

Laporan Modul 2: Laravel Fundamentasl

Mata Kuliah: Workshop Web Lanjut

Nama: M. Iqbal Sayuti

NIM: 2024573010057

Kelas: TI-2C

Abstrak

Tuliskan ringkasan singkat tentang isi laporan ini dan tujuan Anda membuat laporan. - Praktikum ini bertujuan untuk mempelajari fundamental Laravel, termasuk konsep MVC (Model, View, Controller), routing, middleware, request dan response, serta penggunaan Blade Templating Engine dalam pembuatan tampilan web. Praktikum dilakukan dengan membuat aplikasi sederhana, mulai dari menambahkan route, membuat controller, hingga menampilkan view. Hasil praktikum menunjukkan bahwa Laravel memudahkan pengembangan aplikasi web yang terstruktur, efisien, dan mudah dipelihara, serta mendukung validasi input secara otomatis.

1. Dasar Teori

- Apa itu MVC (Model, View, Controller).
 - MVC adalah sebuah pola arsitektur dalam membuat sebuah aplikasi dengan cara memisahkan kode menjadi tiga bagian yang terdiri dari Model, View, Controller
- Konsep Routing di Laravel.
 - Routing di Laravel adalah mekanisme untuk menentukan bagaimana aplikasi merespons sebuah request (misalnya akses URL).
 - Semua route biasanya didefinisikan di folder routes/, terutama pada file web.php (untuk web) dan api.php (untuk API).
 - Route bisa diarahkan ke closure, controller, atau bahkan view langsung.
 - Laravel mendukung berbagai metode HTTP seperti GET, POST, PUT, DELETE, serta grouping (middleware, prefix, namespace).
- Fungsi Middleware.
 - Middleware menyediakan mekanisme untuk memfilter permintaan HTTP yang masuk ke aplikasi Anda. Middleware bertindak sebagai lapisan yang berada di antara siklus request dan response.
- Bagaimana cara Laravel menangani Request dan Response.
 - Laravel menyediakan cara sederhana untuk mengembalikan response. Kita dapat mengembalikan string, view, JSON, atau bahkan file download, Routing di Laravel mendefinisikan bagaimana aplikasi merespons permintaan (request) yang masuk. Routing memetakan URI ke aksi controller atau closure. Route di Laravel didefinisikan di file routes/web.php untuk permintaan web dan di routes/api.php untuk route API.
- Peran Controller dan View.
 - Controller : Bagian yang bertugas untuk menghubungkan serta mengatur model dan view agar dapat saling terhubung.
 - View : Bagian yang bertugas untuk menampilkan informasi dalam bentuk Graphical User Interface (GUI).

- Fungsi Blade Templating Engine.
 - membuat tampilan web yang bersih, efisien, sederhana 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

