



Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis
Semester : 4 (empat) / 6 (enam)
Pertemuan ke- : 1 (satu)

JOBSHEET 02

1. Routing

Langkah-langkah Praktikum:

- a. Pada bagian ini, kita akan membuat dua buah route dengan ketentuan sebagai berikut.

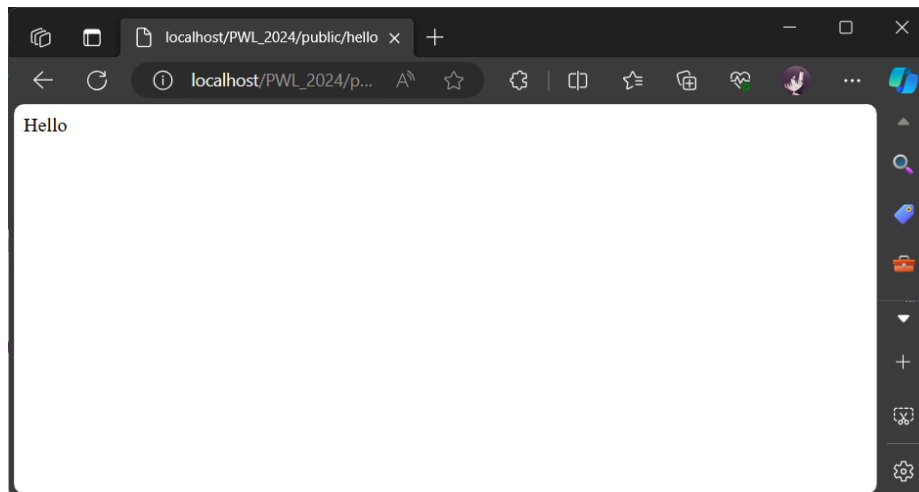
No	Http Verb	Url	Fungsi
1	get	/hello	Tampilkan String Hello ke browser.
2	get	/world	Tampilkan String World ke browser

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'; });
```

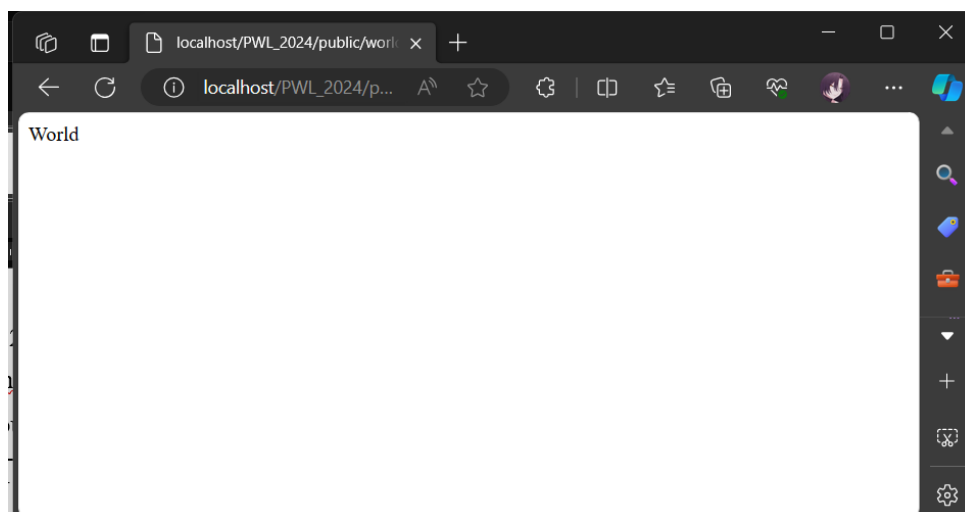
```
16 Route::get('/hello', function () {  
17     return 'Hello';  
18 });  
19
```



- 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';  
});
```

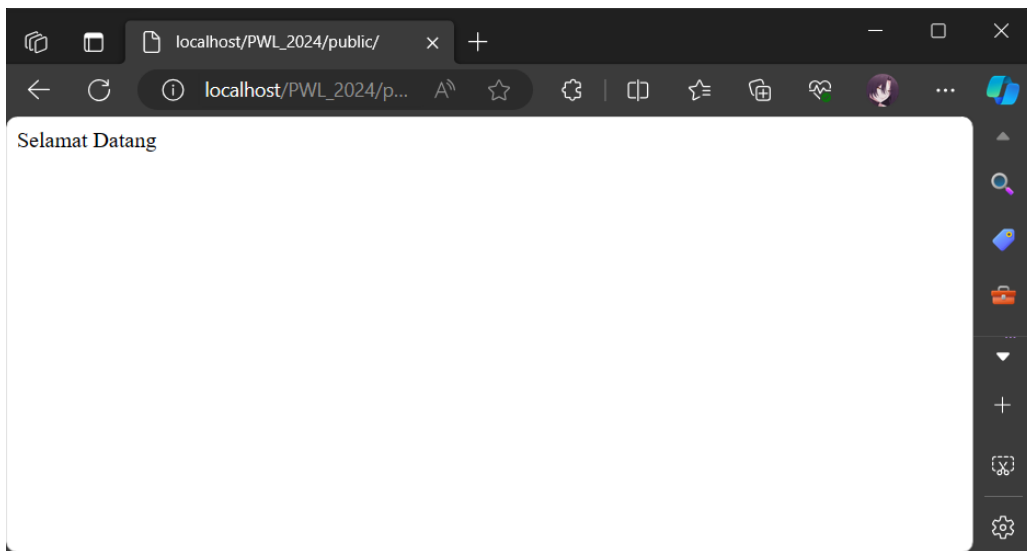
```
20 Route::get('/world', function () {  
21     return 'World';  
22 });
```





