

## Chapter 3

# Instalasi dan Konfigurasi CodeIgniter

Agar dapat menggunakan CodeIgniter, Anda harus menginstall dan melakukan konfigurasi terhadap CodeIgniter terlebih dahulu. Instalasi CodeIgniter sangatlah mudah. Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Hal-hal yang harus dipersiapkan dalam menginstall CodeIgniter adalah

## Mempersiapkan Web Server

Seperti yang kita tahu bahwa PHP adalah bahasa pemrograman website yang berjalan disisi server oleh karena itu untuk dapat menjalankan website yang dibuat dengan menggunakan PHP, di komputer harus terinstall aplikasi web server yang mendukung PHP. Banyak sekali aplikasi web server yang beredar, salah satu web server yang sangat terkenal dan juga bersifat bebas adalah web server Apache, sebuah web server yang digunakan pada sebagian server yang ada di internet.

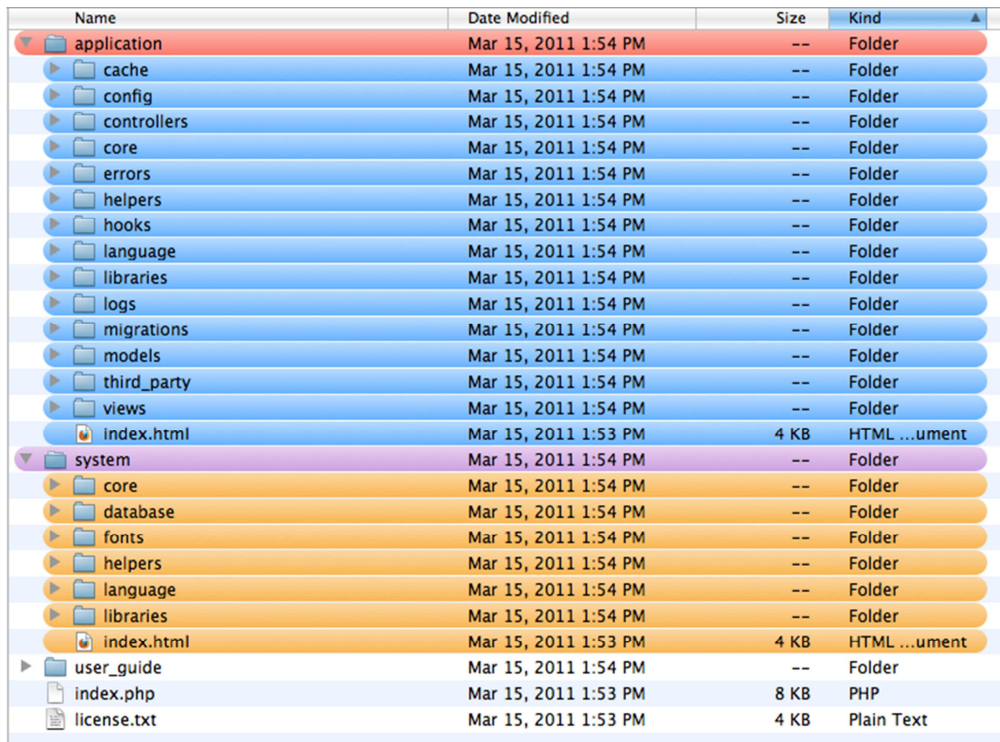
Untuk website yang melibatkan basis data sebagai tempat penyimpanan datanya maka tidak perlu bingung karena CodeIgniter juga mendukung berbagai jenis server basis data, yaitu MySQL, MySQLi, MS SQL, PostgreSQL, Oracle, dan SQLite. CodeIgniter juga bisa dijalankan di semua sistem operasi yang bisa menjalankan aplikasi-aplikasi di atas, baik Windows, Linux, BSD dan yang lainnya.

## Instalasi CodeIgniter

Instalasi CodeIgniter sangat mudah. Meskipun namanya instalasi tetapi karena CodeIgniter adalah aplikasi berbasis website maka sebenarnya yang perlu dilakukan adalah meng-copy folder aplikasi CodeIgniter ke dalam folder **htdocs** atau **DocumentRoot** dari web server yang telah diinstall sebelumnya. Berbeda dengan instalasi software pada umumnya.

