

PRAKTIKUM PEMROGRAMAN WEB LANJUT

TUGAS PERTEMUAN 2

ROUTING, CONTROLLER, DAN VIEW

DISUSUN OLEH :

GILANG PURNOMO

NIM:2341720042



PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

FEBRUARI 2025

Routing

- Basic Routing

Langkah -Langkah Praktikum:

- 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.

Jawab :



Hello World

Jadi, jika menambahkan kode diatas akan membuat route sederhana dengan mengakses url: `http://pwl_2024.test/hello` yang akan menampilkan teks teks “Hello World” di halaman browser.

- 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.

Jawab :



World

Jadi, ini sama seperti perintah pada soal sebelumnya, tetapi path dan teksnya di ganti dengan world. Ketika memanggil url: `http://pwl_2024.test/world` maka yang ditampilkan di halaman browser yaitu teks “World”.

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

Jawab :

Kode program:

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

Output :



Selamat Datang

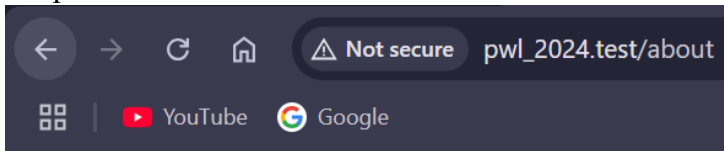
- g. Kemudian buatlah route `'/about'` yang akan menampilkan NIM dan nama Anda.

Jawab :

Kode program :

```
Route::get('/about', function () {  
    return 'Nama : Gilang Purnomo <br> NIM : 2341720042';  
});
```

Output :



Nama : Gilang Purnomo

NIM : 2341720042

- Route Parameters

Langkah -Langkah Praktikum:

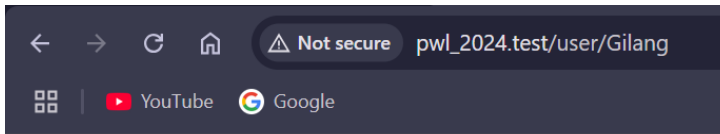
Untuk membuat routing dengan parameter dapat dilakukan dengan cara berikut ini.

- 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.

Jawab :



Nama saya Gilang

Jadi, dari kode tersebut bahwa {name} ini adalah parameter dinamis dalam route, yang berarti nilainya bisa berubah sesuai inputannya. Kalau \$name ini akan mengambil nilai dari parameter tersebut.

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

Jawab :



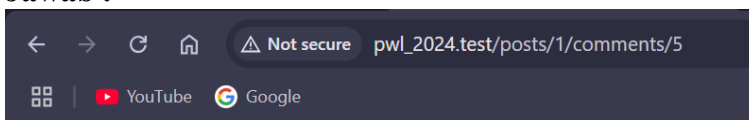
Laravel menampilkan halaman errornya, karena route /user/ ini harus ada parameternya dalam Url. Jika tanpa parameter, Laravel tidak dapat mencocokkan request dengan routenya, sehingga permintaan dianggap tidak valid.

- 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.

Jawab :



Pos ke-1 Komentar ke-: 5

Jadi, {post} ini untuk ID postingan dan {comment} ini untuk ID komentar dalam postingan. Oleh karena itu Laravel mengambil nilai 1 sebagai \$postId dan 5 sebagai \$commentId.

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

Jawab :

Kode Program :

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

Output :



Halaman Artikel dengan ID 09

- Optional Parameters

Langkah -Langkah Praktikum:

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.

Jawab :



Nama saya

Hanya muncul “Nama saya” saja, karena parameter {name} tidak diberikan dan defaultnya diberi null. Jadi \$name bernilai null.

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

Jawab :



Nama saya Gilang

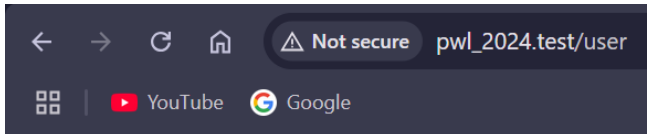
Muncul nama saya. Jadi Laravel mengambil “Gilang” sebagai nilai {name} dan \$name jadi memiliki nilai yaitu “Gilang”.

