

Laporan Modul 2: Laravel Fundamentasl

Mata Kuliah: Workshop Web Lanjut

Nama: [Liza Aulia Siregar]

NIM: [2024573010002]

Kelas: [TI.2C]

Abstrak

Laporan ini merupakan hasil praktikum yang memperkenalkan kita pada fondasi utama dalam membangun Larevel. Untuk memahami routing, middleware, penanganan request/response, controller, view, dan Blade templating yang sangat penting untuk mengembangkan dengan Laravel secara efektif.

1. Dasar Teori

MVC (Model–View–Controller) adalah sebuah pola arsitektur perangkat lunak yang digunakan untuk memisahkan concern dalam aplikasi web, yaitu :

- Model : Yang merepresentasikan data dan logika bisnis. Biasanya dalam Laravel ditangani oleh Eloquent models yang berinteraksi dengan database.
- View : Yang menangani lapisan presentasi yaitu HTML yang ditampilkan kepada pengguna. Laravel menggunakan Blade sebagai template engine untuk view.
- Controller : Yang Mengatur alur antara model dan view. Controller menerima input dari pengguna (melalui route), memprosesnya (seringkali melibatkan model), dan mengembalikan view atau respons yang sesuai.
- Routing di Laravel mendefinisikan bagaimana aplikasi merespons permintaan (request) yang masuk. Routing memetakan URI ke aksi controller atau closure.
- Middleware menyediakan mekanisme untuk memfilter permintaan HTTP yang masuk ke aplikasi. Middleware bertindak sebagai lapisan yang berada di antara siklus request dan response.
- Cara Laravel menangani Request dan Response : **Mengakses Data Request** Dalam mengakses data request, dapat mengakses data request menggunakan objek Request

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
6  use Illuminate\Foundation\Bus\DispatchesJobs;
7  use Illuminate\Foundation\Validation\ValidatesRequests;
8  use Illuminate\Routing\Controller as BaseController;
9
10 class Controller extends BaseController
11 {
12     use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
13 }
14 |
```

Pengguna juga dapat dengan mudah mengakses parameter query, data form, dan payload JSON. Dalam controller atau route closure

```
public function calculate(Request $request)
{
    $validated = $request->validate([
        'number1' => 'required|numeric',
        'number2' => 'required|numeric',
        'operator' => 'required|in:add,sub,mul,div'
    ]);
}
```

Form Request Dalam Laravel memungkinkan kita untuk memvalidasi dan mengotorisasi data request yang masuk melalui class form request dengan menjalankan perintah : `php artisan make:request StoreUserRequest`

Mengembalikan Response Laravel juga menyediakan cara sederhana untuk mengembalikan response dengan mengembalikan string, view, JSON atau bahkan file download.

```
return view('calculator', [
    'result' => $result,
    'number1' => $validated['number1'],
    'number2' => $validated['number2'],
    'operator' => $validated['operator']
]);
```

Peran Controller dan View Controller digunakan untuk mengatur logika aplikasi Anda. File controller berada di dalam `app/Http/Controllers`. Di dalam controller bisa mendefinisikan sebuah method untuk menangani route tertentu.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Routing\Controller as BaseController;

class Controller extends BaseController
{
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
}
```

Di file routes/web.php, kita dapat mendaftarkan route agar terhubung ke method controller tersebut:

```
Route::get('/calculator', [CalculatorController::class, 'index']);
```

View disimpan di dalam resources/views.

Fungsi Blade Templating Engine. Blade merupakan template engine yang disediakan oleh laravel. Ini dapat membawa kemudahan, kejelasan, dan kecepatan ke dunia pengembangan web.

