LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT
ROUTING, CONTROLLER, DAN VIEW

Disusun Oleh :

Qusnul Diah Mawanti 2341760035

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2025

Jurusan Teknologi Informasi – Politeknik Negeri Malang

Jobsheet 2 : Routing, Controller, Dan View
Mata Kuliah : Pemrograman Web Lanjut
Kelas : 2A – SIB
Nama : Qusnul Diah Mawanti
No absen : 25
NIM : 2341760035

Februari 2025

TUJUAN PRAKTIKUM

1. Memahami konsep dasar MVC pada Laravel.
2. Mempelajari implementasi Routing dalam Laravel.
3. Mempelajari penggunaan Controller sebagai perantara Model dan View.
4. Menggunakan View untuk menampilkan data kepada pengguna.

DASAR TEORI

A. MVC Pada Laravel

Laravel menggunakan arsitektur **Model-View-Controller (MVC)** yang terdiri dari:

- **Model:** Mengelola data dan interaksi dengan database.
- **View:** Bertanggung jawab atas tampilan antarmuka pengguna.
- **Controller:** Mengatur logika aplikasi dan menghubungkan Model dengan View.

B. Routing

Routing dalam Laravel digunakan untuk menghubungkan URL dengan aksi tertentu.

1. Praktikum 1: Basic Routing

a. Membuat dua buah route menggunakan project **PWL2025** dengan ketentuan sebagai berikut.

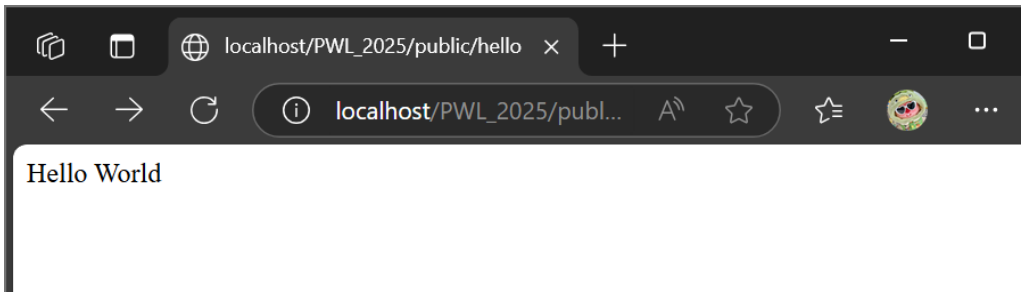
No	Http Verb	Url	Fungsi
1	get	/hello	Tampilkan String Hello ke browser.
2	get	/world	Tampilkan String World ke browser

b. Buka file **`routes/web.php`**. Tambahkan sebuah route dengan URL **/hello** yang akan menampilkan teks "Hello" pada browser.

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

```php
// Route untuk menampilkan "Hello"
Route::get('/hello', function () {
    return 'Hello World';
});
```

c. Buka browser, tuliskan URL untuk memanggil route tersebut: **localhost/PWL_2025/public/hello**. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.



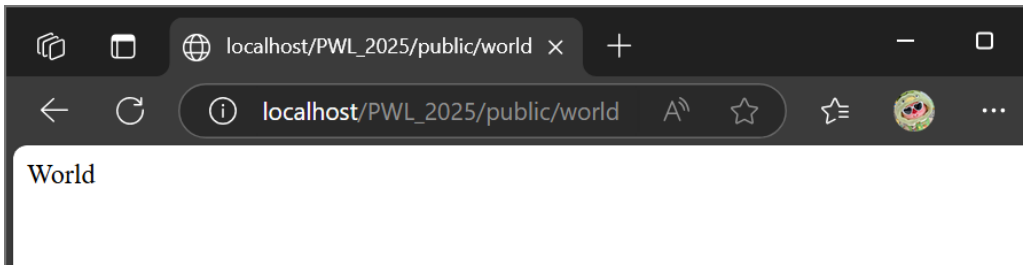
Halaman berhasil muncul dengan teks **Hello World**, berarti route sudah berjalan dengan baik.

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

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

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

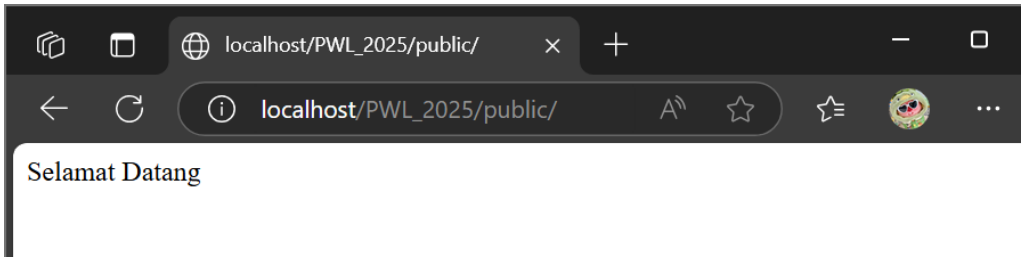
e. Bukalah pada browser, tuliskan URL untuk memanggil route tersebut: **localhost/PWL_2025/public/world**. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.



