

PEMROGRAMAN WEB LANJUT
JOBSHEET-2 - ROUTING CONTROLLER VIEW



Disusun oleh:
Stefanus Ageng Budi Utomo (2241720126)

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN AJARAN 2023/2024

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

```
16 Route::get('/', function () {  
17     return view('welcome');  
18 });  
19  
20 Route::get('/hello', function () {  
21     return "Hello";  
22 });  
23  
24 Route::get('/world', function () {  
25     return "World";  
26 });  
27
```

- 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.



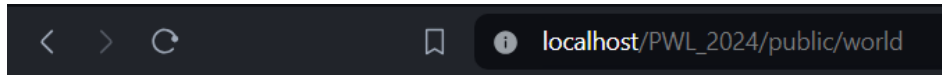
Hello

- pada halaman hello akan terdapat teks hello sesuai dengan yang kita return di route
- 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.



World

- - pada halaman world akan terdapat teks hello sesuai dengan yang kita return di route
- F. Selanjutnya, cobalah membuat route '/' yang menampilkan pesan 'Selamat Datang'.

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



Selamat Datang

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

```
Route::get('/about', function () {  
    return 'NIM 2241720126';  
});
```



NIM 2241720126

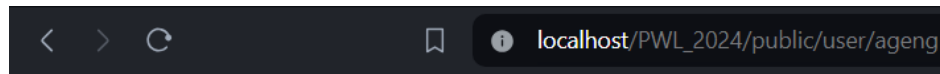
Route Parameters

- 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;  
});
```

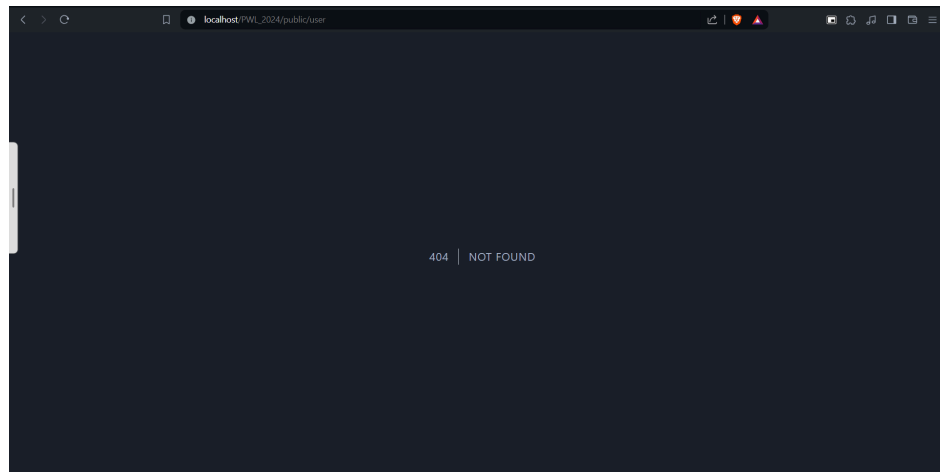
```
Route::get('/user/{name}', function ($name) {  
    return 'Nama saya ' . $name;  
});
```

- 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 ageng

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

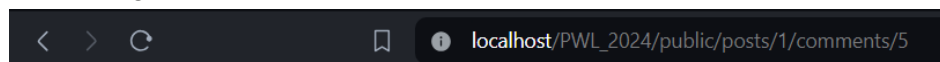


- D. Suatu route, juga bisa menerima lebih dari 1 parameter seperti kode berikut ini. Route menerima parameter \$postId dan juga \$comment.

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

```
Route::get('/posts/{posts}/comments/{comments}', 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.



Pos ke-1 Komentar ke-: 5

- 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 ($articlesId) {  
    return 'Halaman Artikel dengan ID-' . $articlesId;  
});
```



Halaman Artikel dengan ID-27

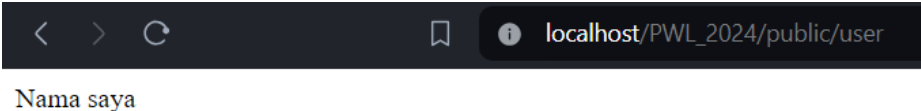
Optional Parameters

- 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;  
});
```

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

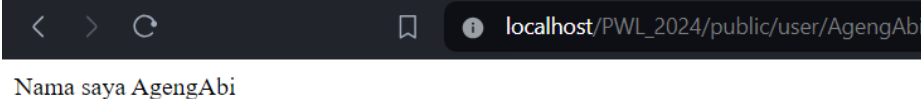
- B. Jalankan kode dengan menuliskan URL: localhost/PWL_2024/public/user/. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



localhost/PWL_2024/public/user

Nama saya

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



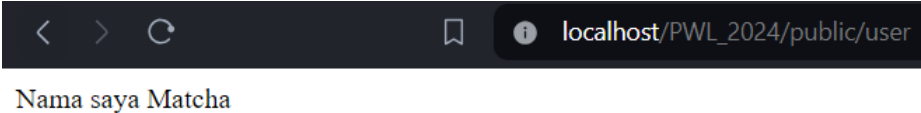
localhost/PWL_2024/public/user/AgengAbi

Nama saya AgengAbi

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

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

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



localhost/PWL_2024/public/user

Nama saya Matcha

Route Name

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



Nama saya profile

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.

