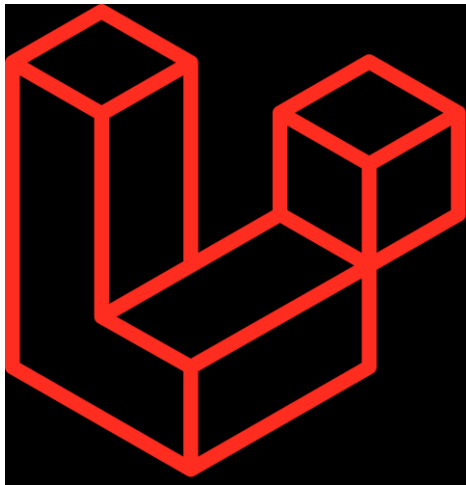


Modul pembelajaran

Laravel 6.8



Aji Syahroni & Zauvik Rizaldi M

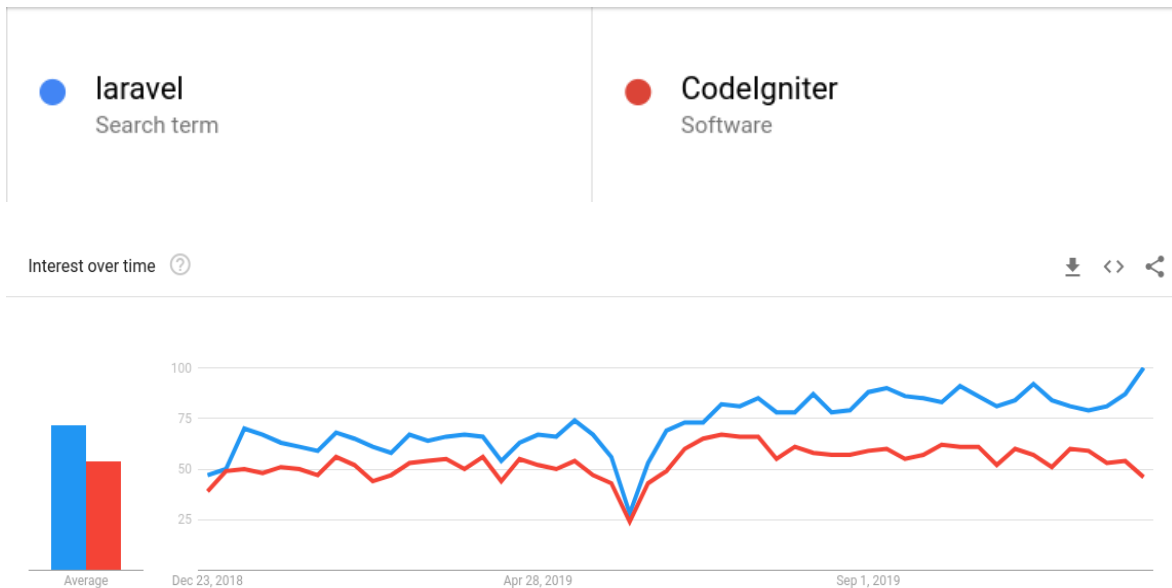
Supported by : [Ngulicode.com](https://ngulicode.com)

19th December, 2019

STAGE I – KENALAN YOOKKKK!

