

1. Instalasi dan Konfigurasi Laravel

Laravel sangat mudah dikonfigurasi untuk mengembangkan web app. Pada bagian ini kita akan bahas apa saja yang harus dipersiapkan untuk memulai menggunakan framework Laravel. Untuk memudahkan pembahasan, kita akan menggunakan penjelasan yang sesederhana mungkin untuk memahami setiap teknik atau istilah yang sering digunakan di Laravel.

1.1 Text Editor

Dalam membuat aplikasi Laravel, saya sarankan menggunakan teks editor [Sublime Text 3](http://www.sublimetext.com)¹ 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 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. Saya sendiri, menggunakan [Vim](http://www.vim.org)⁸ (lebih tepatnya [neovim](https://neovim.io/)⁹) selama menulis buku ini.

Jika akan menggunakan Sublime Text, pastikan untuk menginstal package control agar dapat menginstal package. Panduan cara menginstal package control dapat dibaca di <https://sublime.wbond.net/installation>¹⁰.

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 <http://www.vim.org>

⁹ <https://neovim.io/>

¹⁰ <https://sublime.wbond.net/installation>

Untuk menginstal package di Sublime Text dengan package control dapat dibaca di [dokumentasi](#)¹¹.



Tahapan instalasi plugin Sublime Text ini opsional. Jika bingung, tidak perlu pusing memikirkan langkah ini. Meskipun tidak dilakukan, proses belajar Laravel tidak akan terganggu. Gunakan text editor yang paling dikuasai saja.

1.2 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 5.6.4 keatas

dengan ekstensi OpenSSL, PDO, Mbstring dan Tokenizer. Saya sendiri menggunakan [MAMP](#)¹² untuk OSX, jika Anda pengguna windows bisa menggunakan [XAMPP](#)¹³ (pastikan selalu menggunakan versi terbaru).

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

1.3 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](#)¹⁴, 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.

¹¹<https://sublime.wbond.net/docs/usage>

¹²www.mamp.info

¹³www.apachefriends.org ¹⁴ pear.php.net

Setidaknya ada beberapa masalah dari solusi ini:

- Bagaimana jika web kita membutuhkan lebih dari satu library, misalnya 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..

Untuk memahami composer lebih lanjut, kita dapat mengunjungi [dokumentasi resmi composer](#)¹⁵.

1.3.1 Install Composer

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

1.3.1.1 Windows

Cukup download [composer-setup.exe](#)¹⁶ dan jalankan file instalasi.

1.3.1.2 Unix

Jalankan terminal dan masukkan perintah berikut:

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

Pada windows maupun unix kita dapat mengetes apakah composer sudah terinstall dengan

