

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



OLEH:

LOVELYTA SEKARAYU KRISDIYANTI

KELAS 2B SIB / 11 (2341760081)

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No. 9 Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur

65141

Praktikum

1. 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

Kita akan menggunakan project minggu sebelumnya yaitu PWL_2024.

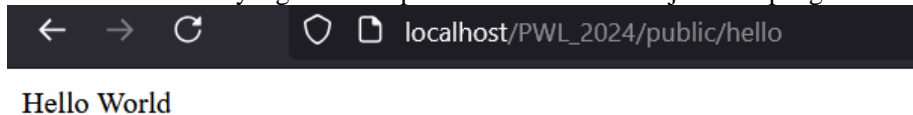
- b. Buka file routes/web.php. Tambahkan sebuah route untuk nomor 1 seperti di bawah ini:

```
<?php

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.

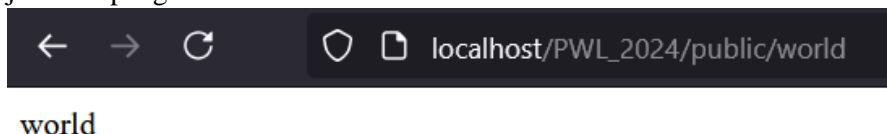


- Get /hello akan mengembalikan teks 'Hello World' sehingga akan mencetak kalimat tersebut seperti gambar di atas

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

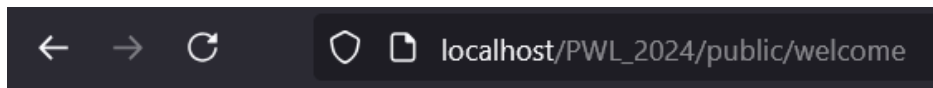
```
Route::get('/world', function () {
    return 'world';
});
```

- 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.



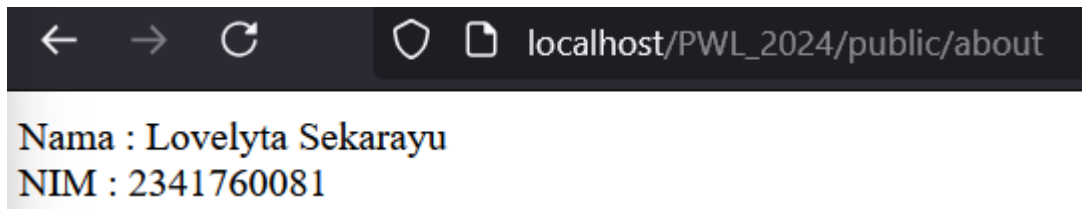
- Get /world akan mengembalikan teks 'world' sehingga akan mencetak kalimat tersebut seperti gambar di atas
- f. Selanjutnya, cobalah membuat route '/' yang menampilkan pesan 'Selamat Datang'.

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



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

