

# Laporan Modul 2: Laravel Fundamentasl

---

**Mata Kuliah:** Workshop Web Lanjut

**Nama:** Kaisan Nurwaled

**NIM:** 2024573010044 **Kelas:** TI-2C

---

## Abstrak

Tuliskan ringkasan singkat tentang isi laporan ini dan tujuan Anda membuat laporan.

Jadi laporan kali ini membahas Route, Controller, dan Blade View, jadi saya lebih paham akan hal-hal yang dibahas itu,, dan juga kami disuruh buat proyek baru, yaitu membuat kalkulator sederhana, dan alhamdulillah berhasil. Setelah itu saya membuat prak 3 yang disuruh mencari tau lebih dalam tentang MVC, Route, Controller, dll. Tujuan dibuatnya laporan ini pastinya untuk mengenal lebih dalam sehingga lebih memahami tentang apa yang dipelajari, bukan hanya sekedar mengenal tetapi lebih mendalami sehingga lebih memahami.

---

## 1. Dasar Teori

- Apa itu MVC (Model, View, Controller).

MVC itu cara kerja aplikasi yang dibagi jadi tiga bagian: Model: mengurus data (misalnya ambil, simpan, hapus). View: menampilkan tampilan ke pengguna. Controller: penghubung antara data (Model) dan tampilan (View).

- Konsep Routing di Laravel.

Routing adalah jalan atau peta yang menghubungkan alamat web (URL) ke apa yang harus dijalankan di aplikasi. Contoh: saat buka /calculator, Laravel akan tahu harus menampilkan halaman kalkulator.

- Fungsi Middleware.

Middleware itu seperti "penjaga pintu". Sebelum masuk ke halaman tertentu, request akan dicek dulu. Misalnya: Apakah sudah login? Apakah punya izin khusus?

- Bagaimana cara Laravel menangani Request dan Response.

Saat pengguna buka halaman, Laravel terima request. Laravel cek aturan (middleware, routing). Data diambil/diolah oleh controller & model. Hasilnya dikirim balik ke pengguna dalam bentuk response (misalnya halaman HTML).

- Peran Controller dan View.

Controller: otaknya aplikasi, mengatur logika dan jalan cerita. View: wajah aplikasi, yaitu tampilan yang dilihat pengguna.

- Fungsi Blade Templating Engine.

Blade itu alat untuk bikin tampilan di Laravel supaya lebih rapi dan mudah. Dengan Blade: Bisa pakai template yang sama di banyak halaman. Bisa pakai @if, @foreach untuk logika sederhana di tampilan. Tulisannya lebih singkat dibanding PHP biasa