- 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.

Jawab :



Nama saya John

Jadi, jika parameter {name} ini tidak diberikan maka \$name akan menggunakan nilai defaultnya yaitu “John”.

Controller

- Membuat Controller

Langkah -Langkah Praktikum:

- 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.

Jawab :



Hello World

Dengan menggunakan controller, pengelolaan logika menjadi terstruktur dibandingkan harus menuliskannya langsung di web.php. Dengan perubahan tersebut, Laravel memanggil method hello() yang ada di WelcomeController ketika /hello diakses.

- 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' <i>PageController : index</i>		
/about		Tampilkan Nama dan NIM <i>PageController : about</i>		
/articles/{id}		Tampilkan halaman dinamis 'Halaman Artikel dengan Id {id}' id diganti sesuai input dari url <i>PageController : articles</i>		

Jawab :

1. Buat file *PageController*

```
PS C:\laragon\www\PWL_2024> php artisan make:controller PageController
INFO Controller [C:\laragon\www\PWL_2024\app\Http\Controllers\PageController.php] created successfully.
```

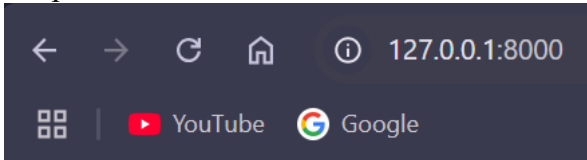
2. Ubah isi *PageController*

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class PageController extends Controller
8  {
9      public function index() {
10         return 'Selamat Datang';
11     }
12
13     public function about() {
14         return 'Nama : Gilang Purnomo <br> NIM : 2341720042';
15     }
16
17     public function articles($id) {
18         return 'Halaman Artikel dengan Id ' . $id;
19     }
20 }
21
```

3. Ubah juga routenya yang berada di *web.php*

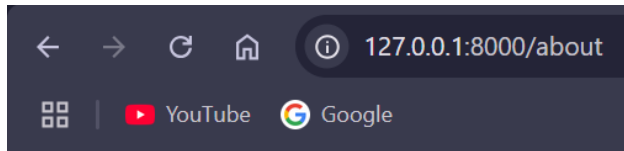
```
Route::get('/', [PageController::class, 'index']);
Route::get('/about', [PageController::class, 'about']);
Route::get('/articles/{id}', [PageController::class, 'articles']);
```

4. Output / :



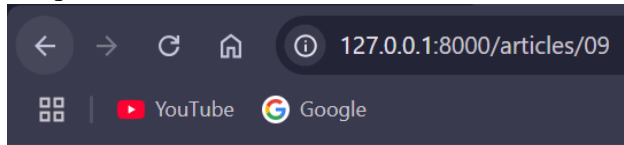
Selamat Datang

Output /about :



Nama : Gilang Purnomo
NIM : 2341720042

Output /articles :



Halaman Artikel dengan Id 09

- 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.

Jawab :

1. Buat filenya di Controller

- *HomeController*

```
PS C:\laragon\www\PWL_2024> php artisan make:controller HomeController --invokable  
INFO Controller [C:\laragon\www\PWL_2024\app\Http\Controllers\HomeController.php] created successfully.
```

- *AboutController*

```
PS C:\laragon\www\PWL_2024> php artisan make:controller AboutController --invokable  
INFO Controller [C:\laragon\www\PWL_2024\app\Http\Controllers\AboutController.php] created successfully.
```

- *ArticleController*

```
PS C:\laragon\www\PWL_2024> php artisan make:controller ArticleController --invokable  
INFO Controller [C:\laragon\www\PWL_2024\app\Http\Controllers\ArticleController.php] created successfully.
```

2. Ubah isi setiap filenya

- *HomeController*

