

# **MENGUASAI CODEIGNITER 3**



---

# MENGUASAI CODEIGNITER 3

## 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 Instalasi CodeIgniter</b>	<b>1</b>
<b>2 Menampilkan Teks</b>	<b>7</b>
<b>3 Penggunaan Controller</b>	<b>15</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 Instalasi CodeIgniter</b>	<b>1</b>
1.1 Instalasi CodeIgniter3	1
1.1.1 CodeIgniter	1
1.1.2 Versi CodeIgniter	1
1.1.3 Kebutuhan Sistem	2
	<b>ix</b>

1.1.4	Langkah-Langkah Instalasi CodeIgniter3	2
1.2	Mengenal Pola Desain MVC	2
1.3	Struktur Direktori CodeIgniter	3
1.3.1	Direktori application	4
1.3.2	Direktori System	4
1.3.3	User_guide	5
1.3.4	Alur Aplikasi CodeIgniter	5
<b>2</b>	<b>Menampilkan Teks</b>	<b>7</b>
2.1	Aplikasi "HELLO WORLD"	7
2.1.1	Nama-nama yang sudah didefinisikan oleh CodeIgniter3	7
2.2	Nama-nama Variabel	7
2.3	Controller	8
2.3.1	Nama-nama Controller	8
2.3.2	View	8
2.4	Model	9
2.5	Menggunakan Controller,Model dan View	10
<b>3</b>	<b>Penggunaan Controller</b>	<b>15</b>
3.1	Penggunaan Controller	15

# DAFTAR GAMBAR

---

1.1	Halaman Welcome dari CodeIgniter	3
1.2	Arsitektur MVC	3
1.3	Tahap Alur CodeIgniter	5
2.1	Perintah Hello World Perintah Lengkap	9
2.2	Perintah Hello World Perintah Simple	9
2.3	Hasil	10
2.4	Membuat views	10
2.5	Menyimpan Direktori View	10
2.6	Hasil dari View	11
2.7	Membuat model	11
2.8	Menyimpan model	11
2.9	Mengubah model ke controllerl	12
2.10	Mengubah model ke controllerl	12

2.11	Controller Hello!	13
2.12	Model Hello!	13
2.13	View Hello!	14
2.14	View Hello!	14

# 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[?].

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



# BAB 1

---

## INSTALASI CODEIGNITER

---

### 1.1 Instalasi CodeIgniter3

#### 1.1.1 CodeIgniter

CodeIgniter adalah *framework* web yang untuk bahasa pemrograman PHP yang dibuat oleh Rick Ellis pada 2006 , penemu dan pendiri EllisLab ([www.ellislab.com](http://www.ellislab.com)) EllisLab adalah suatu tim kerja yang berdiri tahun 2002 dan bergerak dibidang pembuatan *software* dan *tool* untuk para pengembang *web*. CodeIgniter memiliki banyak fitur (fasilitas) yang membantu para pengembang (*developer*) PHP untuk dapat membuat aplikasi web secara cepat dan mudah. dibandingkan dengan *framework* web PHP lainnya, diakui bahwa CodeIgniter memiliki desain yang lebih sederhana bersifat fleksibel(tidak kaku).

#### 1.1.2 Versi CodeIgniter

CodeIgniter memiliki tiga versi yang dikembangkan secara berbeda, yaitu versi 2.2 dan 3.0 kedua versi ini dikembangkan secara bersamaan dan merupakan versi yang



ditetapkan versi yang stabil karena digunakan untuk keperluan dizaman sekarang. adapun juga versi 4.0 yaitu versi pengembangan dari versi 3.0 yang sekarang dirilis.