Sebelum melakukan instalasi yang perlu dilakukan pertama kali adalah mendapatkan kode sumber dari CodeIgniter itu sendiri yang dapat didownload di <http://www.CodeIgniter.com/> (disediakan pula di dalam CD penyerta buku). Selanjutnya letakkan folder hasil ekstrak tadi di DocumentRoot web server, yaitu folder htdocs didalam direktori **C:\xampp\apache\friends\** bagi yang menggunakan XAMPP di Windows atau **/var/www/html** bagi yang menggunakan linux (semua tergantung dimana anda menginstall dan mengkonfigurasi webserver).

Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Adapun struktur utama dari CodeIgniter terbagi menjadi dua bagian, yaitu **application** dan **sistem/core CodeIgniter**. Application adalah tempat kita meletakkan code yang akan dibuat (bewarna merah dan hijau sedangkan sistem/core CodeIgniter yang bewarna ungu) . Folder sistem berisi library-library dan helper bawaan CodeIgniter.



Name	Date Modified	Size	Kind
application	Mar 15, 2011 1:54 PM	--	Folder
cache	Mar 15, 2011 1:54 PM	--	Folder
config	Mar 15, 2011 1:54 PM	--	Folder
controllers	Mar 15, 2011 1:54 PM	--	Folder
core	Mar 15, 2011 1:54 PM	--	Folder
errors	Mar 15, 2011 1:54 PM	--	Folder
helpers	Mar 15, 2011 1:54 PM	--	Folder
hooks	Mar 15, 2011 1:54 PM	--	Folder
language	Mar 15, 2011 1:54 PM	--	Folder
libraries	Mar 15, 2011 1:54 PM	--	Folder
logs	Mar 15, 2011 1:54 PM	--	Folder
migrations	Mar 15, 2011 1:54 PM	--	Folder
models	Mar 15, 2011 1:54 PM	--	Folder
third_party	Mar 15, 2011 1:54 PM	--	Folder
views	Mar 15, 2011 1:54 PM	--	Folder
index.html	Mar 15, 2011 1:53 PM	4 KB	HTML ...ument
system	Mar 15, 2011 1:54 PM	--	Folder
core	Mar 15, 2011 1:54 PM	--	Folder
database	Mar 15, 2011 1:54 PM	--	Folder
fonts	Mar 15, 2011 1:54 PM	--	Folder
helpers	Mar 15, 2011 1:54 PM	--	Folder
language	Mar 15, 2011 1:54 PM	--	Folder
libraries	Mar 15, 2011 1:54 PM	--	Folder
index.html	Mar 15, 2011 1:53 PM	4 KB	HTML ...ument
user_guide	Mar 15, 2011 1:54 PM	--	Folder
index.php	Mar 15, 2011 1:53 PM	8 KB	PHP
license.txt	Mar 15, 2011 1:53 PM	4 KB	Plain Text

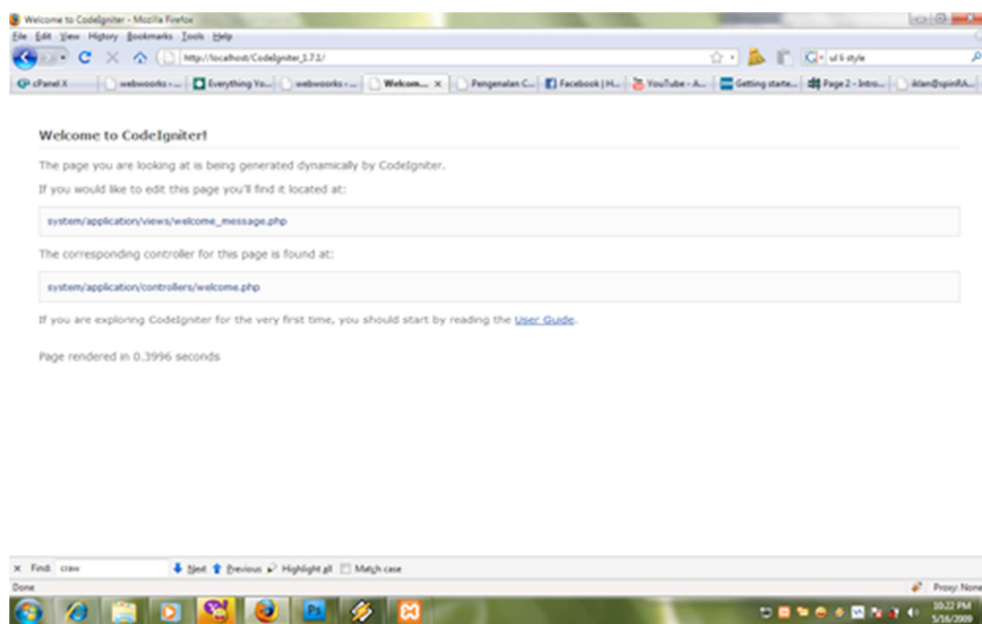
Gambar 4. Struktur Direktori CodeIgniter