```
Route::get('/about', function () {  
    return 'Nama : Lovelyta Sekarayu <br> NIM : 2341760081';  
});
```

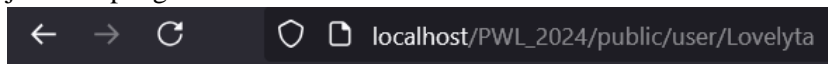


• Route Parameters

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

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

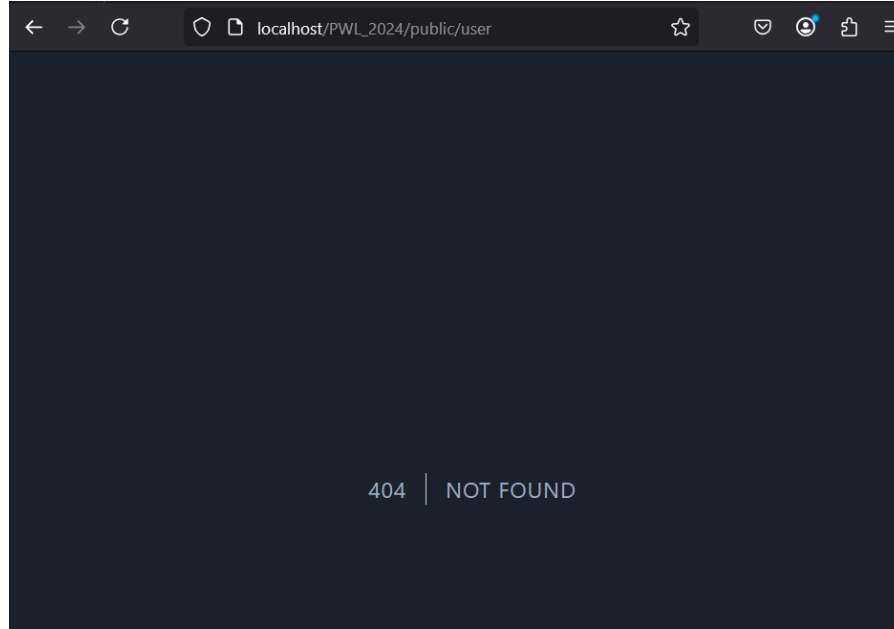
- b. Jalankan kode dengan menuliskan URL untuk memanggil route tersebut: **localhost/PWL_2024/public/user>NamaAnda**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Nama saya Lovelyta

- Ketika user mengganti "NamaAnda" dengan nama tertentu seperti gambar di atas, maka hasilnya nama tersebut muncul. Ini karena, \$name pada return memanggil \$name yang berada pada URL

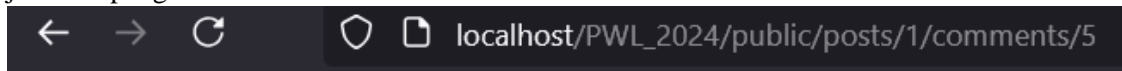
- c. Selanjutnya, coba tuliskan URL: localhost/PWL_2024/public/user/. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda



- Hasilnya not found atau error, karena tidak ada nilai kembalian pada URL yang sangat dibutuhkan di `return 'Nama saya ' . $name;`
- 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 'Post 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.

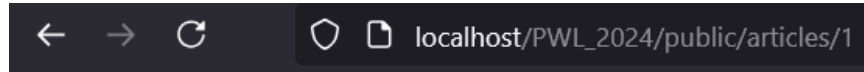


Post ke- 1 Komentar ke- 5

- `{post}` Adalah parameter pertama yang mewakili ID dari sebuah post, `{comment}` adalah parameter kedua yang mewakili ID dari sebuah komentar dalam post
 - `$postId` dan `$commentId` akan mengembalikannya seperti gambar di atas
- 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;
```

```
});
```



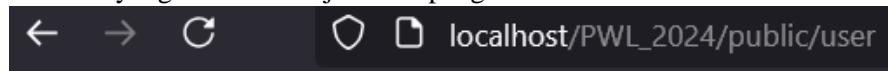
Halaman Artikel dengan ID 1

- **Optional Parameters:**

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

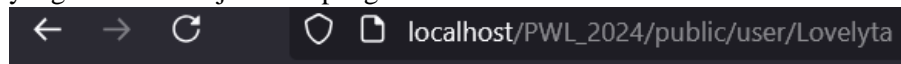
```
Route::get('/user/{name?}', function ($name = null) {  
    return 'Nama saya ' . $name;  
});
```

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



Nama saya

- Jika tidak mengisi nama pada URL, masih bisa dicetak seperti gambar di atas. Ini karena, terdapat (\$name = null) sehingga tetap bisa dicetak meskipun tidak memiliki nilai kembalian
- c. Selanjutnya tuliskan URL: localhost/PWL_2024/public/user>NamaAnda. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda

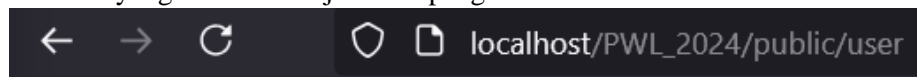


Nama saya Lovelyta

- Kalimat yang dicetak terdapat nama karena memiliki nilai pada URL
- 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.



