

LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT
JOB SHEET 2 : ROUTING, CONTROLLER, DAN VIEW



OLEH :
FAIZA ANATHASYA EKA FALEN
SIB 2A / 12
2341760105

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG
2025

Langkah-langkah Praktikum 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

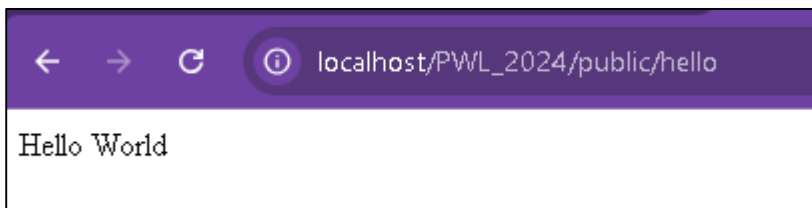
Kita akan menggunakan project minggu sebelumnya yaitu PWL_2024.

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

```
use Illuminate\Support\Facades\Route;

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.



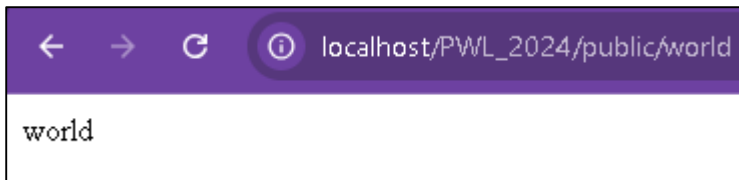
Code yang tambahkan di `route/web.php` pada bagian `get/hello` akan mengembalikan nilai berupa tampilan halaman yang menampilkan tulisan 'Hello World'.

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

```
use Illuminate\Support\Facades\Route;

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

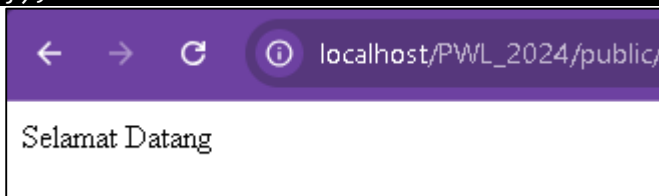
- e. Bukalah pada browser, tuliskan URL untuk memanggil route tersebut: `localhost/PWL_2024/public/world`. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.



Code yang tambahkan di route/web.php pada bagian get/world akan mengembalikan nilai berupa tampilan halaman yang menampilkan tulisan 'World'.

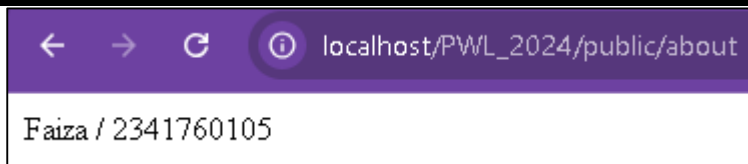
- f. Selanjutnya, cobalah membuat route '/' yang menampilkan pesan 'Selamat Datang'.

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



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

```
Route::get('/about', function () {  
    return 'Faiza / 2341760105';  
});
```



Langkah-langkah Praktikum Route Parameter :

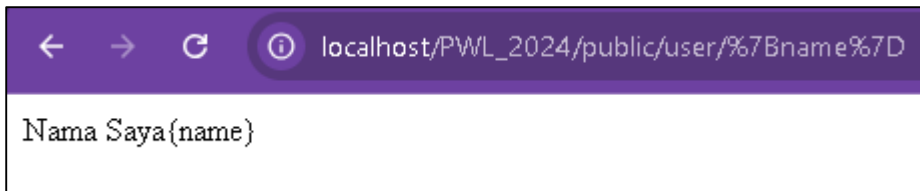
Untuk membuat routing dengan parameter dapat dilakukan dengan cara berikut ini.

