

Laporan Pemrograman Web Lanjut

Jobsheet 02 : Routing, Controller, dan View

Dosen Pengampu : Dimas Wahyu Wibowo, ST.,MT.



Disusun Oleh:

Queenadhynar Azarine Dwipa A.

2341760109

SIB 2B

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023/2024

Langkah-langkah Praktikum :

A. Routing

- **Basic Routing**

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

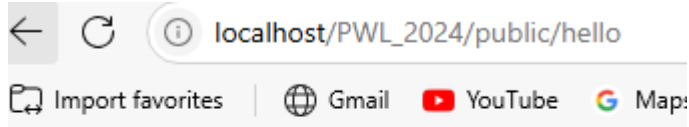
No	Http Verb	Url	Fungsi
1	get	/hello	Tampilkan String Hello ke browser.
2	get	/world	Tampilkan String World ke browser

Dengan menggunakan project minggu sebelumnya yaitu

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

```
// Route Hello
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.

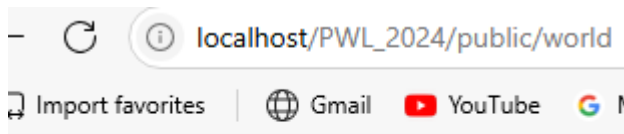


Hello World

- Route::get('/hello') sebuah route yang menangani HTTP GET request ke path /hello
 - function () { return 'Hello World'; } yang akan dijalankan saat route /hello diakses
 - Hasilnya saat menjalankan localhost/PWL_2024/public/hello maka teks “Hello World ” akan ditampilkan seperti gambar diatas
- d. Untuk membuat route kedua, tambahkan route /world seperti di bawah ini:

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

- e. 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.

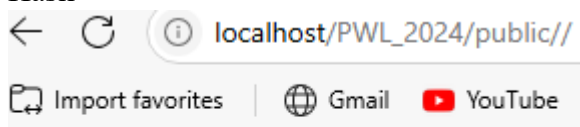


World

- Route::get('/world') sebuah route yang menangani HTTP GET request ke path /world
 - function () { return 'World'; } yang akan dijalankan saat route /world diakses
 - Hasilnya saat menjalankan localhost/PWL_2024/public/world maka teks “World ” akan ditampilkan seperti gambar diatas
- f. Selanjutnya, cobalah membuat route '/' yang menampilkan pesan ‘Selamat Datang’.
- Route

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

➤ Hasil

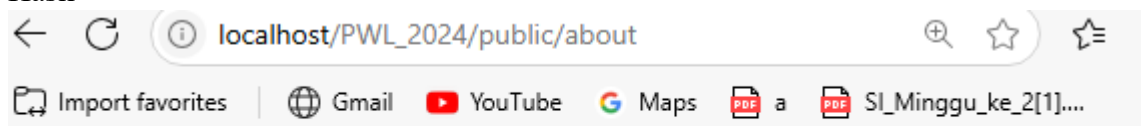


Selamat Datang

- Route::get('/') sebuah route yang menangani HTTP GET request ke path /
 - function () { return 'Selamat Datang'; } yang akan dijalankan saat route /diakses
 - Hasilnya saat menjalankan localhost/PWL_2024/public/ maka teks “Selamat Datang” akan ditampilkan seperti gambar diatas
- g. Kemudian buatlah route '/about' yang akan menampilkan NIM dan nama Anda.
- Route