2. Langkah-Langkah Praktikum

2.1 Praktikum 1 – Route, Controller, dan Blade View

1. Buka project masing-masing menggunakan vscode
2. Buka file routes/web.php dan edit kode nya menjadi seperti berikut:

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers>WelcomeController;
/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', function () {
    return view('welcome');
});
Route::get('/welcome', [WelcomeController::class, 'show']);
```

Route ini untuk memanggil method show() yang ada di dalam WelcomeController ketika user mengunjungi /welcome. 3. Buka git bash dan jalankan perintah berikut untuk membuat controller

```
user@DESKTOP-MESNAC2 MINGW64 ~/OneDrive/Documents/SEMESTER 3/WEB LANJUT
$ php artisan make:controller WelcomeController
```

4. Buka file controller yang telah dibuat tadi Edit menjadi

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class WelcomeController extends Controller
{
    public function show()
    {
        $message = "Welcome to Laravel!";
        return view('mywelcome', ['message' => $message]);
    }
}
```

5. Buat file baru resources/views/mywelcome.blade.php dan isi dengan kode berikut :

```
<!DOCTYPE html>
<html>
    <head>
        <title>Welcome to Laravel</title>
    </head>
    <body>
        <h1>
            Welcome to Laravel 12
        </h1>
        <p>
            Hello, {{$message}}
        </p>
    </body>
</html>
```

6. Jalankan project laravel dengan perintah

```
user@DESKTOP-MESNAC2 MINGW64 ~/OneDrive/Documents/SEMESTER 3/WEB LANJUT/web-lanjut-2024573010002-/projects/helo-larav
el (main)
$ php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

2025-09-16 15:14:00
```

7. Jalankan aplikasi dengan mengetikkan http:localhost:8000/welcome di browser



2.2 Praktikum 2 – Membuat Aplikasi Sederhana "Calculator"

1. Tambahkan route untuk kalkulator.

2. Buat controller CalculatorController dengan perintah

```
user@DESKTOP-MESNAC2 MINGW64 ~/OneDrive/Documents/SEMESTER 3/WEB LANJUT
$ php artisan make:controller CalculatorController
```

dan isi dengan kode berikut :

```
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' => $validated['number1'] != 0
27                 ? ($validated['number1'] / $validated['number2'])
28                 : 'Error: Division by 0',
29         };
30
31         return view('calculator', [
32             'result' => $result,
33             'number1' => $validated['number1'],
34             'number2' => $validated['number2'],
35             'operator' => $validated['operator']
36         ]);
37     }
38 }
```

3. Tambahkan di resources/views/calculator.blade.php


```

<!DOCTYPE html>
<html>
<head>
    <title>Laravel Calculator</title>
</head>
<body>
    <h1>Kalkulator Sederhana</h1>

    @if ($errors->any())
        <div style="color: red;">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form method="POST" action="{{ route('calculator.calculate') }}">
        @csrf
        <input type="number" name="number1" value="{{ old('number1', $number1 ?? '') }}" placeholder="Angka Pertama" required>

        <select name="operator" required>
            <option value="add" {{ ($operator ?? '') == 'add' ? 'selected' : '' }}>+</option>
            <option value="sub" {{ ($operator ?? '') == 'sub' ? 'selected' : '' }}>-</option>
            <option value="mul" {{ ($operator ?? '') == 'mul' ? 'selected' : '' }}>*</option>
            <option value="div" {{ ($operator ?? '') == 'div' ? 'selected' : '' }}>/</option>
        </select>

        <input type="number" name="number2" value="{{ old('number2', $number2 ?? '') }}" placeholder="Angka Kedua" required>
        <button type="submit">Hitung</button>
    </form>

    @isset($result)
        <h3>Hasil: {{ $result }}</h3>
    @endisset
</body>
</html>