Apa itu Laravel

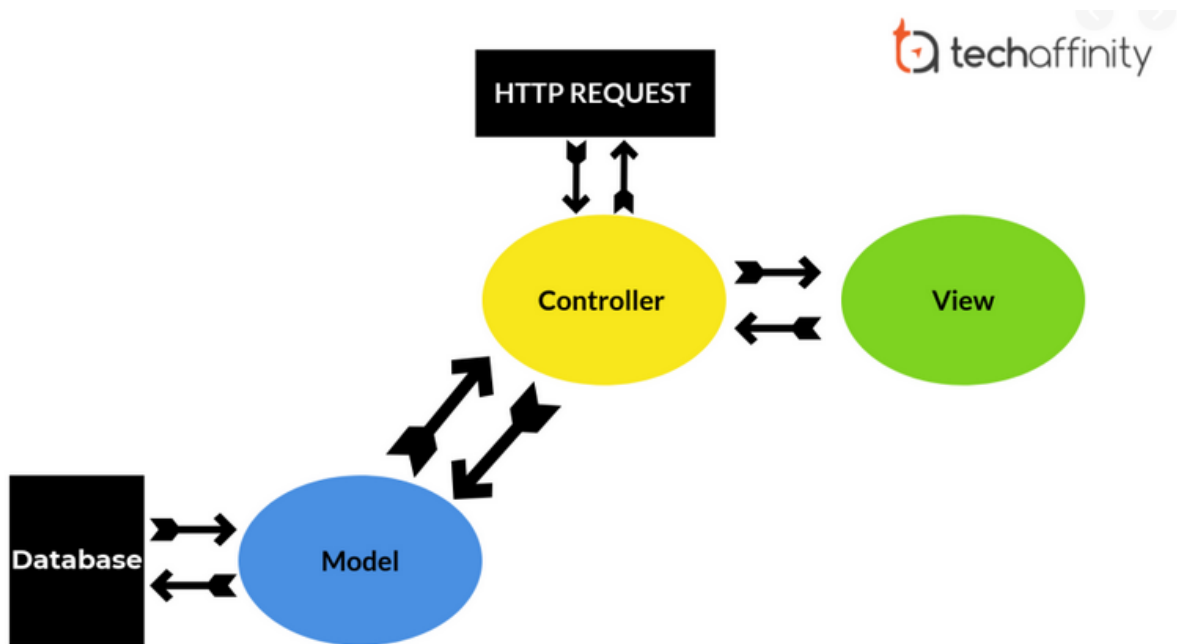
Laravel adalah framework PHP dibuat oleh **Taylor Otwell**, laravel merupakan framework yang populer dan banyak digunakan karena kemudahan dan kecepatan dalam pengembangannya. Laravel memiliki dukungan dokumentasi resmi yang lengkap selain itu laravel juga memiliki dukungan komunitas dan library yang sangat banyak, tak heran jika laravel memiliki banyak pengguna yang loyal. Tingkat kepopuleritasan laravel mengalahkan **Codeigniter** di Google Trend



Laravel dapat dikembangkan menjadi aplikasi **fullstack**, artinya menggabungkan **front-end** dan **back-end** dalam satu aplikasi, ataupun **hanya aplikasi backend** saja.

Software Arsitektur

Dalam pembuatan aplikasi berbasis Laravel, arsitektur yang digunakan adalah pattern/pola MVC (Model View Controller), maksud dari MVC adalah memisahkan kode-kode menjadi logic logic khusus sesuai fungsinya masing-masing yaitu model (**prototype data**), view (**tampilan ui/ux**), controller (**logic proses**). Berikut adalah ilustrasi MVC :



MVC pattern , source : <https://techaffinity.com/blog/mvc-architecture-benefits-of-mvc/>

Keunggulan laravel

1. A framework with expressive, elegant syntax.
2. 3 Different ways to perform query :
 - a. Eloquent (ini cantik bgt suer)
 - b. Query Builder
 - c. DB Raw (native-native sql club)

Lihat perbandingan query berikut :

Plain query (misalkan)	SELECT * FROM books
Eloquent	<code>Book::get();</code>
Query Builder	<code>DB::table('books')->get();</code>
DB Raw	<code>DB::select(DB::raw('select * from books'));</code>

Ayo dipilih dipilih mau style yang mana?...

3. Artisan Command

Digunakan untuk mempermudah dan mempercepat kita dalam melakukan sesuatu hal di laravel, misalnya : mengeksekusi method tertentu, membuat file model, controller, migration dan seeder, run project, implementasi template n etc. Banyak bgt cuyyyyy fungsinya ini, dan tentu nya sangat mempermudah hidup agar hidup menjadi lebih baik wkwk.

