

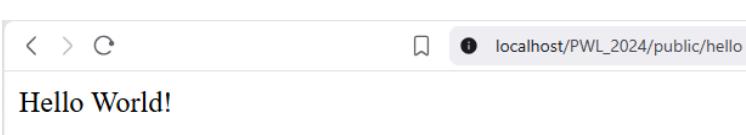


Mata Kuliah : Pemograman Web Lanjut  
Program Studi : D4 – Teknik Informatika  
Semester : 4

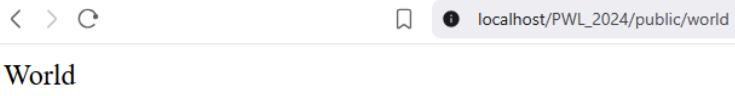
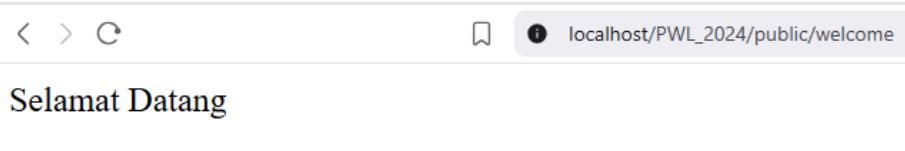
Kelas : 2D  
NIM : 244107020014  
Nama : Siti Nikmatus Sholihah  
Jobsheet Ke- : 2

## Routing

### Basic Routing

Langkah	Keterangan												
1.	<p>Pada bagian ini, kita akan membuat dua buah route dengan ketentuan sebagai berikut.</p> <table border="1"><thead><tr><th>No</th><th>Http Verb</th><th>Url</th><th>Fungsi</th></tr></thead><tbody><tr><td>1</td><td>get</td><td>/hello</td><td>Tampilkan String Hello ke Browser</td></tr><tr><td>2</td><td>get</td><td>/world</td><td>Tampilkan String World ke Browser</td></tr></tbody></table> <p>Kita akan menggunakan project minggu sebelumnya yaitu PWL_2024</p>	No	Http Verb	Url	Fungsi	1	get	/hello	Tampilkan String Hello ke Browser	2	get	/world	Tampilkan String World ke Browser
No	Http Verb	Url	Fungsi										
1	get	/hello	Tampilkan String Hello ke Browser										
2	get	/world	Tampilkan String World ke Browser										
2.	<p>Buka file routes/web.php. Tambahkan sebuah route untuk nomor 1 seperti dibawah ini:</p> <pre>routes &gt; web.php &gt; ... 1 &lt;?php 2 3 use Illuminate\Support\Facades\Route; 4 5 Route::get('/hello', function () { 6     return 'Hello World!'; 7 }); 8</pre>												
3.	<p>Buka browser, tuliskan url untuk memanggil route tersebut: localhost/PWL_2024/public/hello</p>  <p>Dari percobaan yang telah dilakukan , ketika menambahkan kode berikut pada routes/web.php:</p> <pre>Route::get('/hello', function () {     return 'Hello World!';</pre>												



	<p>}); Kemudian mengakses halaman url <a href="localhost/PWL_2024/public/hello">localhost/PWL_2024/public/hello</a> di browser, maka sistem akan menampilkan Teks Hello World!</p>
4.	<p>Untuk membuat route kedua, tambahkan route/world seperti di bawah ini:</p> <pre>routes &gt; web.php &gt; ... 1 &lt;?php 2 3 use Illuminate\Support\Facades\Route; 4 5 Route :: get('/world', function () { 6     return 'World'; 7});</pre>
5.	<p>Buka browser, dan tuliskan url untuk memanggil route tersebut: <a href="localhost/PWL_2024/public/world">localhost/PWL_2024/public/world</a></p>  <p>Dari percobaan yang sudah saya lakukan, ketika url /world diakses, sistem berhasil menampilkan teks ‘World’ sesuai dengan route yang sudah ditentukan pada file routes/web.php</p>
6.	<p>Selanjutnya, cobalah membuat route ‘/’ yang menampilkan pesan ‘Selamat Datang’.</p> <p>Jawab:</p> <pre>routes &gt; web.php &gt; ... 1 &lt;?php 2 3 use Illuminate\Support\Facades\Route; 4 5 Route::get('/welcome', function(){ 6     return 'Selamat Datang'; 7});</pre> <p>url : <a href="localhost/PWL_2024/public/welcome">localhost/PWL_2024/public/welcome</a></p> 
7.	<p>Kemudian buatlah route ‘/about’ yang akan menampilkan NIM dan nama Anda.</p> <p>Jawab:</p>



