

Laporan Modul 2: Laravel Fundamentasl

Mata Kuliah: Workshop Web Lanjut

Nama: [Imam Nawawi]

NIM: [2024573010025]

Kelas: [TI-2C]

Abstrak

laporan ini berisi Tentang bagaimana membuat kalkulator dan tujuan saya membuat laporan ini , untuk mengedukasikan/ membagikan bagaimana cara membuat kalkulator di php

1. Dasar Teori

- Model-View-Controller (MVC) adalah sebuah pola desain perangkat lunak yang membagi logika aplikasi menjadi tiga komponen utama: Model (mengelola data dan logika bisnis), View (menampilkan data kepada pengguna), dan Controller (menghubungkan Model dan View serta menangani alur kerja aplikasi).
 - Routing di laravel dapat diartikan bagaimana aplikasi merespons request yang masuk. Routing Menetakan URI ke aksi controller atau closure. rout di laravel didefinisikan di file routes/web.php untuk permintaan web dan di routes /api.php untuk route API
 - Fungsi middleware Laravel adalah sebagai perantara atau filter yang memproses request HTTP sebelum mencapai logika aplikasi inti (seperti controller), serta dapat memproses response setelah logika dijalankan, sebelum dikirim ke pengguna. Dan beberapa penggunaan umum middleware di laravel adalah (Authentication/auth, Logging, CORS, Input sanitization)
 - Laravel menangani request dan response melalui siklus hidup aplikasi yang terstruktur. Permintaan dari pengguna masuk melalui entry point public/index.php, lalu diproses oleh Kernel HTTP yang meneruskan request melalui serangkaian middleware. Akhirnya, request akan menuju routing dan controller yang menentukan logika aplikasi, kemudian controller mengembalikan objek response, yang selanjutnya akan diproses kembali melalui middleware sebelum dikirim ke browser pengguna
 - Peran Controller dan View.
 - Controller Mengatur logika aplikasi dan menjadi penghubung antara model(data) dan view(tampilan). Controller menerima request dari user untuk memproses data(seperti mengambil database lewat model)
- View Bagian yang menampilkan data ke pengguna. View berisi kode HTML/CSS/JS , Serta blade yang menerima data dari controller untuk ditampilkan ke dalam browser
- Fungsi Blade Templating Engine.
 - Blade adalah sistem template bawaan Laravel yang memungkinkan Anda menulis kode PHP di dalam file tampilan dengan sintaks yang lebih rapi dan aman.

- Menyediakan fitur seperti template inheritance (@extends, @section, @yield), komponen dan partials, loop & conditional (@foreach, @if), dan escaped output ({{ }}) untuk mencegah serangan XSS.
- Memudahkan pemisahan logika dan tampilan, sehingga kode lebih terstruktur dan mudah dipelihara.

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

- Langkah-Langkah dalam menambahkan code dalam route/web.php
 1. Buka project nya yaitu hello-laravel
 2. Lalutambahkan kode berikut(untuk menambah route pada routes):



```
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']);
```

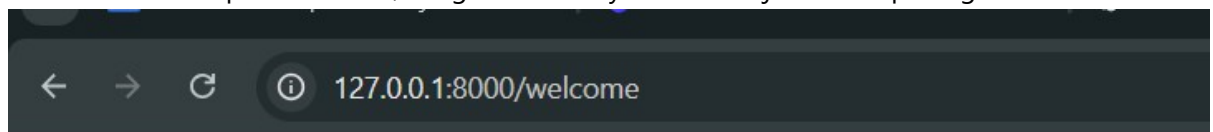
isi code dari gambar tersebut yaitu berfungsi sebagai peta rute untuk laravel anda/ routes/web.php yang menentukan URL apa yang tersedia di web-app dan aksi apa yang dijalankan

- kemudian buat view mywelcome.blade.php. dengan contoh code nya seperti pada gambar yg ada dibawah:

```
eb-lanjut-2024573010025 > project > helo-laravel > resources > views > mywelcome.blade.php > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4
5     <title>Welcome to Laravel </title>
6 </head>
7 <body>
8     <h1>
9         Welcome to Laravel 12
10    </h1>
11    <p>
12        Hello, {{$message}}
13    </p>
14 </body>
15 </html>
```

Code itu untuk menampilkan bagian welcome page(beranda dengan penulisan Welcome to laravel 12)

- Dan hasil dari outputan di atas , Paeg welcome nya akan menjadi seerti pada gambar berikut:



Welcome to Laravel 12

Hello, Welcome To Laravel!

