

# **CERDAS MENGUASAI CODEIGNITER**



---

# CERDAS MENGUASAI CODEIGNIT

## Dalam 24 Jam

---

**Rolly M. Awangga**  
Informatics Research Center



**Kreatif Industri Nusantara**

***Penulis:***

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

***Editor:***

M. Yusril Helmi Setyawan

***Penyunting:***

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

***Desain sampul dan Tata letak:***

Deza Martha Akbar

***Penerbit:***

Kreatif Industri Nusantara

***Redaksi:***

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

***Distributor:***

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara  
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat  
menahan lelahnya  
belajar, Maka kamu harus  
sanggup menahan  
perihnya Kebodohan.’  
Imam Syafi’i*

# CONTRIBUTORS

---

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia



# CONTENTS IN BRIEF

---

<b>1</b>	<b>Mengenal CodeIgniter</b>	<b>1</b>
<b>2</b>	<b>CodeIgniter 4</b>	<b>5</b>





# DAFTAR ISI

---

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
<b>1 Menenal CodeIgniter</b>	<b>1</b>
1.1 CodeIgniter	1
1.2 Fungsi CodeIgniter	2
1.3 Kelebihan CodeIgniter	2
1.4 Kelebihan CodeIgniter	3
	<b>ix</b>

<b>2</b>	<b>CodeIgniter 4</b>	<b>5</b>
2.1	CodeIgniter 4 pre alpha bukan CodeIgniter 3	5
2.2	apa yang masih ada? dan apa yang telah berubah?	6
2.2.1	Tidak ada perbedaan dalam memuat objek	6
2.2.2	Memuat Model dan Perpustakaan dalam pengontrol	6
2.2.3	Basis data dan input	7
2.2.4	Perpustakaan sebagai antarmuka untuk kelas CodeIgniter tidak ada	8
Daftar Pustaka		11
Index		13

# DAFTAR GAMBAR

---



# DAFTAR TABEL

---



# Listings

---





# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa



# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat  
Februari, 2019*



# ACKNOWLEDGMENTS

---

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.



# ACRONYMS

---

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association





# GLOSSARY

---

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald



# SYMBOLS

---

- $A$  Amplitude
- $\&$  Propositional logic symbol
- $a$  Filter Coefficient
  
- $\mathcal{B}$  Number of Beats



# INTRODUCTION

---

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$



# BAB 1

---

## MENGENAL CODEIGNITER

---

### 1.1 CodeIgniter

Codeigniter merupakan suatu Web Application Framework (WAF) yang di bentuk khusus untuk mempermudah para developer web baik ahli maupun bagi kaum awam dalam mengembangkan dan membuat apilkasi berbasis web.

Codeigniter itu sendiri berisi beberapa kumpulan kode berupa pustaka (library) dan alat (tools) yang dipadukan/digabungkan sedemikian rupa menjadi suatu kerangka kerja (framework). Dalam Codeigniter yang merupakan framework web untuk Bahasa pemrograman PHP terkhususnya merupakan buatan yang dibuat oleh Rick Ellis pada tahun 2006. Yang juga merupakan penemu dan pendiri dari EllisLab.

EllisLab itu sendiri merupakan suatu tim kerja yang berdiri pada tahun 2002 dan bergerak di bidang pembuatan software dan tool untuk para pengembang web. Dan sejak tahun 2014 sampai sekarang, ELisLab telah menyerahkan hak kepemilikan Codeigniter terhadap British Columbia Institute of Technology (BCIT) untuk proses pengembangan lebih lanjut kedepannya.

Pada saat ini situs web resmi dari Codeigniter telah berubah dari [www.ellislab.com](http://www.ellislab.com) menjadi [www.codeigniter.com](http://www.codeigniter.com). Sehingga dapat mempermudah dalam pencarian men-



genai fitur-fiturnya. Codeigniter memiliki banyak fitur yang membantu dalam pengembangan PHP untuk dapat membuat aplikasi web secara mudah, cepat dan efisien.

Dibandingkan dengan framework web PHP lainnya, Codeigniter memiliki desain yang lebih sederhana dan bersifat fleksibel (tidak kaku). Dan Codeigniter mengizinkan para pengembang untuk menggunakan framework secara parsial atau secara keseluruhan. Sehingga codeigniter masih memberi kebebasan kepada pengembang untuk menulis bagian-bagian kode tertentu di dalam aplikasi menggunakan cara konvensional (tanpa framework).

