

Laporan Modul 3: Laravel Controller

Mata Kuliah: Workshop Web Lanjut

Nama: KHAIRATIL AINI

NIM: 2024573010031

Kelas: TI 2C

Abstrak

- Laporan ini membahas tentang Controller dalam laravel sangat penting untuk manangani logika aplikasi, yang menerima permintaan dari rute, memproses input, dan mengembalikan respons dan juga ada berbagai jenis controller, cara membuatnya, dan praktik terbaik untuk mengorganisasir logika controller di laravel 12.
 - Tujuan laporan ini adalah untuk mendokumentasikan proses pembelajaran dan pemahaman terhadap penggunaan Controller dalam laravel, khususnya dalam menangani alur kerja dari request hingga response.
-

1. Dasar Teori

- Controller adalah Dalam pola MVC (Model-View-Controller), sebuah controller bertindak sebagai jembatan antara model dan view. Controller menangani input pengguna, berinteraksi dengan model untuk data, dan mengembalikan respons yang benar, yang seringkali merender sebuah view. Controller juga menerima permintaan dari pengguna melalui rute (route), memproses input menggunakan logika yang diperlukan (termasuk validasi dan interaksi dengan model), lalu mengembalikan hasil berupa tampilan, JSON, atau respons lainnya
- Konsep controller di Laravel adalah Controller yang berperan sebagai penghubung antara Model dan View dalam pola arsitektur MVC (Model-View-Controller). Controller bertugas untuk menerima request dari pengguna melalui route, memproses logika aplikasi, berinteraksi dengan model jika diperlukan, dan mengembalikan response berupa tampilan (view) atau data. Dengan adanya controller, logika bisnis aplikasi terpisah dari tampilan sehingga kode menjadi lebih terstruktur, bersih, dan mudah dikelola.
- Fungsi controller adalah Fungsi utama Controller dalam Laravel adalah sebagai pengatur logika aplikasi yang menghubungkan antara route, model, dan view dalam pola arsitektur MVC (Model-View-Controller).
- Route pada controller adalah Laravel menyediakan fitur Route Grouping untuk mempermudah pengaturan rute yang ditangani oleh satu controller. Dengan menggunakan Route::controller()->group(), beberapa rute dapat digabung dalam satu blok kode tanpa harus menulis controller yang sama berulang kali. Hal ini menerapkan prinsip Don't Repeat Yourself (DRY) dan membuat struktur rute menjadi lebih efisien serta mudah dibaca. Pengelompokan rute juga membantu dalam pemeliharaan kode saat aplikasi semakin berkembang.
- Request dan response adalah Laravel menangani request dengan mengarahkan setiap permintaan pengguna ke rute yang sesuai. Rute tersebut kemudian meneruskan request ke controller yang bertanggung jawab untuk memproses data, memanggil model jika perlu, dan akhirnya mengembalikan response. Response dapat berupa tampilan HTML, JSON, redirect, maupun teks biasa. Proses ini

menunjukkan bagaimana Laravel mengelola komunikasi dua arah antara pengguna dan server secara efisien dan terstruktur.

2. Langkah-Langkah Praktikum

Tuliskan langkah-langkah yang sudah dilakukan, sertakan potongan kode dan screenshot hasil.

2.1 Praktikum 1 – Menangani Request dan Response View di laravel 12

- Buka projec laravel composer yang sudah dibuat dengan nama lab-view, lalu buat sebuah controller dengan php artisan make:controllerDemoController.

```
app > Http > Controllers > DemoController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class DemoController extends Controller
8  {
9      // simple data passing
10     public function hello()
11     {
12         $name = 'Laravel Learner';
13         return view('hello', ['name' => $name]);
14     }
15     // parameterized route
16     public function greet($name)
17     {
18         return view('greet', ['name' => ucfirst($name)]);
19     }
20
21     //query string
22     public function search(Request $request)
23     {
24         $keyword = $request->query('q', 'name');
25         return view('search', ['keyword' => $keyword]);
26     }
27 }
28
```

- Tambahkan route pada routes/web.php.

```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\DemoController;
5
6  Route::get('/hello', [DemoController::class, 'hello']);
7  Route::get('/greet/{name}', [DemoController::class, 'greet']);
8  Route::get('/search', [DemoController::class, 'search']);
9
10 Route::get('/', function () {
11     return view('welcome');
12 });
13
```

```
resources > views > 🗂 hello.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Hello</title>
5  </head>
6  <body>
7  |   <h1>Hello, {{ $name }}!</h1>
8  </body>
9  </html>
```