```
// Route about  
Route::get('/about', function () {  
    return 'NIM : 2341760109, Nama : Queenadhynar Azarine Dwipa Andiyani';  
});
```

➤ Hasil



NIM : 2341760109, Nama : Queenadhynar Azarine Dwipa Andiyani

- Route::get('/about') sebuah route yang menangani HTTP GET request ke path /about
- function () { return 'about'; } yang akan dijalankan saat route /aboutdiakses

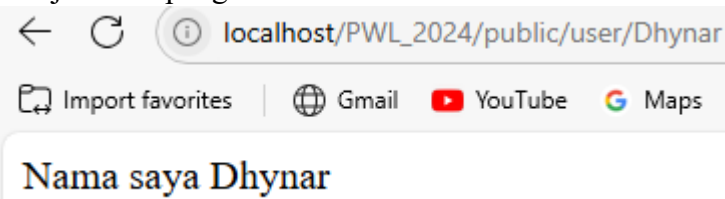
- Hasilnya saat menjalankan localhost/PWL_2024/public/about maka teks “NIM : 2341760109, Nama : Queenadhynar Azarine Dwipa Andiyani” akan ditampilkan seperti gambar diatas

• Route Parameters

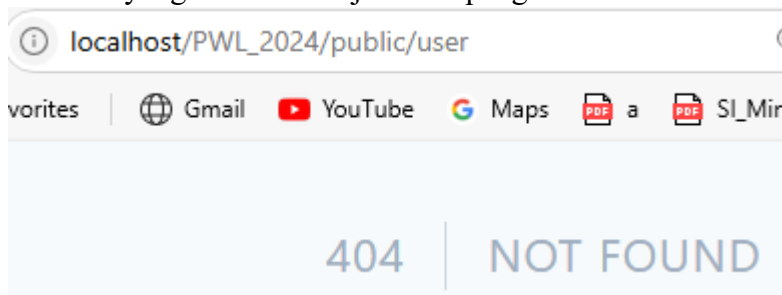
- Kita akan memanggil route /user/{name} sekaligus mengirimkan parameter berupa nama user \$name seperti kode di bawah ini.

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

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



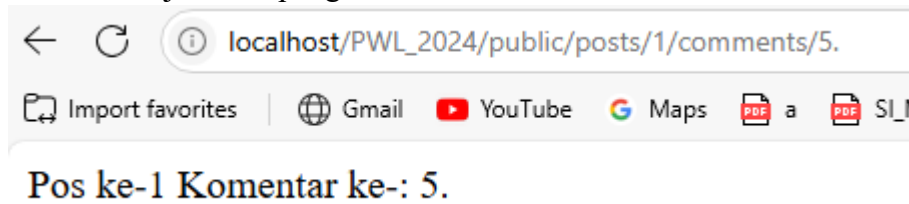
- Route diatas memiliki parameter {name} yang dapat menerima nilai, jika menjalankan localhost/PWL_2024/public/user/Dhynar maka hasil yang muncul “**Nama saya Dhynar**”. Seperti hasil diatas
- Selanjutnya, coba tuliskan URL: localhost/PWL_2024/public/user/. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda



- Hasilnya error/404 not found dikarenakan route yang diatas membutuhkan parameter {name}, tetapi URL /PWL_2024/public/user/ tidak menyertakan parameter tersebut
- Suatu route, juga bisa menerima lebih dari 1 parameter seperti kode berikut ini. Route menerima parameter \$postId dan juga \$comment