Nama saya John

- Terdapat nama yang dicetak karena terdapat nilai kembalian pada (\$name = 'John')

2. 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 C:\laragon\www\PWL_2024> php artisan make:controller WelcomeController
```

```
INFO Controller [C:\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;
```

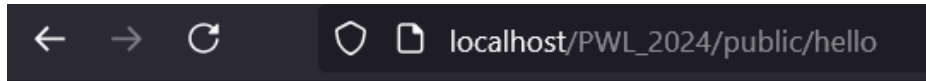
```
use Illuminate\Http\Request;
```

```
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::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

- Route akan memanggil class WelcomeController sehingga output yang keluar adalah yang terdapat pada class WelcomeController.php
- 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		

- Buat Controller

```
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.
```

- Controller

app > Http > Controllers > 🐞 PageController.php > 📁 PageController > 📖 about

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  class PageController extends Controller
6  {
7      //Method untuk halaman utama
8      public function index()
9      {
10         return "Selamat Datang";
11     }
12
13     // Method untuk halaman about
14     public function about()
15     {
16         return "Nama: [Isi Nama] <br> NIM: [Isi NIM]";
17     }
18
19     // Method untuk halaman artikel berdasarkan ID
20     public function articles($id)
21     {
22         return "Halaman Artikel dengan Id {$id}";
23     }
24 }
```

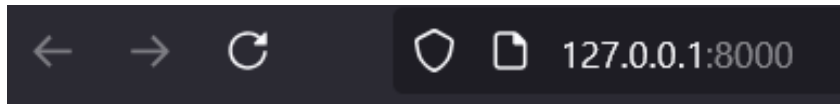
- Routes

routes > 🐞 web.php > ...

```
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\ItemController;
5  use App\Http\Controllers>WelcomeController;
6  use App\Http\Controllers\PageController;
7
8  // Rute untuk halaman utama
9  Route::get('/', [PageController::class, 'index']);
10
11 // Rute untuk halaman about
12 Route::get('/about', [PageController::class, 'about']);
13
14 // Rute untuk halaman artikel dengan parameter ID
15 Route::get('/articles/{id}', [PageController::class, 'articles']);
16
```

- Menjalankan program:

- <http://127.0.0.1:8000/> untuk menampilkan halaman utama



Selamat Datang

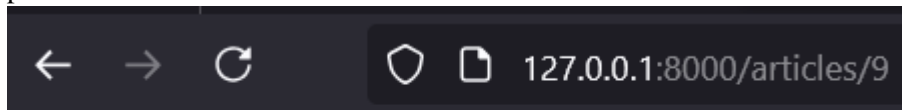
- <http://127.0.0.1:8000/about> untuk menampilkan halaman about



Nama : Lovelyta Sekarayu

NIM : 2341760081

- <http://127.0.0.1:8000/articles/{id}> untuk menampilkan halaman artikel dengan parameter ID



Halaman Artikel dengan ID 9

- 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.

- Buat Controller

```
PS C:\laragon\www\PWL_2024> php artisan make:controller HomeController --i  
nvokable
```

```
INFO Controller [C:\laragon\www\PWL_2024\app\Http\Controllers\HomeCont  
roller.php] created successfully.
```

```
PS C:\laragon\www\PWL_2024> php artisan make:controller AboutController --  
invokable
```

```
INFO Controller [C:\laragon\www\PWL_2024\app\Http\Controllers\AboutCon  
troller.php] created successfully.
```

```
PS C:\laragon\www\PWL_2024> php artisan make:controller ArticleController  
--invokable
```

```
INFO Controller [C:\laragon\www\PWL_2024\app\Http\Controllers\ArticleC  
ontroller.php] created successfully.
```

- Controller

➤ Halaman Utama

app > Http > Controllers > 🐞 HomeController.php > ...