```
1 <?php  
2  
3 namespace App\Http\Controllers;  
4  
5 use Illuminate\Http\Request;  
6  
7 class HomeController extends Controller  
8 {  
9     public function __invoke()  
10     {  
11         return 'Selamat Datang';  
12     }  
13 }  
14
```

- AboutController

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class AboutController extends Controller
8  {
9      public function __invoke()
10     {
11         return 'Nama: Gilang Purnomo <br> NIM: 2341720042';
12     }
13 }
14
```

- ArticleController

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class ArticleController extends Controller
8  {
9      public function __invoke($id)
10     {
11         return 'Halaman Artikel dengan Id ' . $id;
12     }
13 }
14
```

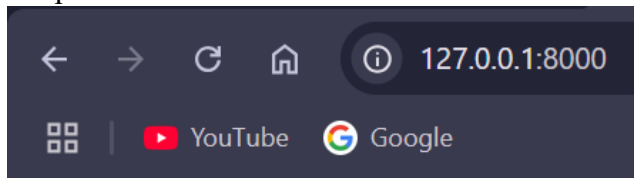
3. Ubah route di web.php

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

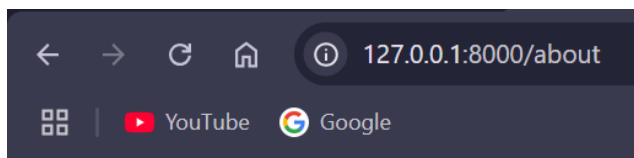
```
Route::get('/about', AboutController::class);
```

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

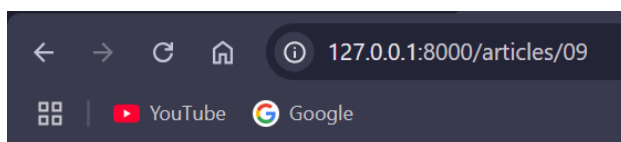
4. Output :



Selamat Datang



Nama: Gilang Purnomo
NIM: 2341720042



Halaman Artikel dengan Id 09

- Resource Controller

Langkah -Langkah Praktikum:

- a. 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.

Jawab :

```
PS C:\laragon\www\PWL_2024> php artisan make:controller PhotoController --resource
INFO Controller [C:\laragon\www\PWL_2024\app\Http\Controllers\PhotoController.php] created successfully.
```

- 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);
```

Jawab :

```
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	web
	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

Jawab :

```
PS C:\laragon\www\PWL_2024> php artisan route:list

GET|HEAD / ..... HomeController
POST _ignition/execute-solution ignition.executeSolution > Spatie\LaravelIgnition > Exec...
GET|HEAD _ignition/health-check ignition.healthCheck > Spatie\LaravelIgnition > HealthCheckC...
POST _ignition/update-config ignition.updateConfig > Spatie\LaravelIgnition > UpdateConf...
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 > CsrfCookieController@sh...
GET|HEAD user/{name?} .....
GET|HEAD world .....

Showing [19] 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'
]);
```

View

- Membuat View

Langkah -Langkah Praktikum:

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

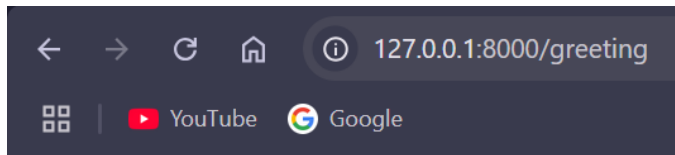
```
<!-- View pada resources/views/hello.blade.php -->
<html>
  <body>
    <h1> Hello, {{ $name }} </h1>
  </body>
</html>
```

- 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.

Jawab :



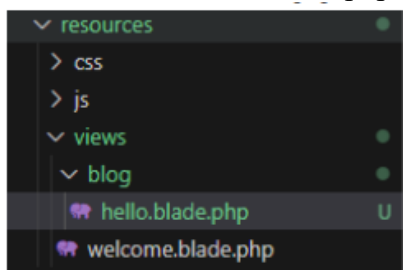
Hello, Gilang Purnomo

Setelah membuka url : <http://127.0.0.1:8000/greeting>, halaman akan menampilkan teks yang berada di view yang bernama hello.blade.php. Variabel \$name yang dikirim dari route juga akan ditampilkan pada view.

- View dalam direktori

Langkah -Langkah Praktikum:

- 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 () {
```

```

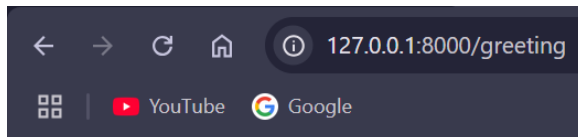
        return view('blog.hello', ['name' => 'Andi']);
    });

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

