

Laporan Modul 3: Laravel Controller

Mata Kuliah: Workshop Web Lanjut

Nama: Bunga Alfa Zahrah **NIM:** 2024573010023 **Kelas:** TI-2C

Abstrak

- Laporan ini membahas penerapan Controller pada framework Laravel 12 sebagai bagian dari pola arsitektur Model-View-Controller (MVC). Tujuan dari praktikum ini adalah untuk memahami peran controller dalam mengelola logika aplikasi, menangani request dan response, serta menghubungkan antara route, model, dan view.
 - Melalui tiga percobaan utama—menangani request dan response, membuat aplikasi sederhana menggunakan controller, serta mengelompokkan route dengan prefix dan namespace—praktikan mempelajari bagaimana struktur aplikasi Laravel dapat diorganisir secara efisien.
 - Hasil dari praktikum menunjukkan bahwa penggunaan controller dalam Laravel mempermudah pengelolaan logika program, meningkatkan keteraturan kode, serta mendukung pengembangan aplikasi web yang terstruktur dan mudah dipelihara.
-

1. Dasar Teori

- Apa itu Controller
 - 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.
 - Jenis-Jenis Controller(Pada Laravel) Dalam laravel mendukung beberapa jenis controller untuk mendukung berbagai perkembangan dalam web
 - 1.Basic Controllers(Controller dasar) Controller standar yang memiliki beberapa metode publik yang diantaranya(index, show, store, update, destroy)
 - 2.Resource Controller(Controller sumber daya) Controller yang dirancang untuk operasi CRUD pada sumber daya(resource) dengan metode-metode standar seperti index, create, store, show, edit, update, destroy Contoh pembuatannya hampir sama dengan yg dasar cuman ada tambahan "--resource" di akhir
 - 3.Invokable Controller(Single-action Controller) digunakan ketika sebuah controller hanya melakukan satu tugas tunggal dan tidak perlu banyak metode.
 - Apa itu view,json,redirect,respon kostum dalam laravel
 - View digunakan untuk menampilkan tampilan(file.blade.php) ke pengguna biasanya dipakai untuk aplikasi web berbasis antarmuka HTML
 - JSON response()->json() digunakan untuk mengirim data dalam format JSON, biasanya untuk API atau AJAX request
 - redirect() digunakan untuk mengalihkan (memindahkan) pengguna ke route atau URL lain.
-

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

- Tambahkan route pada routes/web.php. Editkan di routes/web.php menjadi seperti pada gambar berikut.

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DemoController;

Route::get('/hello', [DemoController::class, 'hello']);
Route::get('/greet/{name}', [DemoController::class, 'greet']);
Route::get('/search', [DemoController::class, 'search']);

Route::get('/', function () {
    return view('welcome');
});
```

dan itu mengarah ke Demo controller , kemudian buat controller nya(DemoController).

- Buat controller DemoController.php Create DemoController pada GitBash yang ada pada vscode lalu ketikkan sesuai dengan gambar berikut

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class DemoController extends Controller
{
    Tabnine | Edit | Test | Explain | Document
    public function hello(){
        $name = 'Laravel Learner';
        return view('hello', ['name' => $name]);
    }
    Tabnine | Edit | Test | Explain | Document
    public function greet($name){
        return view('greet', ['name' => ucfirst($name)]);
    }
    Tabnine | Edit | Test | Explain | Document
    public function search(Request $request){
        $keyword = $request->query('q', 'none');
        return view('search', ['keyword' => $keyword]);
    }
}

```

- Buat view sederhana pada resources/views. buat file blade ini pada resources/views: - hello.blade.php
Buatlah code nya seperti berikut:

```

<!DOCTYPE html>
<html>
    <head>
        <title>Hello</title>
    </head>
    <body>
        <h1>Hello, {{ $name }}!</h1>
    </body>
</html>

```

- greet.blade.php Buatlah

code nya seperti berikut:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Greet</title>
  </head>
  <body>
    <h1>Nice to meet you, {{ $name }}!</h1>
  </body>
</html>
```

search.blade.php Buatlah code nya seperti berikut:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Search</title>
  </head>
  <body>
    <h1>You searched for: <strong>{{ $keyword }}</strong></h1>
  </body>
</html>
```

- Jalankan aplikasi dan tunjukkan hasil di browser. <http://127.0.0.1:8000/hello> Tampilan Browser

Hello, Laravel Learner!

<http://127.0.0.1:8000/greet/Bunga> (Pada url ini bebas tergantung codingannya yang anda inputkan.)
Tampilan Browser

Nice to meet you, Bunga!

<http://127.0.0.1:8000/search?q=laravel> Tampilan Browser

You searched for: laravel

2.2 Praktikum 2 – Membuat Aplikasi Sederhana "Calculator"

- Tambahkan route PageController dalam web.php. Editkan di routes/web.php menjadi seperti pada gambar berikut.

