

# Laporan Modul 2: Laravel Fundamentasl

---

**Mata Kuliah:** Workshop Web Lanjut

**Nama:** [Khalisha Zafira]

**NIM:** [2024573010040]

**Kelas:** [TI\_2C]

---

## Abstrak

Tuliskan ringkasan singkat tentang isi laporan ini dan tujuan Anda membuat laporan. [ Laporan ini berisi penjelasan mengenai pembuatan aplikasi kalkulator sederhana menggunakan framework Laravel. Dalam aplikasi ini, user dapat melakukan operasi aritmatika dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian. Proses perhitungan dilakukan melalui alur Route → Controller → View, sedangkan hasilnya ditampilkan pada halaman antarmuka berbasis Blade. Validasi input juga diterapkan untuk memastikan data yang dimasukkan sesuai, misalnya mencegah kesalahan seperti pembagian dengan nol.

Tujuan Pembuatan Laporan Tujuan dari laporan ini adalah:

- Mendokumentasikan proses pembuatan aplikasi kalkulator sederhana.
  - Menunjukkan bagaimana Laravel menangani request, response, validasi input, serta pengolahan data.
  - Memberikan gambaran penerapan konsep dasar MVC (Model-View-Controller) dalam sebuah program sederhana.]
- 

## 1. Dasar Teori

- Apa itu MVC (Model, View, Controller). [ MVC adalah singkatan dari Model-View-Controller, sebuah pola arsitektur perangkat lunak yang digunakan untuk memisahkan concern dalam aplikasi web

Model: Merepresentasikan data dan logika bisnis. Dalam Laravel, biasanya ditangani oleh Eloquent models yang berinteraksi dengan database

View: Menangani lapisan presentasi- yaitu HTML yang ditampilkan kepada pengguna. Laravel menggunakan Blade sebagai templatr engine untuk view

Controller: Mengatur alur anatar model dan view. Controller menerima input dari pengguna (melalui route) memprosesnya (seringkali melibatkan model),dan mengembalikan view atau respons yang sesuai]

- Konsep Routing di Laravel. [ Routing di Laravel mendefinisikan bagaimana aplikasi merespons permintaan yang masuk. Routing memetakan URI ke aksi controller atau closure.Route di Laravel didefinisikan di file routes/web.php]
- Fungsi Middleware. [ Middleware berfungsi sebagai perantara antara request dan response. Dia bisa memeriksa, memfilter, atau memodifikasi request sebelum masuk ke controller, dan juga response sebelum dikirim ke user (contoh: autentikasi, logging, CSRF protection). ]

- Bagaimana cara Laravel menangani Request dan Response. [ - Request dari user masuk ke public/index.php.
    - Lalu diteruskan ke Kernel, di situ dijalankan middleware.
    - Setelah lolos middleware, request diarahkan ke route → controller → logic aplikasi.
    - Hasil dari controller dikembalikan sebagai response yang bisa berupa HTML, JSON, dll, lalu dikirim ke browser.]
  - Peran Controller dan View. [ Controller: mengatur logika aplikasi, menerima request, memproses data (misalnya lewat model), lalu menentukan apa yang harus ditampilkan.  
  
View: menampilkan data dalam bentuk antarmuka (UI) untuk user, biasanya berisi HTML + Blade syntax.  
]
  - Fungsi Blade Templating Engine. [ Blade memudahkan pembuatan view di Laravel dengan:
    - Mendukung template inheritance (layout dasar bisa diwarisi).
    - Sintaks sederhana (@if, @foreach, dll).
    - Bisa menyisipkan variabel PHP dengan aman. → Intinya mempermudah pemisahan logika dan tampilan.]
- 

## 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.

```
project > helo-laravel > routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\WelcomeController;
5
6
7  Route::get('/', function () {
8      return view('welcome');
9  });
10
11 Route::get('/welcome', [WelcomeController::class, 'show']);
12
```

- Buat controller WelcomeController.

```
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 }
```

- Buat view mywelcome.blade.php.

```
project > helo-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.

Screenshot Hasil:



# Welcome to laravel 12

Hello, Welcome to Laravel!

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

- Tambahkan route untuk kalkulator.

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

- Buat controller CalculatorController.

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

```

27
28     return view('calculator', [
29         'result' => $result,
30         'number1' => $validated['number1'],
31         'number2' => $validated['number2'],
32         'operator' => $validated['operator']
33     ]);
34
35 }

```

- Tambahkan view calculator.blade.php.

