

Laravel
4.2

Seminggu Belajar Laravel



Rahmat Awaludin

Seminggu Belajar Laravel

Laravel itu framework PHP yang bikin hidup programmer lebih menyenangkan. Jadi, belajarnya juga mesti menyenangkan.

Rahmat Awaludin

This book is for sale at <http://leanpub.com/seminggubelajarlaravel>

This version was published on 2015-01-02



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](#)

Tweet This Book!

Please help Rahmat Awaludin by spreading the word about this book on [Twitter](#)!

The suggested tweet for this book is:

Hei, gue baru aja download buku Seminggu Belajar Laravel. Keren nih, buat belajar framework Laravel!

The suggested hashtag for this book is [#seminggubelajarlaravel](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#seminggubelajarlaravel>

Untuk istriku tercinta Irna Rahyu dan jagon kecilku Shidqi Abdullah Mubarak.

Contents

Hari 1 : Instalasi dan Konfigurasi Laravel	1
Text Editor	1
Kebutuhan Sistem	1
Homestead	1
Composer	3
Instalasi Laravel	7
Konfigurasi	10
Menjalankan Web Server	10
Ringkasan	13

Hari 1 : Instalasi dan Konfigurasi Laravel

Laravel sangat mudah dikonfigurasi untuk mengembangkan sebuah aplikasi. Pada bagian ini akan saya jelaskan apa saja yang harus dipersiapkan untuk memulai menggunakan framework Laravel. Untuk memudahkan pemahaman, saya tidak akan menjelaskan beberapa istilah secara rinci.

Text Editor

Selama mengembangkan aplikasi di buku ini, saya menggunakan teks editor [Sublime Text](http://www.sublimetext.com) ³ dengan plugin [Laravel Blade Highlighter](http://github.com/Medalink/laravel-blade) ² untuk menampilkan syntax highlighting blade dan [Emmet](http://emmet.io) ³ untuk mempercepat mengetik HTML. Jika Anda lebih menyukai sebuah IDE, saya sarankan menggunakan [PHPStorm](http://www.jetbrains.com/phpstorm) ⁴ atau [AksiIDE](http://aksiide.com) ⁵ karya om [Luri Darmawan](https://www.facebook.com/luridarmawan) ⁶, seorang anak bangsa yang merupakan member yang disegani di grup [PHP Indonesia](https://www.facebook.com/groups/35688476100) ⁷.

Jika Anda akan menggunakan Sublime Text, pastikan untuk menginstal package control agar Anda dapat menginstal package. Panduan cara menginstal package control dapat anda baca di <https://sublime.wbond.net/installation> ⁸.

Untuk menginstal package di Sublime Text dengan package control dapat dibaca di <https://sublime.wbond.net/docs/usage> ⁹.

Kebutuhan Sistem

Laravel mendukung penggunaan web server apache dan nginx. Pada buku ini, saya menggunakan web server Apache. Pastikan PHP yang Anda gunakan sudah versi 5.4 keatas dengan ekstensi MCrypt. Saya sendiri menggunakan [MAMP](http://www.mamp.info) ¹⁰ untuk OSX, jika Anda pengguna windows bisa menggunakan [XAMPP](http://www.apachefriends.org) ¹¹.

Untuk database Laravel dapat menggunakan database MySQL, PostgreSQL, SQLServer atau SQLite.

Homestead

Jika Anda belum menggunakan Homestead, sebaiknya segera belajar menggunakannya. Homestead merupakan package Vagrant yang akan memudahkan Anda membuat virtual machine untuk development aplikasi PHP. Dengan menggunakan homestead, Anda tidak perlu repot-repot mengurus error karena ada extensi PHP yang belum aktif dan error karena masalah web server lainnya.

