

Laporan Modul 2: Laravel Controller

Mata Kuliah: Workshop Web Lanjut

Nama: Ilham Syahdienar

NIM: 2024573010029 **Kelas:** TI-2C

Abstrak

Laporan ini berisi pemahaman tentang controller pada laravel, yaitu bagian menerima permintaan dari rute, memproses input (tempat logic), dan mengembalikan respons ke user.

1. Dasar Teori

- Apa itu Controller?
 - Controller adalah salah satu komponen inti dari MVC(Model, View, Controller) yang berfungsi sebagai penghubung antara user (View) ke model yang nantinya akan di kembalikan lagi ke View dalam bentuk response. Controller ini akan banyak berisi logika – logika dalam menyusun suatu fungsi tertentu. Contohnya adalah aktivitas CRUD (Create, Read, Update, Delete) yang prosesnya berjalan di dalam Controller.
 - Perintah artisan untuk membuat controller yaitu
 - `php artisan make:controller PageController`
 - Untuk menggunakan Controller tersebut maka kita perlu mendaftarkannya di dalam file route:
 - `use App\Http\Controllers\PageController; Route::get('/home', [PageController::class, 'home']);`
 - Jenis - jenis Controller
 - Resource Controller
 - Resource dirancang berguna untuk jika ingin menggunakan operasi CRUD secara efisien, dimana laravel akan menyediakan metode bawaan seperti index, create, store, show, edit, update, dan destroy yang bisa langsung digunakan
 - Invokable Controller
 - Invokable controller cocok untuk tugas yang hanya membutuhkan satu metode, contohnya seperti menghitung total diskon atau mengirim email.
 - Nested Controller
 - Nested Controller digunakan untuk mengelola relasi antar entitas, seperti review untuk setiap buah.
 - Custom Controller
 - Controller ini digunakan untuk kebutuhan khusus, seperti menampilkan penawaran spesial.
-

2. Langkah-Langkah Praktikum

- 2.1 Praktikum 1 - Menangani Request dan Response View di Laravel 12

- 1. Buat project baru di dalam folder projects dengan nama lab-view, buat dengan perintah composer seperti ini:
 - `composer create-project laravel/laravel lab-view`
- 2. Selanjutnya kita buat controller di terminal/git bash, dengan nama controllernya adalah WelcomeController. Untuk membuat controller kita dapat gunakan perintah:
 - `php artisan make:controller WelcomeController`
- 3. Setelah itu buka file controller yang telah dibuat di `app/http/controller/WelcomeController`, lalu edit kode program di file tersebut seperti

```
projects > lab-view > 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' => $name]);
14     }
15
16     //Parameterized route
17     public function greet($name)
18     {
19         return view('greet', ['name' => ucfirst($name)]);
20     }
21
22     //Query String
23     public function search(Request $request)
24     {
25         $keyword = $request->query('q', 'none');
26         return view('search', ['keyword' => $keyword]);
27     }
28 }
```

gambar dibawah ini:

- 4. selanjutnya kita akan mendefinisakan rute, dengan cara cari file web.php pada folder routes, lalu edit kode program di dalam file web.php seperti gambar di bawah ini:

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

- 5. Di `resources/views` kita akan membuat 3 file blade untuk view yang sederhana. beri nama file tersebut adalah:
 - `hello.blade.php`
 - `greet.blade.php`
 - `search.blade.php`

- 6. selanjutnya kita akan mengedit atau menambahkan kode program di masing masing file blade.

```
projects > lab-view > 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>
```

- hello.blade.php

```
projects > lab-view > resources > views > greet.blade.php > html
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>
```

- greet.blade.php

- search.blade.php

```
projects > lab-view > 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>
```

- 7. Setelah melakukan langkah 1 hingga 6, maka pada langkah terakhir yaitu menjalankan server laravel dengan perintah `php artisan serve`.
- 8. Buka browser lalu ketik kunjungi url url ini
 - <http://127.0.0.1:8000/hello>, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:



Hello, Laravel Learner!

- <http://127.0.0.1:8000/greet/ilham>, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:



Nice to meet you, Ilham!

- <http://127.0.0.1:8000/search?q=laravel>, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:



You searched for: laravel

-
- 2.2 Praktikum 2 – Menggunakan Group Route
 - 1. Buat project baru di dalam folder projects dengan nama lab-group, buat dengan perintah composer seperti ini:

- `composer create-project laravel/laravel lab-group`
- 2. Selanjutnya kita buat controller di terminal/git bash, dengan nama controllernya adalah PageController. Untuk membuat controller kita dapat gunakan perintah:
- `php artisan make:controller PageController`
- 3. Setelah itu buka file controller yang telah dibuat di `app/http/controller/PageController`, lalu edit kode program di file tersebut seperti gambar dibawah ini:

```

projects > lab-group > 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