Codeigniter menganut arsitektur Model-View-Controller (MVC), yang memisahkan antar bagian kode untuk penanganan proses bisnis dengan bagian kode untuk keperluan presentasi (tampilan). Dengan menggunakan pola desain ini, memungkinkan para pengembang untuk mengerjakan aplikasi berbasis web secara bersama (teamwork). Para pengembang web lebih bisa berfokus pada bagian masing-masing tanpa mengganggu bagian yang lain. Sehingga aplikasi yang dibangun akan selesai lebih cepat dan menghemat waktu yang digunakan.

## 1.2 Fungsi CodeIgniter

CodeIgniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai framework pilihannya. Fungsi dari CodeIgniter di antaranya adalah sebagai berikut :

1. Mempercepat dan mempermudah kita dalam pembuatan website.
2. Menghasilkan struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file phpnya.
3. Memberikan standar coding sehingga memudahkan kita atau orang lain untuk mempelajari kembali system aplikasi yang dibangun.

## 1.3 Kelebihan CodeIgniter

CodeIgniter memiliki beberapa kelebihan dibandingkan dengan framework lainnya diantaranya adalah sebagai berikut :

1. Berukuran sangat kecil. File download nya hanya sekitar 2MB, itupun sudah includedokumentasinya yang sangat lengkap.
2. Dokumentasi yang bagus. Saat anda mendownloadnya, telah disertakan dengan dokumentasi yang berisi pengantar, tutorial, bagaimana panduan penggunaan, serta referensi dokumentasi untuk komponen-komponennya.
3. Kompatibilitas dengan Hosting. CodeIgniter mampu berjalan dengan baik pada hampir semua platform hosting. CodeIgniter juga mendukung database-database paling umum, termasuk MySQL.

4. Tidak ada aturan coding yang ketat. Terserah anda jika anda hanya ingin menggunakan Controller, tanpa View, atau tidak menggunakan Model, atau tidak salah satu keduanya. Namun dengan menggunakan ketiga komponennya adalah pilihan lebih bijak.
5. Kinerja yang baik. Codeigniter sangat cepat bahkan mungkin bisa dibilang merupakan framework yang paling cepat yang ada saat ini.
6. Sangat mudah diintegrasikan. CodeIgniter sangat mengerti tentang pengembangan berbagai library saat ini. Karenanya CodeIgniter memberikan kemudahan untuk diintegrasikan dengan library-library yang tersedia saat ini.
7. Sedikit Konfigurasi. Konfigurasi CodeIgniter terletak di folder application/config. CodeIgniter tidak membutuhkan konfigurasi yang rumit, bahkan untuk mencoba menjalankannya, tanpa melakukan konfigurasi sedikitpun ia sudah bisa berjalan.
8. Mudah dipelajari. Disamping dokumentasi yang lengkap, ia juga memiliki berbagai forum diskusi.

## 1.4 Kelebihan CodeIgniter

Ada kelebihan dan juga pasti ada kekurangan pada CodeIgniter kekurangan yang dimiliki merupakan beberapa pendapat orang-orang yang pernah menggunakannya dan kekurangan ini yang menunjang untuk adanya perbaikan kedepannya, dan kekurangan dari CodeIgniter itu sendiri adalah sebagai berikut :

1. CodeIgniter tidak ditujukan untuk pembuatan web dengan skala besar.
2. Library yang sangat terbatas. Hal ini dikarenakan sangat sulit mencari plugin tambahan yang terverifikasi secara resmi, karena pada situsnya CodeIgniter tidak menyediakan plugin-plugin tambahan untuk mendukung pengembangan aplikasi dengan CI.
3. Belum adanya editor khusus CodeIgniter, sehingga dalam melakukan create project dan modul-modulnya harus berpindah-pindah folder.



## BAB 2

---

# CODEIGNITER 4

---

### 2.1 CodeIgniter 4 pre alpha bukan CodeIgniter 3

Semenjak BCIT mengambil CodeIgniter di bawah perlindungannya, mereka mencoba bergerak maju untuk memperbaiki beberapa permasalahan yang ada pada sebelumnya. CodeIgniter versi 3 dapat dikatakan sangat kompatibel dengan versi 2. Bahkan dengan mengubah modal di sana-sini Kita dapat memigrasi aplikasi dalam "jiffy". Pada saat ini CodeIgniter 4 telah dibuka untuk umum dalam versi pra-alfa akan tetapi mengenai *Source Code* tidak begitu lengkap dan bahkan perkembangannya cukup lama.