```
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(Request $request)
10     {
11         return 'Selamat Datang';
12     }
13 }
14
```

➤ About

app > Http > Controllers > 🐞 AboutController.php > ...

```
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(Request $request)
10     {
11         return "Nama: [Isi Nama] <br> NIM: [Isi NIM]";
12     }
13 }
14
```

➤ halaman artikel dengan parameter ID

```

app > Http > Controllers > ArticleController.php > ...
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(Request $request)
10     {
11         $id = $request->input('id');
12         return "Halaman Artikel dengan Id $id";
13     }
14 }

```

- Routes

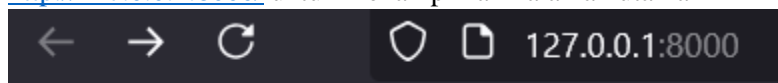
```

routes > web.php > ...
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\ItemController;
5  use App\Http\Controllers>WelcomeController;
6  use App\Http\Controllers\PageController;
7  use App\Http\Controllers\HomeController;
8  use App\Http\Controllers\AboutController;
9  use App\Http\Controllers\ArticleController;
10
11 // untuk halaman utama
12 Route::get('/', HomeController::class);
13
14 // untuk halaman about
15 Route::get('/about', AboutController::class);
16
17 // untuk halaman artikel berdasarkan ID
18 Route::get('/articles/{id}', ArticleController::class);

```

- Menjalankan program

- <http://127.0.0.1:8000/> untuk menampilkan halaman utama



Selamat Datang

- <http://127.0.0.1:8000/about> untuk menampilkan halaman about



Nama : Lovelyta Sekarayu
NIM : 2341760081

- <http://127.0.0.1:8000/articles/{id}> untuk menampilkan halaman artikel dengan parameter ID



Halaman Artikel dengan ID 8

❖ Resource Controller

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

```
PS C:\laragon\www\PWL_2024> 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

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

GET|HEAD / ..... ignition.executeSolution ..... ignition.executeSolution ..... Spatie\LaravelIgnition ..... ExecuteSolutionController
GET|HEAD /_ignition/health-check ..... ignition.healthCheck ..... Spatie\LaravelIgnition ..... HealthCheckController
POST /_ignition/update-config ..... ignition.updateConfig ..... Spatie\LaravelIgnition ..... UpdateConfigController
GET|HEAD /about ..... ..... ..... WelcomeController@hello
GET|HEAD /api/user ..... ..... ..... .....
GET|HEAD /articles/{id} ..... ..... ..... .....
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
```

- 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.

3. VIEW

❖ Membuat View

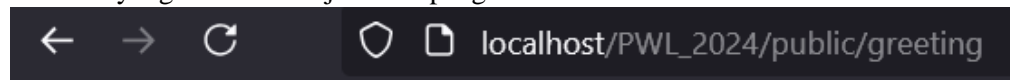
- a. Pada direktori app/resources/views, buatlah file hello.blade.php.

```
<!-- 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)

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

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

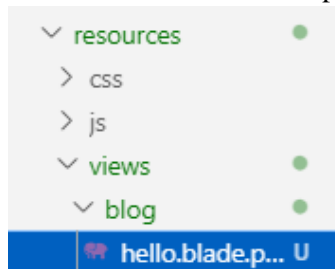


Hello, Lovelyta Sekarayu

- ['name' => 'Lovelyta Sekarayu'] mengirimkan variabel name dengan nilai "Lovelyta Sekarayu". Sehingga Blade akan mengganti {{ \$name }} dengan "Lovelyta Sekarayu" dalam tampilan.

