

LAPORAN PRAKTIKUM

Pemrograman Web Lanjut

Jobsheet - 2 : ROUTING, CONTROLLER, DAN VIEW



Nama : Ghoffar Abdul Ja'far

NIM : 2341720035

Kelas : 2F

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023/2024

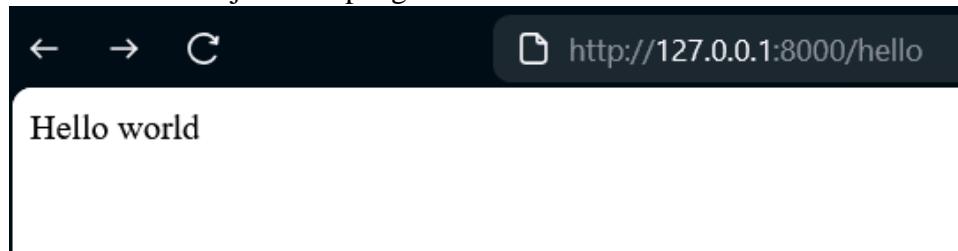
1. Routing

1.1. Basic Routing

- a. Pada bagian ini, kita akan membuat dua buah route dengan ketentuan sebagai berikut.
Kita akan menggunakan project minggu sebelumnya yaitu PWL_2024.
- b. Buka file routes/web.php. Tambahkan sebuah route untuk nomor 1 seperti di bawah ini:

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

- c. Buka browser, tuliskan URL untuk memanggil route tersebut: localhost/PWL_2024/public/hello. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.

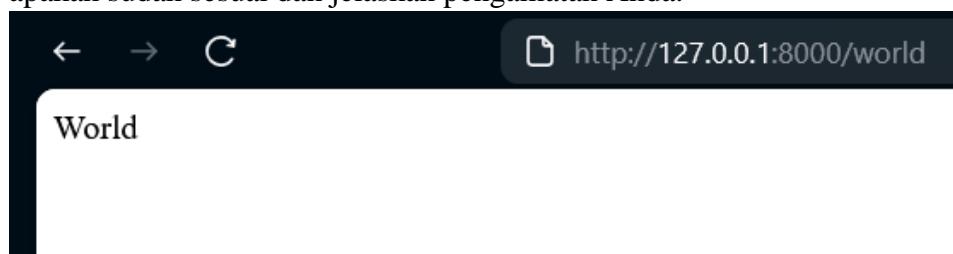


Penjelasan: Menampilkan teks hello world

- d. Untuk membuat route kedua, tambahkan route /world seperti di bawah ini:

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

Bukalah pada browser, tuliskan URL untuk memanggil route tersebut: localhost/PWL_2024/public/world. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.



Penjelasan: Menampilkan teks world

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

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

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

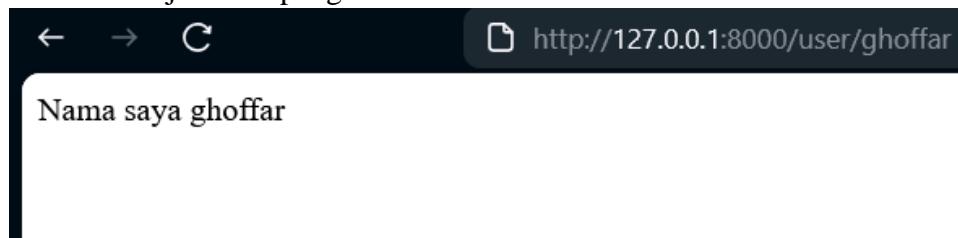
```
Route::get('/about', function () {
    return '2341720035 Ghoffar Abdul';
});
```