- 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  Route::get('/', function () {
7      return view('welcome');
8  });
9
10 Route::get('/mywelcome', [WelcomeController::class, 'show']);
```

- Buat controller WelcomeController.

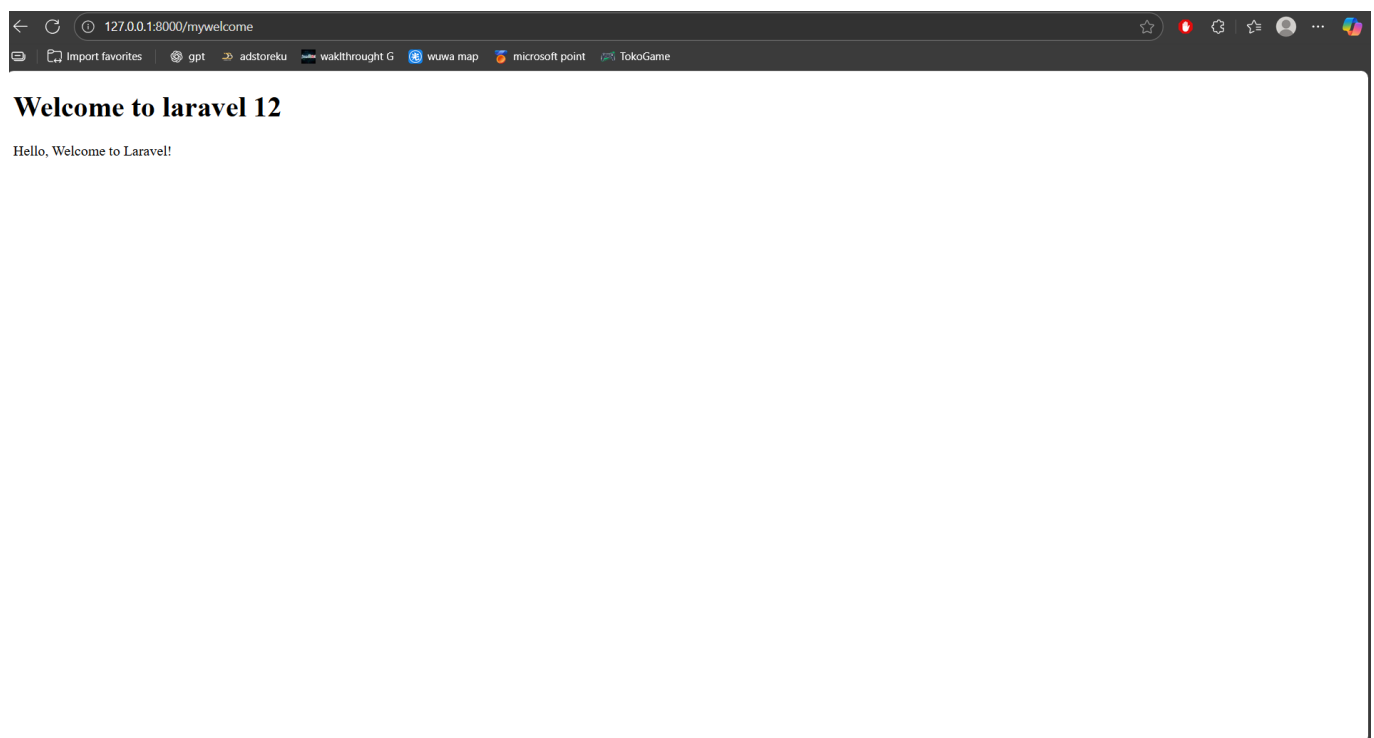
```
project > helo-laravel > app > Http > Controllers > WelcomeController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class WelcomeController extends Controller
8  {
9      public function show()
10     {
11         $message = "Welcome to Laravel!";
12         return view('mywelcome', ['message' => $message]);
13     }
14 }
```

- Buat view mywelcome.blade.php.

```
project > helo-laravel > resources > views > mywelcome.blade.php
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>
```

- Jalankan aplikasi dan tunjukkan hasil di browser.

Screenshot Hasil:



2.2 Praktikum 2 – Membuat Aplikasi Sederhana "Calculator"

- Tambahkan route untuk kalkulator.

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

- Buat controller CalculatorController.

```

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 zero',
27         };
28
29         return view('calculator', [
30             'result' => $result,
31             'number1' => $validated['number1'],
32             'number2' => $validated['number2'],
33             'operator' => $validated['operator'],
34         ]);
35     }
36 }

```

- Tambahkan view calculator.blade.php.

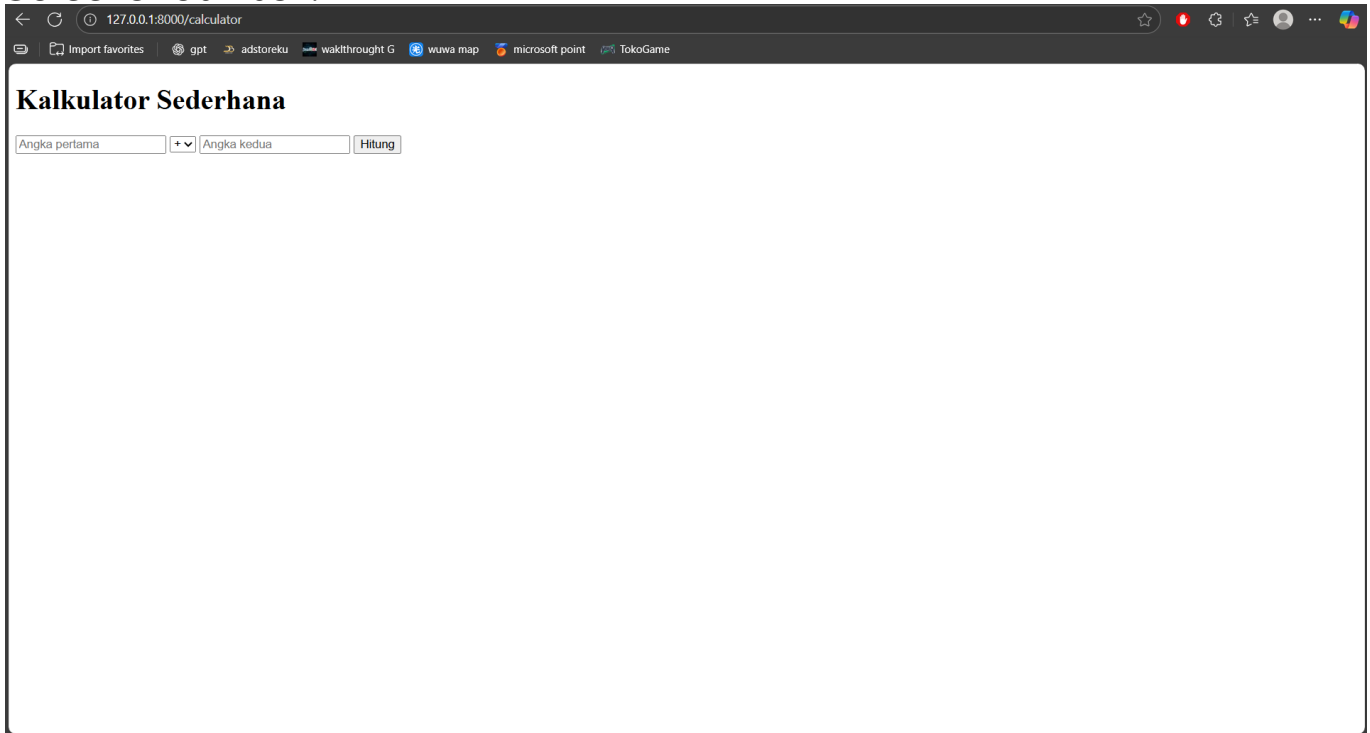
```

