



Tutorial Laravel Dasar Part I

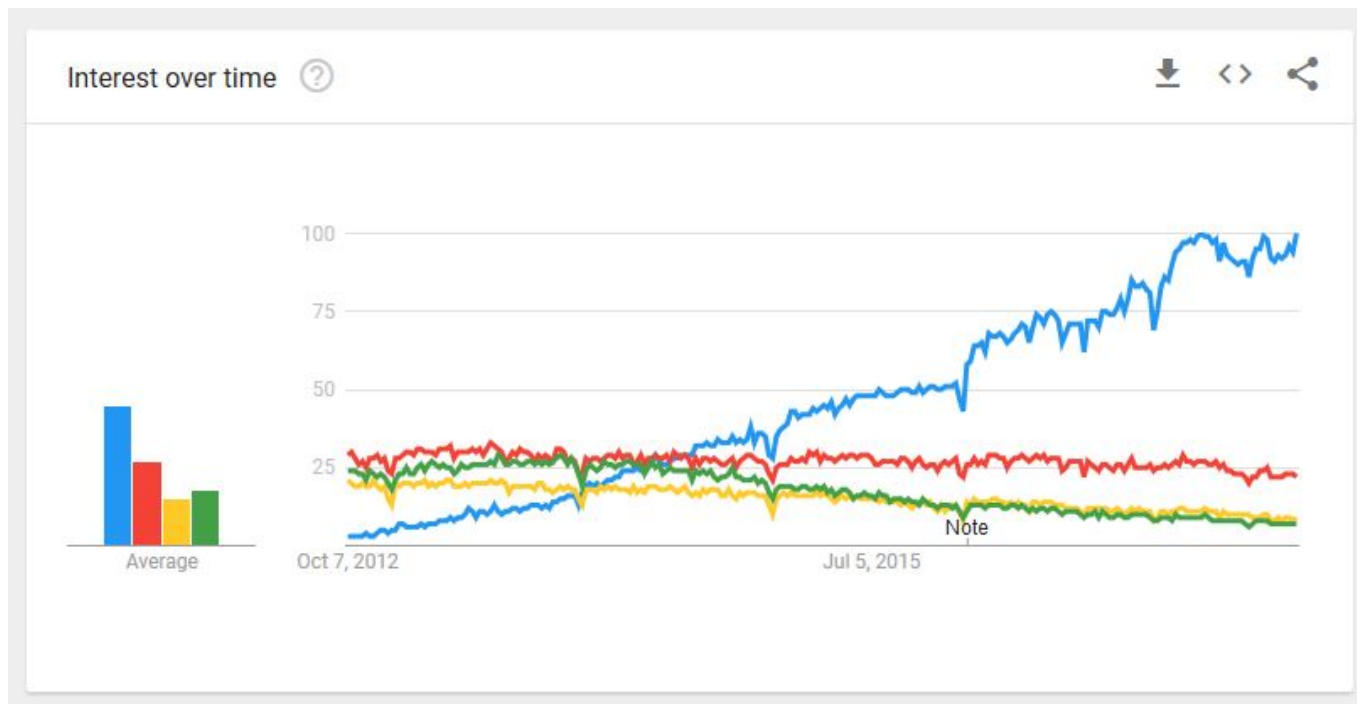
Oleh: Cecep Yusuf

kali ini kita akan belajar mempelajari Laravel mulai dari dasar hingga mahir. Untuk koleksi tutorial utama saya (Tutorial CakePHP Dasar Part I s/d IV) sepertinya tidak akan saya update lagi, berhubung kini saya sudah hijrah ke framework Laravel.



Halo teman-teman semua, mohon maaf sudah sekian lamanya saya tidak kontribusi membuat tutorial lagi di sini. Jadi selama kurang lebih hampir 1 tahun ini saya lebih banyak mantau situs ini dibandingkan dengan membuat tutorial baru. Dikarenakan banyak kesibukan lain di luar, saya jadi benar-benar sedikit waktu untuk membuat tutorial. Namun diusahakan mulai sekarang saya mau aktif lagi membuat tutorial di situs kesayangan kita ini :) Tapi jangan khawatir, karena situs ini masih terawat dengan baik kok, hanya saja saya yang kurang aktif membuat tutorial hehehe.

Baiklah, untuk kali ini kita akan belajar mempelajari Laravel mulai dari dasar hingga mahir. Untuk koleksi tutorial utama saya (Tutorial CakePHP Dasar Part I s/d IV) sepertinya tidak akan saya update lagi, berhubung kini saya sudah hijrah ke framework Laravel. Kenapa hijrah? Itu bukan semata-mata saya meninggalkan CakePHP dan kemudian beralih ke yang baru lho, melainkan saya pikir melihat update dan popularitas framework saat ini Laravel lebih baik.