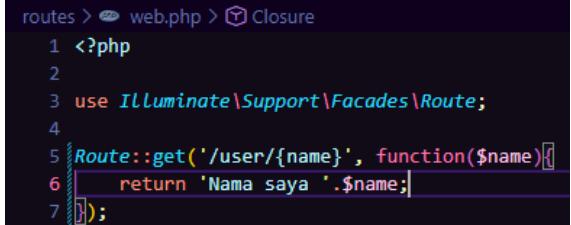
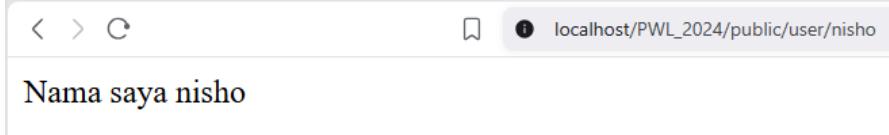
routes > web.php > Closure

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 Route::get('/about', function(){
6     return 'Nama : Siti Nikmatus Sholihah <br> NIM : 244107020014 <br>';
7 });

url : localhost/PWL2024/public/about
```

Nama : Siti Nikmatus Sholihah  
NIM : 244107020014

## Route Parameters

Langkah	Keterangan
1.	Kita akan memanggil route /user/nisho sekaligus mengirimkan parameter berupa nama user \$name seperti kode di bawah ini.   <p>routes &gt; web.php &gt; Closure</p> <pre>1 &lt;?php 2 3 use Illuminate\Support\Facades\Route; 4 5 Route::get('/user/{name}', function(\$name){ 6     return 'Nama saya ' . \$name; 7 });  url : localhost/PWL_2024/public/user/nisho</pre>
2.	Jalankan kode dengan menuliskan URL untuk memanggil route tersebut: <a href="localhost/PWL_2024/public/user/nisho">localhost/PWL_2024/public/user/nisho</a> .   <p>Nama saya nisho</p> <p>Dari percobaan yang sudah saya lakukan, ketika url <a href="localhost/PWL_2024/user/nisho">localhost/PWL_2024/user/nisho</a> diakses, sistem berhasil menampilkan teks ‘Nama saya nisho’, yang berarti {name} adalah sebuah parameter dinamis yang otomatis terisi dengan nilai yang kita tuliskan saat mengakses url.</p>
3.	Selanjutnya, coba tuliskan URL: <a href="localhost/PWL_2024/public/user/">localhost/PWL_2024/public/user/</a>



	<p>Saat mengakses url <a href="localhost/PWL_2024/public/user/">localhost/PWL_2024/public/user/</a>, halaman akan menampilkan pesan error 404   NOT FOUND, karena route tersebut membutuhkan parameter, sedangkan parameter belum diisi.</p>
4.	<p>Suatu route, juga bisa menerima lebih dari 1 parameter seperti kode berikut ini. Route menerima parameter \$postId dan juga \$comment.</p> <pre>routes &gt; web.php &gt; ... 1 &lt;?php 2 3 use Illuminate\Support\Facades\Route; 4 5 Route::get('/posts/{post}/comments/{comment}', function(\$postId, \$commentId){ 6     return 'Pos ke-' . \$postId. " Komentar ke-" . \$commentId; 7});</pre>
5.	<p>Jalankan kode dengan menuliskan URL untuk memanggil route tersebut: <a href="localhost/PWL_2024/public/posts/1/comments/5">localhost/PWL_2024/public/posts/1/comments/5</a></p>
	<p>Dari percobaan yang sudah saya lakukan, ketika url <a href="localhost/PWL_2024/public/post/1/comments/5">localhost/PWL_2024/public/post/1/comments/5</a> diakses, sistem berhasil menampilkan teks ‘Pos ke-1 Komentar ke-5, yang berarti sebuah route bisa memiliki lebih dari 1 parameter dinamis yaitu {post} dan {comment}. function(\$postId, \$commentId) adalah fungsi yang menerima dua parameter yakni \$postId dan \$commentId yang akan diisi dengan nilai dari parameter {post} dan {comment} saat url diakses.</p>
6.	<p>Kemudian buatlah route /articles/{id} yang akan menampilkan output “Halaman Artikel dengan ID {id}”, ganti id sesuai dengan input dari url.</p> <p>Jawab:</p>



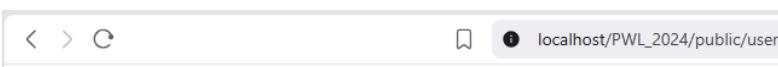
```
routes > web.php > Closure
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 Route::get('/articles/{id}', function($id){
6     return 'Halaman artikel dengan ID '.$id;
7 });