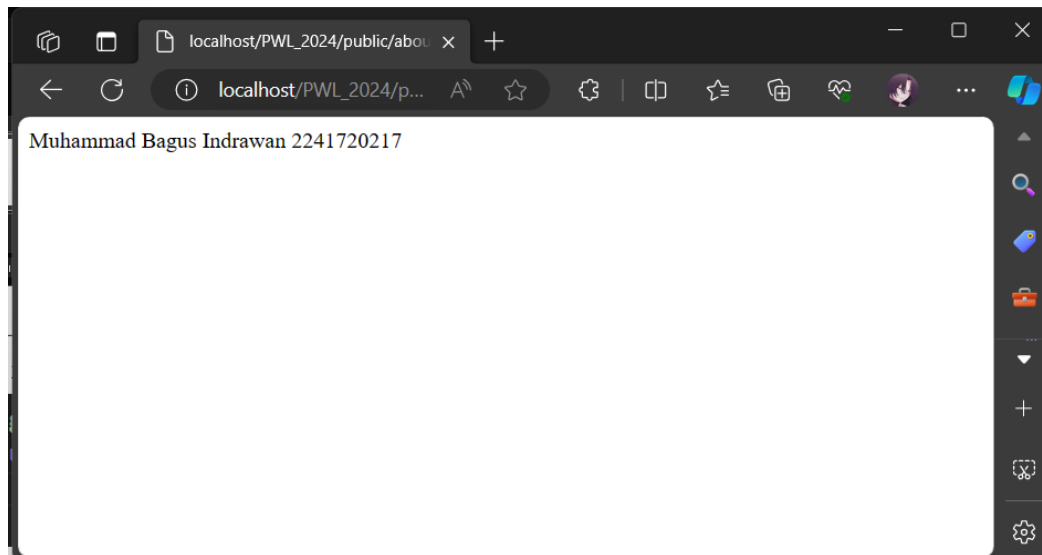
- e. Bukalah pada browser, tuliskan URL untuk memanggil route tersebut: localhost/PWL_2024/public/world. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.
- f. Selanjutnya, cobalah membuat route '/' yang menampilkan pesan 'Selamat Datang'.

```
24 Route::get('/', function () {  
25     return 'Selamat Datang';  
26 });
```



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

```
28 Route::get('/about', function () {  
29     return 'Muhammad Bagus Indrawan 2241720217';  
30 });
```



- Route Parameters

Terkadang saat membuat sebuah URL, kita perlu mengambil sebuah parameter yang merupakan bagian dari segmen URL dalam route kita. Misalnya, kita membutuhkan nama user yang dikirim melalui sebuah URL.

Langkah-langkah Praktikum:

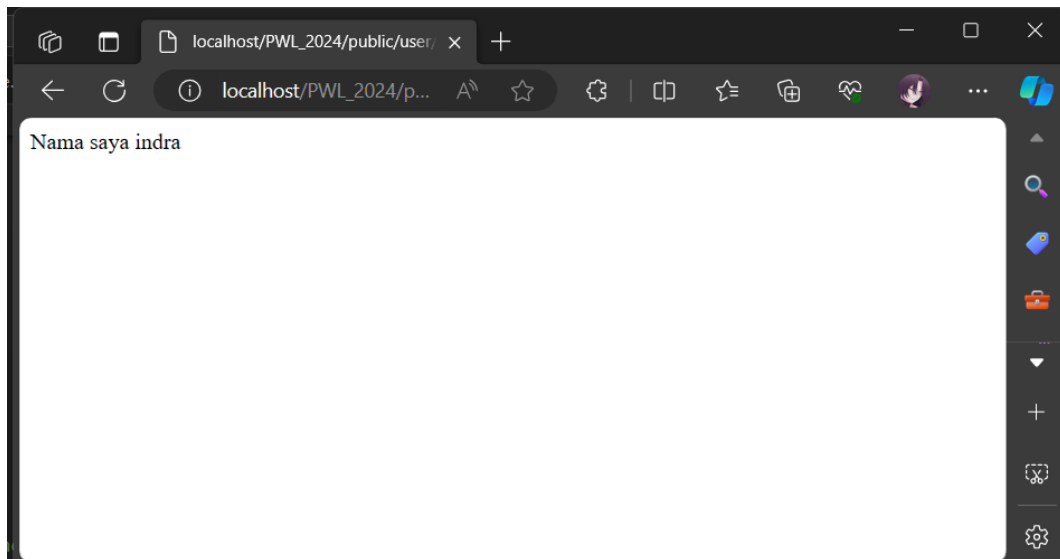
Untuk membuat routing dengan parameter dapat dilakukan dengan cara berikut ini.