```

project > calculator > resources > views > calculator.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <title>Laravel Calculator</title>
6  </head>
7
8  <body>
9      <h1>Kalkulator Sederhana</h1>
10
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          <input type="number" name="number1" value="{{ old('number1', $number1 ?? '') }}" placeholder="Angka P
24
25          <select name="operator" required>
26              <option value="add" {{ ($operator ?? '') == 'add' ? 'selected' : '' }}>+</option>
27              <option value="sub" {{ ($operator ?? '') == 'sub' ? 'selected' : '' }}>-</option>
28              <option value="mul" {{ ($operator ?? '') == 'mul' ? 'selected' : '' }}>*</option>
29              <option value="div" {{ ($operator ?? '') == 'div' ? 'selected' : '' }}>/</option>
30
31          </select>
32
33          <input type="number" name="number2" value="{{ old('number2', $number2 ?? '') }}" placeholder="Angka K
34          <button type="submit">Hitung</button>
35      </form>
36
37      @isset($result)
38          <h3>Hasil: {{ $result }}</h3>
39      @endisset
40
41  </body>
42 </html>

```

- Jalankan aplikasi dan coba dengan beberapa input berbeda. Screenshot Hasil:

- Calculator Penambahan:

← → ↻

⛔ http://127.0.0.1:8000/calculator

## Kalkulator Sederhana

12

+

56

Hitung

**Hasil: 68**

- Calculator Pengurangan:

← → ↻

⛔ http://127.0.0.1:8000/calculator

## Kalkulator Sederhana

14

-

5

Hitung

**Hasil: 9**

- Calculator Perkalian:

← → ↻

⛔ http://127.0.0.1:8000/calculator

## Kalkulator Sederhana

14

\*

5

Hitung

**Hasil: 70**

- Calculator Pembagian:



# Kalkulator Sederhana

14 / 2 Hitung

**Hasil: 7**

---

## 3. Hasil dan Pembahasan

Jelaskan apa hasil dari praktikum yang dilakukan.

- Apakah aplikasi berjalan sesuai harapan? [ Aplikasi berjalan sesuai harapan jika alur Route → Controller → View sudah benar, logika di Controller sesuai kebutuhan, dan data dari user divalidasi dengan benar sebelum diproses.

Laravel sendiri membantu memastikan alur ini konsisten dengan sistem routing, middleware, validasi, dan Blade untuk tampilan.]

- Apa yang terjadi jika ada input yang salah (misalnya pembagian dengan 0)? [ Kalau tidak ada validasi atau error handling, maka program akan menghasilkan error / exception (contoh: Division by zero error).

Laravel punya exception handler, jadi error biasanya ditampilkan dalam bentuk halaman error (di mode development) atau pesan generik (di mode production)]

- Bagaimana validasi input bekerja di Laravel? [ Laravel punya sistem Validator untuk memastikan data yang masuk sesuai aturan. Validasi bisa dilakukan: Langsung di Controller dengan `$request->validate([...])`. Atau menggunakan Form Request Class khusus. Jika input tidak valid → otomatis redirect kembali dengan pesan error. Jika valid → data diteruskan ke proses selanjutnya (misalnya simpan ke database).]
- Apa peran masing-masing komponen (Route, Controller, View) dalam program yang dibuat? [- Route: Menentukan URL mana yang akan memanggil controller atau action tertentu. → ibarat peta jalan dari request ke logic.

Controller: Mengatur logika aplikasi: menerima request dari Route, mengolah data (bisa lewat Model), lalu menentukan View atau Response apa yang diberikan ke user.

View: Bertugas menampilkan hasil ke user, biasanya berupa HTML + Blade. → Jadi user hanya melihat tampilan, tanpa tahu logika di belakangnya.]

---

## 4. Kesimpulan

Tulis kesimpulan dari praktikum ini [ Dari praktikum ini dapat disimpulkan bahwa framework Laravel mampu mempermudah pembuatan aplikasi berbasis web dengan konsep MVC (Model-View-Controller). Penerapan Route, Controller, dan View membuat alur program menjadi lebih terstruktur, sedangkan penggunaan Blade templating engine membantu dalam penyusunan tampilan.

Selain itu, penerapan validasi input sangat penting untuk mencegah error, seperti kasus pembagian dengan nol. Melalui praktikum kalkulator sederhana ini, pemahaman mengenai cara Laravel menangani request, response, serta error handling dapat diperoleh dengan lebih nyata dan aplikatif.]

---

## 5. Referensi

Cantumkan sumber yang Anda baca (buku, artikel, dokumentasi) — minimal 2 sumber. Gunakan format sederhana (judul — URL). [ Laravel Documentation — Validation <https://laravel.com/docs/10.x/validation>

Laravel Documentation — Routing <https://laravel.com/docs/10.x/routing>

Welling, J. & Thomson, L. — PHP and MySQL Web Development (5th Edition)]

---