# Jurus 5:

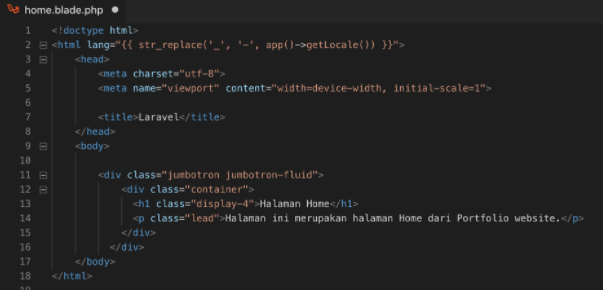
# Kuasai Tampilan (view) : CSS dan HTML

# 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 Boostrap agar menghasilkan tampilan yang lebih professional. Menggunakan Template di View Buat sebuah file home.blade.php dan letakkan dalam folder view. Pada file ini akan dibuat sebuah halaman sederhana, dengan memanfaatkan bootstrap. Misalnya dengan menggunakan Jumbotron pada contoh ini:

[**https://getbootstrap.com/docs/4.2/components/jumbotron/**](https://getbootstrap.com/docs/4.2/components/jumbotron/)

****

**Jombotron Code**

**Manfaat Parameter Config**

**Ikuti langkah berrikut untuk melakukan konfigurasi sistem pada laravel :**

**• Buka.env file**

**• Ganti APP\_NAME dengan Protfolio**

**• Akess APP\_NAME dari Home dengan menggunakan perintah :**

**1. {{config(app.name)}}**

**Dengan parameter ke -2 sebagai default value :**

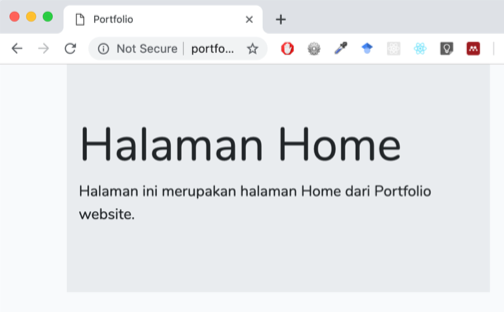
**2. {{config(‘app.name’, ‘Protfolio’)}}**

**Code yang dibuat akan terlihat seperti gambar berikut ini :**

****

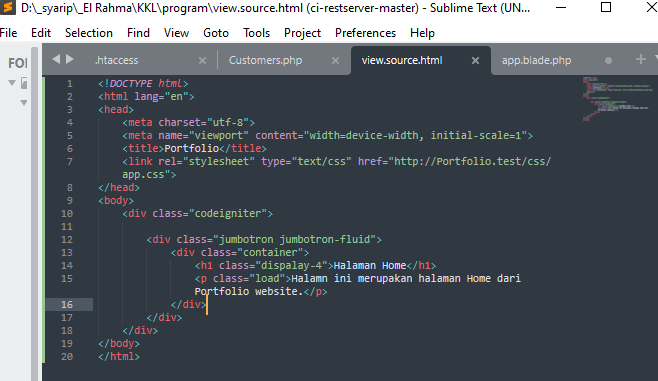
## Pengenalan Bootsrap

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

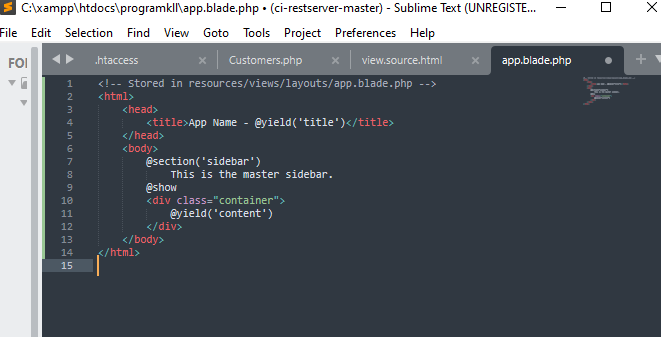
**<link rel="stylesheet" href="{{asset('css/app.css')}}">**

Tampilan Home dengan Boottsrap CSS

Apabila dicermati, Maka kode sumber dari halaman ini akan tampil sebagai berikut :

****

**Mendefinisikan Blade**

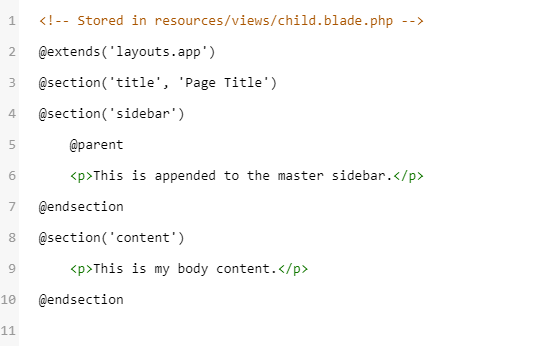
Potongan program berikut dalam file app.blade.php yang disimpan dalam sebuah folder baru dalam folders Views yang diberi nama Layouts.

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.

