

# Laporan Modul 3: Laravel Controller

---

**Mata Kuliah:** Workshop Web Lanjut

**Nama:** Maila Aziza **NIM:** 2024573010024 **Kelas:** TI 2C

---

## Abstrak

Laporan ini bertujuan untuk membahas tentang controller yang sangat penting untuk menangani logika aplikasi dalam laravel. Controller berfungsi untuk menerima rute, memproses input, dan mengembalikan respons.

---

## 1. Dasar Teori

- Apa itu Controller Controller Mengatur alur antara model dan view. Controller menerima input dari pengguna melalui route, memprosesnya, dan mengembalikan view atau respons yang sesuai.
- Jenis-jenis Controller pada Laravel laravel mendukung beberapa jenis controller:
  1. Basic Controllers
  2. Resource Controllers
  3. Invokable Controllers
  4. Pengelompokan Rute dengan Controller
  5. Injeksi Permintaan dan Injeksi Ketergantungan
  6. Validasi Permintaan dalam Controller
  7. Mengembalikan Respons dari Controller

## 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

- Buat controller pada 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 }
```

- Definisikan Route

```
<?php

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

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

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

- Buat view sederhana.

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

1. hello.blade.php

2. greet.blade.php

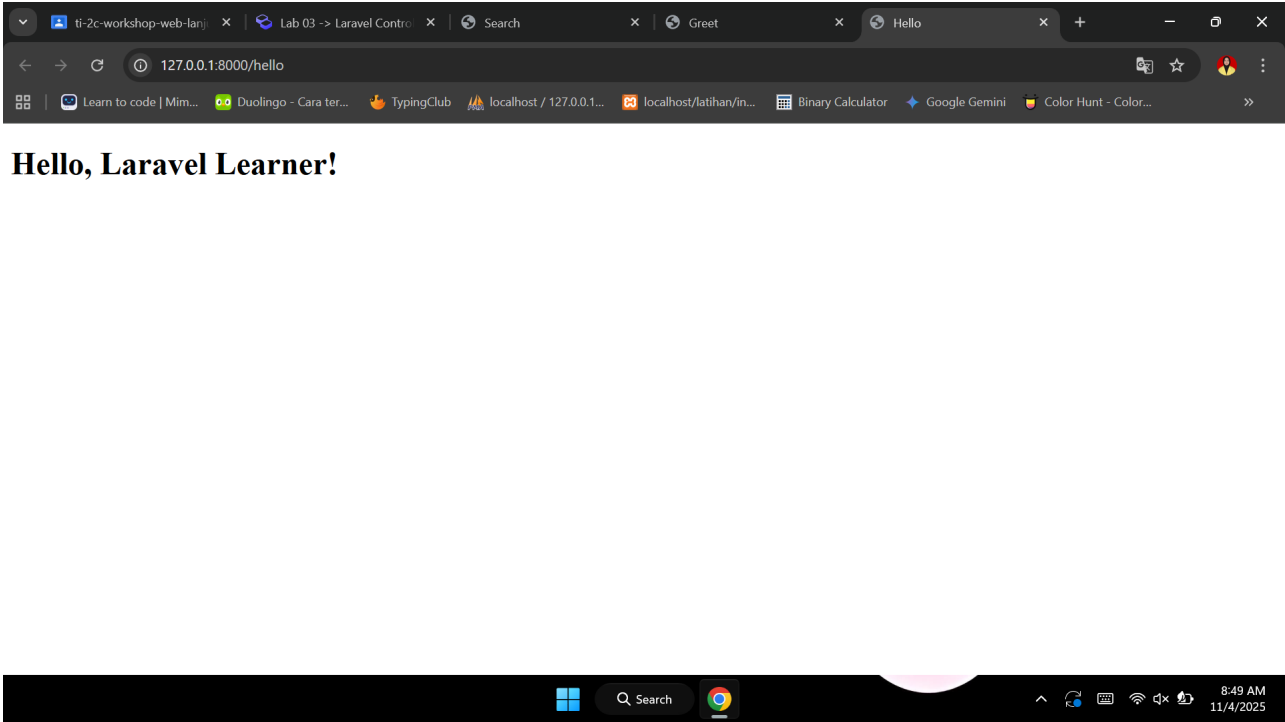
```
<!DOCTYPE html>
<html>
|   <head>
|       <title>Greet</title>
|   </head>
|   <body>
|       <h1>Nice to meet you, {{ $name }}!</h1>
|   </body>
</html>
```

