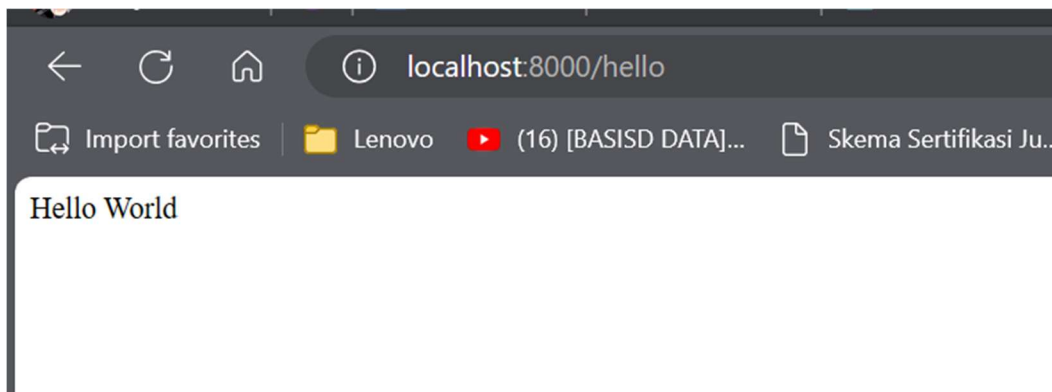


Basic Routing

- a. Buka file routes/web.php. Tambahkan sebuah route untuk nomor 1 seperti di bawah

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 Route::get('/hello', function () {
6     return 'Hello World';
7 });
```

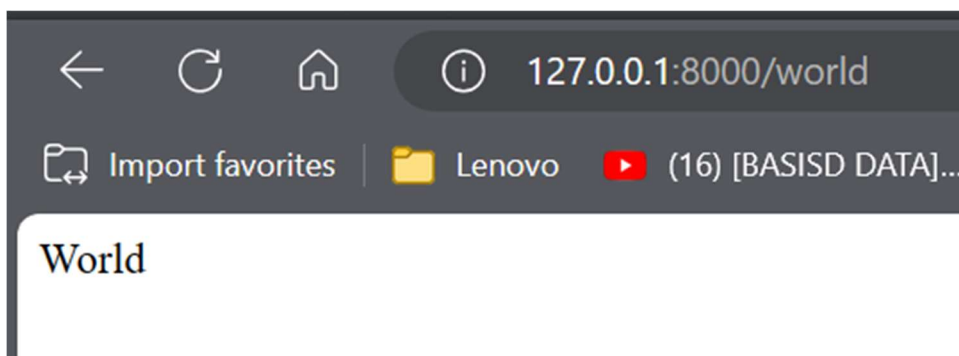
- b. Buka browser, tuliskan URL untuk memanggil route tersebut: localhost/PWL_2024/public/hello. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda



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

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

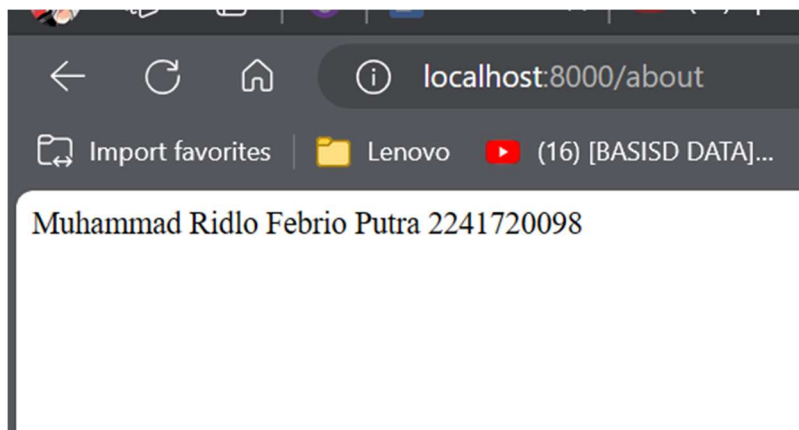
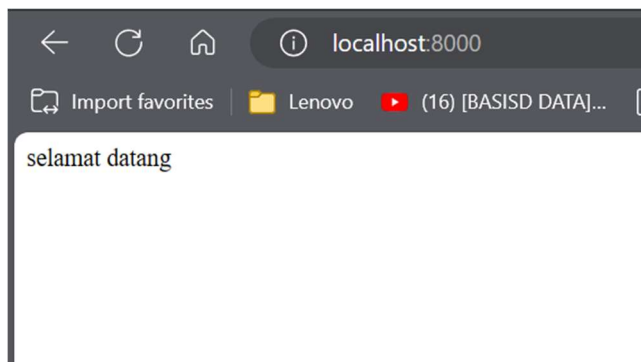
- d. Bukalah pada browser, tuliskan URL untuk memanggil route tersebut: localhost/PWL_2024/public/world. Perhatikan halaman yang muncul apakah sudah sesuai dan jelaskan pengamatan Anda.



- e. Selanjutnya, cobalah membuat route '/' yang menampilkan pesan 'Selamat Datang'. Kemudian buatlah route '/about' yang akan menampilkan NIM dan nama Anda.

