

Nama : Dea Putri Nastiti

NIM : 2241720117

Kelas : TI 2H

Pemrograman Web Lanjut

Jobsheet 2

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

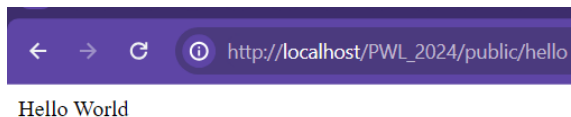
Kita akan menggunakan project minggu sebelumnya yaitu PWL_2024.

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

```
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.

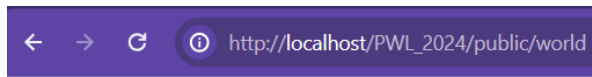


- d. Untuk membuat route kedua, tambahkan route `/world` seperti di bawah ini:

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

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

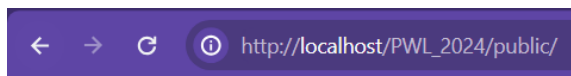
- e. Bukalah pada browser, tuliskan URL untuk memanggil route tersebut: localhost/PWL_2024/public/world. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.



World

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

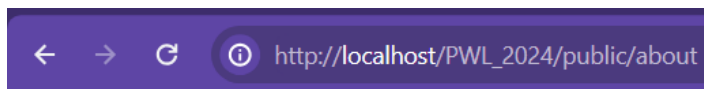
```
Route::get('/', function(){
    return 'Selamat Datang';
});
```



Selamat Datang

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

```
32 Route::get('/about', function(){
33     return '2241720117 <br>Dea Putri Nastiti';
34 });
```



2241720117
Dea Putri Nastiti

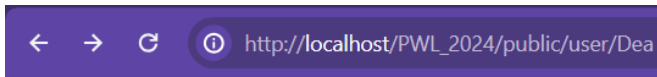
- Route Parameters

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.

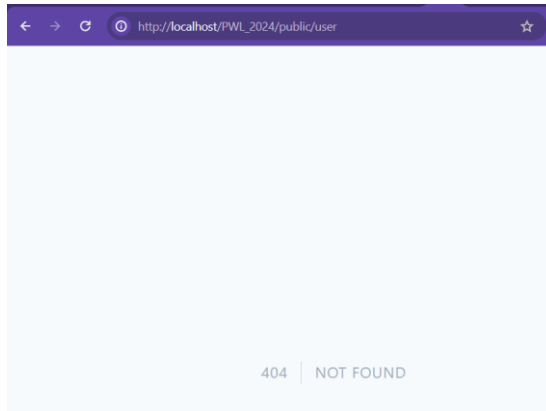
```
Route::get( 'user/ {name}', function
($name) {
    return 'Nama saya
'.$name; });
```

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



Nama saya Dea

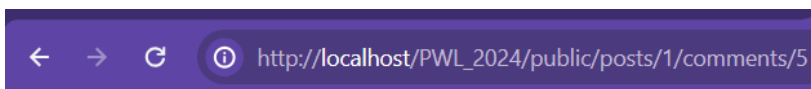
- c. Selanjutnya, coba tuliskan URL: **localhost/PWL_2024/public/user/**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda



- 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; });
```

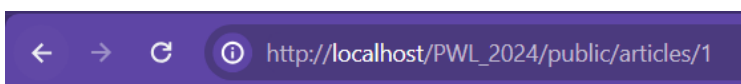
- e. Jalankan kode dengan menuliskan URL untuk memanggil route tersebut: **localhost/PWL_2024/public/posts/1/comments/5**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