### 1.1.3 Kebutuhan Sistem

Untuk menjalankan CodeIgniter sangat disarankan untuk memakai server dengan PHP tipe 5.4 atau lebih baru. Walaupun keadaan yang sangat memaksa CodeIgniter dapat berjalan pada PHP minimal versi 5.2.4. Menjalankan CodeIgniter pada PHP versi lama sangat tidak dianjurkan, karena dapat memicu gangguan keamanan dan mengurangi fitur yang ada CodeIgniter

### 1.1.4 Langkah-Langkah Instalasi CodeIgniter3

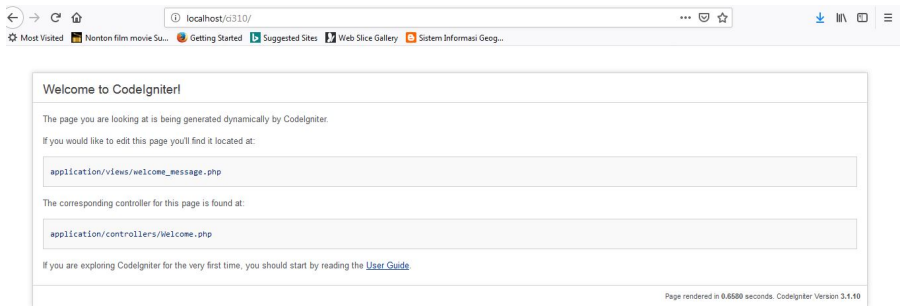
Disini untuk versi codeIgniter yang dipakai adalah CodeIgniter Versi 3.10. untuk mempermudah pembelajaran untuk menginstalasi CodeIgniter, ikuti langkah-langkah berikut ini :

1. Langkah 1 : Download CodeIgniter dari [https ://codeigniter.com](https://codeigniter.com)
2. Langkah 2 : Ekstrak file CodeIgniter-3.1.10.zip ke folder  
C: \XAMPP\htdocs
3. Langkah 3 : Rename folder CodeIgniter-3.1.10.zip sesuai dengan yang anda inginkan. untuk contohnya nama dari folder ini akan diubah menjadi folder ci310
4. Langkah 4 : Didalam folder ci310, terdapat user guide. folder itu berisi tentang dokumentasi atau petunjuk pemakaian CodeIgniter.
5. Langkah 5 : Jangan lupa kondisi server (XAMPP) harus aktif , buka halaman <http://localhost/ci310/> apabila instalasi codeIgniter berhasil, maka Anda akan mendapatkan tampilan seperti gambar dibawah ini :

## 1.2 Mengenal Pola Desain MVC

*Model, View, Controller* atau dengan singkatan *MVC* adalah nama dari suatu metodologi atau pola desain *design pattern* yang digunakan untuk merelasikan data dan *user-interface* aplikasi secara efisien. pola awal MVC digunakan untuk rancang-bangun aplikasi desktop, khususnya untuk aplikasi-aplikasi yang dikembangkan menggunakan C++, Java, dan Smalltalk. Dalam Pola MVC, komponen aplikasi dibagi menjadi tiga bagian yaitu :

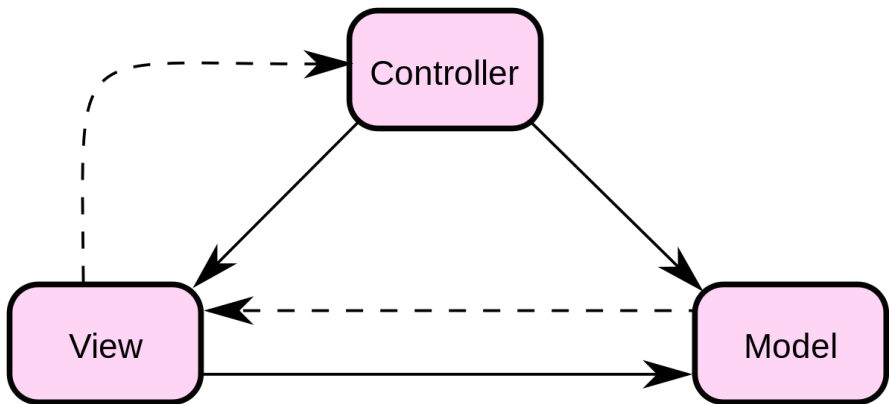
1. *Model* yang mempresentasikan struktur data.
2. *View* yang merupakan representasi keluaran(*output*) dari suatu model.