Ini ada beberapa command yang sering dipake :

Note : btw baca sekilas aja, command-command ini nantinya akan kita gunakan setelah instalasi cuy, dan tanda [] adalah optional option.

Command	Fungsi
Php artisan serve	Run project
Php migrate:[option]	Migrasi schema yang sudah dibuat ke database
Php artisan db:seed [--class=YourClass]	Mengisi data fake di database untuk testing Option -class : Jika class seeder yang digunakan bukan default laravel/DatabaseSeeder

<pre> Php artisan make:migration migration_name </pre>	Membuat file migrasi
<pre> Php artisan make:seeder seeder_name </pre>	Membuat file seeder
<pre> Php artisan make:controller controller_name [--resource] </pre>	Membuat controller Option -resource : Berarti membuat template dari controller yang dibuat
<pre> Php artisan make:model [--mc][--resource] </pre>	Membuat model Option -mc : Berarti membuat model, migration dan controller secara serentak
<pre> Php artisan config:cache </pre>	Untuk menghapus config cache yg lama dan melakukan cache config yang baru, command ini sangat direkomendasikan setelah melakukan perubahan setting di dalam folder /config atau setelah merubah setting di file .env Walaupun php sudah menggunakan PSR-4(standar untuk autoload file), tapi terkadang ada masalah saat auto caching nya (pengalaman pribadi nih wkwk, uda susah2 debugging code yg error mana, eh taunya cache settingan yg baru tidak terload dengan benar)

4. Pretty good documentation (Opini penulis nih)

Penulis sendiri sebelumnya sempat menggunakan React Js, Vue Js ,Angular , Ci, Slim, dan laravel/lumen. Dari semua dokumentasi yang framework yg disebutkan, penulis sangat suka dokumentasi Laravel. Dari yang basic sampe bedah arsitektur dan core dari framework ada semua coy, ditambah lagi ada tutorial vidio, maupun forum di official media partner Laravel yaitu di laracast.

STAGE 2 - Getting Started!

Prerequisites

Untuk menginstall Laravel di sistem komputer kita, ada beberapa persyaratan yang enviroment harus dipenuhi, yaitu :

1. PHP >= 7.2.5 recommended 7.3.* dengan ekstensi php sebagai berikut :

- BCMath PHP Extension
- Ctype PHP Extension
- JSON PHP Extension
- Mbstring PHP Extension
- OpenSSL PHP Extension
- PDO PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

Ribet? Wkwkwk langsung install package dari Xampp(windows), atau Lampp(Linux). Jangan lupa install yang latest version ya!..

2. Install composer

Composer merupakan dependency management/package management php. Jika node.js punya NPM, maka composer ini adalah NPM nya si PHP.

Depedency Manager adalah software yang memudahkan kita dalam memanajemen ataupun menginstal paket-paket yang kita butuhkan hanya dengan command di cli (dulu sebelum ada ini, para developer harus nge clone dr github/source lain untuk install paket-paket yang dibutuhkan).

Untuk instalasi composer di windows maupun os lain, silahkan kunjungi link berikut : <https://getcomposer.org/download/>

3. Install Database(optional)

Jika kalian memang ingin membuat website dinamis, maka diperlukan database untuk menyimpan data-data. Kalau di tutorial ini pastinya butuh dongg yaaa. Bagi yang udah instal Xampp/Lampp database nya udah embed kok.