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

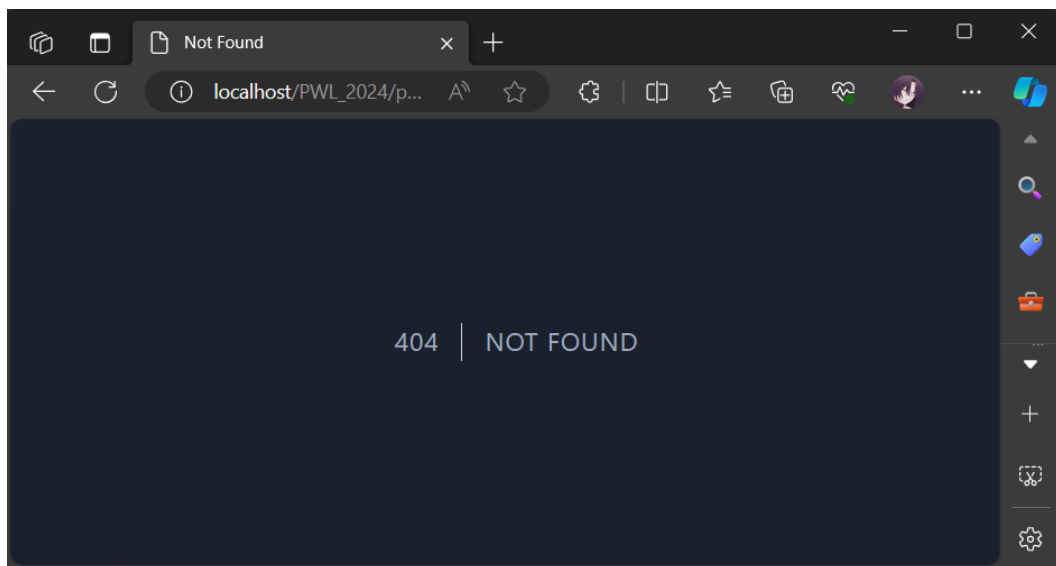
```
Route::get( '/user/{name}', function  
($name) {  
    return 'Nama saya  
    '.$name; } );
```

```
32 Route::get('/user/{name}', function ($name) {  
33     return 'Nama saya '.$name;  
34 });
```

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



- c. URL: `localhost/PWL_2024/public/user/`.



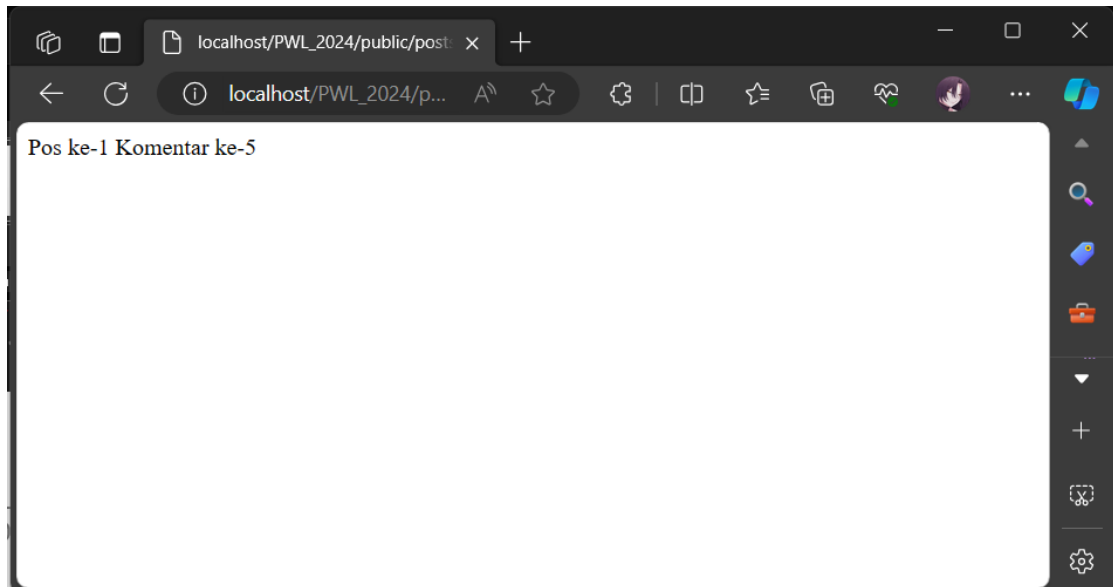
- d. Suatu route, juga bisa menerima lebih dari 1 parameter seperti kode berikut ini. Route menerima parameter `$postId` dan juga `$comment`.