Hal pertama yang saya perhatikan adalah ada beberapa perubahan pada CodeIgniter sebelumnya. ini tidak terlihat seperti CodeIgniter seperti yang kita gunakan juga. BCIT memutuskan untuk membersihkan beberapa fungsi yang tidak begitu penting untuk membangun kembali sebuah framework yang lebih simple dan cepat di mengerti pada kalangan programmer. untuk pendokumentasinya dan juga perawatannya dilakukan perkembangan itu sendiri, butuh beberapa saat untuk mencari tahu bagaimana semuanya cocok pada Framework yang disebut CodeIgniter 4, yang hanya akan berjalan mulai dengan PHP versi 7. Jadi pertama-tama cari tahu apakah penye-

dia host Kita mendukung atau jika sudah diinstal dan berjalan di server lokal Kita sendiri. CodeIgniter 4 TIDAK kompatibel dengan CodeIgniter 3.

Semuanya sekarang berhubungan dengan ruang nama. Tidak ada lagi *this*— *load*— *model(sesuatu)* tetapi Kita harus terbiasa dengan penggunaan dan kembali memangkas referensi ruang nama.

## 2.2 apa yang masih ada? dan apa yang telah berubah?

Dalam ulasan kecil dan tidak lengkap ini, saya tidak akan berbicara tentang pandangan, karena saya belum sampai sejauh itu dan saya menggunakan CodeIgniter terutama sebagai server REST ke aplikasi berbasis Ext JS. Saya akan berbicara tentang pandangan setelah benar-bener CodeIgniter 4 di rilis dan bias di publikasikan sepenuhnya. CodeIgniter masih merupakan kerangka kerja MVC *model-view-controller* dan tim pengembangan sangat jelas tentang HMVC (*hierarki-model-tampilan-controller*, itu tidak akan diimplementasikan. HMVC itu sendiri merupakan kumpulan triad MVC tradisional yang beroperasi sebagai satu aplikasi. Setiap triad sepenuhnya independen dan dapat dieksekusi tanpa kehadiran yang lain.

Apa yang akan Kita perhatikan adalah struktur jalur yang berbeda. Kita sekarang akan menemukan jalur aplikasi, sistem dan publik. Itu perlu penjelasan. Gagasan di balik semua ini adalah bahwa jalur publik harus menjadi jalur root dari aplikasi web Kita. Ini berarti bahwa sistem dan aplikasi akan disembunyikan dari mata publik. Ini meningkatkan keamanan. Juga di belakang mata publik adalah folder dapat ditulis, yang memiliki akses tulis, tetapi tidak dapat diakses langsung oleh URL.

### 2.2.1 Tidak ada perbedaan dalam memuat objek

CodeIgniter melakukan pemuatan Model, Perpustakaan, dan objek lainnya dengan cara yang sama. Memuat Model atau Pustaka secara otomatis, serta objek lain, dapat diatur di dalam *"Autoload.php"* di folder *"config"*. Di sana Kita memiliki *array psr4* dan *array classmap*, yang dapat digunakan untuk memuat suatu objek secara otomatis. Tidak ada perbedaan antara kelas ketika datang ke pemuatan otomatis. Kita harus memberi tahu apa namespace itu dan di mana ia dapat ditemukan.

```
classmap = [
    'TMDB_API' => APPPATH . 'ThirdParty/tmdb-v3.php'
];
```

Sampel tersebut menunjukkan bagaimana *namespace* dibuat untuk objek non-CodeIgniter di folder *"ThirdParty"* di dalam folder *"application"*.

### 2.2.2 Memuat Model dan Perpustakaan dalam pengontrol

Kutipan di bawah ini menunjukkan bagaimana Kita memuat model dan pustaka di controller. Sampel memiliki kelas model bernama *"TmdbModel"* dan kelas perpustakaan bernama *"Tmdb"*. Perhatikan deklarasi gunakan. Kita harus terbiasa dengan

itu (saya lakukan), tetapi Kita akan terbiasa dengan cepat. Saya harus mendeklarasikan antarmuka CodeIgniter HTTP karena saya mendapatkan kesalahan pada konstruk induk ketika saya tidak menambahkan variabel *"request"* dan *"response"*. Meninggalkan parameter tidak berfungsi.

```
<?php
namespace App\Controllers;

use App\Models\TmdbModel;
use App\Libraries\Tmdb;

use CodeIgniter\HTTP\RequestInterface;
use CodeIgniter\HTTP\ResponseInterface;

class Movies extends \CodeIgniter\Controller
{
    protected $tmdbModel;
    protected $tmdb;

    public function __construct(RequestInterface $request,
        ResponseInterface $response)
    {
        parent::__construct($request, $response);

        $this->tmdbModel = new TmdbModel();
        $this->tmdb = new Tmdb();

        helper('download');
    }
}
```

Seperti yang kita lihat kelas model dan kelas perpustakaan dinyatakan sebagai objek baru. kita tidak lagi menggunakan *this- > load- > model* dan *this- > load- > library*, yang sebenarnya juga menciptakan objek.