<p>● laravel Search term</p>	<p>● codeignit... Search term</p>	<p>● cakephp Search term</p>	<p>● yii Search term</p>
----------------------------------	---------------------------------------	----------------------------------	------------------------------

Apakah kalian memperhatikan gambar di atas? Ya, Laravel unggul lebih jauh dibandingkan framework-framework yang lain. Luar biasa? Sangat. Oleh karena itu saya mulai hijrah ke Laravel dari CakePHP selama beberapa tempo terakhir ini.

Berikut adalah artikel yang dibuat oleh mas David mengenai Laravel, cekidot:

http://jagocoding.com/tutorial/433/Berkenalan_Dengan_Framework_Laravel

Insy Allah saya akan menyempatkan waktu untuk membuat artikel mengenai kelebihan dan kekurangan Laravel dibandingkan dengan framework lain dengan lebih detail. Pada tutorial ini kita akan fokus ke praktik secara teknis menggunakan framework populer ini.

Selamat datang di **Tutorial Laravel Dasar Part I by Cheyuz.**

Persiapan Instalasi Laravel

Untuk instalasi Laravel, kita membutuhkan beberapa komponen berikut, yaitu:

1. PHP

Minimal versi 5.6.4 untuk Laravel 5.4

Minimal versi 7.0 untuk Laravel 5.5

- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension

- Tokenizer PHP Extension
- XML PHP Extension

1. MySQL Database
2. Composer
3. IDE / Text Editor

PHP & MySQL

Untuk PHP, kita akan menggunakan XAMPP saja untuk development. Dengan menginstall XAMPP, kita tidak perlu direpotkan instalasi PHP, MySQL, dan web server Apache secara manual. Untuk instalasi Xampp, silahkan download saja di sini:

<https://www.apachefriends.org/download.html>

Download dan install sesuai OS yang terpasang di komputer kita ya. Untuk tutorial kali ini kita akan fokus ke instalasi Laravel, maka dari itu kita asumsikan bahwa kita sudah menginstall aplikasi Xampp pada system operasi **Windows 64 bit**. Kemudian kita asumsikan juga untuk folder webrootnya misalnya berada di **C:\xampp\htdocs** biar sepakat kita pakai drive C saja.

Composer

Composer sudah menjadi tren baru pada komunitas PHP. Kehadiran composer mampu menarik perhatian para developer PHP karena menawarkan kemudahan dalam proses development aplikasi kita. Composer memudahkan kita dalam menginstall library external yang dibutuhkan dalam project kita. Kita cukup menambahkan satu bari perintah dan menjalankan composer maka secara otomatis composer akan mendownload dan membuat autoload class dari library yang kita butuhkan sehingga kita bisa langsung menggunakannya tanpa perlu membuat class loadernya lagi.

Untuk instalasi Composer, silahkan lihat tutorial di bawah ini:

http://jagocoding.com/tutorial/384/Instalasi_Composer_untuk_Windows

Text Editor

Teman-teman bebas memilih mau pakai text editor apa saja. Untuk framework Laravel saya sarankan menggunakan **PHP Storm** karena integrasi dengan framework Laravel sangat sangat baik dan juga sangat cerdas. Sayangnya PHP Storm tidak gratis, kita harus membelinya karena license IDE ini bukanlah freeware. Jika teman-teman ingin yang *gratisan* silahkan pakai **Netbeans** saja, selain gratis, kecerdasan IDEnya setara dengan PHP Storm, hanya saja menurut saya integrasi dengan Laravel agak kurang dibandingkan dengan PHP Storm.

