

LAPORAN JOBSHEET 2

ROUTING, CONTROLLER, DAN VIEW

MATA KULIAH PEMROGRAMAN WEB LANJUT

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Disusun oleh :

Dahnier Davina SIB-2A / 2341760023

JURUSAN TEKNOLOGI INFORMASI

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

POLITEKNIK NEGERI MALANG


2025

A. Langkah Praktikum Basic Routing

1. Buka file routes/web.php. Tambahkan sebuah route untuk nomor 1 seperti di bawah ini:

```
Route::get('/hello', function () {  
    return 'Hello World';  
});  
//menggunakan controller
```

2. Buka browser, tuliskan URL untuk memanggil route tersebut: localhost/PWL_2025/public/hello. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.




Hello World

3. Untuk membuat route kedua, tambahkan route /world seperti di bawah ini

```
Route::get('/world', function () {  
    return 'World';  
});
```


4. Bukalah pada browser, tuliskan URL untuk memanggil route tersebut: localhost/PWL_2025/public/world. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.



World

5. Selanjutnya, cobalah membuat route '/' yang menampilkan pesan 'Selamat Datang'


```
Route::get('/opening', function () {  
    return 'Selamat Datang';  
});
```



Selamat Datang

6. Kemudian buatlah route '/about' yang akan menampilkan NIM dan nama Anda

```
Route::get('/about', function () {  
    return 'Dahniar Davina dengan NIM 23417600243';  
});
```



Dahniar Davina dengan NIM 23417600243

B. Langkah Praktikum Route Parameters

1. Kita akan memanggil route `/user/{name}` sekaligus mengirimkan parameter berupa nama user `$name` seperti kode di bawah ini.

```
Route::get('/user/{name}', function ($name) {  
    return 'Nama saya '.$name;  
});
```

Nama saya {name}

2. Jalankan kode dengan menuliskan URL untuk memanggil route tersebut: `localhost/PWL_2024/public/user>NamaAnda`. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda

Nama saya Dahniar

3. Selanjutnya, coba tuliskan URL: `localhost/PWL_2024/public/user/`. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda

404 | NOT FOUND

4. Suatu route, juga bisa menerima lebih dari 1 parameter seperti kode berikut ini. Route menerima parameter `$postId` dan juga `$comment`.

```
Route::get('/posts/{post}/comments/{comment}', function  
($postId, $commentId) {  
    return 'Pos ke-' . $postId . " Komentar ke-: " . $commentId;  
});
```

5. Jalankan kode dengan menuliskan URL untuk memanggil route tersebut: `localhost/PWL_2024/public/posts/1/comments/5`. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

Pos ke-1 Komentar ke-: 5

6. Kemudian buatlah route `/articles/{id}` yang akan menampilkan output “Halaman Artikel dengan ID {id}”, ganti id sesuai dengan input dari url.

```
Route::get('/articles/{id}', function ($id) {  
    return 'Halaman Artikel dengan ID ' . $id;  
});
```

C. Langkah Praktikum Optional Parameters

1. Kita akan memanggil route /user sekaligus mengirimkan parameter berupa nama user \$name dimana parameternya bersifat opsional.

```
Route::get('/user/{name?}', function ($name=null) {  
    return 'Nama saya '.$name;  
});
```

2. Jalankan kode dengan menuliskan URL: localhost/PWL_2024/public/user/. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

Nama saya

3. Selanjutnya tuliskan URL: localhost/PWL_2024/public/user>NamaAnda. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda

Nama saya Dahniar

4. Ubah kode pada route /user menjadi seperti di bawah ini.

```
Route::get('/user/{name?}', function ($name='John') {  
    return 'Nama saya '.$name;  
});
```

5. Jalankan kode dengan menuliskan URL: localhost/PWL_2024/public/user/. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

Nama saya John

D. Langkah Praktikum Membuat Controllers

1. Untuk membuat controller pada Laravel telah disediakan perintah untuk menggenerate struktur dasarnya. Kita dapat menggunakan perintah artisan diikuti dengan definisi nama controller yang akan dibuat.

```
windows@DESKTOP-PE7RQP7 MINGW64 /c/laragon/www/PWL_2025 (main)  
$ php artisan make:controller WelcomeController  
  
INFO Controller [C:\laragon\www\PWL_2025\app\Http\Controllers\WelcomeController.php] created successfully.
```

2. Buka file pada `app/Http/Controllers/WelcomeController.php`. Struktur pada controller dapat digambarkan sebagai berikut:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class WelcomeController extends Controller
{
    //
}
```


3. Untuk mendefinisikan action, silahkan tambahkan function dengan access public. Sehingga controller di atas menjadi sebagai berikut:

```
class WelcomeController extends Controller
{
    public function hello() {
        return 'Hello World';
    }
}
```

4. Setelah sebuah controller telah didefinisikan action, kita dapat menambahkan controller tersebut pada route. Ubah route `/hello` menjadi seperti berikut:

```
// Controllers //
Route::get('hello', [WelcomeController::class, 'hello']);
```

5. Buka browser, tuliskan URL untuk memanggil route tersebut: `localhost/PWL_2024/public/hello`. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Hello World

6. Modifikasi hasil pada praktikum poin 2 (Routing) dengan konsep controller. Pindahkan logika eksekusi ke dalam controller dengan nama `PageController`.

Resource	POST	GET	PUT	DELETE
/		Tampilkan pesan 'Selamat Datang' PageController : index		
/about		Tampilkan Nama dan NIM PageController : about		

/articles/{id}		Tampilkan halaman dinamis 'Halaman Artikel dengan Id {id}' id diganti sesuai input dari url PageController : articles		
----------------	--	--	--	--

7. Modifikasi kembali implementasi sebelumnya dengan konsep Single Action Controller. Sehingga untuk hasil akhir yang didapatkan akan ada HomeController, AboutController dan ArticleController. Modifikasi juga route yang digunakan

```
Windows@DESKTOP-PE7RQP7 MINGW64 /c:/laragon/www/PWL_2025 (main)
$ php artisan make:controller PageController

INFO Controller [C:\laragon\www\PWL_2025\app\Http\Controllers\PageController.php] created successfully.
```

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PageController extends Controller
{
    public function index() {
        return 'Welcome!';
    }
    public function about() {
        return "Nama: Dahniar Davina <br> NIM: 2341760023";
    }
    public function articles($id) {
        return "Halaman Artikel dengan ID " . $id;
    }
}
```

```
use App\Http\Controllers\PageController; // mengimpor Pagecontroller
```

```
//Page Controller//
Route::get('/', [PageController::class,'index']);
Route::get('/about', [PageController::class,'about']);
Route::get('/articles/{id}', [PageController::class,'articles']);
```

Welcome!

Nama: Dahniar Davina
NIM: 2341760023

Halaman Artikel dengan ID 2341760023

E. Langkah Praktikum Resource Controllers

1. Untuk membuatnya dilakukan dengan menjalankan perintah berikut ini di terminal.

```
$ php artisan make:controller PhotoController --resource  
INFO Controller [C:\laragon\www\PWL_2025\app\Http\Controllers\PhotoController.php] created successfully.
```

Perintah ini akan men-generate sebuah controller dengan nama PhotoController yang berisi method method standar untuk proses CRUD.

2. Setelah controller berhasil degenerate, selanjutnya harus dibuatkan route agar dapat terhubung dengan frontend. Tambahkan kode program berikut pada file web.php.

```
use App\Http\Controllers\PhotoController; // mengimpor Photocontroller
```

```
//Photo Controller//  
Route::resource('photos', [PhotoController::class]);
```

3. Jalankan cek list route (php artisan route:list) akan dihasilkan route berikut ini.

```
C:\laragon\www\PWL_2025>php artisan route:list  
GET|HEAD / ..... PageController@index  
POST _ignition/execute-solution ..... ignition.executeSolution > Spatie\LaravelIgnition > ExecuteSolutionController  
GET|HEAD _ignition/health-check ..... ignition.healthCheck > Spatie\LaravelIgnition > HealthCheckController  
POST _ignition/update-config ..... ignition.updateConfig > Spatie\LaravelIgnition > UpdateConfigController  
GET|HEAD about ..... PageController@about  
GET|HEAD api/user .....  
GET|HEAD articles/{id} ..... PageController@articles  
GET|HEAD hello ..... WelcomeController@hello  
GET|HEAD items ..... items.index > ItemController@index  
POST items ..... items.store > ItemController@store  
GET|HEAD items/create ..... items.create > ItemController@create  
GET|HEAD items/{item} ..... items.show > ItemController@show  
PUT|PATCH items/{item} ..... items.update > ItemController@update  
DELETE items/{item} ..... items.destroy > ItemController@destroy  
GET|HEAD items/{item}/edit ..... items.edit > ItemController@edit  
GET|HEAD opening .....  
GET|HEAD photos ..... photos.index > PhotoController@index  
POST photos ..... photos.store > PhotoController@store  
GET|HEAD photos/create ..... photos.create > PhotoController@create  
GET|HEAD photos/{photo} ..... photos.show > PhotoController@show  
PUT|PATCH photos/{photo} ..... photos.update > PhotoController@update  
DELETE photos/{photo} ..... photos.destroy > PhotoController@destroy  
GET|HEAD posts/{post}/comments/{comment} ..... photos.edit > PhotoController@edit  
GET|HEAD sanctum/csrf-cookie ..... sanctum.csrf-cookie > Laravel\Sanctum > CsrfCookieController@show  
GET|HEAD user/{name?} .....  
GET|HEAD user/{name} .....  
GET|HEAD world .....  
  
Showing [28] routes
```