3. search.blade.php

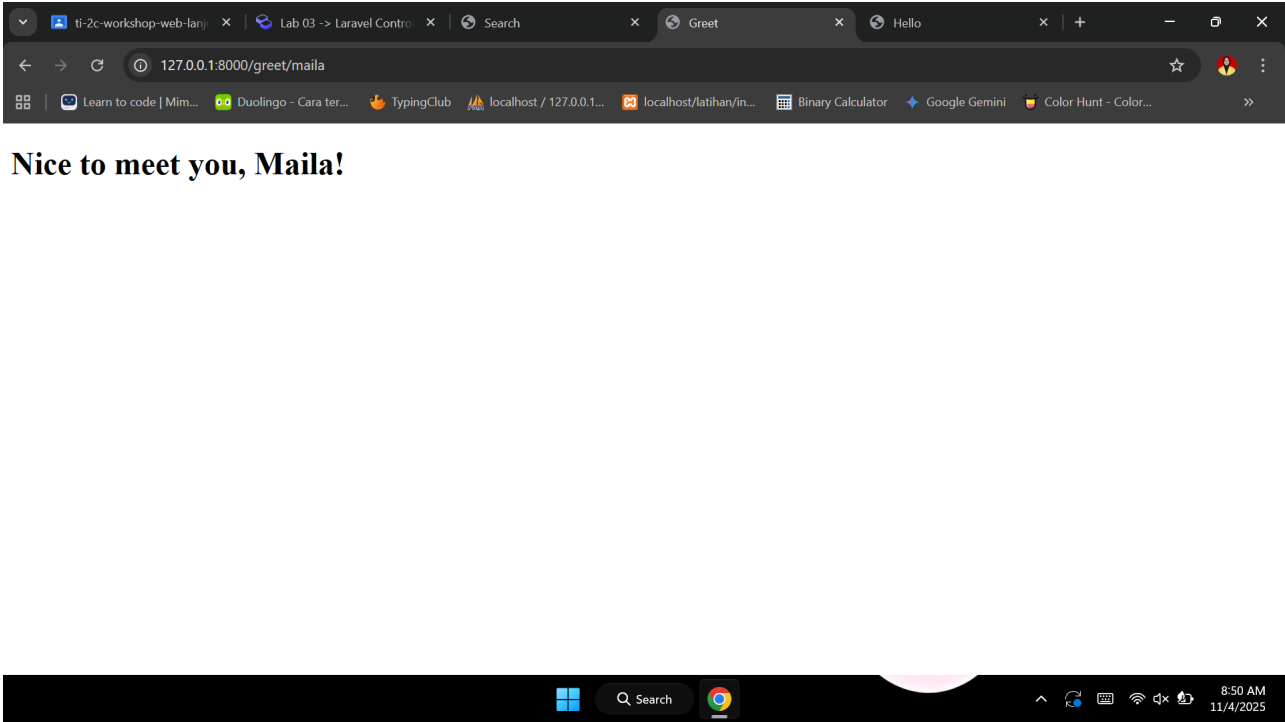
```
<!DOCTYPE html>
<html>
|   <head>
|       <title>Search</title>
|   </head>
|   <body>
|       <h1>You searched for: <strong>{{ $keyword }}</strong></h1>
|   </body>
</html>
```

- Jalankan aplikasi dan tunjukkan hasil di browser. Screenshot Hasil:

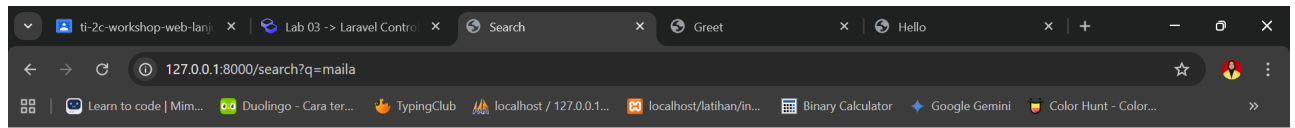
1. hello



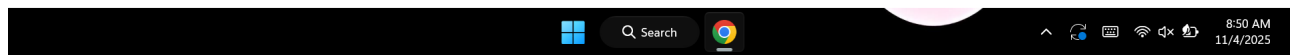
2. greet



### 3. search



**You searched for: maila**



## 2.2 Praktikum 2 – Menggunakan Group Route

- Buat controller pada app/Http/Controllers/PageController.php.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PageController extends Controller
{
    public function home()
    {
        $message = "Welcome to the Home Page!";
        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'));
    }
}
```

- Definisikan Route

```
<?php

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

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

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

- Buat view sederhana.