```
// Route dengan 1 parameter lebih
Route::get('/posts/{post}/comments/{comment}', function ($postId, $commentId) {
    return 'Pos ke- '.$postId." Komentar ke-: " . $commentId;
});
```

- 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.

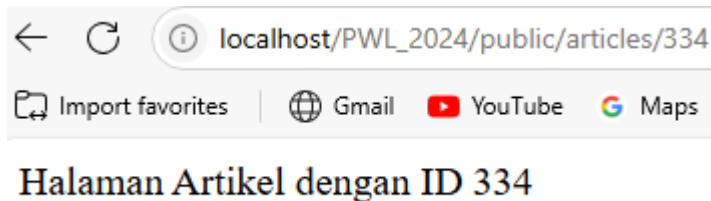


- Route diatas memiliki 2 Paramater yang dapat menerima nilai sesuai dengan yang isi pada parameter. Seperti hasil diatas parameter post diisi dengan angka 1 dan paremeter comments diisi dengan angka 5 maka saat dijalankan URL: localhost/PWL_2024/public/posts/1/comments/5. Maka akan menghasilkan teks seperti diatas
- f. Kemudian buatlah route /articles/{id} yang akan menampilkan output “Halaman Artikel dengan ID {id}”, ganti id sesuai dengan input dari url.

- Route

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

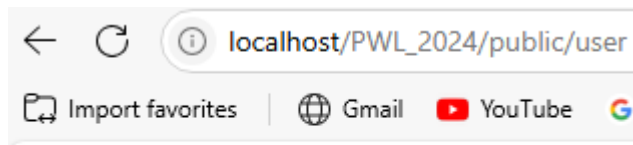
- Hasil



- Route dengan /articles/{id} dimana parameter id akan menerima input id dari url. Seperti contoh diatas saat memasukkan localhost/PWL_2024/public/articles/334 maka akan memunculkan hasil “Halaman Artikel dengan ID 334”
- **Optional Parameters**
 - a. Kita akan memanggil route /user sekaligus mengirimkan parameter berupa nama user \$name dimana parameternya bersifat opsional.

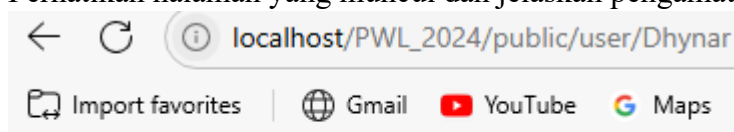
```
// Route User
Route::get('/user/{name?}', function ($name=null) {
    return 'Nama saya ' . $name;
});
```

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



Nama saya

- Parameter {name?} adalah parameter opsional, yang berarti bisa diisi atau tidak
 - Jika parameter {name} diberikan dalam URL maka akan berisi nilai yang telah ditentukan, namun jika tidak diisi maka akan bernilai null seperti hasil di atas. Hanya memunculkan hasil “**Nama Saya**”
- c. Selanjutnya tuliskan URL: localhost/PWL_2024/public/user>NamaAnda. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda

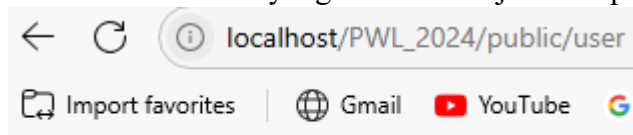


Nama saya Dhynar

- Di atas merupakan contoh hasil jika URL diberi isi “**Dhynar**” maka akan memunculkan teks “Nama saya Dhynar” sesuai URL yang diberi isi
- d. Ubah kode pada route /user menjadi seperti di bawah ini.

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

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



Nama saya John

- Untuk route di atas jika parameter tidak diisi hasil yang keluar akan memiliki nilai default “**John**”. Sehingga saat menjalankan URL: localhost/PWL_2024/public/user/ maka akan muncul hasil “**Nama saya John**”
- **Route Name**
Route name biasanya digunakan untuk mempermudah kita dalam pemanggilan route saat membangun aplikasi. Kita cukup memanggil name dari route tersebut.

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

Route::get(
    '/user/profile',
    [UserProfileController::class, 'show']
)->name('profile');

// Generating URLs...
$url = route('profile');

// Generating Redirects...
return redirect()->route('profile');
```

- **Route Group dan Route Prefixes**

Beberapa route yang memiliki atribut yang sama seperti middleware yang sama dapat dikelompokkan menjadi satu kelompok untuk mempermudah penulisan route selain digunakan untuk middleware masih ada lagi penggunaan route group untuk route yang berada dibawah satu subdomain. Contoh penggunaan route group adalah sebagai berikut:

```
Route::middleware(['first', 'second'])->group(function () {
    Route::get('/', function () {
        // Uses first & second middleware...
    });

    Route::get('/user/profile', function () {
        // Uses first & second middleware...
    });
});

Route::domain('{account}.example.com')->group(function () {
    Route::get('user/{id}', function ($account, $id) {
        //
    });
});

Route::middleware('auth')->group(function () {
    Route::get('/user', [UserController::class, 'index']);
    Route::get('/post', [PostController::class, 'index']);
    Route::get('/event', [EventController::class, 'index']);
});
```

Route Prefixes

Pengelompokan route juga dapat dilakukan untuk route yang memiliki prefix (awalan) yang sama. Untuk pembuatan route dengan prefix dapat dilihat kode seperti di bawah ini

```
Route::prefix('admin')->group(function () {
    Route::get('/user', [UserController::class, 'index']);
    Route::get('/post', [PostController::class, 'index']);
    Route::get('/event', [EventController::class, 'index']);
});
```

- **Redirect Routes**

Untuk melakukan redirect pada laravel dapat dilakukan dengan menggunakan Route::redirect cara penggunaannya dapat dilihat pada kode program dibawah ini

```
Route::redirect('/here', '/there');
```

Redirect ini akan sering digunakan pada kasus kasus CRUD atau kasus lain yang membutuhkan redirect.