**Gambar 1.1** Halaman Welcome dari CodeIgniter

### 3. *Controller* Komponen yang bertugas mengambil masukan (*model dan view*)

Secara umum, pola MVC dapat digambarkan sebagai berikut :



**Gambar 1.2** Arsitektur MVC

## 1.3 Struktur Direktori CodeIgniter

Dalam paket distribusinya, *framework* CodeIgniter memiliki tiga direktori, yaitu :

1. application
2. system
3. user\_guide

### 1.3.1 Direktori application

Direktori application adalah direktori yang digunakan untuk menempatkan file dari aplikasi yang akan kita buat. Kita perlu menempatkan daftar *model,controller,view* yang dibuat didalam direktori ini. Penempatan file kedalam direktori application harus diklarifikasikan sesuai ketentuan yang sudah ditetapkan oleh CodeIgniter. Berikut dibawah ini adalah daftar sub-direktori dan penjelasan-penjelasan yang terdapat didalam direktori application

1. cache berguna untuk menyimpan halaman-halaman yang telah dibuka sebelumnya dan disembunyikan *cached*)
2. config, berisi daftar file konfigurasi yang akan digunakan oleh aplikasi yang kita kembangkan
3. controllers, berisi daftar *file controller*
4. core, digunakan untuk menempatkan daftar file kelas induk atau kelas dasar yang nantinya akan digunakan oleh aplikasi.
5. helpers, digunakan untuk menempatkan daftar file helper(pustaka dalam bentuk fungsi) yang didefinisikan sendiri.
6. hooks,digunakan untuk menyimpan file pendukung aplikasi
7. language, dalam direktori ini kita dapat mendefinisikan konstanta tertentu dalam bahasa yang diinginkan.
8. libraries, berisi file library (pustaka dalam bentuk kelas) yang kita definisikan sendiri.
9. logs, digunakan oleh CodeIgniter untuk menyimpan *file log* (catatan) yang secara otomatis akan ditulis ketika terjadi kesalahan
10. models, berisi daftar *model* yang kita perlukan oleh aplikasi.
11. third\_party, digunakan untuk menyimpan *plugin* yang dikembangkan oleh pihak ketiga.
12. views, berisi daftar file *view* yang diperlukan oleh aplikasi.

### 1.3.2 Direktori System

Direktori *system* berisi daftar file kode yang disediakan oleh CodeIgniter, yang diklasifikasikan berdasarkan fungsinya. Sub-direktori yang ada di dalam direktori *system* adalah sebagai berikut :

1. core, berisi daftar file yang merupakan kelas-kelas inti dalam CodeIgniter, seperti CI\_Controller,Ci\_model dan lain-lain.

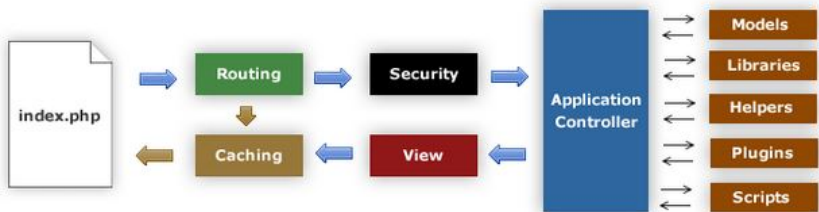
2. database, berisi daftar *file driver* dan *tool* yang digunakan untuk keperluan akses *database*
3. fonts, berisi daftar *file font*
4. helper, berisi daftar *file helper* standar yang disediakan oleh CodeIgniter.
5. language, berisi daftar *file* bahasa (untuk keperluan translasi bahasa).
6. libraries, berisi daftar *file* pustaka kelas standar yang disediakan oleh CodeIgniter.

### 1.3.3 User\_guide

Direktori ini hanya berisi dokumentasi CodeIgniter yang ditulis dalam format HTML. anda dapat memindahkan direktori ini ke direktori lain.

### 1.3.4 Alur Aplikasi CodeIgniter

Berikut ini adalah tahap alur aplikasi yang ditulis menggunakan ini CodeIgniter digambarkan seperti dibawah ini :



**Gambar 1.3** Tahap Alur CodeIgniter

Keterangan :

1. Index.php berfungsi sebagai controller depan, inisialisasi sumber daya dasar yang dibutuhkan untuk menjalankan CodeIgniter.
2. Router memeriksa HTTP request untuk menentukan apa yang harus dilakukan dengan hal itu.
3. Jika file cache ada, dikirim langsung ke browser, melewati pelaksanaan sistem normal.
4. Keamanan Sebelum controller aplikasi dimuat, permintaan HTTP dan setiap pengguna mengirimkan data disaring untuk keamanan.