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

1. home.blade.php

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

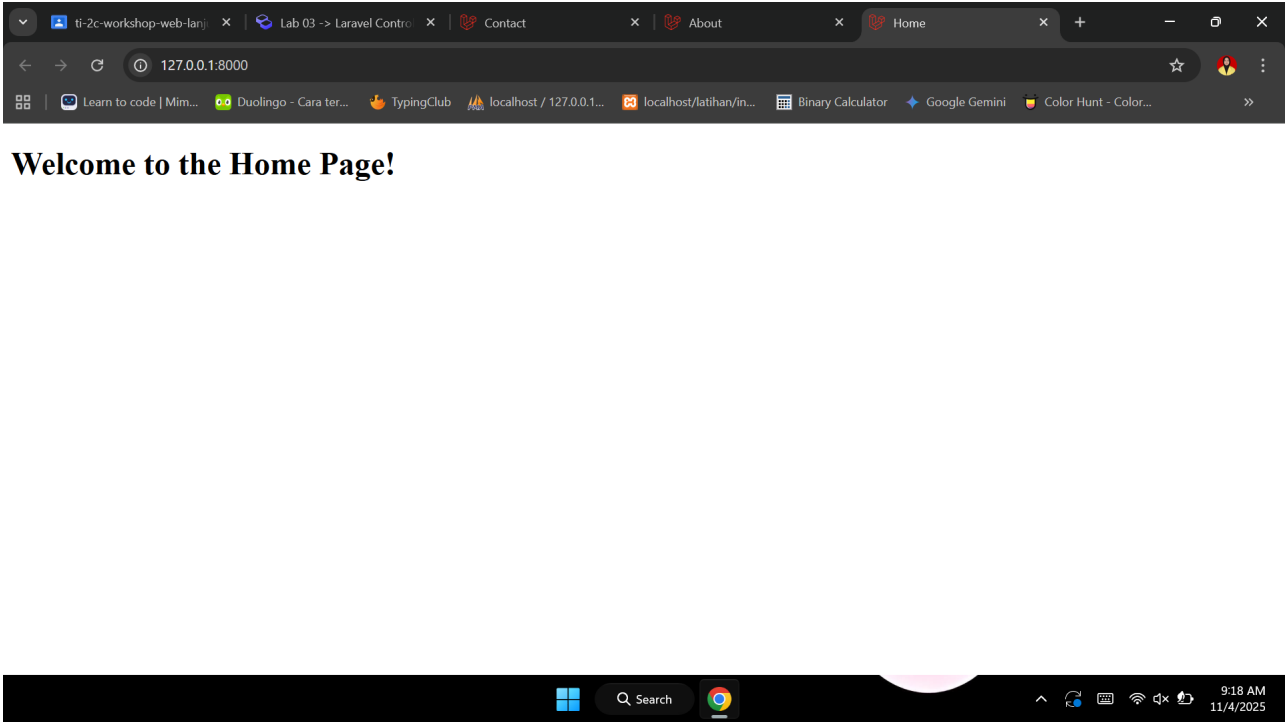
2. about.blade.php

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

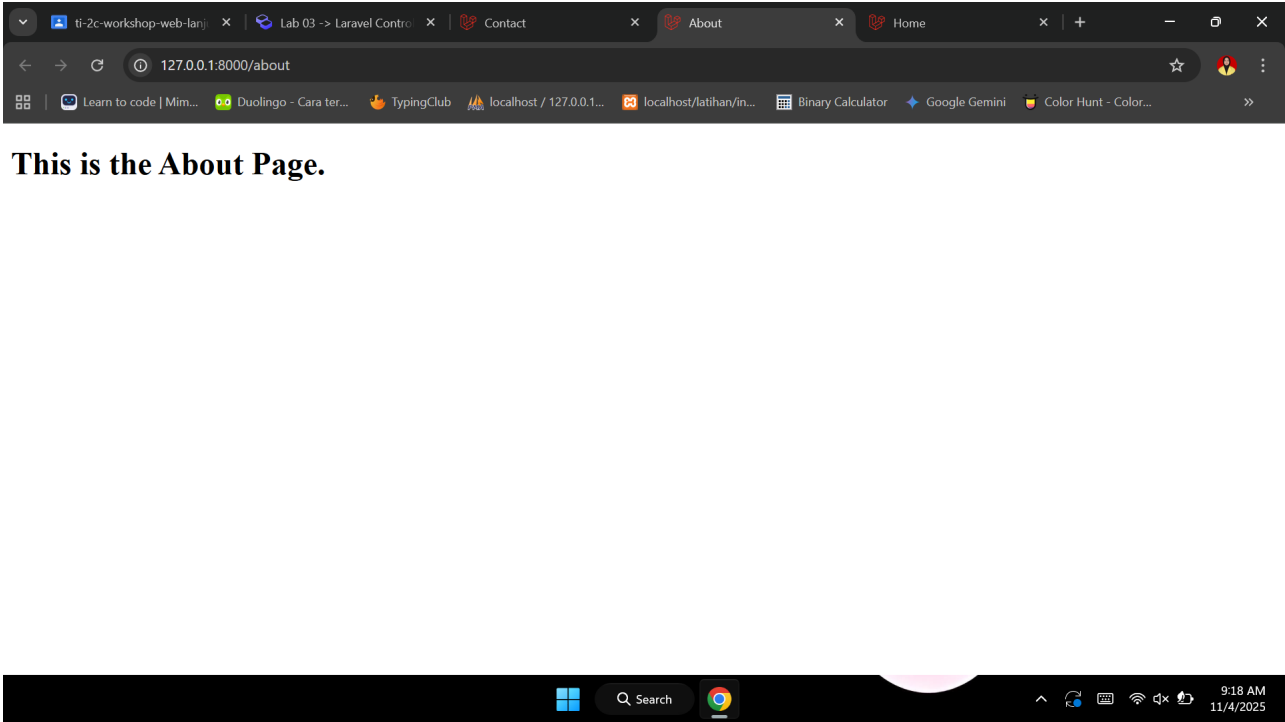
3. contact.blade.php

- Jalankan aplikasi dan tunjukkan hasil di browser. Screenshot Hasil:

1. home

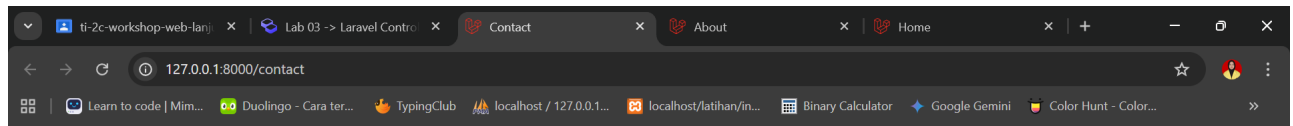


2. about





### 3. contact



**Reach us through the Contact Page.**

## 2.3 Praktikum 3 – Pengelompokan Prefix dengan Namespace Rute di Laravel 12

- Definisikan Kelompok Rute dengan Prefix dan Namespace Controller

```
<?php

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

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

Route::prefix('admin')->group(function(){
    Route::get('/dashboard', [DashboardController::class, 'index'])->name('admin.dashboard');
    Route::get('/users', [UserController::class, 'index'])->name('admin.users');
    Route::get('/users/{id}', [UserController::class, 'show'])->name('admin.users.show');
});
```

- Tambahkan Aksi ke Controller Kita akan membuat aksi sederhana yang mengembalikan view dengan pesan selamat datang. Berikut kode untuk

## 1. DashboardController.php:

```
<?php

namespace App\Http\Controllers\Admin;

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

class DashboardController extends Controller
{
    public function index()
    {
        return view('admin.dashboard', ['message' => 'welcome to Admin Dashboard']);
    }
}
```

## 2. UserController.php:

```
<?php

namespace App\Http\Controllers\Admin;

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

class UserController extends Controller
{
    public function index()
    {
        $users = ['Ria', 'Lie', 'Jhon'];
        return view('admin.users.index', compact('users'));
    }

    public function show($id)
    {
        $user = "User #" . $id;
        return view('admin.users.show', compact('user'));
    }
}
```

- Buat view sederhana.