url : localhost/PWL_2024/public/articles/1
```

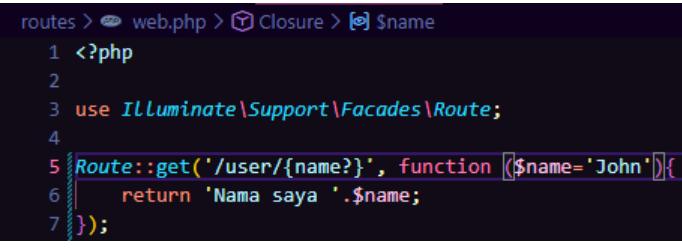


Halaman artikel dengan ID 1

## Optional Parameters

Langkah	Keterangan
1.	Kita akan memanggil route /user sekaligus mengirimkan parameter berupa nama user \$name dimana parameternya bersifat opsional.  <pre>routes &gt; web.php &gt; Closure 1 &lt;?php 2 3 use Illuminate\Support\Facades\Route; 4 5 Route::get('/user/{name?}', function (\$name=null){ 6     return 'Nama saya '.\$name; 7 });  </pre>
2.	Jalankan kode dengan menuliskan URL: <a href="localhost/PWL_2024/public/user/">localhost/PWL_2024/public/user/</a> .   <p>Nama saya</p> <p>Dari percobaan yang sudah saya lakukan, ketika url <a href="localhost/PWL_2024/public/user/">localhost/PWL_2024/public/user/</a> diakses, sistem menampilkan teks 'Nama saya ', yang berarti {name?} adalah sebuah parameter dinamis yang bersifat opsional, ditandai dengan tanda ?. function(\$name=null) adalah fungsi yang menerima parameter \$name yang memiliki nilai default null jika tidak ada nilai yang diisi saat mengakses url.</p>



3.	<p>Selanjutnya tuliskan URL: <a href="localhost/PWL_2024/public/user/nisho">localhost/PWL_2024/public/user/nisho</a>.</p>  <p><b>Nama saya nisho</b></p> <p>Dari percobaan yang saya lakukan, Ketika url <a href="localhost/PWL_2024/public/user/nisho">localhost/PWL_2024/public/user/nisho</a> diakses, maka sistem akan menampilkan teks ‘Nama saya nisho’. Hal ini menunjukkan bahwa nilai \$name diisi berdasarkan parameter yang dikirim melalui url.</p>
4.	<p>Ubah kode pada route /user menjadi seperti di bawah ini.</p>  <pre>routes &gt; web.php &gt; Closure &gt; \$name 1 &lt;?php 2 3 use Illuminate\Support\Facades\Route; 4 5 Route::get('/user/{name?}', function (\$name = 'John') { 6     return 'Nama saya ' . \$name; 7 });  </pre>
5.	<p>Jalankan kode dengan menuliskan URL: <a href="localhost/PWL_2024/public/user/">localhost/PWL_2024/public/user/</a>.</p> <p>Jawab:</p> <p>Dari percobaan yang sudah saya lakukan, ketika url <a href="localhost/PWL_2024/public/user/">localhost/PWL_2024/public/user/</a> diakses, sistem menampilkan teks ‘Nama saya John’, karena pada kode pada file web.php nilai dari parameter \$name diisi dengan ‘John’.</p>

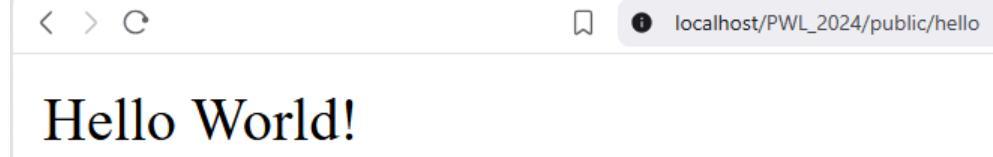


## Controller

### Membuat Controller

Langkah	Keterangan
1.	<p>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.</p> <pre>D:\laragon\www\PWL_2024&gt;php artisan make:controller WelcomeController INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\WelcomeController.php] created successfully.</pre>
2.	<p>Buka file pada app/Http/Controllers/WelcomeController.php. Struktur pada controller dapat digambarkan sebagai berikut:</p> <pre>app &gt; Http &gt; Controllers &gt; WelcomeController.php &gt; ... 1 &lt;?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class WelcomeController extends Controller 8 { 9     // 10 }</pre>
3.	<p>Untuk mendefinisikan action, silahkan tambahkan function dengan access public. Sehingga controller di atas menjadi sebagai berikut:</p> <pre>app &gt; Http &gt; Controllers &gt; WelcomeController.php &gt; WelcomeController &gt; hello 1 &lt;?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class WelcomeController extends Controller 8 { 9     public function hello(){} 10    return 'Hello World!'; 11 } 12 }</pre>
4.	<p>Setelah sebuah controller telah didefinisikan action, kita dapat menambahkan controller tersebut pada route. Ubah route /hello menjadi seperti berikut:</p> <pre>Route::get('/hello', [WelcomeController::class, 'hello']);</pre>



5.	<p>Buka browser, tuliskan URL untuk memanggil route tersebut:</p> <p><code>localhost/PWL_2024/public/hello</code></p>  <p><b>Hello World!</b></p> <p>Dari percobaan yang sudah saya lakukan, ketika url <code>localhost/PWL_2024/public/hello</code> diakses, laravel memproses request berdasarkan route yang telah didefinisikan pada file web.php. Route tersebut memanggil method <code>hello()</code> pada WelcomeController yang kemudian mengembalikan teks “Hello World!” sebagai response ke browser.</p>
6.	<p>Modifikasi hasil pada praktikum poin 2 (Routing) dengan konsep controller. Pindahkan logika eksekusi ke dalam controller dengan nama PageController.</p> <p>a. index</p>  <pre>app &gt; Http &gt; Controllers &gt; PageController.php &gt; ... 1 &lt;?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class PageController extends Controller 8 { 9     public function index(){ 10         return "Selamat Datang"; 11     } 12 } 13 </pre> <p><code>Route::get('/', [PageController::class, 'index']);</code></p> <p>url : <code>localhost/PWL_2024/public/</code></p>  <p><b>Selamat Datang</b></p> <p>b. about</p>