- **View Routes**

Laravel juga menyediakan sebuah route khusus yang memudahkan dalam membuat sebuah routes tanpa menggunakan controller atau callback function. Routes ini langsung menerima input berupa url dan mengembalikan view / tampilan. Berikut ini cara membuat view routes.

```
Route::view('/welcome', 'welcome');
Route::view('/welcome', 'welcome', ['name' => 'Taylor']);
```

Pada view routes diatas /welcome akan menampilkan view welcome dan pada route kedua /welcome akan menampilkan view welcome dengan tambahan data berupa variabel name.

B. Controller

- **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 D:\laragon\www\PWL_2024> php artisan make:controller WelcomeController
```

```
INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\WelcomeController.php] created successfully.
```

- 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:

```
<?php

namespace App\Http\Controllers;

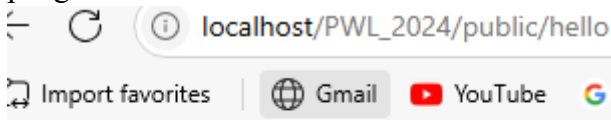
use Illuminate\Http\Request;

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 Hello
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.



Hello World

- Pada class WelcomeController extends Controller
Mendeklarasikan sebuah controller yaitu WelcomeController, extends berarti controller ini mewarisi fungsi dari Controller utama Laravel. hello() mengembalikan string 'Hello World' ketika dipanggil melalui route.
Pada route [WelcomeController::class, 'hello'] menunjukkan bahwa route /hello akan memanggil method hello() di WelcomeController
- f. 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		

- Membuat controller pada Laravel

```
PS D:\laragon\www\PWL_2024> php artisan make:controller PageController

INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\PageController.php] created successfully.
```

- Mendefinisikan action, sehingga menjadi sebagai berikut :

```
class PageController extends Controller
{
    public function index() {
        return 'Selamat Datang';
    }

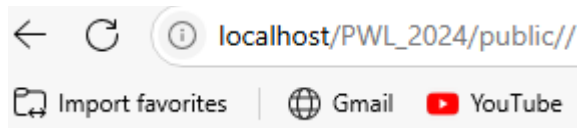
    public function about() {
        return "NIM : 2341760109, Nama : Queenadhynar Azarine Dwipa Andiyani";
    }

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

- Setelah controller didefinisikan action, lalu menambahkan controller pada **resource /**
- Route

```
// Route '/'
Route::get('/', [PageController::class, 'index']);
```

- Hasil



localhost/PWL_2024/public/

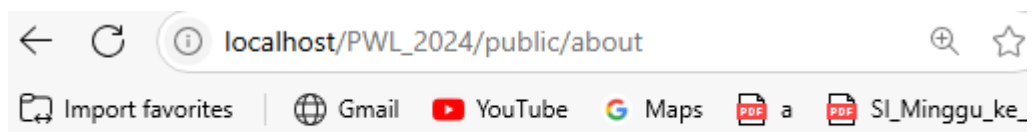
Import favorites | Gmail | YouTube

Selamat Datang

resource /about

- Route

```
// Route about
Route::get('/about', PageController::class, 'about');
```



localhost/PWL_2024/public/about

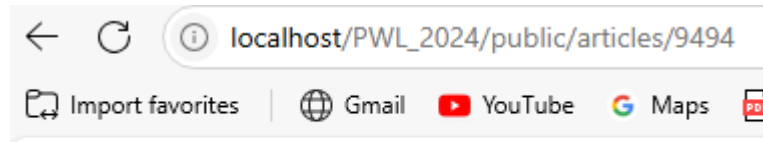
Import favorites | Gmail | YouTube | Maps | a | SI_Minggu_ke_

NIM : 2341760109, Nama : Queenadhynar Azarine Dwipa Andiyani

Resource /articles/{id}

- Route

```
// Route Halaman Artikel
Route::get('/articles/{id}', [PageController::class, 'articles']);
```



Halaman Artikel dengan ID 9494

- 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

- Membuat controller

```
PS D:\laragon\www\PWL_2024> php artisan make:controller HomeController

INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\HomeController.php] created successfully.