2.2 Praktikum 2 – Membuat Aplikasi Sederhana "Calculator"

- Buatlah project baru di folder bernama calculator, kemudian setelah sudah di install ke folder calculator, tambahkan route ke routes/web.php dan contoh code nya ada pada gambar berikut:

```

1  <?php
2
3  use App\Http\Controllers\CalculatorController;
4  use Illuminate\Support\Facades\Route;
5
6  Route::get('/', function () {
7      return view('welcome');
8  });
9
10 });
11
12 Route::get('/calculator', [CalculatorController::class, 'index']);
13 Route::post('/calculator', [CalculatorController::class, 'calculate']) ->name('calculator.calculate');
14
15
16

```

- Membuat controller CalculatorController. Buat seperti ini pertama buka terminal lebih dahulu dengan code seperti ini: "php artisan make:controller CalculatorController" lalu buka app/Http/Contollers/CalculatorController dan ubah code nya menjaidi seperti ini:

```

project > calculator > app > Http > Controllers > CalculatorController.php > CalculatorController > index
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
30         return view('calculator', [
31             'result' => $result,
32             'number1' => $validated['number1'],
33             'number2' => $validated['number2'],
34             'operator' => $validated['operator']
35         ]);
36     }
37 }

```

- Menambahkan code di view calculator.blade.php. Pertama buat file di resources/views lalu buat calculator.blade.php Setelah itu isikan dengan code seperti ini:

```

project > calculator > resources > views > calculator.blade.php > html > body > h1
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Laravel Calculator</title>
5 </head>
6 <body>
7 <h1>Kalkulator Sederhana</h1>
8
9
10 @if ($errors->any())
11 <div style="color:red;">
12 <ul>
13     @foreach ($errors->all() as $error)
14         <li>{{ $error }}</li>
15     @endforeach
16 </ul>
17 </div>
18 @endif
19
20 <form method="POST" action="{{ route('calculator.calculate') }}">
21 @csrf
22 <input type="number" name="number1" value="{{ old('number1', $number1 ?? '') }}" placeholder="Angka Pertama" required>
23
24 <select name="operator" required>
25 <option value="add" {{ ($operator ?? '') == 'add' ? 'selected' : '' }}>+</option>
26 <option value="sub" {{ ($operator ?? '') == 'sub' ? 'selected' : '' }}>-</option>
27 <option value="mul" {{ ($operator ?? '') == 'mul' ? 'selected' : '' }}>*</option>
28 <option value="div" {{ ($operator ?? '') == 'div' ? 'selected' : '' }}>/</option>
29 </select>
30
31 <input type="number" name="number2" value="{{ old('number2', $number2 ?? '') }}" placeholder="Angka Kedua" required>
32 <button type="submit">Hitung</button>
33 </form>
34
35 @isset($result)
36 <h3>Hasil: {{ $result }}</h3>
37 @endisset
38
39 </body>
40 </html>

```

View ini berisi form untuk input user dan menampilkan hasil perhitungan. juga menangani error validasi

- Hasil dari outputan code di atas ' -Halaman beranda nya:

Kalkulator Sederhana

-Hasil percobaan calculator

Kalkulator Sederhana

| | | | |
|--------------------------------|------------------------------------|---------------------------------|---------------------------------------|
| <input type="text" value="7"/> | <input type="button" value="+ v"/> | <input type="text" value="11"/> | <input type="button" value="Hitung"/> |
|--------------------------------|------------------------------------|---------------------------------|---------------------------------------|

Hasil: 18

Screenshot Hasil:

3. Hasil dan Pembahasan

Jelaskan apa hasil dari praktikum yang dilakukan.

- Sesuai dengan harapan meskipun banyak problem yang saya tidak terlalu mengerti seperti di env nya