Halaman berhasil muncul dengan teks **World**, berarti route sudah berjalan dengan baik.

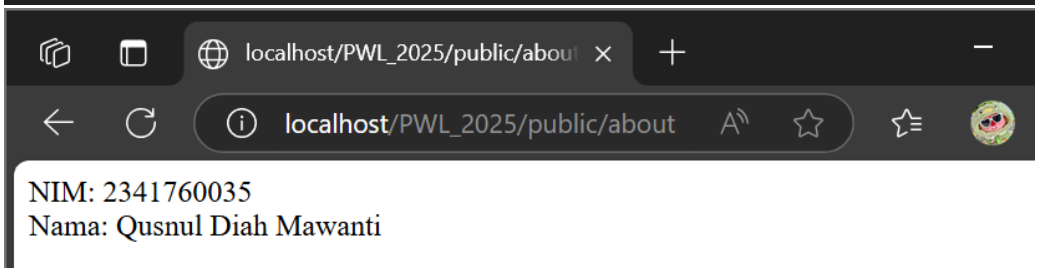
f. Selanjutnya , cobalah membuat route **'/'** yang menampilkan pesan **'Selamat Datang'**.

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



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

```
// Route '/about' untuk menampilkan 'NIM' dan 'Nama'
Route::get('/about', function () {
    return 'NIM: 2341760035 <br> Nama: Qusnul Diah Mawanti';
});
```

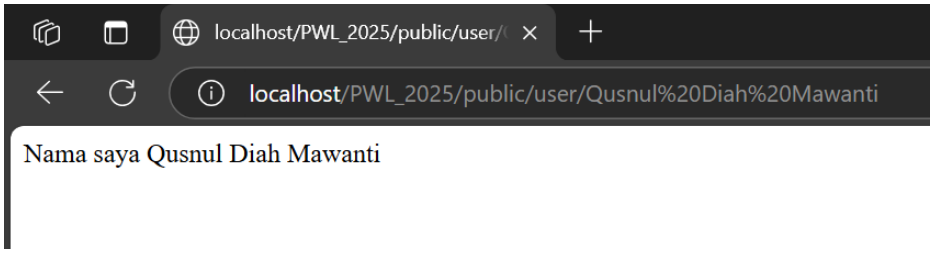


2. Praktikum 2: Route Parameter

a. Memanggil route ***/user/{name}*** sekaligus mengirimkan parameter berupa nama user **$name** seperti kode di bawah ini.

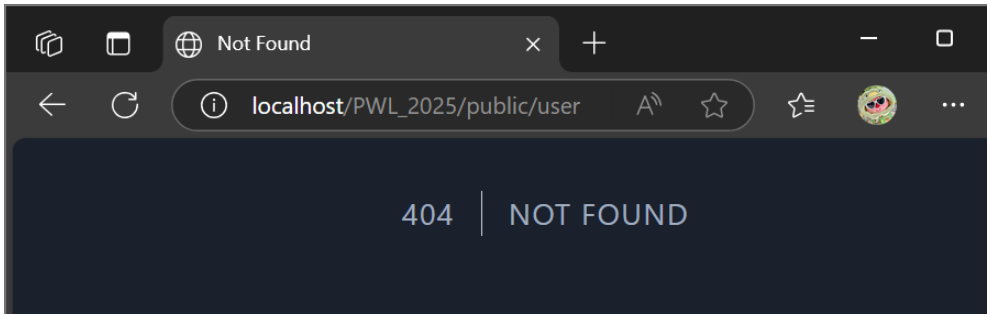
```
// Route dengan satu parameter
Route::get('/user/{name}', function ($name) {
    return "Nama saya " . $name;
});
```

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



Hasil pengamatan adalah akan muncul teks "Nama saya Qusnul Diah Mawanti", dengan parameter **{name}** diganti sesuai input di URL.

c. Selanjutnya, coba tuliskan URL: **localhost/PWL_2025/public/user/**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

