**Laporan Jobsheet Pemrograman Web Lanjut**

**JOBSHEET 02**

**ROUTING, CONTROLLER, DAN VIEW**

**Dosen Pembimbing:**

**Dimas Wahyu Wibowo, S.T., M.T.**



**Disusun oleh:**

**Achmad Maulana Hamzah / 2A**

**2341720172**

**JURUSAN TEKNOLOGI INFORMASI**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI MALANG**

**2024**

1. **Routing**

* **Basic Routing**

1. Pada bagian ini, kita akan membuat dua buah route dengan ketentuan sebagai berikut.

Kita akan menggunakan project minggu sebelumnya yaitu PWL\_2024.

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

use Illuminate\Support\Facades\Route;

Route::get('/hello', function () {

return 'Hello World';

});

1. Buka browser, tuliskan URL untuk memanggil route tersebut:

localhost/PWL\_2024/public/hello.

Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.



Muncul pesan Hello World didalam website

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

use Illuminate\Support\Facades\Route;

Route::get('/world', function () {

return 'World';

});

1. Bukalah pada browser, tuliskan URL untuk memanggil route tersebut:

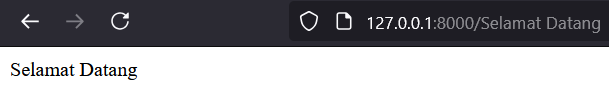
localhost/PWL\_2024/public/world.

Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.

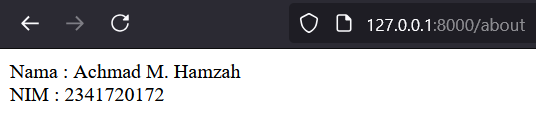
****

Muncul pesan World didalam website

1. Selanjutnya, cobalah membuat route ’/’ yang menampilkan pesan ‘Selamat Datang’.



1. Kemudian buatlah route ‘/about’ yang akan menampilkan NIM dan nama Anda.



* **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;

});

1. Jalankan kode dengan menuliskan URL untuk memanggil route tersebut:

localhost/PWL\_2024/public/user/NamaAnda. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

****

Muncul pesan dengan nama saya didalam website

1. Selanjutnya, coba tuliskan URL: localhost/PWL\_2024/public/user/. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda



Tidak muncul apa apa didalam website

1. 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;

});

1. 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.



Muncul pesan diatas didalam website

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

* **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;

});

1. Jalankan kode dengan menuliskan URL: localhost/PWL\_2024/public/user/.

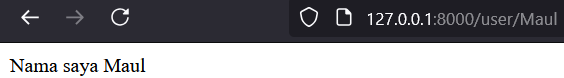
Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Hanya muncul pesan seperti berikut

1. Selanjutnya tuliskan URL: localhost/PWL\_2024/public/user/NamaAnda. Perhatikan

halaman yang muncul dan jelaskan pengamatan Anda



Muncul pesan diikuti nama saya

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

Route::get('/user/{name?}', function ($name='John') {

return 'Nama saya '.$name;

});

1. Jalankan kode dengan menuliskan URL: localhost/PWL\_2024/public/user/.

Perhatikan halaman yang muncul dan jelaskan pengamatan Anda



Muncul pesan dengan nama john karena parameter nama diberi kata John

1. **Controller**

* **Membuat Controller**

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.

php artisan make:controller WelcomeController

1. 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

{

//

}

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

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class WelcomeController extends Controller

{

public function hello() {

return 'Hello World';

}

}

1. 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’]);

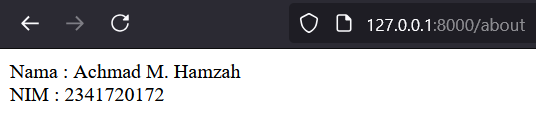
1. Buka browser, tuliskan URL untuk memanggil route tersebut:

localhost/PWL\_2024/public/hello. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



1. Modifikasi hasil pada praktikum poin 2 (Routing) dengan konsep controller. Pindahkan

logika eksekusi ke dalam controller dengan nama PageController.



1. 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.

