

# Modul Pelatihan Framework CodeIgniter



Oleh :

Afwan Auliyar Rakhman, ST.

([www.afwan.net](http://www.afwan.net))

**PASSIONIT**

**“IT Consultant & IT Training Bandung”**

**Head office : Komplek Mekarsari Endah blok A-5 Baleendah.**

**[www. Passionit.co.id](http://www.Passionit.co.id)**

**Bandung, 2011.**

# MENGENAL FRAMEWORK CODE IGNITER (CI)

---

## Apa itu Framework Code Igniter (CI) ?!?

CodeIgniter adalah salah satu framework php yang populer. Framework itu sendiri adalah suatu kerangka kerja yang berupa sekumpulan folder yang memuat file-file php yang menyediakan class libraries, helpers, plugins dan lainnya. Framework menyediakan konfigurasi dan teknik coding tertentu. Code Igniter tergolong framework dengan ukuran kecil dan cukup mudah di kuasai. CI juga datang dengan manual yang tergolong lengkap.

Modul ini menggunakan codeigniter versi 2.0.0, sehingga penggunaan-penggunaan codeigniter dalam membangun aplikasi web menggunakan codeigniter versi 2.0.0.

## Instalasi Framework CI

Setelah mendapatkan file kompresi CI, kemudian ekstrak dan ditempatkan di directory web servernya. Untuk contoh yang saat ini letakkan di directory c:\xampp\htdocs.

## Struktur CI

Karena CI adalah sebuah php framework yang berupa kumpulan folder dan file php, java script,css,txt maka CI dijalankan under web dan harus dengan web server. Program CI cukup di letakkan di bawah folder directory web server anda.

Struktur file dari Framework CI sebagai berikut :

- Application
  - Cache
  - Config
  - Controllers
  - Core
  - Errors
  - Helpers
  - Hooks
  - Language

- Libraries
- Logs
- Models
- Third\_party
- Views
- System
  - Core
  - Database
  - Font
  - Helpers
  - Language
  - Libraries
- User\_guide

Untuk bisa membangun web, file-file konstruksinya diletakkan di folder Application. Struktur aplikasi CI ada di dalam folder **application/**. Berikut adalah folder-folder yang sering digunakan dalam pembangunan web dengan CI :

a. Config

Merupakan folder untuk konfigurasi aplikasi yang akan dibangun. Di folder ini diletakkan konfigurasi tentang base URL, database, routing URL.

b. Controllers

Merupakan folder yang berisi file-file control, dimana nantinya file-file didalam folder ini akan dieksekusi ketika pemanggilan di URL.

c. Errors

Merupakan folder error handling, yang mengatur ketika aplikasi yang dibangun mengalami error.

d. Helpers

Merupakan folder yang berisi fungsionalitas yang memudahkan developer untuk membangun aplikasi lebih mudah.

e. Models

Merupakan folder yang berisi fungsionalitas untuk pemodelan terhadap aplikasi yang nantinya akan dibangun. Yang bisa dieksekusi oleh controller.

f. Views

Merupakan folder yang berisi tentang file-file template untuk tampilan dari website yang akan dibangun.

## Setting Framework CI

Untuk mensetting framework CI, dilakukan di folder system/application/config. Kemudian edit file config.php dengan mengeset base URLnya seperti : <http://localhost/ci/> (d disesuaikan dengan nama foldernya ).

## User guide CI

Sebagai framework, CI memberikan user guide yang lengkap dengan langkah-langkah nya dalam satu paket download CI. User guide ini, digunakan untuk memandu kita dalam develop web menggunakan CI. Untuk bisa menggunakan user guide ini cukup masuk melalui browser, kemudian ketikkan alamat dimana web server CI berada dengan menambahkan /user\_guide diakhirnya, contoh : [http://localhost/ci/user\\_guide](http://localhost/ci/user_guide) .