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

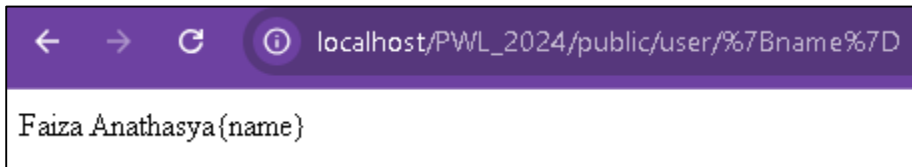
```
Route::get( '/user/{name}', function  
( $name ) {  
    return 'Nama saya  
' . $name; } );
```

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

Blm ada nama



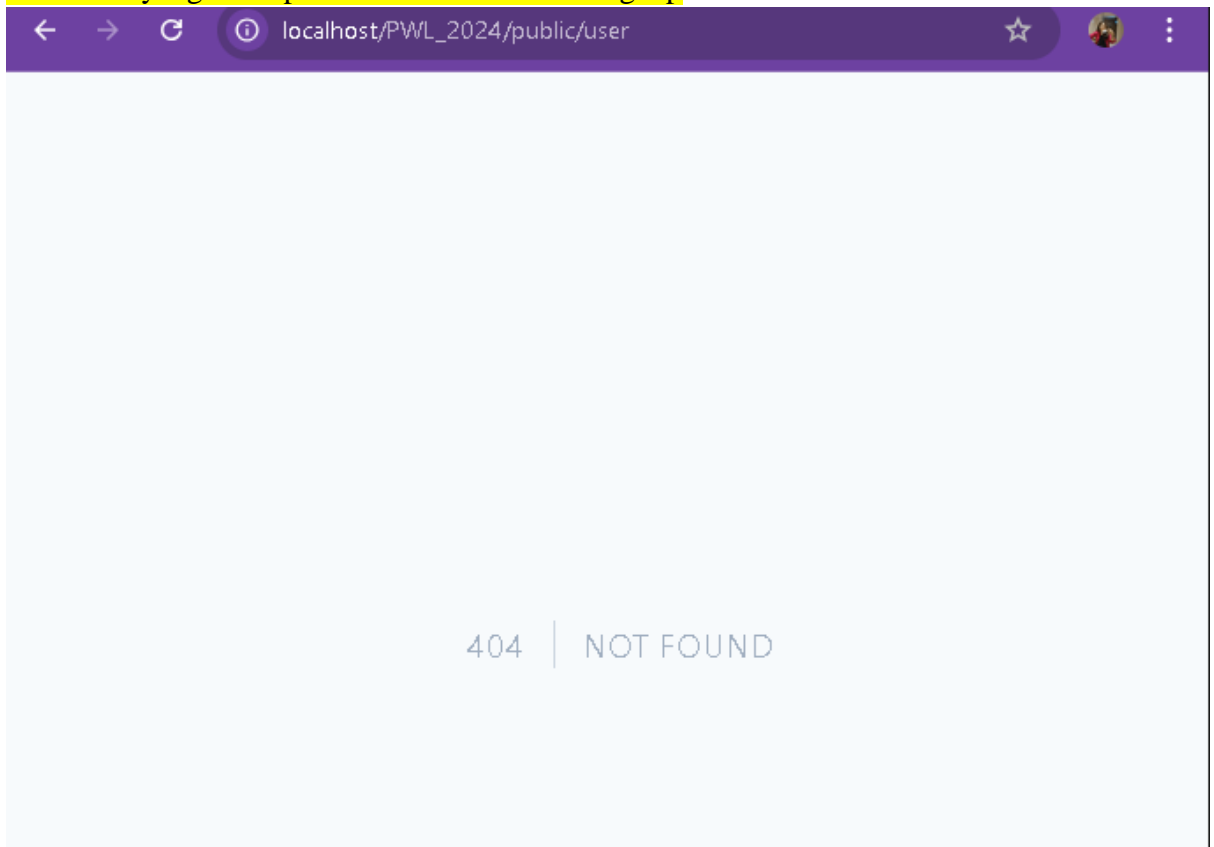
Bernama



Tampilan akan berisi nilai yang diisikan pada url.

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

Tidak ada yang ditampilkan karena url tidak lengkap



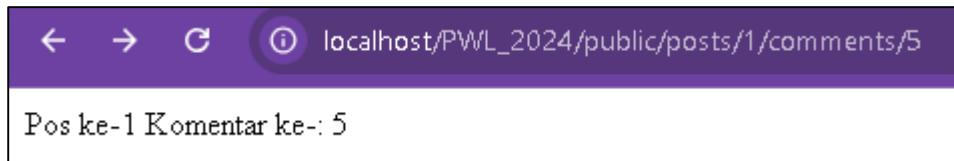
- d. Suatu route, juga bisa menerima lebih dari 1 parameter seperti kode berikut ini.