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>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', $number1 ?? '') }}" placeholder="Angka pertama" required>
24
25          <select name="operator" required>
26              <option value="add" {{ ($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          </select>
31
32          <input type="number" name="number2" value="{{ old('number2', $number2 ?? '') }}" placeholder="Angka kedua" required>
33          <button type="submit">Hitung</button>
34      </form>
35
36      @isset($result)
37          <h3>Hasil {{ $result }}</h3>
38      @endisset
39  </body>
40 </html>

```

- Jalankan aplikasi dan coba dengan beberapa input berbeda.

Screenshot Hasil:



3. Hasil dan Pembahasan

Jelaskan apa hasil dari praktikum yang dilakukan.

- Apakah aplikasi berjalan sesuai harapan?
 - ya, aplikasi berjalan sesuai code yang telah diketikkan
- Apa yang terjadi jika ada input yang salah (misalnya pembagian dengan 0)?
 - akan menampilkan "Hasil Error: Division by zero"
- Bagaimana validasi input bekerja di Laravel?
 - Validasi input di Laravel bekerja dengan mengecek data request terhadap aturan (rules) yang ditentukan. Jika valid, data diproses; jika gagal, Laravel otomatis mengembalikan error ke user (biasanya redirect dengan pesan error).
- Apa peran masing-masing komponen (Route, Controller, View) dalam program yang dibuat?
 - Route : Route di Laravel didefinisikan di file routes/web.php untuk permintaan web dan di routes/api.php untuk route API.
 - Controller : Bagian yang bertugas untuk menghubungkan serta mengatur model dan view agar dapat saling terhubung.
 - View : Bagian yang bertugas untuk menampilkan informasi dalam bentuk Graphical User Interface (GUI).

4. Kesimpulan

Tulis kesimpulan dari praktikum ini

Laravel memudahkan pembuatan aplikasi web dengan menerapkan konsep MVC (Model, View, Controller) yang memisahkan logika, tampilan, dan data. Route berfungsi untuk menangani request dari pengguna, Controller memproses logika aplikasi, dan View menampilkan

informasi secara interaktif. Penggunaan Blade Templating Engine membuat pembuatan tampilan menjadi lebih rapi, efisien, dan mudah dipelihara.

5. Referensi

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

- Apa Itu MVC? Pahami Konsepnya dengan Baik — <https://www.dicoding.com/blog/apa-itu-mvc-pahami-konsepnya/>
 - Modul 2 - Laravel Fundamentals — <https://hackmd.io/@mohdrzu/B1zwKEK5xe#Praktikum-1---Route-Controller-dan-Blade-View>
-