❖ View dalam direktori

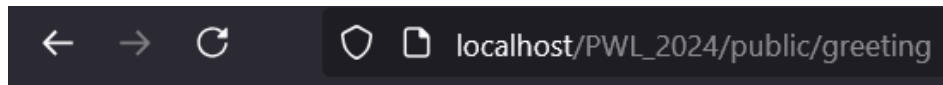
- Buatlah direktori blog di dalam direktori views.
- Pindahkan file hello.blade.php ke dalam direktori blog.



- c. Selanjutnya lakukan perubahan pada route.

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

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



Hello, Lovelyta Sekarayu

- Fungsi `view('blog.hello', ['name' => 'Lovelyta Sekarayu'])`; `blog.hello` berarti Laravel mencari file `resources/views/blog/hello.blade.php`. Variabel `$name` dikirimkan ke view dengan nilai 'Lovelyta Sekarayu'.

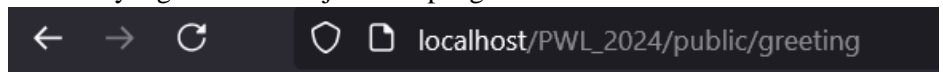
❖ Menampilkan View dari Controller

- 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' => 'Lovelyta Sekarayu']);
    }
}
```
- Ubah route `/greeting` dan arahkan ke `WelcomeController` pada fungsi `greeting`.

```
Route::get('/greeting', [WelcomeController::class, 'greeting']);
```
- Jalankan code dengan membuka url `localhost/PWL_2024/public/greeting`. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Hello, Lovelyta Sekarayu

- Laravel akan menjalankan method `greeting()` dalam `WelcomeController` saat URL `/greeting` diakses.
- ❖ Meneruskan data ke view
- 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');
    }
}

```

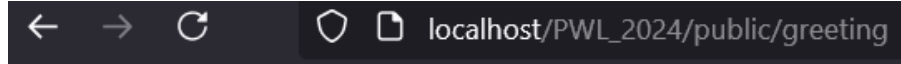
- b. Ubah hello.blade.php agar dapat menampilkan dua parameter.

```

<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

- ->with () digunakan untuk menambahkan lebih dari satu variabel.
- Saat URL /greeting diakses, method greeting() dari WelcomeController akan dipanggil. Kemudian View blog.hello akan menerima variabel name dan occupation untuk ditampilkan.

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
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

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.

JAWABAN

1. Jalankan composer create-project Laravel/Laravel="10.3" POS pada terminal

