



# 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 work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

## **Tweet This Book!**

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

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

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=#seminggubelajararavel>

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

# Contents

<b>Hari 1 : Instalasi dan Konfigurasi Laravel</b> . . . . .	<b>1</b>
Kebutuhan Sistem . . . . .	1
Composer . . . . .	1
Instalasi Laravel . . . . .	4
Konfigurasi . . . . .	7
Menjalankan Web Server . . . . .	7
Ringkasan . . . . .	10

# 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.

## 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 5.3.7 keatas. Saya sendiri menggunakan [MAMP](http://www.mamp.info)<sup>1</sup> untuk OSX, jika Anda pengguna windows bisa menggunakan [XAMPP](http://www.apachefriends.org)<sup>2</sup>.

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

## 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)<sup>3</sup>, 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:

- 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](https://getcomposer.org/doc/00-intro.md)<sup>4</sup>.

## Install Composer

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

---

<sup>1</sup>[www.mamp.info](http://www.mamp.info)

<sup>2</sup>[www.apachefriends.org](http://www.apachefriends.org)

<sup>3</sup>[pear.php.net](http://pear.php.net)

<sup>4</sup><https://getcomposer.org/doc/00-intro.md>

## Windows

Cukup download [composer-setup.exe](#)<sup>5</sup> dan jalankan file instalasi.

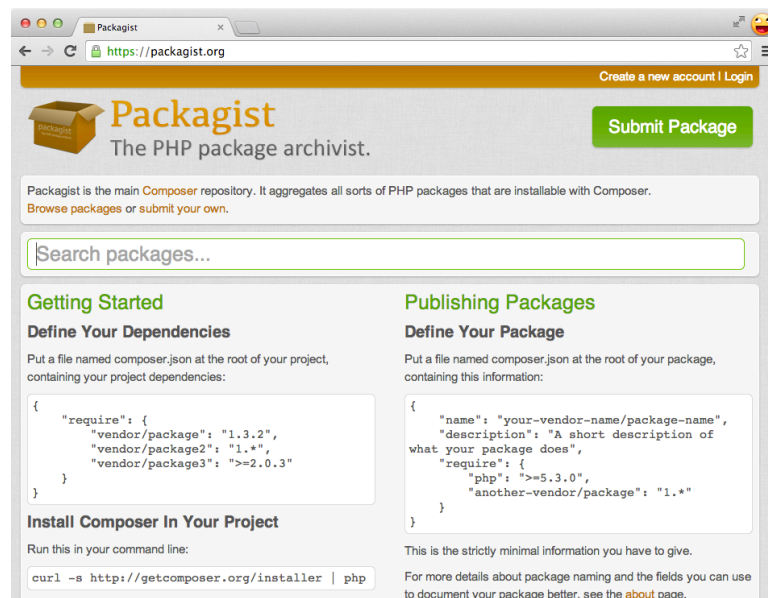
## \*nix

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](#)<sup>6</sup>. Tentunya, kita juga dapat menyiapkan repositori package private menggunakan [satis](#)<sup>7</sup>.



Packagist.org

Composer menggunakan file dengan [format JSON](#)<sup>8</sup>. JSON merupakan format standar untuk menyimpan data name⇒value yang sudah sangat umum digunakan untuk transfer data. Contoh syntax JSON terlihat seperti ini:

---

<sup>5</sup><https://getcomposer.org/Composer-Setup.exe>

<sup>6</sup><http://packagist.org>

<sup>7</sup><https://github.com/composer/satis>

<sup>8</sup><http://json.org>

#### 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

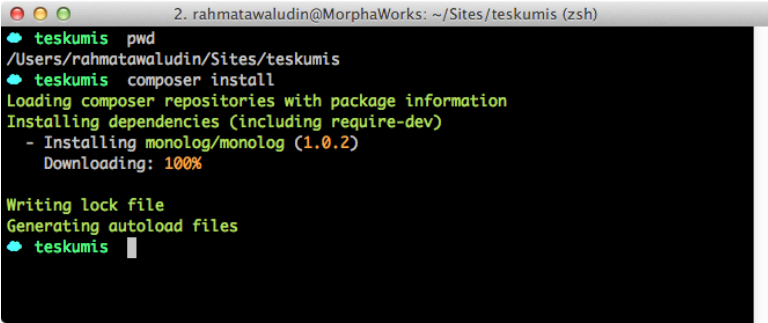
```
1 {  
2     "require": {  
3         "monolog/monolog": "1.0.*"  
4     }  
5 }
```