```

4. Jalankan aplikasi dan coba dengan beberapa input berbeda. **Penjumlahan**

Lab 02 -> Laravel Fundamentals | Laravel Calculator

127.0.0.1:8000/calculator

Kalkulator Sederhana

52 + 32 Hitung

Hasil: 84

Lab 02 -> Laravel Fundamentals | Laravel Calculator

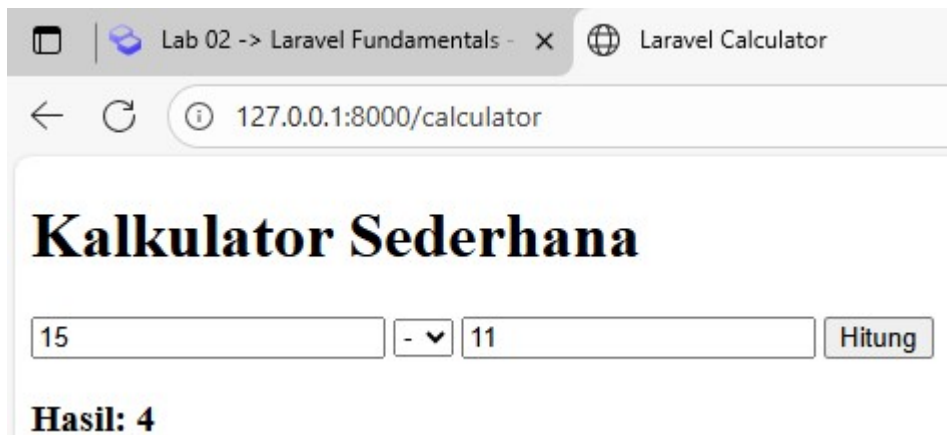
127.0.0.1:8000/calculator

Kalkulator Sederhana

11 + 15 Hitung

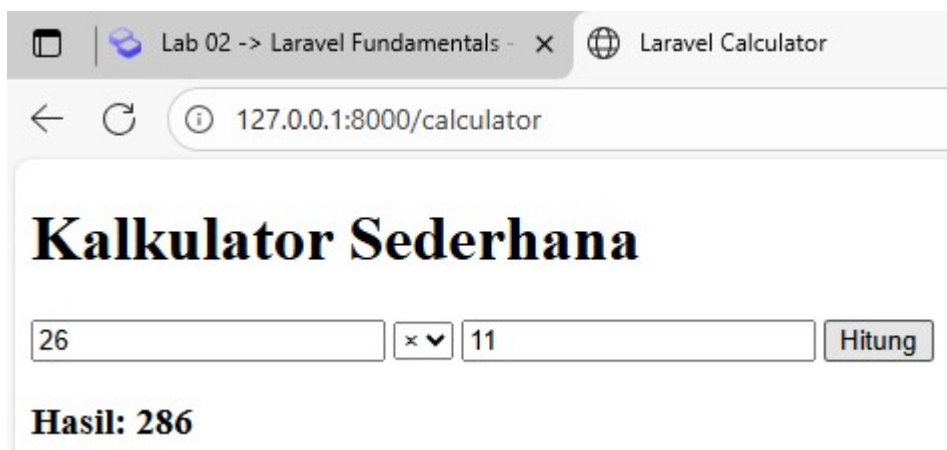
Hasil: 26

Pengurangan



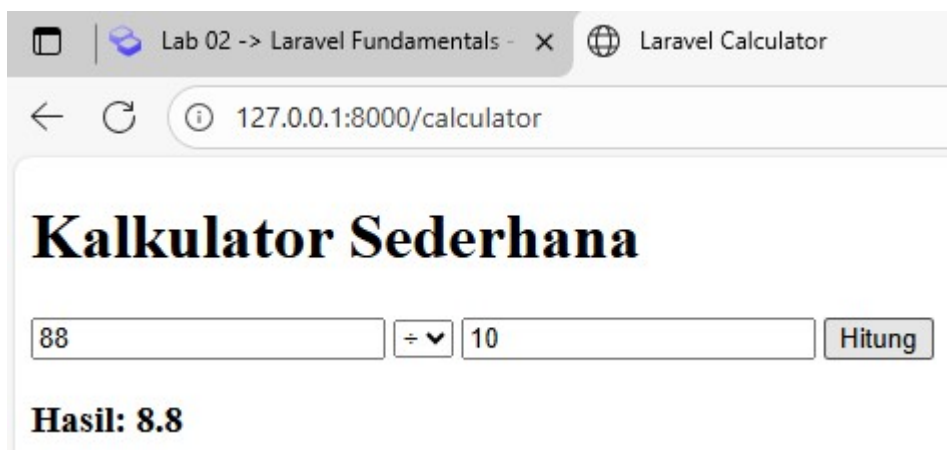
The screenshot shows a web browser with two tabs: 'Lab 02 -> Laravel Fundamentals - X' and 'Laravel Calculator'. The address bar shows '127.0.0.1:8000/calculator'. The page title is 'Kalkulator Sederhana'. Below the title, there are two input fields: the first contains '15' and the second contains '11'. Between them is a dropdown menu with a minus sign '-' selected. To the right is a 'Hitung' button. Below the inputs, the result is displayed as 'Hasil: 4'.