```
C:\laragon\www>composer create-project laravel/laravel="10.3.*" POS
Creating a "laravel/laravel=10.3.*" project at "./POS"
Installing laravel/laravel (v10.3.3)
 - Installing laravel/laravel (v10.3.3): Extracting archive
Created project in C:\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)
 - 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 (2.17.0)
 - Locking fruitcake/php-cors (v1.3.0)
```

2. Pindah ke folder POS dan jalankan php artisan serve

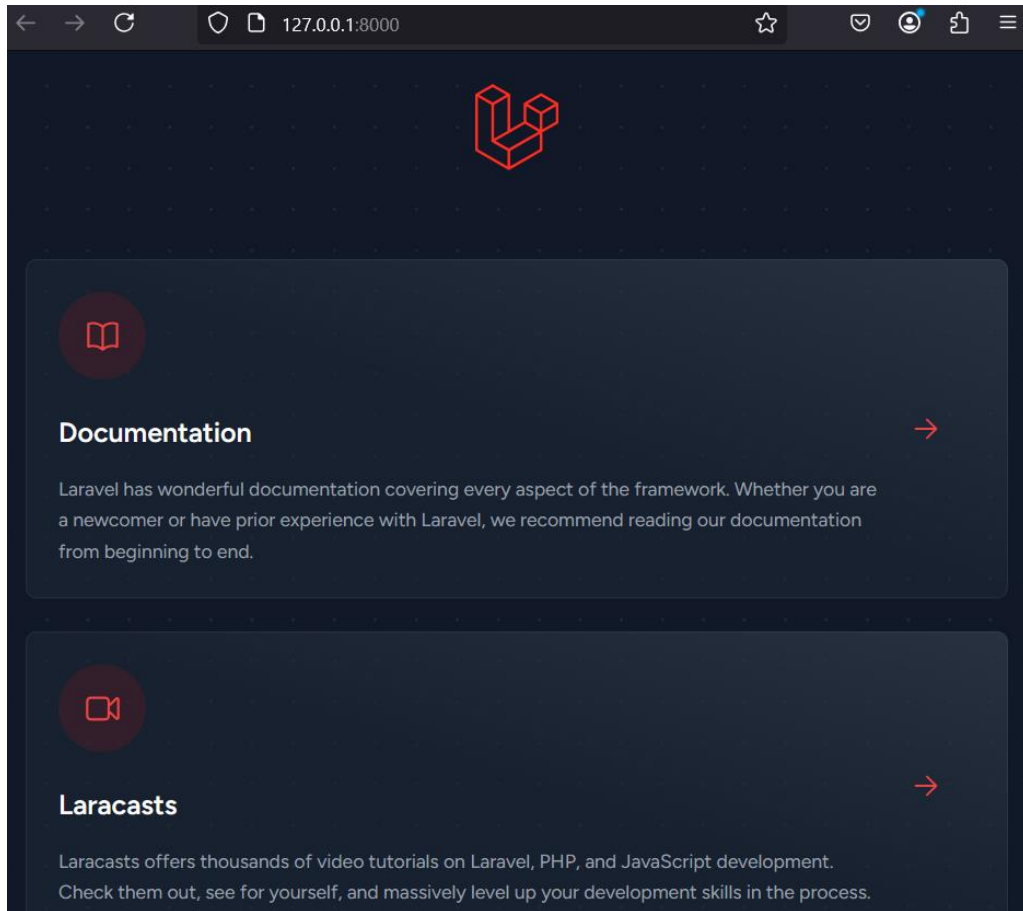
```
C:\laragon\www>cd POS
C:\laragon\www\POS>php artisan serve

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

Press Ctrl+C to stop the server

2025-02-24 08:24:04 ..... ~ 5s
```


Hasil:



3. Membuat 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 ProductController
```

```
INFO Controller [C:\laragon\www\POS\app\Http\Controllers\ProductController.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.
```

```
PS C:\laragon\www\POS> █
```

4. Membuat route

routes > web.php > Closure

```
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\HomeController;
5  use App\Http\Controllers\ProductController;
6  use App\Http\Controllers\SalesController;
7  use App\Http\Controllers\UserController;
8
9  //Home
10 Route::get('/', [HomeController::class, 'index']);
11
12 //Product dengan Prefix
13 Route::prefix('products')->group(function () {
14     Route::get('/food-beverages', [ProductController::class, 'foodBeverages'])->name('category.food-beverage');
15     Route::get('/beauty-health', [ProductController::class, 'beautyHealth'])->name('category.beauty-health');
16     Route::get('/home-care', [ProductController::class, 'homeCare'])->name('category.home-care');
17     Route::get('/baby-kid', [ProductController::class, 'babyKid'])->name('category.baby-kid');
18 });
19
20 //User dengan parameter
21 Route::get('/user/{id}/name/{name}', [UserController::class, 'showProfile'])->name('user.profile');
22
23 //Sales
24 Route::get('/sales', [SalesController::class, 'index'])->name('sales');
```

5. Mengisi controller

➤ HomeController.php

app > Http > Controllers > HomeController > 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 index(){
10         return view('home');
11     }
12 }
```

➤ ProductController.php

app > Http > Controllers > ProductController.php > ProductController >

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class ProductController extends Controller
8  {
9      public function foodBeverages(){
10         return view ('products.food-beverages');
11     }
12
13     public function beautyHealth(){
14         return view ('products.beauty-health');
15     }
16
17     public function homeCare(){
18         return view ('products.home-care');
19     }
20
21     public function babyKid(){
22         return view ('products.baby-kid');
23     }
24 }
```