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

## 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
```

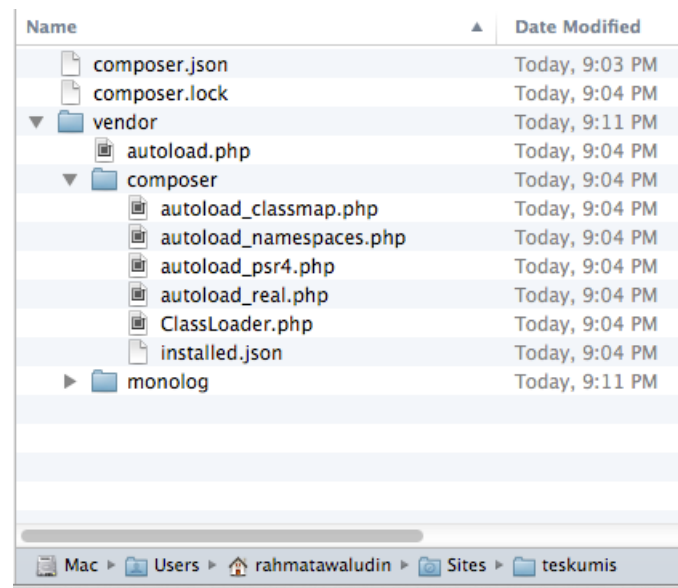


```
2. rahmatawaludin@MorphaWorks: ~/Sites/teskumis (zsh)  
teskumis pwd  
/Users/rahmatawaludin/Sites/teskumis  
teskumis composer install  
Loading composer repositories with package information  
Installing dependencies (including require-dev)  
- Installing monolog/monolog (1.0.2)  
  Downloading: 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 strutur 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)<sup>9</sup>.