Perkalian



The screenshot shows the same web browser interface. The address bar and tabs are identical. The page title is 'Kalkulator Sederhana'. The input fields now contain '26' and '11', and the dropdown menu has a multiplication sign 'x' selected. The 'Hitung' button is still present. The result displayed is 'Hasil: 286'.

Pembagian

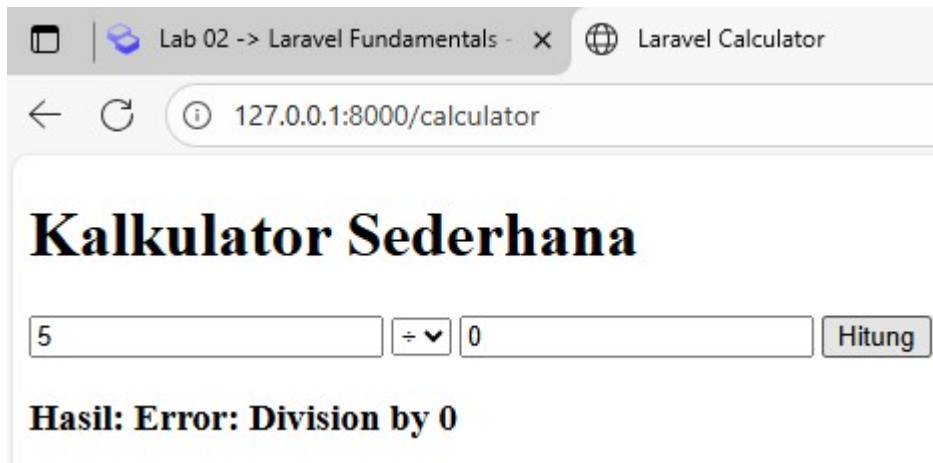


The screenshot shows the same web browser interface. The address bar and tabs are identical. The page title is 'Kalkulator Sederhana'. The input fields now contain '88' and '10', and the dropdown menu has a division sign '÷' selected. The 'Hitung' button is still present. The result displayed is 'Hasil: 8.8'.

3. Hasil dan Pembahasan

Jelaskan apa hasil dari praktikum yang dilakukan.

- Apakah aplikasi berjalan sesuai harapan? Aplikasi yang telah dilakukan pada praktikum ini berjalan sesuai harapan.
- Apa yang terjadi jika ada input yang salah (misalnya pembagian dengan 0)? Hasilnya akan eror, seperti yang terlihat pada gambar ini



- Bagaimana validasi input bekerja di Laravel? Validasi yang ada di CalculatorController.php, memastikan yang diinputkan sesuai aturan sebelum perhitungan. Jika inputan dari user tidak sesuai aturan, user akan langsung dibalikkan ke form dengan pesan error
- Apa peran masing-masing komponen (Route, Controller, View) dalam program yang dibuat? **Route** berperan sebagai penghubung URL yang akan diakses user yang akan dijalankan di dalam aplikasi **Controller** berperan sebagai mengelolah logika program, seperti menerima input, melakukan validasi, dan menyiapkan hasil **View** untuk menampilkan hasil dari inputan user.

4. Kesimpulan

Dalam praktikum ini memperkenalkan kita pada pondasi utama dalam membangun aplikasi Laravel. Untuk lebih memahami routing, penanganan request/response, controller, view, dan Blade tempating yang sangat penting untuk mulai mengembangkan dengan Laravel secara efektif.

5. Referensi

- Modul 2 Laravel Fundamental - <https://hackmd.io/@mohdrzu/B1zwKEK5xe>
- Blade Templating Engine : Mengoptimalkan Pengembangan Web dengan Efisien - <https://baraka.uma.ac.id/blade-templating-engine-mengoptimalkan-pengembangan-web-dengan-efisien/#:~:text=Blade%20Templating%20Engine%20membawa%20kemudahan,web%20yang%20kompl eks%20dengan%20mudah>