4. Pada route list semua route yang berhubungan untuk crud photo sudah di generate oleh laravel. Jika tidak semua route pada resource controller dibutuhkan dapat dikurangi dengan mengupdate route pada web.php menjadi seperti berikut ini.

```
C:\laragon\www\PWL_2025>php artisan route:list  
GET|HEAD / ..... PageController@index  
POST _ignition/execute-solution ..... ignition.executeSolution > Spatie\LaravelIgnition > ExecuteSolutionController  
GET|HEAD _ignition/health-check ..... ignition.healthCheck > Spatie\LaravelIgnition > HealthCheckController  
POST _ignition/update-config ..... ignition.updateConfig > Spatie\LaravelIgnition > UpdateConfigController  
GET|HEAD about ..... PageController@about  
GET|HEAD api/user .....  
GET|HEAD articles/{id} ..... PageController@articles  
GET|HEAD hello ..... WelcomeController@hello  
GET|HEAD items ..... items.index > ItemController@index  
POST items ..... items.store > ItemController@store  
GET|HEAD items/create ..... items.create > ItemController@create  
GET|HEAD items/{item} ..... items.show > ItemController@show  
PUT|PATCH items/{item} ..... items.update > ItemController@update  
DELETE items/{item} ..... items.destroy > ItemController@destroy  
GET|HEAD items/{item}/edit ..... items.edit > ItemController@edit  
GET|HEAD photos ..... photos.index > PhotoController@index  
GET|HEAD photos/{photo} ..... photos.show > PhotoController@show  
GET|HEAD photos/{photo}/edit ..... photos.update > PhotoController@update  
GET|HEAD posts/{post}/comments/{comment} ..... photos.destroy > PhotoController@destroy  
GET|HEAD sanctum/csrf-cookie ..... sanctum.csrf-cookie > Laravel\Sanctum > CsrfCookieController@show  
GET|HEAD user/{name?} .....  
GET|HEAD user/{name} .....  
GET|HEAD world .....  
  
Showing [24] routes
```

F. Langkah Praktikum Membuat View

1. Pada direktori app/resources/views, buatlah file hello.blade.php.

```
1 <!-- View pada resources/views/hello.blade.php -->
2 <html>
3 <body>
4 |   <h1>Hello, {{ $name }}</h1>
5 </body>
6 </html>
7
```

2. View tersebut dapat dijalankan melalui Routing, dimana route akan memanggil View sesuai dengan nama file tanpa 'blade.php'. (Catatan: Gantilah Andi dengan nama Anda)

```
/ view route//
Route::get('/greeting', function () {
    return view('hello', ['name' => 'Dahnian']);
});
```

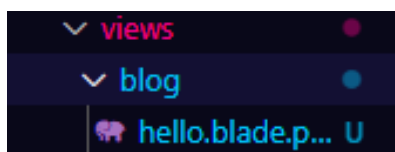
3. Jalankan code dengan membuka url localhost/PWL_2024/public/greeting. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Hello, Dahnian

G. Langkah Praktikum View dalam Direktori

1. Buatlah direktori blog di dalam direktori views.
2. Pindahkan file hello.blade.php ke dalam direktori blog.



3. Selanjutnya lakukan perubahan pada route.

```
//view perubahan //
Route::get('/greeting', function () {
    return view('blog.hello', ['name' => 'Dahnian']);
});
```

4. Jalankan code dengan membuka url localhost/PWL_2024/public/greeting. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Hello, Dahnian

H. Langkah Praktikum Menampilkan View dari Controller


1. Buka WelcomeController.php dan tambahkan fungsi baru yaitu greeting

```
class WelcomeController extends Controller
{
    public function hello() {
        return 'Hello World';
    }
    public function greeting(){
        return view('blog.hello', ['name' => 'Dahniar']);
    }
}
```

2. Ubah route /greeting dan arahkan ke WelcomeController pada fungsi greeting.

```
Route::get('/greeting', [WelcomeController::class, 'greeting']);
```

3. Jalankan code dengan membuka url localhost/PWL_2024/public/greeting.
Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



I. Langkah Praktikum Meneruskan Data ke View

1. Buka WelcomeController.php dan tambahkan ubah fungsi greeting

```
class WelcomeController extends Controller
{
    public function hello() {
        return 'Hello World';
    }
    public function greeting(){
        return view('blog.hello')
            ->with(['name', 'Dahniar'])
            ->with('occupation', 'Astronaut');
    }
}
```

2. Ubah hello.blade.php agar dapat menampilkan dua parameter

```
<!-- View pada resources/views/hello.blade.php -->
<html>
<body>
    <h1>Hello, {{ $name }}</h1>
    <h1>You are {{ $occupation }}</h1>
</body>
</html>
```

3. Jalankan code dengan membuka url localhost/PWL_2024/public/greeting.
Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

Hello, Dahnlar

You are Amazing!