PS D:\laragon\www\PWL_2024> php artisan make:controller AboutController

INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\AboutController.php] created successfully.

PS D:\laragon\www\PWL_2024> php artisan make:controller ArticleController

INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\ArticleController.php] created successfully.
```

- Mendefinisikan action, sehingga menjadi sebagai berikut :
HomeController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    public function index() {
        return 'Selamat Datang';
    }
}
```

AboutController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AboutController extends Controller
{
    public function about() {
        return "NIM : 2341760109, Nama : Queenadhynar Azarine Dwipa Andiyani";
    }
}
```

ArticleController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ArticleController extends Controller
{
    public function articles($id) {
        return 'Halaman Artikel dengan ID '.$id;
    }
}
```

- Setelah controller didefinisikan action, lalu menambahkan Single Action controller pada **resource /, about, dan article**

```
// Route dengan Single Action Controller
// Route '/'
Route::get('/', [HomeController::class, 'index']);
// Route about
Route::get('/about', [AboutController::class, 'about']);
// Route Halaman Artikel
Route::get('/articles/{id}', [ArticleController::class, 'articles']);
```

- Hasil yang muncul

localhost/PWL_2024/public//

Import favorites | Gmail | YouTube

Selamat Datang

localhost/PWL_2024/public/about

Import favorites | Gmail | YouTube | Maps | a | SI_Minggu_ke_2[1]....

NIM : 2341760109, Nama : Queenadhynar Azarine Dwipa Andiyani

localhost/PWL_2024/public/articles/0705

Import favorites | Gmail | YouTube | Maps

Halaman Artikel dengan ID 0705

- **Resource Controller**

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

```
PS D:\laragon\www\PWL_2024> php artisan make:controller PhotoController --resource
```

```
INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\PhotoController.php] created successfully.
```

Perintah ini akan generate sebuah controller dengan nama PhotoController yang berisi method 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;

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

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

```
PS D:\laragon\www\PhL_2024> php artisan route:list

GET|HEAD / ..... HomeController@index
POST _ignition/execute-solution ignition.executeSolution > Spatie\LaravelIgnition > ExecuteSoluti...
GET|HEAD _ignition/health-check ignition.healthCheck > Spatie\LaravelIgnition > HealthCheckController
POST _ignition/update-config ignition.updateConfig > Spatie\LaravelIgnition > UpdateConfigControl...
GET|HEAD about ..... AboutController@about
GET|HEAD api/user .....
GET|HEAD articles/{id} ..... ArticleController@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
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 photos/{photo}/edit ..... photos.edit > PhotoController@edit
GET|HEAD posts/{post}/comments/{comment} .....
GET|HEAD sanctum/csrf-cookie ..... sanctum.csrf-cookie > Laravel\Sanctum > CsrfCookieController@show
GET|HEAD user/{name?} .....
GET|HEAD user/{name} .....
GET|HEAD world .....

Showing [27] 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']);
```

C. View

- **Membuat View**

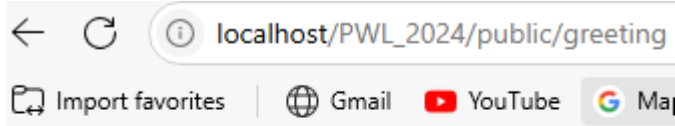
- a. Pada direktori app/resources/views, buatlah file hello.blade.php.