Membuat laravel 6.8 project

Instalasi projek laravel bisa melalui 2 metode, metode 1 melalui laravel installer dan metode 2 menggunakan composer

1. Metode 1 (Laravel installer)

Laravel installer adalah paket khusus composer yang digunakan untuk membuat dan manajemen project laravel

Menambahkan paket **laravel installer** dengan composer

```
composer global require laravel/installer
```

Pembuatan projek laravel dengan laravel installer, nama projek laravel yang dibuat adalah **blog**

```
laravel new blog
```

2. Metode 2 (Composer create project)

Pembuatan project laravel juga dapat dilakukan **tanpa** menginstall **laravel installer**, akan tetapi dapat langsung dibuat dengan composer create project, nama projek laravel yang dibuat adalah **blog**

```
composer create-project --prefer-dist laravel/laravel blog
```

Menjalankan project laravel

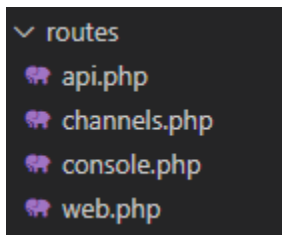
Untuk menjalankan projek laravel di browser maka buka cmd/terminal lokasi projek anda berada, ketikan perintah `php artisan serve` , kemudian buka <http://localhost:8000> di browser maka akan mengarahkan pada projek laravel anda

```
Laravel development server started: http://127.0.0.1:8000
```


Struktur Dasar project Laravel

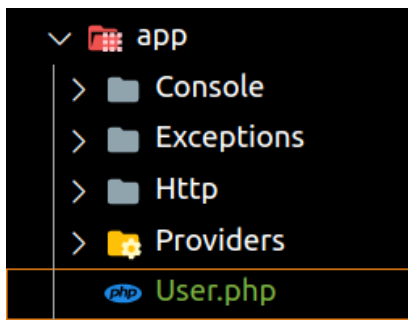
Karena modul ini ditujukan untuk beginner, untuk sekarang maka kalian hanya perlu mengetahui lokasi dan penggunaan Route, Model, Controller, View, Migration dan Seeder.

1. **Routing** sendiri adalah proses pengiriman data maupun informasi ke pengguna melalui sebuah permintaan yang dilakukan kepada alamat yang sudah terdaftar, lalu alamat tersebut akan memproses dari permintaan kita tadi. Letak route berada di `/routes`



Terdapat banyak jenis route laravel. Tapi karena di modul ini kita akan membuat basic aplikasi CRUD dan fullstack web, maka yang kita perlukan sementara hanya routing di file `web.php`

2. **Model** , model adalah representasi dari **data yang dihimpun dari database** terletak pada folder `/app`.



Model akan menjadi perantara antara Database dan Controller. Jadi di file ini, kita melakukan pengolahan data plain dari database kita sebelum di kasih “logic” pada controller nantinya.

Dalam sampel bawaan laravel sudah terdapat 1 file model bernama User.php, untuk pembuatan model dengan menggunakan bantuan php artisan yaitu, `php artisan make:model model_name`

Laravel memiliki fitur yang bernama **Laravel Naming Convetion** untuk mempercepat pekerjaan para pemakainya. Fitur ini sangat-sangat usefull bagi yang sudah paham konsepnya, tapi bagi yang belum paham, bisa jadi bencana bro, karna fitur ini laravel bakalan keliatan kayak pesulap dahhh. Jika kita belum tau secretnya, pasti bakalan puyeng setengah mati wkwk.

Laravel Naming Convetion pada model memungkinkan kita membuat sebuah model yang auto konek ke table kita tanpa mendeklarasikan nama table kita, pada file model! Ajaib gak tuh? Ini yang membuat para beginner selalu mikir “loh ini model gua kok kaga konek-konek sama table gua sih!”.