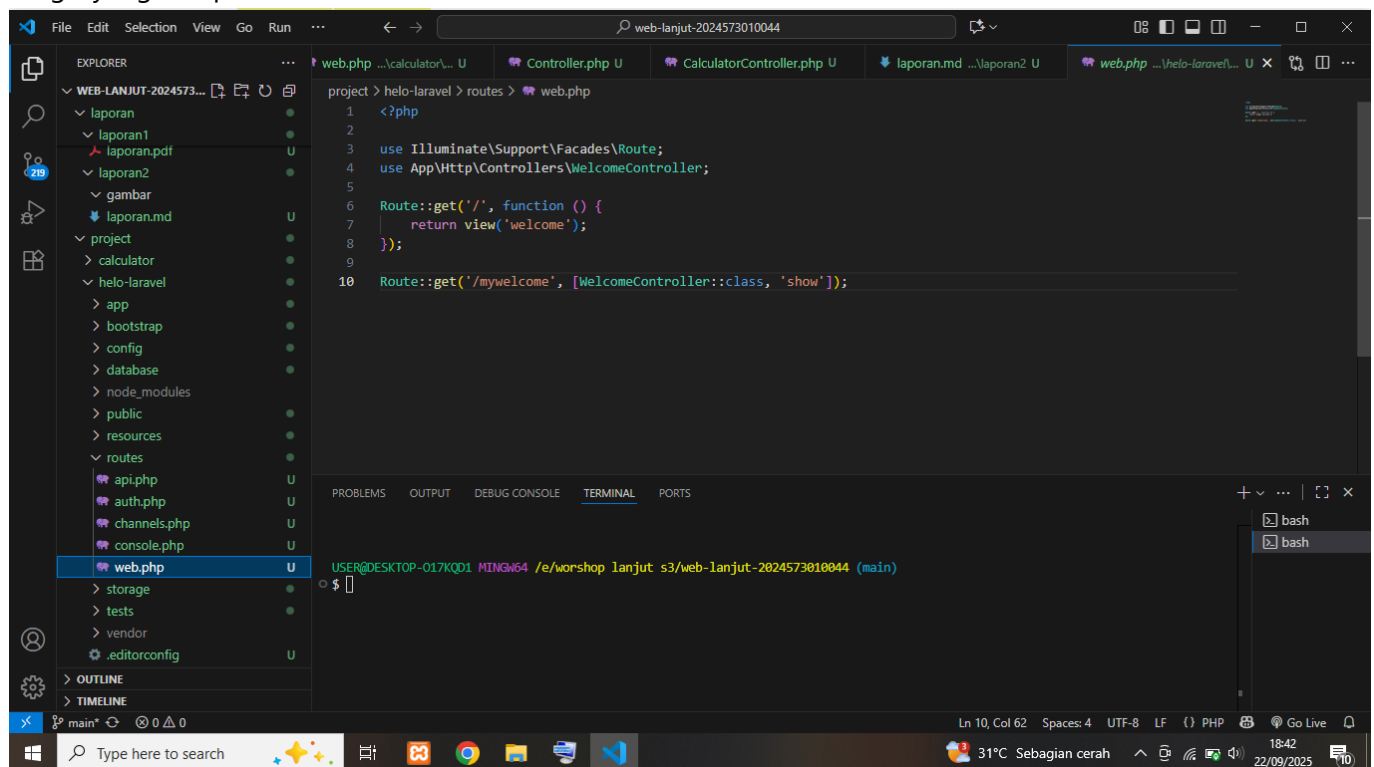
## 2. Langkah-Langkah Praktikum

Tuliskan langkah-langkah yang sudah dilakukan, sertakan potongan kode dan screenshot hasil.

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

- Tambahkan route pada routes/web.php.

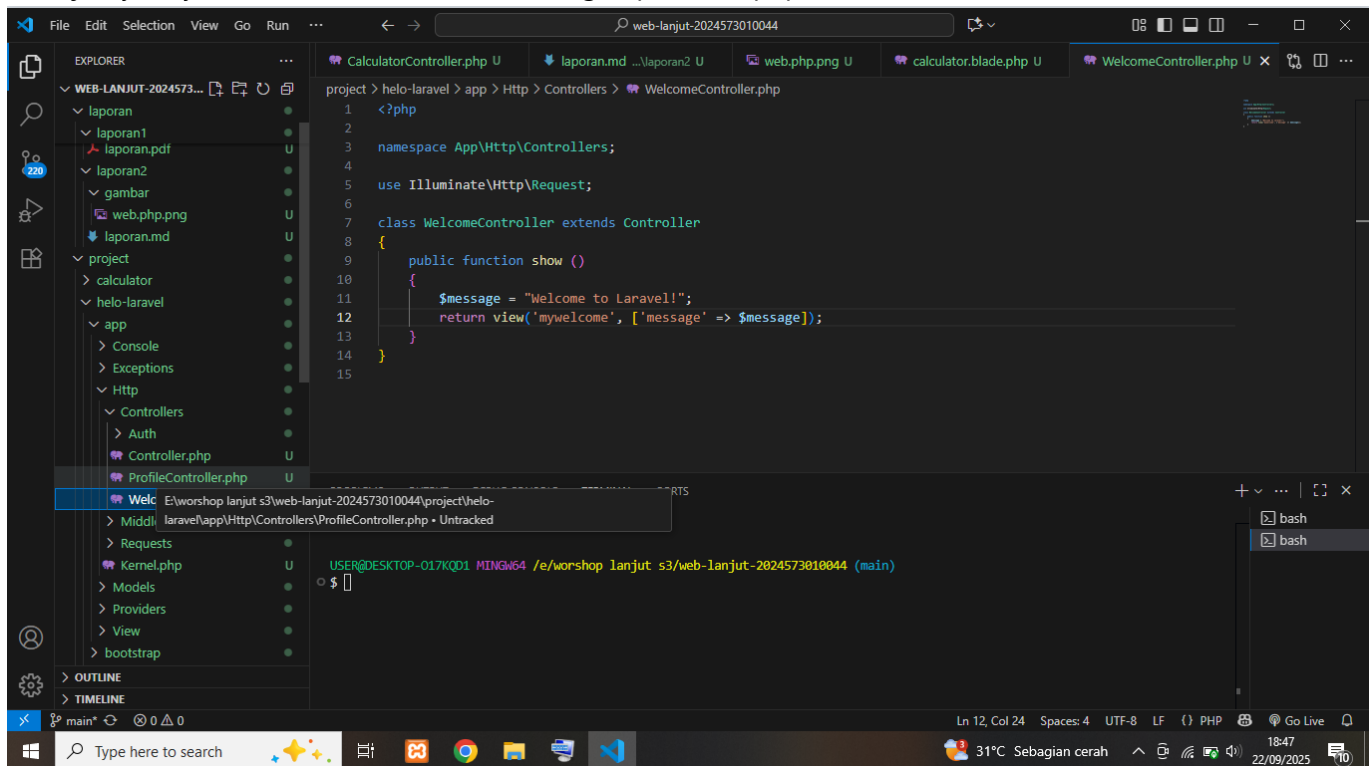
Pertama-tama, menambahkan routes baru pada web.php, ini berfungsi untuk menghubungkan URL dengan fungsi yang ada pada controller.



```
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']);
```

- Buat controller WelcomeController.