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

- 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)

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

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

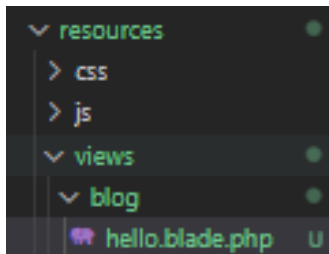


Hello, Dhynar

- Sintaks {{ \$name }} fitur Blade Template untuk menampilkan data yang dikirim dari controller atau route, fungsi callback (function ()...)mengembalikan tampilan view bernama hello dan mengirimkan data name dengan nilai 'Dhynar' Karena variabel name diisi dengan 'Dhynar' maka teks akan muncul seperti hasil diatas

- **View dalam Direktori**

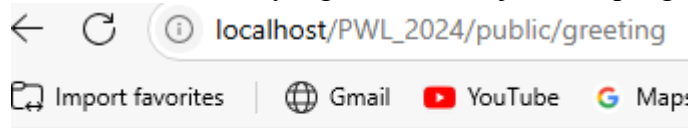
- a. Buatlah direktori blog di dalam direktori views.
- b. Pindahkan file hello.blade.php ke dalam direktori blog.



- c. Selanjutnya lakukan perubahan pada route.

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

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



Hello, Andi

- Sama dengan code sebelumnya namun yang membedakan adalah view memanggil blog.hello karena file hello.blade.php dipindah ke dalam folder blog. Juga variabel yang diisi dengan 'Andi' sehingga menghasilkan teks seperti diatas

- **Menampilkan View dari Controller**

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

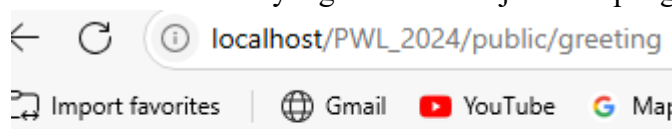
```
class WelcomeController extends Controller
{
    public function hello() {
        return 'Hello World';
    }

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

- 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



Hello, Andi

- Method greeting() akan dipanggil lalu merender view blog.hello, Laravel akan meneruskan data name = 'Andi' ke view. Saat url dijalankan maka akan keluar hasil seperti diatas

- **Meneruskan data ke view**

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

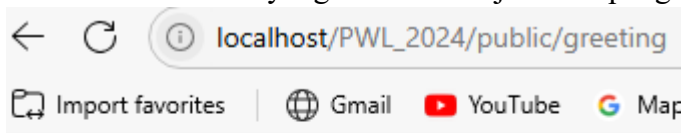
```
class WelcomeController extends Controller
{
    public function hello() {
        return ('Hello World');
    }

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

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

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

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



Hello, Andi

You are Astronaut

- Method greeting() akan dipanggil lalu merender view blog.hello, ->with() untuk mengirimkan dua variabel yaitu name = 'Andi', occupation = 'Astronaut'
- Blade template menampilkan data yang dikirim controller
{{ \$name }} menjadi "Andi" {{ \$occupation }} menjadi "Astronaut"


SOAL PRAKTIKUM

1. Jalankan Langkah-langkah Praktikum pada jobsheet di atas. Lakukan sinkronisasi perubahan pada project PWL_2024 ke Github.
2. Buatlah project baru dengan nama POS. Project ini merupakan sebuah aplikasi Point of Sales yang digunakan untuk membantu penjualan.

- Membuat Project POS (Point of Sales)

```
PS D:\laragon\www> composer create-project laravel/laravel POS
Creating a "laravel/laravel" project at "../POS"
Cannot use laravel/laravel's latest version v11.6.1 as it requires php
^8.0.
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in D:\laragon\www\POS
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.12.1)
```


