

LAPORAN PRAKTIKUM
MATA KULIAH PEMROGRAMAN WEB LANJUT

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.

JOBSHEET II ROUTING, CONTROLLER, DAN VIEW



Disusun oleh :

Nama : Vita Eka Saraswati

NIM : 2341760082

No Absen : 29

JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
POLITEKNIK NEGERI MALANG

2025

PRAKTIKUM I : ROUTING

A. Basic Routing

1. Menambahkan route /hello

```
6 Route::get(uri: '/hello', action: function (): string {  
7     return 'Hello World';  
8 });  
9
```

← → ↻ ⓘ localhost/PWL2025/public/hello

Hello World

2. Menambahkan route /world

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

← → ↻ ⓘ localhost/PWL2025/public/world

World

3. Menambahkan route /home dengan pesan 'Selamat Datang'

```
14 Route::get(uri: '/home', action: function (): string {  
15     return 'Selamat Datang';  
16 });  
17
```

← → ↻ ⓘ localhost/PWL2025/public/home

Selamat Datang

4. Menambahkan route /about yang menampilkan NIM masing – masing

```
19 // Route untuk "/about" menampilkan NIM dan Nama  
20 Route::get(uri: '/about', action: function (): string {  
21     return 'NIM: 2341760082 - Nama: Vita Eka Saraswati';  
22 });
```

← → ↻ ⓘ localhost/PWL2025/public/about

NIM: 2341760082 - Nama: Vita Eka Saraswati

B. Route Parameters

1. Memanggil route /user/name dan mengirimkan parameter berupa nama user \$name

```
28 Route::get(uri: '/user/{name}', action: function ($name): string {  
29     return 'Nama saya, ' . $name;  
30 });
```

localhost/PWL2025/public/user/Vita

Nama saya Vita

2. Route yang menerima lebih dari 1 parameter. Route menerima parameter \$postId dan juga \$comment

```
33 Route::get(uri: '/posts/{post}/comments/{comment}', action: function  
34 ($postId, $commentId): string {  
35     return 'Pos ke-' . $postId . " Komentar ke-: " . $commentId;  
36 });
```

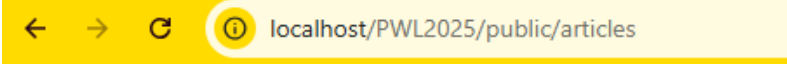
localhost/PWL2025/public/posts/1/comments/5

Pos ke-1 Komentar ke-: 5

3. Membuat route /articles/{id} yang akan menampilkan output “Halaman Artikel dengan ID {id}”, ganti id sesuai dengan input dari url.

```
39 // Route untuk menampilkan daftar artikel  
40 Route::get(uri: '/articles', action: function (): string {  
41     $articles = [  
42         1 => 'Artikel 1: Cara Belajar Laravel untuk Pemula',  
43         2 => 'Artikel 2: Panduan Dasar Routing di Laravel',  
44         3 => 'Artikel 3: Mengenal Blade Template di Laravel',  
45         4 => 'Artikel 4: Cara Menggunakan Controller dan Middleware',  
46         5 => 'Artikel 5: Database Migration dan Eloquent ORM'  
47     ];  
48  
49     $output = "<h2>Daftar Artikel</h2><ul>";  
50     foreach ($articles as $id => $title) {  
51         $output .= "<li><a href='/articles/$id'>$title</a></li>";  
52     }  
53     $output .= "</ul>";  
54  
55     return $output;  
56 });  
57
```


```
58 // Route untuk menampilkan artikel berdasarkan ID
59 Route::get(uri: '/articles/{id}', action: function ($id): string {
60     $articles = [
61         1 => 'Artikel 1: Cara Belajar Laravel untuk Pemula',
62         2 => 'Artikel 2: Panduan Dasar Routing di Laravel',
63         3 => 'Artikel 3: Mengenal Blade Template di Laravel',
64         4 => 'Artikel 4: Cara Menggunakan Controller dan Middleware',
65         5 => 'Artikel 5: Database Migration dan Eloquent ORM'
66     ];
67
68     if (array_key_exists(key: $id, array: $articles)) {
69         return "Halaman Artikel dengan ID $id - " . $articles[$id];
70     } else {
71         return "Artikel dengan ID $id tidak ditemukan.";
72     }
73 });
74
```