Controller

Membuat controller

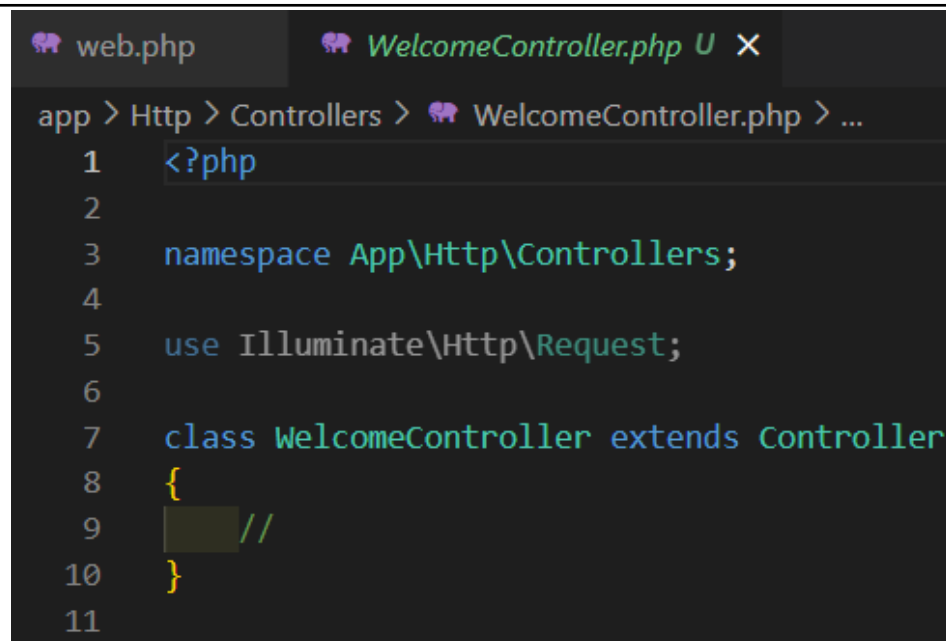
- A. Untuk membuat controller pada Laravel telah disediakan perintah untuk menggenerate struktur dasarnya. Kita dapat menggunakan perintah artisan diikuti dengan definisi nama controller yang akan dibuat.

```
php artisan make:controller WelcomeController
```

```
- PS D:\laragon\www\PWL_2024> 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
{
    //
}
```

A screenshot of a code editor with a dark theme. The top bar shows two tabs: 'web.php' and 'WelcomeController.php' (which is active and has a close button). The breadcrumb navigation shows the path: 'app > Http > Controllers > WelcomeController.php > ...'. The code content is as follows:

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class WelcomeController extends Controller
8  {
9      //
10 }
11
```

- 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';
    }
}
```

```
web.php M WelcomeController.php U X
app > Http > Controllers > WelcomeController.php > Welc
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class WelcomeController extends Controller
8  {
9      public function hello()
10     {
11         return 'Hello World';
12     }
13 }
14
```

- 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']);
```

```
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

- 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		
--	--	--	--	--

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

    public function about()
    {
        return 'Nama Ageng, NIM 2241720126';
    }

    public function articles($id)
    {
        return "Halaman Artikel dengan Id {$id}";
    }
}
```

< > ↻ ⓘ 127.0.0.1:8000

Selamat Datang

< > ↻ ⓘ 127.0.0.1:8000/about

Nama Ageng, NIM 2241720126

< > ↻ ⓘ 127.0.0.1:8000/articles/27

Halaman Artikel dengan Id 27

- 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.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    /**
     * Handle the incoming request.
     */
    public function __invoke(Request $request)
    {
        return 'Selamat Datang';
    }
}
```

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AboutController extends Controller
{
    /**
     * Handle the incoming request.
     */
    public function __invoke(Request $request)
    {
        return 'Nama Ageng, NIM 2241720126';
    }
}
```

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ArticleController extends Controller
{
    /**
     * Handle the incoming request.
     */
    public function __invoke($id, Request $request)
    {
        return "Halaman Artikel dengan Id {$id}";
    }
}
```

- web.php

```
use App\Http\Controllers\HomeController;
use App\Http\Controllers>AboutController;
use App\Http\Controllers\ArticleController;

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

< > ↺ ⓘ 127.0.0.1:8000

- Selamat Datang

< > ↺ ⓘ 127.0.0.1:8000/about

- Nama Ageng, NIM 2241720126

< > ↺ ⓘ 127.0.0.1:8000/articles/27

- Halaman Artikel dengan Id 27

Resource Controller

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

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