```
public function about(){
    return "Nama : Siti Nikmatus Sholihah <br> NIM : 244107020014 <br>";
}
```

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

url: [localhost/PWL\\_2024/public/about](http://localhost/PWL_2024/public/about)



Nama : Siti Nikmatus Sholihah  
NIM : 244107020014

c. article

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

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

url : [localhost/PWL\\_2024/public/articles/1](http://localhost/PWL_2024/public/articles/1)



Halaman artikel dengan ID 1

7. 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**

```
D:\laragon\www\PWL_2024>php artisan make:controller HomeController
INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\HomeController.php] created successfully.
```



```
app > Http > Controllers > HomeController.php > HomeController > index
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class HomeController extends Controller
8 {
9     public function index(){
10         return "Selamat Datang";
11     }
12 }
```

routing :

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

url : localhost/PWL\_2024/public/



### b. AboutController.php

```
D:\laragon\www\PWL_2024>php artisan make:controller AboutController
INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\AboutController.php] created successfully.
```

```
app > Http > Controllers > AboutController.php > AboutController > about
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class AboutController extends Controller
8 {
9     public function about(){
10         return "Nama : Siti Nikmatus Sholihah <br> NIM : 244107020014 <br>";
11     }
12 }
```

routing :

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



url : [localhost/PWL\\_2024/public/about](http://localhost/PWL_2024/public/about)



Nama : Siti Nikmatus Sholihah  
NIM : 244107020014

#### c. ArticleController

```
D:\laragon\www\PWL_2024>php artisan make:controller ArticleController
[INFO] Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\ArticleController.php] created successfully.
```