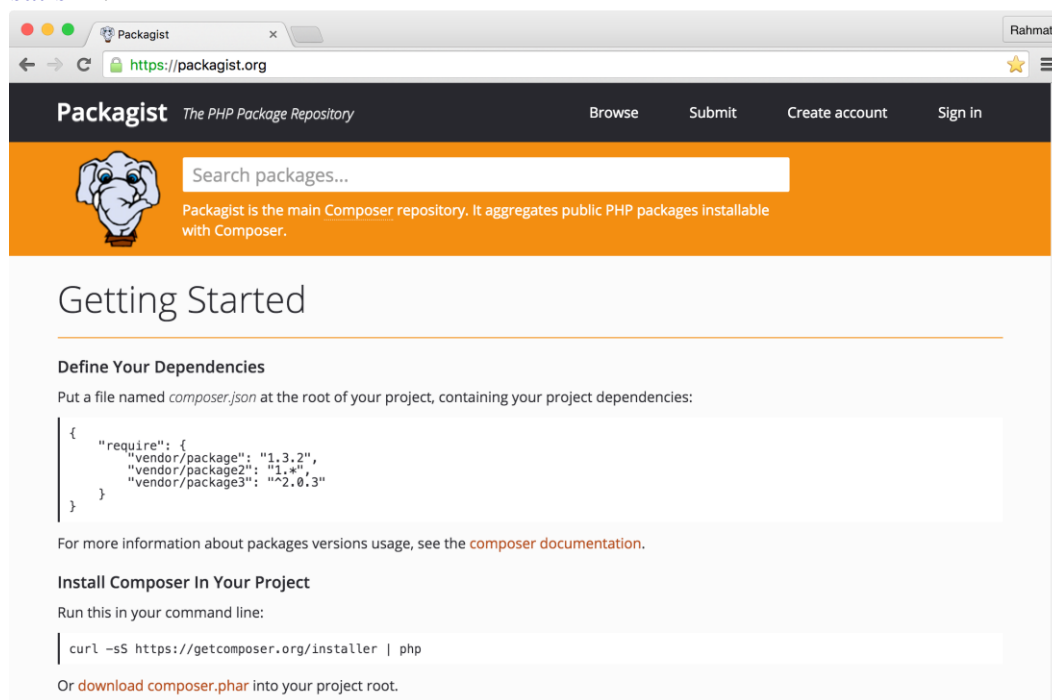
perintah: `composer -V`

¹⁵ <https://getcomposer.org/doc/00-intro.md>

¹⁶ <https://getcomposer.org/Composer-Setup.exe>

1.3.2 Penggunaan Composer

Secara default, composer akan menggunakan package yang teregister di packagist.org¹⁷ . Selain packagist, kita juga dapat menyiapkan repositori package private menggunakan [satis](#)¹⁸ .



Packagist.org

Composer menggunakan file dengan format **JSON**¹⁹ . JSON merupakan format standar untuk menyimpan data `name => value` yang umum digunakan untuk transfer data. Contoh syntax JSON seperti ini:

-
- 17 <http://packagist.org>
 - 18 <https://github.com/composer/satis>
 - 19 <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` (file ini harus kita buat). Berikut contoh isi `composer.json` : `composer.json`

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

File `composer.json` berfungsi untuk menentukan library apa saja yang akan kita install pada project PHP yang kita buat. Pada syntax diatas, pada bagian `require` kita masukkan nama `package` yang kita butuhkan (`monolog/monolog`) dan `versi` yang diinginkan (`1.0.*`).

Monolog adalah sebuah library yang biasa digunakan untuk bekerja dengan file log. Disini, instalasi Monolog hanya sebagai contoh penggunaan composer untuk menginstal library PHP, tidak ada hubungannya dengan Laravel.

1.3.2.1 Install Package

Untuk menginstall package dengan composer, pindahkan file `composer.json` diatas ke sebuah folder. Disini saya akan menggunakan folder di `~/Code/coba-composer`.

Untuk pengguna Windows, misalnya dapat menggunakan `C:\coba-composer`. Lalu jalankan perintah berikut di dalam folder tersebut:

```
composer install
```

```
2. rahmatawaludin@MalesCast: ~/Code/coba-composer (zsh)
> ls
composer.json

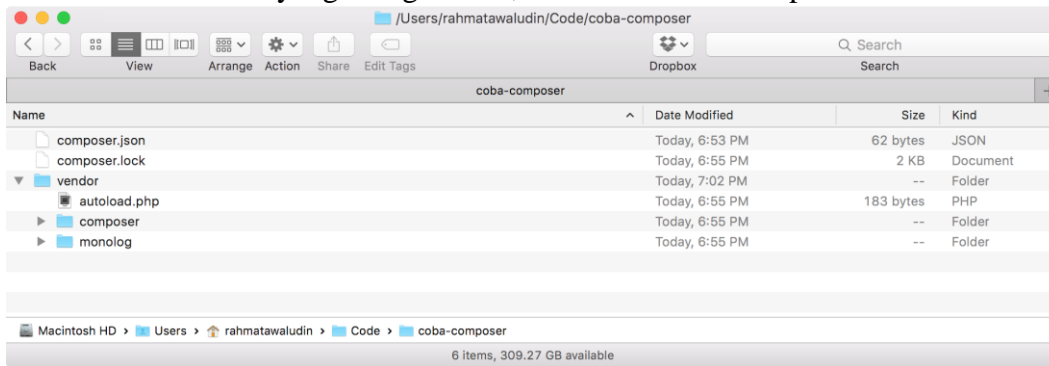
~/Code/coba-composer
> composer install
Loading composer repositories with package information
Installing dependencies (including require-dev)
- Installing monolog/monolog (1.0.2)
  Loading from cache

Writing lock file
Generating autoload files

~/Code/coba-composer 1m 4s
> _
```