```

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

Jawab :



Hello, Gilang Purnomo

Jadi ini kurang lebih sama seperti praktikum diatas, tetapi beda pemanggilan filenya. Dengan memakai view('blog.hello'), maka Laravel mencari filenya di resources/views/blog/hello.blade.php. Menggunakan notasi (.) di view() adalah cara untuk mengakses view dalam subdirektori

- Menampilkan View dari Controller

Langkah -Langkah Praktikum:

- 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']);
    }
}

```

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

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

```

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

```

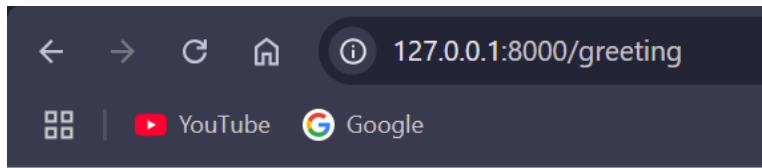
Route::get('/greeting', [WelcomeController::class, 'greeting']);

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.

Jawab :



Hello, Gilang Purnomo

Jadi, Route /greeting diarahkan ke file WelcomeController dan memanggil fungsi greeting(). Lalu Laravel mencari resources/views/blog/hello.blade.php untuk tampilan.

- Meneruskan data ke view

Langkah -Langkah Praktikum:

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

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

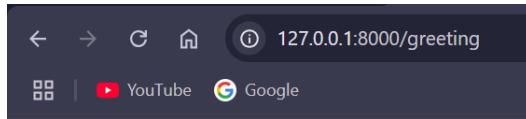
    public function greeting(){
        return view('blog.hello')
            ->with('name','Andi')
            ->with('occupation','Astronaut');
    }
}
```

- 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.

Jawab :