# MEMBEDAH ARSITEKTUR CODE IGNITER (CI)

---

## Skema pemanggilan CI

Untuk memanggil aplikasi website yang dibangun menggunakan CI sesuai dengan arsitektur pemanggilan di browser adalah sebagai berikut :

`http:// [server_web]/[folder]/index.php/[controller]/[function]`

contoh : <http://localhost/ci/index.php/bio/hello>

Default [function] adalah index, jika tidak di panggil, contoh :

<http://localhost/ci/index.php/bio> sama dengan <http://localhost/ci/index.php/bio/index>

Nilai default yang dipanggil oleh CI adalah controller.

## Konsep Pemodelan CI

Framework CI menggunakan konsep MVC dimana konsep ini memisahkan antara logic, database dan tampilan. Keuntungan menggunakan MVC akan membuat coding logic lebih simple, karena dipisah dengan coding tampilannya dan membuat programmer dapat bekerja terpisah dengan designer. Programmer mengerjakan logic dan designer mengerjakan coding tampilan.

Model merupakan code struktur data, berupa modul-modul fungsi pengolahan data. Script sql dapat diletakkan disini. View merupakan code untuk menampilkan tampilan suta program. Tampilan dapat berupa Web page, header, footer dan apa saja yang berjenis tampilan. Controller merupakan code untuk logic, algoritma dan sebagai penghubung antara model, view, dan sumber lain yang di perlukan untuk mengolah HTTP request dan generate web page.

CI menerapkan pola MVC yang flexible, karena model dapat tidak digunakan. Kita dapat hanya menggunakan Controller dan View saja dalam menggunakan CI tanpa Model. Jika kita tidak memerlukan pemisahan didalam struktur data maka kita dapat tidak menggunakan model dalam pembangunan web dengan CI.

## Controller

Controller pada CI memiliki fungsi pengendali logic yang mengatur antara pemodelan database dan tampilan. Pada CI controller dibuat didalam folder system/application/controllers, dengan aturan pembuatan controller sebagaia berikut :

- Nama controller harus diawali huruf besar dan sesuai dengan nama file.
- Nama constructor bisa mengikuti dengan nama controllernya.

Contoh pembuatan controller :

```
<?php
class Hello extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
    }

    function index()
    {
    }
}
?>
```

Class Hello yang merupakan class controller yang dibangun, sedangkan function Hello merupakan konstruktor mengikuti penulisan dari class controller di atasnya.

## View

Membuat view sama seperti membangun html biasa. Karena view ini yang nantinya akan bisa digunakan untuk menampilkan dari layout aplikasi website yang akan dibangun. Di view inilah script-script javascript bisa dimasukkan, css, maupun flash juga bisa di tempatkan di view ini.

Contoh pembuatan view :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>My test CI</title>
</head>
<body>
Hello world !!
```

```
</body>
</html>
```

Fungsi dari view adalah menangkap variable yang berasal dari controller. Variable tersebut bisa langsung di tampilkan yang disesuaikan dengan layoutnya.

## Model

Membuat class model hampir sama dengan membuat controller, hanya saja yang berbeda adalah parameter extends dan class parent nya. Model ini lebih didefinisikan sebagai sekumpulan fungsi yang bisa diakses oleh banyak controller. Dengan begitu model lebih merepresentasikan fungsi yang sama. Contoh pembuatan model adalah sebagai berikut :

```
<?php
class Biodata extends CI_Model
{
    function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    function insert($data)
    {
        $this->db->insert('biodata',$data);
    }
}
?>
```

Fungsi yang ada dimodel adalah fungsi general yang akan dipakai oleh banyak controller, seperti akses database, maupun fungsi-fungsi general lainnya.

## “Hello World” di CI

Sebagai langkah awal kita akan membuat “hello world” yang akan dieksekusi oleh browser ketika dipanggil dan memunculkan tulisan “hello world”.

- a. Masuk ke direktori system/application/controllers dan buatlah file php didalamnya. Contoh :  
hello.php
- b. Kemudian buka file tersebut dengan editor, dan masukkan code seperti berikut :