```
routes > web.php > Closure
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  Route::get('/hello', function () {
6      return 'Hello World';
7  });
8
9  Route::get('/', function () {
10     return 'selamat datang';
11 });
12
13 Route::get('/about', function () {
14     return 'Muhammad Ridlo Febrio Putra 2241720098';
15 });
```

f. Hasil

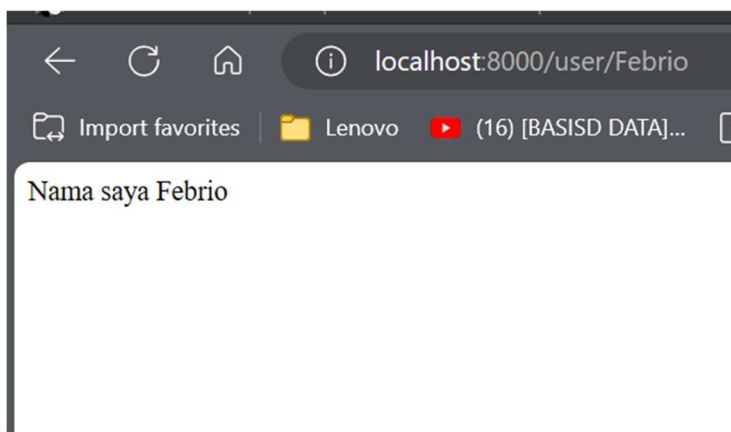


Route Parameter

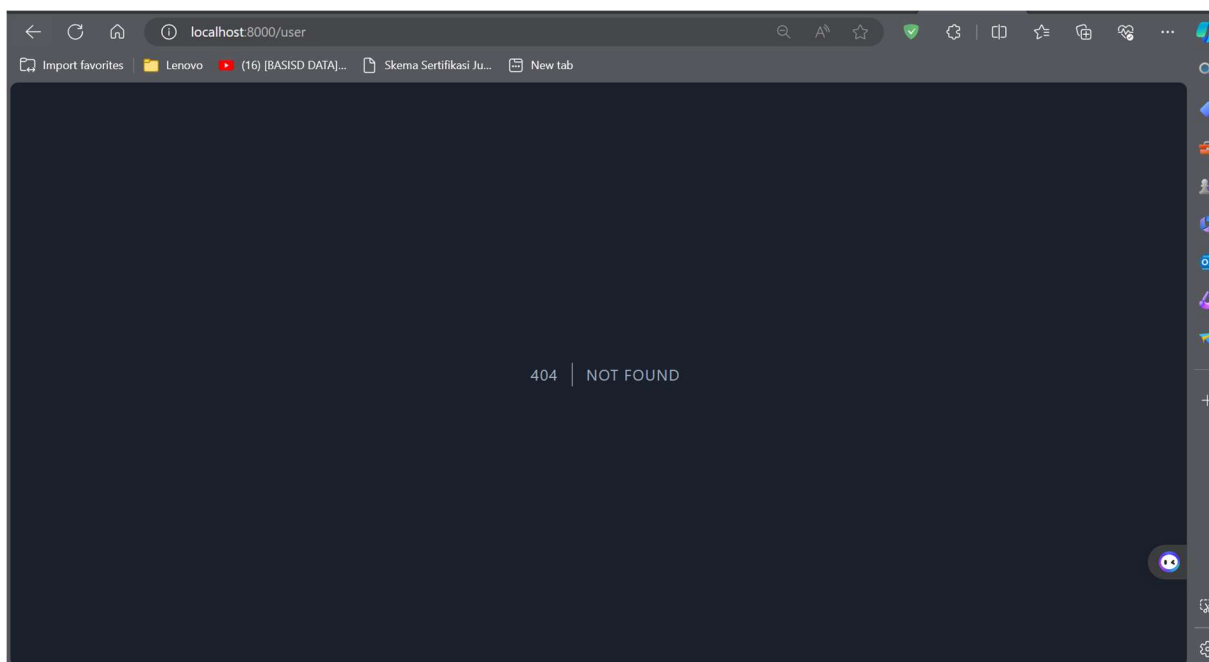
- Kita akan memanggil route `/user/{name}` sekaligus mengirimkan parameter berupa nama user `$name` seperti kode di bawah ini.