composer install

Perintah diatas akan melakukan instalasi package aplikasi yang kita tulis di bagian `require`. Cobalah buka folder yang kita gunakan, akan terlihat isian seperti berikut:



Isi folder setelah composer install

- 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.

1.3.2.2 Update package

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

```
composer update
```

Untuk menginstall package baru, kita juga dapat menjalankan perintah:

```
composer require <nama-package>
```

Perintah composer lainnya dapat dilihat dengan perintah :

```
composer --help
```

Atau cek di [manual composer](#)²⁰ .

Untuk memahami lebih jauh tentang composer, bisa juga menonton video yang telah saya buat di [Malescast](#)²¹.

1.4 Instalasi Laravel

Sebagaimana disampaikan di bagian sebelumnya, Laravel diinstall menggunakan composer. Gunakan perintah ini untuk membuat project laravel di folder webapp dengan versi laravel 5.3:

²⁰ <https://getcomposer.org/doc>

²¹<https://malescast.com/Teknik-PHP-Modern/dependency-management-dengan-composer>

```
1 composer create-project laravel/laravel=5.3 --prefer-dist webapp
```



Jika kita ingin menginstall versi terbaru dari Laravel, gunakan perintah:

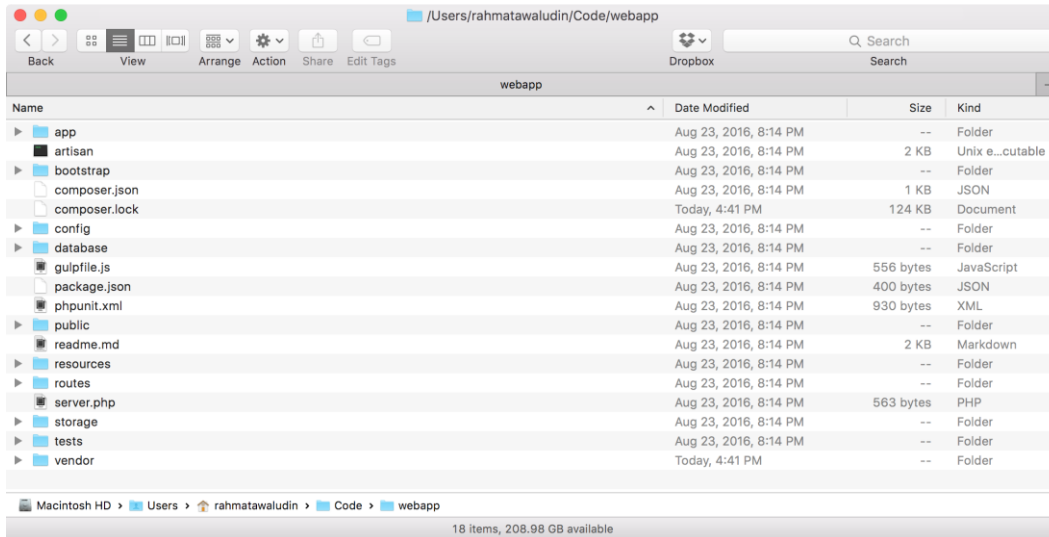
```
i$ composer create-project laravel/laravel --prefer-dist webapp
```

Akan muncul output seperti ini:

```
Installing laravel/laravel (v5.3.0)
- Installing laravel/laravel (v5.3.0) Downloading: 100%

Created project in webapp
> php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev) ....
....
Writing lock file
Generating autoload files
> Illuminate\Foundation\ComposerScripts::postUpdate
> php artisan optimize
Generating optimized class loader
> php artisan key:generate
Application key [base64:++HHOg9mCiJvBEWYkPB9NwZMAIDBNm7dZGHmDPqAJKrk=] set successfully.
```