¹www.sublimetext.com

²<http://github.com/Medalink/laravel-blade>

³<http://emmet.io>

⁴<http://www.jetbrains.com/phpstorm>

⁵<http://aksiide.com>

⁶<https://www.facebook.com/luridarmawan>

⁷<https://www.facebook.com/groups/35688476100>

⁸<https://sublime.wbond.net/installation>

⁹<https://sublime.wbond.net/docs/usage>

¹⁰www.mamp.info

¹¹www.apachefriends.org

Beberapa aplikasi yang sudah disediakan oleh Homestead :

- Ubuntu 14.04
- PHP 5.5
- Nginx
- MySQL
- Postgres
- Node (dengan Bower, Grunt, dan Gulp)
- Redis
- Memcached
- Beanstalkd
- Laravel Envoy
- Fabric + HipChat Extension

Untuk menggunakan homestead Anda perlu menginstall [Virtual Box](#)¹² dan [Vagrant](#)¹³. Silahkan diinstall dua aplikasi tersebut terlebih dahulu.

Selanjutnya ikuti langkah berikut :

1. Install vagrant box untuk homestead dengan perintah

```
1 $ vagrant box add laravel/homestead
```

Perintah ini akan mendownload box atau Virtual Machine ubuntu yang sudah terkonfigurasi dari Internet. Tunggulah hingga prosesnya selesai.

1. Setelah berhasil mendownload box homestead, kini Anda perlu melakukan clone repository github dari homestead. Pastikan Anda sudah menginstall Git di komputer Anda. Jalankan perintah ini di folder yang Anda inginkan bisa di folder yang Anda inginkan.

```
1 $ git clone https://github.com/laravel/homestead.git Homestead
```

1. Agar kita dapat mengakses homestead dengan SSH, kita perlu memberitahu homestead lokasi file ssh kita. Pada folder homestead yang baru di clone, buka file Homestead.yaml. Ubah isian authorize dan keys dengan lokasi file id_rsa Anda. Biasanya settingan default sudah sesuai dengan lokasi id_rsa di komputer kita.
2. Homestead bekerja dengan cara memetakan folder di host OS dengan guest OS (virtual machine). Dengan pemetaan ini, kita dapat coding di host OS dan aplikasi kita tetap dijalankan dari guest OS.

¹²<https://www.virtualbox.org/wiki/Downloads>

¹³<http://www.vagrantup.com/downloads.html>

Cek kembali file `Homestead.yaml`, terlihat ada isian folders dengan isian default `map: ~/Code` dan `to: /home/vagrant/Code`. Pada isian ini dijelaskan folder `~/Code` di host OS akan dipetakan ke folder `/home/vagrant/Code` di guest OS. Silahkan sesuaikan isian `map` dengan lokasi dimana Anda akan menempatkan project Laravel.

Sedikit gambaran untuk pengguna Xamp, karena biasanya project web disimpan di folder `C:\xampp\htdocs` maka Anda cukup mengubah `map: ~/Code` menjadi `map: C:\xampp\htdocs`.

1. Ini bagian yang paling menarik. Setiap web yang berjalan dengan homestead, akan kita akses dengan URL yang kita inginkan. Cek kembali isian `Homestead.yaml`, terdapat isian `sites` dengan isian default `map: homestead.app` dan `to: /home/vagrant/Code/Laravel/public`.

Maksud dari isian ini adalah ketika di host OS kita mengakses `http://homestead.app:8000` maka homestead akan melayani web dari folder `/home/vagrant/Code/Laravel/public`. Karena folder `/home/vagrant/Code` merupakan `map` dari `~/Sites`, maka URL ini sebenarnya mengambil file dari `~/Sites/Laravel/public`.

Jadi, jika Anda membuat aplikasi Laravel di `~/Sites/webapp` dan ingin mengaksesnya dengan URL `http://webapp.site:8000`. Langkah yang Anda lakukan adalah menambahkan isian di `sites` ini. Dengan isian `map: webapp.site` dan `to: /home/vagrant/Code/webapp/public`.