```
15  
16  
17 Route::get('/user/{name}', function ($name) {  
18     return 'Nama saya ' . $name;  
19     });  
20
```

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



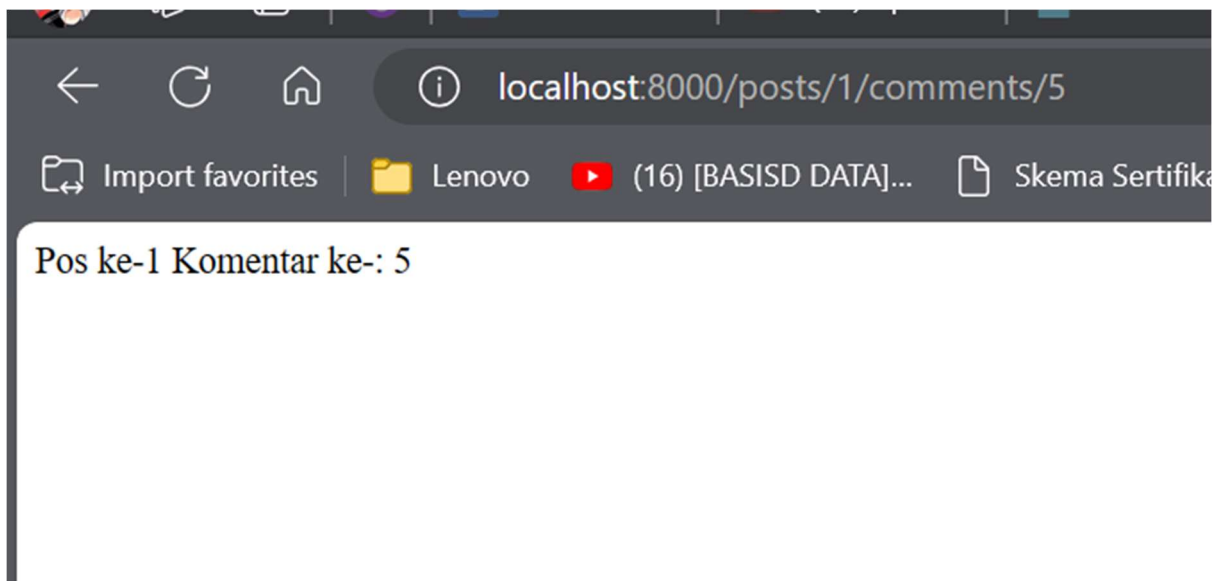
- Selanjutnya, coba tuliskan URL: `localhost/PWL_2024/public/user/`. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda



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