1.2. Route Parameters

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

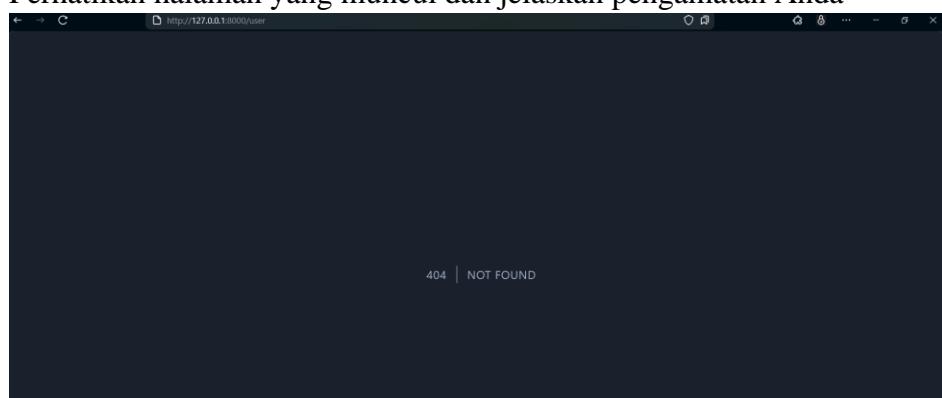
```
Route::get('/user/{name}', function ($name) {
    return 'Nama saya ' . $name;
});
```

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



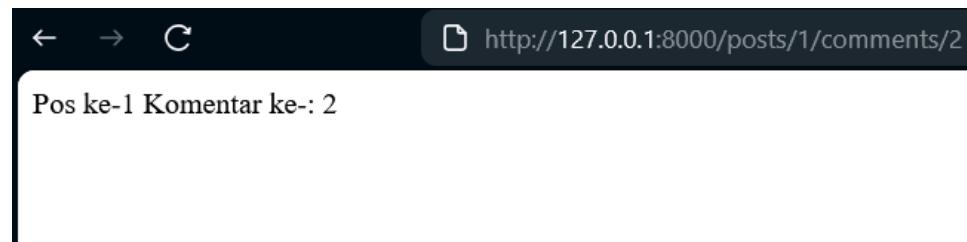
Penjelasan: Menampilkan teks yang menampilkan nama yang sesuai dengan yang diketik pada link

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



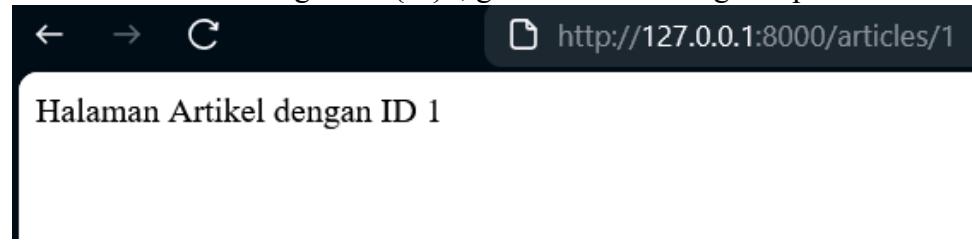
Penjelasan: laman tidak dapat dicari karena tidak ada parameter yang dimasukkan

- d. Suatu route, juga bisa menerima lebih dari 1 parameter seperti kode berikut ini. Route menerima parameter \$postId dan juga \$comment.
- e. 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.



Penjelasan: dapat menampilkan lebih dari 1 parameter

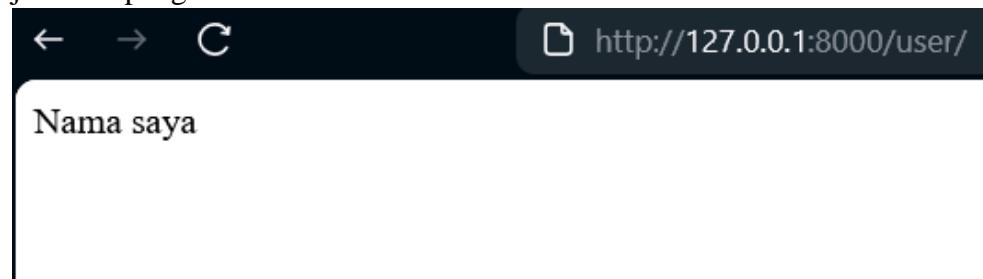
- f. Kemudian buatlah route /articles/{id} yang akan menampilkan output “Halaman Artikel dengan ID {id}”, ganti id sesuai dengan input dari url.



