

Laporan Modul 2: Laravel Fundamentals

Mata Kuliah: Workshop Web Lanjut

Nama: Ilham Syahdienar

NIM: 2024573010029 **Kelas:** TI-2C

Abstrak

Laporan ini berisi pembelajaran tentang MVC, yaitu Model, Views, Controller. dilaporan ini juga berisi 2 percobaan, percobaan menampilkan teks dengan routes yang dibuat sendiri, dan membuat kalkulator sederhana.

1. Dasar Teori

- Apa itu MVC?
 - MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dan presentasi. Dengan komponen Model, View, Controller kita bisa menangani pengembangan aplikasi untuk aspek-aspek tertentu
- Konsep Routing di Laravel
 - Routing di Laravel yaitu mekanisme yang menghubungkan URL atau alamat pada browser dengan logika atau fungsi tertentu di aplikasi. Istilah Sederhana dari routing yaitu seperti peta yang mengarahkan request ke tempat yang benar di aplikasi anda.
- Fungsi Middleware
 - keamanan = dapat melindungi data sensitif dan mengontrol hak akses pengguna.
 - Pemfilteran Data = untuk memanipulasi data permintaan sebelum sebuah action dalam controller
 - logging dan audit = digunakan untuk mencatat aktivitas pengguna atau melakukan audit terhadap permintaan yang masuk.
- Bagaimana cara Laravel menangani Request dan Response.
 - Ketika user mengakses sebuah URL, Laravel akan menangani request tersebut melalui serangkaian proses yang disebut lifecycle. Proses ini dimulai dari HTTP kernel, yang bertanggung jawab menginisialisasi aplikasi, hingga akhirnya menghasilkan response yang dikirimkan kembali ke browser.
 - Middleware: Bagian ini mengelola request sebelum mencapai logika utama aplikasi, seperti memeriksa autentikasi atau filtering data.
 - Routing: Setelah request melewati middleware, Laravel akan mencocokkan URL yang diminta dengan route yang telah didefinisikan.
 - Controller dan Views: Laravel kemudian memproses logika di controller, mengakses data dari model jika diperlukan, lalu mengirimkan hasilnya melalui view kepada browser.
- Peran Controller dan View
 - peran Controller ada sebagai tempat memproses atau mengeksekusi logika yang diperlukan sedangkan View sebagai tempat untuk menampilkan hasil yang sudah di eksekusi oleh controller kepada user.
- Fungsi Blade Templating Engine

- berfungsi untuk menyederhanakan proses pembuatan tampilan (view) web yang dinamis, rapi, dan aman, dengan cara memisahkan logika PHP dari HTML melalui sintaks yang singkat dan fitur fitur otomatis yang membantu produktivitas pengembang.

2. Langkah-Langkah Praktikum

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

- 1. Buka project yang telah dibuat pada pertemuan sebelumnya menggunakan vscode.
- 2. Cari file web.php pada folder routes, lalu edit kode program di dalam file web.php seperti gambar di bawah ini:

```
web-lanjut-2024573010029 > projects > helo-laravel > 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('/welcome', [WelcomeController::class, 'show']);
```

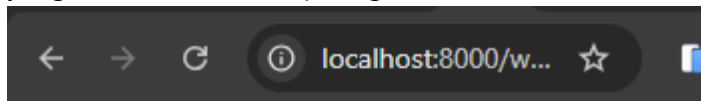
- 3. Selanjutnya kita buat controller di terminal/git bash, untuk membuat controller kita dapat gunakan perintah: `php artisan make:controller WelcomeController`
- 4. Setelah itu buka file controller yang telah dibuat, lalu edit kode program di file tersebut seperti gambar dibawah ini:

```
ut-2024573010029 > projects > helo-laravel > app > Http > Controllers > WelcomeCo
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 }
```

- 5. Di `resources/views` buat file baru dengan nama `mywelcome.blade.php`, lalu isi file tersebut dengan kode program seperti gambar di bawah ini:

```
web-lanjut-2024573010029 > projects > helo-laravel > resources > views > my
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>
```

- 6. Jalankan project laravel yang kita buat dengan mengetik `php artisan serve` di terminal atau git bash
- 7. Buka browser lalu ketik `http://localhost:8000/welcome`, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:



Welcome to Laravel 12

Hello, Welcome to Laravel!