Selanjutnya saya membuat controller baru dengan perintah: `php artisan make:controller WelcomeController`

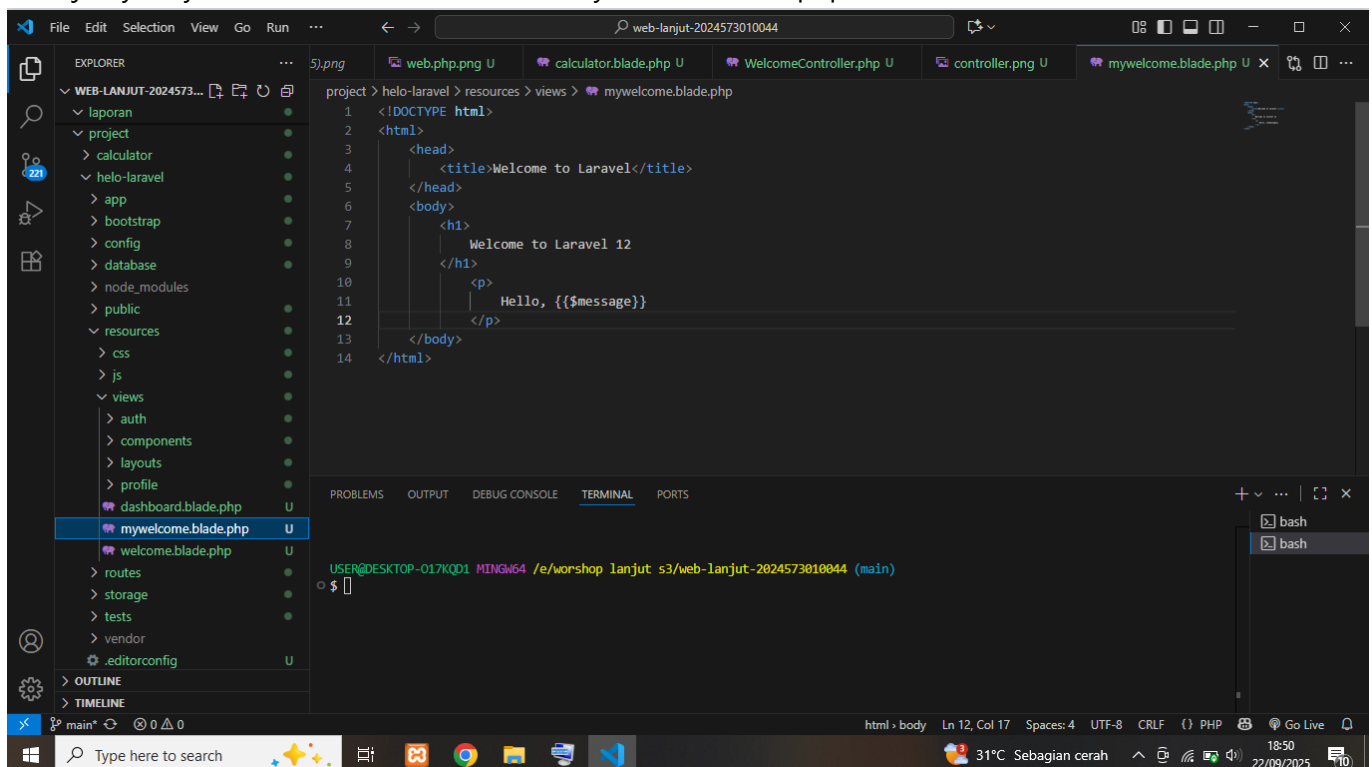


The screenshot shows the Visual Studio Code interface with the Explorer panel on the left displaying the project structure. The main editor shows the newly created `WelcomeController.php` file in the `app/Http/Controllers` directory. The code defines a `WelcomeController` class extending `Controller` with a `show()` method that returns a view named `mywelcome` with a message. The terminal at the bottom shows the command `php artisan make:controller WelcomeController` being executed successfully.

```
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
```

- Buat view `mywelcome.blade.php`.

Selanjutnya saya membuat file view bernama `mywelcome.blade.php` di folder `resources/views`.