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



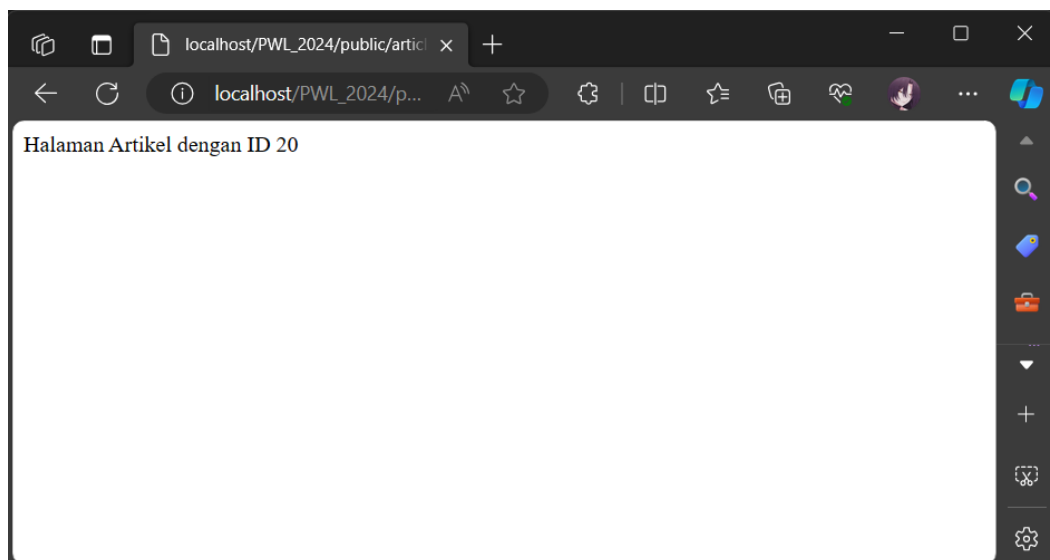
```
36 Route::get('/posts/{post}/comments/{comment}', function ($postId, $commentId) {  
37     return 'Pos ke-' . $postId . " Komentar ke-" . $commentId;  
38 });
```

- e. URL: `localhost/PWL_2024/public/posts/1/comments/5`.



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

```
40 Route::get('/articles/{id}', function ($id) {  
41     return 'Halaman Artikel dengan ID ' . $id;  
42 });
```





Langkah-langkah Praktikum:

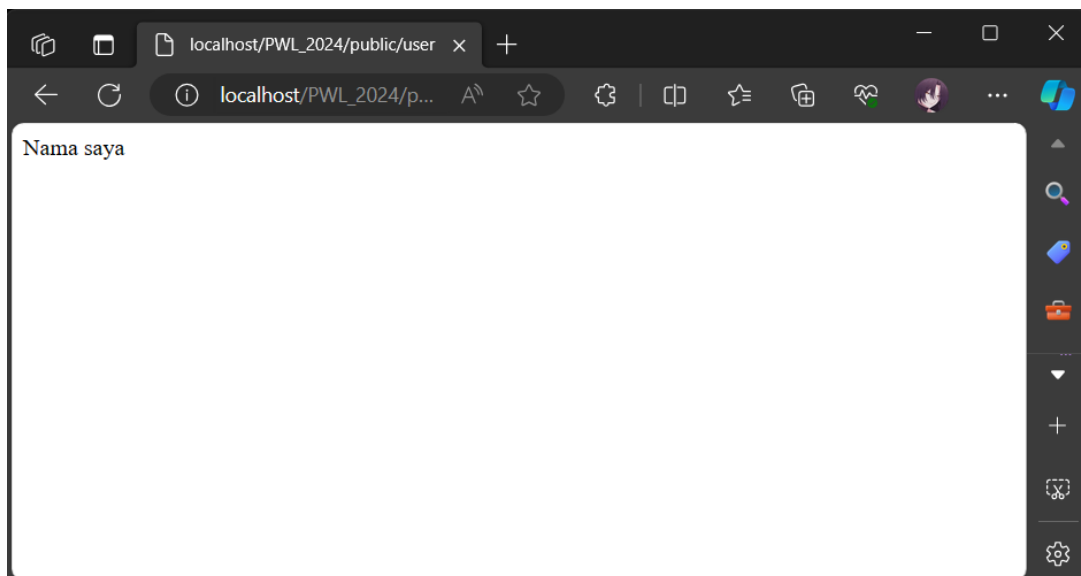
Untuk membuat routing dengan optional parameter dapat dilakukan dengan cara berikut ini.

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

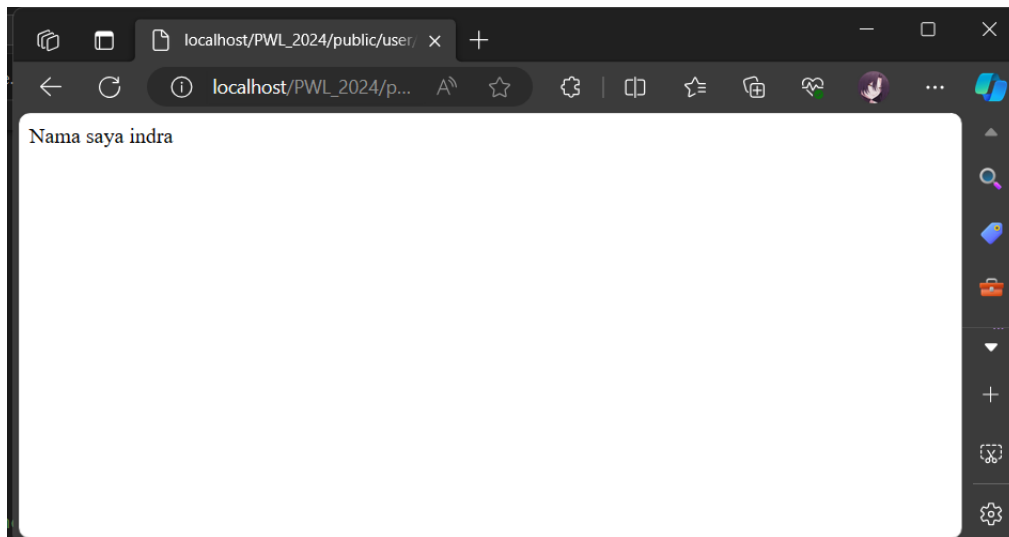
```
Route::get( '/user/{name?}', function ($name=null) {  
    return 'Nama saya '.$name;  
});
```

```
44 Route::get('/user/{name?}', function ($name=null) {  
45 |     return 'Nama saya '.$name;  
46 | });|
```