5. Controller beban model, perpustakaan inti, pembantu, dan sumber daya lain yang diperlukan untuk memproses permintaan khusus.
6. Lihat yang diselesaikan diberikan kemudian dikirim ke browser web untuk dilihat. Jika caching diaktifkan, pandangan-cache pertama sehingga pada permintaan selanjutnya dapat disajikan.

## BAB 2

---

# MENAMPILKAN TEKS

---

### 2.1 Aplikasi "HELLO WORLD"

Hello World adalah syntax atau script untuk mengetahui aturan sebuah tools tersebut, dimana perintah ini adalah teks yang selalu digunakan dalam pengujian awal suatu program untuk menampilkan output.

#### 2.1.1 Nama-nama yang sudah didefinisikan oleh CodeIgniter3

Hal yang perlu diperhatikan dalam menentukan nama (baik nama controller, fungsi, variabel, maupun konstanta) kita tidak boleh menggunakan nama-nama yang sudah didefinisikan oleh CodeIgniter.

### 2.2 Nama-nama Variabel

Untuk mendefinisikan variabel, Anda seharusnya tidak menggunakan nama berikut :

1. \$config

2. \$db
3. \$lang

## 2.3 Controller

controller adalah sebuah file class yang berujuan dapat berhubungan dengan sebuah URI.

### 2.3.1 Nama-nama Controller

Berikut ini daftar nama yang tidak dapat Anda gunakan untuk mendefinisikan nama *Controller*

1. ci\_controller
2. Default
3. Index

mari coba membuat controller sederhana untuk dapat melihat bagaimana controller bekerja. buat file dengan nama hello.php, dan isikan code seperti dibawah ini. Dalam tulisan saya sebelumnya adalah tutorial instalasi codeigniter, sekarang kita lanjut ke cara menampilkan Hello World pada CodeIgniter, kenapa hello world?

Disini kita akan menggunakan *Controller* sebagai komponen utama yang memproses permintaan user. untuk memulainya, kita dapat menjalankan aplikasi Text Editor yang dimiliki. lalu kita bisa menuliskan kode, disini ada 2 perintah yang akan saya masukan yaitu untuk perintah lengkap dan perintah *simplenya* tersebut seperti ini:

Kita hanya tinggal memilih yang mana yang akan kita masukan dari kedua perintah diatas pada gambar tersebut kemudian simpan file dengan nama hello.php dan ditempatkan di dalam direktori ??

C:\XAMPP\htdocs\ci310\application\controllers\\setelah disimpan, kita dapat menjalankan controller diatas dengan menuliskan url berikut : <http://localhost/ci310/index.php/hello> permintaan diatas akan mengeksekusi metode index() yang terdapat didalam kelas Hello. hasil yang diperoleh akan seperti ini :

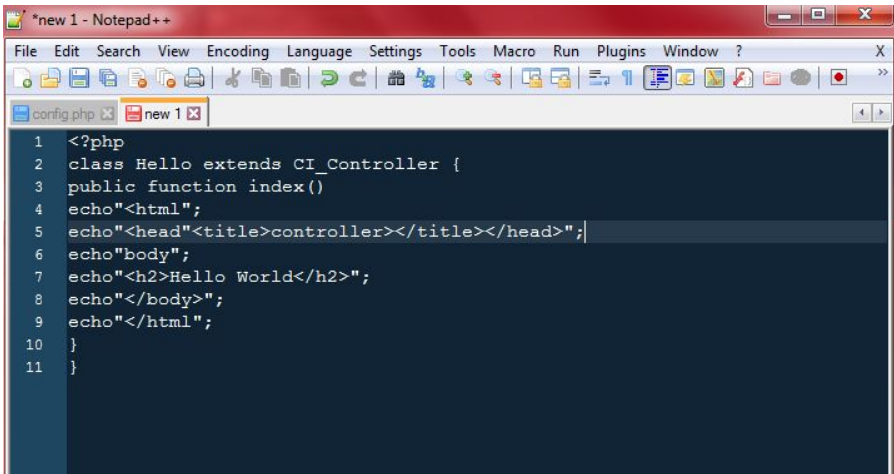
### 2.3.2 View

Proses yang menampilkan hasil output dari suatu model. Cara membuat view pada codeigniter yaitu pada gambar dibawah ini : Setelah membuat kode program seperti gambar diatas simpan *file* diatas dengan nama yang bebas sesuai keinginan masing-masing (misalnya:belajarview.php) dan tempatkan pada direktori berikut ?? :