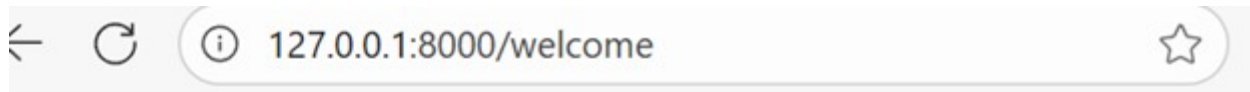


The screenshot shows the Visual Studio Code interface with the Explorer panel on the left displaying the project structure. The main editor shows the newly created `mywelcome.blade.php` file in the `resources/views` directory. The code is a basic HTML template with a title `Welcome to Laravel` and a paragraph `Hello, {{ $message }}`. The terminal at the bottom shows the command `php artisan make:controller WelcomeController` being executed successfully.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Welcome to Laravel</title>
5 </head>
6 <body>
7     <h1>
8         Welcome to Laravel 12
9     </h1>
10     <p>
11         Hello, {{ $message }}
12     </p>
13 </body>
14 </html>
```

- Jalankan aplikasi dan tunjukkan hasil di browser.

Screenshot Hasil:



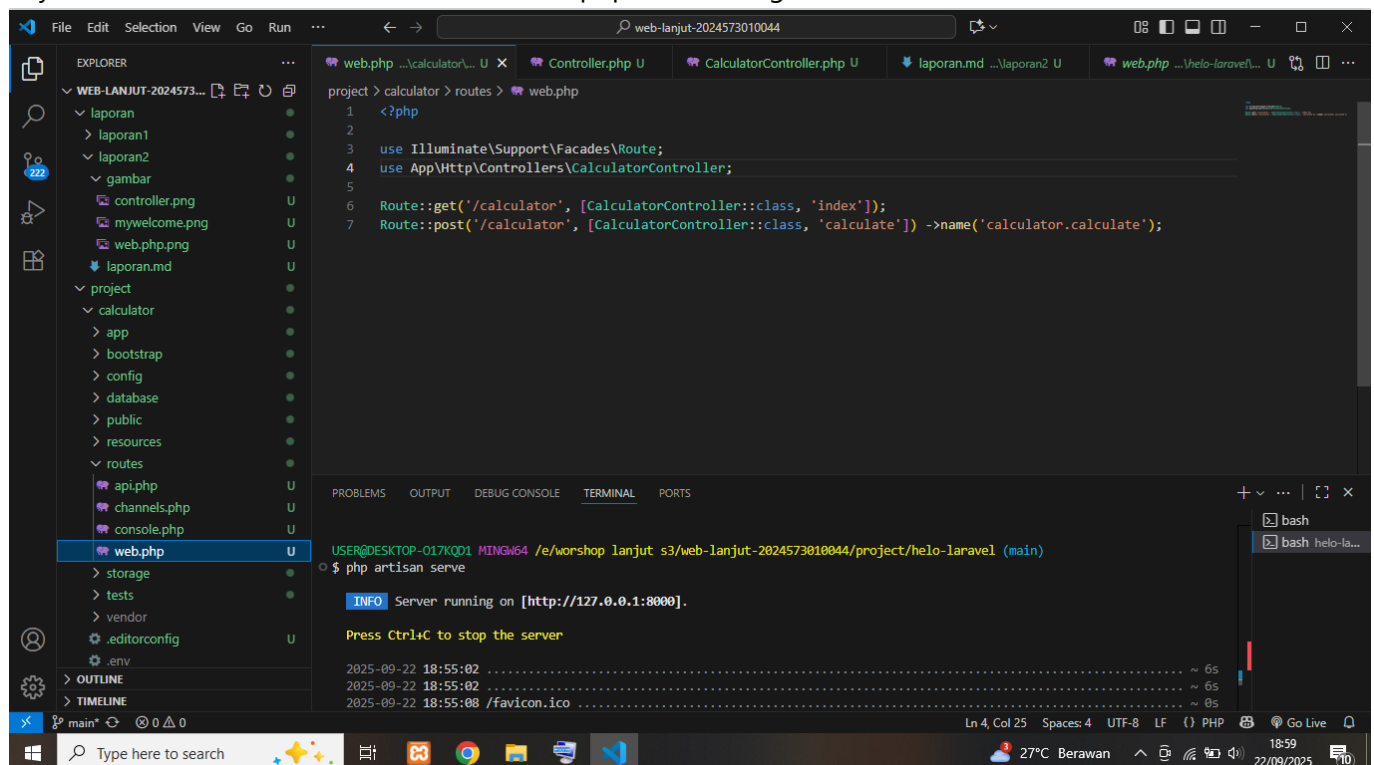
# Welcome to Laravel!

Hello, Welcome to laravel!

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

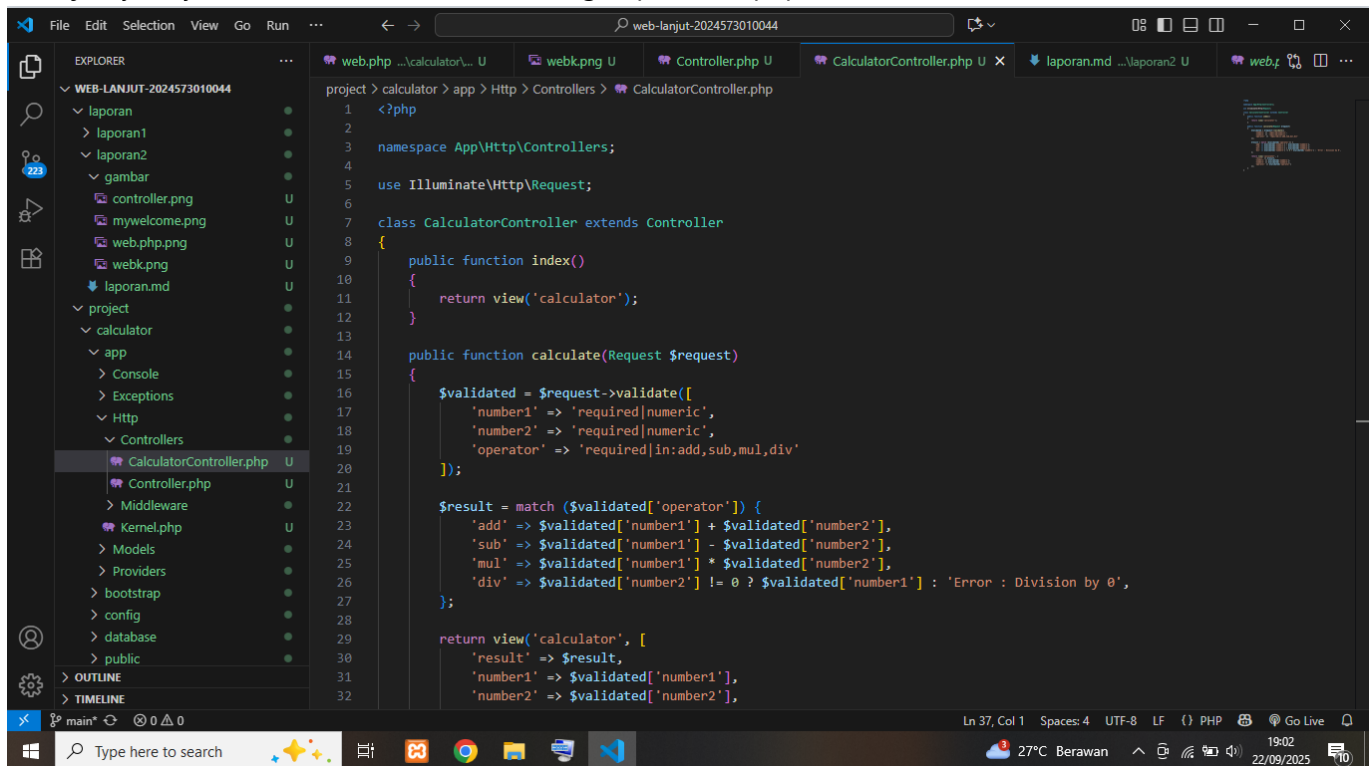
- Tambahkan route untuk kalkulator.

Saya menambahkan route baru di routes/web.php untuk mengakses kalkulator:

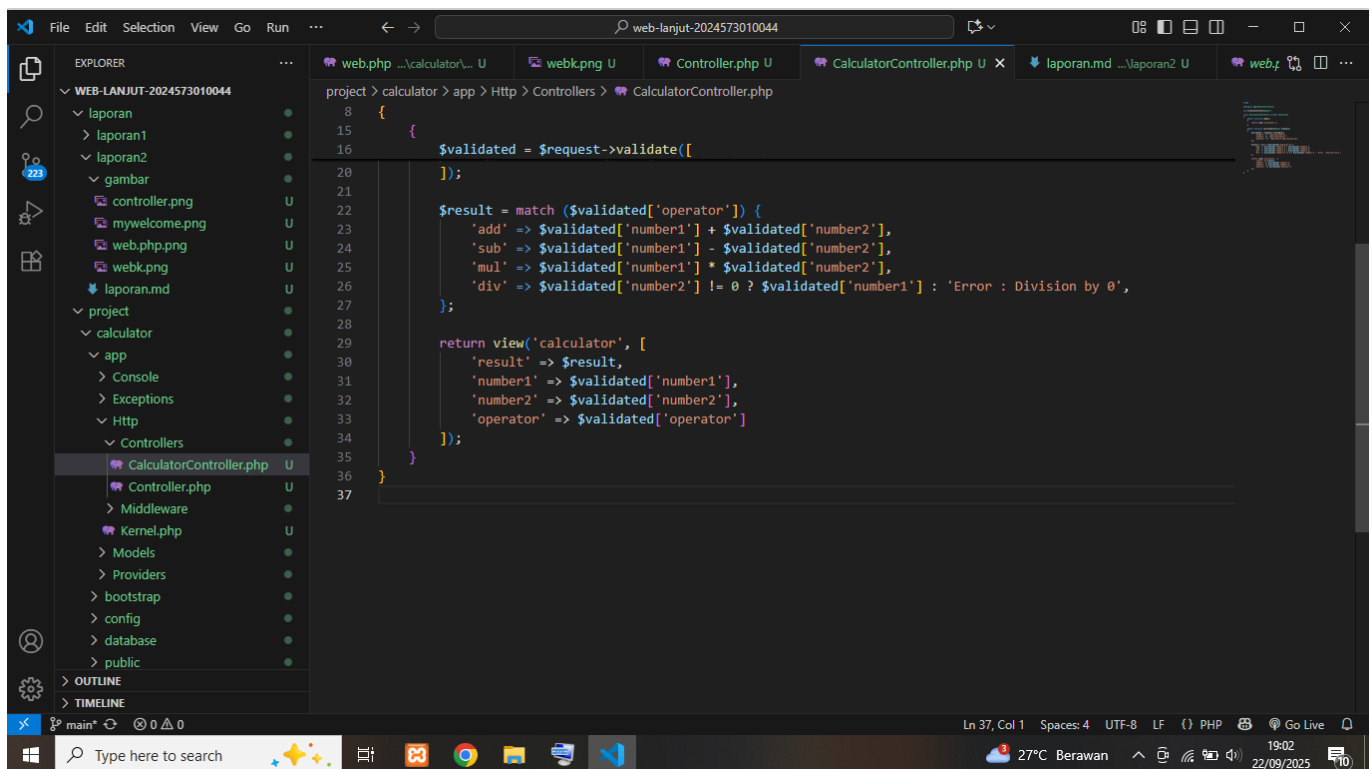


- Buat controller CalculatorController.

Selanjutnya saya membuat controller baru dengan perintah: `php artisan make:controller WelcomeController`