Hello, Gilang Purnomo

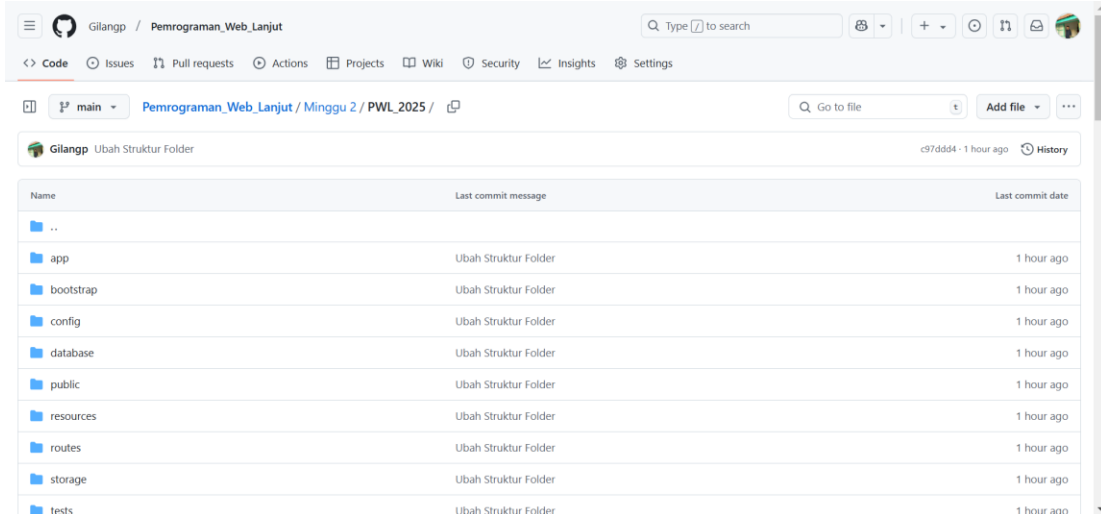
You are Astronaut

Penggunaan method with() untuk mengirim beberapa variabel ke view. Variabel name bernilai “Gilang Purnomo” dan occupation bernilai "Astronaut". Lalu {{ \$name }} akan menampilkan nilai dari variabel name, sedangkan {{ \$occupation }} akan menampilkan nilai dari variabel occupation.

SOAL PRAKTIKUM

1. Jalankan Langkah-langkah Praktikum pada jobsheet di atas. Lakukan sinkronisasi perubahan pada project PWL_2024 ke Github.

Jawab :



2. Buatlah project baru dengan nama POS. Project ini merupakan sebuah aplikasi Point of Sales yang digunakan untuk membantu penjualan.

Jawab :

```
C:\laragon\www
λ 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 ^8.2 which is not satisfied by your platform.
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
```

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

1	Halaman Home Menampilkan halaman awal website
2	Halaman Products Menampilkan daftar product (route prefix) /category/food-beverage /category/beauty-health /category/home-care /category/baby-kid
3	Halaman User Menampilkan profil pengguna (route param) /user/{id}/name/{name}
4	Halaman Penjualan Menampilkan halaman transaksi POS

Jawab :

- Route :

```
use App\Http\Controllers\HomeController;
use App\Http\Controllers\ProductController;
use App\Http\Controllers\UserController;
use App\Http\Controllers\SalesController;
```

- Controller :

```
PS C:\laragon\www\POS> php artisan make:controller HomeController
INFO Controller [C:\laragon\www\POS\app\Http\Controllers\HomeController.php] created successfully.

PS C:\laragon\www\POS> php artisan make:controller ProductControllerphp
INFO Controller [C:\laragon\www\POS\app\Http\Controllers\ProductControllerphp.php] created successfully.

PS C:\laragon\www\POS> php artisan make:controller UserController
INFO Controller [C:\laragon\www\POS\app\Http\Controllers\UserController.php] created successfully.

PS C:\laragon\www\POS> php artisan make:controller SalesController
INFO Controller [C:\laragon\www\POS\app\Http\Controllers\SalesController.php] created successfully.
```

- View :

```
views
├── category
│   ├── baby-kid.blade.php
│   ├── beauty-health.blade.php
│   ├── food-beverage.blade.php
│   └── home-care.blade.php
├── sales
│   └── index.blade.php
└── user
    ├── profile.blade.php
    └── home.blade.php
```

4. Route tersebut menjalankan fungsi pada Controller yang berbeda di setiap halaman.

Jawab :

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

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']);
});

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

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

5. Fungsi pada Controller akan memanggil view sesuai halaman yang akan ditampilkan.

Jawab :

- Home Controller :

```
class HomeController extends Controller
{
    public function index()
    {
        return view('home');
    }
}
```

- Product Controller

```
class ProductController extends Controller
{
    public function foodBeverage()
    {
        return view('category.food-beverage');
    }

    public function beautyHealth()
    {
        return view('category.beauty-health');
    }

    public function homeCare()
    {
        return view('category.home-care');
    }

    public function babyKid()
    {
        return view('category.baby-kid');
    }
}
```

- User Controller :

```
class UserController extends Controller
{
    public function profile($id, $name)
    {
        return view('user.profile', compact('id', 'name'));
    }
}
```

- Sales Controller :

```
class SalesController extends Controller
{
    public function index()
    {
        return view('sales.index');
    }
}
```

Output :