```
20
21 Route::get('/posts/{post}/comments/{comment}', function
22     ($postId, $commentId) {
23     return 'Pos ke-' . $postId . " Komentar ke-: " . $commentId;
24     });
25
```

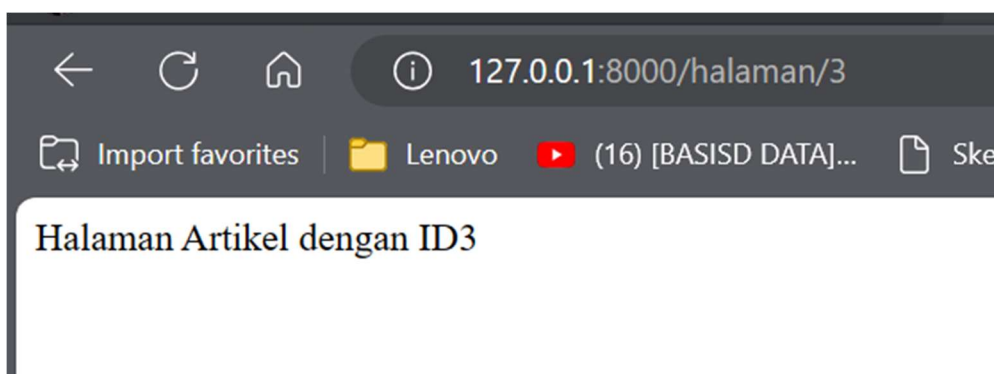
- e. Jalankan kode dengan menuliskan URL untuk memanggil route tersebut: localhost/PWL_2024/public/posts/1/comments/5. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



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

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

- g. Hasil

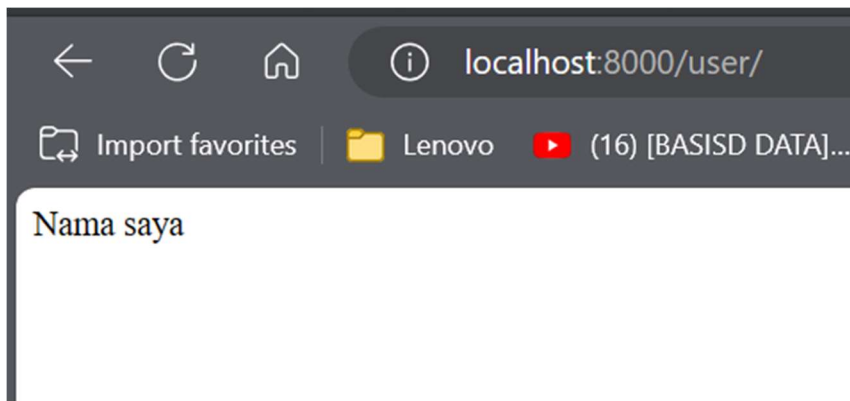


Optional Parameters

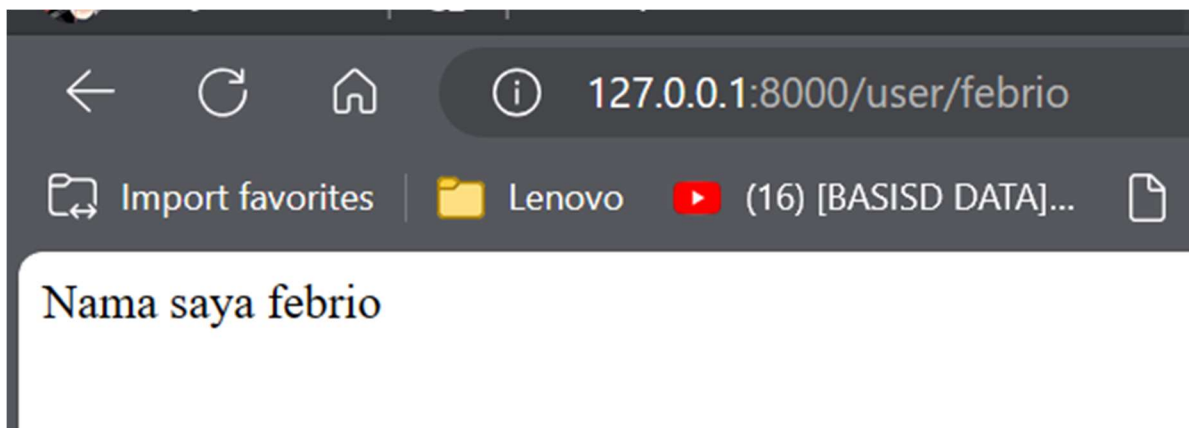
- a. Kita akan memanggil route /user sekaligus mengirimkan parameter berupa nama user \$name dimana parameternya bersifat opsional.