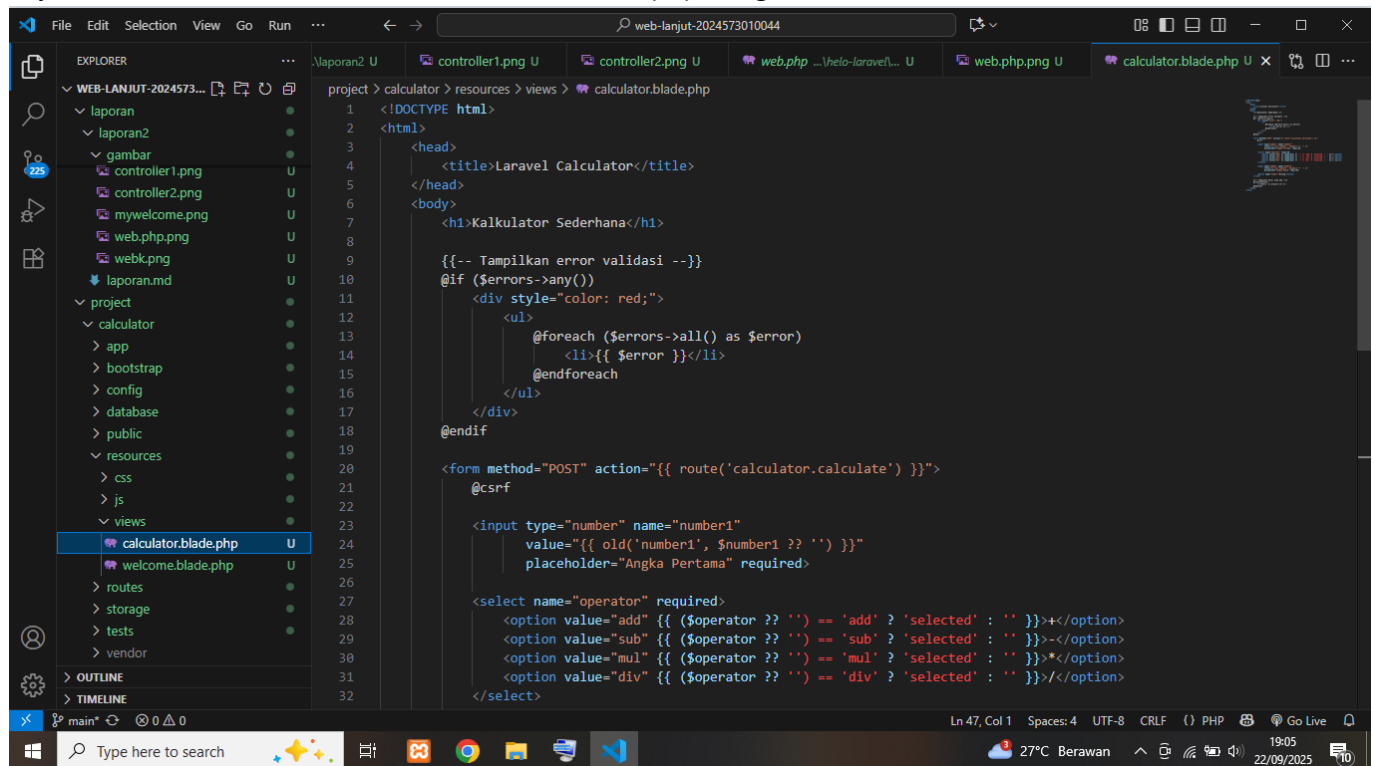
```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class CalculatorController extends Controller
8 {
9     public function index()
10     {
11         return view('calculator');
12     }
13
14     public function calculate(Request $request)
15     {
16         $validated = $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['number2'] != 0 ? $validated['number1'] : 'Error : Division by 0',
27         };
28
29         return view('calculator', [
30             'result' => $result,
31             'number1' => $validated['number1'],
32             'number2' => $validated['number2'],
33         ]);
34     }
35 }
```