Penjelasan: dapat menampilkan id artikel sesuai dengan parameter yang dimasukkan

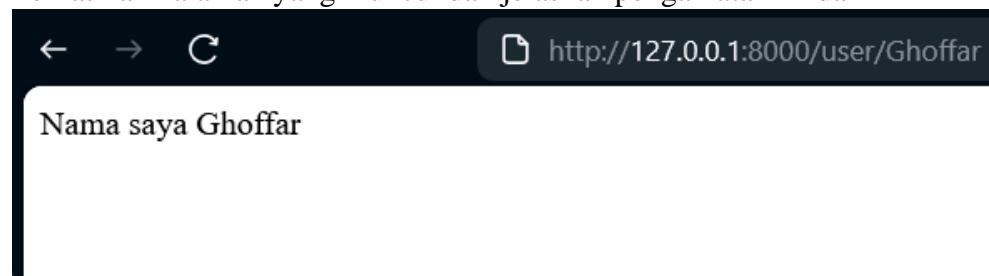
1.3. Optional Parameters

- a. Kita akan memanggil route /user sekaligus mengirimkan parameter berupa nama user \$name dimana parameternya bersifat opsional.
- b. Jalankan kode dengan menuliskan URL:
localhost/PWL_2024/public/user/. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Penjelasan: dapat mengakses halaman meskipun tanpa memasukkan parameter karena terdapat default parameter

- c. Selanjutnya tuliskan URL: **localhost/PWL_2024/public/user/NamaAnda**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda

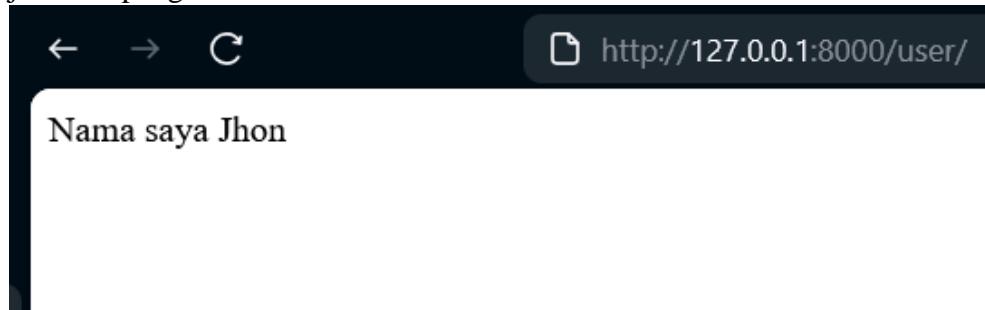


Penjelasan: dapat menampilkan nama

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

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

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



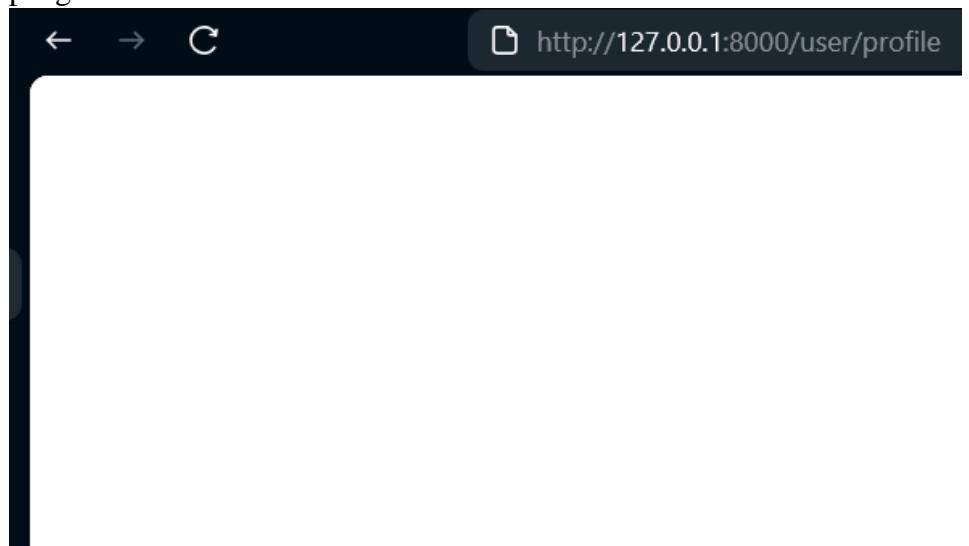
Penjelasan: terdapat nama default yaitu jhon

1.4. Route Name

- a. Buat kode seperti di bawah ini, dimana route /user/profile akan disederhanakan dengan route name menjadi 'profile'.

```
Route::get('/user/profile', function() {  
    //  
})->name('profile');
```

- b. Buka URL sesuai dengan nama route yang telah dibuat dan jelaskan pengamatan Anda.