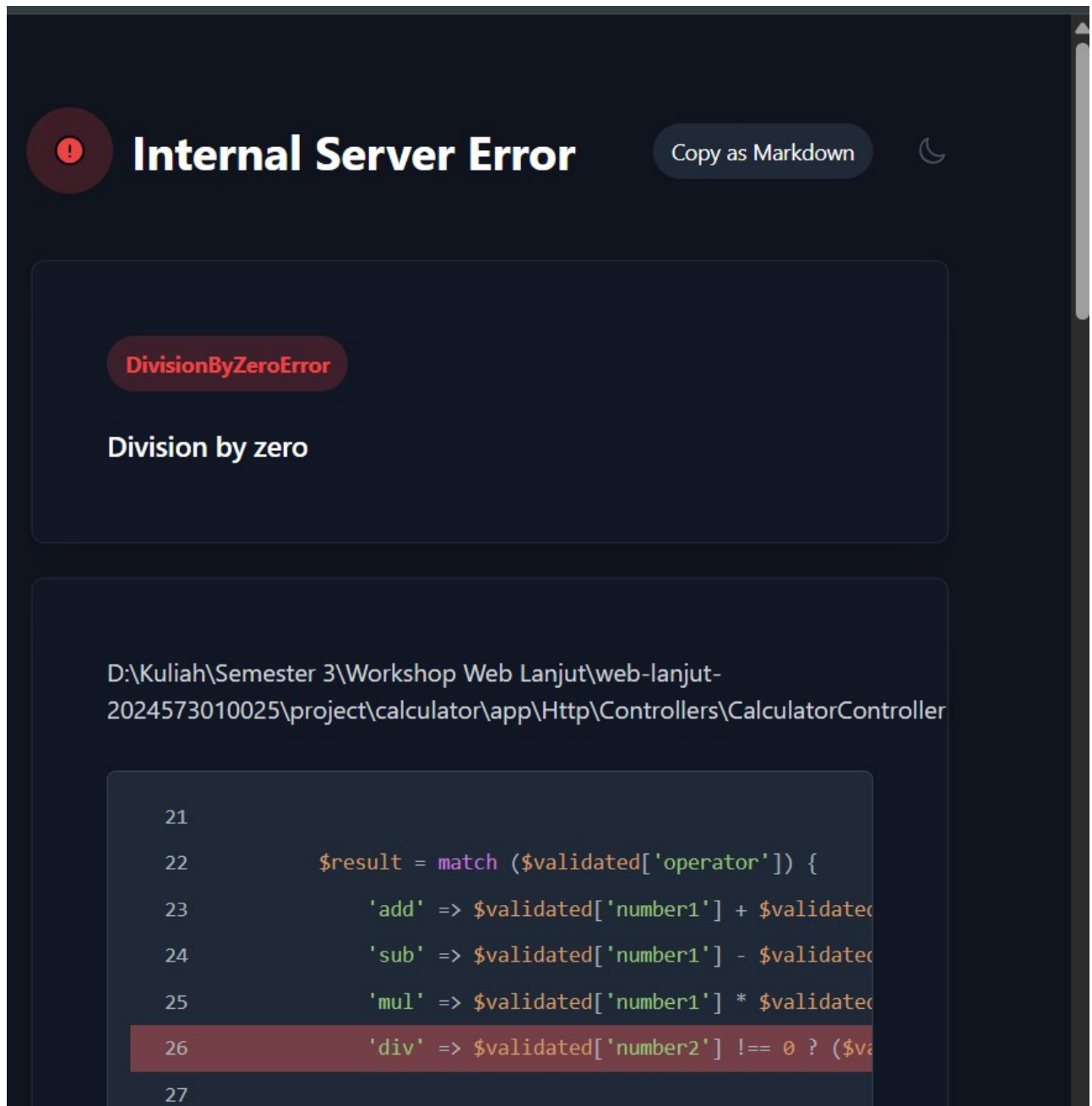
```
DB_CONNECTION=file
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=null
DB_USERNAME=root
DB_PASSWORD=

SESSION_DRIVER=file
SESSION_LIFETIME=120
SESSION_ENCRYPT=false
SESSION_PATH=/
SESSION_DOMAIN=null

BROADCAST_CONNECTION=log
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
```

menjadi seperti ini:

- Apa yang terjadi jika ada input yang salah (misalnya pembagian dengan 0)? Maka akan terjadi error seperti pada gambar ini



- Bagaimana validasi input bekerja di Laravel?
 - Di Laravel, validasi input bekerja dengan cara memeriksa data yang dikirim dari request (misalnya dari form) agar sesuai dengan aturan yang kita tentukan sebelum data diproses atau disimpan.
- Apa peran masing-masing komponen (Route, Controller, View) dalam program yang dibuat?
 1. Route Berfungsi Menjadi “pintu masuk” aplikasi. Dan Perannya: Mencocokkan URL/HTTP method yang diminta pengguna (GET, POST, dll.) dengan aksi yang tepat, misalnya memanggil controller tertentu atau langsung menampilkan view.

Ibarat: Petunjuk jalan yang menentukan ke mana request diarahkan.

2. Controller berfungsi untuk mengolah logika bisnis.

Perannya: Menerima data dari Route, memvalidasi/olah data (misalnya ambil dari database melalui Model), lalu menyiapkan data untuk dikirim ke View.

Ibarat: Otak atau "juru masak" yang menyiapkan hidangan sesuai pesanan.

3. View Berfungsi Menampilkan output ke pengguna.

Perannya: File blade (.blade.php) yang memuat HTML/CSS/JS untuk menyajikan data yang sudah diolah Controller.

Ibarat: Piring saji yang menampilkan makanan agar enak dilihat.

4. Kesimpulan

Fitur Lengkap laravel memudahkan user untuk membuat sebuah web dengan adanya fitur seperti Menyediakan routing, middleware, templating (Blade), ORM Eloquent untuk database, autentikasi, serta sistem artisan command yang mempercepat pengembangan sebuah web.

5. Referensi

- <https://www.everand.com/book/874828246/Laravel-12-Training-Kit-A-Practical-Guide-to-Modern-Web-Development>
 - <https://santrikoding.com/tutorial-laravel-12-2-cara-install-laravel-12>
-