```
<?php
class Hello extends CI_Controller
{
    Function index()
    {
        $this->load->view('helloWorld');
    }
}
?>
```

Untuk penulisan code yang harus diperhatikan :

- Nama class harus sama dengan nama file dan diawali dengan huruf besar, contoh :  
class Hello
  - Function index adalah function default
- c. Masuk ke system/application/views, dan buatlah file dengan nama helloWorld.  
Nama file disesuaikan dengan nama view yang dipanggil di controller, yang dipanggil dengan code \$this->load->view('[nama\_file\_view]');
  - d. Kemudian buka file tersebut dengan editor, kemudian tuliskan code sebagai berikut :

```
<?php
Echo "Hello World";
?>
```

- e. Kemudian buka browser, lalu akses melalui URL [http://localhost/\[nama\\_folder\]/index.php/hello](http://localhost/[nama_folder]/index.php/hello)
  - [Nama\_folder] diisi dengan alamat folder dimana CI diletakkan
  - hello merupakan nama controller yang sebelumnya dibuat.

## Helper

Helper merupakan pengkodean yang bisa membantu pengembang web sehingga pengkodean lebih simple dan lebih ringkas. Selaian disediakan oleh CI, kita juga bisa membuat helper sendiri dengan mengikuti aturan yang ada di helper-helper CI lainnya. Referensi untuk bisa menggunakan helper yang



disediakan oleh CI bisa dilihat di user guide dengan table of contents adalah helper reference. Beberapa helper yang disediakan oleh CI yang sering digunakan :

#### A. Form helper

Form helper digunakan untuk membantu pembuatan form, dan untuk bisa menggunakan form helper load terlebih dahulu dengan code : `$this->load->helper('form')`.

Helper yang bisa digunakan :

Helper	keterangan	Contoh
<code>form_open('controller/fungsi')</code>	Helper untuk membuka form	<code>echo form_open('input/data')</code>
<code>form_input('name','value')</code>	Helper untuk inputan berupa text	<code>echo form_input('username')</code>
<code>form_password('name','value')</code>	Helper untuk inputan berupa password	<code>echo form_password('pass')</code>
<code>form_submit()</code>	Helper untuk memberikan button submit	<code>echo form_submit()</code>
<code>form_close()</code>	Helper untuk menutup form.	<code>echo form_close()</code>
<code>form_open_multipart('action')</code>	Sama seperti <code>form_open</code> , tetapi fungsinya khusus untuk upload file.	<code>echo form_open_multipart('input/data')</code>
<code>form_hidden('name','value')</code>	Helper untuk inputan berupa text yang tersembunyi	<code>echo form_hidden('id',1234)</code>

Untuk lebih lengkap form helper apa saja yang bisa digunakan, dapat dilihat di user guide di table of content : helper reference > form helper. Contoh penggunaan form helper :

```
<?php echo form_open('form/kali');?>
<br />
Form Perkalian
<br />
<br />
```

```

<p>Nilai Pertama :
<?php echo form_input('nilai1'); ?>
</p>
    <p>Nilai Kedua :
        <?php echo form_input('nilai2'); ?>
    </p>
<p>
    <?php echo form_submit('submit','Kali'); ?>
</p>
<?php echo form_close(); ?>

```

## B. URL helper

URL helper digunakan untuk memudahkan pengembang yang bekerja dengan URL,, dan untuk bisa menggunakan URL helper load terlebih dahulu dengan code :