Daftar Artikel

- [Artikel 1: Cara Belajar Laravel untuk Pemula](#)
- [Artikel 2: Panduan Dasar Routing di Laravel](#)
- [Artikel 3: Mengenal Blade Template di Laravel](#)
- [Artikel 4: Cara Menggunakan Controller dan Middleware](#)
- [Artikel 5: Database Migration dan Eloquent ORM](#)

Jika user menambahkan parameter 1 pada url link akan menampilkan output seperti di bawah ini



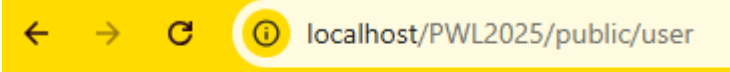
Halaman Artikel dengan ID 1 - Artikel 1: Cara Belajar Laravel untuk Pemula

C. Optional Parameters

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

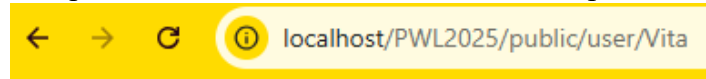
```
39 //OPTIONAL PARAMETERS
40 //A. Memanggil route user sekaligus mengirimkan parameter berupa nama user
41 Route::get(uri: '/user/{name?}', action: function ($name=null): string {
42     return 'Nama saya, ' . $name;
43 });
```

Jika tanpa parameter hanya memunculkan “Nama saya” karena mengirimkan variable `$name` bernilai null



Nama saya,

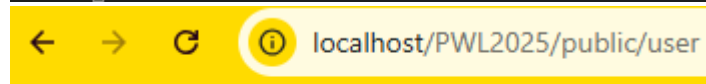
Jika parameter ditambah Nama, hasil akan seperti dibawah ini



Nama saya Vita

2. Merubah kode pada route /user menjadi seperti di bawah ini.

```
45 //B. Modifikasi name menjadi John
46 Route::get(uri: '/user/{name?}', action: function ($name='John'): string {
47     return 'Nama saya, ' . $name;
48 });
```



Nama saya, John

PRAKTIKUM II : CONTROLLER

A. Membuat Controller

1. Membuat Controller

```
PS F:\laragon\www\PWL2025> php artisan make:Controller WelcomeController

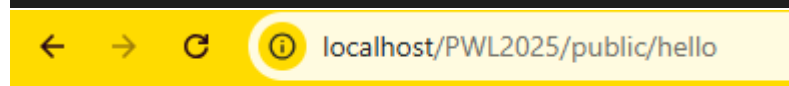
INFO Controller [F:\laragon\www\PWL2025\app\Http\Controllers\WelcomeController.php] created successfully.
```

2. Menambahkan function dengan access public untuk mendefinisikan action pada file app/Http/Controllers/WelcomeController.php

```
7 class WelcomeController extends Controller
8 {
9     0 references | 0 overrides
10     public function hello(): string{
11         return 'Hello, World!';
12     }
13 }
```

3. Setelah sebuah controller telah didefinisikan action, menambahkan controller tersebut pada route. Ubah route /hello menjadi seperti berikut:

```
107 Route::get(uri: '/hello', action: [WelcomeController::class, 'hello']);
```



Hello, World!

Output akan menampilkan teks “Hello,World!” sesuai dengan definisi fungsi pada WelcomeController yang terhubung melalui route

4. Modifikasi hasil pada praktikum poin 2 (Routing) dengan konsep controller. Pindahkan logika eksekusi ke dalam controller dengan nama PageController

- Membuat PageController

```
PS F:\laragon\www\PWL2025> php artisan make:controller PageController

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

- Definisi function pada file PageController