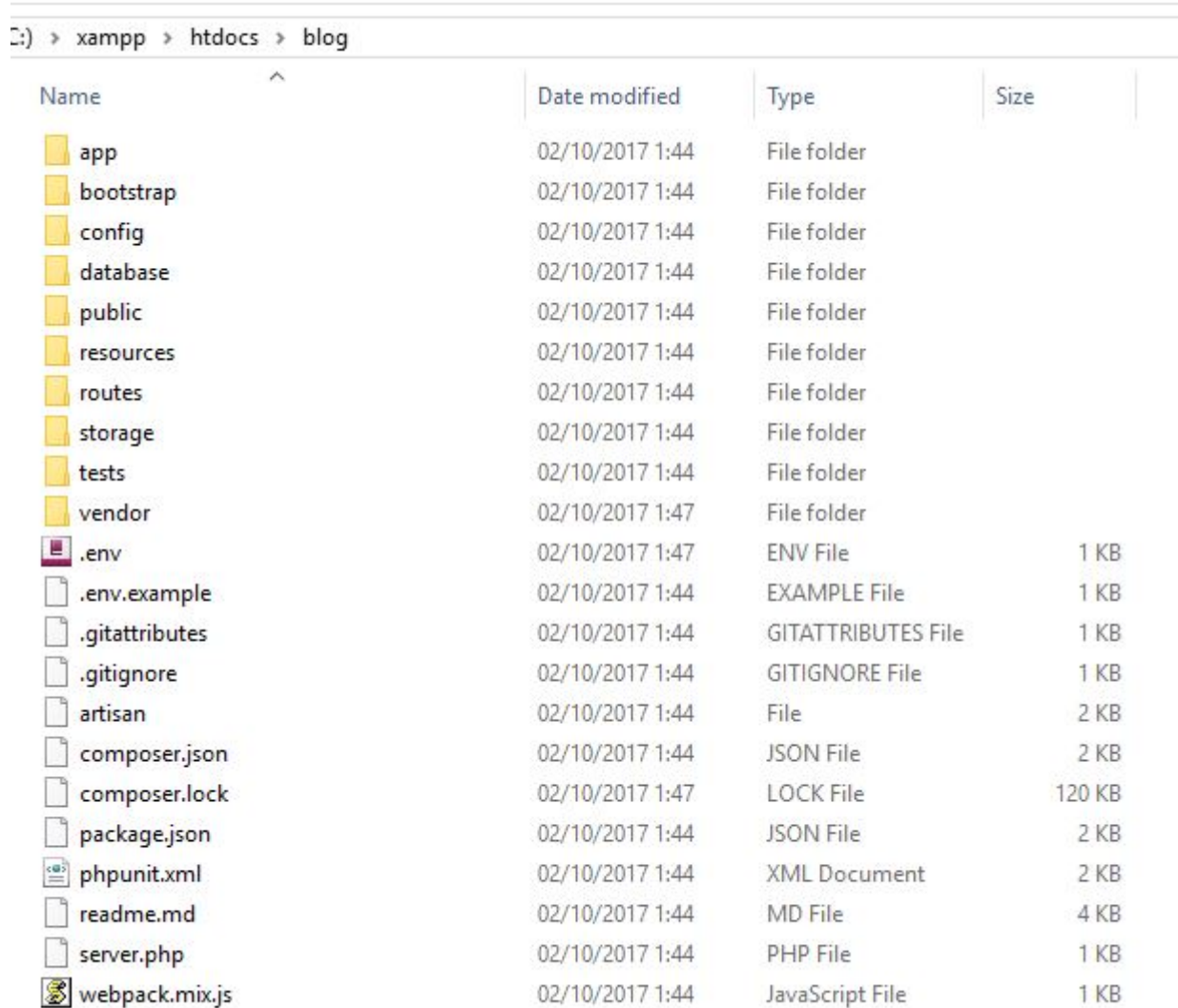
Memulai Instalasi

Pastikan semua kebutuhan sudah terpasang dengan baik di komputer kita. Jika sudah terinstall, silahkan buka Command Prompt di drive ini **C:\xampp\htdocs**. Setelah itu, ketik command berikut:

```
composer create-project --prefer-dist laravel/laravel blog "5.4.*"
```

Selanjutnya Composer akan mendownload skeleton dari framework Laravel, berupa folders dan files

yang siap pake (starter) untuk Laravel.



Name	Date modified	Type	Size
app	02/10/2017 1:44	File folder	
bootstrap	02/10/2017 1:44	File folder	
config	02/10/2017 1:44	File folder	
database	02/10/2017 1:44	File folder	
public	02/10/2017 1:44	File folder	
resources	02/10/2017 1:44	File folder	
routes	02/10/2017 1:44	File folder	
storage	02/10/2017 1:44	File folder	
tests	02/10/2017 1:44	File folder	
vendor	02/10/2017 1:47	File folder	
.env	02/10/2017 1:47	ENV File	1 KB
.env.example	02/10/2017 1:44	EXAMPLE File	1 KB
.gitattributes	02/10/2017 1:44	GITATTRIBUTES File	1 KB
.gitignore	02/10/2017 1:44	GITIGNORE File	1 KB
artisan	02/10/2017 1:44	File	2 KB
composer.json	02/10/2017 1:44	JSON File	2 KB
composer.lock	02/10/2017 1:47	LOCK File	120 KB
package.json	02/10/2017 1:44	JSON File	2 KB
phpunit.xml	02/10/2017 1:44	XML Document	2 KB
readme.md	02/10/2017 1:44	MD File	4 KB
server.php	02/10/2017 1:44	PHP File	1 KB
webpack.mix.js	02/10/2017 1:44	JavaScript File	1 KB

Setelah skeleton framework Laravel sudah terbuat, kita jalankan web server bawaan [PHP](#) dengan menggunakan **artisan** Laravel. Artisan adalah *command line* atau perintah-perintah yang dijalankan via command prompt/terminal yang nyediain beberapa perintah yang bisa kita pakai selama develop aplikasi Laravel.