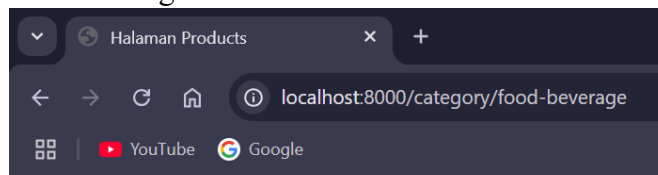
- Halaman Home



Selamat Datang di Sistem Point of Sales (POS)

- Halaman Products

- food-beverage

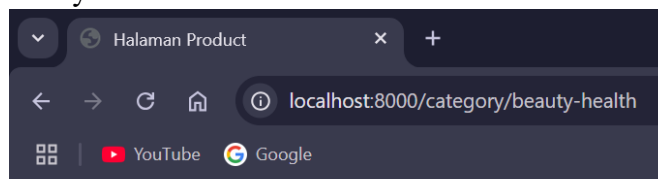


Kategori Makanan & Minuman

Produk Tersedia:

- Makanan ringan & camilan
- Makanan instan
- Minuman sehat & jus
- Susu & produk olahan susu
- Bahan makanan pokok

- beauty-health

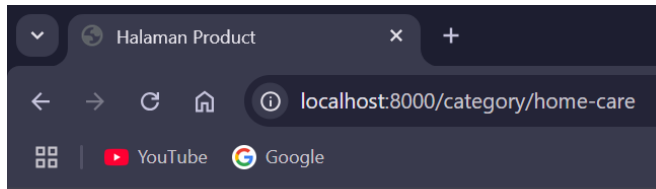


Kategori Kecantikan & Kesehatan

Produk Tersedia:

- Produk perawatan kulit (skincare)
- Produk perawatan rambut
- Vitamin & suplemen kesehatan
- Alat kecantikan & kosmetik

- home-care

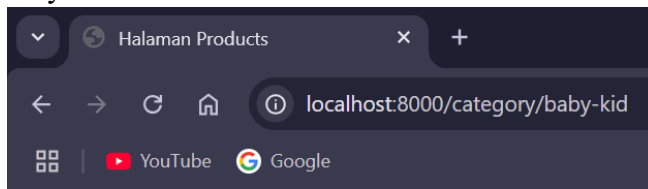


Kategori Perawatan Rumah

Produk Tersedia:

- Pembersih lantai & kaca
- Deterjen & pewangi pakaian
- Pembersih dapur & kamar mandi
- Alat kebersihan rumah tangga

- baby-kid

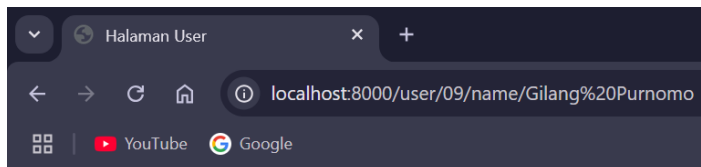


Kategori Bayi & Anak

Produk Tersedia:

- Popok bayi
- Susunya & Makanan bayi
- Mainan
- Pakaian bayi dan anak

- Halaman User

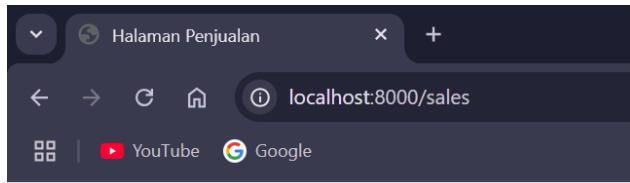


Profil Pengguna

ID: 09

Name: Gilang Purnomo

- Halaman Penjualan



Halaman Transaksi Penjualan

Informasi Transaksi

- Nama Produk :
- Jumlah :
- Harga :
- Metode Pembayaran :
- Total Pembayaran :

6. Simpan setiap perubahan yang dilakukan pada project POS pada Git, sinkronisasi perubahan ke Github.

Jawab :