```
25  
26 Route::get('/user/{name?}', function ($name=null) {  
27     return 'Nama saya '.$name;  
28 });  
29
```

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



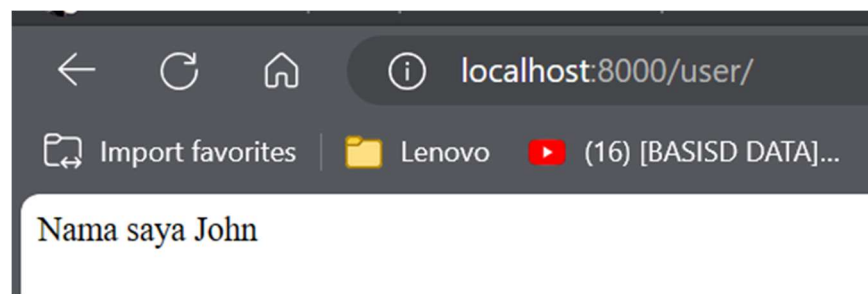
- c. Selanjutnya tuliskan URL: localhost/PWL_2024/public/user>NamaAnda. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda



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

```
0 Route::get('/user/{name?}', function ($name='John') {  
1     return 'Nama saya '.$name;  
2 });  
3
```

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



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.

```
PS D:\Laravel Kuliah\PWL_2024> php artisan make:controller WelcomeController  
  
INFO Controller [D:\Laravel Kuliah\PWL_2024\app\Http\Controllers\WelcomeController.php] created successfully.
```

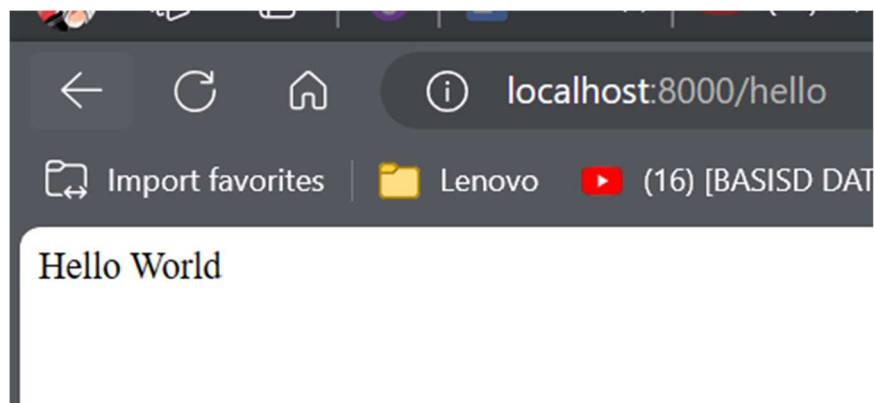
- b. Buka file pada app/Http/Controllers/WelcomeController.php.
- c. Untuk mendefinisikan action, silahkan tambahkan function dengan access public. Sehingga controller di atas menjadi sebagai berikut:

```
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
  
2 references | 0 implementations  
class WelcomeController extends Controller  
{  
    0 references | 0 overrides  
    public function hello() {  
        return 'Selamat Datang';  
    }  
}
```

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

```
74  
75 Route::get('/hello', [WelcomeController::class, 'hello']);  
76  
77
```

- 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

```
1 reference | 0 implementations
class PageController extends Controller
{
    0 references | 0 overrides
    public function hello() {
        return 'Selamat Datang';
    }
    0 references | 0 overrides
    public function about(){
        $nama = "Muhammad Ridlo Febrio Putra";
        $nim = "2241720098";

        return $nama . " dan " . $nim;
    }
    0 references | 0 overrides
    public function artitel($id){
        if ($id != null) {
            $halaman = $id;
        }
        return "Halaman Artikel dengan yang ke". $halaman;
    }
}
```

- 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.
- a. HomeController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

2 references | 0 implementations
class HomeController extends Controller
{
    1 reference | 0 overrides
    public function hello() {
        return 'Selamat Datang';
    }
}
```

- b. AboutController


```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

2 references | 0 implementations
class AboutController extends Controller
{
    1 reference | 0 overrides
    public function about(){
        $nama = "Muhammad Ridlo Febrio Putra";
        $nim = "2241720098";

        return $nama . " dan " . $nim;
    }
}
```