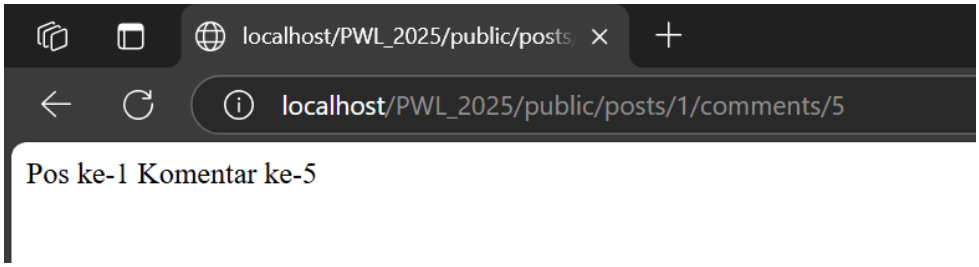


Hasil pengamatan jika mengakses **localhost/PWL_2025/public/user/**, akan muncul **404 Not Found**, karena Laravel mengharapkan adanya parameter **{name}.**

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

```
// Route dengan dua parameter {post} dan {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_2025/public/posts/1/comments/5**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

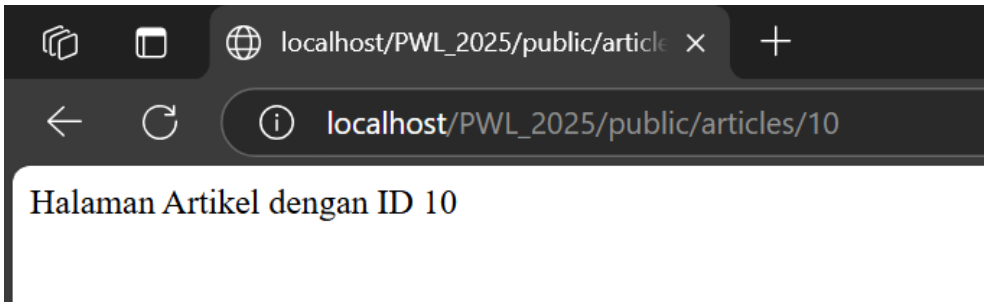


Hasil pengamatan adalah akan muncul teks **"Pos ke-1 Komentar ke-5"** sesuai dua parameter yang ada.

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

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

Akan muncul teks **"Halaman Artikel dengan ID 10"**, dengan angka **10** sesuai input URL.

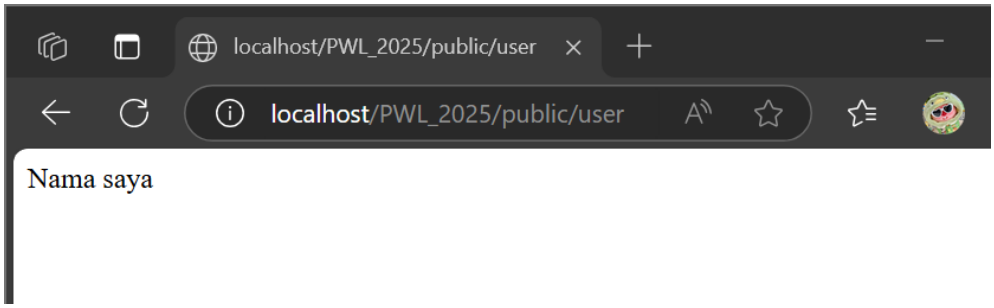


3. Praktikum 3: Optional Parameters

a. Memanggil route **/user** sekaligus mengirimkan parameter berupa nama user **$name** dimana parameternya bersifat opsional.

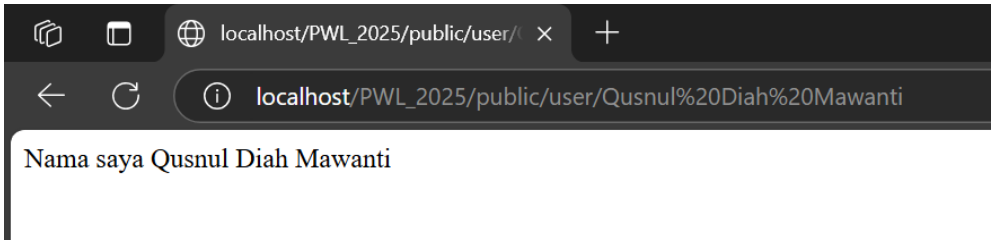
```php
// Route dengan parameter optional
Route::get('/user/{name?}', function ($name = null) {
    return "Nama saya " . $name;
});
```

b. Jalankan kode dengan menuliskan URL: **localhost/PWL_2025/public/user/**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Jika parameter tidak diberikan, maka **$name** akan menjadi **null**, sehingga hasilnya bisa menjadi "Nama saya " tanpa nilai setelahnya.

c. Selanjutnya tuliskan URL**: localhost/PWL_2025/public/user/NamaAnda**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.

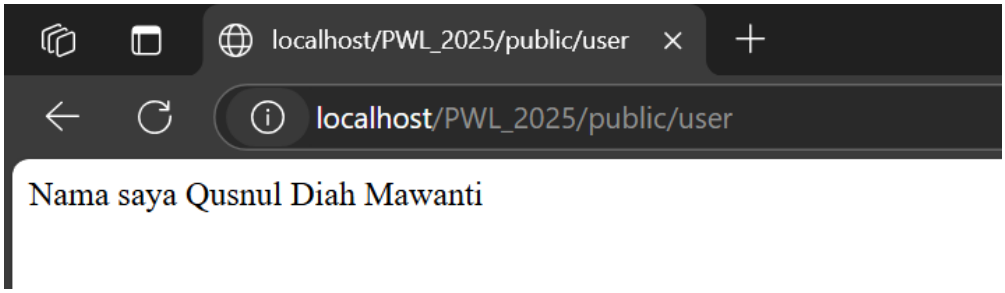


Halaman akan berubah sesuai input dari URL.

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