Route menerima parameter \$postId dan juga \$comment.

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

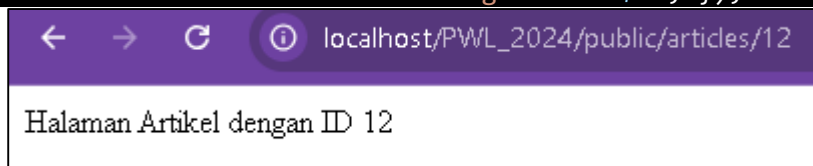
- e. Jalankan kode dengan menuliskan URL untuk memanggil route tersebut: **localhost/PWL_2024/public/posts/1/comments/5**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

Dapat menuliskan isi tampilan pada url



- f. 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; });
```



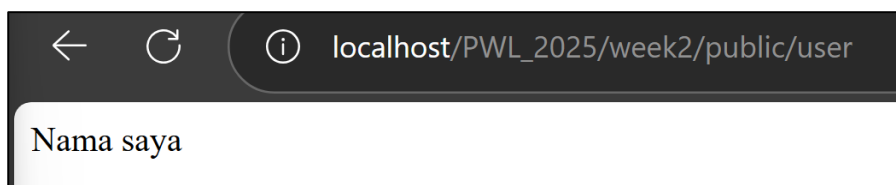
Langkah-langkah Praktikum Optional Parameter :

Untuk membuat routing dengan optional parameter dapat dilakukan dengan cara berikut ini.

- a. Kita akan memanggil route `/user` sekaligus mengirimkan parameter berupa nama user `$name` dimana parameternya bersifat opsional.

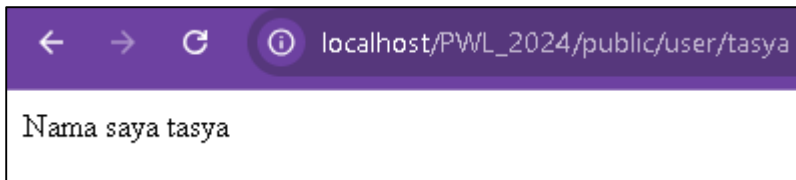
```
Route::get('/user/{name?}', function  
($name=null) {  
    return 'Nama saya '.$name;  
});
```

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



Halaman menampilkan output Nama Saya.

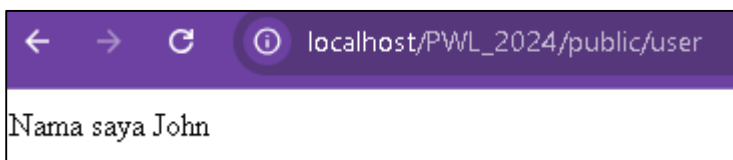
- c. Selanjutnya tuliskan URL: **localhost/PWL_2024/public/user>NamaAnda**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda



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

```
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.



Langkah-langkah Praktikum 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.

```
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:

```
<?php
namespace App\Http\Controllers;

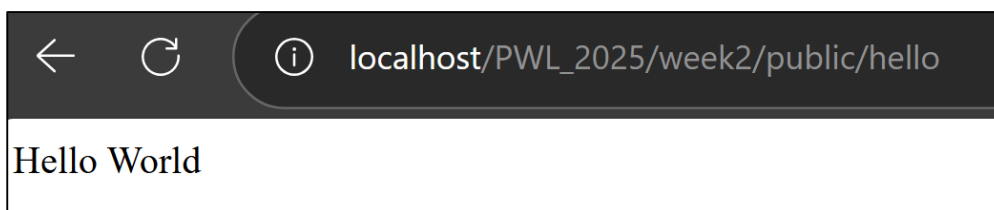
use Illuminate\Http\Request;

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::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.



- 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		

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

← ↻ ⓘ localhost/PWL_2025/week2/public/

Selamat Datang

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

← ↻ ⓘ localhost/PWL_2025/week2/public/about

Nama: Faiza Anathasya, NIM: 2341760105

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

← ↻ ⓘ localhost/PWL_2025/week2/public/articles/1

Halaman Artikel dengan Id 1

- 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.