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

1. dashboard.blade.php

```
<!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>
```

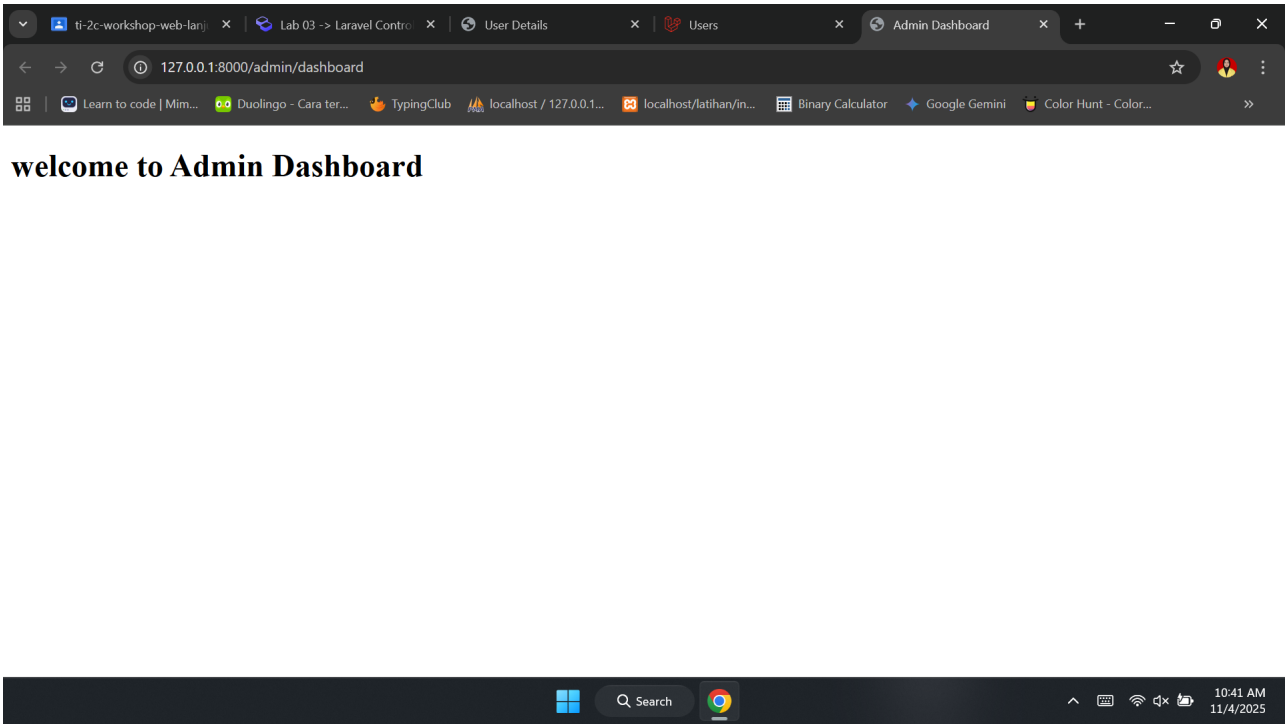
2. users/index.blade.php

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

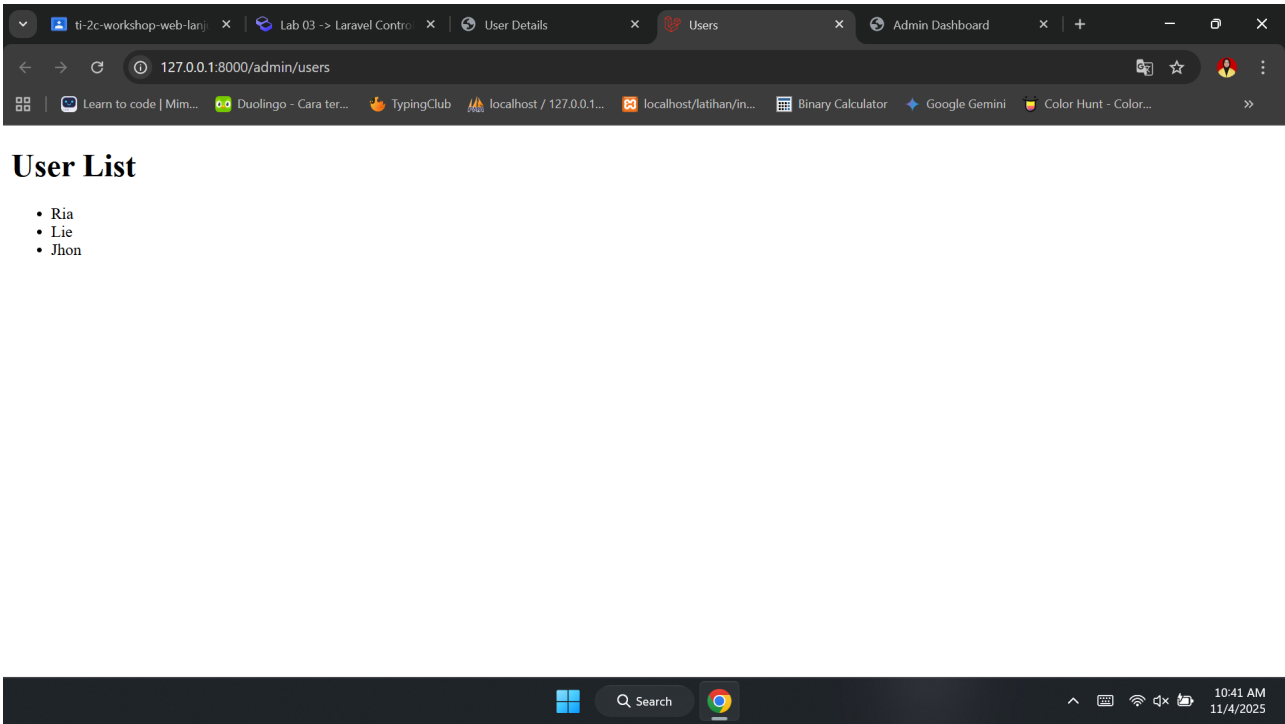
3. users/show.blade.php

- Jalankan aplikasi dan tunjukkan hasil di browser. Screenshot Hasil:

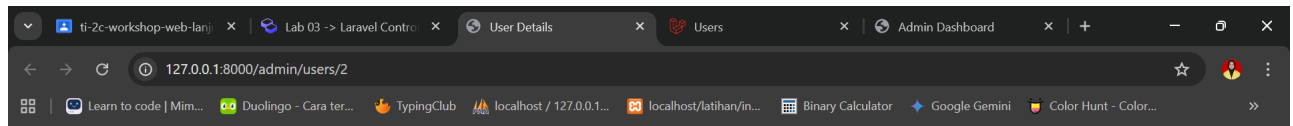
1. Admin Dashboard



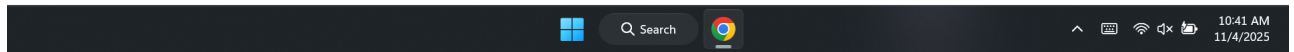
2. Users



### 3. User Details



#### Details for: User #2



### 3. Hasil dan Pembahasan

- Aplikasi berhasil dijalankan dan semua route yang didefinisikan dapat diakses melalui browser dan menampilkan tampilan yang diinginkan.
- Controller mengelola data dan meneruskannya ke View. Data yang dikirim dari controller dapat ditampilkan pada halaman view menggunakan sintaks Blade seperti `{{ $variable }}`.
- Pengelompokan Route berhasil dilakukan dengan Prefix dan Namespace. Dengan menggunakan prefix dan namespace, struktur route menjadi lebih terorganisir dan memudahkan pengelolaan banyak halaman.
- Penggunaan Blade sebagai template engine memudahkan pembuatan tampilan. Blade memberikan fitur seperti `@extends`, `@section`, dan `@include` yang membuat kode lebih modular dan mudah dirawat.

### 4. Kesimpulan

Dalam praktikum ini mempelajari cara 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. Membahas cara menangani permintaan, memvalidasi input, dan mengembalikan berbagai jenis respons.

### 5. Referensi

Laravel Controller — <https://hackmd.io/@mohdrzu/H1sB73dnxg#Modul-3---Laravel-Controller> Bagaimana Grup Rute Laravel Dapat Menyederhanakan dan Mengatur Kode Anda — <https://redberry.international/laravel-route-group-organize-your-code/>