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



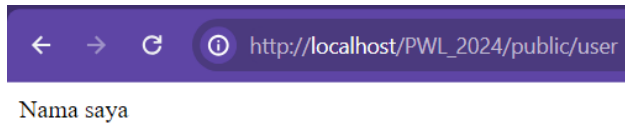
Halaman Artikel dengan Id 1

- Optional Parameters

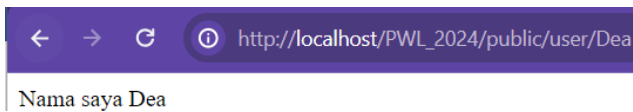
- a. Kita akan memanggil route `/user` sekaligus mengirimkan parameter berupa nama user `$name` dimana parameternya bersifat opsional.

```
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.



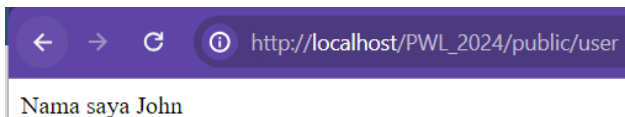
- c. Selanjutnya tuliskan URL: **localhost/PWL_2024/public/user>NamaAnda**. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda



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

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

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



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.

```
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;

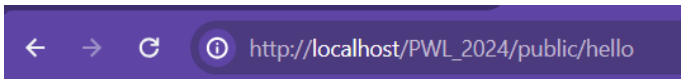
use Illuminate\Http\Request;

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.



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		

```

PhotoController.php × PageController.php × ArticleController.php
app > Http > Controllers > PageController.php > PageController > artic
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class PageController extends Controller
8  {public function index(){
9      return 'Selamat Datang';
10 }
11
12 public function about(){
13     return 'Dea Putri Nastiti <br>2241720117';
14 }
15
16 public function articles($artichel){
17     return 'Halaman Artikel dengan Id '.$artichel;
18 }
19     //
20 }

```

```

39  Route::get('/', [PageController::class,'index']);
40

```

← → ↺ ⓘ http://localhost/PWL_2024/public/
Selamat Datang

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

← → ↻ ⓘ http://localhost/PWL_2024/public/about

Dea Putri Nastiti
2241720117

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

← → ↻ ⓘ http://localhost/PWL_2024/public/articles/17

Halaman Artikel dengan Id 17

- 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.

```
AppController.php AboutController.php HomeController.php ×
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 {public function index(){
9     return 'Selamat Datang';
10 }
11 //
12 }
13
```

```
41 Route::get('/', [HomeController::class, 'index']);
42
```

← → ↻ ⓘ http://localhost/PWL_2024/public/

Selamat Datang

```

oller.php | AboutController.php X | HomeController.php |
pp > 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  {public function about(){
9      return 'Dea Putri Nastiti <br>2241720117';
10 }
11     //
12 }

```

```

48 Route::get('/about', [AboutController::class, 'about']);

```

← → ↻ ⓘ http://localhost/PWL_2024/public/about

Dea Putri Nastiti
2241720117

```

PageController.php | ArticleController.php X | WelcomeContro
pp > 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  {public function articles($artichel){
9      return 'Halaman Artikel dengan Id '.$artichel;
10 }
11     //
12 }

```

```

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

```

← → ↻ ⓘ http://localhost/PWL_2024/public/articles/17

Halaman Artikel dengan Id 17

- Resource Controller

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

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

Perintah ini akan generate sebuah controller dengan nama PhotoController yang berisi method method standar untuk proses CRUD.

```
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.
```

```
web.php PhotoController.php X hello.blade.php  
app > Http > Controllers > PhotoController.php > ...  
1 <?php  
2  
3 namespace App\Http\Controllers;  
4  
5 use Illuminate\Http\Request;  
6  
7 class PhotoController extends Controller  
8 {  
9     /**  
10      * Display a listing of the resource.  
11      */  
12     public function index()  
13     {  
14         //  
15     }  
16 }
```