Jalankan servernya dengan command berikut:

```
php artisan serve
```

Jangan lupa untuk masuk terlebih dahulu ke folder **blog** yang ada di htdocs. Jadi lengkapnya seperti ini di Command Prompt:

```
c:\xampp\htdocs\blog>php artisan serve
```

Maka setelah itu akan ada tanda bahwa web server built in sedang aktif:

```
c:\xampp\htdocs\blog>php artisan serve
```

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

Silahkan buka alamat berikut di browser kita:

<http://localhost:8000>

Maka browser akan menampilkan halaman selamat datang dari Laravel seperti gambar di bawah ini:

Laravel

DOCUMENTATION

LARACASTS

NEWS

FORGE

GITHUB

Untuk instalasi telah selesai. Selamat datang di Laravel! Sangat mudah, kan?

Untuk tahapan berikutnya kita akan membuat database **db_blog** dengan menggunakan MySQL.

Setting Up Database

Untuk database, kita menggunakan MySQL sebagai RDBMSnya. Di dalam paket Xampp, MySQL sudah tersedia, sehingga kita tinggal jalankan MySQL Servernya kemudian buka phpMyAdmin di browser.

Buatlah database dengan nama **db_blog**, kemudian copy paste SQL query di bawah ini untuk membuat tabel-tabel yang dibutuhkan.

```
CREATE TABLE IF NOT EXISTS `contents` (  
  `id` int(11) NOT NULL,  
  `title` varchar(125) NOT NULL,  
  `excerpt` text NOT NULL,
```

```

`content` text NOT NULL,
`user_id` int(11) DEFAULT NULL,
`created_at` datetime NOT NULL,
`updated_at` datetime NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1;

```

Kita membuat table contents sebagai contoh kasus kali ini. Jadi ceritanya kita akan membuat blog sederhana dengan menggunakan Laravel. Untuk tahap awal kita buat satu table terlebih dahulu, karena kita akan fokus ke CRUD-nya terlebih dahulu. Untuk tutorial selanjutnya kita akan mempelajari tahap demi tahap bagaimana cara merelasikan model, layouting/templating, manajemen user, autentikasi, dsb. Untuk saat ini kita akan belajar bagaimana menampilkan data terlebih dahulu.

Perlu diperhatikan bahwa penamaan table yang baik itu harus **bahasa inggris** dan **plural/jamak**, dikarenakan dengan convention tersebut kita akan dimudahkan dalam integrasi database dengan framework Laravel dan framework lainnya. Jika kita lihat di atas ada field dengan nama **user_id**, ini untuk sementara dikosongkan terlebih dahulu, karena tidak akan kita pakai sekarang. Tapi nanti kita akan pakai untuk relasi ke table lain, namanya users, setelah kita pelajari model association/relation. Untuk tutorial pertama ini kita fokus ke cara menampilkan data yang simpel terlebih dahulu.