```
app > Http > Controllers > ArticleController.php > ArticleController > articles
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class ArticleController extends Controller
8 {
9     public function articles($id){
10         return "Halaman artikel dengan ID ".$id;
11     }
12 }
```

routing :

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

url : [localhost/PWL\\_2024/public/articles/1](http://localhost/PWL_2024/public/articles/1)



Halaman artikel dengan ID 1



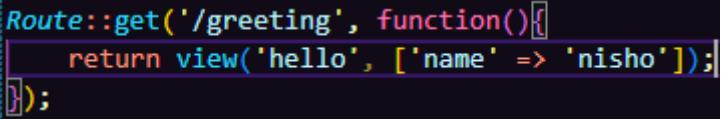
## Resource Controller

Langkah	Keterangan
1.	<p>Untuk membuatnya dilakukan dengan menjalankan perintah berikut ini di terminal.</p> <pre>D:\laragon\www\PWL_2024&gt;php artisan make:controller PhotoController --resource INFO Controller [D:\laragon\www\PWL_2024\app\Http\Controllers\PhotoController.php] created successfully.</pre> <p>Perintah ini akan generate sebuah controller dengan nama PhotoController yang berisi method standar untuk proses CRUD.</p>
2.	<p>Setelah controller berhasil degenerate, selanjutnya harus dibuatkan route agar dapat terhubung dengan frontend. Tambahkan kode program berikut pada file web.php.</p> <pre>use App\Http\Controllers\PhotoController;  Route::resource('photos', PhotoController::class);</pre>
3.	<p>Jalankan cek list route (php artisan route:list) akan dihasilkan route berikut ini.</p> <pre>D:\laragon\www\PWL_2024&gt;php artisan route:list GET HEAD / ..... HomeController@index POST _ignition/execute-solution ignition.executeSolution &gt; Spatie\LaravelIgnition &gt; ExecuteSolutionController GET HEAD _ignition/health-check ..... ignition.healthCheck &gt; Spatie\LaravelIgnition &gt; HealthCheckController POST _ignition/update-config ..... ignition.updateConfig &gt; Spatie\LaravelIgnition &gt; UpdateConfigController GET HEAD about ..... AboutController@about GET HEAD api/user ..... ApiController@user GET HEAD articles/{id} ..... ArticleController@articles GET HEAD hello ..... WelcomeController@hello GET HEAD photos ..... photos.index &gt; PhotoController@index POST photos ..... photos.store &gt; PhotoController@store GET HEAD photos/create ..... photos.create &gt; PhotoController@create GET HEAD photos/{photo} ..... photos.show &gt; PhotoController@show PUT PATCH photos/{photo} ..... photos.update &gt; PhotoController@update DELETE photos/{photo} ..... photos.destroy &gt; PhotoController@destroy GET HEAD photos/{photo}/edit ..... photos.edit &gt; PhotoController@edit GET HEAD posts/{post}/comments/{comment} ..... comments.show &gt; CommentController@show GET HEAD sanctum/csrf-cookie ..... sanctum.csrf-cookie &gt; Laravel\Sanctum &gt; CsrfCookieController@show GET HEAD user/{name?} ..... UserController@name GET HEAD user/{name} ..... UserController@name GET HEAD world ..... WorldController@world  Showing [20] routes</pre>