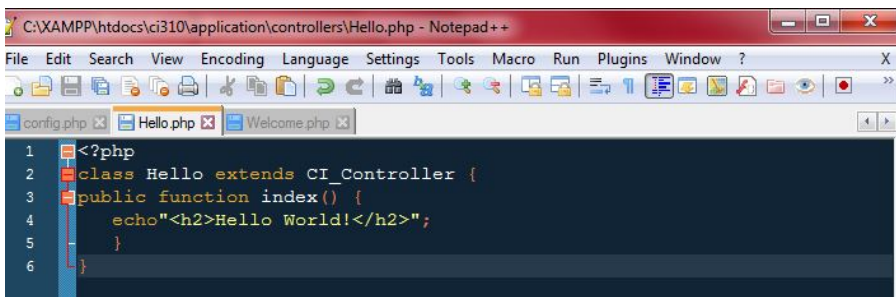
C:\XAMPP\htdocs\ci310\application\views\\seperti pada gambar dibawah ini : Selanjutnya,kita harus mengubah file *Controller*

application\controllers\hello.php\\

dan hasilnya akan menjadi seperti gambar dibawah ini :



**Gambar 2.1** Perintah Hello World Perintah Lengkap



**Gambar 2.2** Perintah Hello World Perintah Simple

## 2.4 Model

Model biasanya digunakan untuk representasi dari struktur data yang terdapat pada CodeIgniter. Aplikasi juga dapat dibuat hanya dengan menggunakan komponen *controller* dan *model* tanpa *view*, meskipun pada praktiknya hal ini seharusnya tidak dilakukan. Untuk membuktikannya, kita dapat membuat *model* (misalnya dengan nama *Hello\_model*) menggunakan kode berikut :

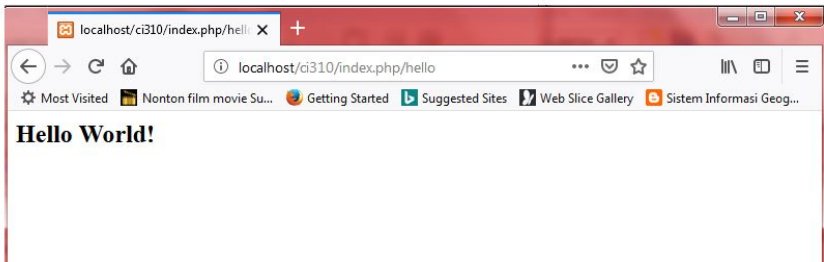
jangan lupa setelah itu kita simpan pada direktori C : \XAMPP\htdocs\ci310\application

Selanjutnya, kita harus mengubah file *Controller*

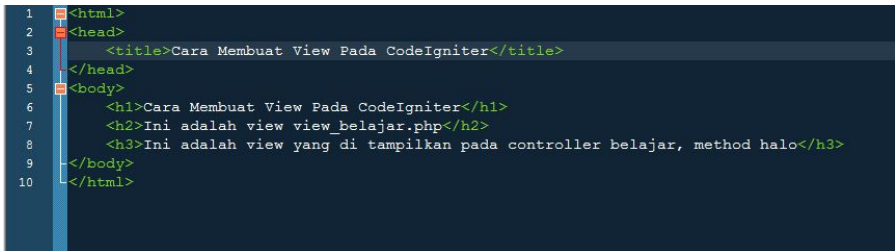
application\controllers\hello.php \\\ seperti pada gambar dibawah ini :

dan setelah itu perintah akan dieksekusi dan hasilnya akan seperti ini :