Home Controller

```
<?php
```




```
namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    // Fungsi bawaan Single Action Controller
    public function __invoke()
    {
        return 'Selamat Datang'; // Menampilkan pesan selamat datang
    }
}
```

About Controller

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AboutController extends Controller
{
    public function __invoke()
    {
        return 'Nama: [Isi Nama Anda], NIM: [Isi NIM Anda]'; // Menampilkan
nama dan NIM
    }
}
```

Article Controller

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

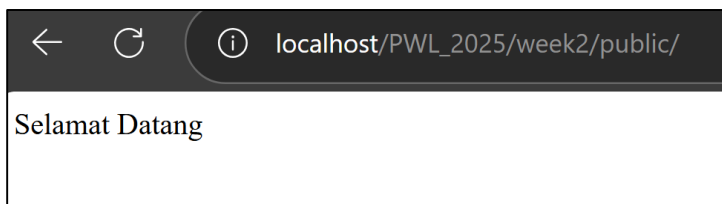
class ArticleController extends Controller
{
```



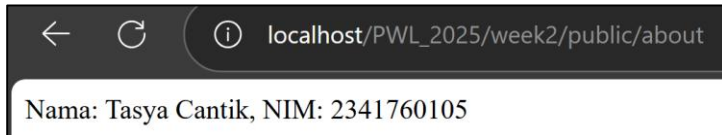
```
public function __invoke($id)
{
    return "Halaman Artikel dengan Id $id"; // Menampilkan halaman artikel
    dengan ID dinamis
}
}
```

Route :

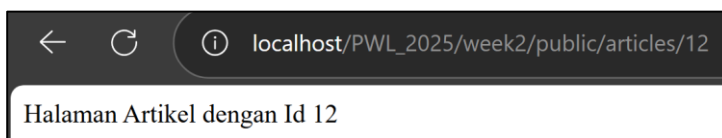
```
Route::get('/', HomeController::class);
```



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



```
Route::get('/articles/{id}', ArticleController::class);
```



Langkah-langkah Praktikum Resource Controller:

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

```
php artisan make:controller PhotoController --resource
```

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

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

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	/api/user		Closure	api
	GET HEAD	photos	photos.index	App\Http\Controllers\PhotoController@index	auth:api
	POST	photos	photos.store	App\Http\Controllers\PhotoController@store	web
	GET HEAD	photos/create	photos.create	App\Http\Controllers\PhotoController@create	web
	GET HEAD	photos/{photo}	photos.show	App\Http\Controllers\PhotoController@show	web
	PUT PATCH	photos/{photo}	photos.update	App\Http\Controllers\PhotoController@update	web
	DELETE	photos/{photo}	photos.destroy	App\Http\Controllers\PhotoController@destroy	web
	GET HEAD	photos/{photo}/edit	photos.edit	App\Http\Controllers\PhotoController@edit	web
	GET HEAD POST PUT PATCH DELETE OPTIONS	specialMahasiswa		Closure	web
	GET POST HEAD	specialUrl		Closure	web

```
php artisan route:list  
  
GET|HEAD / ..... HomeController  
POST _ignition/execute-solution ignition.executeSolution > Spatie\LaravelI...  
GET|HEAD _ignition/health-check ignition.healthCheck > Spatie\LaravelIgnition ...  
POST _ignition/update-config ignition.updateConfig > Spatie\LaravelIgnitio...  
GET|HEAD about ..... AboutController  
GET|HEAD api/user .....  
GET|HEAD articles/{id} ..... ArticleController  
GET|HEAD hello ..... WelcomeController@hello  
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 > CsrfCooki...  
GET|HEAD user/{name?} .....  
GET|HEAD user/{name} .....  
GET|HEAD welcome .....  
GET|HEAD world .....  
  
Showing [21] 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' ]);
```



Langkah-langkah Praktikum 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>
```

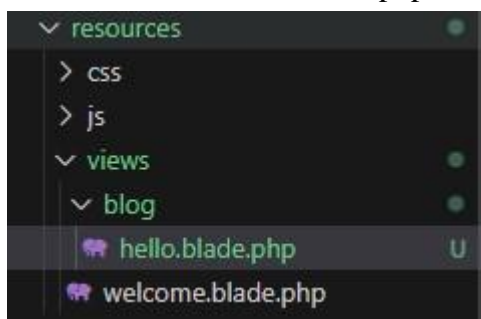
- b. View tersebut dapat dijalankan melalui Routing, dimana *route* akan memanggil View sesuai dengan nama *file* tanpa 'blade.php'. (Catatan: Gantilah Andi dengan nama Anda)

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

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

Langkah-langkah Praktikum 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' =>
'Andi']); });
```



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

Langkah-langkah Praktikum 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.