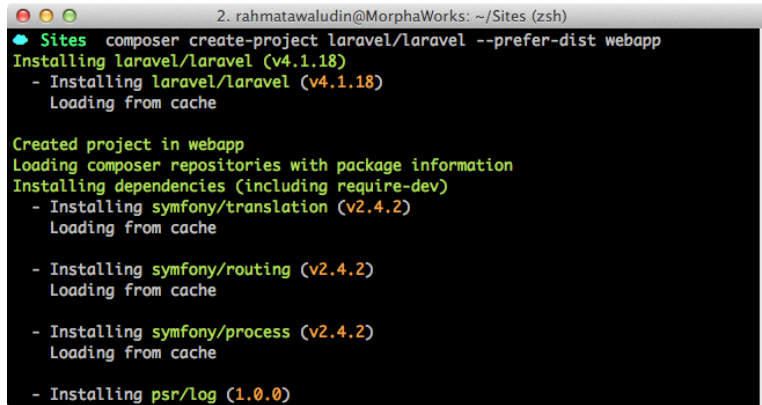
## Instalasi Laravel

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

---

<sup>9</sup><https://getcomposer.org/doc>

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

A terminal window with a dark background and light green text. The title bar shows the user '2. rahmatawaludin@MorphaWorks' and the directory '~/Sites (zsh)'. The output of the command shows the installation of Laravel v4.1.18, which is loaded from the cache. It then creates the project in 'webapp', loads composer repositories, and installs dependencies: symfony/translation (v2.4.2), symfony/routing (v2.4.2), symfony/process (v2.4.2), and psr/log (1.0.0), all of which are also loaded from the cache.

```
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

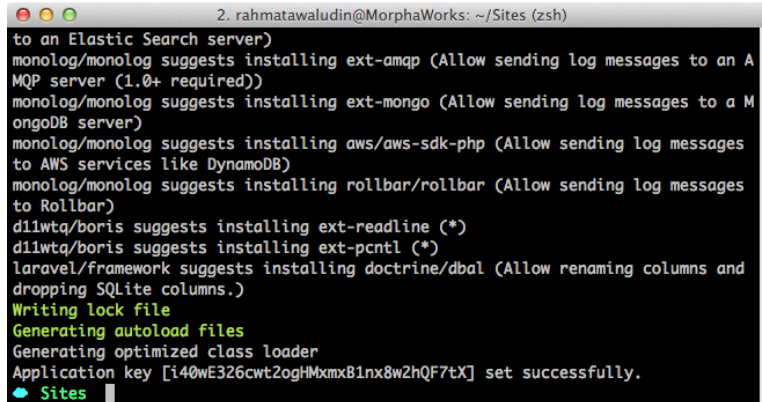
Created project in webapp
Loading composer repositories with package information
Installing dependencies (including require-dev)
- Installing symfony/translation (v2.4.2)
  Loading from cache

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

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

- Installing psr/log (1.0.0)
```

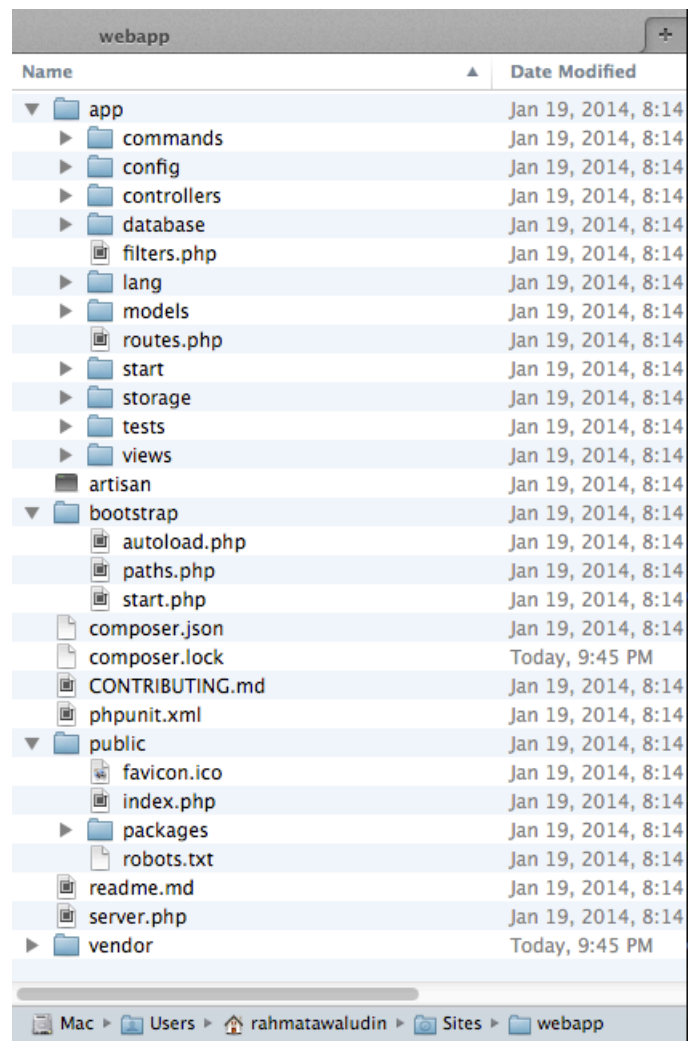
install laravel part 1

A terminal window with a dark background and light green text. The title bar shows the user '2. rahmatawaludin@MorphaWorks' and the directory '~/Sites (zsh)'. The output continues from the previous part, showing suggestions from various packages like monolog/monolog and d11wtq/boris. It then writes the lock file, generates autoload files, and generates an optimized class loader. Finally, it sets the application key successfully.

```
2. rahmatawaludin@MorphaWorks: ~/Sites (zsh)
to an Elastic Search server)
monolog/monolog suggests installing ext-amqp (Allow sending log messages to an A
MQP server (1.0+ required))
monolog/monolog suggests installing ext-mongo (Allow sending log messages to a M
ongoDB 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)<sup>10</sup>

```
$ 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/`:

---

<sup>10</sup><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.