```

9      public function index(): string
10     {
11         return 'Selamat Datang';
12     }
13
14     1 reference | 0 overrides
15     public function about(): string
16     {
17         return 'Nama: Vita Eka, NIM: 2341760082';
18     }
19
20     1 reference | 0 overrides
21     public function articles($id): string
22     {
23         return "Halaman Artikel dengan Id {$id}";
24     }

```

- Menambahkan routes
Sebelum mendefinisikan route lakukan impor terlebih dahulu

```

7      use App\Http\Controllers\PageController; ///Mengimpor PageController unt

```

Berikut routes nya :

```

111     //Controller -memisahkan logika
112     Route::get(uri: '/', action: [PageController::class, 'index']);
113     Route::get(uri: '/about', action: [PageController::class, 'about']);
114     Route::get(uri: '/articles/{id}', action: [PageController::class, 'articles']);

```

- Hasil output

localhost/PWL2025/public/

Selamat Datang

localhost/PWL2025/public/about

Nama: Vita Eka, NIM: 2341760082

localhost/PWL2025/public/articles

Daftar Artikel

- [Artikel 1: Cara Belajar Laravel untuk Pemula](#)
- [Artikel 2: Panduan Dasar Routing di Laravel](#)
- [Artikel 3: Mengenal Blade Template di Laravel](#)
- [Artikel 4: Cara Menggunakan Controller dan Middleware](#)
- [Artikel 5: Database Migration dan Eloquent ORM](#)

5. Modifikasi kembali implementasi sebelumnya dengan konsep Single ActionController. Sehingga untuk hasil akhir yang didapatkan akan ada HomeController, AboutController dan ArticleController. Modifikasi juga route yang digunakan.

- Membuat controller

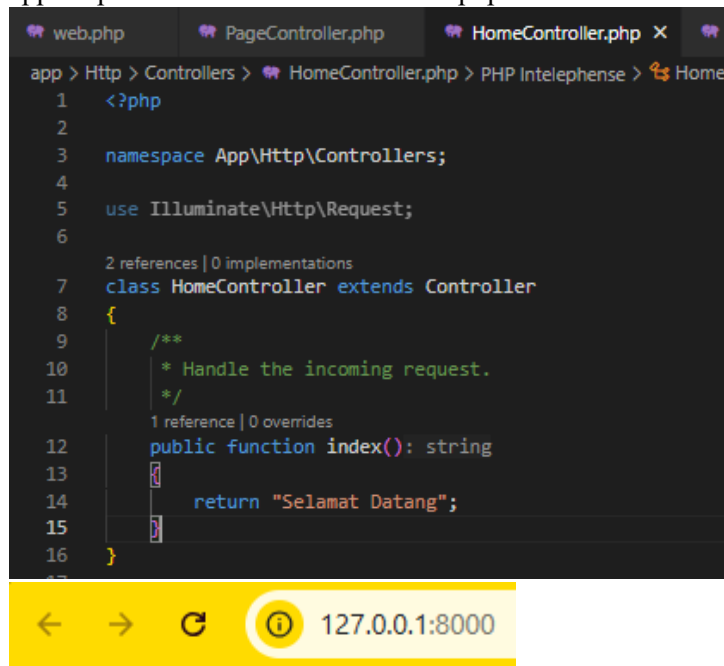
```
PS F:\laragon\www\PwL2025> php artisan make:controller PageController
php artisan make:controller HomeController --invokable
>> php artisan make:controller AboutController --invokable [PageController.php] created successfully.
>> php artisan make:controller ArticleController --invokable
>> F:\laragon\www\PwL2025>

[INFO] Controller [F:\laragon\www\PwL2025\app\Http\Controllers\HomeController.php] created successfully.

[INFO] Controller [F:\laragon\www\PwL2025\app\Http\Controllers\AboutController.php] created successfully.