Ok selanjutnya kita isikan data terlebih dahulu, kita akan coba tampilkan data-data yang sudah diinputkan dengan menggunakan Laravel. Silahkan copy paste SQL query di bawah ini untuk menambahkan data secara manual di MySQL.

```

INSERT INTO `contents` (`id`, `title`, `excerpt`, `content`, `user_id`,
`created_at`, `updated_at`) VALUES
(1, 'Tutorial Laravel Dasar Part I', 'Ini merupakan excerpt/cuplikan',
'Yang ini adalah konten, ini adalah isi konten, isi post yang berisi
konten, karena post adalah konten. Yang ini adalah konten, ini adalah isi
konten, isi post yang berisi konten, karena post adalah konten. Yang ini
adalah konten, ini adalah isi konten, isi post yang berisi konten, karena
post adalah konten.', NULL, '2017-10-02 09:00:00', '2017-10-02 09:00:00'),
(2, 'Tutorial Laravel Dasar Part II', 'Yang ini adalah excerpt Tutorial
Laravel Part II', 'Ini adalah isi dari konten tutorial laravel part 2, isi
kontennya ada di sini. Yuk kita belajar Laravel di sini, di Jagocoding.com.
Ini adalah isi dari konten tutorial laravel part 2, isi kontennya ada di
sini. Yuk kita belajar Laravel di sini, di Jagocoding.com. Ini adalah isi
dari konten tutorial laravel part 2, isi kontennya ada di sini. Yuk kita
belajar Laravel di sini, di Jagocoding.com.', NULL, '2017-10-02 09:00:00',
'2017-10-02 09:00:00'),
(3, 'Tutorial Javascript Dasar', 'Nah yang ini adalah tutorial Javascript',
'Ini merupakan tutorial javascript dasar, yuk kita belajar Javascript di
Jagocoding.com. Belajar javascript itu asyik lho. Javascript berbeda
dengan Java ya. Ini merupakan tutorial javascript dasar, yuk kita belajar
Javascript di Jagocoding.com. Belajar javascript itu asyik lho. Javascript
berbeda dengan Java ya. Ini merupakan tutorial javascript dasar, yuk kita
belajar Javascript di Jagocoding.com. Belajar javascript itu asyik lho.
Javascript berbeda dengan Java ya.', NULL, '2017-10-02 11:00:00',
'2017-10-02 11:00:00'),

```

```
(4, 'Tips dan Trik coding yang Baik', 'Ini merupakan excerpt/cuplikan tips dan trik coding yang baik.', 'Ini merupakan konten untuk tips dan trik coding yang baik. Silahkan belajar di sini, hanya di Jagocoding.com. Yuk kita belajar sama-sama. Ini merupakan konten untuk tips dan trik coding yang baik. Silahkan belajar di sini, hanya di Jagocoding.com. Yuk kita belajar sama-sama. Ini merupakan konten untuk tips dan trik coding yang baik. Silahkan belajar di sini, hanya di Jagocoding.com. Yuk kita belajar sama-sama. Ini merupakan konten untuk tips dan trik coding yang baik. Silahkan belajar di sini, hanya di Jagocoding.com. Yuk kita belajar sama-sama. ', NULL, '2017-10-02 14:00:00', '2017-10-02 14:00:00');
```

Silahkan execute script di atas untuk menambahkan data secara manual. Perhatikan bahwa contoh konten di atas bisa teman-teman custom saja ya, tidak perlu sama dengan contoh di atas, hehehe..

Berikut ini adalah screenshot data yang sudah ditambahkan di phpmyadmin:

title	excerpt	content	user_id	created_at	updated_at
Tutorial Laravel Dasar Part I	Ini merupakan excerpt/cuplikan	Yang ini adalah konten, ini adalah isi konten, isi...	NULL	2017-10-02 09:00:00	2017-10-02 09:00:00
Tutorial Laravel Dasar Part II	Yang ini adalah excerpt Tutorial Laravel Part II	Ini adalah isi dari konten tutorial laravel part 2...	NULL	2017-10-02 09:00:00	2017-10-02 09:00:00
Tutorial Javascript Dasar	Nah yang ini adalah tutorial Javascript	Ini merupakan tutorial javascript dasar, yuk kita ...	NULL	2017-10-02 11:00:00	2017-10-02 11:00:00
Tips dan Trik coding yang Baik	Ini merupakan excerpt/cuplikan tips dan trik codin...	Ini merupakan konten untuk tips dan trik coding ya...	NULL	2017-10-02 14:00:00	2017-10-02 14:00:00

Routing dalam Laravel

Yang perlu dipelajari pertama kali ketika kita develop dengan Laravel adalah routing. Routing/route adalah bentuk endpoint/url yang akan kita pakai untuk membuat fungsi program yang sudah didefine di dalam controller function atau dalam bentuk anonymous function (closure). Dalam case ini kita akan membuat route yang mengarahkan ke controller **ContentsController** dengan nama function **index**.

Mari kita tes apakah di dalam aplikasi kita sudah berjalan dengan baik atau belum, dengan cara buat route baru dengan nama **halo**.

Silahkan buka file **/routes/web.php** di dalam framework Laravel kita, kemudian tambahkan baris route baru dengan tipe GET di bawah route **welcome**.

```
<?php
```

```
/*
```

```
|-----
```

```
-
```

```
| Web Routes
```

```
|-----
```

```
-
```

```
|
```



```
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

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

Route::get('halo', function(){
    return 'Hello world! Halo semuanya, selamat belajar Laravel.';
});
```

Route yang kita buat tersebut tidak mengarah ke controller dan function di controller, namun secara langsung return text "**Hello world! Halo semuanya, selamat belajar Laravel.**" tanpa layout yang didefinisikan, sehingga jika kita buka endpoint "**halo**" di browser, hasilnya akan terlihat text tersebut.