```

- 4. selanjutnya kita akan mendefinisakan rute yang dikelompokkan, dengan cara cari file `web.php` pada folder `routes`, lalu edit kode program di dalam file `web.php` seperti gambar

```

projects > lab-group > routes > web.php > ...
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\PageController;
5
6  // Route::get('/', function () {
7  //     return view('welcome');
8  // });
9
10 Route::controller(PageController::class)->group(function(){
11
12     Route::get('/', 'home')->name('home');
13     Route::get('/about', 'about')->name('about');
14     Route::get('/contact', 'contact')->name('contact');
15 });
16

```

di bawah ini:

- 5. Di `resources/views` buat folder baru bernama `pages` lalu kita akan membuat 3 file blade untuk view yang sederhana. beri nama file tersebut adalah:
- `about.blade.php`
- `contact.blade.php`
- `home.blade.php`

- 6. selanjutnya kita akan mengedit atau menambahkan kode program di masing masing file blade.

```
projects > lab-group > 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>
```

- about.blade.php

```
projects > lab-group > 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>
```

- contact.blade.php

```
projects > lab-group > 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>
```

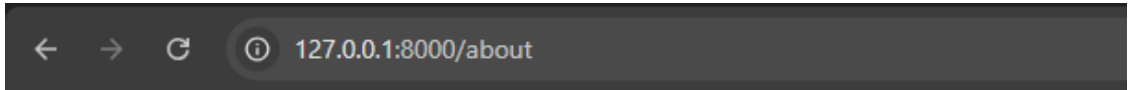
- home.blade.php

- 7. Setelah melakukan langkah 1 hingga 6, maka pada langkah terakhir yaitu menjalankan server laravel dengan perintah `php artisan serve`.
- 8. Buka browser lalu ketik kunjungi url url ini
 - `http://127.0.0.1:8000/`, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:



Welcome to the homepage.

- `http://127.0.0.1:8000/about`, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:



This is the about page.

- `http://127.0.0.1:8000/contact`, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:



reach us through the contact page.

-
- 2.3 Praktikum 3 – Pengelompokan Prefix dengan Namespace Rute di Laravel 12
 - 1. Buat project baru di dalam folder projects dengan nama lab-prefix, buat dengan perintah composer seperti ini:
 - `composer create-project laravel/laravel lab-prefix`
 - 2. Selanjutnya kita buat folder admin Ini akan membuat dua controller di direktori `app/Http/Controllers/`, dengan nama controllernya adalah `UserController` dan `AdminController`. Untuk membuat controllernya kita dapat gunakan perintah:
 - `php artisan make:controller Admin/UserController`
 - `php artisan make:controller Admin/DashboardController`
 - 3. Setelah itu buka file controller yang telah dibuat di `app/http/controller/Admin`, lalu edit kode program di file tersebut seperti gambar dibawah ini:

```

projects > lab-prefix > 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         $users = ['Ria', 'Lie', 'Jhon'];
13         return view('admin.users.index', compact('users'));
14     }
15
16     public function show($id)
17     {
18         $user = "User #" . $id;
19         return view('admin.users.show', compact('user'));
20     }
21 }

```

- UserController

- DashboardController

```

projects > lab-prefix > 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 }
15

```

- 4. selanjutnya kita akan mendefinisakan rute yang dikelompokkan dengan prefix dan namespace controller, dengan cara cari file web.php pada folder routes, lalu edit kode program di dalam file web.php seperti gambar di bawah ini:

```

projects > lab-group > routes > web.php > ...
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\PageController;
5
6  // Route::get('/', function () {
7  //     return view('welcome');
8  // });
9
10 Route::controller(PageController::class)->group(function(){
11
12     Route::get('/', 'home')->name('home');
13     Route::get('/about', 'about')->name('about');
14     Route::get('/contact', 'contact')->name('contact');
15 });
16

```


- 5. Di `resources/views` buat folder baru bernama `admin` lalu di folder `admin` kita akan membuat 1 file blade dan 1 folder `users` untuk view yang sederhana. beri nama file `dashboard.blade.php` di folder `admin`, dan isikan kode ini:

```
projects > lab-prefix > resources > views > admin > 📄 dashb
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Admin Dashboard</title>
5      </head>
6      <body>
7          <h1>{{ $message }}</h1>
8      </body>
9  </html>
```

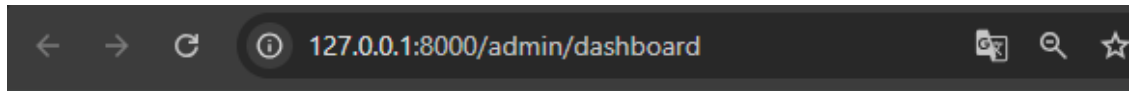
- 6. Di `resources/views/Admin/users` buat 2 file yaitu:
 - `index.blade.php` dan isikan kode berikut:

```
projects > lab-prefix > resources > views > admin > users > 📄 i
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>
```