3. Buatlah beberapa route, controller, dan view sesuai dengan ketentuan sebagai berikut.

1	<p>Halaman Home</p> <p>Menampilkan halaman awal website</p> <ul style="list-style-type: none">➤ Membuat Home Controller <pre>PS D:\laragon\www\POS> php artisan make:controller HomeController</pre> <p>INFO Controller [D:\laragon\www\POS\app\Http\Controllers\HomeController.php] created successfully.</p> <pre>app > Http > Controllers > HomeController.php > HomeController > home</pre> <pre>1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class HomeController extends Controller 8 { 9 public function home() { 10 return view('POS.home'); 11 } 12 }</pre> <ul style="list-style-type: none">➤ Membuat view <pre>resources > views > POS > home.blade.php > ...</pre> <pre>1 <html> 2 <body> 3 <h1>Welcome to Point of Sales</h1> 4 </body> 5 </html> 6</pre> <ul style="list-style-type: none">➤ Membuat Routes <pre>use App\Http\Controllers\HomeController;</pre> <pre>// Home Page Route::get('/home', [HomeController::class, 'home']);</pre> <ul style="list-style-type: none">➤ Hasil untuk Halaman Home saat memanggil localhost/POS/public/home  <p>localhost/POS/public/home</p> <p>Import favorites Gmail YouTube Maps</p> <p>Welcome to Point of Sales</p>
2	<p>Halaman Products</p> <p>Menampilkan daftar product (route prefix)</p> <ul style="list-style-type: none">/category/food-beverage/category/beauty-health/category/home-care/category/baby-kid <ul style="list-style-type: none">➤ Membuat Controller <pre>PS D:\laragon\www\POS> php artisan make:controller ProductController</pre> <p>INFO Controller [D:\laragon\www\POS\app\Http\Controllers\ProductController.php] created successfully.</p>

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ProductController extends Controller
{
    public function foodbeverage() {
        return view('POS.Product.foodbeverage');
    }
    public function beauty() {
        return view('POS.Product.beauty');
    }
    public function homecare() {
        return view('POS.Product.homecare');
    }
    public function baby() {
        return view('POS.Product.baby');
    }
}

```

- Membuat view
food-beverage

```

resources > views > POS > Product > foodbeverage.blade.php > ...
1  <html>
2      <body>
3          <h1>PRODUCT LIST</h1>
4          <h2>Food and Beverage</h2>
5      </body>
6  </html>

```

beauty-health

```

resources > views > POS > Product > beauty.blade.php >
1  <html>
2      <body>
3          <h1>PRODUCT LIST</h1>
4          <h2>Beauty and Health</h2>
5      </body>
6  </html>

```

home-care

```

resources > views > POS > Product > homecare.blade.php >
1  <html>
2      <body>
3          <h1>PRODUCT LIST</h1>
4          <h2>Home Care</h2>
5      </body>
6  </html>

```

baby-kid

```

resources > views > POS > Product > baby.blade.php >
1  <html>
2      <body>
3          <h1>PRODUCT LIST</h1>
4          <h2>Baby and Kid Product</h2>
5      </body>
6  </html>

```

- Membuat Route Prefix

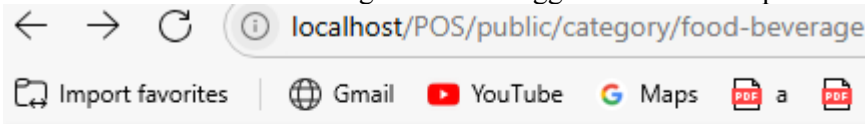
```

use App\Http\Controllers\ProductControllerController;