Penjelasan: Dapat di akses

2. Controllers

2.1. Membuat Controller

- a. 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.

```
PS E:\Kuliah\Pemrograman Web Lanjut\PWL_2025> php artisan make:controller WelcomeController
```

- b. 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
{
    //
}
```

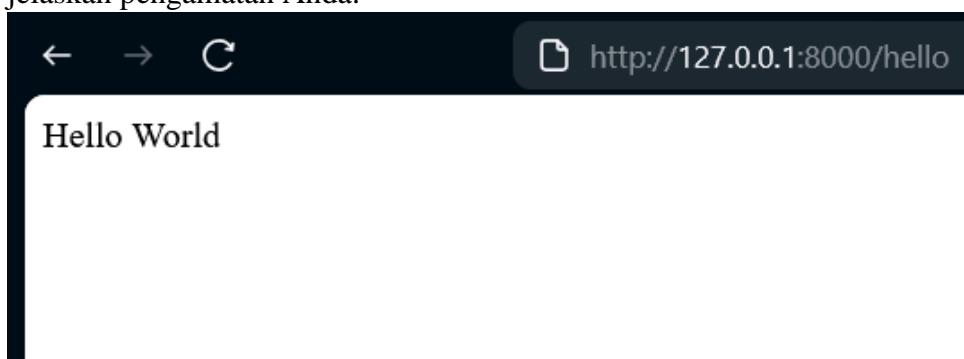
- c. Untuk mendefinisikan action, silahkan tambahkan function dengan access public. Sehingga controller di atas menjadi sebagai berikut:

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

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

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

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



Penjelasan: dapat menampilkan hello world

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

```
public function index() {
    return 'Selamat Datang';
}

public function about() {
    return 'Nama: Ghoffar, Nim: 2341720035';
}

public function articles($id) {
    return 'Halaman Artikel dengan Id'. $id;
}
```

- g. 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

2.2. Resource Controller

- a. Untuk membuatnya dilakukan dengan menjalankan perintah berikut ini di terminal.

```
PS E:\Kuliah\Pemrograman_Web_Lanjut\PWL_2025> php artisan make:controller PhotoController --resource
```

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

- b. 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;

Route::resource('photos', PhotoController::class);
```

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

```

PS E:\Kuliah\Pemrograman_Web_Lanjut\WkL_2025> php artisan route:list
GET|HEAD  photos .....
POST      photos .....
GET|HEAD  photos/create
GET|HEAD  photos/{photo}
PUT|PATCH photos/{photo}
DELETE   photos/{photo}
GET|HEAD  photos/{photo}/edit
GET|HEAD  storage/{path}
GET|HEAD  up .....

photos.index > PhotoController@index
photos.store > PhotoController@store
photos.create > PhotoController@create
photos.show > PhotoController@show
photos.update > PhotoController@update
photos.destroy > PhotoController@destroy
photos.edit > PhotoController@edit
..... storage.local

Showing [9] routes

```

- d. 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.

```

Route::resource('photos', PhotoController::class)->only([
    'index', 'show'
]);

Route::resource('photos', PhotoController::class)->except([
    'create', 'store', 'update', 'destroy'
]);

```

3. View

3.1. Membuat View

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

```

</head>
<body>
    <h1>Hello, {{ $name }}</h1>
</body>

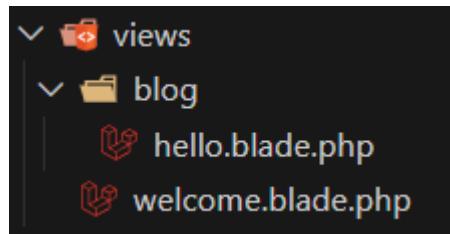
```

- b. View tersebut dapat dijalankan melalui Routing, dimana *route* akan memanggil View sesuai dengan nama *file* tanpa ‘blade.php’. (Catatan: Gantilah Andi dengan nama Anda)
- c. Jalankan code dengan membuka url localhost/PWL_2024/public/greeting. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

Hello, Ghoffar

3.2. View dalam direktori

- a. Buatlah direktori blog di dalam direktori views.
- b. Pindahkan file hello.blade.php ke dalam direktori blog.