Adapun susunan folder CodeIgniter secara default adalah:

- Folder **application**: disinilah aplikasi yang akan kita bangun diletakkan.
  - ✓ Folder **config** - tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari database, router dan autoload aplikasi.
  - ✓ Folder **controllers** - tempat menyimpan semua file controller.
  - ✓ Folder **errors** - tempat menyimpan semua template error aplikasi.
  - ✓ Folder **helpers** - tempat menyimpan helper-helper yang bukan berasal dari CI.
  - ✓ Folder **hooks** - tempat menyimpan hook yang digunakan untuk mengubah alur fungsi dari core CI.
  - ✓ Folder **language** - tempat menyimpan bahasa-bahasa yang akan di gunakan.
  - ✓ Folder **libraries** - tempat menyimpan semua library buatan kita sendiri.
  - ✓ Folder **models** - tempat menyimpan semua model.
  - ✓ Folder **views** - tempat menyimpan semua file view aplikasi.

- Folder **system** menyimpan semua file baik itu file aplikasi yang dibuat maupun core framework-nya.
  - ✓ Folder **cache** - tempat menyimpan semua cache yang dibuat caching library.
  - ✓ Folder **codeigniter** - tempat menyimpan semua file internals CI.
  - ✓ Folder **database** - tempat menyimpan semua driver database drivers dan class yang akan digunakan.
  - ✓ Folder **fonts** - tempat menyimpan semua font yang digunakan image manipulation library.
  - ✓ Folder **helpers** - tempat menyimpan semua helper core CI.
  - ✓ Folder **language** - tempat menyimpan semua language core CI.
  - ✓ Folder **libraries** - tempat menyimpan semua library core CI
  - ✓ Folder **logs** - tempat menyimpan semua logs generated oleh CI.
  - ✓ Folder **plugin** - tempat menyimpan semua plugin core CI.
  - ✓ Folder **scaffolding** - tempat menyimpan semua file yang berfungsi sebagai scaffolding .
- Folder **user\_guide** berisi userguide/manual penggunaan CI.
- File **index.php** file yang akan handle semua request yang dilakukan oleh client.

Setelah meletakkan CodeIgniter ke dalam folder htdocs maka akan didapatkan tampilan seperti di bawah ini, artinya CodeIgniter telah sukses berjalan di aplikasi Anda.



Gambar 5. Tampilan browser ketika sukses menginstall Codeigniter

Adapun checklist yang harus dilakukan untuk menjalankan CodeIgniter secara default adalah

- Pastikan Apache dan PHP telah terinstall dan berjalan di komputer.
- Pastikan peletakkan source code CodeIgniter di folder/direktori web apache (biasanya htdocs) dan memiliki permission setidaknya-tidaknya read only atau kode 644.

## Konfigurasi CodeIgniter

Walaupun CodeIgniter dapat berjalan dengan konfigurasi default, tetapi untuk sebuah aplikasi yang nyata kita harus tetap melakukan konfigurasi, setidaknya pada bagian **base\_url** dan **router**. Pengaturan base\_url dan router sangat berguna ketika proses pengembangan aplikasi yang banyak menggunakan helper dan library.

