

Laporan Modul 2: Laravel Fundamentasl

Mata Kuliah: Workshop Web Lanjut

Nama: Hayzar Muhaiyar **NIM:** 2024573010100 **Kelas:** TI-2C

Abstrak

Laporan ini membahas dasar-dasar penggunaan Laravel pada modul 2 dengan fokus pada konsep MVC, routing, middleware, serta cara Laravel menangani request dan response. Praktikum dilakukan dengan membuat route, controller, dan view menggunakan Blade Templating Engine. Selain itu, dibuat juga aplikasi sederhana berupa "Calculator" untuk memahami alur kerja Laravel dari request hingga response.

1. Dasar Teori

- Apa itu MVC (Model, View, Controller). MVC adalah pola arsitektur perangkat lunak yang memisahkan logika aplikasi menjadi tiga bagian utama: Model: Mengelola data dan logika bisnis. View: Bertugas menampilkan data dalam bentuk antarmuka pengguna. Controller: Menjembatani antara Model dan View dengan menangani input serta mengatur alur aplikasi.
 - Konsep Routing di Laravel. Routing berfungsi untuk menentukan ke mana request pengguna akan diarahkan. Laravel mendukung route sederhana, route dengan parameter, route ke controller, hingga route group. Semua didefinisikan di file routes/web.php.
 - Fungsi Middleware. Middleware adalah lapisan penghubung yang memproses HTTP request sebelum mencapai controller. Contoh penggunaannya adalah autentikasi, validasi request, logging, atau filtering request.
 - Bagaimana cara Laravel menangani Request dan Response. Laravel menggunakan HTTP Kernel untuk menerima request, kemudian melewati middleware, diteruskan ke route, lalu controller, dan hasil akhirnya dikembalikan dalam bentuk response ke browser.
 - Peran Controller dan View. controller: Menangani logika aplikasi, memproses input, dan memilih view yang tepat untuk ditampilkan. View: Bertugas menampilkan hasil ke pengguna. Di Laravel, view menggunakan Blade Templating Engine.
 - Fungsi Blade Templating Engine. Blade adalah template engine Laravel yang memungkinkan pembuatan tampilan dinamis dengan sintaks sederhana, mendukung layouting, inheritance, dan directive seperti @if, @foreach, dll.
-

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. buka folder routes/web.php kemudian isi dengan kode berikut:

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

route ini akan memberitahu laravel untuk memanggil method show() yang ada di dalam file WelcomeController ketika user mengunjungi /welcome.

- Buat controller WelcomeController. Buka terminal(git bash) dan jalankan perintah berikut untuk membuat controller. php artisan make:controller WelcomeController kemudian isi dengan kode berikut

```
project > hello-laravel > app > Http > Controllers > WelcomeController.php
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. buat file bernama mywelcome.blade.php di dalam folder resources/views selanjutnya isi dengan kode berikut



```
project > hello-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 party</title>
7 </head>
8 <body>
9   <h1>welcome to party 12</h1>
10  <p> hello {{$message }}
11  </p>
12 </body>
13 </html>
```

- Jalankan aplikasi dan tunjukkan hasil di browser. ketik php artisan serve di git bash atau terminal

welcome to party 12

hello Welcome to laravel!

Screenshot Hasil:

2.2 Praktikum 2 – Membuat Aplikasi Sederhana "Calculator"

- Tambahkan route untuk kalkulator. buka folder routes/web.php kemudian isi dengan kode berikut:
Route::get('/calculator', [CalculatorController::class, 'index']); Route::post('/calculator',

```
[CalculatorController::class, 'calculate')->name('calculator.calculate');
```

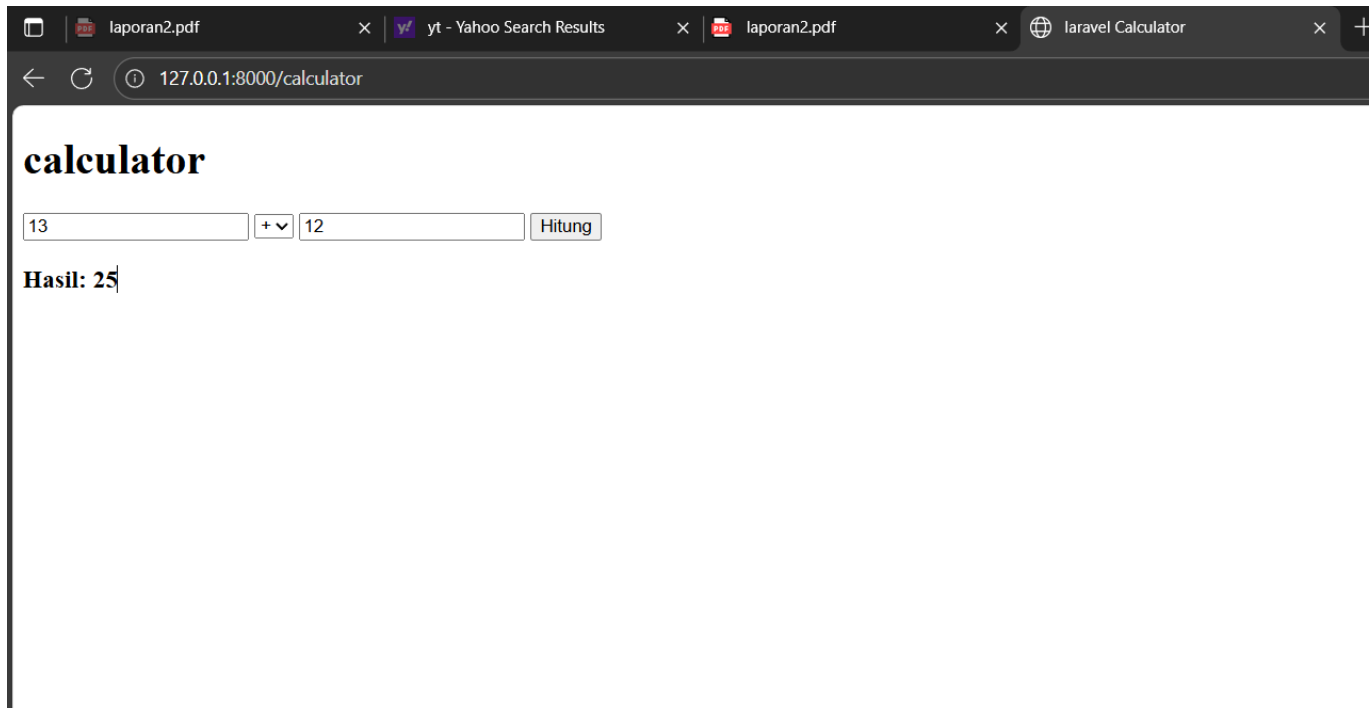
```
project > hello-laravel > routes > web.php
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\welcomeController;
5 use App\Http\Controllers\mywelcomeController;
6
7 Route::get('/', function () {
8     return view('welcome');
9 });
10
11 Route::get('/welcome', [welcomeController::class,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