- h. URL: **localhost/PWL_2024/public/user/**.



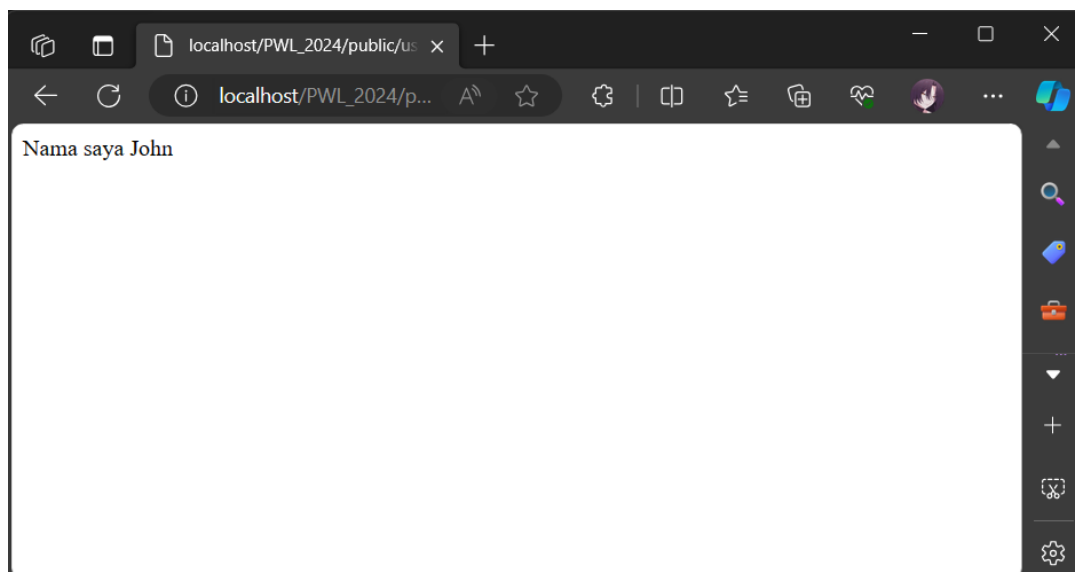
- i. URL: **localhost/PWL_2024/public/user>NamaAnda**.



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

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

- k. URL: **localhost/PWL_2024/public/user/**



2. Controller



Langkah-langkah Praktikum:

- 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 C:\laragon\www\PWL_2024> php artisan make:controller WelcomeController  
INFO Controller [C:\laragon\www\PWL_2024\app\Http\Controllers\WelcomeController.php] created successfully.  
PS C:\laragon\www\PWL_2024> 
```

- 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';
    }
}
```

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  0 references | 0 implementations
8  class WelcomeController extends Controller
9  {
10     0 references | 0 overrides
11     public function hello(){
12         return 'Hello World';
13     }
14 }
```

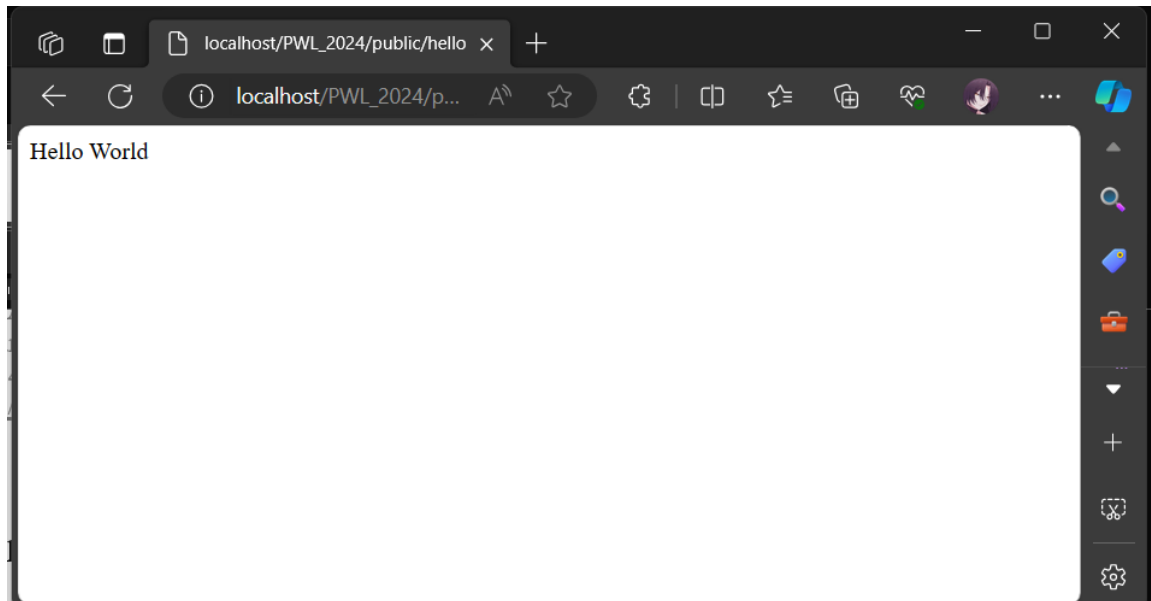
- d. Setelah sebuah controller telah didefinisikan action, kita dapat menambahkan controller tersebut pada route. Ubah route /hello menjadi seperti berikut:

