

Laporan Modul 2: Laravel Fundamentasl

Mata Kuliah: Workshop Web Lanjut

Nama: Firdhila Ananda Syahputri **NIM:** 2024573010014 **Kelas:** TI-2C

Abstrak

Laporan ini menjelaskan dasar Laravel tentang MVC(Model-View-Controller). Pada laporan ini terdapat dua percobaan praktikum, yaitu menampilkan halaman menggunakan route, controller, dan view, serta membuat kalkulator sederhana. Tujuan dari laporan ini adalah agar kita dapat memahami cara kerja laravel dalam mengatur aplikasi.

1. Dasar Teori

- MVC (Model-View_Controller) merupakan desain arsitektur yang membantu menjaga kode tetap terorganisir dan memudahkan pengembangan, pemeliharaan, dan skalabilitas aplikasi. MVC memisahkan aplikasi menjadi tiga komponen utama Model, View, dan Controller. Model merupakan bagian yang bertugas untuk menyiapkan, mengatur, dan mengorganisasikan data yang ada di database. View merupakan bagian yang bertugas untuk menampilkan informasi. Controller merupakan bagian yang bertugas untuk menghubungkan serta mengatur model dan view untuk saling terhubung.
 - Konsep Routing di Laravel. Routing melakukan proses pengiriman data maupun informasi ke pengguna melalui sebuah permintaan yang dilakukan kepada alamat yang sudah terdaftar, kemudian alamat tersebut akan mengembalikan sebuah iutput atau hasil dari proses tersebut.
 - Middlewarre merupakan lapisan perangkat lunak yang berada di antara sistem operasi dan aplikasi. Middleware bertugas sebagai perantara yang memfasilitasi komunikasi dan manajemen data antar aplikasi yang berbeda.
 - Bagaimana cara Laravel menangani Request dan Response. Cara laravel menangani Request yaitu dengan mendeklarasikan Illuminate\Http\Request yang ada di dalam CalculatorController.php. Sementara untuk menangani Response yaitu dengan mengembalikan string, view, JSON, atau bahkan file download. `Route::get('response/', function () { return 'Hello World'; });`
 - View merupakan bagian yang bertugas untuk menampilkan informasi. Controller merupakan bagian yang bertugas untuk menghubungkan serta mengatur model dan view untuk saling terhubung.
 - Blade templating adalah sistem template bawaan laravel yang digunakan untuk membuat tampilan web yang lebih rapi dan terstruktur. Blade bisa membuat header atau footer yang bisa digunakan kembali di banyak halaman tanpa perlu menulis ulang kode.
-

2. Langkah-Langkah Praktikum

2.1 Praktikum 1 – Route, Controller, dan Blade View

- Tambahkan route pada routes/web.php.

1. Masuk ke project hello-laravel
2. Buka file routes/web.php
3. Ubah isinya menjadi seperti gambar dibawah ini

```
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']);
```

- Buat controller WelcomeController.

1. Buka git bash kemudian jalankan perintah dibawah ini untuk membuat controller php artisan make:controller WelcomeController

- Buat view mywelcome.blade.php.

1. Buka folder resource, kemudian klik views
2. Buat file mywelcome.blade.php didalam folder views.
3. Kemudian isi kode untuk file mywelcome.blade.php seperti dibawah ini

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Welcome to Laravel</title>
5      </head>
6      <body>
7          <h1>
8              Welcome to Laravel
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.

1. Masuk ke project calculator
2. Buka file routes/web.php
3. Ubah isinya menjadi seperti gambar dibawah ini

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

- Buat controller CalculatorController.

1. Buka git bash kemudian jalankan perintah dibawah ini untuk membuat controller php artisan make:controller CalculatorController

- Tambahkan view calculator.blade.php.