```php
// Route dengan parameter default
Route::get('/user/{name?}', function ($name = 'Qusnul Diah Mawanti') {
    return "Nama saya " . $name;
});
```

e. Jalankan kode dengan menuliskan URL: **localhost/PWL_2025/public/user/**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Jika tidak ada parameter, Laravel menggunakan nilai **default** ("Qusnul Diah Mawanti"). Jika ada parameter, Laravel menampilkan sesuai input dari URL.

C. Controller

Controller digunakan untuk mengorganisasi logika aplikasi menjadi lebih terstruktur. Logika action aplikasi yang masih ada kaitan dapat dikumpulkan dalam satu kelas Controller. Atau sebuah Controller dapat juga hanya berisi satu buah action. Controller pada Laravel disimpan dalam folder `app/Http/Controllers`.

1. Praktikum 4: 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.

```
C:\laragon\www\PWL_2025>php artisan make:controller WelcomeContr
oller

  INFO  Controller [C:\laragon\www\PWL_2025\app\Http\Controller
s\WelcomeController.php] created successfully.
```

b. Buka file pada **`app/Http/Controllers/WelcomeController.php`**. Struktur pada controller dapat digambarkan sebagai berikut:

app > Http > Controllers > WelcomeController.php > ...

```php
<?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:

app > Http > Controllers > WelcomeController.php > ...

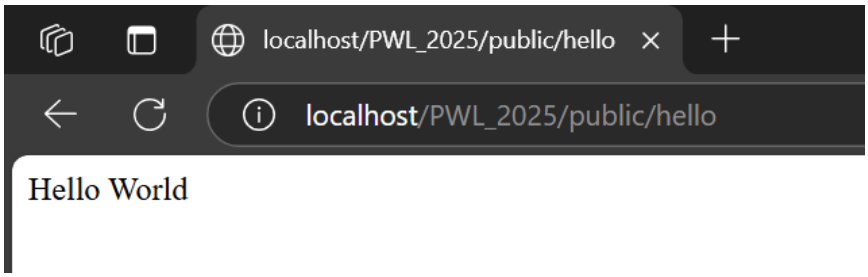
```php
<?php

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:

```
use App\Http\Controllers\WelcomeController;
```

```
// route /hello memanggil method hello() dari WelcomeController
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		

Membuat **PageController**.

```
C:\laragon\www\PWL_2025>php artisan make:controller PageControll
er

  INFO  Controller [C:\laragon\www\PWL_2025\app\Http\Controller
s\PageController.php] created successfully.
```

Menambahkan method pada **app/Http/Controller/PageController.php**.

app > Http > Controllers > PageController.php > ...

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

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

    public function about()
    {
        return "NIM: 2341760035 <br> Nama: Qusnul Diah M";
    }

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

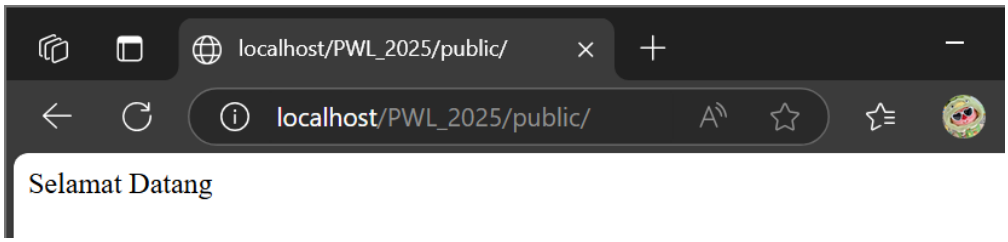
Memanggil **PageController.**

```php
use App\Http\Controllers\PageController;
```