```
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`.

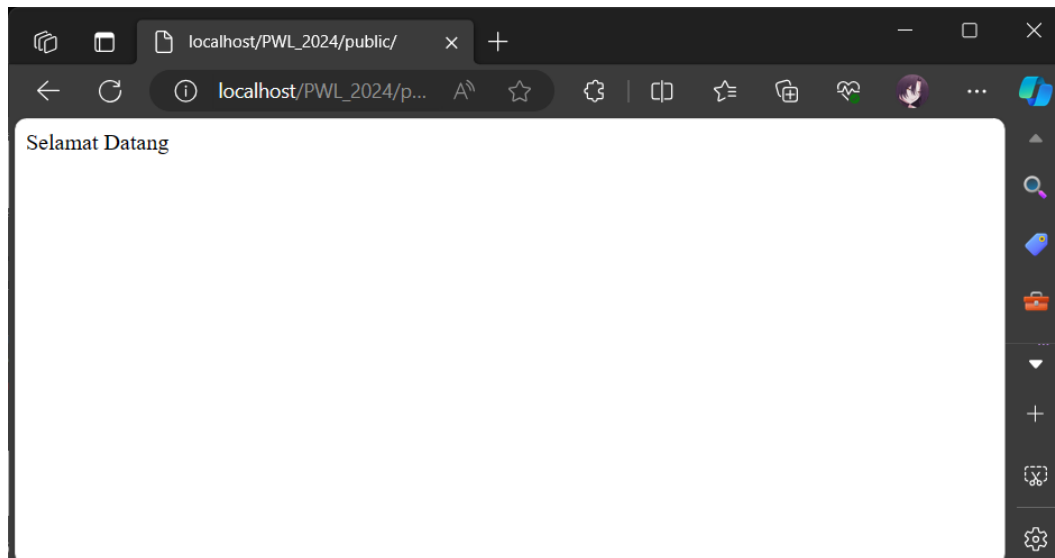


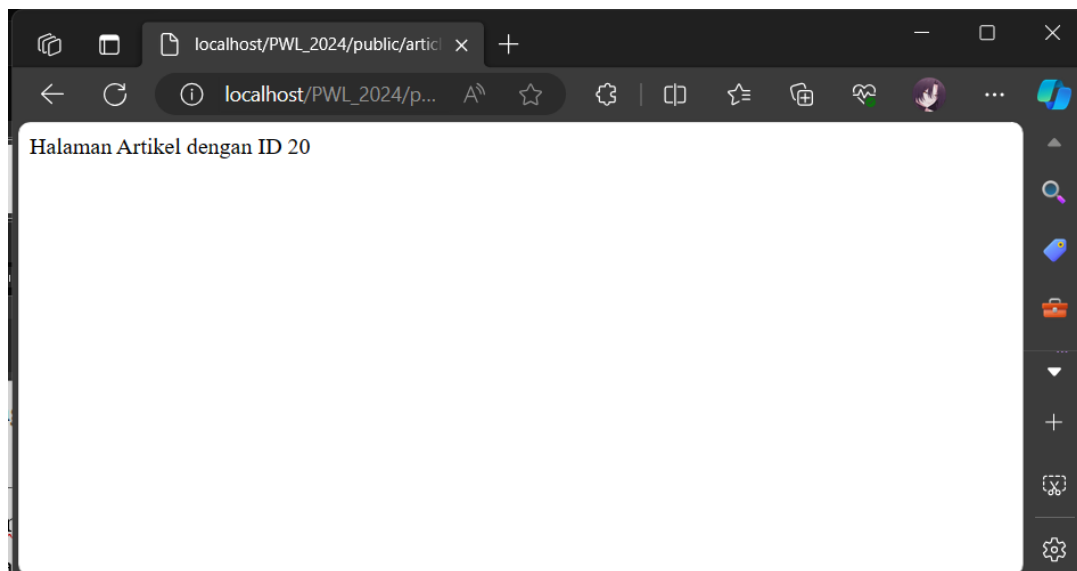
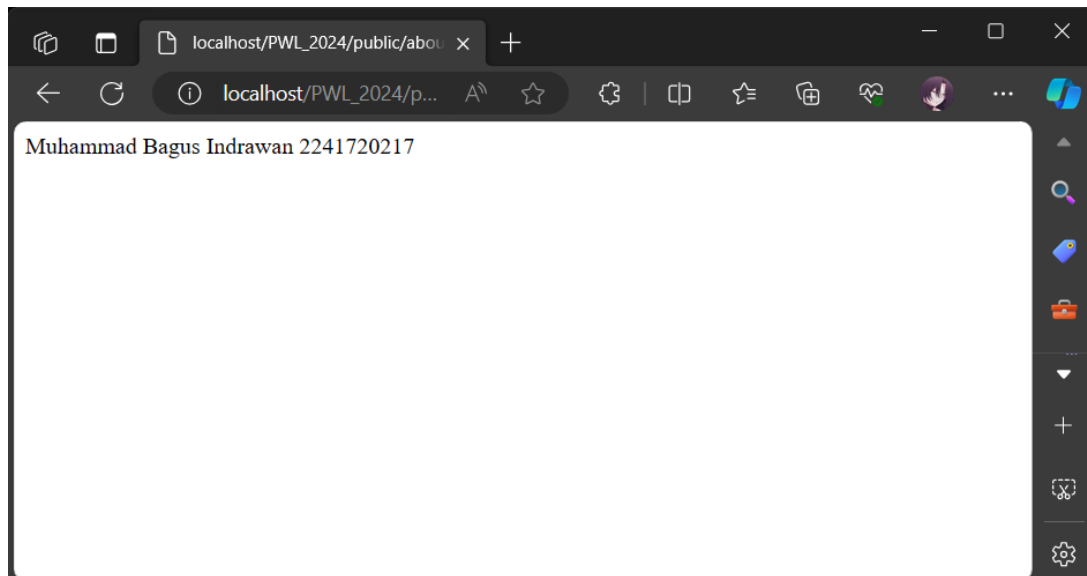
- f. Modifikasi hasil pada praktikum poin 2 (Routing) dengan konsep controller. Pindahkan logika eksekusi ke dalam controller dengan nama PageController.