[INFO] Controller [F:\laragon\www\PwL2025\app\Http\Controllers\ArticleController.php] created successfully.
```

- Menambahkan kode atau function public pada file app/Http/Controllers/HomeController.php

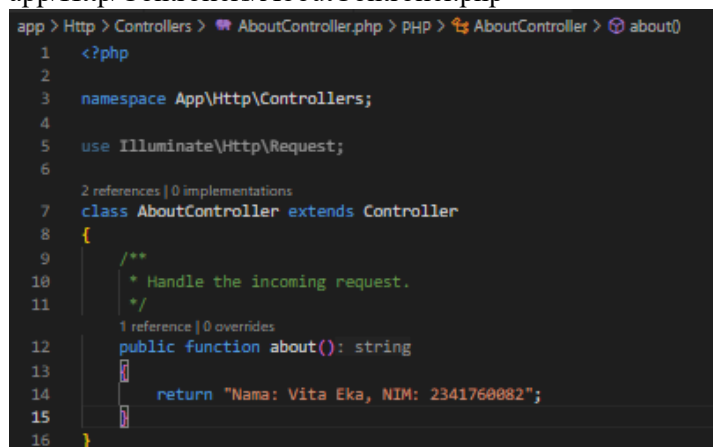


```
web.php | PageController.php | HomeController.php X
app > Http > Controllers > HomeController.php > PHP Intelephense > Home
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  2 references | 0 implementations
  class HomeController extends Controller
8  {
9      /**
10     * Handle the incoming request.
11     */
12     1 reference | 0 overrides
    public function index(): string
13     {
14         return "Selamat Datang";
15     }
16 }
```

← → ↻ ⓘ 127.0.0.1:8000

Selamat Datang

- Menambahkan kode atau function public pada file app/Http/Controllers/AboutController.php

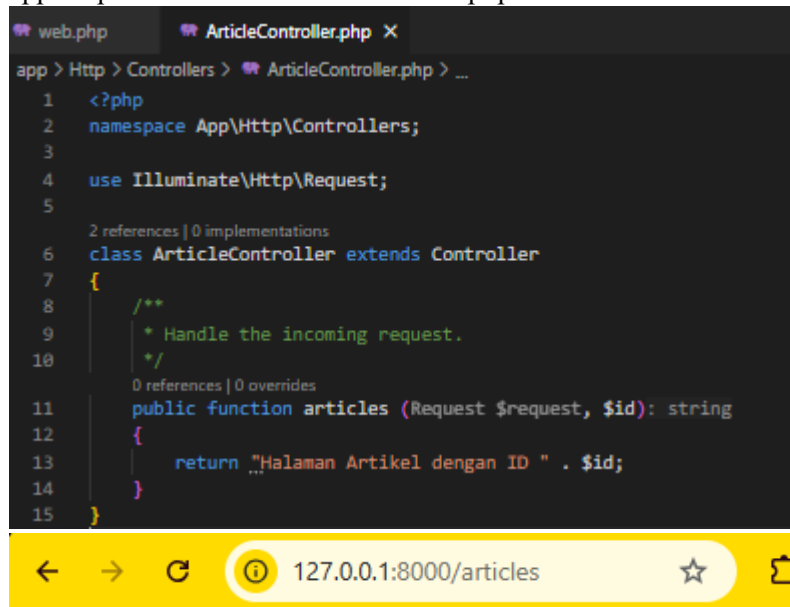


```
app > Http > Controllers > AboutController.php > PHP > AboutController > about()
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  2 references | 0 implementations
  class AboutController extends Controller
8  {
9      /**
10     * Handle the incoming request.
11     */
12     1 reference | 0 overrides
    public function about(): string
13     {
14         return "Nama: Vita Eka, NIM: 2341760082";
15     }
16 }
```



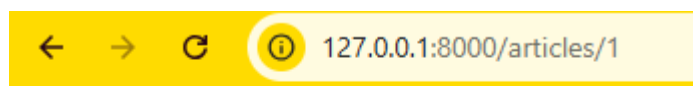
Nama: Vita Eka, NIM: 2341760082

- Menambahkan kode atau function public pada file app/Http/Controllers/ArticleController.php



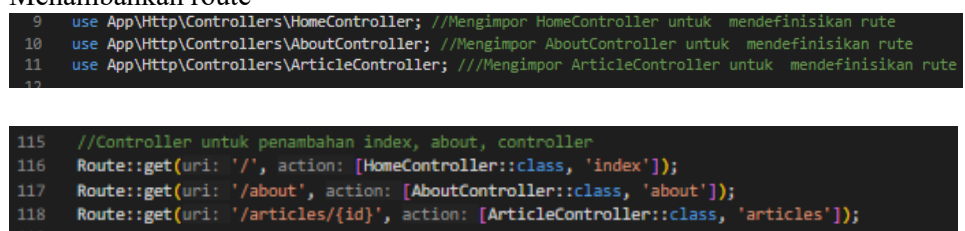
Daftar Artikel

- [Artikel 1: Cara Belajar Laravel untuk Pemula](#)
- [Artikel 2: Panduan Dasar Routing di Laravel](#)
- [Artikel 3: Mengenal Blade Template di Laravel](#)
- [Artikel 4: Cara Menggunakan Controller dan Middleware](#)
- [Artikel 5: Database Migration dan Eloquent ORM](#)



Halaman Artikel dengan ID 1

- Menambahkan route



B. Resource Controller

1. Membuat resource controller