```
8 {
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['number2'] != 0 ? $validated['number1'] : 'Error : Division by 0',
27     };
28
29     return view('calculator', [
30         'result' => $result,
31         'number1' => $validated['number1'],
32         'number2' => $validated['number2'],
33         'operator' => $validated['operator']
34     ]);
35 }
36 }
```

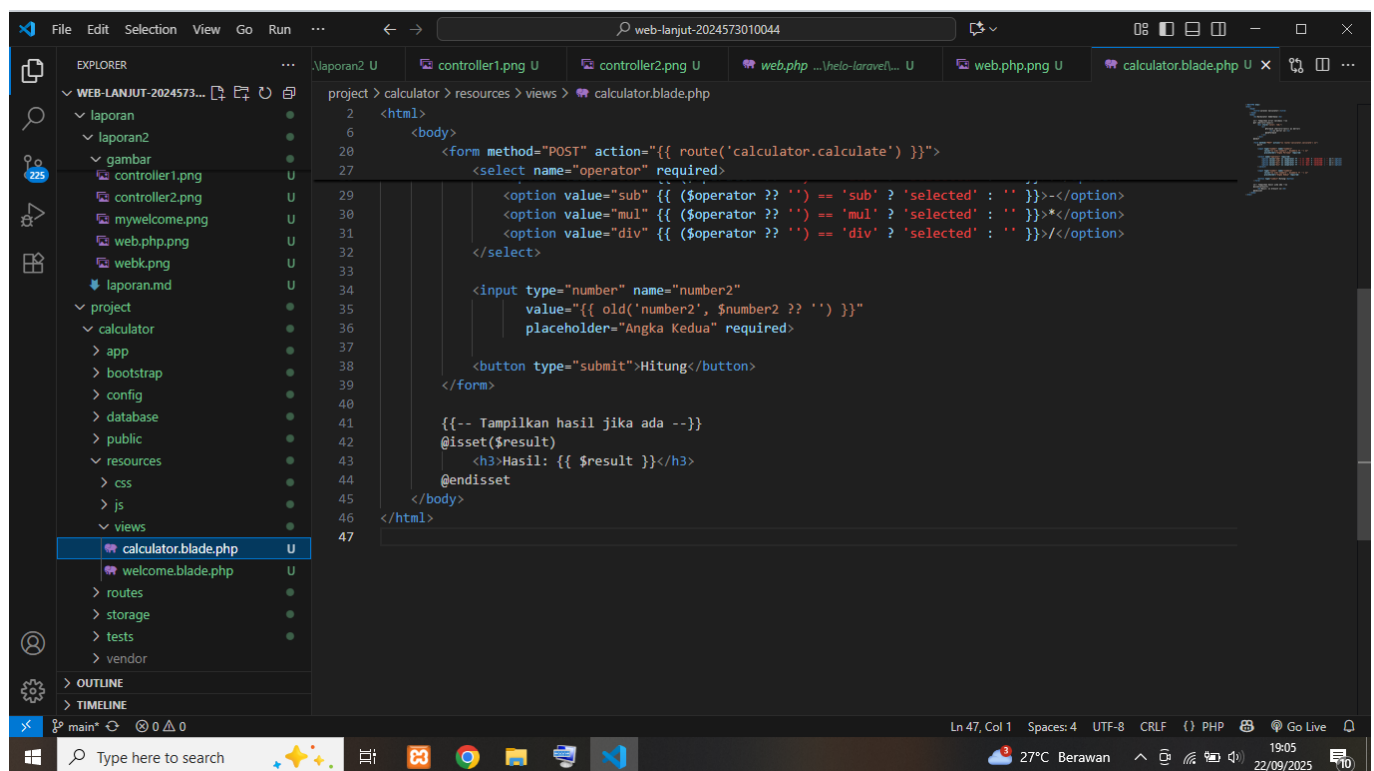
- Tambahkan view calculator.blade.php.

Saya membuat file resources/views/calculator.blade.php dengan isi:



The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows the project structure: laporan2 > gambar > controller1.png, controller2.png, mywelcome.png, web.php.png, webk.png, laporan.md. The project folder contains calculator, app, bootstrap, config, database, public, resources, css, js, and views. The views folder contains calculator.blade.php, welcome.blade.php, routes, storage, tests, and vendor. The calculator.blade.php file is open in the editor, showing the following code:

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



The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows the project structure: laporan2 > gambar > controller1.png, controller2.png, mywelcome.png, web.php.png, webk.png, laporan.md. The project folder contains calculator, app, bootstrap, config, database, public, resources, css, js, and views. The views folder contains calculator.blade.php, welcome.blade.php, routes, storage, tests, and vendor. The calculator.blade.php file is open in the editor, showing the following code:

```
2 <html>
3
4 <body>
5
6 <form method="POST" action="{{ route('calculator.calculate') }}">
7 <select name="operator" required>
8 <option value="add" {{ ($operator ?? '') == 'add' ? 'selected' : '' }}>+</option>
9 <option value="sub" {{ ($operator ?? '') == 'sub' ? 'selected' : '' }}>-</option>
10 <option value="mul" {{ ($operator ?? '') == 'mul' ? 'selected' : '' }}>*</option>
11 <option value="div" {{ ($operator ?? '') == 'div' ? 'selected' : '' }}>/</option>
12 </select>
13
14 <input type="number" name="number2"
15 value="{{ old('number2', $number2 ?? '') }}"
16 placeholder="Angka Kedua" required>
17
18 <button type="submit">Hitung</button>
19 </form>
20
21 {{-- Tampilkan hasil jika ada --}}
22 @isset($result)
23 <h3>Hasil: {{ $result }}</h3>
24 @endisset
25 </body>
26 </html>
```

- Jalankan aplikasi dan coba dengan beberapa input berbeda.

Saya menjalankan aplikasi dengan perintah: cd project, cd calculator, php artisan serve.

Screenshot Hasil:



## Kalkulator Sederhana

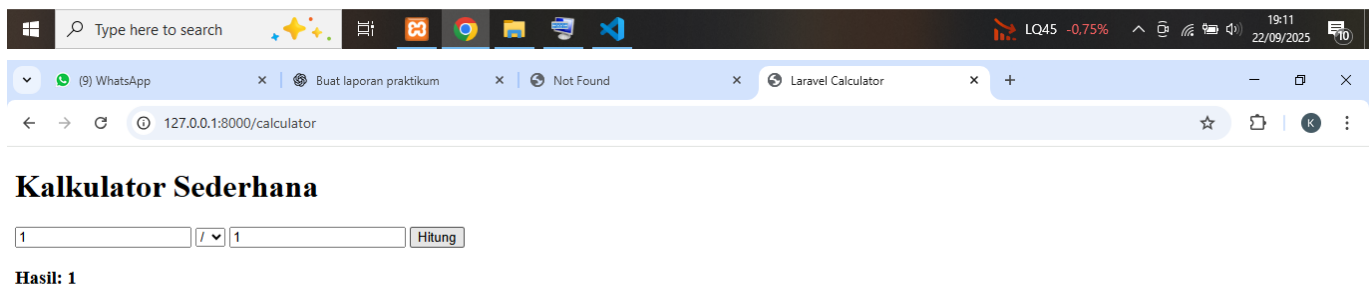
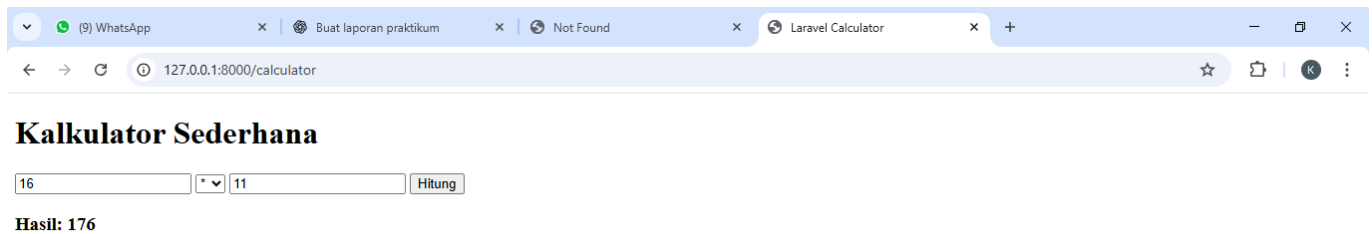
Hasil: 27



## Kalkulator Sederhana

Hasil: 5





### 3. Hasil dan Pembahasan

Jelaskan apa hasil dari praktikum yang dilakukan.

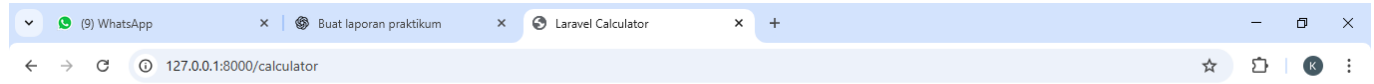
- Apakah aplikasi berjalan sesuai harapan?

Aplikasi kalkulator sederhana berhasil dijalankan. Ketika dimasukkan angka dan memilih operasi hitung, hasil perhitungan dapat ditampilkan di browser mau itu tambah(+), kurang(-), kali(\*), bagi(/).

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



Misalnya dibagi dengan angka 0, maka aplikasi menampilkan pesan Error agar tidak terjadi kesalahan perhitungan.



## Kalkulator Sederhana

1 / 0 Hitung

Hasil: Error : Division by 0



- Bagaimana validasi input bekerja di Laravel?

Validasi input dapat dilakukan dengan Request atau fungsi `validate()` untuk memastikan data yang masuk sesuai aturan, misalnya angka harus diisi dan tidak boleh kosong. Dengan validasi ini, aplikasi lebih aman dari input yang tidak sesuai.

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

Route: mengatur alamat URL yang mengarah ke fungsi tertentu. Controller: tempat logika program dijalankan (proses perhitungan). View: menampilkan hasil ke pengguna dalam bentuk halaman web.

---

## 4. Kesimpulan

Tulis kesimpulan dari praktikum ini

Dari praktikum ini dapat disimpulkan bahwa Laravel mempermudah pembuatan aplikasi web dengan konsep MVC (Model-View-Controller). Route digunakan untuk mengatur alur halaman, Controller berfungsi mengolah logika aplikasi, dan View menampilkan hasil ke pengguna. Aplikasi kalkulator sederhana yang dibuat sudah berjalan sesuai harapan, termasuk menangani input yang tidak valid seperti pembagian dengan nol.

---

## 5. Referensi

Cantumkan sumber yang Anda baca (buku, artikel, dokumentasi) — minimal 2 sumber. Gunakan format sederhana (judul — URL).

Laravel Documentation — <https://laravel.com/docs> Tutorialspoint: Laravel —

<https://www.tutorialspoint.com/laravel> W3Schools Laravel Tutorial — <https://www.w3schools.com/laravel>