Silahkan buka <http://localhost:8000/halo>

Maka browser akan memunculkan text yang sudah kita buat tersebut, seperti gambar di bawah ini:



Membuat Controller, Model, dan Action/Function

Sekarang kita akan membuat controller dan action untuk menampilkan data content yang sudah kita buat di MySQL.

Silahkan buka console/terminal/command prompt dan arahkan ke root aplikasi kita, yaitu **C:\xampp\htdocs\blog**, selanjutnya buatlah controller dengan menggunakan perintah **artisan** sebagai berikut:

```
php artisan make:controller ContentsController
```

Eksekusi command tersebut untuk proses generate controller baru di laravel hingga muncul informasi seperti berikut:

```
C:\xampp\htdocs\blog>php artisan make:controller ContentsController
Controller created successfully.

C:\xampp\htdocs\blog>
```

Selanjutnya lihat direktori **/app/Http/Controllers**, maka akan muncul satu controller baru bernama

ContentsController.php. Buka file tersebut untuk melihat isi filenya.

ContentsController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ContentsController extends Controller
{
    //
}
```

Selanjutnya mari kita buat model **Content** untuk mapping dengan database. Inilah yang disebut **ORM**. Dengan ORM, kita tidak harus membuat kode SQL Native (select bintang blablabla) di program yang kita buat, melainkan dengan menggunakan fungsi-fungsi PHP yang sudah disediakan oleh Laravel, sehingga kita tidak kesulitan jika database kita mengalami perubahan nama table, nama database, bahkan tipe RDBMS itu sendiri.

Buatlah model dengan menggunakan artisan dengan command berikut:

```
php artisan make:model Content
```

Berikut ini adalah tampilan jika kita sudah execution command tersebut:

```
C:\xampp\htdocs\blog>php artisan make:model Content
Model created successfully.
```

```
C:\xampp\htdocs\blog>
```

Kemudian lihatlah direktori **/app**, akan terbuat sebuah file php baru dengan nama **Content.php**. Silahkan buka file tersebut untuk melihat isinya.

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Content extends Model
{
    //
}
```

Ok untuk model sebetulnya cukup begitu saja, karena Laravel secara otomatis akan mapping model dengan database dikarenakan namanya sama.

Sekarang kita akan buat function untuk menampilkan datanya. Silahkan buka controller **ContentsController.php** dan tambahkan kode berikut di bawah namespace:

```
use App\Content;
```

Sehingga kode lengkapnya akan menjadi seperti di bawah ini:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Content;

class ContentsController extends Controller
{
    //
}
```

perintah **use** digunakan ketika kita akan "import" class lain yang mempunyai namespace berbeda. Karena kita akan menggunakan model **Content**, maka kita harus import terlebih dahulu class **Content** tersebut.

Buatlah function **index()** pada controller **ContentsController.php** dengan kode seperti di bawah ini:

ContentsController.php

```
...

function index()
{
    $data = Content::get();
    return $data;
}

...
```

Content::get() berfungsi untuk mengambil data dari model Content dengan settingan default, artinya

kita tidak menambahkan kondisi seperti where, group, join, dsb. Jadi untuk contoh ini kita pakai function `get()` saja untuk menampilkan data. Hasil dari pengambilan data disimpan di variable `$data`, kemudian di-return untuk selanjutnya ditampilkan di view. Perlu kita tahu bahwa pada contoh ini kita tidak membuat view terlebih dahulu, tapi kita akan mengembalikannya dalam bentuk **JSON** dengan return ini.

Untuk melihat hasil dari fungsi ini, silahkan buat route-nya terlebih dahulu di **/routes/web.php** dengan menambahkan kode di bawah ini:

web.php

```
Route::get('contents', 'ContentsController@index');
```

Dengan membuat route tersebut, kita dapat akses URL <http://localhost:8000/contents> untuk pointing ke controller `ContentsController` dan function `index()`. Silahkan save file `web.php` kemudian buka di browser URL tersebut. Sehingga hasilnya seperti ini:

```
localhost:8000/contents
JSON Raw Data Headers
Save Copy
0:
  id: 1
  title: "Tutorial Laravel Dasar Part I"
  excerpt: "Ini merupakan excerpt/cuplikan"
  content: "Yang ini adalah konten, ini adalah isi konten, isi post yang berisi konten, karena post ada konten, karena post adalah konten. Yang ini adalah konten, ini adalah isi konten, isi post y
  user_id:
  created_at: "2017-10-02 09:00:00"
  updated_at: "2017-10-02 09:00:00"
1:
  id: 2
  title: "Tutorial Laravel Dasar Part II"
  excerpt: "Yang ini adalah excerpt Tutorial Laravel Part II"
  content: "Ini adalah isi dari konten tutorial laravel part 2, isi kontennya ada di sini. Yuk kita bel laravel part 2, isi kontennya ada di sini. Yuk kita belajar Laravel di sini, di Jagocoding.c sini. Yuk kita belajar Laravel di sini, di Jagocoding.com."
  user_id:
  created_at: "2017-10-02 09:00:00"
  updated_at: "2017-10-02 09:00:00"
2:
  id: 3
  title: "Tutorial Javascript Dasar"
  excerpt: "Nah yang ini adalah tutorial Javascript"
  content: "Ini merupakan tutorial javascript dasar, yuk kita belajar Javascript di Jagocoding.com. Bel tutorial javascript dasar, yuk kita belajar Javascript di Jagocoding.com. Belajar javascript javascript dasar, yuk kita belajar Javascript di Jagocoding.com. Belajar javascript itu asyi
  user_id:
  created_at: "2017-10-02 11:00:00"
  updated_at: "2017-10-02 11:00:00"
3:
  id: 4
  title: "Tips dan Trik coding yang Baik"
```

Bukalah dengan browser firefox agar tampilan JSON dirapikan seperti tampilan di atas.

Nah, mudah bukan?? Dengan langkah-langkah ini saja kita sudah dapat menampilkan semua data yang ada di table **contents**. Untuk selanjutnya, kita akan membuat View di Laravel.

Membuat View untuk Menampilkan Data

Setiap halaman web pasti mempunyai apa yang disebut dengan View. View merupakan file-file pendukung untuk merepresentasikan/ menampilkan program kita, hasil query dan olahan yang kita buat agar terlihat di browser. Type file untuk view adalah PHP, dengan template **Blade**. **Blade** merupakan salah satu dari sekian banyak view templating yang ada di PHP. Dengan menggunakan Blade, kita tidak dirumitkan dengan kode yang berantakan, mudah dimaintenance sehingga kode menjadi bersih sebersih-bersihnya :D

Ok mari kita buat view yang baru untuk menampilkan data list content ini.

Buatlah satu buah folder di **/resources/views/** dengan nama **contents**, dan di dalam folder **contents** buatlah file dengan nama **index.blade.php**. Artinya, kita akan membuat view **index** pada folder **contents** sehingga alamat filenya sekarang menjadi **/resources/views/contents/index.blade.php**. View ini akan digunakan pada function yang sudah kita buat di atas. Copy paste kode di bawah ini untuk view tersebut.

```
<!DOCTYPE html>
<html lang="{{ app()->getLocale() }}">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- CSRF Token -->
  <meta name="csrf-token" content="{{ csrf_token() }}">

  <title>{{ config('app.name', 'Laravel') }}</title>

  <!-- Styles -->
  <link href="{{ asset('css/app.css') }}" rel="stylesheet">
</head>
<body>
<div id="app">

  <div class="container">
    <h2 class="page-header">List of Content</h2>
    <table class="table table-hover">
      <thead>
        <tr>
          <th>Title</th>
          <th>Excerpt</th>
          <th>Created</th>
          <th>Updated</th>
        </tr>
      </thead>
      <tbody>
        @foreach($data as $d)
          <tr>
            <td>{{ $d->title }}</td>
            <td>{{ $d->excerpt }}</td>
            <td>{{ $d->created_at->format('d/M/Y') }}</td>
            <td>{{ $d->updated_at->format('d/M/Y') }}</td>
          </tr>
        @endforeach
      </tbody>
    </table>
  </div>
</div>

<!-- Scripts -->
```

```
<script src="{{ asset('js/app.js') }}"></script>
</body>
</html>
```

Coba perhatikan kode di atas, kita membuat struktur HTML lengkap. Untuk layouting akan dijelaskan pada materi berikutnya, pada tutorial ini kita membuat view tanpa layouting yang baik, sehingga untuk view index ini saja harus dibuat dari tag HTML pembuka hingga penutup HTML. Sebetulnya inti dari view tersebut mulai dari div class container, di mana di situ terdapat **H2** dan satu buah table yang menampilkan data yang sudah kita tarik dari model.

Dengan menggunakan Blade, kita tidak perlu memakai tag PHP pembuka dan penutup, yaitu `<?php` dan `?>`, melainkan dengan identifier bawaan dari blade, yaitu `{{ }}` untuk fetch variable, dan `@foreach @endforeach` untuk membuat perulangan foreach. Di sini kode yang terlihat sangat bersih, sehingga maintainable.

Selanjutnya kita panggil view tersebut di controller dengan mengubah kode yang sebelumnya menjadi seperti berikut:

```
...

function index()
{
    $data = Content::get();
    return view('contents.index', compact('data'));
}

...
```

Pada kode tersebut, kita mengganti callback dari function memanggil function **view()** yang mana akan memanggil file view yang sudah kita buat tadi. Untuk direktori separator (pemisah folder), jangan menggunakan slash (/) atau backslash (\) melainkan hanya titik (.) saja agar lebih *readable*. Selanjutnya variable **\$data** hasil dari penarikan data di model Content harus dipassing ke view dengan cara memasukkan ke parameter ke dua view yaitu **compact('data')**.

Selanjutnya silahkan refresh browser kita.

Tampilan dari view tersebut adalah seperti gambar di bawah ini:

List of Content

Title	Excerpt	Created	Updated
Tutorial Laravel Dasar Part I	Ini merupakan excerpt/cuplikan	02/Oct/2017	02/Oct/2017
Tutorial Laravel Dasar Part II	Yang ini adalah excerpt Tutorial Laravel Part II	02/Oct/2017	02/Oct/2017
Tutorial Javascript Dasar	Nah yang ini adalah tutorial Javascript	02/Oct/2017	02/Oct/2017
Tips dan Trik coding yang Baik	Ini merupakan excerpt/cuplikan tips dan trik coding yang baik	02/Oct/2017	02/Oct/2017

Gimana jika kita ingin menambahkan pagination pada data tersebut? Yuk kita buat saja sekarang.

Menambahkan Pagination pada View

Untuk menambahkan pagination, ubah function **get()** di controller menjadi **paginate(2)**, artinya kita akan menampilkan 2 row setiap halaman dan selanjutnya akan dibuatkan *paging* dengan navigasi previous, next, dan angka dari page. Kode diubah menjadi seperti ini:

```
...  
  
function index()  
{  
    $data = Content::paginate();  
    return view('contents.index', compact('data'));  
}  
  
...
```

Setelah itu, tambahkan kode **{{ \$data->links() }}** di bawah table pada view **index.blade.php**.

```
...  
  
<div class="container">  
    <h2 class="page-header">List of Content</h2>  
    <table class="table table-hover">  
        <thead>  
            <tr>  
                <th>Title</th>  
                <th>Excerpt</th>  
                <th>Created</th>  
                <th>Updated</th>  
            </tr>  
        </thead>  
        <tbody>  
            @foreach($data as $d)
```



```

        <tr>
            <td>{{ $d->title }}</td>
            <td>{{ $d->excerpt }}</td>
            <td>{{ $d->created_at->format('d/M/Y') }}</td>
            <td>{{ $d->updated_at->format('d/M/Y') }}</td>
        </tr>
    @endforeach
</tbody>
</table>

    {{ $data->links() }}
</div>

...

```

Setelah itu save dan refresh browser kita, maka akan muncul paging dengan limit per page = 2. Sungguh sangat mudah membuat pagination di Laravel. Hehehe...

Kesimpulan

Gimana teman-teman? Apakah enak menggunakan Laravel untuk aplikasi web/website yang akan kalian buat? Menurut saya, Laravel merupakan framework terbaik saat ini dan terus dikembangkan dengan komunitas yang kuat, sehingga itu menjadi alasan saya sekarang memakai framework ini. Pada tutorial kali ini kita sudah mempelajari bagaimana cara install, konfigurasi, membuat MVC, menarik data, dan menampilkannya di View. Saya rasa tutorial kali ini cukup sampai di sini, karena tangan saya mulai pegel, kita lanjut di part berikutnya ya..

Jika ada pertanyaan, silahkan ditanyakan di kolom komentar. Insya Allah serial tutorial ini akan selalu saya update, mohon doanya agar disela kesibukan saya bisa tetap teguh berbagi membuat serial tutorial ini.

Salam hangat, Cheyuz :)

~ Jagocoding.com

Stand by with Me, @cheyuz,
Jagocoding, learn and share your code!

Tentang Penulis



Cecep Yusuf

Hi, my name is Cecep Yusuf. However, in the virtual world I am more likely to use the name Cheyuz, which is an abbreviation of two words "Cecep" and "Yusuf". I am founder of Jagocoding.com, u can view more of me in Cheyuz.id