Langkah-langkah Praktikum 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');
    }
}
```

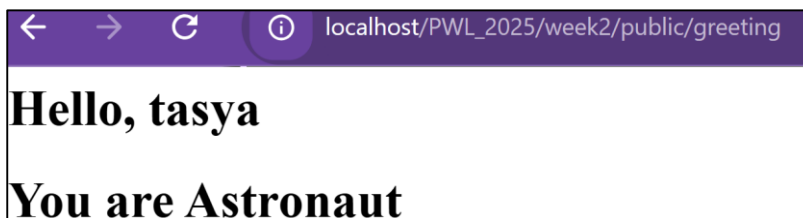


```
}
```

- 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.



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.

- Instalasi dan create project

```
PS C:\laragon\www\PWL_2025\POS> composer create-project --prefer-dist laravel/laravel .
0.*"
>>
Creating a "laravel/laravel" project at "."
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in C:\laragon\www\PWL_2025\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 (0.12.3)
- Locking carbonphp/carbon-doctrine-types (2.1.0)
- Locking dflydev/dot-access-data (v3.0.3)
- Locking doctrine/inflector (2.0.10)
- Locking doctrine/lexer (3.0.1)
- Locking dragonmantank/cron-expression (v3.4.0)
- Locking egulias/email-validator (4.0.3)
- Locking fakerphp/faker (v1.24.1)
- Locking filp/whoops (2.18.0)
- Locking fruitcake/php-cors (v1.3.0)
- Locking graham-campbell/result-type (v1.1.3)
```



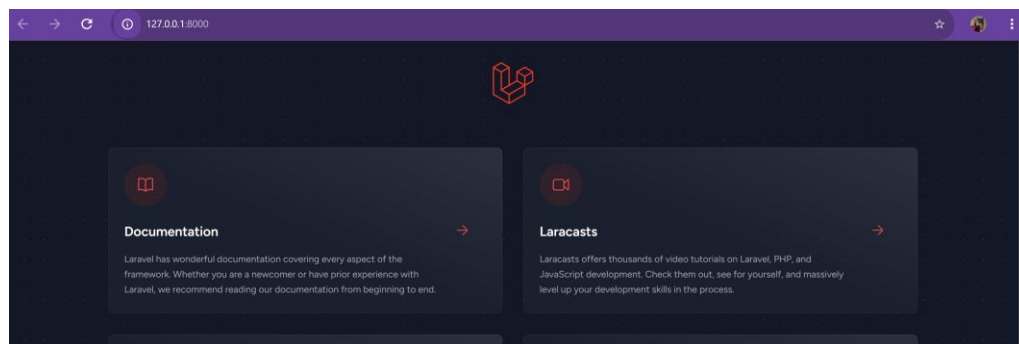
- **Menjalankan php artisan serve**

```
php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

2025-03-17 22:03:48 .....
2025-03-17 22:03:52 /favicon.ico .....
```



- **Membuat Controller**