```
$this->load->helper('url').
```

Helper	Keterangan	contoh
base_url()	Mengembalikan nilai ke base_url yang di setting di config	echo base_url()
site_url()	Mengembalikan nilai site_url yang disetting di config. Mengarah pada URL server yang dipakai	echo site_url('bio/data') artinya : http://example.com/index.php/bio/data
anchor('URL','text')	Membuat standar html anchor link	echo anchor('bio/data','click') artinya : <a href="bio/data">click</a>
Redirect('URL')	Untuk mendirect ke fungsi tertentu di controller tertentu.	redirect('/article/13', 'location', 301);

Untuk lebih lengkap URL helper apa saja yang bisa digunakan, dapat dilihat di user guide di table of content : helper reference > URL helper.

## C. HTML helper

HTML helper digunakan untuk memudahkan pengembang yang bekerja dengan tag-tag HTML, dan untuk bisa menggunakan HTML helper, load terlebih dahulu dengan code :  
`$this->load->helper('html')`.

Beberapa helper yang sering digunakan :

- `img()`

Helper untuk meload gambar. Tag HTML `` merupakan tag yang biasanya digunakan untuk menampilkan gambar. Contoh penggunaan helper `img()`, adalah :

```
$image_properties = array(
    'src' => 'images/picture.jpg',
    'alt' => 'just picture',
    'class' => 'post_images',
    'width' => '200',
    'height' => '200',
    'title' => 'just title'
);
echo img($image_properties);
```

Source diatas sama seperti code dibawah ini :

```

```

- `link_tag()`

Untuk menggantikan tag `<link />`, biasanya digunakan untuk menyebutkan link untuk css, favicon, dll. Contoh penggunaan :

```
echo link_tag('css/mystyles.css');
```

artinya sama dengan :

```
<link href="http://site.com/css/mystyles.css"
rel="stylesheet" type="text/css" />
```

- `br()`

untuk menggantikan tag `<br />`, contoh penggunaan :

```
echo br(3);
```

Artinya sama dengan :

<br/> <br/> <br/>

Untuk lebih lengkap HTML helper apa saja yang bisa digunakan, dapat dilihat di user guide di table of content : helper reference > HTML helper.

#### D. Email Helper

Untuk bisa menggunakan email helper, load terlebih dahulu dengan code sebagai berikut : `$this->load->helper('email')`.

Helper yang bisa digunakan :

- `valid_email('email')`, berfungsi untuk mengecek format email yang benar. Contoh :  

```
$this->load->helper('email');  
if (valid_email('get.motivasy@gmail.com'))  
{  
    echo 'email is valid';  
}  
else  
{  
    echo 'email is not valid';  
}
```
- `send_email('recipient','subject','message')`, berfungsi untuk mengirimkan email.

## Form

Form merupakan bentuk interaksi user dengan system. Dimana user bisa menginputkan data, berupa text, password, pilihan, seleksi, maupun check box. Pembangunan form pada CI bisa menggunakan helper ataupun tanpa menggunakan helper.

- a. Buatlah sebuah controller, diberi nama form.php
- b. Isilah form.php dengan code class controller sebagai berikut :

```
<?php  
Class Form extends CI_Controller  
{  
    function __construct()  
{
```

```

parent::__construct();
$this->load->helper('form');
}

function index()
{
    $this->load->view('formData');
}

function kali()
{
    $nilai1=$this->input->post('nilai1');
    $nilai2=$this->input->post('nilai2');
    $data['nilai1']=$nilai1;
    $data['nilai2']=$nilai2;
    $data['hasil']=$nilai1*$nilai2;
    $this->load->view('hasilData',$data);
}
}
?>

```

Keterangan :

- `$this->load->helper('form')` merupakan inisiasi helper yang akan dipakai di form
  - `$this->load->view('formData')` merupakan pemanggilan view formData.php
  - `$this->load->view('hasilData',$data)` merupakan pemanggilan view hasilData.php dengan membawa nilai dari nilai-nilai yang diinputkan oleh user dan hasil dari perkalian keduanya. Dimana `$data` merupakan variable pembawanya.
  - `$data['nilai1']` , `$data['nilai2']`, `$data['hasil']` merupakan variable array penampung data yang akan di bawa ke view.
- c. Buatlah 2 buah view di system/application/view dan dinamai formData.php dan hasilData.php
- d. Untuk formData.php berfungsi untuk menampilkan form inputan, isilah dengan code berikut :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