2. Agar URL yang kita buat di tahap 4 dapat diakses, kita perlu menambah isian `host file`. Di *nix Anda dapat mengubahnya di `/etc/hosts`, sedangkan di Windows ada di `C:\windows\system32\drivers\etc\hosts`. Tambahkan setiap `hostname` yang kita buat dan petakan ke `127.0.0.1`. Misalnya:

```
1 webapp.site 127.0.0.1
```

1. Terakhir, untuk menjalankan homestead, masuk ke folder homestead dan jalankan perintah:

```
1 $ vagrant up
```

Dengan homestead, Anda tidak akan direpotkan dengan konfigurasi sistem yang berbeda tiap anggota tim. Untuk mengetahui, password default dan konfigurasi lebih lanjut tentang homestead, kunjungi <http://laravel.com/docs/homestead>

Composer

Untuk menginstall laravel kita akan menggunakan composer. Composer adalah aplikasi yang digunakan untuk mengatur package-package dalam mengembangkan sebuah web dengan PHP. Jika dulu, mungkin Anda mengenal yang namanya [PEAR](http://pear.php.net)¹⁵, composer tuh mirip-mirip PEAR lah.

Anggaplah kita belum kenal dengan PEAR/Composer. Jika kita akan mengembangkan sebuah aplikasi web dan membutuhkan library untuk user management misalnya 'UserAuth' maka kita akan download dari webnya, letakkan di folder tertentu (misalnya library), kemudian me-load dengan `require` atau `include` pada class yang kita butuhkan.

Setidaknya ada beberapa masalah dari solusi ini:

¹⁴<http://laravel.com/docs/homestead>

¹⁵pear.php.net

- Bagaimana jika web kita membutuhkan tidak hanya satu library, tapi 40 library? Mau download satu-persatu?
- Bagaimana jika library UserAuth bergantung dengan library lain? misalnya SessionManager dan SessionManager juga bergantung kepada library Session. Dan seterusnya, dan seterusnya..

Pada Laravel, kita akan menggunakan composer tidak hanya untuk menginstall library, tapi framework Laravel itu sendiri diinstall menggunakan composer.

Untuk memahami composer lebih lanjut, Anda dapat mengunjungi [dokumentasi resmi composer](#)¹⁶.

Install Composer

Instalasi composer agak berbeda untuk OS *nix (Linux, OSX, dll) dan Windows, saya jelaskan masing-masing:

Windows

Cukup download [composer-setup.exe](#)¹⁷ dan jalankan file instalasi.

*nix

Jalankan terminal dan masukkan perintah berikut:

- 1 `$ curl -sS https://getcomposer.org/installer | php`
- 2 `$ sudo mv composer.phar /usr/local/bin/composer`

Penggunaan Composer

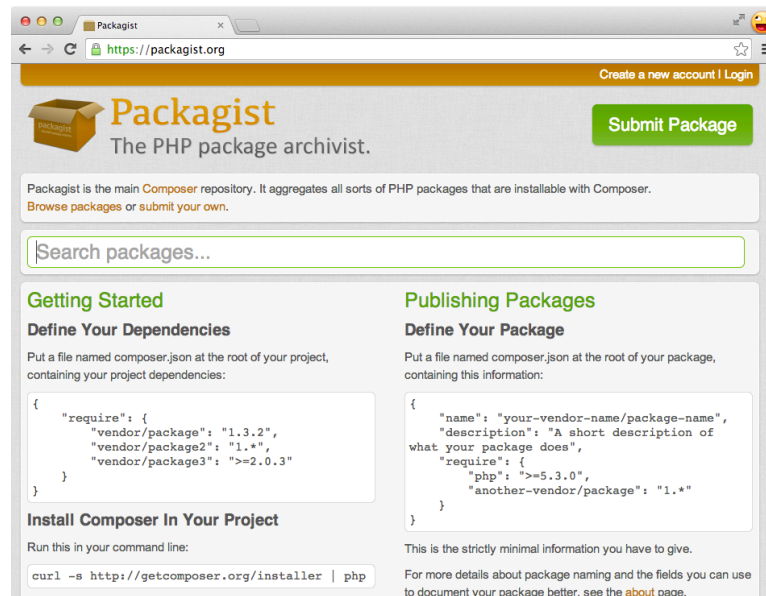
Secara default, composer akan menggunakan package yang teregister di [packagist.org](#)¹⁸. Tentunya, kita juga dapat menyiapkan repositori package private menggunakan [satis](#)¹⁹.