* **Resource Controller**

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

php artisan make:controller PhotoController --resource

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

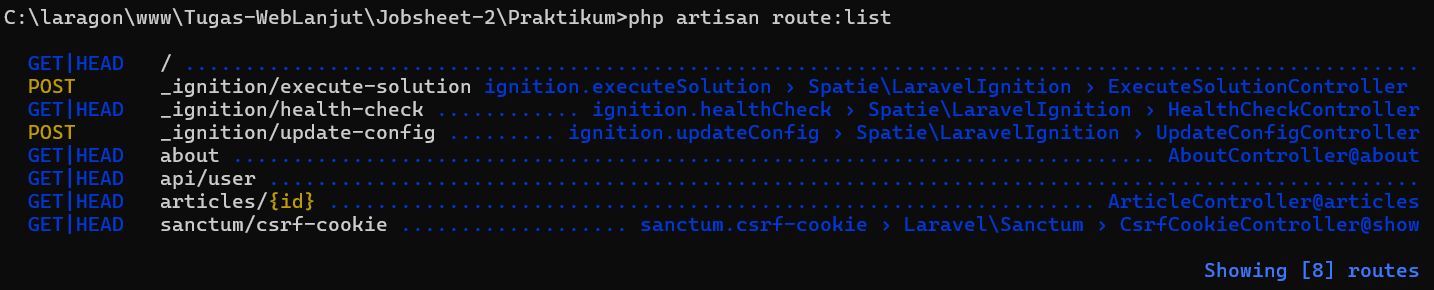
1. Setelah controller berhasil degenerate, selanjutnya harus dibuatkan route agar dapat

terhubung dengan frontend. Tambahkan kode program berikut pada file web.php.

use App\Http\Controllers\PhotoController;

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

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



1. 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'

]);

1. **View**

* **Membuat View**

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

<!-- View pada resources/views/hello.blade.php -->

<html>

<body>

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

</body>

</html>

1. View tersebut dapat dijalankan melalui Routing, dimana route akan memanggil View

sesuai dengan nama file tanpa ‘blade.php’. (Catatan: Gantilah Andi dengan nama Anda)

Route::get('/greeting', function () {

return view('hello', ['name' => 'Andi']);

});

1. Jalankan code dengan membuka url localhost/PWL\_2024/public/greeting. Perhatikan

halaman yang muncul dan jelaskan pengamatan Anda.



Muncul pesan berikut didalam website

* **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.

Route::get('/greeting', function () {

return view('blog.hello', ['name' => 'Andi']);

});

1. Jalankan code dengan membuka url localhost/PWL\_2024/public/greeting. Perhatikan

halaman yang muncul dan jelaskan pengamatan Anda.



Hasilnya sama seperti pada praktikum sebelumnya

* **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' => 'Andi']);

}

}

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

Route::get('/greeting', [WelcomeController::class,

'greeting']);

1. Jalankan code dengan membuka url localhost/PWL\_2024/public/greeting. Perhatikan

halaman yang muncul dan jelaskan pengamatan Anda.



Hasilnya sama seperti praktikum sebelumnya. Hanya saja menampilkan view melalui controller.

* **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','Andi')

->with('occupation','Astronaut');

}

}

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

<html>

<body>

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

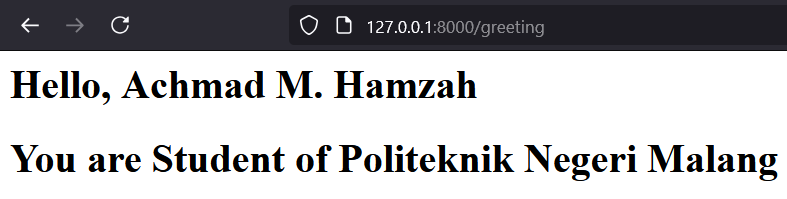
<h1>You are {{ $occupation }}</h1>

</body>

</html>

1. Jalankan code dengan membuka url localhost/PWL\_2024/public/greeting. Perhatikan

halaman yang muncul dan jelaskan pengamatan Anda.



Muncul pesan berikut dengan menggabungkan 2 parameter.