```

```

<title>Test CI</title>
</head>
<body>
<?php echo form_open('form/kali');?>
    <br />
    Form Perkalian
    <br />
    <br />
    <p>Nilai Pertama :
    <?php echo form_input('nilai1'); ?>
    </p>
    <p>Nilai Kedua :
    <?php echo form_input('nilai2'); ?>
    </p>
    <p>
    <?php echo form_submit('submit','Kali'); ?>
    </p>
    <?php echo form_close(); ?>
    <p>&nbsp;</p>
</body>
</html>

```

Keterangan :

- form\_open adalah code untuk membuka form, dengan parameter action berupa form\_open('controller/fungsi').
- form\_input adalah form untuk inputan berupa teks.
- form\_submit merupakan button submit pada sebuah form inputan
- form\_close merupakan code untuk menutup form

- e. Untuk hasilData.php berfungsi menampilkan proses yang sudah dilakukan oleh controller sebelumnya, isilah dengan code sebagai berikut :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
Hasil dari <?php echo $nilai1; ?> X <?php echo $nilai2; ?> adalah <?php
echo $hasil; ?>
<p>&nbsp;</p>
</body>
</html>

```

- f. Untuk pengaksesannya menggunakan URL yang sama seperti pada pemanggilan “hello world” sebelumnya dengan dirubah pada controller dan fungsinya saja. Seperti ini :

<http://localhost/ci/index.php/form>

## Library

Library merupakan class yang sudah dibangun oleh CI. Library ini sifatnya akan mempermudah pembangunan aplikasi website. Library juga bisa dibuat oleh sendiri. Secara umum, library ini letaknya di system/library, dan untuk menggunakan library, diinisiasi terlebih dahulu di konstruktor classnya.

Berikut beberapa library yang sudah disediakan oleh CI :

### A. Library database

Library database ini memungkinkan komunikasi dengan database lebih mudah. Untuk inisiasi dari library ini dengan memasukkan code : `$this->load->database()` di class konstrukturnya.

Berikut beberapa list untuk penggunaan dari library database ini :

Code	Keterangan
<code>\$this-&gt;db-&gt;query([query])</code>	Untuk mengeksekusi sebuah query, [query] diganti dengan query database. Contoh : <code>\$this-&gt;db-&gt;query('select * from biodata')</code>
<code>\$this-&gt;db-&gt;get('[nama_tabel]')</code>	Untuk menampilkan semua isi database.[nama_tabel] diganti dengan tabel yang akan ditampilkan. Contoh : <code>\$this-&gt;db-&gt;get('biodata')</code>
<code>\$this-&gt;db-&gt;where('[kondisi]')</code>	Untuk memberikan nilai kondisi pada sebuah database. Contoh :

	<code>\$this-&gt;db-&gt;where('data',1)</code> Sama dengan where data=1
<code>\$this-&gt;db-&gt;select()</code>	Untuk menyeleksi data-data apa yang akan ditampilkan. Nilai defaultnya adalah * artinya akan menyeleksi semua field dari sebuah database
<code>\$this-&gt;db-&gt;from('['tabel]')</code>	Untuk memilih database mana yang akan di eksekusi. Contoh : <code>\$this-&gt;db-&gt;from('biodata')</code>
<code>\$this-&gt;db-&gt;insert('tabel','data')</code>	Untuk memasukkan data ke database. Contoh : <code>\$this-&gt;db-&gt;insert('biodata',array('nama'=&gt;'afwan'))</code>
<code>\$this-&gt;db-&gt;delete('tabel')</code>	Untuk menghapus data yang diinginkan. Biasanya digabung dengan kondisi tertentu. Contoh : <code>\$this-&gt;db-&gt;where('id',1);</code> <code>\$this-&gt;db-&gt;delete('biodata');</code>
<code>\$this-&gt;db-&gt;update('tabel','data')</code>	Untuk mengupdate nilai dari sebuah database. Biasanya digabung dengan kondisi tertentu. Contoh : <code>\$this-&gt;db-&gt;where('id',1);</code> <code>\$this-&gt;db-&gt;update('biodata',array('nama','afwan'));</code>

Penggunaan library database banyak di gunakan pada class model. User guide untuk library database bisa dilihat di tabel : class reference > Database class . Contoh library database di model :

## B. Library session

Session berguna untuk menyimpan data pengunjung situs secara otomatis. Untuk penggunaan library session ini menggunakan `$this->load->library('session')` di dalam konstruktor classnya.

Contoh penggunaan library session :