• 2.2 Praktikum 2 – Membuat Aplikasi Sederhana "Calculator"

- 1. Buat sebuah project baru di dalam folder `projects` dan beri nama project baru itu adalah `calculator`
- 2. Cari file `web.php` pada folder `routes`, lalu edit kode program di dalam file `web.php` seperti gambar di bawah ini:

```
web-lanjut-2024573010029 > projects > calculator > 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');
```

- 3. Selanjutnya kita buat controller di terminal/git bash, untuk membuat controller kita dapat gunakan perintah: `php artisan make:controller CalculatorController`
- 4. Lalu pada file controller yang sudah dibuat, edit kode program seperti gambar dibawah ini:

```

web-lanjut-2024573010029 > projects > calculator > app > Http > Controllers > CalculatorController.php > ...
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'] / $validated['number2']) : '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 }
37

```

- 5. Jangan lupa untuk membuat file baru di folder `resources/views` dengan nama `calculator.blade.php`, dan isi file tersebut dengan kode program pada gambar dibawah ini:

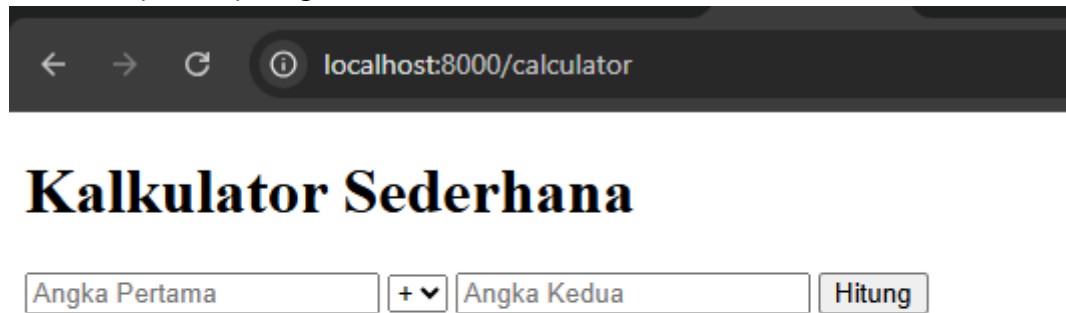
```

web-lanjut-2024573010029 > projects > calculator > resources > views > calculator.blade.php > html
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', $number1 ?? '') }}" placeholder="Angka Pertama" required>
22
23         <select name="operator" required>
24             <option value="add" {{ ($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', $number2 ?? '') }}" placeholder="Angka Kedua" required>
31         <button type="submit">Hitung</button>
32     </form>
33
34     @isset($result)
35         <h3>Hasil: {{ $result }}</h3>
36     @endisset
37 </body>
38 </html>