➤ UserController.php

app > Http > Controllers > UserController.php > UserController > showProfile

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

➤ SalesController.php

```

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

```

6. Membuat View

➤ Home

Resource/views/home.blade.php

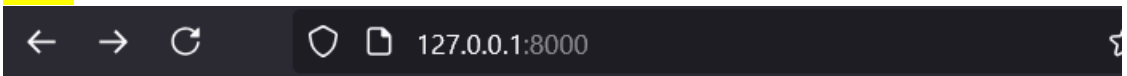
resources > views > home.blade.php > html

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Home</title>
5  </head>
6  <body>
7      <h1>Welcome! This is the home page of POS.</h1>
8      <h2>Product Categories</h2>
9      <ul>
10         <li><a href="{{ route('category.food-beverage') }}">Food & Beverages</a></li>
11         <li><a href="{{ route('category.beauty-health') }}">Beauty & Health</a></li>
12         <li><a href="{{ route('category.home-care') }}">Home Care</a></li>
13         <li><a href="{{ route('category.baby-kid') }}">Baby & Kids</a></li>
14     </ul>
15 </body>
16 </html>

```

Hasil:



Welcome! This is the home page of POS.

Product Categories

- [Food & Beverages](#)
- [Beauty & Health](#)
- [Home Care](#)
- [Baby & Kids](#)

➤ FNB

Resource/views/products/food-beverages.php

resources > views > products > food-beverages.php > html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>FNB</title>
5 </head>
6 <body>
7 |   <h1>Food and Beverage</h1>
8 </body>
9 </html>
```

Hasil:



Food and Beverage

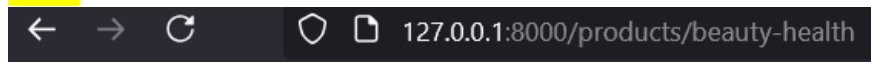
➤ Beauty and Health

Resource/views/products/beauty-health.php

resources > views > products > beauty-health.blade.php >

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Beauty and Health</title>
5 </head>
6 <body>
7 |   <h1>Beauty and Health Products</h1>
8 </body>
9 </html>
```

Hasil:



Beauty and Health Products

➤ Home Care

Resource/views/products/home-care.php

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>
```

Hasil:

← → ↻ 🛡️ 📄 127.0.0.1:8000/products/home-care

Home Care Products

➤ Baby kid

Resource/views/products/baby-kid.php

resources > views > products > baby-kid.blade.php >

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Baby Kid</title>
5 </head>
6 <body>
7 |   <h1>Baby Kid Products</h1>
8 </body>
9 </html>
```

Hasil:

← → ↻ 🛡️ 📄 127.0.0.1:8000/products/baby-kid

Baby Kid Products

➤ Sales

Resource/views/sales/index.blade.php

```
resources > views > sales > index.blade.php > f
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Sales</title>
5 </head>
6 <body>
7 | <h1>Point of Sales (POS)</h1>
8 </body>
9 </html>
```

Hasil:



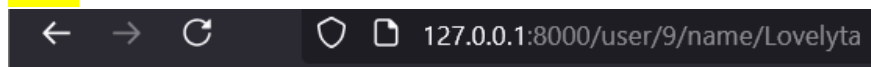
Point of Sales (POS)

➤ User

Resource/views/user/profile.blade.php

```
resources > views > user > profile.blade.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>User Profile</title>
5 </head>
6 <body>
7 | <h1>User Profile</h1>
8 | <p>ID: {{ $id }}</p>
9 | <p>Name: {{ $name }}</p>
10 </body>
11 </html>
```

Hasil:



User Profile

ID: 9

Name: Lovelyta

LINK GITHUB: <https://github.com/vveskr/POS>