```

$data=array(
    'nama'=>'afwan',
    'logged_in'=>TRUE
);
$this->session->set_userdata($data);

```



Logged\_in disetting bernilai true dan ini harus di masukkan dalam nilai array nya.

Sedangkan untuk memanggil session yang sudah dibuat dengan cara sebagai berikut :

```
$this->session->userdata('[nama_array]')
```

contohnya : `$this->session->userdata('nama')`

### C. Library upload file

Library ini merupakan library untuk upload file ke server dan disimpan didalam folder tertentu.

Untuk bisa menggunakan library ini, kita bisa mengkonfigurasinya sendiri sesuai dengan parameter tertentu yang kita inginkan. Berikut setting library upload file :

```
$config['upload_path'] = './uploads/';  
$config['allowed_types'] = 'gif|jpg|png';  
$config['max_size'] = '100';  
$config['max_width'] = '1024';  
$config['max_height'] = '768';  
$config['overwrite'] = true;  
$this->load->library('upload', $config);
```

Parameter-parameter yang bisa digunakan adalah :

Parameter	Keterangan
Upload_path	Merupakan alamat folder dimana file yang dikirim akan disimpan. Bisa berupa absolute path atau relative path
Allowed_types	Jenis file yang bisa di upload
Max_size	Ukuran maksimal file yang bisa dikirim (dalam kilobytes).
Max_width	Ukuran lebar maksimal file yang bisa dikirim (dalam pixel)
Max_heiht	Ukuran tinggi maksimal file yang bisa dikirim (dalam pixel)
Overwrite	Nilainya berupa true atau false. Dimana file yang dikirim akan ditimpa jika nilainya true, jika false maka file yang namanya sama tidak akan di simpan

Setelah setting untuk upload selesai, maka gunakan code : `$this->upload->do_upload()` untuk mengeksekusi file yang telah diupload. Dan code : `$this->upload->data()` untuk mengambil informasi data yang telah diupload. Form yang digunakan untuk upload file berupa form multipart. Contoh penggunaan library class :

- Siapkan folder untuk menampung file yang akan diupload, contoh : folder 'upload', dan diletakkan di bawah root.

- view untuk mengupload file, beri nama upload\_form.php :

```
<html>
<head>
<title>Upload Form</title>
</head>
<body>
<?php echo $error;?>
<?php echo form_open_multipart('upload/do_upload');?>
<input type="file" name="userfile" size="20" />
<br /><br />
<input type="submit" value="upload" />
</form>
</body>
</html>
```

- view ketika sukses upload, diberi nama upload\_success.php :

```
<html>
<head>
<title>Upload Form</title>
</head>
<body>
<h3>Your file was successfully uploaded!</h3>
<ul>
<?php foreach($upload_data as $item => $value):?>
<li><?php echo $item;?>: <?php echo $value;?></li>
<?php endforeach; ?>
</ul>
<p><?php echo anchor('upload', 'Upload Another File!'); ?></p>
</body>
</html>
```

- Controller yang akan mengeksekusi upload file, diberi nama upload.php :