```

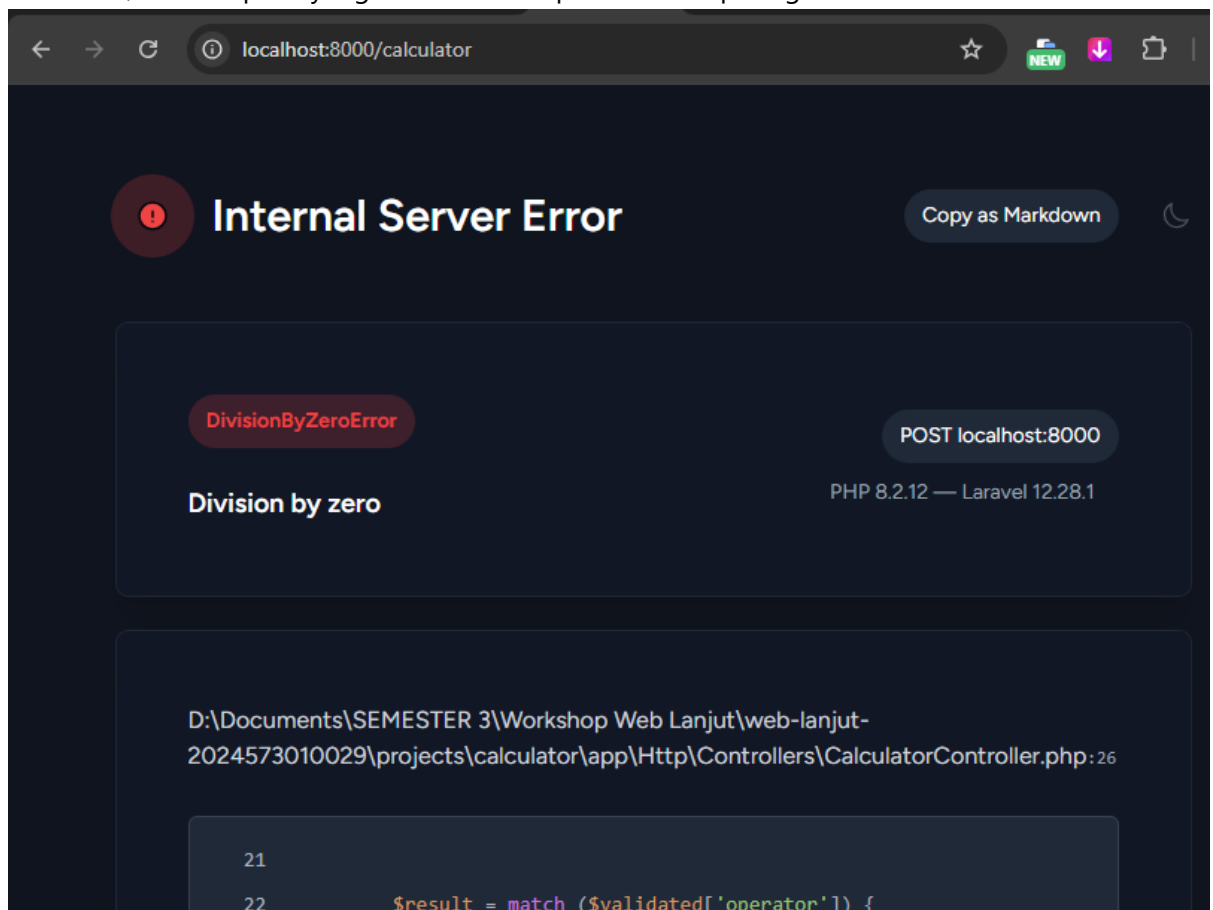
- 6. Setelah melakukan langkah 1 hingga 5, maka pada langkah terakhir yaitu menjalankan server laravel dengan perintah `php artisan serve`, setelah itu buka browser dan akan

keluar tampilan seperti gambar dibawah ini:



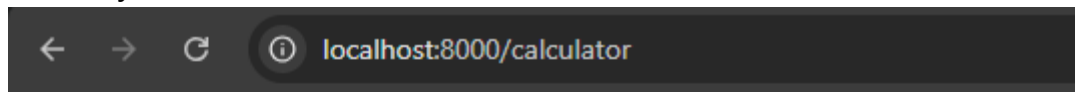
3. Hasil dan Pembahasan

- Apakah aplikasi berjalan sesuai harapan?
 - Alhamdulillah hasil dari praktikum ini sesuai harapan meskipun menjumpai beberapa error yang disebabkan karena terlalu terburu-buru sehingga beberapa kali typo.
- Apa yang terjadi jika ada input yang salah (misalnya pembagian dengan 0)?
 - yang terjadi adalah akan muncul tampilan error yang telah kita atur sesuai dengan logika pada controller, dan tampilan yang telah kita atur pada view. seperti gambar dibawah ini:



- Bagaimana validasi input bekerja di Laravel?

- Validasi yaitu proses memeriksa data yang dikirim dari form atau request agar sesuai dengan aturan (rules) sebelum disimpan atau diproses lebih lanjut. pada praktikum ini, validasi di lakukan di file `CalculatorController`, setelah sesuai dengan aturan maka akan di eksekusi dan diproses.
- Apa peran masing-masing komponen (Route, Controller, View) dalam program yang dibuat?
 - Route = Berperan sebagai tempat untuk pengalamanan yang akan ditampilkan ke user
 - Controller = Beperan sebagai pengatur logika atau validasi yang digunakan saat request dan respon ke user.
 - View = Berperan bekerja sama dengan controller, yaitu saat controller meminta request dari user lalu melakukan validasi atau eksekusi program maka setelah itu akan ditampilkan hasil eksekusi yang dilakukan. dan tugas yang mengatur tampilan atau menyajikan tampilan data tersebut adalah view.
- Hasil Penjumlahan:

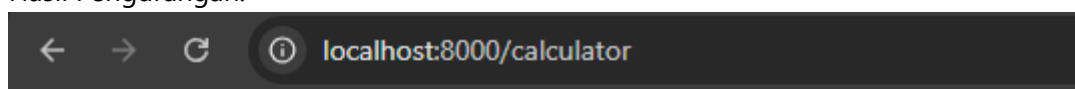


Kalkulator Sederhana

<input type="text" value="20"/>	<input type="button" value="+"/>	<input type="text" value="30"/>	<input type="button" value="Hitung"/>
---------------------------------	----------------------------------	---------------------------------	---------------------------------------

Hasil: 50

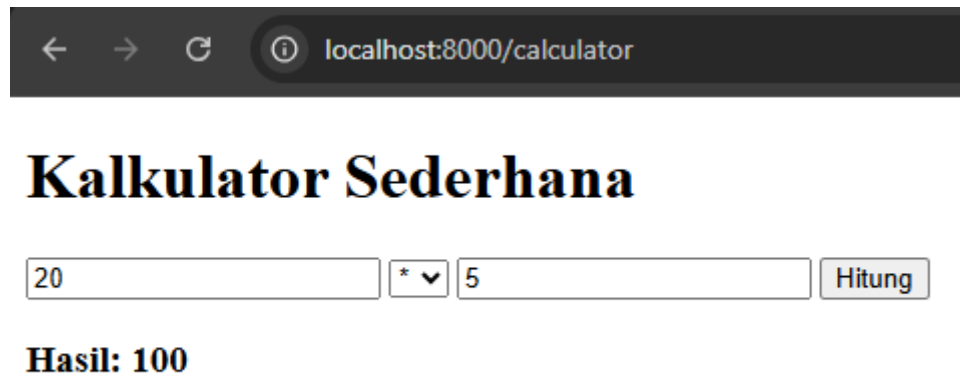
-
- Hasil Pengurangan:



Kalkulator Sederhana

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

Hasil: 10



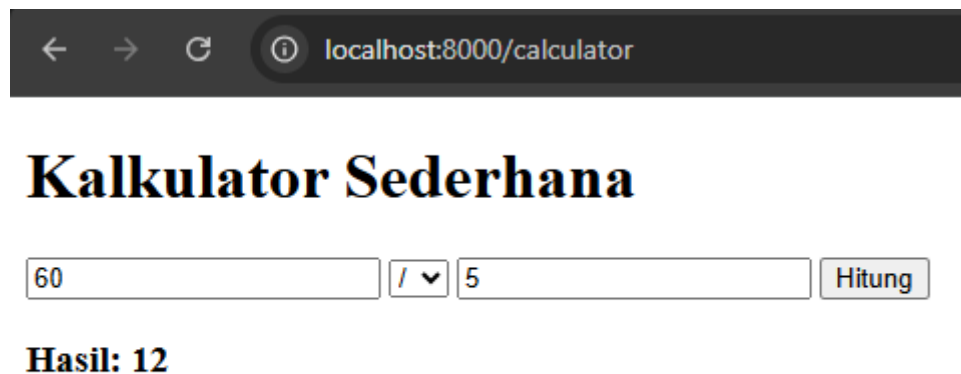
← → ↻ ⓘ localhost:8000/calculator

Kalkulator Sederhana

*

Hasil: 100

- Hasil Perkalian:



← → ↻ ⓘ localhost:8000/calculator

Kalkulator Sederhana

/

Hasil: 12

- Hasil Pembagian:

4. Kesimpulan

- Kesimpulan dari praktikum ini adalah dengan menerapkan MVC pada project yang kita buat itu akan memudahkan kita untuk pengembangan, perbaikan, dan efisiensi yang sangat berguna baik itu dalam hemat tenaga ataupun hemat waktu.

5. Referensi

- <https://codingstudio.id/blog/model-view-controller-adalah/>
- <https://buildwithangga.com/tips/apa-itu-route-pada-framework-laravel>
- <https://buildwithangga.com/tips/apa-itu-laravel-request-lifecycle>
- <https://hackmd.io/@mohdrzu/B1zwKEK5xe>