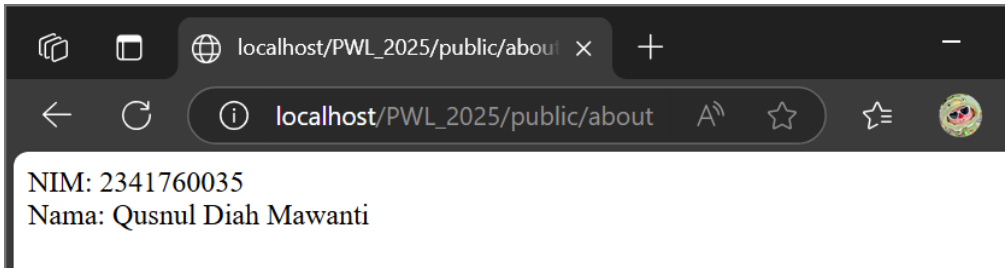
```php
// Route memanggil PageController
Route::get('/', [PageController::class, 'index']);
Route::get('/about', [PageController::class, 'about']);
Route::get('/articles/{id}', [PageController::class, 'articles']);
```

Uji coba Akses.

- **/**



- **/about**



➢ **/articles/{id}**

g. Modifikasi kembali implementasi sebelumnya dengan konsep Single Action Controller. Sehingga untuk hasil akhir yang didapatkan akan ada **`HomeController`, `AboutController` dan `ArticleController`**. Modifikasi juga route yang digunakan.

➢ Membuat Controller Baru dengan method **--invoke()**, yang hanya menangani satu action.

a. **HomeController**

```
C:\laragon\www\PWL2025>php artisan make:controller HomeControlle
r --invokable

  INFO  Controller [C:\laragon\www\PWL2025\app\Http\Controllers
\HomeController.php] created successfully.
```

b. **AboutController**

```
C:\laragon\www\PWL2025>php artisan make:controller AboutControll
er --invokable

  INFO  Controller [C:\laragon\www\PWL2025\app\Http\Controllers
\AboutController.php] created successfully.
```

c. **ArticleController**

```
C:\laragon\www\PWL2025>php artisan make:controller ArticleContro
ller --invokable

  INFO  Controller [C:\laragon\www\PWL2025\app\Http\Controllers
\ArticleController.php] created successfully.
```

➢ Menambahkan method pada setiap controller

d. **app/Http/Controllers/HomeController.php**

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

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

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

e. **app/Http/Controllers/AboutController.php**

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

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AboutController extends Controller
{
    public function __invoke()
    {
        return "NIM: 2341760035 <br> Nama: Qusnul Diah Mawanti";
    }
}
```

f. **app/Http/Controllers/ArticleController.php**

app > Http > Controllers > ArticleController.php > ...

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

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

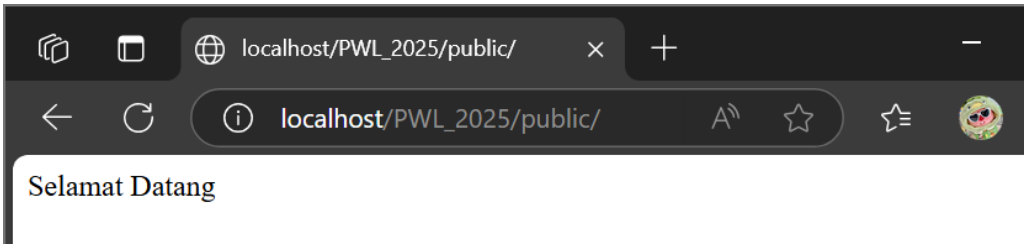
➢ Ubah Route agar sesuai dengan Single Action Controller

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

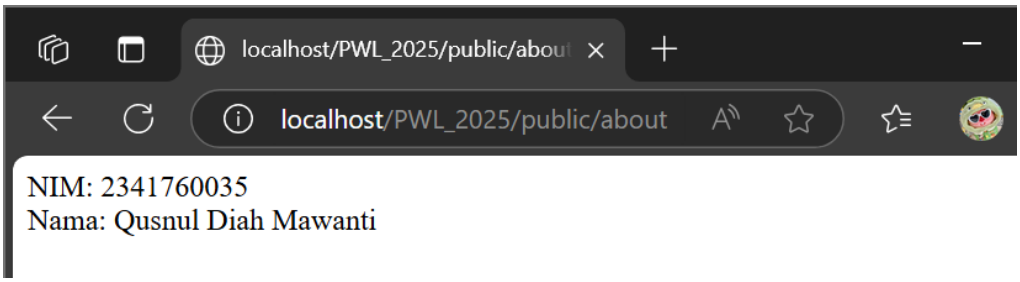
```php
// Route Single Action Controller
Route::get('/', HomeController::class);
Route::get('/about', AboutController::class);
Route::get('/articles/{id}', ArticleController::class);
```

➢ Uji Coba

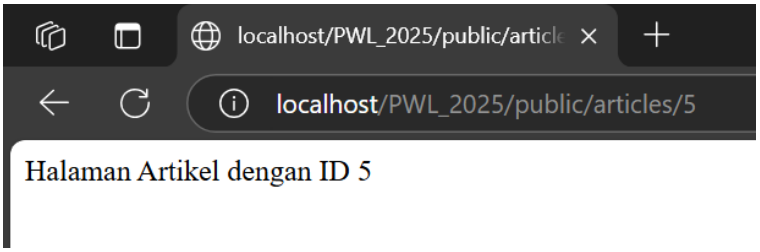
g. **localhost/PWL_2025/public/**



h. **localhost/PWL_2025/public/about**



i. **localhost/PWL_2025/public/articles/5**



2. Praktikum 5: Resource Controller

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

