

Tutorial Code Igniter

1. Pendahuluan

Mengapa kita butuh framework?

Bagi anda yang belum familiar dengan framework, framework adalah sekumpulan fungsi, class, dan aturan-aturan. Berbeda dengan library yang sifatnya untuk tujuan tertentu saja, framework bersifat menyeluruh mengatur bagaimana kita membangun aplikasi. Framework memungkinkan kita membangun aplikasi dengan lebih cepat karena sebagai developer kita akan lebih memfokuskan pada pokok permasalahan sedangkan hal-hal penunjang lainnya seperti koneksi database, form validation, GUI, dan security; umumnya telah disediakan oleh framework. Disamping itu dengan aturan-aturan yang jelas dan harus dipatuhi, aplikasi kita lebih solid, more readable, dan kolaborasi dalam tim dapat lebih mudah dilaksanakan.

Kita sebagai seorang software developer bisa dianalogikan sebagai seorang tukang bangunan. Apabila anda perhatikan, seorang tukang bangunan bisa membuat sebuah rumah. Tidak akan menjadi masalah bila hanya untuk membangun rumah dengan satu atau dua lantai. Tetapi akan menjadi masalah apabila dia mendapatkan pekerjaan untuk membangun sebuah gedung bertingkat. Permasalahan akan menjadi semakin komplek, makin banyak pekerja dan material yang dilibatkan, belum lagi dengan jadwal yang ketat. Kita pun seperti itu. Membangun aplikasi kecil tentu tidak menjadi masalah. Namun bagaimana bagaimana apabila aplikasi kecil kita tersebut dengan makin lama makin bertambah requirementnya sejalan dengan kebutuhan user. Di sini lah peran penting sebuah framework dalam membangun aplikasi.

2. Pembahasan

2.1 Mengetahui Code Igniter

Code Igniter (CI) (www.codeigniter.com) adalah salah satu framework php yang tangguh dan populer yang awalnya ditulis oleh mas Rick Ellis, pendiri dan CEO EllisLab.com, perusahaan yang mengembangkan codeigniter. Saat ini, codeigniter dikembangkan oleh komunitas dan disebarluaskan ke seluruh dunia dengan lisensi bebas. Code Igniter tergolong framework dengan ukuran kecil dan cukup mudah di kuasai. CI juga datang dengan manual yang tergolong lengkap. Code Igniter adalah sebuah framework PHP. 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.

2.2 Mendapatkan Code Igniter

Code Igniter dapat di peroleh secara gratis, dengan mendownload nya di www.codeigniter.com. Versi saat ini adalah versi 1.7.2.

2.4 Kebutuhan Code Igniter

Untuk menggunakan codeigniter dengan maksimal, anda akan memerlukan:

- PHP 4.3.2 atau lebih baru
- Web server, saya rekomendasikan [Apache](#)
- [MySQL](#), MySQLi, MS SQL, Postgre, Oracle, SQLite atau ODBC
- Codeigniter php framework. Anda bisa mendownloadnya di <http://codeigniter.com/download.php>

2.5 Instalasi Code Igniter

Karena berupa folder php, maka instalasi Code Igniter, cukup dengan mengekstrak file kompresi yang di dapat dari situs www.codeigniter.com dan menempatkannya di dalam folder directory web server anda. Kemudian edit file config.php yang ada di application/config/config.php dan set base URL anda, misal : <http://localhost/ci>. Jika anda hendak menggunakan database, edit file database.php yang ada di application/config/database.php. Setting database anda di file tsb.

Setelah di lakukan instalasi dan setting, test lah Code Igniter anda dengan membuka browser anda dan mengetikkan url yang anda taruh di file config.php di atas, misal : <http://localhost/ci>. Jika muncul halaman welcome ala Code Igniter, maka Code Igniter telah siap di gunakan.

2.6 Konsep Model-View-Controller (MVC)

Konsep MVC adalah konsep pemisahan antara logic dengan tampilan dan database.

Manfaat konsep ini adalah, membuat coding logic lebih simple, karena sudah di pisah dengan code untuk tampilan dan membuat programmer dapat bekerja secara terpisah dengan designer. Programmer mengerjakan logic, sedangkan designer berkutut dengan design dan tampilan.