- `show.blade.php`

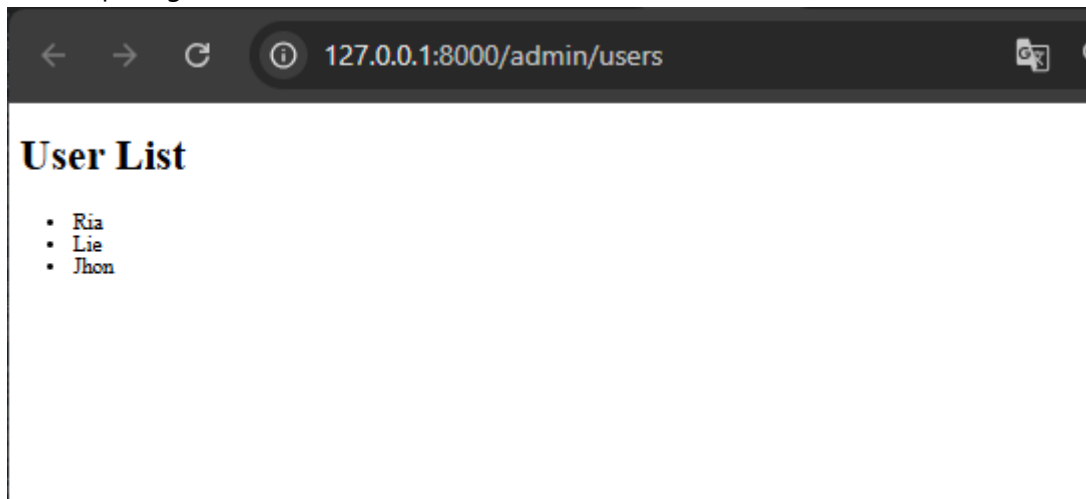
```
projects > lab-prefix > resources > views > admin > users > 📄 show.blade
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>User Details</title>
5      </head>
6      <body>
7          <h1>Details for: {{ $user }}</h1>
8      </body>
9  </html>
```

- 7. Setelah melakukan langkah 1 hingga 6, maka pada langkah terakhir yaitu menjalankan server laravel dengan perintah `php artisan serve`.
- 8. Buka browser lalu ketik kunjungi url url ini
 - `http://127.0.0.1:8000/admin/dashboard`, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:

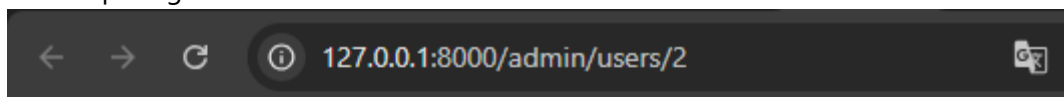


welcome to Admin Dashboard

- <http://127.0.0.1:8000/admin/users>, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:



- <http://127.0.0.1:8000/admin/users/2>, maka nanti akan keluar hasil yang sudah kita coba seperti gambar dibawah ini:



Details for: User #2

3. Hasil dan Pembahasan

- Apakah aplikasi berjalan sesuai harapan?
 - Alhamdulillah hasil dari praktikum ini sesuai harapan meskipun menjumpai beberapa error yang disebabkan karena terlalu terburu-buru sehingga beberapa kali typo.

- Bagaimana pengelompokan rute di Laravel 12 bekerja?
 - Pengelompokan rute di Laravel 12 itu fungsinya supaya rute yang mirip-mirip bisa digabung jadi satu. Misalnya semua rute admin pakai prefix /admin dan middleware auth, jadi nggak perlu nulis berulang. Cukup pakai `Route::prefix('admin')->middleware('auth')->group(function () { ... });`, nanti semua rute di dalamnya otomatis ikut aturan itu. Intinya, biar kode lebih rapi, gampang diatur, dan nggak ribet kalau mau ubah prefix.
- Apa itu prefix?
 - Prefix itu kayak tambahan di depan URL yang dipakai bareng-bareng beberapa rute. Misalnya kamu bikin `prefix('admin')`, berarti semua rute di dalam grup itu URL-nya bakal diawali /admin, contohnya pada praktikum ini yaitu di praktikum 3 yaitu <http://127.0.0.1:8000/admin/users/2>, yang prefix itu bagian **admin**.

4. Kesimpulan

- Kesimpulan dari praktikum ini adalah tentang cara kerja controller di Laravel 12, mulai dari pembuatan, pengelolaan rute, pengiriman data ke view, jenis jenis controller, hingga validasi dan respons. Melalui pembahasan dan praktik yang dilakukan, kita belajar bagaimana mengatur struktur aplikasi agar lebih rapi dan efisien, serta mempelajari cara mengelompokkan rute untuk organisasi yang lebih baik. yang nantinya menjadi bekal penting untuk mempelajari fitur Laravel yang lebih kompleks di bab selanjutnya.

5. Referensi

- <https://buildwithangga.com/tips/apa-itu-controller-pada-laravel>
- <https://informatika.ciputra.ac.id/2019/10/laravel-controller/>
- <https://hackmd.io/@mohdrzu/H1sB73dnxg>