```
PS C:\laragon\www\PWL_2025\POS> php artisan make:controller HomeController

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

PS C:\laragon\www\PWL_2025\POS> php artisan make:controller ProductController

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

PS C:\laragon\www\PWL_2025\POS> php artisan make:controller UserController

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

PS C:\laragon\www\PWL_2025\POS> php artisan make:controller SalesController

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

- **Membuat Route**

```
<?php

// Import class yang diperlukan
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\HomeController;
use App\Http\Controllers\ProductController;
use App\Http\Controllers\SalesController;
use App\Http\Controllers\UserController;

// Route untuk halaman utama (homepage)
```



```
Route::get('/', [HomeController::class, 'index']); // Akses URL: '/',
menjalankan method 'index' di HomeController

// Group route dengan prefix 'products'
Route::prefix('products')->group(function () {

    // Route untuk kategori Food & Beverages
    Route::get('/food-beverages', [ProductController::class,
'foodBeverages'])->name('category.food-beverage');
    // Akses URL: '/products/food-beverages', method 'foodBeverages' di
ProductController, nama route: category.food-beverage

    // Route untuk kategori Beauty & Health
    Route::get('/beauty-health', [ProductController::class,
'beautyHealth'])->name('category.beauty-health');
    // Akses URL: '/products/beauty-health', method 'beautyHealth', nama
route: category.beauty-health

    // Route untuk kategori Home Care
    Route::get('/home-care', [ProductController::class, 'homeCare'])->
name('category.home-care');
    // Akses URL: '/products/home-care', method 'homeCare', nama route:
category.home-care

    // Route untuk kategori Baby & Kid
    Route::get('/baby-kid', [ProductController::class, 'babyKid'])->
name('category.baby-kid');
    // Akses URL: '/products/baby-kid', method 'babyKid', nama route:
category.baby-kid
});

// Route dengan parameter dinamis untuk user
Route::get('/user/{id}/name/{name}', [UserController::class,
'showProfile'])->name('user.profile');
// Akses URL: '/user/123/name/john', menjalankan method 'showProfile'
dengan parameter id dan name, nama route: user.profile

// Route untuk halaman sales
Route::get('/sales', [SalesController::class, 'index'])->name('sales');
// Akses URL: '/sales', menjalankan method 'index' di SalesController,
nama route: sales
```

- **Membuat Views**



✓ **Baby & kids**

```
resources > views > products > baby-kid.blade.php > .
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Baby & Kids</title>
5 </head>
6 <body>
7 | <h1>Baby & Kids Products</h1>
8 </body>
9 </html>
```

✓ **Beauty & Health**

```
resources > views > products > beauty-health.blade.php > ..
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Beauty & Health</title>
5 </head>
6 <body>
7 | <h1>Beauty & Health Products</h1>
8 </body>
9 </html>
```

✓ **Food & Beverages**

```
resources > views > products > food-beverages.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Food & Beverages</title>
5 </head>
6 <body>
7 | <h1>Food & Beverages Products</h1>
8 </body>
9 </html>
```

✓ **Home & care**




```
resources > views > products > home-care.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Home Care</title>
5 </head>
6 <body>
7 | <h1>Home Care Products</h1>
8 </body>
9 </html>
```



✓ **Home**

```
resources > views > home.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Home Page</title>
8 </head>
9 <body>
10  <h1> HI! WELCOME TO HOME PAGE POS</h1>
11  <h3> PRODUCT CATEGORIES : </h3>
12  <ul>
13    <li><a href="{{ route('category.food-beverage') }}">Food & Beverag
14    <li><a href="{{ route('category.beauty-health') }}">Beauty & Healt
15    <li><a href="{{ route('category.home-care') }}">Home Care</a></li>
16    <li><a href="{{ route('category.baby-kid') }}">Baby & Kids</a></li>
17  </ul>
18
19 </body>
20 </html>
```

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

1	<p>Halaman Home</p> <p>Menampilkan halaman awal website</p> 
2	<p>Halaman Products</p> <p>Menampilkan daftar product (route prefix)</p> <p>/category/food-beverage</p>  <p>/category/beauty-health</p> 



	<p>/category/home-care</p> <p>← → ↺ ⓘ localhost/PWL_2025/POS/public/products/home-care</p> <h2>Home Care Products</h2> <p>/category/baby-kid</p> <p>← → ↺ ⓘ localhost/PWL_2025/POS/public/products/baby-kid</p> <h2>Baby & Kids Products</h2>
3	<p>Halaman User</p> <p>Menampilkan profil pengguna (route param)</p> <p>/user/{id}/name/{name}</p> <p>← → ↺ ⓘ 127.0.0.1:8000/user/12/tasya</p> <h2>Profil Pengguna</h2> <p>ID Pengguna: 12</p> <p>Nama Pengguna: tasya</p>
4	<p>Halaman Penjualan</p> <p>Menampilkan halaman transaksi POS</p> <p>← → ↺ ⓘ localhost/PWL_2025/POS/public/sales</p> <h2>Halaman Transaksi Penjualan (POS)</h2> <p>Selamat datang di sistem kasir!</p> <h3>Contoh Produk Terjual</h3> <ul style="list-style-type: none">• Produk: Kopi Susu - Jumlah: 2 - Total: Rp 20.000• Produk: Roti Coklat - Jumlah: 1 - Total: Rp 7.000 <p>Total Pembayaran: Rp 27.000</p>



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.