



Routing dan Controller

Versi Laravel

- Versi Laravel 10 telah rilis, pastikan anda menggunakan versi yang sama agar nantinya bisa melakukan kolaborasi dengan teman saat project akhir

- Instalasi Laravel 9

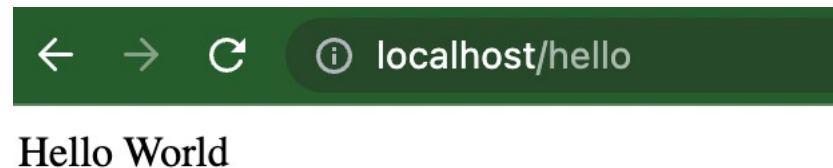
```
composer create-project --prefer-dist  
laravel/laravel="9.*" <project>
```

Pengantar Routing

- Rute = Jalur
- Routing pada Laravel : jalur URL yang bisa diakses oleh pengguna aplikasi dan ke mana jalur itu diproses

contoh

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



- Pengguna mengakses URL akan mengeksekusi callback function yang menampilkan pesan 'Hello World'
- Akan tetapi penggunaan callback function jarang sekali dipakai dalam pembuatan aplikasi sesungguhnya, karena untuk logika yang kompleks menjadikan kode susah di-*maintenance*

Default Route Files

▼ routes

- api.php
- channels.php
- console.php
- web.php

- `routes/web.php` digunakan untuk web interface
- `routes/api.php` digunakan untuk membuat routing API, membuat core service API dengan menggunakan Laravel
- `routes/channels.php` digunakan membuat routing yang bersifat broadcasting event melalui websocket, seperti notification
- `routes/console.php` digunakan untuk membuat routing command yang berjalan di terminal
- Secara umum aplikasi yang dibuat cukup dengan `routes/web.php` dan `routes/api.php`. Bahkan jika aplikasi tidak perlu menyediakan API hanya menggunakan `routes/web.php` saja.

Syntax Dasar Routing

```
Route::verb("/path", callback);
```

- **verb** yang merupakan HTTP verb, umumnya terdiri dari get, post, put, delete, options, patch
 - **Route::get**(\$uri, \$callback); → retrieve data from the server
 - **Route::post**(\$uri, \$callback); → sends data to the server → write new data
 - **Route::put**(\$uri, \$callback); → update an existing resource → write and update
 - **Route::patch**(\$uri, \$callback); → request body only needs to contain the specific changes to the resource
 - **Route::delete**(\$uri, \$callback); → delete a resource specified by its URI → delete data
 - **Route::options**(\$uri, \$callback); → describe communication options for the target resource
- **path** yang berupa URL setelah nama domain aplikasi yang diakses oleh pengguna
- **callback** yang dapat berupa callback function atau controller action yang menjalankan logika ketika path diakses oleh pengguna

Multiple Routes

- **Match Method** : digunakan ketika membutuhkan beberapa route yang berbagi URI yang sama.

```
Route::match(['get', 'post'], '/', function () {  
    // ...  
});
```

Basic Routing

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5
6 Route::get('/hello', function () {
7     return 'Hello';
8 });
9
10
```

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

← → ↻ ⚠ Not Secure | 0.0.0.0/hello/

Hello

Router Method

- Semua http verb dapat dipasang method router
- Endpoint / url router sebaiknya mengikuti best practice berikut ini dimana sebuah resource dapat dilayani dengan fungsi berbeda pada setiap http verb nya

Resource	POST	GET	PUT	DELETE
/mahasiswa	Membuat record mahasiswa baru	Mengambil Daftar Mahasiswa	Update banyak data mahasiswa	Delete banyak data mahasiswa
/mahasiswa/{id}	Error	Tampilkan Data Satu Mahasiswa	Update data mahasiswa jika ada data dengan id yang dikirim	Delete satu data mahasiswa

Router Method

```
Route::get('mahasiswa', function ($id) {
});

Route::post('mahasiswa', function ($id) {
});

Route::put('mahasiswa', function ($id) {
});

Route::delete('mahasiswa', function ($id) {
});

Route::get('mahasiswa/{id}', function ($id) {
});

Route::put('mahasiswa/{id}', function ($id) {
});

Route::delete('mahasiswa/{id}', function ($id) {
})
```

- Perlu diketahui laravel dapat mendukung satu route yang memiliki lebih dari satu http verb atau memiliki semua http verb

Required Parameters

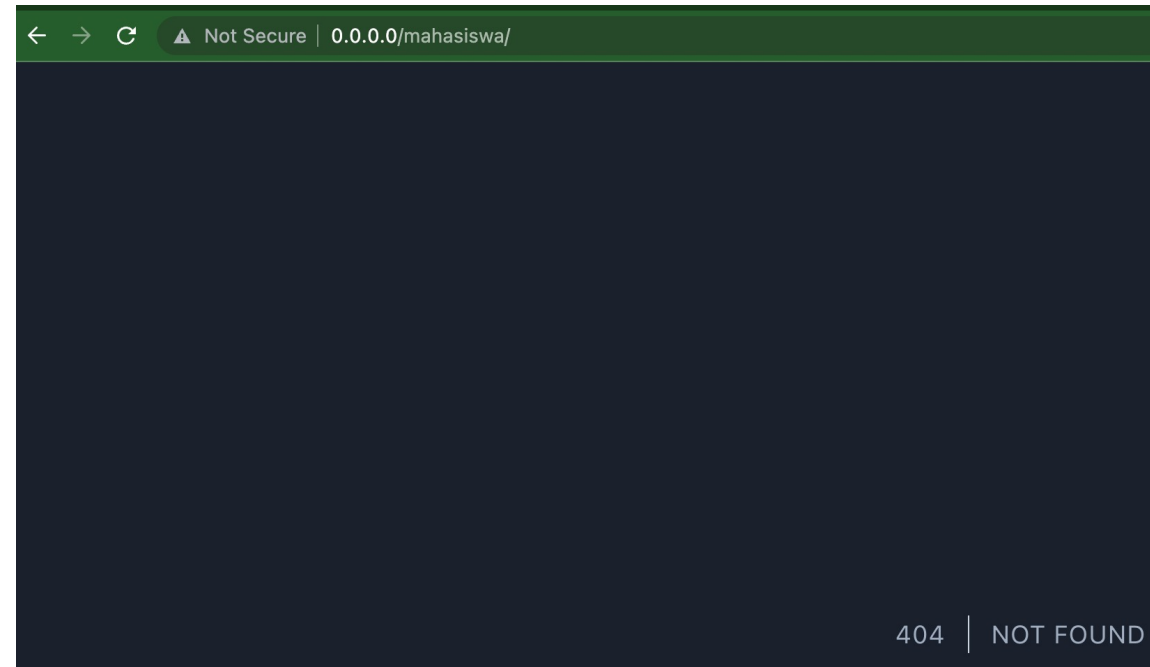
- Terkadang perlu menangkap segmen URI di dalam route. Misalnya, mungkin perlu mengambil ID mahasiswa dari URL.
- Parameter menggunakan {}