//Product Page
Route::prefix('category')->group(function(){
    Route::get('/food-beverage', [ProductController::class, 'foodbeverage']);
    Route::get('/beauty-health', [ProductController::class, 'beauty']);
    Route::get('/home-care', [ProductController::class, 'homecare']);
    Route::get('/baby-kid', [ProductController::class, 'baby']);
});

```

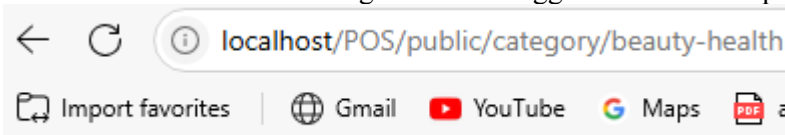
- Hasil
 untuk Halaman Food - Beverage saat memanggil localhost/POS/public/ category/food-beverage



PRODUCT LIST

Food and Beverage

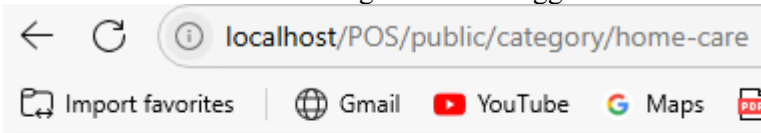
untuk Halaman Food - Beverage saat memanggil localhost/POS/public /category/beauty-health



PRODUCT LIST

Beauty and Health

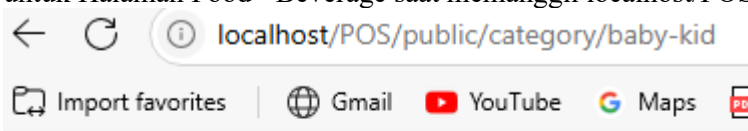
untuk Halaman Food - Beverage saat memanggil localhost/POS/public /category/home-care



PRODUCT LIST

Home Care

untuk Halaman Food - Beverage saat memanggil localhost/POS/public /category/baby-kid



PRODUCT LIST

Baby and Kid Product

- 3 Halaman User
 Menampilkan profil pengguna (route param)

/user/{id}/name/{name}

➤ Membuat Controller

```
PS D:\laragon\www\POS> php artisan make:controller UserController
```

INFO Controller [D:\laragon\www\POS\app\Http\Controllers\UserController.php] created successfully.

```
app > Http > Controllers > UserController.php > ...
```

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class UserController extends Controller
8 {
9     public function user($id = null, $name = null) {
10         return view('POS.user', ['id' => $id, 'name' => $name]);
11     }
12 }
13
14
```

➤ Membuat View

```
resources > views > POS > user.blade.php > html > body > h2
```

```
1 <html>
2 <body>
3     <h1>USER PAGE</h1>
4     <h2>User ID : {{ $id }}</h2>
5     <h2>Nama : {{ $name }}</h2>
6 </body>
7 </html>
```

➤ Membuat Route Paramater

```
// User Page
Route::get('/user/{id}/{name}', [UserController::class, 'user']);
```

➤ Hasil untuk Halaman user saat memanggil localhost/POS/public/user/{id}/name/{name}
Tampilan saat id dan name null

localhost/POS/public/user

USER PAGE

User ID :

Nama :

Tampilan jika user memasukkan id dan nama

localhost/POS/public/user/994/Dhynar

USER PAGE

User ID : 994

Nama : Dhynar

4 Halaman Penjualan
Menampilkan halaman transaksi POS

➤ Membuat Controller

```
PS D:\laragon\www\POS> php artisan make:controller SaleController

INFO Controller [D:\laragon\www\POS\app\Http\Controllers\SaleController.php] created successfully.

app > Http > Controllers > SaleController.php > SaleController > sales
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class SaleController extends Controller
8  {
9      public function sales() {
10         return view('POS.sale');
11     }
12 }
```

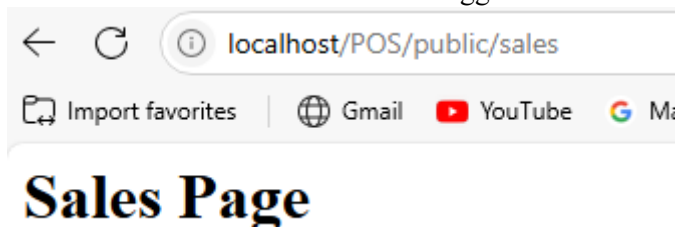
➤ Membuat View

```
resources > views > POS > sale.blade.php > html > body > h1
1  <html>
2      <body>
3          <h1>Sales Page</h1>
4      </body>
5  </html>
```

➤ Membuat Route

```
// Sales Page
Route::get('/sales', [SaleController::class, 'sales']);
```

➤ Hasil untuk Halaman user saat memanggil localhost/ POS/public/sales



4. Route tersebut menjalankan fungsi pada Controller yang berbeda di setiap halaman.
5. Fungsi pada Controller akan memanggil view sesuai halaman yang akan ditampilkan.
6. Simpan setiap perubahan yang dilakukan pada project POS pada Git, sinkronisasi perubahan ke Github.

Link github <https://github.com/dhynarazarine/POS>