```
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;
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 auth: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

```
PS D:\laragon\www\PWL_2024> php artisan route:list
GET|HEAD / ..... HomeController
POST _ignition/execute-solution ..... 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 ..... AboutController
GET|HEAD api/user ..... 
GET|HEAD articles/{id} ..... ArticleController
GET|HEAD hello ..... WelcomeController@hello
ANY here ..... Illuminate\Routing > RedirectController
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/{posts}/comments/{comments} ..... 
GET|HEAD sanctum/csrf-cookie ..... sanctum.csrf-cookie > Laravel\Sanctum > CsrfCookieController@show
GET|HEAD user/profile ..... profile
GET|HEAD user/{name} ..... 
GET|HEAD world ..... 
```

- 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'
]);
```

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

Route::resource('photos', PhotoController::class)->only([
    'index', 'show'
]);
Route::resource('photos', PhotoController::class)->except([
    'create', 'store', 'update', 'destroy'
]);
```

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>
```



The screenshot shows a code editor with two tabs: 'web.php M' and 'hello.blade.php U'. The 'hello.blade.php' tab is active, showing the following code:

```
resources > views > hello.blade.php > ...
1  <html>
2
3  <body>
4  <h1>Hello, {{ $name }}</h1>
5  </body>
6
7  </html>
8
```

- 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']);
});
```

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

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



Hello, Ageng

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, Ageng

Meneruskan data ke view

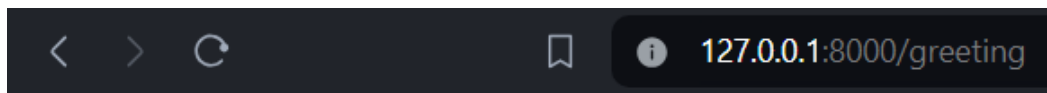
- A. Buka WelcomeController.php dan tambahkan ubah fungsi greeting.

```
class WelcomeController extends Controller
{
public function hello(){
return('Hello World');
}
public function greeting(){
return view('blog.hello')
Jobsheet 02 – PWL 2023/2024 Hal. 15 / 16
->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, Ageng

You are students

Tugas 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.

```
D:\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 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 (0.11.0)
- Locking carbonphp/carbon-doctrine-types (2.1.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.10)
- Locking doctrine/lexer (3.0.1)
- Locking dragonmantank/cron-expression (v3.3.3)
- Locking egulias/email-validator (4.0.2)
- Locking fakerphp/faker (v1.23.1)
- Locking filp/whoops (2.15.4)
- Locking fruitcake/php-cors (v1.3.0)
- Locking graham-campbell/result-type (v1.1.2)
- Locking guzzlehttp/guzzle (7.8.1)
- Locking guzzlehttp/promises (2.0.2)
- Locking guzzlehttp/psr7 (2.6.2)
- Locking guzzlehttp/uri-template (v1.0.3)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v10.46.0)
- Locking laravel/pint (v1.14.0)
```

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.


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

Route::prefix('category')->group(function () {
    Route::get('/', [ProductController::class, 'index']);
    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}', function ($id = 27, $name = "Ageng") {
    return view('user-profile', ['id' => $id, 'name' => $name]);
});

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

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

Halaman home

127.0.0.1:8000

Welcome to POS

This is the home page

	Title	Route
1	Home	/
2	Products	/category
3	User-profile	/user/{id}/name/{name} (default id=27 name=Ageng)
4	Penjualan	/penjualan

Halaman Product

127.0.0.1:8000/category

Product Category

Category
Food and Beverage
Beauty and Health
Home Care
Baby and Kid

127.0.0.1:8000/category/food-beverage

Food and Beverage Products

ID	Nama	Harga
1	Apel	Rp 10.000
2	Jeruk	Rp 8.000
3	Semangka	Rp 35.000

127.0.0.1:8000/category/beauty-health

Beauty and Health Products

ID	Nama	Harga
1	Shampo	Rp 60.000
2	Pembersih Wajah	Rp 120.000
3	Suplemen	Rp 150.000

127.0.0.1:8000/category/home-care

Home Care Products

ID	Nama	Harga
1	Sabun Piring	Rp 25.000
2	Deterjen	Rp 80.000
3	Pembersih Multi Fungsi	Rp 50.000

127.0.0.1:8000/category/baby-kid

Baby and Kid Products

ID	Nama	Harga
1	Pampers	Rp 100.000
2	Tisu Basah Bayi	Rp 55.000
3	Mainan Balok	Rp 150.000

Halaman User

127.0.0.1:8000/user/27/name/Ageng

User Profile

ID	Name
27	Ageng

Halaman Penjualan

127.0.0.1:8000/penjualan

Penjualan

ID	Product ID	User ID	Quantity
1	1	1	5
2	2	2	3
3	3	3	2
4	4	4	1
5	5	5	4

6. Simpan setiap perubahan yang dilakukan pada project POS pada Git, sinkronisasi perubahan ke Github.
 - Link Repo : <https://github.com/AgengAbi/POS.git>