c. ArticleController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

2 references | 0 implementations
class ArticleController extends Controller
{
    1 reference | 0 overrides
    public function artitel($id){
        if ($id != null) {
            $halaman = $id;
        }
        return "Halaman Artikel dengan yang ke". $halaman;
    }
}
```

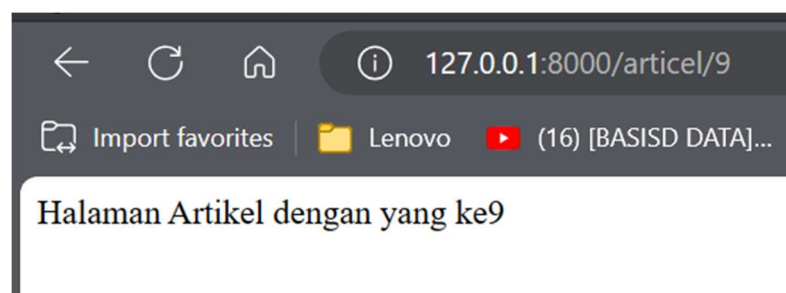
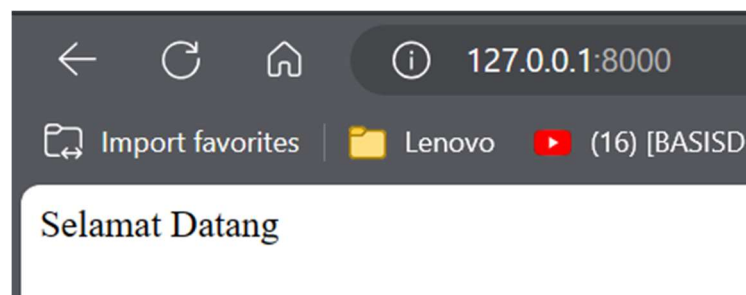
d. Route

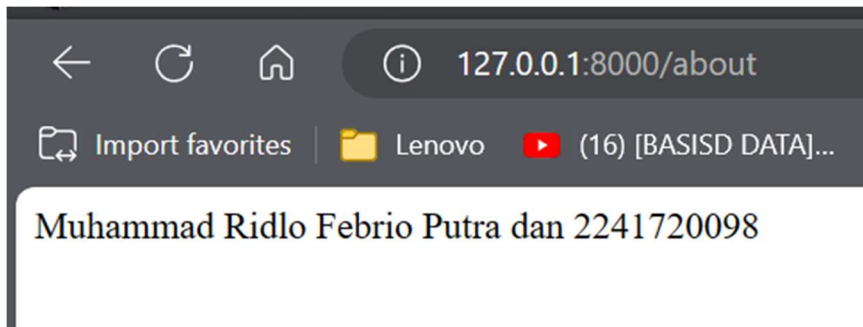
```
Route::get('/', [HomeController::class,'hello']);

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

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

Hasil





Resource Controller

- Untuk membuatnya dilakukan dengan menjalankan perintah berikut ini di terminal (php artisan make:controller PhotoController --resource)
- Setelah controller berhasil degenerate, selanjutnya harus dibuatkan route agar dapat terhubung dengan frontend. Tambahkan kode program berikut pada file web.php

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

- 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'
]);
```

View

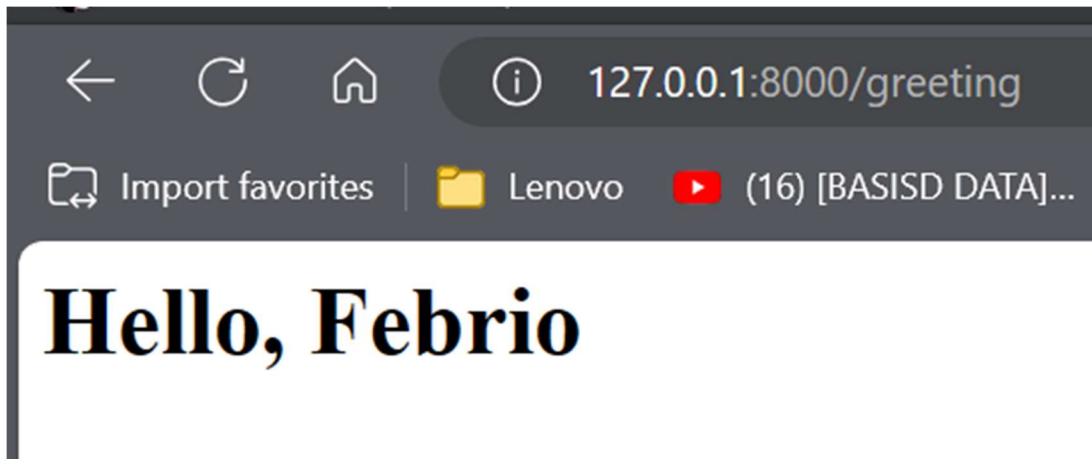
- Pada direktori app/resources/views, buatlah file hello.blade.php

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

- 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' => 'Febrio']);
});
```