Jika Anda telah mendapatkan output seperti ini, itu tandanya Laravel telah berhasil didownload. Ini struktur folder yang akan kita dapatkan:



Struktur folder Laravel

Kita juga dapat mengecek versi Laravel yang terinstall dengan perintah:

```
php artisan -V
// Laravel Framework version 5.3.4
```

Terlihat pada output diatas, kita mendapatkan versi 5.3.4.

1.5 Konfigurasi

Setelah Laravel terinstall pastikan folder `storage` dan `bootstrap/cache` dapat diakses oleh web server. Seharusnya, ini sudah diset dengan benar. Jika menggunakan unix, kita dapat menjalankan perintah berikut untuk mengeset ulang hak akses kedua folder tersebut

```
sudo chmod -R 777 storage sudo chmod -R
777 bootstrap/cache
```

Semua konfigurasi Laravel disimpan di folder `config`. Misalnya, konfigurasi database akan disimpan di file `config/database.php`.

Dalam mengisi file konfigurasi Laravel, kita dapat langsung mengisinya atau menggunakan file `.env`. Misalnya, untuk mengisi konfigurasi database pada koneksi dengan nama `mysql`, kita akan menemui isian seperti ini:

```
config/database.php
```

```
'database' => env('DB_DATABASE',  
    'forge'),  
'username' => env('DB_USERNAME',  
    'forge'),  
'password' => env('DB_PASSWORD', ''),
```

Untuk mengisi ketiga field tersebut, kita dapat langsung mengubah isinya atau menggunakan file `.env`. Caranya, kita buat isian seperti ini (sesuaikan dengan konfigurasi database):

`.env`

```
DB_DATABASE=Larapus DB_USERNAME=rahmat  
DB_PASSWORD=r4ha51a
```

Dalam development yang menggunakan version control misalnya Git, file ini tidak akan di commit. Sehingga, isian file `.env` akan berbeda tergantung environment mesin tiap developer.



Git?

Git merupakan salah satu tools yang bisa kita gunakan untuk menyimpan source code dengan lebih terstruktur. Jika belum paham Git, sebaiknya segera belajar setelah menyelesaikan buku ini. Setiap developer saat ini, wajib menguasai Git.

1.6 PHP builtin web server

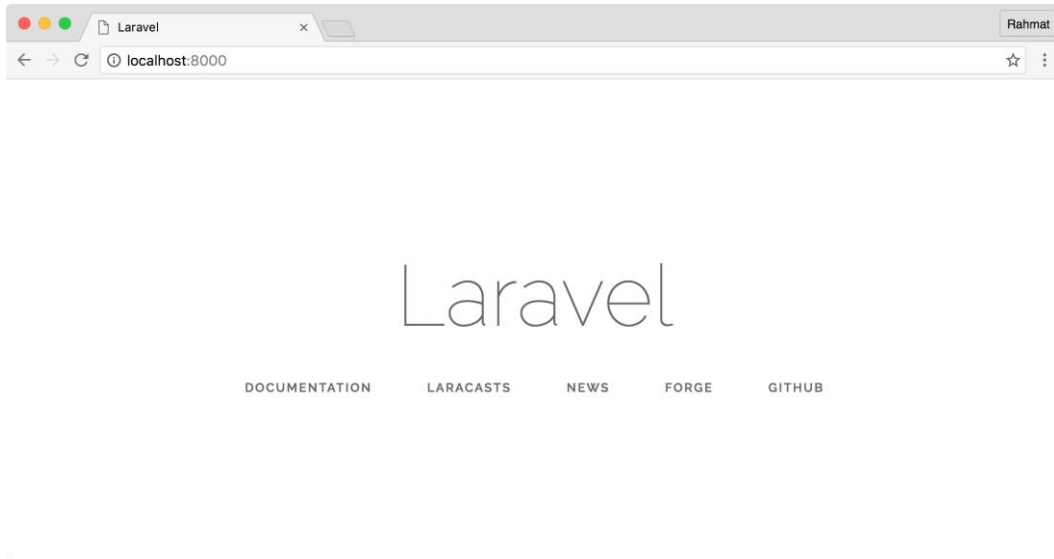
Selama proses pengembangan, kita dapat menggunakan builtin web server untuk mengakses Laravel. Dengan teknik ini, kita tidak perlu menggunakan web server seperti Apache atau Nginx.