1. Buka folder resource.
2. klik views.
3. kemudian buat file yang namanya calculator.blade.php

Screenshot Hasil:

- Penjumlahan



Kalkulator Sederhana

320 + 2 Hitung

Hasil: 322

- Pengurangan

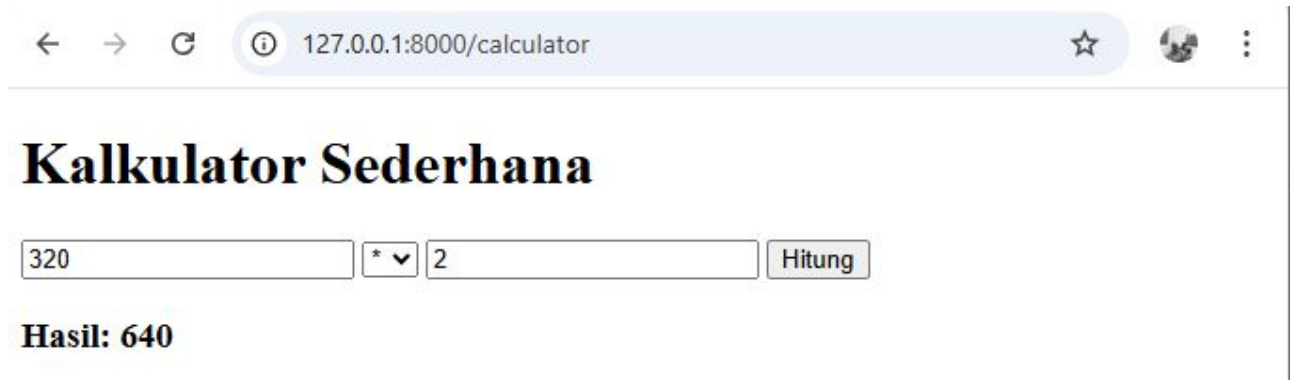


Kalkulator Sederhana

320 - 2 Hitung

Hasil: 318

- Perkalian



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/calculator". The page title is "Kalkulator Sederhana". Below the title, there are two input fields: the first contains "320" and the second contains "2". Between them is a dropdown menu showing a multiplication symbol (*). To the right of the second input field is a button labeled "Hitung". Below the inputs, the result is displayed as "Hasil: 640".

- Pembagian



The screenshot shows the same web browser window as before, but the dropdown menu now shows a division symbol (/). The input fields still contain "320" and "2". The "Hitung" button is still present. Below the inputs, the result is displayed as "Hasil: 160".

3. Hasil dan Pembahasan

- Aplikasi web berjalan sesuai harapan. Halaman berhasil ditampilkan di browser sesuai dengan kode yang ditulis. Pada praktikum membuat kalkulator, aplikasi dapat melakukan semua operasi penjumlahan, pengurangan, perkalian dan pembagian dengan benar.
- Jika kita input yang salah, misalkan angka 0 pada operasi pembagian, maka aplikasi akan menampilkan "Hasil: Error: Division by 0"



The screenshot shows the web browser window with the address bar displaying "127.0.0.1:8000/calculator". The page title is "Kalkulator Sederhana". Below the title, there are two input fields: the first contains "10" and the second contains "0". Between them is a dropdown menu showing a division symbol (/). To the right of the second input field is a button labeled "Hitung". Below the inputs, the result is displayed as "Hasil: Error: Division by 0".

- Cara validasi input bekerja yaitu ketika user klik hitung, datanya akan dikirim ke Calculator Controller. Di controller, laravel akan cek data sudah sesuai. Kalau datanya ada yang salah, maka akan menampilkan pesan error. Tapi kalau datanya benar, maka data tersebut dapat dihitung sesuai operator yang dipilih. Kemudian hasilnya akan ditampilkan.

- Controller, Mengatur alur antara model dan view. Controller menerima input dari pengguna melalui route, memprosesnya, dan mengembalikan view atau respons yang sesuai. Route, menyesuaikan alamat URL. View, Menangani lapisan presentasi, yaitu HTML yang ditampilkan kepada pengguna. Laravel menggunakan blade sebagai template engine untuk view.
-

4. Kesimpulan

Dari praktikum ini dapat disimpulkan bahwa laravel memudahkan kita dalam membuat web karena sudah memakai MVC yang rapi dan terstruktur. Pada praktikum ini berhasil menampilkan halaman sederhana dan membuat kalkulator sederhana yang bisa melakukan penjumlahan, perkalian, pengurangan dan pembagian.

5. Referensi

- <https://www.codepolitan.com/blog/apa-itu-middleware-pengertian-contoh-dan-fungsinya/>
 - <https://medium.com/@muhammad.ibrahim/belajar-laravel-kupas-tuntas-routing-di-laravel-f8dfac2e64fd>
 - <https://www.dicoding.com/blog/apa-itu-mvc-pahami-konsepnya/>
-