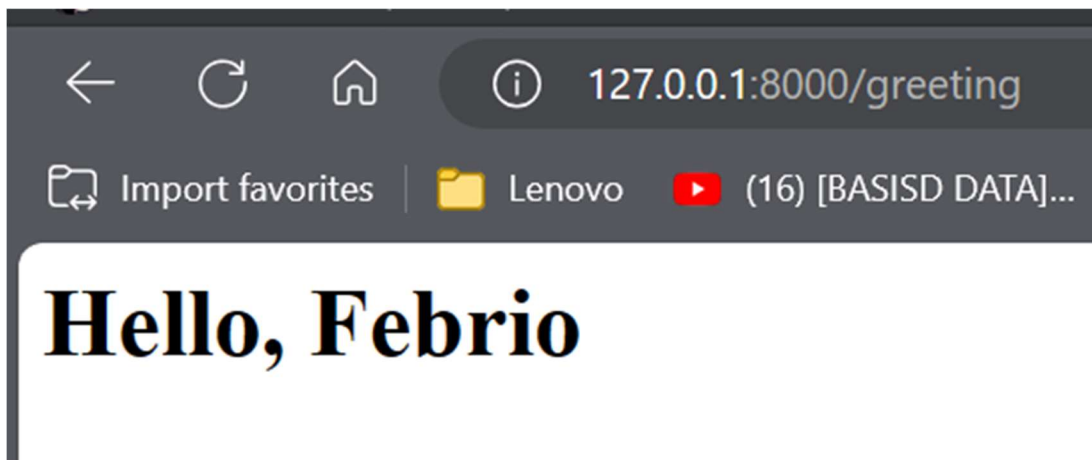
- Jalankan code dengan membuka url localhost/PWL_2024/public/greeting. Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.



- d. Buatlah direktori blog di dalam direktori views.
- e. Pindahkan file hello.blade.php ke dalam direktori blog
- f. Selanjutnya lakukan perubahan pada route.

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

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



Menampilkan View dari Controller

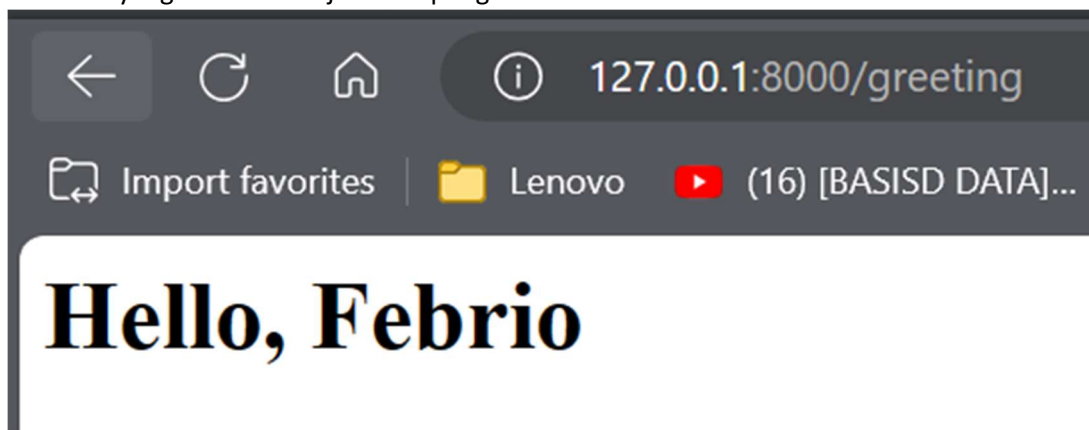
- a. Buka WelcomeController.php dan tambahkan fungsi baru yaitu greeting.

```
1 reference | 0 overrides
public function greeting(){
    return view('blog.hello', ['name' => 'Febrio']);}
```

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

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

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



Meneruskan data ke view

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

```
1 reference | 0 overrides
public function greeting(){
    return view('blog.hello')
        ->with(['name', 'Febrio'])
        ->with('occupation', 'Astronaut');}
}
```

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

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

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