4.	<p>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.</p> <pre>use App\Http\Controllers\PhotoController;  Route::resource('photos', PhotoController::class) -&gt; only(['index', 'show']);  Route::resource('photos', PhotoController::class) -&gt; except(['create', 'store', 'update', 'destroy']);</pre>



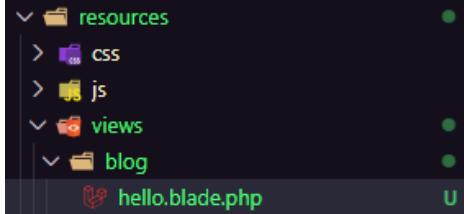
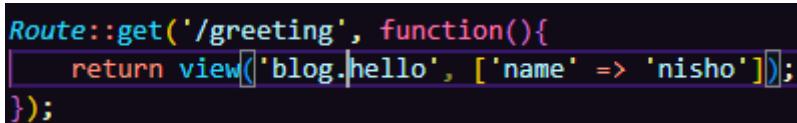
## View

### Membuat View

Langkah	Keterangan
1.	Pada direktori app/resources/views, buatlah file hello.blade.php. 
2.	View tersebut dapat dijalankan melalui Routing, dimana route akan memanggil View sesuai dengan nama file tanpa ‘blade.php’. 
3.	Jalankan code dengan membuka url <a href="http://localhost/PWL_2024/public/greeting">localhost/PWL_2024/public/greeting</a>  <p>Dari hasil percobaan yang saya lakukan, ketika mengakses url <a href="http://localhost/PWL_2024/public/greeting">localhost/PWL_2024/public/greeting</a> diakses, sistem menampilkan teks “Hello, nisho”. Hal ini karena route memanggil view “Hello” dan mengirimkan variabel name dengan nilai “nisho”, yang kemudian ditampilkan menggunakan sintaks Blade{{ \$name }}.</p>

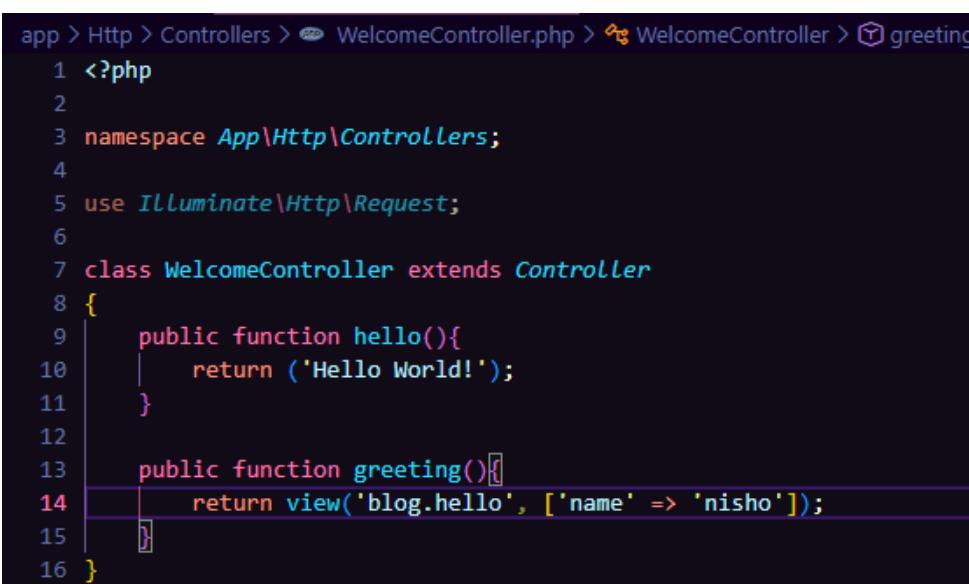
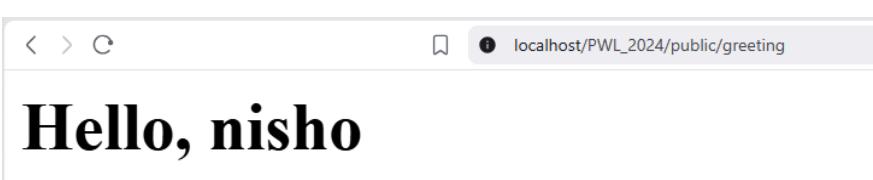


### View dalam direktori

Langkah	Keterangan
1.	Buatlah direktori blog di dalam direktori views.
2.	Pindahkan file hello.blade.php ke dalam direktori blog. 
3.	Selanjutnya lakukan perubahan pada route. 
4.	Jalankan code dengan membuka url <a href="http://localhost/PWL_2024/public/greeting">localhost/PWL_2024/public/greeting</a> . Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.  <p>Dari percobaan yang saya lakukan, setelah file hello.blade.php dipindahkan ke folder blog, route diubah menjadi view('blog.hello'). Saat url diakses, halaman tetap menampilkan ‘Hello, nisho’. Hal ini menunjukkan bahwa Laravel menggunakan tanda titik(.) untuk mengakses view di dalam subdirektori.</p>



## Menampilkan View dari Controller

Langkah	Keterangan
1.	Buka WelcomeController.php dan tambahkan fungsi baru yaitu greeting.  <pre>app &gt; Http &gt; Controllers &gt; WelcomeController.php &gt; WelcomeController &gt; greeting 1 &lt;?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class WelcomeController extends Controller 8 { 9     public function hello(){ 10         return ('Hello World!'); 11     } 12 13     public function greeting(){ 14         return view('blog.hello', ['name' =&gt; 'nisho']); 15     } 16 }</pre>