Sebenarnya kita bisa meng-override rule **Laravel Naming convention** dengan menambahkan properti protected \$table = books. Ada banyak sekali jenis properti pada laravel model. Di dokumentasi laravel, hal ini termasuk kedalam pembahasan Eloquent ORM, check this out <https://laravel.com/docs/5.3/eloquent>

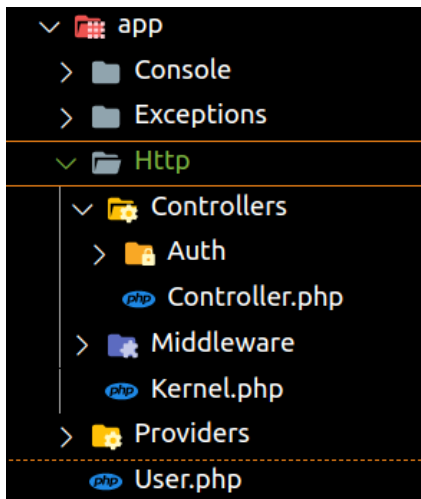
Aturan penamaan pada model agar auto terkoneksi adalah:

Misalkan kita punya table bernama ‘books’

Maka nama model nya harus Book.php (Uppercase huruf awal dan menghilangkan s di akhir kata). Nama model ini juga wajib harus musti sama dengan nama class di dalam nya.

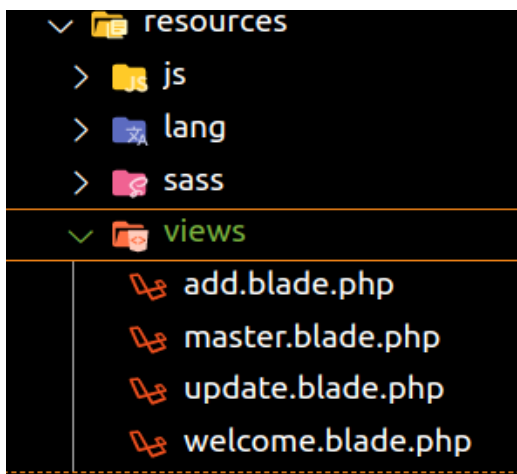
3. **Controller**, controller adalah pengolahan logika program berada (perhitungan, logika program dll). Controller terletak di folder / **app** / **http** / **controllers**.

Dalam sampel bawaan laravel belum terdapat file controller, pembuatan file controller dengan menggunakan perintah artisan, yaitu `php artisan make:controller controller_name`

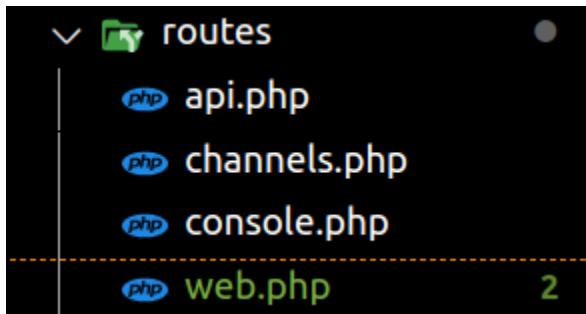


4. **Lokasi view**, view atau tampilan berada pada folder resources / **views**. File view pada laravel berekstensi **.blade.php**

Dalam sampel bawaan laravel belum terdapat sample file view bawaan bernama `welcome.blade.php`



Agar halaman **view** bisa diakses pada browser maka harus terdaftar pada routing, perhatikan pada **routes / web.php**



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

LETS CODE!

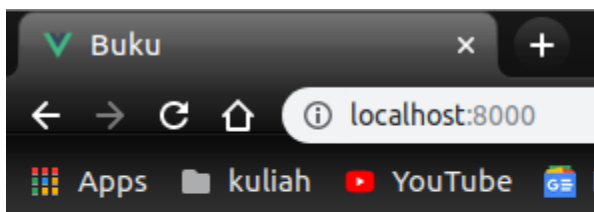
Basic View Laravel

Templating di laravel menggunakan templating yang bernama **blade**, **blade** berfungsi untuk mempermudah penulisan **view** menggunakan php, dalam **blade** dapat menggunakan **perulangan**, **percabangan**, dan **standar layouting** dengan menjaga penulisan kode agar tetap rapi dan terstruktur.