Model → Merupakan code struktur data. Model berisi fungsi di dalam pengolahan database.

Script Sql masuk di sini.

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.

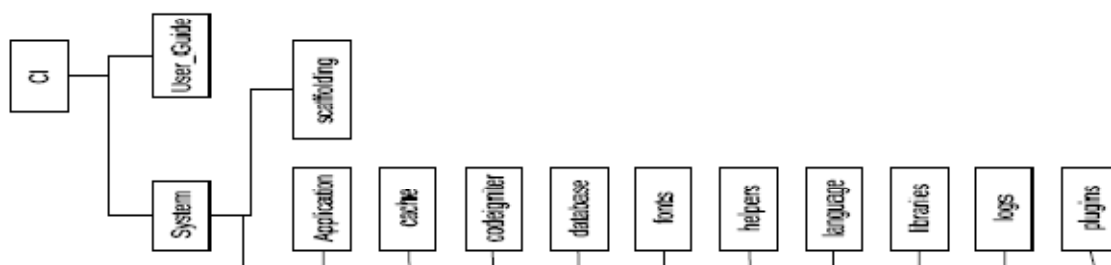
CI menerapkan pola MVC yang flexible, karena model dapat tidak di gunakan. Anda dapat hanya menggunakan Controller dan View saja dalam menggunakan CI tanpa Model. Jika anda tidak memerlukan pemisahan di dalam struktur data dan database atau menganggap penggunaan model hanya menambah kompleks aplikasi dengan keuntungan yang kurang sebanding, maka anda dapat tidak menggunakan model.

2.7 Struktur file CI

CI adalah sebuah php framework yang berupa kumpulan folder dan file php, java script,css,txt dan file berbasis web lainnya dengan setting tertentu untuk menggunakannya dan menyediakan library dan helper yang dapat di manfaatkan di dalam pemrograman php.

CI di jalankan under web dan harus dengan web server. Program CI cukup di letakkan di bawah folder directory web server anda.

Berikut adalah struktur file CI :



2.8 Coding dengan Code Igniter

2.8.1 Membuat “Hello World” di CI

Berikut kita akan mempraktekan pemrograman php dengan CI. Kita akan menampilkan tulisan “Hello World” ke dalam browser kita. Langkah-langkah nya adalah sebagai berikut :

a. Buka Text Editor anda. Lalu buatlah sebuah controller di dalam direktory CI anda, yaitu di : system/application/controllers.

b. Tulis lah code sbb :

```
<?php
class Hello extends controller
{
function index()
{
$this->load->view('helloview');
}
}?
```

Beri nama file hello.php.

Yang musti di perhatikan :

- Nama class harus di awali huruf besar.
- Nama class harus sama dengan nama file.
- Function index adalah function default. Di url tidak perlu di tuliskan nama function index nya. Kalau nama function selain index, perlu di tulis di url nya.

c. Buat file baru dengan nama helloview.php di system/application/views

Tulis code sbb :

```
<?
echo “Hello World”;
?>
```

d. Misal base_url : <http://localhost/ci>, maka jalankan di url anda menjadi : <http://localhost/ci/hello>. Maka akan muncul hasil nya.

2.8.2 Membuat Form dengan CI

Form pada CI dapat di buat dengan menggunakan helper atau dengan syntax html biasa.

Pada kesempatan kali ini, kita akan mencoba membuat form dengan menggunakan helper.

Buatlah sebuah controller. Ketik code berikut :

```
<?php
class Form extends Controller
{
function index()
{
$this->load->view('form');
}
function jumlah()
{
$angka1=$this->input->post('angka1');
$angka2=$this->input->post('angka2');
$hasil=$angka1+$angka2;
echo "Hasil nya adalah : $hasil";
echo "<br />".anchor("form/index","Kembali");
}
} ?>
```

Beri nama file “form.php”.