```
1 Route::get('mahasiswa/{id}', function ($id) {  
2     return "mahasiswa ke ".$id;  
3 });
```



mahasiswa ke 3



Optional Parameter



```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\WelcomeController;
5
6
7 Route::get('/hello/{page?}', function ($page=1) {
8     return "Hello, halaman ".$page;
9 });
10
11
```

- parameter opsional diberikan tanda ? dan dapat dibuat nilai awal nya pada callback function atau controller function nya



localhost/hello/50

Hello, halaman 50

- Jika diganti post

Symfony\Component\HttpKernel\Exception\MethodNotAllowedHttpException

The GET method is not supported for route hello/50. Supported methods: POST.

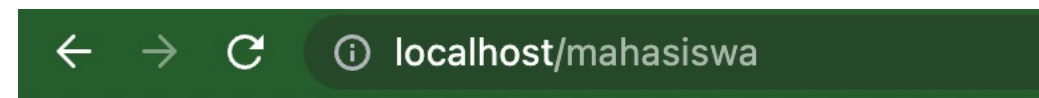
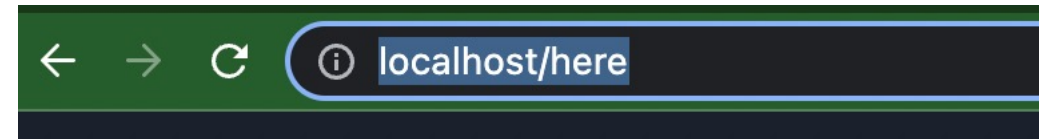
Dependency Injection

- Route pada laravel dapat dilakukan dependency injection dimana sebuah route dapat dititipkan dependency yang dibutuhkan
- Laravel akan meresolve dependency ini secara otomatis dan ikut dikirim di callback function yang dimiliki route

```
4 use Illuminate\Http\Request;
5
6 Route::get('/mahasiswa/{id}', function (Request $request, string $id) {
7     return 'mahasiswa '.$id;
8 });
```

Redirect Routes

- redirect sering menggunakan pada kasus CRUD



mahasiswa

View Routes

- Laravel juga menyediakan sebuah route khusus yang memudahkan anda membuat sebuah routes tanpa menggunakan controller atau callback function routes ini langsung menerima input berupa url dan mengembalikan view / tampilan.

```
Route::view('/welcome', 'welcome');  
Route::view('/welcome', 'welcome', ['name' => 'Taylor']);
```

Route Name

- Sebuah route bisa diberi nama custom untuk mempermudah pembuatan url pada saat melakukan coding

```
Route::get('/mahasiswa/profile', function () {  
    return "Profil mahasiswa ";  
})->name('profile');
```

Route dengan Controller

- Penggunaan callback function jarang sekali dipakai dalam pembuatan aplikasi sesungguhnya, karena untuk logika yang kompleks menjadikan kode susah di-*maintenance*
- **SOLUSI : Controller**
- Harus membuat sebuah controller yang berisi method yang anda sebutkan di router

Controller

- Controller digunakan untuk mengorganisasi logika aplikasi menjadi lebih terstruktur.
- Logika action aplikasi yang masih ada kaitan dapat dikumpulkan dalam satu kelas Controller. Atau sebuah Controller dapat juga hanya berisi satu buah action. Controller pada Laravel disimpan dalam folder `app/Http/Controllers`.
- Membuat Controller

```
php artisan make:controller WelcomeController
```

Controller

```
● Mamluatuls-MacBook-Air:PWLP2 mamluatulhaniah$ php artisan make:controller AboutController  
  INFO Controller [app/Http/Controllers/AboutController.php] created successfully.  
○ Mamluatuls-MacBook-Air:PWLP2 mamluatulhaniah$
```

```
app > Http > Controllers > AboutController.php > ...  
1  <?php  
2  
3  namespace App\Http\Controllers;  
4  
5  use Illuminate\Http\Request;  
6  
7  1 reference | 0 implementations  
8  class AboutController extends Controller  
9  {  
10     //  
11 }
```

Route dengan Controller

Routing

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\WelcomeController;
5
6 Route::get('/word', [WelcomeController::class, 'word']);
7
```

← → ↻ ⚠ Not Secure | 0.0.0.0/word/

Hello World

Controller

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class WelcomeController extends Controller
8 {
9     //
10    public function word() {
11        return 'Hello World';
12    }
13
14 }
15
```