¹⁶<https://getcomposer.org/doc/00-intro.md>

¹⁷<https://getcomposer.org/Composer-Setup.exe>

¹⁸<http://packagist.org>

¹⁹<https://github.com/composer/satis>



Packagist.org

Composer menggunakan file dengan [format JSON](http://json.org)²⁰. JSON merupakan format standar untuk menyimpan data name⇒value yang sudah sangat umum digunakan untuk transfer data. Contoh syntax JSON terlihat seperti ini:

contoh struktur JSON

```
1 {
2     name1 : {
3         subname1 : value,
4         subname2 : value
5     },
6     name2 : value
7 }
```

Composer menggunakan format json ini pada file bernama composer . json. Berikut contoh isi composer . json :

composer.json

```
1 {
2     "require": {
3         "monolog/monolog": "1.0.*"
4     }
5 }
```

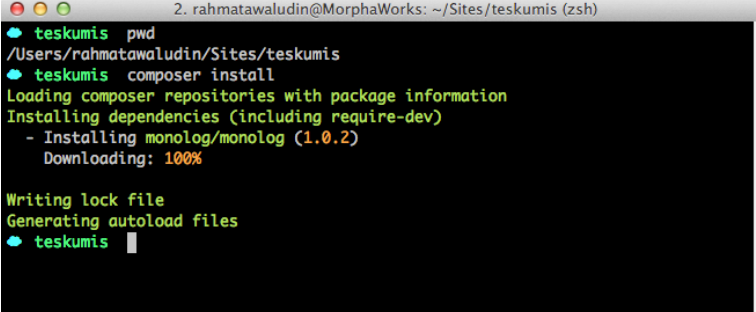
Sebagaimana terlihat pada syntax diatas, pada bagian require kita masukkan nama package yang kita butuhkan (monolog/monolog) dan versi yang diinginkan (1.0.*).

²⁰<http://json.org>

Install Package

Untuk menginstall package dengan composer, pindahkan file `composer.json` diatas ke sebuah folder. Lalu jalankan perintah berikut di dalam folder tersebut:

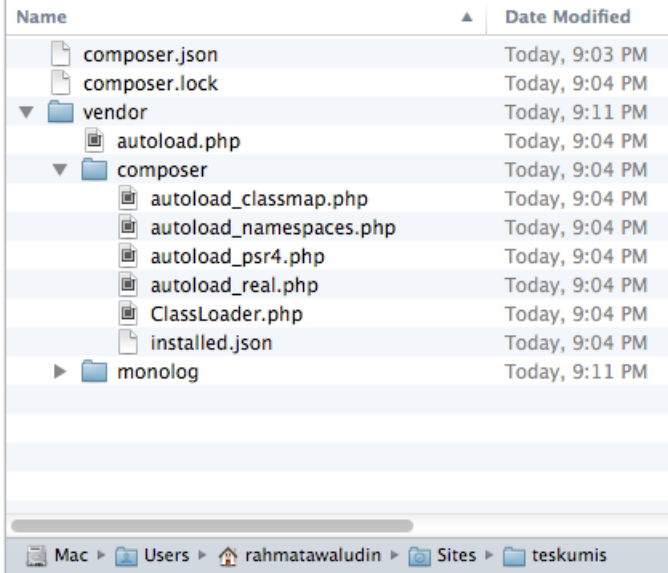
```
1 $ composer install
```



```
2. rahmatawaludin@MorphaWorks: ~/Sites/teskumis (zsh)
teskumis$ pwd
/Users/rahmatawaludin/Sites/teskumis
teskumis$ composer install
Loading composer repositories with package information
Installing dependencies (including require-dev)
- Installing monolog/monolog (1.0.2)
  Downloading: 100%
Writing lock file
Generating autoload files
teskumis$
```

composer install

Perintah diatas akan melakukan instalasi package aplikasi yang kita tulis di bagian `require`. Setelah dieksekusi strukur folder kita akan berubah:



Name	Date Modified
composer.json	Today, 9:03 PM
composer.lock	Today, 9:04 PM
▼ vendor	Today, 9:11 PM
autoload.php	Today, 9:04 PM
▼ composer	Today, 9:04 PM
autoload_classmap.php	Today, 9:04 PM
autoload_namespaces.php	Today, 9:04 PM
autoload_psr4.php	Today, 9:04 PM
autoload_real.php	Today, 9:04 PM
ClassLoader.php	Today, 9:04 PM
installed.json	Today, 9:04 PM
▶ monolog	Today, 9:11 PM

composer structure

- Folder `vendor` menyimpan package yang dibutuhkan, sebagaimana yang ditulis di bagian `require`
- File `vendor/autoload.php` dapat digunakan untuk mendapatkan fitur autoloading.
- File `composer.lock` berfungsi mencatat versi package yang saat ini sedang kita gunakan, jangan hapus/edit file ini, karena perintah `composer install` bergantung pada file ini.

Update package

Jika package baru telah ditambah pada bagian `require` atau versi package yang digunakan dirubah, gunakan perintah ini untuk memperbaharui package yang kita gunakan:

```
1 $ composer update
```

Perintah composer lainnya dapat dilihat dengan perintah :

```
1 $ composer --help
```

Atau cek di [manual composer](https://getcomposer.org/doc)²¹.

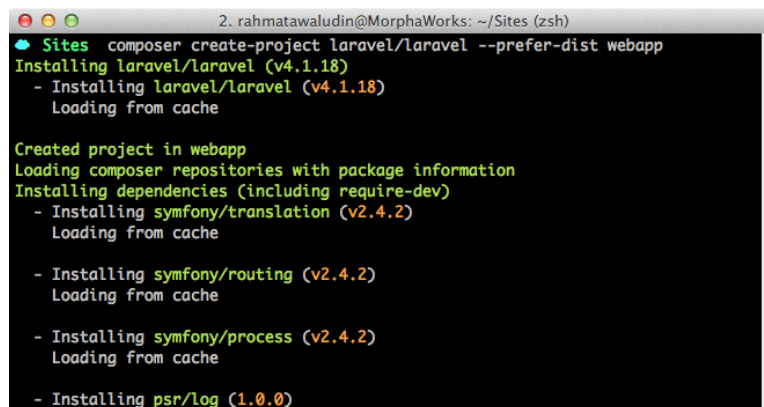
Instalasi Laravel

Sebagaimana disampaikan di bagian sebelumnya, Laravel diinstall menggunakan composer. Gunakan perintah ini untuk membuat project laravel di folder webapp dengan versi laravel 4.2:

```
1 $ composer create-project laravel/laravel=4.2 --prefer-dist webapp
```

Jika Anda ingin menginstal versi terbaru dari Laravel, gunakan perintah:

```
1 $ composer create-project laravel/laravel --prefer-dist webapp
```



```
2. rahmatawaludin@MorphaWorks: ~/Sites (zsh)
Sites composer create-project laravel/laravel --prefer-dist webapp
Installing laravel/laravel (v4.1.18)
- Installing laravel/laravel (v4.1.18)
  Loading from cache

Created project in webapp
Loading composer repositories with package information
Installing dependencies (including require-dev)
- Installing symfony/translation (v2.4.2)
  Loading from cache

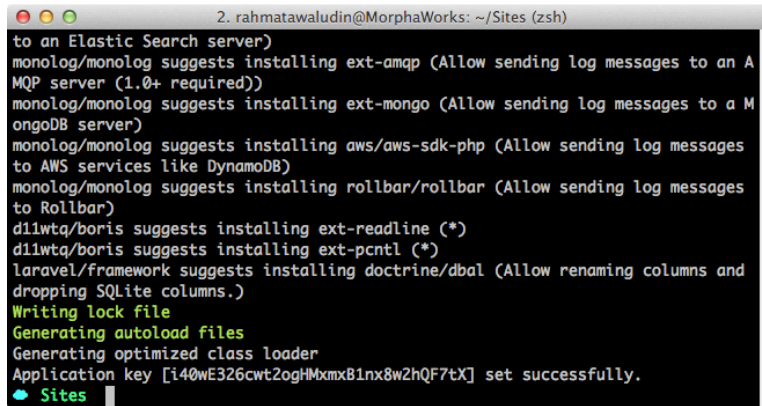
- Installing symfony/routing (v2.4.2)
  Loading from cache

- Installing symfony/process (v2.4.2)
  Loading from cache

- Installing psr/log (1.0.0)
```