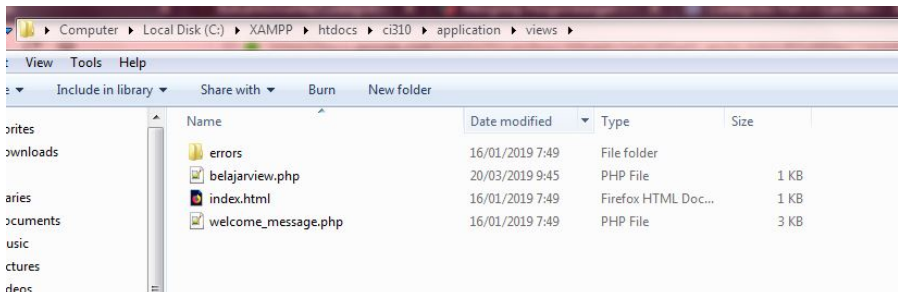




Gambar 2.3 Hasil



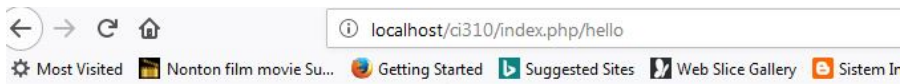
Gambar 2.4 Membuat views



Gambar 2.5 Menyimpan Direktori View

## 2.5 Menggunakan Controller, Model dan View

Aplikasi dengan komponen *Controller*, *model*, *view* merupakan aplikasi yang lengkap. pada praktiknya, teknik inilah yang pada umumnya digunakan oleh aplikasi yang dikembangkan menggunakan CodeIgniter. untuk itu kita akan mencoba mendemonstrasikan dengan melibatkan file ketiga komponen utama ini yaitu??



## Cara Membuat View Pada CodeIgniter

Ini adalah view `view_belajar.php`

Ini adalah view yang di tampilkan pada controller belajar, method hallo

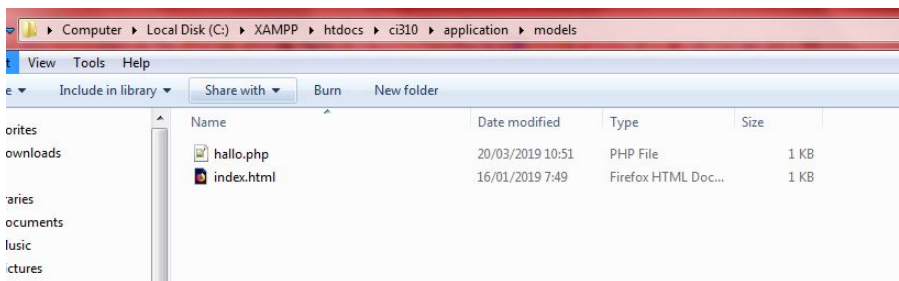
**Gambar 2.6** Hasil dari View

```

1 <?php
2 class Hello_model extends CI_Model {
3
4     //mendefinisikan properti dengan nama $str
5     public $str = 'HALLO SEMUA';
6 }