```
PS F:\laragon\www\PNL2025> php artisan make:controller PhotoController --resource  
  
INFO Controller [F:\laragon\www\PNL2025\app\Http\Controllers\PhotoController.php] created successfully.
```

2. Membuat route agar dapat terhubung dengan frontend. Tambahkan kode program berikut pada file web.php.

```
12 use App\Http\Controllers\PhotoController; ///Mengimpor PhotoController untuk mendefinisikan rute  
  
119 Route::resource(name: 'photos', controller: PhotoController::class);  
120
```

3. Menjalankan cek list route (php artisan route:list) akan dihasilkan route berikut ini.

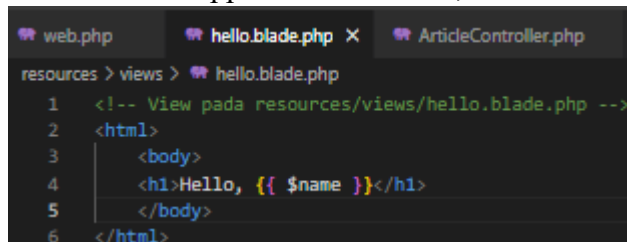
```
PS F:\laragon\www\PNL2025> php artisan route:list  
  
GET|HEAD / ..... HomeController@index  
POST _ignition/execute-solution ..... ignition.executeSolution > Spatie\LaravelIgnition > ExecuteSolutionController  
GET|HEAD _ignition/health-check ..... ignition.healthCheck > Spatie\LaravelIgnition > HealthCheckController  
POST _ignition/update-config ..... ignition.updateConfig > Spatie\LaravelIgnition > UpdateConfigController  
GET|HEAD about ..... AboutController@about  
GET|HEAD api/user .....  
GET|HEAD articles .....  
GET|HEAD articles/{id} ..... ArticleController@articles  
GET|HEAD hello ..... WelcomeController@hello
```

4. 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.

PRAKTIKUM III : VIEW

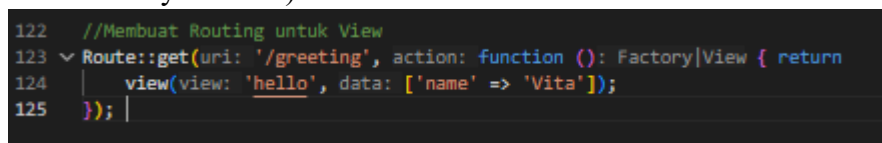
A. Membuat View

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

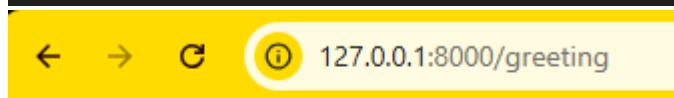


```
1 <!-- View pada resources/views/hello.blade.php -->
2 <html>
3   <body>
4     <h1>Hello, {{ $name }}</h1>
5   </body>
6 </html>
```

2. View tersebut dapat dijalankan melalui Routing, dimana *route* akan memanggil View sesuai dengan nama *file* tanpa 'blade.php'. (Catatan: Gantilah Andi dengan nama Anda yaitu Vita)