install laravel part 1

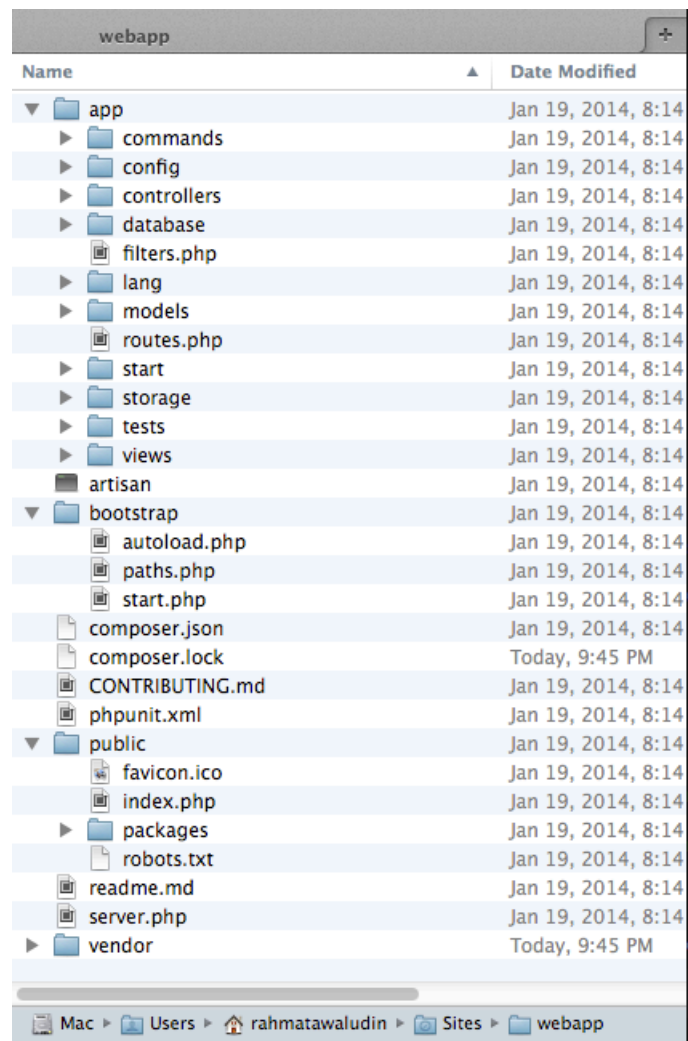
²¹<https://getcomposer.org/doc>

A terminal window with a dark background and light green text. The window title is '2. rahmatawaludin@MorphaWorks: ~/Sites (zsh)'. The output shows various suggestions from Composer for installing extensions like ext-amqp, ext-mongo, aws/aws-sdk-php, rollbar/rollbar, ext-readline, and ext-pcntl. It also shows suggestions from Laravel/framework for installing doctrine/dbal. The final steps shown are 'Writing lock file', 'Generating autoload files', 'Generating optimized class loader', and 'Application key [i40wE326cwt2oghHmx8B1nx8w2hQF7tX] set successfully.' followed by a green prompt 'Sites ' with a cursor.

```
to an Elastic Search server)
monolog/monolog suggests installing ext-amqp (Allow sending log messages to an AMQP server (1.0+ required))
monolog/monolog suggests installing ext-mongo (Allow sending log messages to a MongoDB server)
monolog/monolog suggests installing aws/aws-sdk-php (Allow sending log messages to AWS services like DynamoDB)
monolog/monolog suggests installing rollbar/rollbar (Allow sending log messages to Rollbar)
d11wtq/boris suggests installing ext-readline (*)
d11wtq/boris suggests installing ext-pcntl (*)
laravel/framework suggests installing doctrine/dbal (Allow renaming columns and dropping SQLite columns.)
Writing lock file
Generating autoload files
Generating optimized class loader
Application key [i40wE326cwt2oghHmx8B1nx8w2hQF7tX] set successfully.
Sites 
```

install laravel part 2

Perintah ini akan menginstall framework laravel dan dependency packagenya.