```
<?php
```

```

class Upload extends CI_Controller {
    function __construct()
    {
        parent::__construct();
        $this->load->helper(array('form', 'url'));
    }

    function index()
    {
        $this->load->view('upload_form', array('error' => ' ' ));
    }

    function do_upload()
    {
        $config['upload_path'] = './uploads/';
        $config['allowed_types'] = 'gif|jpg|png';
        $config['max_size']      = '100';
        $config['max_width']    = '1024';
        $config['max_height']    = '768';
        $this->load->library('upload', $config);
        if ( ! $this->upload->do_upload())
        {
            $error = array('error' => $this->upload->display_errors());
            $this->load->view('upload_form', $error);
        }
        else
        {
            $data = array('upload_data' => $this->upload->data());
            $this->load->view('upload_success', $data);
        }
    }
}
?>

```

## Desain & layout di CI

Desain dan layout diletakkan di view, dan tidak menggunakan aturan yang terlalu kompleks, sama seperti pembangunan tampilan di web biasa. Beberapa hal yang bisa dilakukan oleh view untuk membuat layout di CI, adalah :

### a. Insert CSS

Memasukkan css apa yang akan dipakai bisa menggunakan helper atau tidak. Contoh :

- Menggunakan helper  

```
<?php echo link_tag('css/mystyles.css'); ?>
```
- Tidak menggunakan helper  

```
<link href="<?php=base_url()?>css/mystyles.css" rel="stylesheet" type="text/css" />
```

Keterangan :

- Base\_url() akan memberikan nilai yang mengacu pada settingan di config.
- Folder css diletakkan sejajar dengan folder system

### b. Insert Javascript

Menggunakan javascript bisa langsung dimasukkan dengan code berikut :

```
<script type="text/javascript" src="<?php=base_url();?>js/test.js">
</script>
```

Keterangan :

- Folder js diletakkan sejajar dengan system.

### c. Passing variable

Passing variable merupakan variable yang dilempar dari controller untuk ditampilkan di view.

Contoh penggunaan passing variable :

- Di controller  

```
$data['nilai']='afwan auliyar';
$this->load->view('viewData',$data);
```
- Di view  