## 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
```

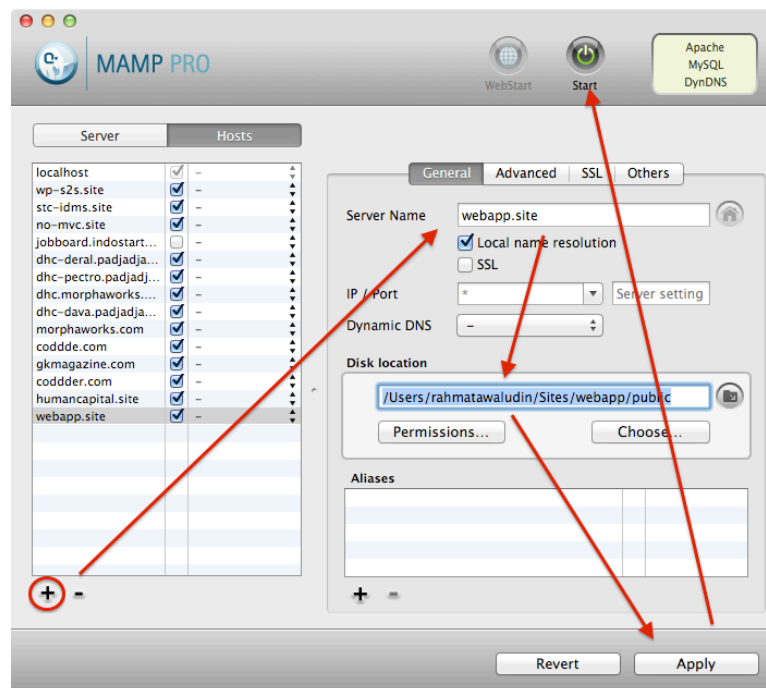
Setelah Anda menjalankan perintah diatas, aplikasi laravel dapat diakses di http://localhost:8000. Kekurangan dari PHP Builtin web server adalah ketika terjadi perubahan pada source code, terkadang server harus di restart.

### 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

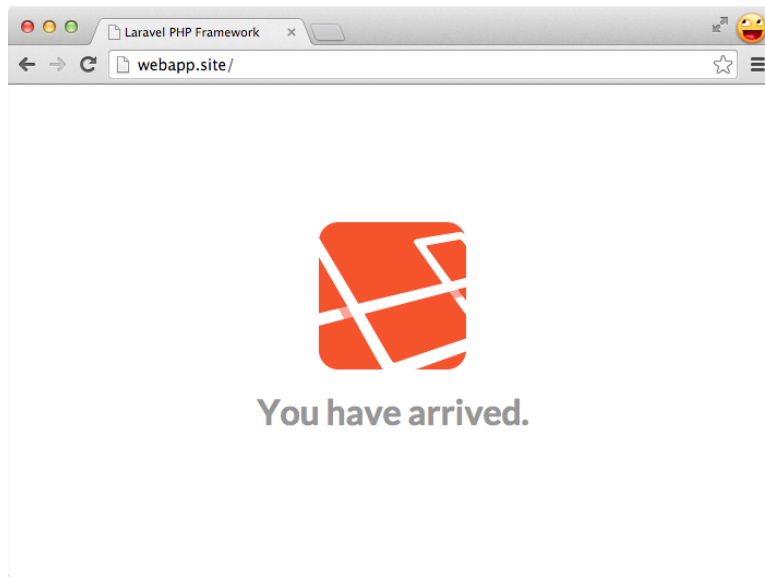
---

```
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

Pada hari 1 ini, saya harap Anda telah dapat memahami:

- Penggunaan composer untuk development php modern
- Instalasi laravel
- Konfigurasi virtualhost

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