- c. Selanjutnya lakukan perubahan pada route.

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

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



3.3. Menampilkan View dari Controller

- a. Buka WelcomeController.php dan tambahkan fungsi baru yaitu greeting.

```
public function greeting(){
    return view('blog.hello', ['name' => 'Ghoffar']);
}
```

- b. Ubah route /greeting dan arahkan ke WelcomeController pada fungsi greeting.

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

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



3.4. Meneruskan data ke view

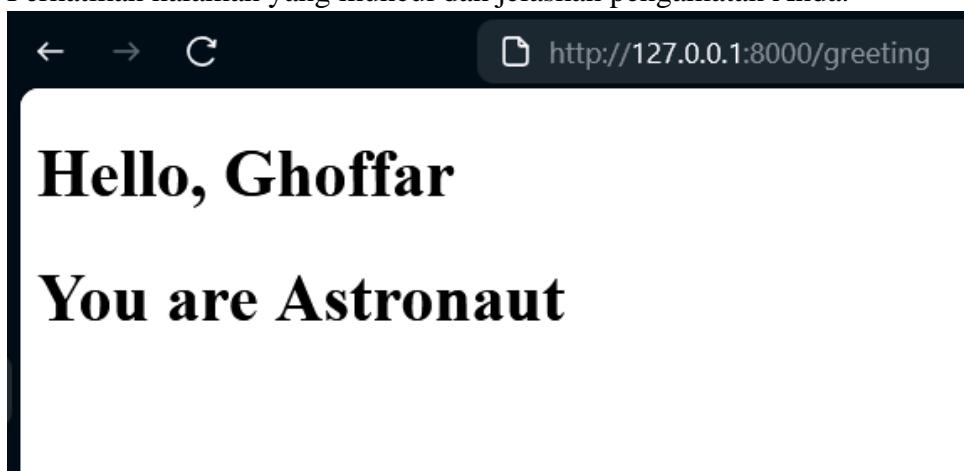
- a. Buka WelcomeController.php dan tambahkan ubah fungsi greeting.

```
public function greeting(){
    return view('blog.hello')
        ->with('name','Ghoffar')
        ->with('occupation','Astronaut');
}
```

- b. Ubah hello.blade.php agar dapat menampilkan dua parameter.

```
<body>
    <h1>Hello, {{ $name }}</h1>
    <h1>You are {{ $occupation }}</h1>
</body>
```

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

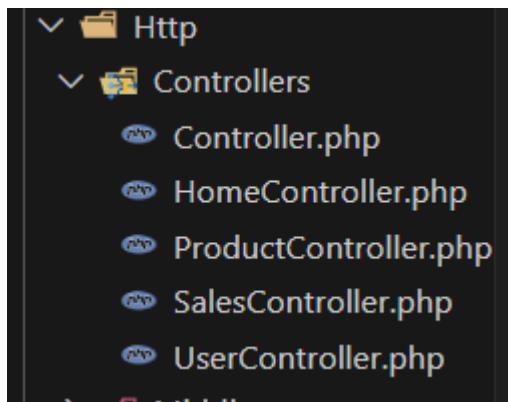


Soal Praktikum

1. Routes

```
// home
Route::get('/', [HomeController::class, 'index']);
Route::get('/category', [HomeController::class, 'products']);
// category product
Route::prefix('category')->group(function () {
    Route::get('food-beverage', [ProductController::class, 'foodbeverage']);
    Route::get('beauty-health', [ProductController::class, 'beautyhealth']);
    Route::get('home-care', [ProductController::class, 'homecare']);
    Route::get('baby-kid', [ProductController::class, 'babykid']);
});
// User
Route::get('/user/{id}/name/{name}', [UserController::class, 'index']);
// Sales
Route::get('/sales', [SalesController::class, 'index']);
```

2. Controller



3. View