laravel structure

Jika Anda menggunakan OS *nix, instalasi laravel dapat pula dilakukan dengan menggunakan `laravel.phar`, caranya:

1. Download [Laravel phar](http://laravel.com/laravel.phar)²²

```
$ wget http://laravel.com/laravel.phar
```

2. Rename file yang telah didownload menjadi `laravel`

```
$ mv laravel.phar laravel
```

3. Pindahkan file tersebut ke `/usr/local/bin/`:

²²<http://laravel.com/laravel.phar>

```
$ sudo mv laravel /usr/local/bin/
```

4. Untuk membuat project laravel, jalan perintah :

```
$ laravel new webapp
```

Konfigurasi

Setelah Laravel terinstall pastikan folder app/storage dapat diakses oleh web server. Cara sederhananya, jalankan perintah ini:

```
$ sudo chmod -R 777 app/storage
```

Jangan lupa isi juga konfigurasi database Anda (nama database, username, password) di app/config/database.php. Tentunya database harus Anda buat sendiri di aplikasi database yang Anda gunakan.

Untuk memudahkan selama pengembangan, aktifkan mode debug di app/config/app.php ubah isian debug menjadi true.

Menjalankan Web Server

Web yang dikembangkan dengan Laravel dapat diakses menggunakan PHP builtin web server atau virtual host.

PHP builtin web server

Jalankan perintah berikut di folder webapp:

```
$ php artisan serve
```

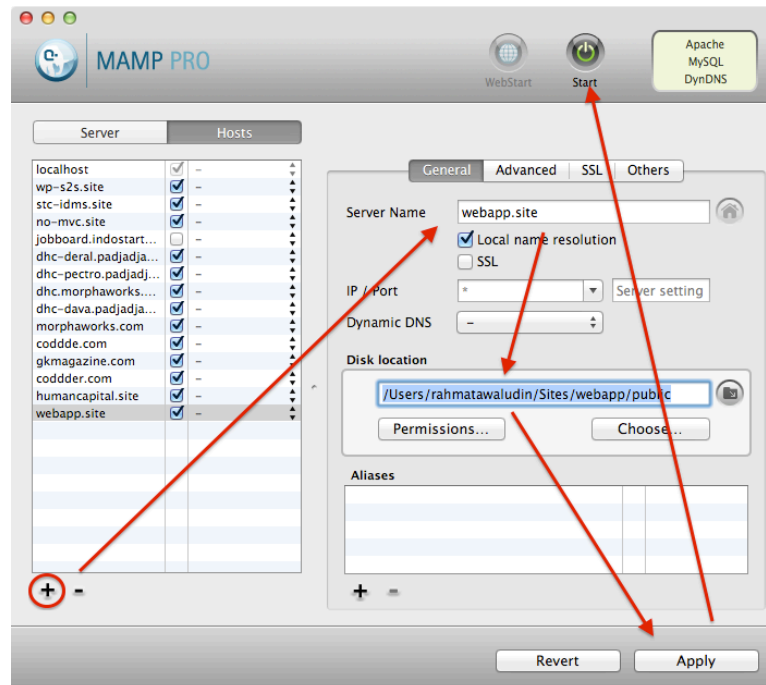
Setelah Anda menjalankan perintah diatas, aplikasi laravel dapat diakses di <http://localhost:8000>. Kekurangan dari PHP Builtin web server adalah ketika terjadi perubahan pada source code, terkadang server harus di restart.

Virtual Host

Menggunakan virtual host, aplikasi dapat diakses dengan url seperti <http://webapp.site>, <http://www.webapp.com>, dan sebagainya walupun masih berada di lokal. Saya sendiri lebih sering menggunakan virtualhost ketika mengembangkan web dengan Laravel. Berikut cara membuat virtual host:

MAMP

1. Menggunakan MAMP Pro, buka menu **Hosts**
2. Klik tombol **[+]**
3. Isi bagian **Server name** dengan url yang kita inginkan
4. Isi **Disk Location** dengan alamat folder public di webapp
5. Klik **Apply**
6. Klik **Start** untuk merestart server apache.