```
C:\laragon\www\PWL2025>php artisan make:controller PhotoControll
er --resource

  INFO  Controller [C:\laragon\www\PWL2025\app\Http\Controllers
\PhotoController.php] created successfully.
```

Perintah ini akan men 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 untuk PhotoController
Route::resource('photos', PhotoController::class);
```

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

```
C:\laragon\www\PWL2025>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
  GET|HEAD        photos ..... photos.index › PhotoController@index
  POST            photos ..... photos.store › PhotoController@store
  GET|HEAD        photos/create ..... photos.create › PhotoController@create
  GET|HEAD        photos/{photo} ..... photos.show › PhotoController@show
  PUT|PATCH       photos/{photo} ..... photos.update › PhotoController@update
  DELETE          photos/{photo} ..... photos.destroy › PhotoController@destroy
  GET|HEAD        photos/{photo}/edit ..... photos.edit › PhotoController@edit
  GET|HEAD        posts/{post}/comments/{comment} .....
  GET|HEAD        sanctum/csrf-cookie ..... sanctum.csrf-cookie › Laravel\Sanctum › CsrfCookieController@show
  GET|HEAD        user/{name?} .....
  GET|HEAD        user/{name} .....
  GET|HEAD        world .....

                                                                                Showing [20] 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.

```php
// Hanya menggunakan index dan show
Route::resource('photos', PhotoController::class)->only(['index', 'show']);

// Mengecualikan Method Tertentu
Route::resource('photos', PhotoController::class)->except(['create', 'store', 'update', 'destroy']);
```

D. View

Dalam kerangka kerja Laravel, View merujuk pada bagian dari aplikasi web yang bertanggung jawab untuk menampilkan antarmuka pengguna kepada pengguna akhir. View pada dasarnya adalah file template yang digunakan untuk menghasilkan HTML yang akan ditampilkan kepada pengguna.

1. Praktikum 6: Membuat View

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

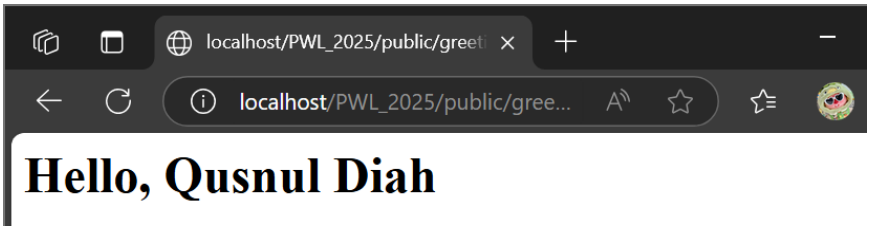
resources > views > hello.blade.php > ...

```blade
<!-- 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)

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

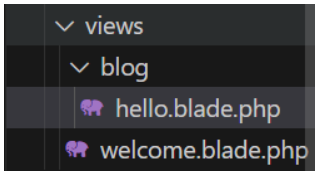
c. Jalankan code dengan membuka url **localhost/PWL_2025/public/greeting**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



2. Praktikum 7: View Dalam Direktori

a. Buatlah direktori **blog** di dalam direktori **views**.