- Buat controller CalculatorController. buka terminal atau gitbash kemudian ketik kode berikut php artisan make:controller WelcomeController
- Tambahkan view calculator.blade.php. buat file bernama calculator.blade.php di dalam folder resources/views selanjutnya isi dengan kode berikut

```
ade.php U mywelcome.png U laporan2.md U web.php ...Calculator\... U web.php ...hello-laravel\... M CalculatorController.php U
projects > projects > Calculator > resources > views > calculator.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>laravel Calculator</title>
7 </head>
8 <body>
9     <h1>calculator</h1>
10    @if ($errors->any())
11    <div style="color: red;">
12        <ul>
13            @foreach ($errors->all() as $error)
14                <li>
15                    {{ $error}}
16                </li>
17            @endforeach
18        </ul>
19    </div>
20    @endif
21    <form method="POST" action="{{ route('calculator.calculate') }}">
22        @csrf
23        <input type="number" name="number1" value="{{ old('number1', $number1 ?? '') }}" placeholder="Angka Pertama" required>
24        <select name="operator" required>
25            <option value="add" {{ ($operator ?? '') == 'add' ? 'selected' : '' }}>+</option>
26            <option value="add" {{ ($operator ?? '') == 'sub' ? 'selected' : '' }}>-</option>
27            <option value="add" {{ ($operator ?? '') == 'mul' ? 'selected' : '' }}>*</option>
28            <option value="add" {{ ($operator ?? '') == 'div' ? 'selected' : '' }}>/</option>
29        </select>
30
31        <input type="number" name="number2" value="{{ old('number2', $number2 ?? '') }}" placeholder="Angka Kedua" required>
32        <button type="submit">Hitung</button>
33    </form>
34    @if (isset($result))
35    <h3>Hasil: {{ $result }}</h3>
36    @endisset
37 </body>
38 </html>
```

- Jalankan aplikasi dan coba dengan beberapa input berbeda. buka terminal atau gitbash kemudian ketik kode berikut php artisan make:controller WelcomeController;

Screenshot Hasil:



3. Hasil dan Pembahasan

Jelaskan apa hasil dari praktikum yang dilakukan.

- Apakah aplikasi berjalan sesuai harapan? Aplikasi berjalan sesuai harapan. Route dapat mengarahkan request ke controller yang tepat, lalu menampilkan view menggunakan Blade.
- Apa yang terjadi jika ada input yang salah (misalnya pembagian dengan 0)? Pada kalkulator, operasi penjumlahan, pengurangan, perkalian, dan pembagian dapat dijalankan dengan baik. Jika pengguna mencoba membagi dengan angka 0, sistem akan menampilkan pesan error karena validasi sudah ditangani di controller.
- Bagaimana validasi input bekerja di Laravel? Validasi input di Laravel dilakukan dengan `$request->validate()`, yang otomatis memberikan feedback jika ada input yang tidak sesuai aturan.
- Apa peran masing-masing komponen (Route, Controller, View) dalam program yang dibuat? Peran komponen dalam program: Route → Mengarahkan request ke controller yang sesuai. Controller → Memproses logika perhitungan dan menentukan view yang akan ditampilkan. View → Menampilkan hasil kalkulasi dalam format HTML.

4. Kesimpulan

Tulis kesimpulan dari praktikum ini. Dari praktikum ini dapat disimpulkan bahwa: Laravel mempermudah pengembangan aplikasi web dengan konsep MVC. Routing berfungsi sebagai penghubung antara URL dan controller. Controller mengatur logika bisnis aplikasi, sementara View mengatur tampilan dengan bantuan Blade Templating Engine. Middleware berperan penting dalam memproses request sebelum mencapai controller. Contoh Proyek Kecil "Calculator" berhasil

menunjukkan bagaimana alur request → controller → view bekerja secara menyeluruh di Laravel.

5. Referensi

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

chatgpt.com <https://hackmd.io/@mohdrzu/B1zwKEK5xe>

<https://laravel.com/docs>