```
122 //Membuat Routing untuk View
123 Route::get(uri: '/greeting', action: function (): Factory|View { return
124 |   view(view: 'hello', data: ['name' => 'Vita']);
125 });
```



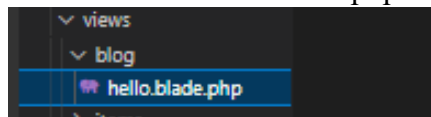
127.0.0.1:8000/greeting

Hello, Vita

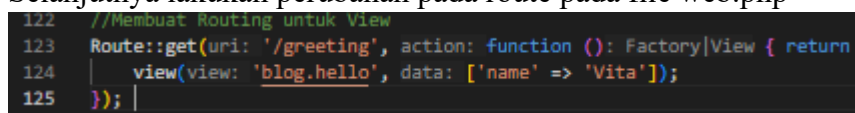
Jawab : Halaman memunculkan output/view seperti diatas, route melakukan generate dan memanggil file hello.blade.php untuk ditampilkan ke user

B. View dalam Direktori

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



3. Selanjutnya lakukan perubahan pada route pada file web.php



```
122 //Membuat Routing untuk View
123 Route::get(uri: '/greeting', action: function (): Factory|View { return
124 |   view(view: 'blog.hello', data: ['name' => 'Vita']);
125 });
```

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



127.0.0.1:8000/greeting

Hello, Vita

Jawab : Halaman akan memunculkan view dari file hello.blade.php. Route web.php generate dan memanggil file hello.blade.php untuk ditampilkan ke user

C. Menampilkan View dari Controller

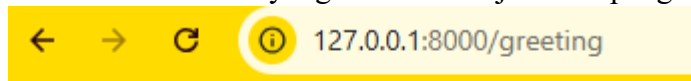
1. Buka WelcomeController.php dan tambahkan fungsi baru yaitu greeting

```
7 class WelcomeController extends Controller
8 {
9     1 reference | 0 overrides
10     public function hello(): string
11     {
12         return 'Hello World';
13     }
14
15     0 references | 0 overrides
16     public function greeting(): Factory|View
17     {
18         return view(view: 'blog.hello', data: ['name' => 'Vita']);
19     }
20 }
```

2. Ubah route /greeting dan arahkan ke WelcomeController pada fungsi greeting.

```
122 //Membuat Routing untuk View
123 Route::get(uri: '/greeting', action: [WelcomeController::class,
124     'greeting']);
```

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



Hello, Vita

Jawab : Sebelumnya route tanpa menggunakan controller, sedangkan pada langkah ini menggunakan function controller untuk memanggil file hello dalam direktori blog

D. Meneruskan data ke view

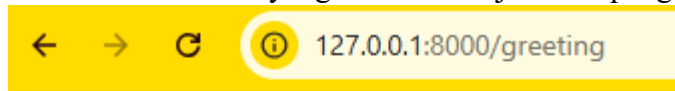
1. Buka WelcomeController.php dan tambahkan ubah fungsi greeting.

```
7 class WelcomeController extends Controller
8 {
9     1 reference | 0 overrides
10     public function hello(): string
11     {
12         return 'Hello World';
13     }
14
15     1 reference | 0 overrides
16     public function greeting(): View
17     {
18         return view(view: 'blog.hello')
19             ->with(key: 'name', value: 'Andi')
20             ->with(key: 'occupation', value: 'Astronaut');
21     }
22 }
```

2. Ubah hello.blade.php agar dapat menampilkan dua parameter.

```
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>
```

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



Hello, Andi

You are Astronaut

Jawab : Pada langkah ini bertujuan meneruskan data ke dalam view untuk ditampilkan pada user dengan lebih dari 1 parameter menggunakan with. With dapat digunakan untuk input data dalam parameter occupation

SOAL PRAKTIKUM

1. Jalankan Langkah-langkah Praktikum pada jobsheet di atas. Lakukan sinkronisasi perubahan pada project PWL2025 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 :

A. Halaman Home (Menampilkan halaman awal website)

- Membuat HomeController

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

- Menambahkan method pada HomeController