```
echo $nilai;
```

Keterangan :

- Nilai variable di controller harus berupa array dengan indeks tertentu, dalam contoh diatas `$data['nilai']`.
- Variable di view merupakan indeks dari array variable yang dilempar ke view. Dalam contoh diatas 'nilai' adalah indeks dari `$data`, maka 'nilai' tersebut dijadikan variable di view.

**d. Penempatan image**

Untuk bisa menggunakan image, bisa menggunakan URL helper. Folder image bisa ditempatkan sejajar dengan folder system dan bisa menggunakan helper `base_url()` sebagai alamat URL image yang akan di gunakan.

# MEMBANGUN APLIKASI WEB DENGAN CODE IGNITER (CI)

---

Pada bagian ini kita akan mencoba membangun sebuah arsitektur web menggunakan CI. Aplikasi yang akan dibangun kali ini merupakan aplikasi “Pendaftaran Pelatihan Online”. Aplikasi ini menggambarkan proses pendaftaran untuk mengikuti pelatihan secara online. Langkah-langkah pembuatan aplikasi :

- a. Menyiapkan database ‘pelatihan’, dengan membuat tabel peserta, tabel pelatihan dengan kolomnya sebagai berikut :

Tabel peserta :

Field	type	attribute
Id_biodata	int	Primary, auto_increment
Id_pelatihan	int	
nama	Varchar(30)	
alamat	Varchar(100)	
email	Varchar(20)	
No_hp	Varchar(20)	

Tabel pelatihan:

Field	type	attribute
Id_pelatihan	int	Primary, auto_increment
Nama_pelatihan	Varchar(50)	
biaya	Varchar(20)	
tentor	Varchar(50)	
quota	Int(5)	

- b. Mengatur url\_base di config.php dan database di database.php yang kedua file tersebut di system/application/config.
- c. Membuat model peserta, kemudian buatlah fungsi-fungsi sebagai berikut :

- insertData : isilah dengan fungsi untuk memasukkan data ke tabel peserta
- updateData : isilah dengan fungsi untuk mengupdate data ke tabel peserta
- deleteData : isilah dengan fungsi untuk menghapus data ke tabel peserta
- getAllData : isilah dengan fungsi untuk menampilkan semua data dari tabel peserta
- getData : isilah dengan fungsi untuk menampilkan sebuah data yang spesifik dari tabel peserta

d. Membuat model pelatihan, kemudian buatlah fungsi-fungsi dasar sebagai berikut :

- insertData : isilah dengan fungsi untuk memasukkan data ke tabel pelatihan
- updateData : isilah dengan fungsi untuk mengupdate data ke tabel pelatihan
- deleteData : isilah dengan fungsi untuk menghapus data ke tabel pelatihan
- getAllData : isilah dengan fungsi menampilkan semua data dari tabel pelatihan
- getData : isilah dengan fungsi untuk menampilkan sebuah data yang spesifik dari tabel pelatihan

e. Membuat controller cPeserta, kemudian buatlah fungsi di controller sebagai berikut :

- Index : berisi halaman home, deskripsi tentang pelatihan
- viewDaftar : berfungsi untuk menampilkan form pendaftaran, dengan mempassing data dari model pelatihan
- viewEdit : berfungsi untuk menampilkan form edit peserta
- daftarPeserta : berfungsi untuk memasukkan data dari form pendaftaran
- viewAllPeserta : berfungsi untuk memperlihatkan data peserta keseluruhan
- viewPeserta : berfungsi untuk memperlihatkan peserta per pelatihan
- editPeserta : berfungsi untuk mengedit peserta yang sudah mendaftar.

f. Membuat controller cPelatihan, kemudian buatlah fungsi di controller sebagai berikut :

- Index : berisi halaman untuk menampilkan seluruh data pelatihan
- viewCreate : berfungsi untuk menampilkan form untuk create pelatihan
- viewEdit : berfungsi untuk menampilkan form edit pelatihan
- createPelatihan: berfungsi untuk memasukkan data dari form create
- viewAllPelatihan : berfungsi untuk memperlihatkan data pelatihan keseluruhan

- viewPelatihan : berfungsi untuk memperlihatkan detail dari setiap pelatihan
  - editPelatihan : berfungsi untuk mengedit pelatihan yang sudah ada.
- g. Membuat view dengan membuat file di system/application/views formDaftar.php, viewPeserta.php, formEditPeserta.php, home.php, viewPelatihan.php, viewCreatePelatihan.php, formEditPelatihan.php . Deskripsi masing-masing file :
- formDaftar : berisi sebuah form untuk pendaftaran peserta pelatihan meliputi nama, alamat, email, no HP dan pelatihan
  - viewPeserta.php : untuk melihat data peserta yang ikut per pelatihan.
  - formEditPeserta.php : untuk mengedit data peserta yang sudah ikut mendaftar
  - home.php : untuk menampilkan deskripsi pelatihan di halaman depan
  - viewPelatihan.php : untuk menampilkan semua data pelatihan yang ada
  - viewCreatePelatihan.php : untuk membuat daftar pelatihan baru
  - formEditPelatihan.php : untuk mengedit data pelatihan yang ada

Pembahasan detail code akan diberikan melalui saat sesi pelatihan.

#### Referensi :

- a. [www.ilmukomputer.com](http://www.ilmukomputer.com)
- b. Code Igniter user guide, [http://www.codeigniter.com/user\\_guide](http://www.codeigniter.com/user_guide), Lisensi : [http://codeigniter.com/user\\_guide/license.html](http://codeigniter.com/user_guide/license.html)
- c. Membangun elemen form dengan codeigniter, <http://www.a2fahmi.com>