File konfigurasi terletak dalam folder **application/config**. Adapun file-file yang terdapat dalam direktori tersebut dan sering digunakan antara lain:

- **Config.php.** Pada file konfigurasi config.php berisi konfigurasi secara umum mengenai CodeIgniter, seperti peletakkan baseurl, suffix, frontcontroller, serta metode yang digunakan URI dan lain-lain. Adapun konfigurasi-konfigurasi yang perlu diperhatikan adalah :

- **\$config['base\_url']** - Konfigurasi ini berisi alamat url sebuah aplikasi. Jika menggunakan helper url maka konfigurasi ini harus di-set dengan benar. Contoh: aplikasi Anda akan diakses dengan menggunakan domain www.contoh.com/app\_ci maka pada konfigurasi ini harus diisi:

```
$config['base_url'] = "http://www.contoh.com/app_ci/";
```

Tetapi jika ingin menggunakan base url yang lebih fleksibel maka dapat menggantinya dengan variabel server. Contoh:

```
$config['base_url'] = "http://".$_SERVER['HTTP_HOST'].  
str_replace(basename($_SERVER['SCRIPT_NAME']), "",  
$_SERVER['SCRIPT_NAME']);
```

- **\$config['index\_php']** - Konfigurasi ini berisi file yang menjadi frontcontroller. Konfigurasi ini berhubungan dengan base\_url. Jika menggunakan **.htaccess** untuk mempercantik url maka isi variabel ini harus dikosongkan.
- **\$config['uri\_protocol']** - Konfigurasi ini bertujuan untuk menentukan bagaimana library URI bekerja. CodeIgniter dapat menangkap URI yang di berikan melalui 4 cara yaitu **PATH\_INFO**, **QUERY\_STRING**, **REQUEST\_URI** dan **ORIG\_PATH\_INFO**. Masing-masing cara mempunyai kelebihan dan kekurangan masing-masing, bahkan tidak semua web server mendukung semua cara tersebut, oleh karena itu secara default digunakan pilihan auto. Tetapi jika aplikasi membutuhkan sesuatu yang lebih *custom* maka pemilihan metode URI secara langsung akan lebih baik. Lebih lanjut akan dibahas dibagian library.
- **\$config['url\_suffix']** - Konfigurasi ini bertujuan untuk menambahkan akhiran pada url. Contoh Anda mempunyai sebuah controller page, maka controller tersebut akan diakses melalui http://localhost/index.php/page, dengan menambahkan url\_suffix berisi "html" maka url tadi dapat juga diakses melalui http://localhost/index.php/page.html.

- **\$config['language']** - Secara default CodeIgniter sudah mendukung banyak bahasa/multy language. Kita dapat mengubah pesan-pesan yang ada di dalam CodeIgniter dengan bahasa yang kita kehendaki. Untuk mengubah bahasa tersebut cukup dengan mendownload paket bahasa yang diinginkan lalu uraikan di dalam direktori **system/languages/[nama\_lang]** lalu Anda tinggal mengubah nama\_lang di config.
- **\$config['enable\_hooks']** - Konfigurasi ini bertujuan mengaktifkan/menonaktifkan hook pada CodeIgniter. Hook dapat dikatakan event-event yang terjadi pada CodeIgniter, dimana kita bisa meletakkan fungsi di dalamnya. Hook akan bermanfaat sekali ketika Anda ingin mengubah perilaku CodeIgniter maupun untuk *logging event*. Contoh: Anda ingin mengubah urutan *loading library* dimana Anda sudah meng-extend library router untuk menggunakan database untuk aturan routing-nya. Artinya Anda harus meload library database sebelum library loader. Hal tersebut bisa dilakukan melalui hook.
- **\$config['subclass\_prefix'] = 'MY\_'**. Jika kita ingin mengubah/mengextend library CodeIgniter maka library tersebut harus memiliki prefik yang sama dengan konfigurasi ini.
- **\$config['permitted\_uri\_chars']**. Konfigurasi ini bertujuan untuk keamanan CodeIgniter. Konfigurasi ini menentukan karakter apa saja yang boleh digunakan di dalam uri.
- **\$config['log\_threshold']**. Konfigurasi ini menentukan bagaimana sistem logging CodeIgniter bekerja. Sistem logging ini sangat membantu dalam proses pengembangan terutama ketika *debugging*. Jika di set 0 maka tidak ada proses logging error di CodeIgniter. Jika di set 1 maka yang dicatat hanyalah pesan-pesan kesalahan yang termasuk kesalahan PHP. Jika di set 2 maka akan menampilkan semua pesan debug dan pesan kesalahan CodeIgniter dan PHP. Jika di set 3 maka logging-logging yang berisi informasi seperti sebuah library telah di load juga akan ditampilkan. Jika di set 4 maka semuanya akan dilog mulai dari error, pesan debug sampai yang bersifat informasi.
- **\$config['log\_path']**. Dikonfigurasi ini kita dapat menentukan dimana log akan diletakkan. Jika diisi kosong maka akan diletakkan di system/logs. Harus dingat Anda harus mengubah tingkat hak akses dari direktori tersebut menjadi dapat ditulisi / *writable* jika kita menggunakan fasilitas logging.
- Konfigurasi session. Session di CodeIgniter menggunakan cookies jadi kita dapat mengeset waktu hidup cookies/expire, nama cookies dan lain-lain melalui konfigurasi ini.

```
$config['sess_cookie_name']='Nama cookie';
$config['sess_expiration']=7200;
$config['sess_encrypt_cookie']=FALSE;
$config['sess_use_database']=FALSE;
$config['sess_table_name']='session_table';
$config['sess_match_ip']=FALSE;
$config['sess_match_useragent']=TRUE;
$config['sess_time_to_update']=300;
```