```

**Gambar 2.7** Membuat model



**Gambar 2.8** Menyimpan model

Kemudian masukan perintah file model  
dan terakhir perintah untuk view  
Maka untuk hasil akhirnya akan seperti ini :

```
<?php
class Hello extends CI_Controller {
    public function index() {
        //memuat model dengan nama 'hallo'
        $this->load->model('hallo_model');

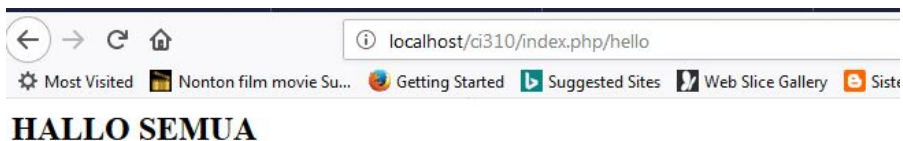
        //menangkap objek dari kelas hallo_model
        // yang telah dimuat ke variabel $model
        $model = $this->hallo_model;

        echo "<html>";
        echo "<head><title>Controller dan Model</title></head>";
        echo "<body>";

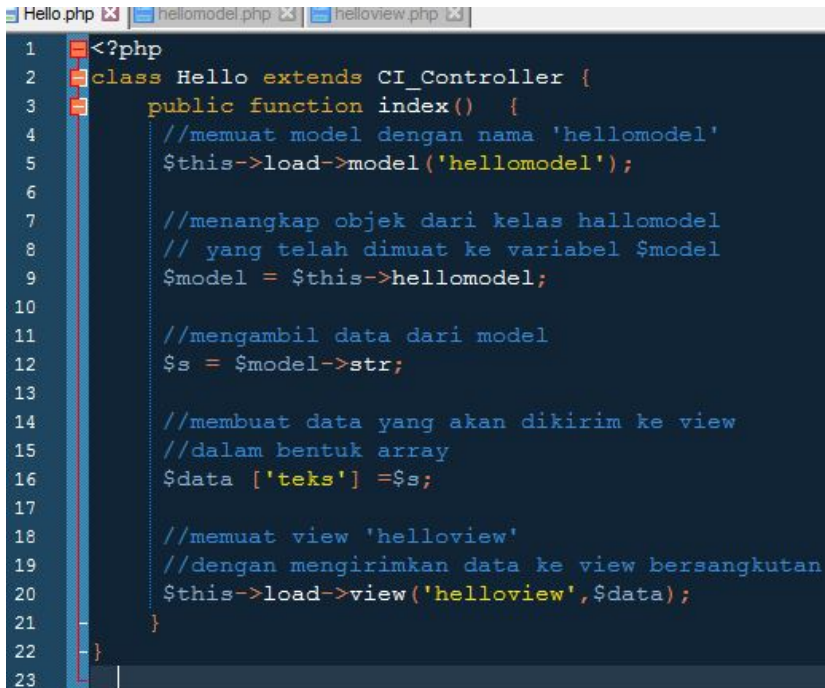
        //memanggil properti $str milik objek $model
        echo "<h2>". $model->str. "</h2>";

        echo "</body>";
        echo "</html>";
    }
}
```

**Gambar 2.9** Mengubah model ke controllerl

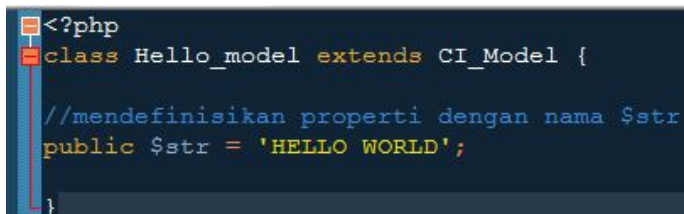


**Gambar 2.10** Mengubah model ke controllerl



```
1 <?php
2 class Hello extends CI_Controller {
3     public function index() {
4         //memuat model dengan nama 'hellomodel'
5         $this->load->model('hellomodel');
6
7         //menangkap objek dari kelas hellomodel
8         // yang telah dimuat ke variabel $model
9         $model = $this->hellomodel;
10
11         //mengambil data dari model
12         $s = $model->str;
13
14         //membuat data yang akan dikirim ke view
15         //dalam bentuk array
16         $data ['teks'] =$s;
17
18         //memuat view 'helloview'
19         //dengan mengirimkan data ke view bersangkutan
20         $this->load->view('helloview',$data);
21     }
22 }
23
```

Gambar 2.11 Controller Hello



```
<?php
class Hello_model extends CI_Model {

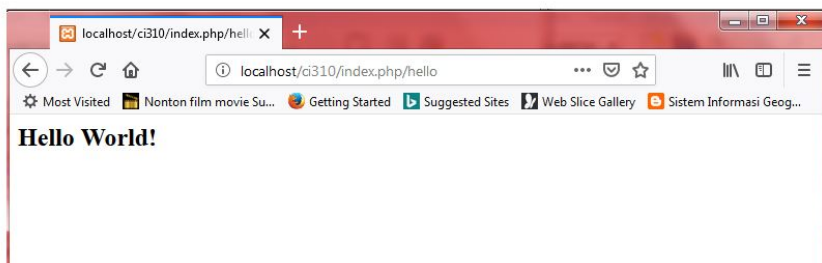
    //mendefinisikan properti dengan nama $str
    public $str = 'HELLO WORLD';

}
```

Gambar 2.12 Model Hello

```
1 <html>
2 <head>
3   <title>Controller,Model,dan View</title>
4 </head>
5 <body>
6   <h2><?php echo,$teks;?></h2>
7 </body>
8 </html>
```

Gambar 2.13 View Hellol



Gambar 2.14 Akhir

## BAB 3

---

# PENGUNAAN CONTROLLER

---

### 3.1 Penggunaan Controller

Perintah navigasi direktori