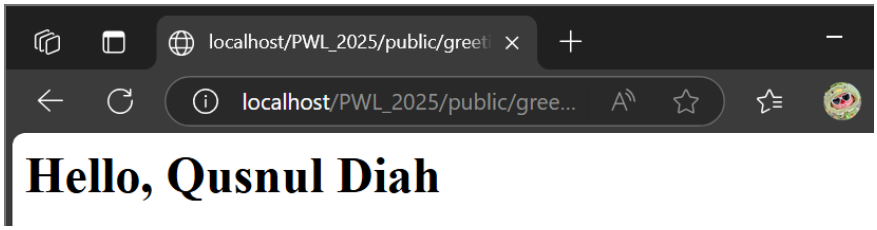
b. Pindahkan file **hello.blade.php** ke dalam direktori blog.



c. Selanjutnya lakukan perubahan pada **route**.

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

d. Jalankan code dengan membuka url **localhost/PWL_2025/public/greeting**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Laravel bisa menggunakan View dalam subfolder

3. Praktikum 8: Menampilkan View dari Controller

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

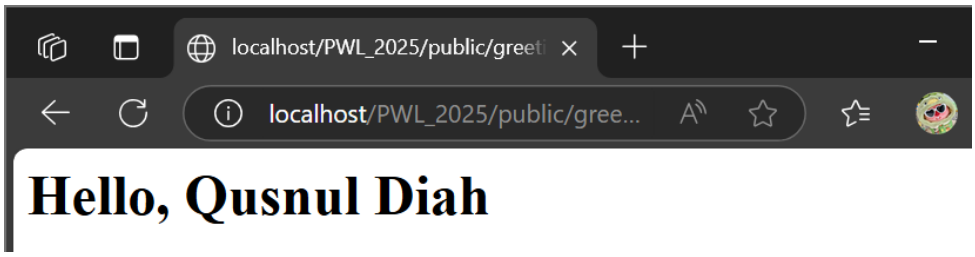
app > Http > Controllers > WelcomeController.php > ...

```php
<?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' => 'Qusnul Diah']);
    }
}
```

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

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

c. Jalankan code dengan membuka url **localhost/PWL_2025/public/greeting**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Laravel akan memanggil method **greeting()** dalam **WelcomeController**

4. Praktikum 9: Meneruskan Data Ke View

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

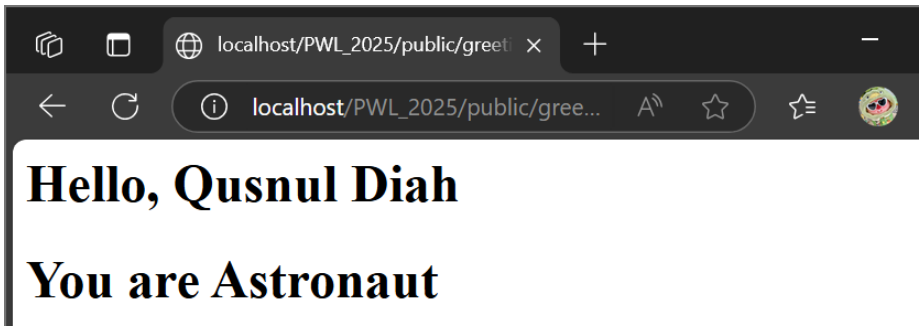
```php
public function greeting()
{
    return view('blog.hello')
        ->with('name', 'Qusnul Diah')
        ->with('occupation', 'Astronaut');
}
```

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

resources > views > blog > hello.blade.php > ...

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

c. Jalankan code dengan membuka url **localhost/PWL_2025/public/greeting**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



Modifikasi method **greeting()** di **WelcomeController** untuk mengirim lebih dari satu parameter ke View

E. 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.

```
C:\laragon\www>composer create-project "laravel/laravel:^10.0" POS
Creating a "laravel/laravel:^10.0" 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 (0.12.3)
  - Locking carbonphp/carbon-doctrine-types (2.1.0)
```

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

a. **Membuat Controller**

- **HomeController.php**

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

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

- **ProduckController.php**

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


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

- **UserController.php**

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

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

➢ **SalesController.php**

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

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

b. **Menambahkan Route di routes/web.php**

routes > web.php

```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\HomeController;
use App\Http\Controllers\ProductController;
use App\Http\Controllers\UserController;
use App\Http\Controllers\SalesController;

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

// Halaman Products dengan prefix
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']);
});

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

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

c. **Menambahkan Method Pada Controller**

➢ **HomeController.php**

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

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

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

➢ **ProduckController.php**

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

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

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

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

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

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

➢ **UserController.php**

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

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

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

➢ **SalesController.php**

app > Http > Controllers > SalesController.php > ...

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

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

d. **Membuat File View**

➢ **View Home**

resources > views > home.blade.php > ...