- **Autoload.php.** Konfigurasi ini bertujuan untuk menentukan sumber daya apa yang akan di-load secara otomatis. Cara penggunaannya sederhana, misalnya kita ingin me-load library database, pagination dan lain-lain secara otomatis maka kita tinggal mengubah `$autoload['libraries']` menjadi :

```
$autoload['libraries']=array('database','session','pagination')
```

- **Routes.php.** Konfigurasi di file ini bertujuan untuk menentukan kemana routing oleh library route akan dilakukan. Hal paling sederhana yang harus dilakukan adalah mengubah default controller (controller yang akan dibuka ketika tidak ada uri yang diberikan oleh browser). Misalnya website kita beralamat [www.koder.web.id](http://www.koder.web.id). Maka ketika membuka website tersebut maka secara otomatis CodeIgniter akan mengalihkan ke controller default, karena tidak disertakan di dalam url kita. Adapun yang perlu diubah adalah

```
$route['default_controller']="welcome";
```

## Kesepakatan Coding (*Coding Standart*) CodeIgniter

Sebelum melakukan coding menggunakan codeigniter maka ada baiknya kita mengetahui apasaja kesepakatan-kesepakatan yang ada di codeigniter. Kesepakatan-kesepakatan tersebut akan membuat kode kita lebih mudah dipahami oleh developer lainnya . Adapun kesepakatan tersebut diantaranya :

- **PHP Closing Tag**

Ketika kita menulis library, helper, controller ataupun model maka sebaiknya tidak menggunakan tanda penutup pada dokumen php `?>`. Hal tersebut dilakukan untuk mencegah adanya spasi atau karakter yang tidak diinginkan pada code kita sehingga membuat aplikasi error. Kita juga disarankan untuk memberikan informasi tentang akhir dokumen dan berisi path dokumen tersebut.

**Contoh salah:**

```
<?php echo "Here's my code!"; ?>
```

**Contoh Benar:**

```
<?php echo "Here's my code!";
```

```
/* End of file myfile.php */
```

```
/* Location: ./system/modules/mymodule/myfile.php */
```

- **Penamaan Class dan Method**

Penamaan Class harus dimulai dengan huruf besar. Jika class menggunakan beberapa kata maka kata-kata tersebut dipisahkan menggunakan underscore dan bukan camelcase.

**Contoh salah:**

```
class superclass  
class SuperClass
```

**Contoh Benar:**

```
class Super_class
```

Aturan diatas juga berlaku untuk method contohnya

**Contoh kurang tepat:**

```
function fileproperties() //Tidak deskriptif dan memiliki underscore  
function fileProperties() // Tidak deskriptif dan underscore CamelCase  
function getfileproperties() // Kurang underscore  
function getFileProperties() // menggunakan CamelCase  
get_the_file_properties_from_the_file() // terlalu panjang
```

**Contoh Tepat:**

```
function get_file_properties() //deskriptif, pakai underscore, dan huruf  
kecil
```