# Jurus 6:

# Membuat Controller

Pertama-tama, masih tetap dengan jendela command, jalankan perintah

php artisan make:controller HomeController

Perintah tersebut akan membuat sebuah file controller baru yang bernama HomeController. Jika sudah berhasil, maka file controller tersebut dapat kita lihat di direktori **app/Http/Controllers/**. Jika dibuka, maka kita akan menemukan sebuah class kosong yang dapat kita isi dengan sembarang method.

Sebagai contoh, saya membuat sebuah method *index* dengan isi echo untuk menampilkan teks.

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller

{

//

public function index(){

echo 'Hai haiii.. Saya sedang belajar Laravel';

}

}

Setelah membuat controller tersebut, di laravel kita WAJIB mendefinisikan route untuk menentukan alamat pemanggilan dan baris kode apa yang akan dipanggil. Route dapat kita buat di  /routes/web.php. Isi route default bisa kita hapus, dan diganti menjadi seperti ini :

<?php

#Cara baca : memanggil root (/) = memanggil method index di HomeController

Route::get('/', 'HomeController@index');

Kalau sudah, maka hasilnya adalah seperti berikut ini

Untuk menampilkan sebuah tampilan/view, kita dapat mengaturnya dengan membuat sebuah view template. Tempat melatakkan view ada di direktori resource/views/. Sekarang kita akan membuat sebuah file bernama home.blade.php. "blade" itu sendiri merupakan ekstensi khusus untuk menjalankan template engine bawaan Laravel sehingga kita dapat membuat tampilan dengan sangat mudah. Tapi topik ini akan kita bahas lebih detail di lain waktu~

File home.blade.php tersebut bisa kita isi dengan teks apa saja, seperti misalnya dibawah ini :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Contoh File Template</title>

</head>

<body>

<h1>Halaman Home</h1>

</body>

</html>

Kalau view sudah dibuat, kita akan memanggil view tersebut melalui controller. Laravel menyediakan fungsi view() untuk menampilkannya melalui controller. Sekarang kita kembali ke HomeController, dan hapus baris yang berisi echo dan diganti menjadi

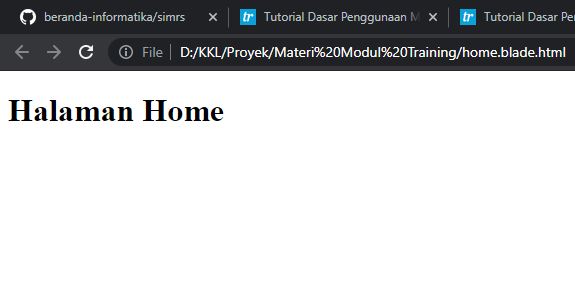
...

//memanggil view yang bernama home.blade.php

return view('home');

...

Hasilnya adalah seperti ini :



Seperti framework lainnya, kita bisa mengirim parameter tambahan ke view tersebut. Cara mengirim parameter adalah dengan menggunakan method tambahan ->with([...]) yang berisi array parameter yang ingin dikirim. Sebagai contoh, return view('home') tadi dapat kita lengkapi dengan parameter seperti ini :

....

return view('home')->with([

'title' => 'Contoh Judul Halaman',

'paragraf' => 'Lorem ipsum dolor sit amet, consectetur adipisicing elit. Id, dignissimos!'

]);

....

Contoh diatas akan mengirimkan parameter bernama title dan paragraf ke view "home", dan parameter tersebut dapat diakses dengan nama variabel *$title*dan *$paragraf*. Untuk melihat hasilnya, sekarang file View home.blade.php bisa kita update menjadi :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Contoh File Template</title>

</head>

<body>

<h1>{{ $title }}</h1>

<p>

{!! $paragraf !!}

</p>

</body>

</html>

Kita melihat ada kurung yang modelnya sedikit berbeda diatas. ada kurung {{...}}, dan juga kurung {!! ... !!}. Pada dasarnya kedua kurung tersebut gunanya sama-sama menampilkan variabel, akan tetapi bedanya kurung {{ ... }} akan mengescape karakter yang dimunculkan, sedangkan kurung {!! ... !!} menampilkan data mentah-mentah tanpa escaping.