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		



```
app > Http > Controllers > PageController.php > PHP Intelephense > PageController
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  0 references | 0 implementations
8  class PageController extends Controller
9  {
10     0 references | 0 overrides
11     public function index(){
12         return 'Selamat Datang';
13     }
14
15     0 references | 0 overrides
16     public function about(){
17         return 'Muhammad Bagus Indrawan 2241720217';
18     }
19
20     0 references | 0 overrides
21     public function articles($id){
22         return 'Halaman Artikel dengan ID '.$id;
23     }
24 }
```





- 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.



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

Langkah-langkah Praktikum:

- 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.

- 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);
```

```
use App\Http\Controllers\PhotoController;
```

```
Route::get('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
	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 C:\laragon\www\PWL_2024> php artisan route:list
```

```
GET|HEAD / ..... HomeController@index
POST _ignition/execute-solution ..... ignition.executeSolution > Spatie\LaravelIgnition > ExecutesSolutionController
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/{id} ..... ArticleController@articles
GET|HEAD hello ..... WelcomeController@hello
GET|HEAD photos .....
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 [14] 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' ]);
```

3. View

Langkah-langkah Praktikum:

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

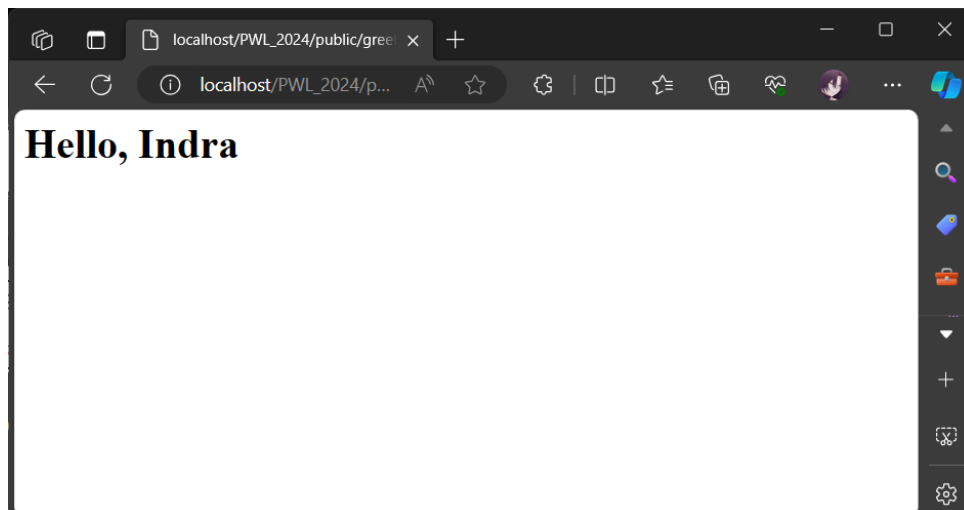


- b. View tersebut dapat dijalankan melalui Routing, dimana *route* akan memanggil View sesuai dengan nama *file* tanpa 'blade.php'. (Catatan: Gantilah Andi dengan nama Anda)

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