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\PageController;

Route::controller(PageController::class)->group(function () {
    Route::get('/', 'home')->name('home');
    Route::get('/about', 'about')->name('about');
    Route::get('/contact', 'contact')->name('contact');
});
```

- Buat controller PageController. Buat PageController pada Gitbash, yg tersimpan pada folder "app/http/controller", lalu buat code nya seperti ini

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class PageController extends Controller
{
    public function home(){
        $message = "Welcome to the homepage.";
        return view('pages.home', compact('message'));
    }
    public function about(){
        $message = "This is the about page.";
        return view('pages.about', compact('message'));
    }
    public function contact(){
        $message = "Reach us through the contact page.";
        return view('pages.contact', compact('message'));
    }
}
```

- Tambahkan view home.blade.php, about.blade, contact blade.php buat file blade ini pada resources/views, lalu buat folder baru dalam views pages menjadi resources/views/pages

- about.blade.php

Buatlah code nya seperti berikut:

```
<!DOCTYPE html>
<html>
  <head>
    <title>About</title>
  </head>
  <body>
    <h1>{{ $message }}</h1>
  </body>
</html>
```

- contact.blade.php Buatlah code nya

```
<!DOCTYPE html>
<html>
  <head>
    <title>Contact</title>
  </head>
  <body>
    <h1>{{ $message }}</h1>
  </body>
</html>
```

seperti berikut:

- home.blade.php Buatlah

```
<!DOCTYPE html>
<html>
  <head>
    <title>Home</title>
  </head>
  <body>
    <h1>{{ $message }}</h1>
  </body>
</html>
```

code nya seperti berikut:

- Jalankan aplikasi dan coba dengan beberapa input berbeda. <http://127.0.0.1:8000/> home Tampilan Browser :

Welcome to the homepage.

http://127.0.0.1:8000/about Tampilan Browser :

This is the about page.

http://127.0.0.1:8000/contact Tampilan Browser :

Reach us through the contact page.

2.3 Praktikum 3 – Pengelompokan Prefix dengan namespace Rute di Laravel 12

- Tambahkan route untuk prefix dan namespace Controller Editkan di routes/web.php menjadi seperti ini.

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Admin\DashboardController;
use App\Http\Controllers\Admin\UserController;

Tabnine | Edit | Test | Explain | Document
Route::prefix('admin')->group(function () {
    Route::get('/dashboard', [DashboardController::class, 'index']);
    Route::get('/users', [UserController::class, 'index']);
    Route::get('/user/{id}', [UserController::class, 'show']);
});

Tabnine | Edit | Test | Explain | Document
Route::get('/', function () {
    return view('welcome');
});
```

- Buat controller DashboardController dan user. ini akan membuat dua controller di direktori app/Http/Controllers/Admin
- Tambahkan aksi ke controller membuat dashboardcontroller dan usercontroller

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;

class DashboardController extends Controller
{
    Tabnine | Edit | Test | Explain | Document
    public function index()
    {
        return view('admin.dashboard', ['message' => 'Welcome to the Admin Dashboard']);
    }
}
```

o

```
<?php

namespace App\Http\Controllers\Admin;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
class UserController extends Controller
{
    Tabnine | Edit | Test | Explain | Document
    public function index(){
        $users = ['Ria', 'Lie', 'Jon'];
        return view('admin.users.index', compact('users'));
    }
    Tabnine | Edit | Test | Explain | Document
    public function show($id){
        $user = "User #" . $id;
        return view('admin.users.show', compact('user'));
    }
}
```

o

- Buat view sederhana buatkan di folder resources/views/admin/user dan ada 3 file

1. index editkan code nya seperti ini :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Users</title>
  </head>
  <body>
    <h1>User List</h1>
    <ul>
      @foreach ($users as $user)
        <li>{{ $user }}</li>
      @endforeach
    </ul>
  </body>
</html>
```

2. show editkan code nya seperti ini :

```
<!DOCTYPE html>
<html>
  <head>
    <title>User Details</title>
  </head>
  <body>
    <h1>Details for: {{ $user }}</h1>
  </body>
</html>
```

3. dashboard editkan code nya seperti ini :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Admin Dashboard</title>
  </head>
  <body>
    <h1>{{ $message }}</h1>
  </body>
</html>
```

- Jalankan aplikasi `http://127.0.0.1:8000/admin/dashboard` Output:

Welcome to the Admin Dashboard

User List

- Ria
- Lie
- Jon

`http://127.0.0.1:8000/admin/users` Output:

`http://127.0.0.1:8000/admin/users/2` Output:!

Details for: User #2

3. Hasil dan Pembahasan

Jelaskan apa hasil dari praktikum yang dilakukan.

- Apakah aplikasi berjalan sesuai harapan? Pada praktikum ini, ketiga praktikum aplikasi nya berjalan sesuai harapan saya, tidak ada kendala
- Apa peran masing-masing komponen (Route, Controller, View) dalam program yang dibuat? Peran route: Menentukan arah alur permintaan (request) ke bagian aplikasi yang sesuai — biasanya ke Controller. Peran Controller: Mengatur logika aplikasi — menjadi penghubung antara Route (input dari user) dan View (tampilan ke user). Peran View: Menampilkan hasil akhir ke pengguna (biasanya dalam bentuk HTML).

4. Kesimpulan

- Dalam bab ini, kita telah membahas dasar-dasar controller di Laravel 12, termasuk cara membuatnya, mendefinisikan rute, dan meneruskan data ke view. Kita juga mengeksplorasi berbagai jenis controller, seperti resource controller dan invokable controller, serta mempelajari cara mengelompokkan rute untuk organisasi yang lebih baik.
- Kita juga membahas cara menangani permintaan, memvalidasi input, dan mengembalikan berbagai jenis respons. Terakhir, kita menyelesaikan lab praktik untuk memperkuat konsep-konsep ini dengan membuat aplikasi web sederhana yang mendemonstrasikan penanganan permintaan dan pengelompokan rute.

5. Referensi

- <https://laravel.com/docs/12.x/controllers>
 - <https://santrikoding.com/tutorial-laravel-12-2-cara-install-laravel-12>
 - <https://dev.to/robin-ivi/laravel-12-a-beginner-friendly-guide-3of>
 - <https://hmsifmipauntan.com/laravel-12-evolusi-framework-php-untuk-pengembangan-aplikasi-modern>
-