```
2 references | 0 implementations
7 class HomeController extends Controller
8
1 reference | 0 overrides
9 public function index(): Factory|View // Method untuk menampilkan halaman utama
10 {
11     return view(view: 'home.home');
12 }
13
```

- Mendefinisikan route HomeController pada file web.php

```
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\HomeController; //mendefinisikan route home
5
```

- Membuat function route HomeController pada file web.php

```
22 //1. ROUTE HALAMAN HOME
23 Route::get(uri: '/home', action: [HomeController::class, 'index']);
24
```

- View untuk halaman home

```
resources > views > home > home.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Home</title>
5 </head>
6 <body>
7 | <h1>Welcome</h1>
8 | <p>Selamat Datang di Sistem Point of Sales!</p>
9 </body>
10 </html>
```

- Output pada web browser



Welcome

Selamat Datang di Sistem Point of Sales!

B. Halaman Products (Menampilkan daftar product (route prefix))

- Membuat ProductController

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

- Menambahkan method pada ProductController

```
7 class ProductController extends Controller
8 {
9     // Menampilkan halaman kategori Food & Beverage
10    0 references | 0 overrides
11    public function foodBeverage(): Factory|View
12    {
13        return view(view: 'products.food-beverage');
14    }
15
16    // Menampilkan halaman kategori Beauty & Health
17    0 references | 0 overrides
18    public function beautyHealth(): Factory|View
19    {
20        return view(view: 'products.beauty-health');
21    }
22
23    // Menampilkan halaman kategori Home Care
24    0 references | 0 overrides
25    public function homeCare(): Factory|View
26    {
27        return view(view: 'products.home-care');
28    }
29
30    // Menampilkan halaman kategori Baby & Kid
31    0 references | 0 overrides
32    public function babyKid(): Factory|View
33    {
34        return view(view: 'products.baby-kid');
35    }
36 }
```

- Mendefinisikan route ProductController pada file web.php

```
5 use App\Http\Controllers\ProductController; //mendefinisikan route product
```

- Membuat function route ProductController pada file web.php

```

27 //2. ROUTE HALAMAN PRODUCT DGN PREFIX
28 // Halaman Products dengan Prefix
29 Route::prefix(prefix: '/category')->group(callback: function (): void {
30     Route::get(uri: '/food-beverage', action: [ProductController::class, 'foodBeverage']);
31     Route::get(uri: '/beauty-health', action: [ProductController::class, 'beautyHealth']);
32     Route::get(uri: '/home-care', action: [ProductController::class, 'homeCare']);
33     Route::get(uri: '/baby-kid', action: [ProductController::class, 'babyKid']);
34 });
35

```

- View untuk halaman product
 - o baby-kid.blade.php

```

resources > views > products > baby-kid.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Baby Kids Section</title>
5 </head>
6 <body>
7 | <h1>Produk Perlengkapan Khusus Bayi dan Anak</h1>
8 | <p>Daftar produk kategori ini.</p>
9 </body>
10 </html>
11

```

- o beauty-health.blade.php

```

resources > views > products > beauty-health.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Beauty Health Section</title>
5 </head>
6 <body>
7 | <h1>Produk Perlengkapan Khusus Perawatan Kecantikan</h1>
8 | <p>Daftar produk kategori ini.</p>
9 </body>
10 </html>
11

```

- o food-beverage.blade.php

```

resources > views > products > food-beverage.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Food Beverage Section</title>
5 </head>
6 <body>
7 | <h1>Produk Perlengkapan Khusus Makanan & Minuman</h1>
8 | <p>Daftar produk kategori ini.</p>
9 </body>
10 </html>
11

```

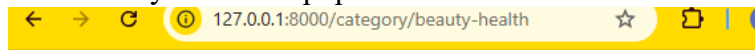
- o home-care.blade.php

