

Jurus 5: Kuasai Tampilan (view) : CSS dan HTML Membuat View (tampilan)

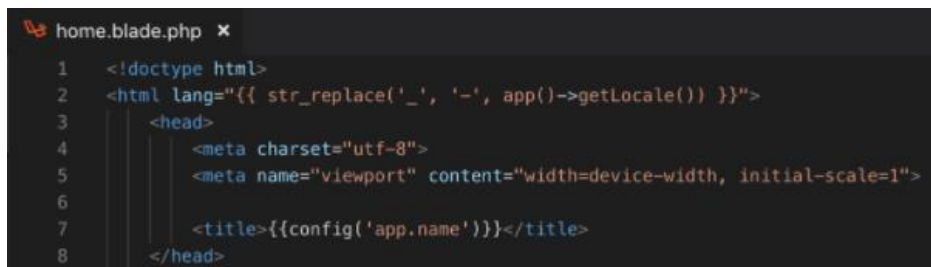
Tutorial Halaman Web dengan Laravel View dan Blade

Pada tutorial ini akan dibuat tampilan dari halaman web dengan memanfaatkan CSS dan framework CSS Bootstrap agar menghasilkan tampilan yang lebih professional.

Parameter Config

Ikuti langkah berikut untuk melakukan konfigurasi sistem pada laravel :

- Buka **.env** file
 - Ganti APP_NAME dengan Portal Berita
 - Akses APP_NAME dari Home dengan menggunakan perintah :
1. `{{config(app.name)}}`
Dengan parameter ke -2 sebagai default value :
 2. `{{config('app.name', 'Portal Berita')}}`
Code yang dibuat akan terlihat seperti gambar berikut ini :



```
home.blade.php x
1 <!doctype html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6
7 <title>{{config('app.name')}}</title>
8 </head>
```

Bootstrap

- o Laravel menggabungkan Bootstrap css dalam project yang di-generate.
- o Bootstrap css tersedia dalam file **/public/assets/bootstrap-3.3.7-dist/css/**
- o Untuk menambahkan Bootstrap dalam web yang dibuat, pada halaman html bagian head dapat ditambahkan:

```
<link href="{{ asset('/assets/bootstrap-3.3.7-dist/css/bootstrap.min.css') }}" rel="stylesheet">
```

Membuat Layout

- Buat sebuah folder layout di dalam folder views
- Buat sebuah file bernama dashboard.blade.php dalam folder tersebut
- Isikan program struktur sebagai berikut:

```
<html>
<head>
  <title>App Name - @yield('title')</title>
</head>
<body>
  @section('sidebar')
    This is the master sidebar.
  @show
  <div class="container">
    @yield('content')
  </div>
</body>
</html>
```

Berikut adalah penjelasan dari potongan program tersebut:

- Blade merupakan pengaturan tampilan dengan menggunakan HTML markup, dengan penambahan beberapa directive dari Laravel.
- Pada contoh sebelumnya terlihat directive pada bagian @section dan @yield
- Directive @section mendefinisikan sebuah bagian (section) dari isi halaman web
- Directive @yield digunakan untuk menampilkan isi dari bagian tersebut.

Membuat Tampilan (view) home

Buat sebuah file bernama [home.blade.php](#) dalam folder view. Adapun isi dari file tersebut adalah sebagai berikut:

```
1  @extends('layout.dashboard')
2  @section('content')
3  <div class="container" >
4
5      <div class="jumbotron">
6          <div class="container">
7              <h1>Hello, world!</h1>
8              <p>Contents ...</p>
9              <p>
10                 <a class="btn btn-primary btn-lg">Learn more</a>
11             </p>
12          </div>
13      </div>
14  </div>
15  @endsection
```

Berikut adalah penjelasan dari potongan program tersebut:

- View yang extend sebuah Blade layout akan mengganti setiap bagian (section) dari layout. Content yang dituliskan pada layout dapat disertakan dengan menggunakan directive @parent dalam section.
- Pada penggunaan @yield, default value dapat diatur atau dikirimkan dengan menuliskannya di argument kedua, contohnya: @yield('section', 'Default Content')

Jadi manakah yang sebaiknya dipergunakan, Yield atau Section?

- Pada contoh sebelumnya terlihat penggunaan @yield dan @section yang serupa.
- Intinya untuk hal sederhana gunakan @yield, sedangkan untuk kebutuhan yang lebih advanced bisa digunakan @section... @show.
- Karena semua yang bisa dilakukan dengan @yield bisa dilakukan dengan @section... @show, namun tidak sebaliknya.

Menampilkan View

Blade views yang telah dibuat dapat ditampilkan dengan perintah view yang dapat diakses secara global.

```
Route::get('/', function () {  
    return view('home');  
});
```

Seperti framework lainnya, laravel 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  
adipiscing 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>
```

Dari skrip diatas terlihat 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.