- 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);
```

```
71 use App\Http\Controllers\PhotoController;  
72  
73 Route::resource('photos', PhotoController::class);  
74
```

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

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

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

GET|HEAD      / ..... HomeController@index
POST          _ignition/execute-solution ignition.execut...
GET|HEAD      _ignition/health-check ignition.healthChec...
POST          _ignition/update-config ignition.updateCon...
GET|HEAD      about ..... AboutController@about
GET|HEAD      api/user .....
GET|HEAD      articles/{id} .. ArticleController@articles
GET|HEAD      greeting ..... WelcomeController@greeting
GET|HEAD      hello ..... WelcomeController@hello
GET|HEAD      photos photos.index > PhotoController@index
POST          photos photos.store > PhotoController@store
GET|HEAD      photos/create photos.create > PhotoControl...
GET|HEAD      photos/{photo} photos.show > PhotoControll...
PUT|PATCH    photos/{photo} photos.update > PhotoContro...
DELETE        photos/{photo} photos.destroy > PhotoContr...
GET|HEAD      photos/{photo}/edit photos.edit > PhotoCon...
GET|HEAD      posts/{post}/comments/{comment} .....
GET|HEAD      sanctum/csrf-cookie sanctum.csrf-cookie > ...
GET|HEAD      user/{name?} .....
GET|HEAD      user/{name} .....
GET|HEAD      world .....

Showing [21] routes
```

- d. Pada route list semua route yang berhubungan untuk crud photo sudah di generate oleh laravel. Jika tidak semua route pada resource controller dibutuhkan dapat dikurangi dengan mengupdate route pada web.php menjadi seperti berikut ini.

```
Route::resource('photos', PhotoController::class)->only([
    'index', 'show'
]);

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

```
75 Route::resource('photos', PhotoController::class)->only([
76     'index', 'show'
77 ]);
78
79 Route::resource('photos', PhotoController::class)->except([
80     'create', 'store', 'update', 'destroy' ]);
```

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>
```

```
resources > views > blog > hello.blade.php > ...
1 <!-- View pada resources/views/hello.blade.php -->
2 <html>
3   <body>
4     <h1>Hello, {{ $name }}</h1>
5     <h1>You are, {{ $occupation }}</h1>
6   </body>
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']);
});
```

```
82 Route::get('/greeting', function () {
83     return view('hello', ['name' => 'Dea']);
84 });
```

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

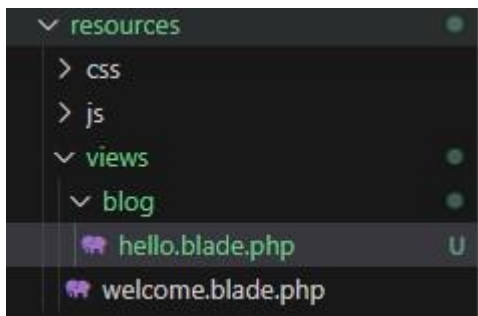
← → ↻ ⓘ http://localhost/PWL_2024/public/greeting

Hello, Dea

You are, Student

- View dalam direktori

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

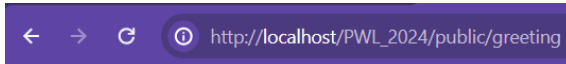


- c. Selanjutnya lakukan perubahan pada route.

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

```
86 Route::get('/greeting', function () {  
87     return view('blog.hello', ['name' => 'Dea']);  
88 });  
89
```

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



Hello, Dea

You are, Student

- 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']);  
    }  
}
```



```
1 <?php  
2  
3 namespace App\Http\Controllers;  
4  
5 use Illuminate\Http\Request;  
6  
7 class WelcomeController extends Controller  
8 { public function hello(){  
9     return 'Hello World';  
10 }  
11 public function greeting(){  
12     return view('blog.hello', ['name' => 'Dea']);  
13 }  
14 }  
15  
16 //  
17 }
```

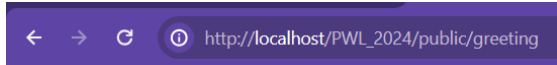
- b. Ubah route /greeting dan arahkan ke WelcomeController pada fungsi greeting.