```

resources > views > products > home-care.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>Home Care Section</title>
5 </head>
6 <body>
7 | <h1>Produk Perlengkapan Khusus Perlengkapan Rumah</h1>
8 | <p>Daftar produk kategori ini.</p>
9 </body>
10 </html>
11

```

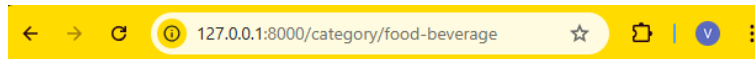
- Output pada web browser
 - o baby-kid.blade.php



Produk Perlengkapan Khusus Perawatan Kecantikan

Daftar produk kategori ini

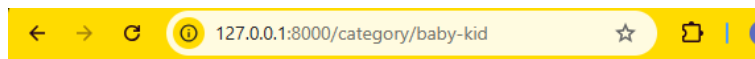
- o beauty-health.blade.php



Produk Perlengkapan Khusus Makanan & Minuman

Daftar produk kategori ini.

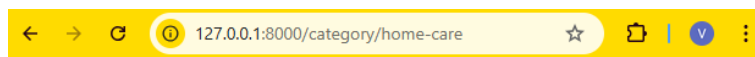
- o food-beverage.blade.php



Produk Perlengkapan Khusus Bayi dan Anak

Daftar produk kategori ini.

- o home-care.blade.php



Produk Perlengkapan Khusus Perlengkapan Rumah

Daftar produk kategori ini.

C. Halaman User (Menampilkan profil pengguna (route param))

- Membuat UserController

```
PS F:\laragon\www\POS> php artisan make:controller UserController;

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

- Menambahkan method pada UserController

```
2 references | 0 implementations
7 ~ class UserController extends Controller
8 ~
1 reference | 0 overrides
9 ~ public function show($id, $name): Factory|View
10 ~ {
11 ~     return view(view: 'user.profil', data: compact(var_name: 'id', var_names: 'name'));
12 ~ }
13 ~
```


- Mendefinisikan route UserController pada file web.php

```
6 use App\Http\Controllers\UserController; //mendefinisikan route user
```

- Membuat function route ProductController pada file web.php

```
35 //3. ROUTE USER
36 Route::get(uri: '/user/{id}/name/{name}', action: [UserController::class, 'show']);
37
```

- View untuk halaman user

```
resources > views > users > profilbalde.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 | <title>User Profile</title>
5 </head>
6 <body>
7 | <h1>Profile User</h1>
8 | <p>ID: {{ $id }}</p> <!-- menerima ID yang dikirim dari controller -->
9 | <p>Name: {{ $name }}</p> <!-- menerima Nama yang dikirim dari controller -->
10 </body>
11 </html>
12
```

- Output pada web browser

← → ↻ ⓘ 127.0.0.1:8000/user/1/name/Vita

Profile User

ID: 1

Name: Vita

D. Halaman Penjualan (Menampilkan halaman transaksi POS)

- Membuat SalesController

```
PS F:\laragon\www\POS> php artisan make:controller SalesController;

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

- Menambahkan method pada SalesController

```
app > Http > Controllers > SalesController.php > PHP Intelephense > S
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 1 reference | 0 implementations
8 class SalesController extends Controller
9 {
10 | 0 references | 0 overrides
11 | public function index(): Factory|View
12 | {
13 | | return view(view: 'sales.index');
14 | }
```

- Mendefinisikan route SalesController pada file web.php

```
7 use App\Http\Controllers\SalesController; //mendefinisikan route sales
8
```

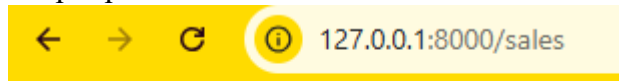
- Membuat function route UserController pada file web.php

```
39 //4. ROUTE SALES/PENJUALAN
40 Route::get(uri: '/sales', action: [SalesController::class, 'index']);
41
42 Route::get(uri: '', action: function (): Factory|View {
43     return view(view: 'welcome');
44 });
45
```

- View untuk halaman transaksi POS

```
web.php index.blade.php X profil.blade.php
resources > views > sales > index.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <title>Point of Sales</title>
5 </head>
6 <body>
7     <h1>Halaman Penjualan</h1>
8     <p>Fitur transaksi POS.</p>
9 </body>
10 </html>
```

- Output pada web browser



Halaman Penjualan

Fitur transaksi POS.