2.	Ubah route /greeting dan arahkan ke WelcomeController pada fungsi greeting 
3.	Jalankan code dengan membuka url <a href="http://localhost/PWL_2024/public/greeting">localhost/PWL_2024/public/greeting</a> . Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.  <p>localhost/PWL_2024/public/greeting</p> <h1>Hello, nisho</h1> <p>Dari percobaan yang saya lakukan, setelah method greeting() ditambahkan pada WelcomeController dan route /greeting diarahkan ke WelcomeController::class, 'greeting', halaman tetap menampilkan "Hello, nisho" saat url diakses. Hal ini menunjukkan bahwa view dapat dipanggil melalui controller, sehingga alur program menjadi lebih terstruktur sesuai konsep MVC, di mana route mengarah ke controller dan controller menampilkan view.</p>



## Meneruskan data ke view

Langkah	Keterangan
1.	Buka WelcomeController.php dan tambahkan ubah fungsi greeting
	<pre>app &gt; Http &gt; Controllers &gt; WelcomeController.php &gt; WelcomeController 1 &lt;?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class WelcomeController extends Controller 8 { 9     public function hello(){ 10         return ('Hello World!'); 11     } 12 13     public function greeting(){ 14         return view('blog.hello') 15             -&gt;with('name', 'Nisho') 16             -&gt;with('occupation', 'Mahasiswa'); 17     } 18 }</pre>
2.	Ubah hello.blade.php agar dapat menampilkan dua parameter
	<pre>resources &gt; views &gt; blog &gt; hello.blade.php &gt; html &gt; body &gt; h1 1 &lt;html&gt; 2     &lt;body&gt; 3         &lt;h1&gt;Hello, {{\$name}}&lt;/h1&gt; 4         &lt;h1&gt;You are {{\$occupation}}&lt;/h1&gt; 5     &lt;/body&gt; 6 &lt;/html&gt;</pre>
3.	Jalankan code dengan membuka url <a href="http://localhost/PWL_2024/public/greeting">localhost/PWL_2024/public/greeting</a> . Perhatikan halaman yang muncul dan jelaskan pengamatan Anda.
	 <p>The screenshot shows a web browser window. The address bar contains the URL "localhost/PWL_2024/public/greeting". The main content area of the browser displays two lines of text: "Hello, Nisho" and "You are Mahasiswa", both in large, bold, black font.</p> <p>Dari percobaan yang sudah saya lakukan, setelah fungsi greeting() diubah untuk mengirim dua data menggunakan with(), yaitu name dan occupation, serta file hello.blade.php diperbarui untuk menampilkan kedua variabel tersebut, halaman menampilkan "Hello, Nisho" dan "You are Mahasiswa" saat url diakses. Hal ini</p>



	menunjukkan bahwa controller dapat meneruskan lebih dari satu data ke view, dan Blade dapat menampilkan beberapa variabel yang dikirimkan tersebut dengan benar.
--	--