

MODUL 2

ROUTES DAN CONTROLLER



CAPAIAN PEMBELAJARAN

1. Bisa membuat route dengan fungsi
2. Bisa membuat route dan controller



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Sistem Operasi Linux/Window 10
2. PHP 5.x, diutamakan versi 7.x keatas
3. Composer
4. XAMPP (PHP, Apache, MySQL)
5. Text Editor Notepad++ atau Visual Studio Code



DASAR TEORI

Route bagian berfungsi menangani *request* dari *user* ke aplikasi kemudian mengarahkan aplikasi untuk memanggil halaman/*resource* tertentu. Di Laravel, semua perintah route terdapat di dalam sebuah file di `app/http/routes.php`.

Route merupakan salah satu fitur yang ada pada framework Laravel untuk mengatur setiap permintaan dari user. Jadi setiap URI yang akan diakses wajib didaftarkan pada route. Tujuannya adalah memberi batasan view mana saja yang bisa diakses serta parameter apa saja yang disertakan.

Route juga menyediakan *gateway* ke dalam aplikasi. Kata kerja HTTP dapat digunakan untuk menentukan tindakan yang harus dilakukan oleh route saat dipanggil. Misalnya, kata kerja GET http digunakan untuk mengambil sumber daya ketika kata kerja POST http digunakan untuk membuat sumber daya. Route lain yang umum digunakan adalah DELETE untuk menghapus sumber daya dan PUT untuk memperbarui sumber daya yang ada.

Rute di Laravel 5.3 ke atas terletak di direktori routes. Laravel memiliki tiga (3) kategori yaitu :

- `api.php` - digunakan untuk menentukan route untuk *Application Programming Interface (APIs)*.
- `console.php` - digunakan untuk menentukan route untuk *command line*
- `web.php` - digunakan untuk menentukan rute untuk aplikasi web.

Untuk memahami cara kerja route sederhana ditunjukkan seperti pada gambar 2.1.



Gambar 2.1 Cara kerja route



PRAKTIK

1. Copykan hasil praktikum pertemuan 1, ke folder htdocs
2. Route menggunakan fungsi, buka file /routes/web.php, tambahkan seperti kode program berikut;

File web.php

```
<?php
....
Route::get('webKu/', function () {
    return "Halo Apa kabar.....?";
});
?>
```

Akhir File web.php

Uji dengan perintah <http://localhost:8000/webKu/> amati hasilnya

3. Menggunakan variabel \$logic sebagai parameter kedua. Misalnya, cuplikan di atas juga bisa ditulis seperti berikut:

//route/web.php

```
$logic = function()
{
    return 'STMIK AKAKOM Yogyakarta.....!';
};
```

```
Route::get('webKu1', $logic);
```

Uji dengan perintah <http://localhost:8000/webKu1/> amati hasilnya

4. Route menggunakan parameter, parameter bisa digunakan untuk memasukkan **placeholder** ke dalam definisi route, parameter yang disertakan dalam bentuk variabel yang dituliskan dengan cara `{$nama_variabel}`.

Tambahkan file `web.php` seperti berikut:

```
Route::get('/buku/{judul}', function($judul)
{
    return "Buku <b>{$judul}</b> adalah termasuk buku komputer.";
});
```

Uji dengan perintah <http://localhost:8000/buku/Buku PHP>, amati hasilnya

5. Menghapus persyaratan untuk route indeks buku dengan menggunakan parameter opsional. Parameter ini bisa dibuat secara opsional dengan menambahkan tanda tanya (?) di akhir nama variabel.

```
Route::get('/buku/{judul?}', function($judul=null)
{
    if ($judul == null) return 'Buku tak berjudul.';
    return "Buku <b>{$judul}</b> adalah termasuk buku komputer.";
});
```

Untuk menguji pertama jalankan url <http://localhost:8000/buku> dan kedua <http://localhost:8000/buku/Master Yii>,

6. **Route menggunakan file views**, View adalah file yang diberi nama menggunakan ekstensi `.php` dan biasanya diletakkan di dalam direktori `app/views`.

Buatlah file `pepatah.php` simpan ke `resources/views/..`, seperti berikut:

File `pepatah.php`

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Coba laravel!</title>
    </head>
    <body>
        <p>Sedikit demi sedikit lama-lama menjadi bukit</p>
    </body>
```

</html>

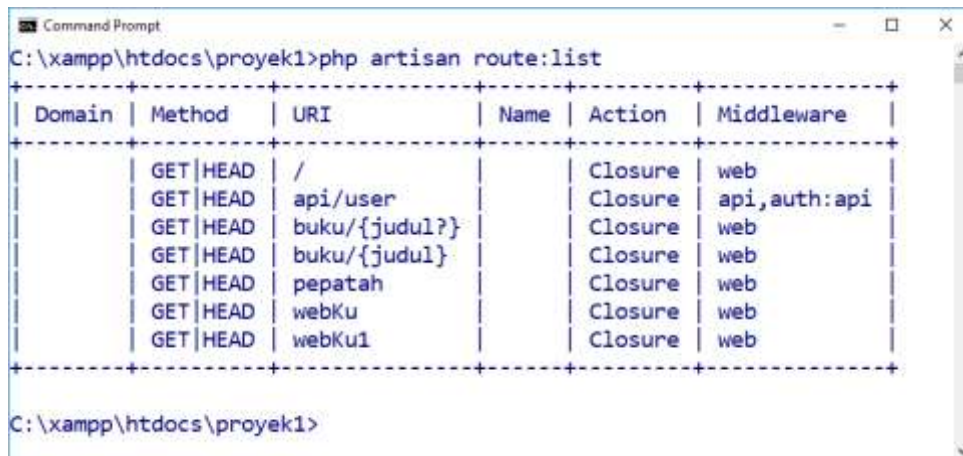
akhir file pepatah.php

Kemudian tambahkan file routes/web.php seperti berikut:

```
Route::get('/pepatah', function()  
{  
    return View::make('pepatah');  
});
```

Untuk menguji jalankan url <http://localhost:8000/pepatah> amati hasilnya

7. Melihat daftar route menggunakan artisan command seperti berikut:

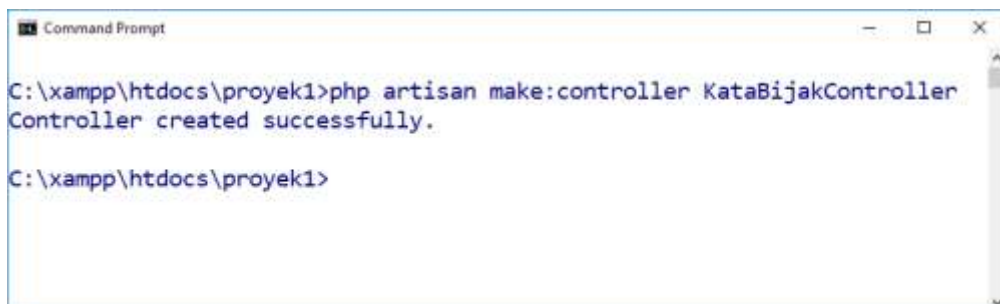


```
Command Prompt  
C:\xampp\htdocs\proyek1>php artisan route:list
```

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	api/user		Closure	api,auth:api
	GET HEAD	buku/{judul?}		Closure	web
	GET HEAD	buku/{judul}		Closure	web
	GET HEAD	pepatah		Closure	web
	GET HEAD	webKu		Closure	web
	GET HEAD	webKu1		Closure	web

```
C:\xampp\htdocs\proyek1>
```

8. Route menggunakan Controller, gunakan perintah artisan seperti berikut:



```
Command Prompt  
C:\xampp\htdocs\proyek1>php artisan make:controller KataBijakController  
Controller created successfully.  
C:\xampp\htdocs\proyek1>
```

9. Buka file app/Http/Controller/KataBijakController.php, dan tambahkan seperti berikut:

File KataBijakControlle.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class KataBijakController extends Controller
{
    public function kata()
    {
        echo "Rajin Pangkal Pandai";
    }
}

```

Akhir File KataBijakControlle.php

10. Untuk memanggil controller tambahkan file web.php seperti program berikut:

```

....
Route::get('kata-bijak/kata', 'KataBijakController@kata');
...

```

Untuk menguji jalankan url <http://localhost:8000/kata-bijak/kata> amati hasilnya

Jalankan *command line* php artisan route:list, amati hasilnya

Jika web.php ditambahkan seperti berikut:

```

Route::get('kata-bijak/pepatah', 'KataBijakController@kataPepatah')
->name('kata-bijak.pepatah');

```

Jalankan *command line* php artisan route:list, apa bedanya dengan sebelumnya?



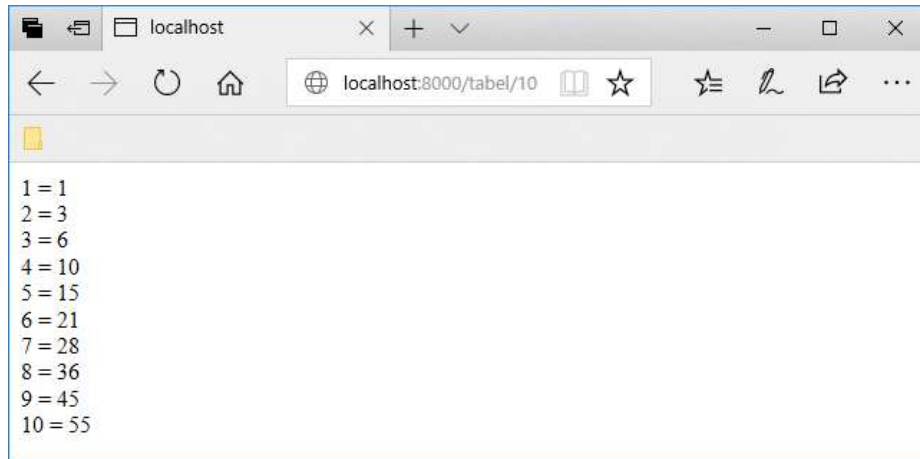
LATIHAN

1. Buatlah route untuk menampilkan teks

Orang menamakan cinta sebagai sesuatu hal yang imajiner. Nyatanya, hal imajiner ini justru membawa Anda kepada kehidupan yang baik. Dengan cinta, Anda bisa merasakan simpati dan empati, dan membuat **hidup bahagia**. Hidup bersama cinta dipandu dengan ilmu pengetahuan akan membuat hidup Anda seimbang.

2. Buatlah route dengan memanggil file views coba.php dengan isi teks di atas.
3. Buatlah route dengan fungsi parameter { n } untuk $1+2+3+4+5+...+n$

Jika memanggil url <http://localhost:8000/tabel/10>, 10 adalah nilai n, maka hasil keluaran seperti berikut



4. Buatlah route menggunakan GET dan menambahkan parameter {\$nilai}, nilai akan diproses oleh fungsi `function ($nilai)`, kuluaran \$nilai pangkat 2, dengan pemanggilan <http://localhost/nilai/2> hasilnya 4



TUGAS

1. Buatlah route yang menggunakan perintah :
 - a) GET,
 - b) POST,
 - c) PUT,
 - d) DELETE



REFERENSI

1. <https://www.w3schools.in/laravel-tutorial/routing/>
2. Cheers, KodeBlog Team, Laravel 5 Routing <https://kode-blog.io/laravel-5-routing>