Mengubah tampilan `welcome.blade.php`

Masuk pada folder `resources / views / welcome.blade.php` , lakukan perubahan konten pada `welcome.blade.php`, lalu refresh halaman browser <http://localhost:8000>

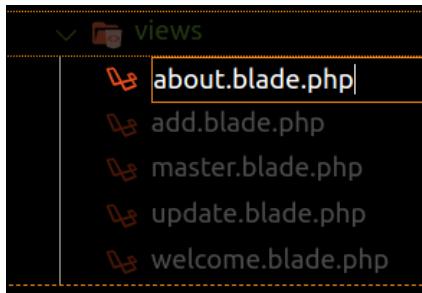
```
welcome.blade.php x
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7
8     <title>Buku</title>
9
10    <!-- Fonts -->
11
12 </head>
13
14 <body class="container">
15
16     Ini adalah halaman welcome.blade.php
17
18 </body>
19
20 </html>
```



Ini adalah halaman `welcome.blade.php`

Menambahkan halaman pada view laravel

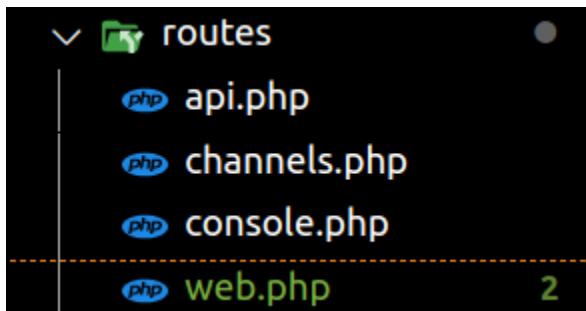
Buat file bernama `about.blade.php` pada folder `resources / views` , kemudian isi konten dari `about.blade.php`



`about.blade.php`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>About</title>
</head>
<body>
  <h1>ini adalah halaman about</h1>
</body>
</html>
```

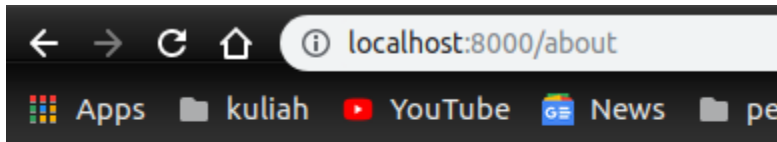
Tambahkan routing bernama `‘/about’` halaman anda pada `routes / web.php`



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

```
});
```

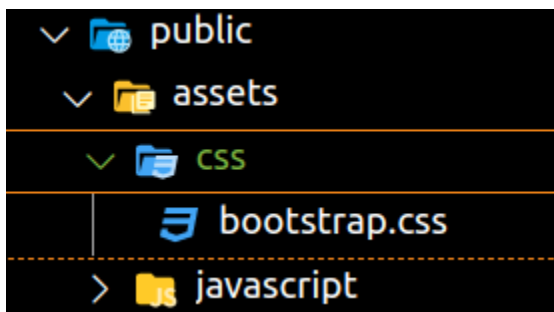
Buka di browser <http://localhost:8000/about>



ini adalah halaman about

Menambahkan bootstrap pada blade laravel

Untuk menambahkan asset seperti css / javascript / icon dll, buka folder public tambahkan semua file asset dalam folder tersebut, misalnya bootstrap css/javascript dll



Setelah file asset ditambahkan dalam folder public, langkah selanjutnya file asset dapat langsung diimpor dalam file **.blade.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>About</title>
  <link rel="stylesheet" type="text/css"
href="assets/css/bootstrap.css">
```

```
</head>
<body class="container">
  <h1>ini adalah halaman about</h1>
</body>
</html>
```

Mini task

Buatlah 3 view laravel untuk seperti berikut

read.blade.php

No.	Nama buku	Pengarang	Penerbit	Tahun terbit	aksi
1	Algoritma	Ajik	Jogja penerbit	2019	<button>update</button> <button>delete</button>