```
<html>
<body>
    <h1>Selamat Datang di Aplikasi POS</h1>
</body>
</html>
```

➢ **View Product**

- **food-beverage.blade.php**

resources > views > product > food-beverage.blade.php > ...

```
<html>
<body>
    <h1>Daftar Produk - Food & Beverage</h1>
</body>
</html>
```

- **beauty-health.blade.php**

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

```
<html>
<body>
    <h1>Daftar Produk - Beauty & Health</h1>
</body>
</html>
```

- **home-care.blade.php**

resources > views > product > home-care.blade.php > ...

```
<html>
<body>
    <h1>Daftar Produk - Home Care</h1>
</body>
</html>
```

- **baby-kid.blade.php**

resources > views > product > baby.kid.blade.php > ...

```
<html>
<body>
    <h1>Daftar Produk - Baby & Kid</h1>
</body>
</html>
```

➢ **View User Profile**

resources > views > user > profile.blade.php > ...

```
<html>
<body>
    <h1>Profil Pengguna</h1>
    <p>ID: {{ $id }}</p>
    <p>Nama: {{ $name }}</p>
</body>
</html>
```

➢ **View Sales**

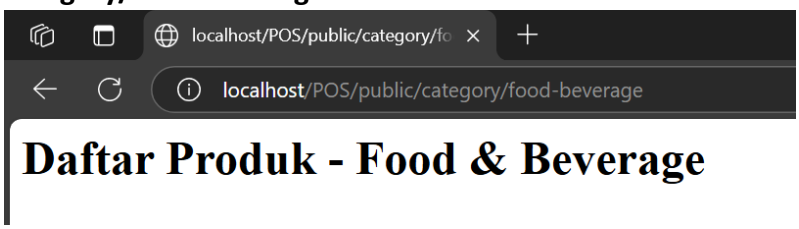
```
resources > views > sales > index.blade.php > ...
1   <html>
2   <body>
3       <h1>Halaman Transaksi POS</h1>
4   </body>
5   </html>
```

e. **Uji Coba**

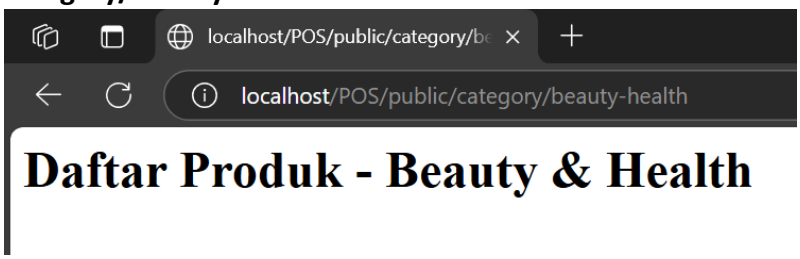
➢ **Home**

➢ **Product**

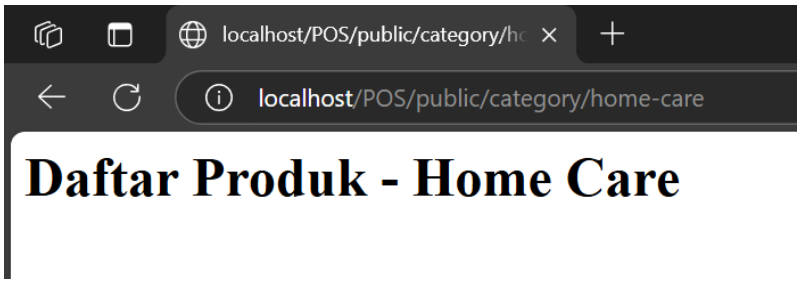
- **category/food-beverage**



- **category/beauty-health**

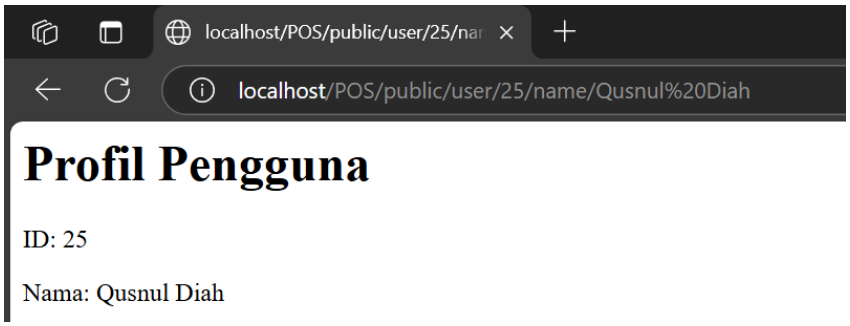


- **category/home-care**



- **Category/baby-kid**

- **User**



- **Sales**

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