**Menggunakan Blade**

Berikut adalah contoh penggunaan Blade Template



Berikut adalah penjelasan dari potongan program tersebut:

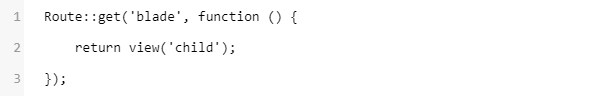
* View yang extend sebuah Blade layout akan mengganti setiap bagian (section) dari layout. Content yang dituliskan pada layout dapat disertakan dengan menggunakan directive @parent dalam section.
* Pada penggunaan @yield, default value dapat diatur atau dikirimkan dengan menuliskannya di argument kedua,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

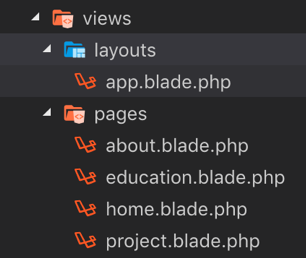
**Menampilkan Blade**

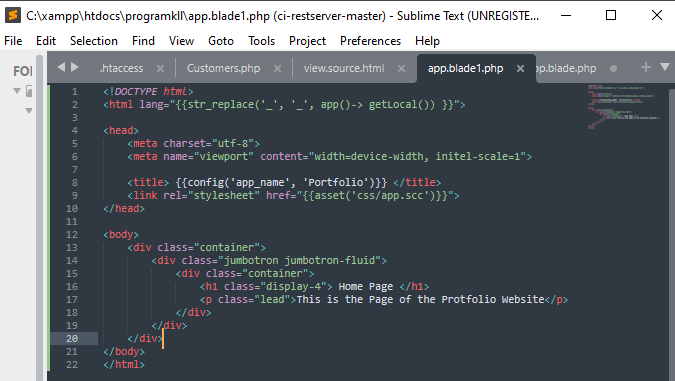
Blade views yang telah dibuat dapat ditampilkan dengan perintah view yang dapat diakses secara global.



**Layout**

* Buat sebuah folder layouts di dalam folder views
* Buat sebuah file bernama app.blade.php dalam folder tersebut
* Copy paste struktur utama dari halaman home.blade.php yang dibuat ke file tersebut.



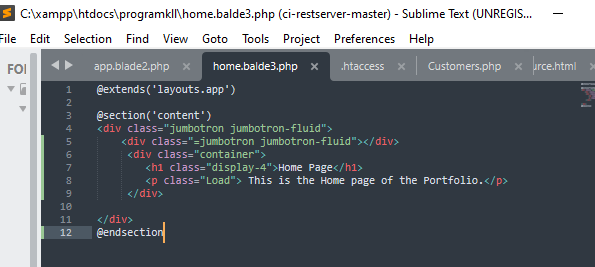


Isi File app.blade.php

**Silahkan mengganti bagian isi dari container dengan code menggunakan @yield seperti tampak pada gambar berikut:**

****

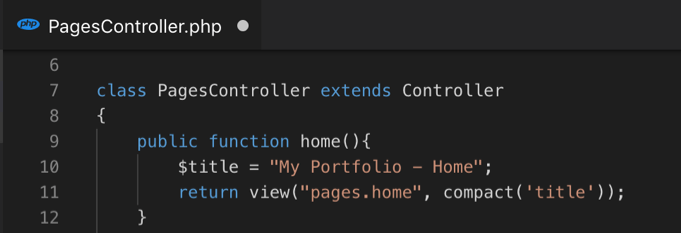
Silahkan mengubah isi dari **home.blade.php** dengan kode seperti terlihat pada gambar :



**Mengirim Value ke Blade**

Berikut adalah beberapa cara untuk mengirimkan Value ke Blade:

* Menggunakan compact(‘namavariabel’)



Blade Mneggunakan Compact

* Menggunakan ->width(‘namavar’, ‘vardikirim’). Parameter pertama adalah nama variabel untuk menerima di dalam blade, parameter kedua adalah variable dari value yang akan dikirimkan.



Balade menggunakan With

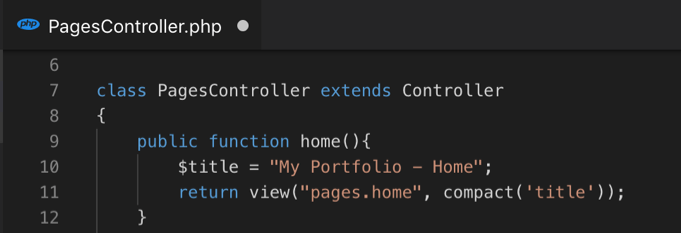
* Menggunakan ->width(‘namavar’, ‘var\_array\_dikirim’). Parameter pertama adalah nama variabel untuk menerima di dalam blade, parameter kedua adalah variable dari value yang akan dikirimkan berupa array.



Blade Menggunakan With dengan 2 Parameter

**Menerima multi Value dari Blade**

Berikut adalah potongan program yang dipergunakan untuk mengambil muti value yang dikirimkan menggunakan Blade.



Mendapatkan Multi Value dengan Blade