```
Route::get('/greeting', [WelcomeController::class,
```

```
'greeting']]);
```

```
90 Route::get('/greeting', [WelcomeController::class,  
91 'greeting']);  
92
```

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



Hello, Dea

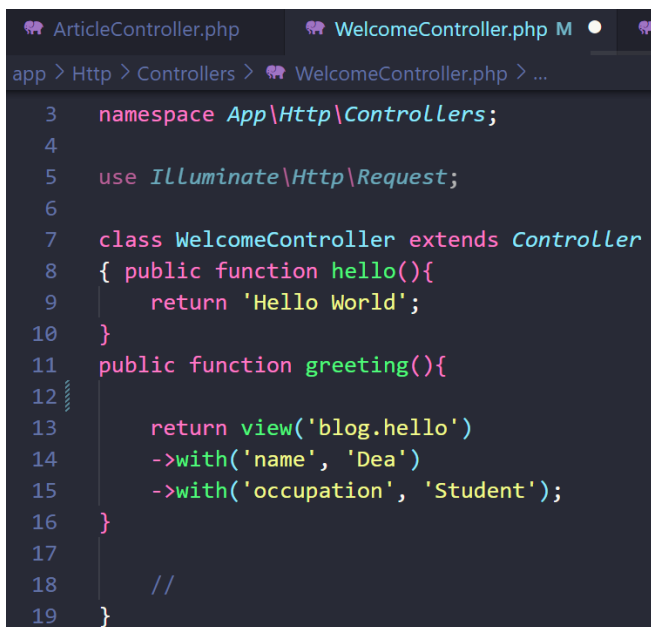
You are, Student

- **Meneruskan data ke View**

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

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

```
        ->with('name', 'Andi')  
        ->with('occupation', 'Astronaut');  
    }  
}
```

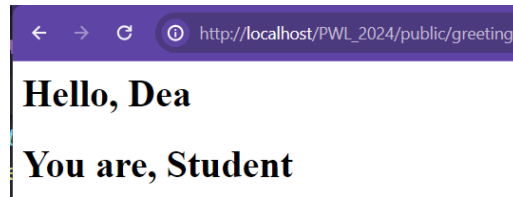


```
ArticleController.php WelcomeController.php M  
app > Http > Controllers > WelcomeController.php > ...  
3 namespace App\Http\Controllers;  
4  
5 use Illuminate\Http\Request;  
6  
7 class WelcomeController extends Controller  
8 { public function hello(){  
9     return 'Hello World';  
10 }  
11 public function greeting(){  
12  
13     return view('blog.hello')  
14     ->with('name', 'Dea')  
15     ->with('occupation', 'Student');  
16 }  
17  
18 //  
19 }
```

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

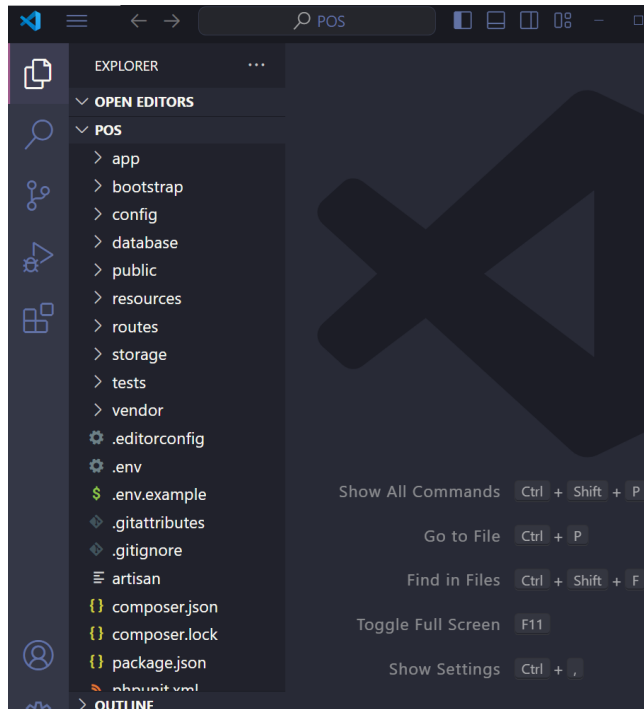
```
<html>
<body>
<h1>Hello, {{ $name }}</h1>
<h1>You are {{ $occupation }}</h1>
</body>
</html>
```

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



Soal Praktikum

1. Jalankan Langkah-langkah Praktikum pada jobsheet di atas. Lakukan sinkronisasi perubahan pada project PWL_2024 ke Github.
2. Buatlah project baru dengan nama POS. Project ini merupakan sebuah aplikasi Point of Sales yang digunakan untuk membantu penjualan.



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

1 Halaman Home

Menampilkan halaman awal web