- Buat view sederhana hello.blade.php

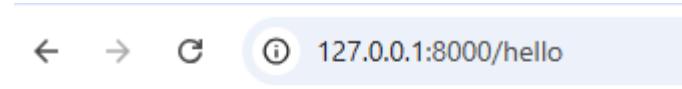
- Buat view sederhana greet.blade.php.

```
resources > views > 🗂 greet.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Greet</title>
5  </head>
6  <body>
7  |   <h1>Nice to meet you, {{ $name }}!</h1>
8  </body>
9  </html>
```

- Buat view sederhana search.blade.php

```
resources > views > 🗂 search.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Search</title>
5  </head>
6  <body>
7  |   <h1>You searched for: ,strong>{{ $keyword }}</strong></h1>
8  </body>
9  </html>
```

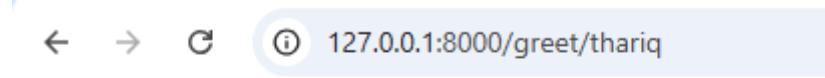
Jalankan aplikasi dan tunjukkan hasil di browser.



127.0.0.1:8000/hello

Hello, LaraveL Learner!

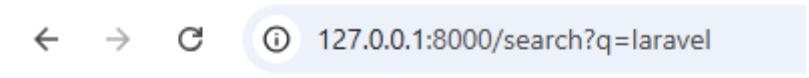
- Hasil hello.blade.php



127.0.0.1:8000/greet/thariq

Nice to meet you, Thariq!

- Hasil greet.blade.php



127.0.0.1:8000/search?q=laravel

You searched for: ,strong>laravel

- Hasil search.blade.php

2.2 Praktikum 2 – Menggunakan Group Route

- Tambahkan route untuk web.php

```
routes > 🗃 web.php
1   <?php
2
3   use Illuminate\Support\Facades\Route;
4   use App\Http\Controllers\PageController;
5
6   Route::controller(PageController::class)->group(function () {
7       Route::get('/', 'home')->name('home');
8       Route::get('/about', 'about')->name('about');
9       Route::get('/contact', 'contact')->name('contact');
10  });

```

- Buat controller PageController

```
app > Http > Controllers > PageController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class PageController extends Controller
8  {
9      public function home()
10     {
11         $message = "Welcome to the homepage.";
12         return view('pages.home', compact('message'));
13     }
14
15     public function about()
16     {
17         $message = "This is the about page.";
18         return view('pages.about', compact('message'));
19     }
20
21     public function contact()
22     {
23         $message = "Reach us through the contact page.";
24         return view('pages.contact', compact('message'));
25     }
26 }
27
```

- Tambahkan view home.blade.php didalam folder: resources/views/ lalu tambah folder pages/

```
resources > views > pages > home.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |  <title>Home</title>
5  </head>
6  <body>
7  |  <h1>{{ $message }}</h1>
8  </body>
9  </html>
```

```
resources > views > pages > about.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |  <title>About</title>
5  </head>
6  <body>
7  |  <h1>{{ $message }}</h1>
8  </body>
9  </html>
```

- tambahkan about.blade.php

```
resources > views > pages > contact.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Contact</title>
5  </head>
6  <body>
7  |   <h1>{{ $message }}</h1>
8  </body>
9  </html>
```

- tambahkan contact.blade.php

Jalankan aplikasi dan tunjukkan hasil di browser.

← → ⌂ ⓘ 127.0.0.1:8000

Welcome to the homepage.

- Hasil home.blade.php

← → ⌂ ⓘ 127.0.0.1:8000/about

This is the about page.

- Hasil about.blade.php

← → ⌂ ⓘ 127.0.0.1:8000/contact

Reach us through the contact page.

- Hasil contact.blade.php

2.3 Praktikum 3 - Pengelompokan Prefix dengan Namespace Rute di laravel 12

- Buat controller dengan Namespace . php artisan make:controller Admin/DashboardController

```
app > Http > Controllers > Admin > DashboardController.php
1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7
8  class DashboardController extends Controller
9  {
10     public function index()
11     {
12         return view('admin.dashboard', ['message' => 'Welcome to Admin Dashboard']);
13     }
14 }
```

```
. php artisan make:controller Admin/UserController
app > Http > Controllers > Admin > UserController.php
1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7
8  class UserController extends Controller
9  {
10     public function index()
11     {
12         $users = ['Ria', 'Lie', 'Jon'];
13         return view('admin.users.index', compact('users'));
14     }
15     public function show($id)
16     {
17         $user = "User #" . $id;
18         return view('admin.users.show', compact('user'));
19     }
20 }
```

- Tambahkan route untuk web.php

```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\Admin\DashboardController;
5  use App\Http\Controllers\Admin\UserController;
6
7  Route::prefix('admin')->group(function () {
8      Route::get('/dashboard', [DashboardController::class, 'index'])->name('admin.dashboard');
9      Route::get('/users', [UserController::class, 'index'])->name('admin.users');
10     Route::get('/users/{id}', [UserController::class, 'show'])->name('admin.users.show');
11  });
12 });
13 Route::get('/', function () {
14     return view('welcome');
15 });
16
```