Insert.blade.php

Nama Buku

Pengarang

Penerbit

Tahun terbit

Submit

Update.blade.php

Nama Buku

Pengarang

Penerbit

Tahun terbit

Update

Membuat layout master pada blade laravel

Layout master adalah layout/view yang digunakan sebagai **acuan/standar** dalam pembuatan layout-layout lain

Langkah pertama buatlah file bernama **master.blade.php** , file ini nantinya digunakan sebagai acuan oleh layout lain

```
<body class="container">
  <div>
    <h1 class="display-4">Data buku</h1>
    <a href="add">+ add buku</a>
    <div class="my-5"></div>
  </div>

  @yield('main_content')
</body>
```

```
<body class="container">

  @extends('master')
  @section('main_content')
    <table class="table">
      <thead>
        <tr>
          <th>No.</th>
```

```

        <th>Nama buku</th>
        <th>Pengarang</th>
        <th>Penerbit</th>
        <th>Tahun terbit</th>
        <th>aksi</th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>1</td>
        <td>Algoritma</td>
        <td>Ajik</td>
        <td>Jogja penerbit</td>
        <td>2019</td>
        <td>
            <a href="update" class="btn btn-sm btn-
success">update</a>
            <a href="#" class="btn btn-sm btn-danger">delete</a>
        </td>
    </tr>
</tbody>
</table>
@stop
</body>

```

No.	Nama buku	Pengarang	Penerbit	Tahun terbit	aksi
1	Algoritma	Ajik	Jogja penerbit	2019	<div>update</div> <div>delete</div>

Data buku

[+ add buku](#)

Mengirim data string dari routing atau controller ke view

Untuk melakukan pengiriman data atau variabel dari **routing** atau **controller** ke **view** maka data yang dikirim harus dikemas dalam **format array assosiatif**

Routes / web.php

```
Route::get('/about', function () {  
  
    $my_name = "aji syahroni";  
    return view('about', ['nama' => $my_name]);  
  
});
```

Untuk menampilkan data yang dikirimkan oleh **routing** atau **controller** maka harus ditampung dalam expressi **{{ }}**

Resources / views / about.blade.php

```
<body class="container">  
    <h1>ini adalah halaman about</h1>  
    <p>created by : {{ $nama }}</p>  
</body>
```

Mengirim data array dari routing atau controller ke view

Routes / web.php

```
Route::get('/about', function () {

    $my_name = "aji syahroni";
    $book = [
        "algoritma" => 12,
        "data mining" => 30,
        "AI" => 33,
        "image processing" => 55,
        "RPL" => 100
    ];

    return view('about', ['nama' => $my_name, 'buku' => $book]);

});
```

Untuk menampilkan data array dalam file blade dapat menggunakan perulangan `@foreach` yang disediakan oleh blade php

Resources / views / about.blade.php

```
<body class="container">
    <h1>ini adalah halaman about</h1>
    <p>created by : {{ $nama }}</p>

    <ul>
        @foreach($buku as $key=>$value)
            <li>{{ $key }} - {{ $value }} </li>
        @endforeach
    </ul>

</body>
```



ini adalah halaman about

created by : aji syahroni

- algoritma - 12
- data mining - 30
- AI - 33
- image processing - 55
- RPL - 100

Percabangan pada view

Selain perulangan, percabangan if dan switch dapat dilakukan dalam file blade.php

```
<body class="container">
  <h1>ini adalah halaman about</h1>
  <p>created by : {{ $nama }}</p>

  <ul>
    @foreach($buku as $key=>$value)
      <li>
        {{ $key }} -

        @if($value > 50)
          <span class="badge badge-success">{{ $value }} </span>
        @else
          <span class="badge badge-info">{{ $value }} </span>
        @endif

      </li>
    @endforeach
  </ul>

</body>
```

ini adalah halaman about

created by : aji syahroni

- algoritma - 12
- data mining - 30
- AI - 33
- image processing - 55
- RPL - 100