```
welcome.blade.php x HomeController.php web.php
resources > views > welcome.blade.php > html > body > h2
1 <!DOCTYPE html>
2 <html Lang="{{str_replace('_', '-', app()->getLocale())}}">
3     <head>
4         <meta charset="utf-8">
5         <meta name="viewport" content="width=device-width, initial-scale=1">
6         <title>POS</title>
7     </head>
8     <body>
9         <h1>Selamat Datang</h1>
10        <h2>di Toko Serba ada {{ $name }}</h2>
11        <h3>Selamat Berbelanja</h3>
12    </body>
13 </html>
```

site

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

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

```

welcome.blade.php × HomeController.php × web.php
app > Http > Controllers > HomeController.php > HomeController > home
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class HomeController extends Controller
8  {
9      public function home(){
10         return view('welcome')
11         ->with('name', 'Dea');
12     }
13     //
14 }

```

```

21 Route::get('/home', [HomeController::class, 'home']);
22

```



2 Halaman Products Menampilkan daftar product (route prefix)

```

routes > web.php > ...
24 Route::prefix('/products')->group(function(){
25     Route::get('/category/foodBeverage', [ProductController::class, 'foodBeverage']);
26     Route::get('/category/beautyHealth', [ProductController::class, 'beautyHealth']);
27     Route::get('/category/homeCare', [ProductController::class, 'homeCare']);
28     Route::get('/category/babyKid', [ProductController::class, 'babyKid']);
29
30 });

```

```

resources > views > category.blade.php > ...
3  <!DOCTYPE html>
4  <html Lang="en">
5  <head>
6      <meta charset="UTF-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Product Category</title>
9  </head>
10 <body>
11     <h1>Products {{ $klasifikasi }}</h1>
12     <ul>
13         @foreach ($products as $product)
14             <li>{{ $product }}</li>
15         @endforeach
16     </ul>
17 </body>
18 </html>

```