Jalankan perintah berikut di folder webapp:

```
php artisan serve  
// Laravel development server started on http://localhost:8000/
```

Untuk mengaksesnya, kita dapat mengunjungi `http://localhost:8000`.

Akan muncul tampilan seperti berikut:



Berhasil menjalankan simple web server

Untuk mematikan server, dapat menggunakan `ctrl+c` di terminal.

Salah satu kekurangannya adalah kita harus me-restart server ketika merubah file konfigurasi (`.env`).

1.7 Konfigurasi Database

Pada buku ini, kita akan menggunakan database MySQL. Cara yang paling mudah untuk pengguna OSX adalah dengan menggunakan [Mamp](https://www.mamp.info)¹. Untuk pengguna Linux atau Windows bisa menggunakan [Xampp](https://www.apachefriends.org)².

Setelah server MySQL berjalan, buatlah database baru (bisa menggunakan [phpMyAdmin](#)²⁴ atau [Sequel Pro](#)²⁵). Catat nama database, username dan password yang akan digunakan. Kemudian kita isi field berikut pada file `.env` (sesuaikan dengan data Anda):

`.env`

```
DB_DATABASE=Larapus DB_USERNAME=rahmat
DB_PASSWORD=r4ha51a
```

Setelah melakukan perubahan ini, pastikan untuk me-restart server.