Setup VirtualHost di MAMP PRO

XAMPP

1. Buka file hosts yang ada di alamat C:\WINDOWS\system32\drivers\etc\hosts
2. Di bagian paling bawah tambahkan alamat IP Address localhost 127.0.0.1 dan nama domain yang dibuat misalnya webapp.site

C:\WINDOWS\system32\drivers\etc\hosts

```

1  ....
2  127.0.0.1    webapp.site
3  ....

```

3. Buka file httpd.conf yang ada di alamat C:\xampp\apache\conf\httpd.conf
4. Cari bagian Directory, jika aplikasi kita berada di C:/xampp/htdocs/webapp isi seperti ini

C:\xampp\apache\conf\httpd.conf

```
1 <Directory "C:/xampp/htdocs/webapp/public">
2     Options Indexes FollowSymLinks Includes ExecCGI
3     AllowOverride All
4     Order allow,deny
5     Allow from all
6     Require all granted
7 </Directory>
```

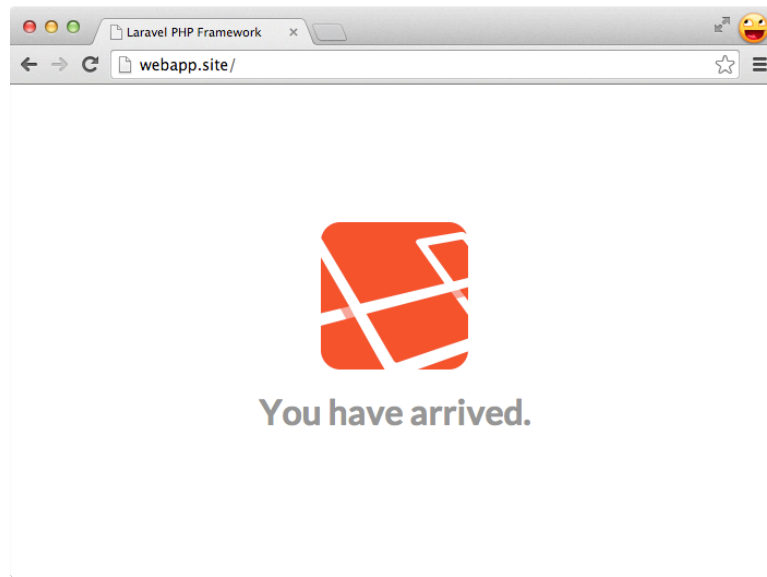
5. Buka file httpd-vhosts.conf yang ada di alamat C:\xampp\apache\conf\extra
6. Tambahkan setingan di bawah ini untuk membedakan website yang dipanggil dengan localhost dan website yang dipanggil dengan virtual host

C:\xampp\apache\conf\extra\httpd-vhosts.conf

```
1 NameVirtualHost *:80
2
3 #VirtualHost untuk webapp.site
4
5 <VirtualHost *:80>
6     DocumentRoot C:/xampp/htdocs/webapp/public
7     ServerName webapp.site
8 </VirtualHost>
9
10 #Untuk localhost yang biasa
11
12 <VirtualHost *:80>
13     DocumentRoot C:/xampp/htdocs
14     ServerName localhost
15 </VirtualHost>
```

7. Restart Apache pada XAMPP Control Panel dengan klik tombol **stop** kemudian klik tombol **start**.

Setelah berhasil, Anda dapat mengakses aplikasi di <http://webapp.site>.



Berhasil setup virtualhost

Ringkasan

Di Hari 1 ini, saya harap Anda telah memahami bagaimana melakukan setup sebuah project Laravel, poin-poin yang telah kita bahas yaitu:

- Text Editor yang digunakan
- Penggunaan composer untuk development php modern
- Instalasi laravel
- Konfigurasi virtualhost

Pada hari 2 kita akan mempelajari konsep Routing dan MVC pada Laravel. Semangat! :)