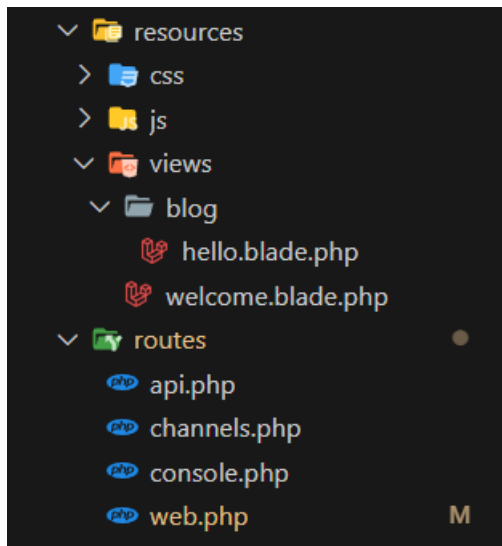
```
Route::get('/greeting', function(){  
    return view('hello', ['name' => 'Indra']);  
});
```

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



Langkah-langkah Praktikum:

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

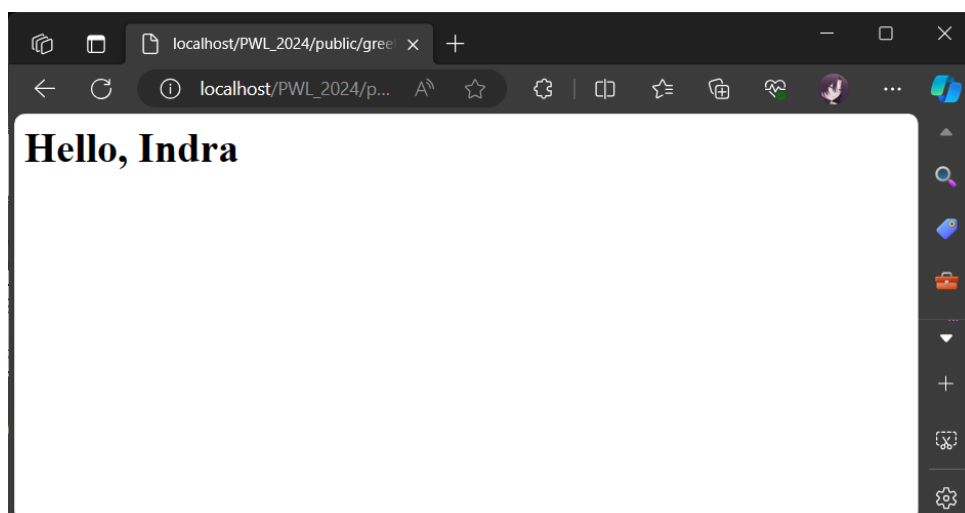


- c. Selanjutnya lakukan perubahan pada route.

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

```
90 Route::get('/greeting', function(){  
91     return view('blog.hello', ['name' => 'Indra']);  
92 });
```

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



Langkah-langkah Praktikum:



- 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']);
    }
}
```

```

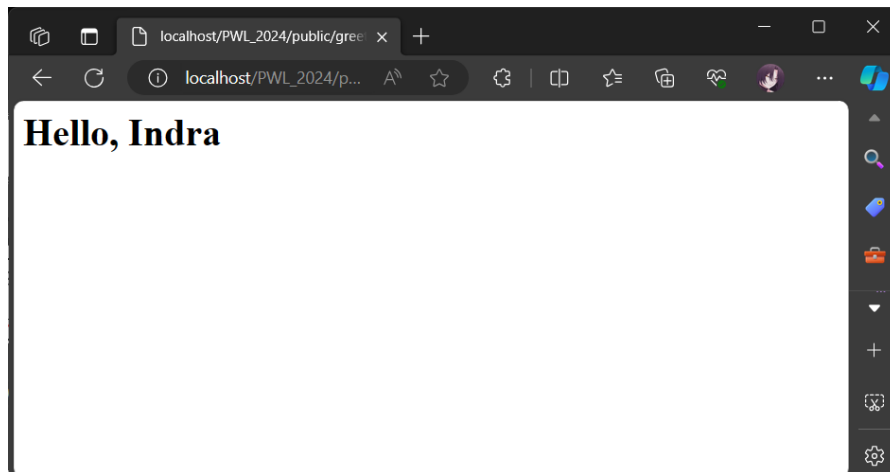
6
3 references | 0 implementations
7 class WelcomeController extends Controller
8 {
9     1 reference | 0 overrides
10     public function hello(){
11         return 'Hello World';
12     }
13     1 reference | 0 overrides
14     public function greeting(){
15         return view('blog.hello')
16         ->with('name', 'Indra');
17     }
18 }
```

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

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

```
93
94 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.



Langkah-langkah Praktikum:

- 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');
    }
}
```

```
3 references | 0 implementations
7  class WelcomeController extends Controller
8  {
9      1 reference | 0 overrides
10     public function hello(){
11         return 'Hello World';
12     }
13     1 reference | 0 overrides
14     public function greeting(){
15         return view('blog.hello')
16             ->with('name', 'Indra')
17             ->with('occupation','Programmer');
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>
```

```
PWL_2024 > 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>
```

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