```

app > Http > Controllers > ProductController.php > ProductController > foodBeverage
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class ProductController extends Controller
8  {
9      public function category()
10     {
11         $products = ['foodBeverage', 'beautyHealth', 'homeCare', 'babyKid'];
12         return view('category', ['klasifikasi' => 'Kategori Produk', 'products' => $products]);
13     }
14     public function foodBeverage()
15     {
16         $products = ['Ayam Geprek', 'Bakso', 'Tahu Telur'];
17         return view('category', ['klasifikasi' => 'Food Beverage', 'products' => $products]);
18     }
19     public function beautyHealth()
20     {
21         $products = ['Face Wash', 'Toner', 'Serum'];
22         return view('category', ['klasifikasi' => 'Beauty Health', 'products' => $products]);
23     }
24     public function homeCare()
25     {
26         $products = ['Sabun', 'Sapu', 'Handsantizer'];
27         return view('category', ['klasifikasi' => 'Home Care', 'products' => $products]);
28     }
29     public function babyKid()
30     {
31         $products = ['Minyak Telon', 'Susu', 'Baju Bayi'];
32         return view('category', ['klasifikasi' => 'Baby Kid', 'products' => $products]);
33     }
34 }

```

/category/food-beverage

← → ↻ http://localhost/POS/public/products/category/foodBeverage

Products Food Beverage

- Ayam Geprek
- Bakso
- Tahu Telur

/category/beauty-health

← → ↻ http://localhost/POS/public/products/category/beautyHealth

Products Beauty Health

- Face Wash
- Toner
- Serum

/category/home-care

← → ↻ http://localhost/POS/public/products/category/homeCare

Products Home Care

- Sabun
- Sapu
- Handsantizer

	<div><div>/category/baby-kid</div><div><div>← → ↺ http://localhost/POS/public/products/category/babyKid</div><div><h2>Products Baby Kid</h2><ul style="list-style-type: none">• Minyak Telon• Susu• Baju Bayi</div></div></div>
3	<div><div>Halaman User</div><div>Menampilkan profil pengguna (route param)</div><div>/user/{id}/nama/{nama}</div><div><pre>resources > views > user.blade.php > html > head 1 <!DOCTYPE html> 2 <html lang="{{str_replace('_', '-', app()->getLocale())}}"> 3 <head> 4 <meta charset="utf-8"> 5 <meta name="viewport" content="width=device-width, initial-scale=1"> 6 <title>POS</title> 7 </head> 8 <body> 9 <h1>Halo!!</h1> 10 <h2>User Profile</h2> 11 <p>ID : {{ \$user['id'] }}</p> 12 <p>Nama : {{ \$user['nama'] }}</p> 13 <h4>Selamat Berbelanja</h4> 14 </body> 15 </html></pre><div><pre>app > Http > Controllers > UserController.php > UserController > userProfile 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class UserController extends Controller 8 { 9 public function userProfile(\$id, \$nama){ 10 \$user = ['id' => \$id, 'nama' => \$nama]; 11 return view('user', ['user' => \$user]); 12 } 13 // 14 }</pre><pre>32 Route::get('/user/{id}/nama/{nama}', [UserController::class, 'userProfile']);</pre></div><div><div>← → ↺ http://localhost/POS/public/user/2241720117/nama/Dea</div><div><h2>Halo!!</h2><h3>User Profile</h3><p>ID : 2241720117</p><p>Nama : Dea</p><p>Selamat Berbelanja</p></div></div></div></div>
4	<div><div>Halaman Penjualan</div><div>Menampilkan halaman transaksi POS</div></div>

```
resources > views > sales.blade.php > html
1 <!DOCTYPE html>
2 <html lang="{{str_replace('_', '-', app()->getLocale())}}">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>PENJUALAN</title>
7 </head>
8 <body>
9 <h1>Sales Transactions</h1>
10 <table border="1">
11 <thead>
12 <tr>
13 <th>Kode</th>
14 <th>Nama Produk</th>
15 <th>Jumlah</th>
16 <th>Total</th>
17 </tr>
18 </thead>
19 <tbody>
20 @foreach ($sales as $penjualan)
21 <tr>
22 <td>{{ $penjualan['kode'] }}</td>
23 <td>{{ $penjualan['nama'] }}</td>
24 <td>{{ $penjualan['jumlah'] }}</td>
25 <td>{{ $penjualan['total'] }}</td>
26 </tr>
27 @endforeach
28 </tbody>
29 </table>
30 </body>
31 </html>
```

```
app > Http > Controllers > salesController.php > salesController > sales
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class salesController extends Controller
8 {
9     public function sales(){
10         $penjualan = [
11             ['kode' => 001, 'nama' => 'Face Wash', 'jumlah' => 3, 'total' => 100000],
12             ['kode' => 002, 'nama' => 'Ayam Geprek', 'jumlah' => 15, 'total' => 150000],
13             ['kode' => 003, 'nama' => 'Baju Bayi', 'jumlah' => 1, 'total' => 50000],
14             ['kode' => 004, 'nama' => 'Sabun', 'jumlah' => 5, 'total' => 25000]
15         ];
16         return view('sales', ['sales' => $penjualan]);
17     }
18     //
19 }
```

```
35 Route::get('/sales', [salesController::class, 'sales']);
```

← → ↻ http://localhost/POS/public/sales

Sales Transactions

Kode	Nama Produk	Jumlah	Total
1	Face Wash	3	100000
2	Ayam Geprek	15	150000
3	Baju Bayi	1	50000
4	Sabun	5	25000

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.