¹ <https://www.mamp.info>

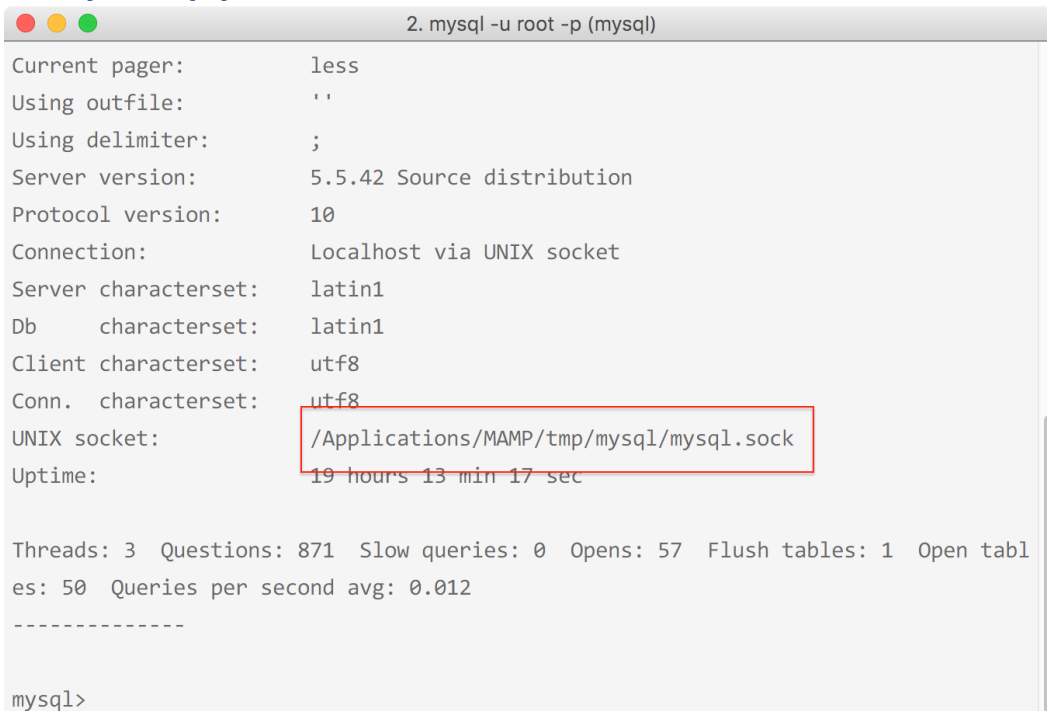
² <https://www.apachefriends.org>

1.7.1 Error koneksi database?

Menggunakan teknik ini, terkadang menyebabkan koneksi database error dengan pesan “Can’t connect to local MySQL server through socket ...”. Tidak selalu terjadi sih. Penyebabnya karena Laravel tidak dapat menentukan letak file socket mysql. Solusinya, kita harus menemukan letak file socket dengan login ke mysql dan jalankan `status` sehingga muncul output seperti berikut:

²⁴ <https://www.phpmyadmin.net>

²⁵ <http://www.sequelpro.com>



```
2. mysql -u root -p (mysql)
Current pager:      less
Using outfile:      ''
Using delimiter:    ;
Server version:    5.5.42 Source distribution
Protocol version:  10
Connection:        Localhost via UNIX socket
Server characterset: latin1
Db characterset:   latin1
Client characterset: utf8
Conn. characterset: utf8
UNIX socket:        /Applications/MAMP/tmp/mysql/mysql.sock
Uptime:             19 hours 13 min 17 sec

Threads: 3  Questions: 871  Slow queries: 0  Opens: 57  Flush tables: 1  Open tabl
es: 50  Queries per second avg: 0.012
-----
mysql> _
```

Lokasi file socket

Bila masih tidak ditemukan, carilah file konfigurasi mysql pada sistem dan temukan konfigurasi untuk lokasi file socket.

Selanjutnya, kita tambahkan isian `unix_socket` pada konfigurasi koneksi database dengan alamat yang kita temukan. Misalnya, hasil akhirnya akan seperti berikut:

```
....
'mysql' => [
  'driver' => 'mysql',
  'host' => env('DB_HOST', 'localhost'), 'database' =>
  env('DB_DATABASE', 'forge'),
  'username' => env('DB_USERNAME', 'forge'),
  'password' => env('DB_PASSWORD', ''),
  'charset' => 'utf8',
  'collation' => 'utf8_unicode_ci',
  'prefix' => '',
  'strict' => false,
```

```
'unix_socket' => '/Applications/MAMP/tmp/mysql/mysql.sock',  
], ....
```



Sudah cukup.