Buat folder dan file dibawah resources/views/admin/. kemudian buat file-file berikut:

```
resources > views > admin > dashboard.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Admin Dashboard</title>
5  </head>
6  <body>
7  |   <h1>{{ $message }}</h1>
8  </body>
9  <html>
```

- dashboard.blade.php
- Tambah folder users dengan file index.blade.php

```
resources > views > admin > users > index.blade.php
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  |   |   <title>Users</title>
5  |   </head>
6  |   <body>
7  |   |   <h1>User List</h1>
8  |   |   <ul>
9  |   |   |   @foreach ($users as $user)
10 |   |   |   |   <li>{{ $user }}</li>
11 |   |   |   @endforeach
12 |   |   </ul>
13 |   </body>
14 |   </html>
```

```
resources > views > admin > users > show.blade.php
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  |   |   <title>User Details</title>
5  |   </head>
6  |   <body>
7  |   |   <h1>Detail for: {{ $user }}</h1>
8  |   </body>
9  </html>
```

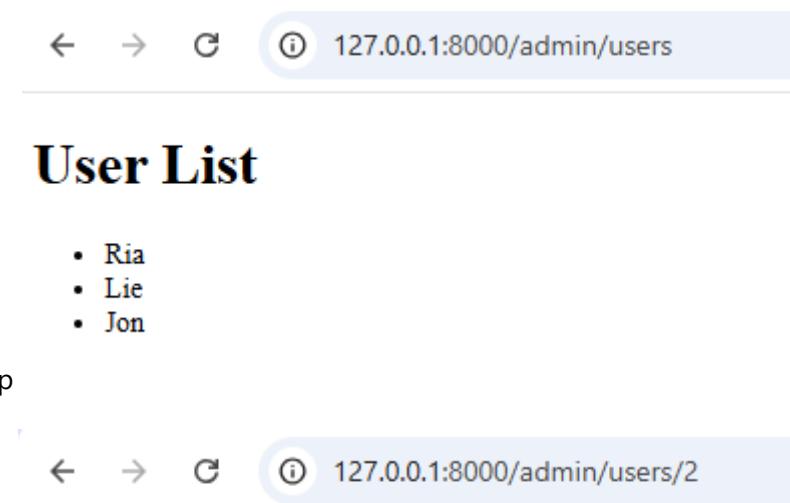
- tambahkan file show.blade.php

Jalankan aplikasi dan tunjukkan hasil di browser.

← → ⌛ 127.0.0.1:8000/admin/dashboard

Welcome to Admin Dashboard

- Hasil dashboard.blade.php



User List

- Ria
- Lie
- Jon

• Hasil index.blade.php



Detail for: User #2

- Hasil show.blade.php

3. Hasil dan Pembahasan

Dari serangkaian praktikum yang dilakukan pada modul Controller di Laravel 12, diperoleh pemahaman dan hasil nyata tentang bagaimana Laravel mengatur alur kerja antara request, controller, dan view dengan cara yang efisien dan terstruktur.

- Dari praktikum 1 : berhasil membuat controller sederhana democontroller yang menangani request dari pengguna dan mengembalikan response berupa tampilan view.
- praktikum 2 : berhasil dibuat controller dengan beberapa metode (PageController) dan rute yang dikelompokkan menggunakan.
- praktikum 3 : berhasil dibuat controller di dalam sub-namespace Admin (Admin/DashboardController dan Admin/UserController) serta rute dengan prefix admin. Dari ketiga praktikum tersebut, diperoleh hasil bahwa controller memiliki peran utama dalam mengatur logika dan alur kerja aplikasi Laravel.

4. Kesimpulan

Kesimpulan dari praktikum modul Controller pada Laravel ini adalah bahwa Controller berperan sangat penting dalam mengatur alur kerja antara request, logika aplikasi, dan tampilan (view). Melalui praktikum yang dilakukan, dipahami bahwa controller berfungsi untuk menerima permintaan dari pengguna, memproses data, dan mengirimkan hasilnya ke view agar dapat ditampilkan di browser. Selain itu, penggunaan route group, prefix, dan namespace membantu dalam mengelompokkan rute dan controller sehingga struktur aplikasi menjadi lebih rapi, efisien, dan mudah dikelola. Dengan memahami konsep controller ini, pengembang dapat membangun aplikasi Laravel yang lebih terorganisir, modular, serta mudah dikembangkan sesuai kebutuhan proyek.

5. Referensi

1. <https://laravel.com/docs/11.x/controllers>
2. <https://laravel.com/docs/12.x/routing>