Baris kode terakhir dalam contoh di atas adalah *helper* atau unduh. Ini adalah cara memuat pembantu. Di sini saya memiliki pembantu unduh, yang saya salin dari inti CodeIgniter 3 ke dalam folder pembantu aplikasi saya. Dalam versi final, saya berharap banyak pembantu inti dari versi 3 akan menjadi bagian dari inti versi 4.

### 2.2.3 Basis data dan input

Kelas basis data telah dimodifikasi. Beberapa fungsi hilang dan beberapa kini memiliki metode yang berbeda (tapi mungkin belum semuanya diimplementasikan).

```
$query->num_rows() dirubah menjadi:
count($query->getResult())
```

`$query->firstRow()` dirubah menjadi:

```
$row = $query->getRow(0);
```

`$query->result()` dirubah menjadi:

```
$query->getResult()
```

`$this->db->insert('table', $record)` dirubah menjadi:

```
$this->db->table('table')->insert($record);
```

Kelas input digantikan oleh kelas permintaan, ini artinya:

```
$search = $this->input->post('search');
```

dirubah menjadi:

```
$search = $this->request->getPost('search');
```

## 2.2.4 Perpustakaan sebagai antarmuka untuk kelas CodeIgniter tidak ada

Saya sering menggunakan kelas pustaka sebagai antarmuka ke paket atau kelas pihak ketiga seperti TCPDF (pembuatan PDF) atau PHPExcel (pembuatan Excel). Saya meletakkan konten paket pihak ketiga kedalam folder pihak ketiga (seperti sekarang *"ThirdParty"* dan membuat kelas perpustakaan CodeIgniter yang memperluas kelas utama pihak ketiga. Dalam aplikasi TMDB saya, saya juga memiliki sesuatu seperti itu. Saya memiliki perpustakaan bernama Tmdb dan di folder *"ThirdParty"*, saya memiliki skrip PHP *"tmdb-v3.php"* yang tidak ada hubungannya dengan CodeIgniter dan berisi kelas dengan panggilan API TMDB. Sampel di bawah ini menunjukkan bagaimana saya melakukannya di CodeIgniter 4. Dalam folder *Libraries*: Tmdb.php

```
<?php
namespace App\Libraries;
use TMDB_API; // relates to the Autoload.php config

class Tmdb extends TMDB_API {
    private $API_Key = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx';
    private $defaultFormat = self::JSON;
    private $defaultLang = 'en';

    function __construct() {
        parent::__construct(
            $this->API_Key,
            $this->defaultFormat,
            $this->defaultLang
        );
    }
}
```

```
}
```

Referensi TMDB\_API di *Autoload.php* di folder "*config*".

```
public function __construct()
{
    parent::__construct();
    ....
    $classmap = [
        'TMDB_API' => APPPATH . 'ThirdParty/tmdb-v3.php'
    ];
    ....
}
...
}
```

Penggunaan dalam model (TmdbModel.php)

```
<?php
namespace App\Models;
use App\Libraries\Tmdb; // Library Tmdb
class TmdbModel extends \CodeIgniter\Model
{
    protected $tmdb; // Library Tmdb goes here

    function __construct()
    {
        parent::__construct();

        $this->tmdb = new Tmdb(); // declare Tmdb as a new object
    }

    ...

    function getTopRated() {
        return $this->tmdb->TvTopRated(); // usage of library f
    }

    ...
    function AddBoxoffice($info = array())
    {
        $record = array(
            'id'          => $info['id'],
            'name'        => $info['name'],
            'popularity'  => $info['popularity']
        );
    }
}
```



```
        $this->db->table('tmdb_boxoffice')->insert('tmdb_boxoff  
    }  
  
    ...  
}
```

# DAFTAR PUSTAKA

---

1. R. Awangga, "Sampeu: Servicing web map tile service over web map service to increase computation performance," in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.



# Index

---

disruptif, xxix  
modern, xxix