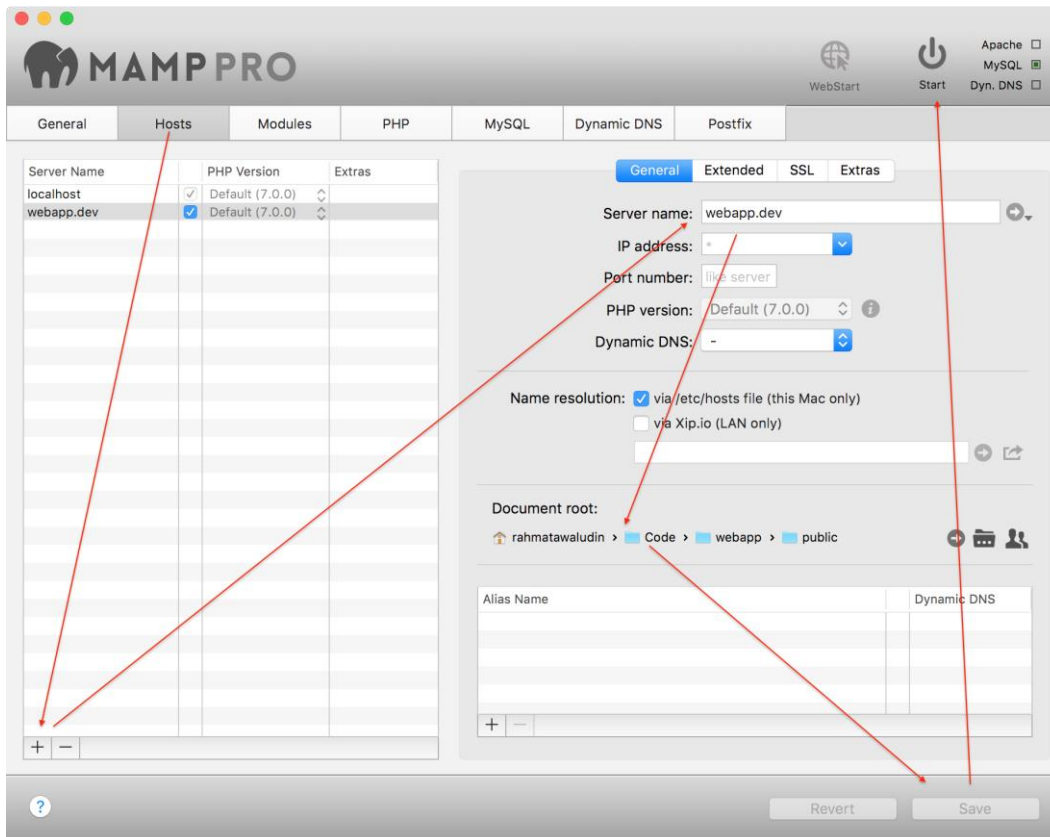
Menggunakan teknik builtin web server, sudah cukup untuk mempelajari isi buku ini. Topik selanjutnya merupakan tambahan bagi yang ingin mendalami bagaimana menjalankan virtual host. Saran saya, sekarang langsung lanjut ke bab selanjutnya dan kunjungi lagi dua topik dibawah setelah buku selesai.

1.8 Virtual Host

Menggunakan virtual host, aplikasi dapat diakses dengan url seperti <http://webapp.dev>, <http://www.webapp.com>, dan sebagainya walupun masih berada di lokal. Berikut cara membuat virtual host:

1.8.1 MAMP

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



Konfigurasi virtual host Mamp

1.8.2 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.dev`

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

```

1  ....
2  127.0.0.1    webapp.dev
3  ....

```



Catatan untuk Windows Terkadang file `hosts` ini tidak bisa langsung disimpan di tempatnya. Solusinya, simpan terlebih dahulu file ini di desktop, lantas replace file aslinya.

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>
```

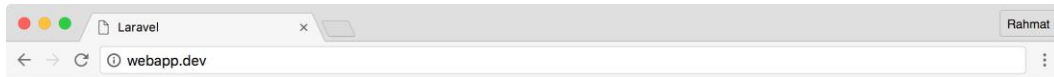
5. Buka file `httpd-vhosts.conf` yang ada di alamat `C:\xampp\apache\conf\extra`
6. Tambahkan setingan di bawah ini untuk membedakan website yang dipanggil dengan `localhost` dan website yang dipanggil dengan `virtual host`

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

```
1 NameVirtualHost *:80
2
3 #VirtualHost untuk webapp.dev
4
5 <VirtualHost *:80>
6 DocumentRoot C:/xampp/htdocs/webapp/public
7 ServerName webapp.dev
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.dev`.



Laravel

DOCUMENTATION LARACASTS NEWS FORGE GITHUB

Berhasil setup virtualhost



Homestead?

Ada satu teknik lagi untuk menjalankan Laravel yaitu menggunakan homestead. Teknik ini sangat bermanfaat ketika pengerjaan Laravel dalam sebuah tim. Tujuannya agar konfigurasi tiap anggota tim selalu sama. Pembahasan homestead tidak dibahas dibuku ini, untuk memahaminya dapat mengunjungi [dokumentasi](#)²⁶ atau baca di buku [Menyelami Framework Laravel](#)²⁷ .

²⁶ <https://laravel.com/docs/5.3/homestead> ²⁷ <https://leanpub.com/bukularavel>

1.9 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 di Laravel. Semangat!

:)