Buat sebuah view dengan nama “form.php” . Ketikkan code sbb :

```
<html>
<head>
<title>Membuat Form input penjumlahan dengan CI</title>
</head>
<body>
<table>
<? echo form_open('form/jumlah');
    $data1=array(
        'name' => 'angka1',
        'size'=>'15');
    echo "<tr><td>Angka 1</td><td> :</td><td>".form_input($data1)."</td></tr>";
    $data2=array(
        'name' => 'angka2',
        'size'=>'15');
```

```

        echo "<tr><td>Angka                2</td><td>                :</td><td>
".form_input($data2)."</td></tr>";
        echo "<tr><td>".form_submit('mysubmit','Ok') ;
        echo " ".form_reset('mysubmit','Clear')." </td></tr>";
    ?>
</table>
</body>
</html>

```

Pada contoh di atas kita load helper di file autoload.php untuk load secara otomatis.

```
$this->load->helper('form');
```

Helper yang kita gunakan, kita input di view. Syntax yang di gunakan adalah :

```
- form_open ('form/jumlah')
```

Syntax di atas sama dengan syntax html berikut :

```
<form method="post" action="http://localhost/ciprivate/index.php/form">
```

atau, gunakan base_url(), untuk mengantisipasi perubahan pada url.

```
<form method="post" action="<?=base_url();?>index.php/form">
```

Nilai base_url() di atas mengambil nilai base_url di file config.php .

```
- form_input ('username','sofwan')
```

Minimal, kita memasukkan nama field dan value pada syntax form_input. Jika menggunakan parameter yang lebih banyak, yaitu :

```

$data1=array(
    'name' => 'angka1',
    'size'=>'15');
echo form_input($data1);

```

```
- form_submit('my submit','OK');
```

identik dengan :

```
<input type="submit" value="OK">
```

```
- form_reset
```

Sama penggunaan nya dengan form_reset.

2.8.3. Menggunakan library CI

A. Library Database.

Menggunakan library ini, kita perlu mengaktifkan

```
$this->load->database() atau dengan $this->
```

Untuk loading otomatis, kita perl mengaktifkan library tsb dengan cara memberikan perintah

```
$this->load->library('database');
```

A.1. Query standard dengan hasil multiple (object

```
$query="select nama,alamat,telp from mahasiswa";
```

```
$hasil=$this->db->query("$query");
```

```
foreach ($hasil as $row)
```

```
{  
    echo $row->nama;  
    echo $row->alamat;  
    echo $row->telp;  
}
```

```
echo "Hasil Nilai :". $hasil->num_rows();
```

A.2. Query standard dengan hasil multiple (Array version)

```
$query="select nama,alamat,telp from mahasiswa";
```

```
$hasil=$this->db->query("$query");
```

```
foreach ($hasil as $row)
```

```
{  
    echo $row['nama'];  
    echo $row['alamat'];  
    echo $row['telp'];  
}
```

```
echo "Hasil Nilai :". $hasil->num_rows();
```

A.3. Query dengan hasil tunggal

Untuk menampilkan sebuah hasil query, kita dapat menggunakan syntax sbb :

```
$query->row()
```

Contoh :

```
$query="select nama from mahasiswa limit 1";
```

```
$hasil=$this->db->query("$query");
```

```
$hasilnya=$hasil->row();
```

```
echo $hasilnya->nama;
```

A.4. Query Bindings

Yaitu membuat query dengan nilai yang dapat berubah-ubah.

```
$sql= "select * from mahasiswa where nama=? And kota=?";
```

```
$this->db->query($sql,array('andi','jakarta'));
```

A.5. Active Record Class

CI menggunakan teknik active record, di samping juga dapat memproses perintah sql. Cara ini adalah cara lain yang di gunakan di CI untuk memproses query.

Perintah-perintah di dalam Active record class adalah sbb :

A.5.1 Selecting Data

```
- $this->db->get()
```

Fungsi : Untuk menampilkan semua isi tabel mahasiswa.

Contoh :

```
- $this->db->get('mahasiswa'); // Sama dengan : select * from mahasiswa.
```

Parameter pertama dan kedua memuat limit dan offset, yaitu :

```
- $query=$this->db->get('mytable',10,20)
```

Menghasilkan : *select * from mytable limit 10,20*

```
- $this->db->get_where();
```

Fungsi : Untuk menampilkan semua isi tabel dengan di tambah kondisi where

Contoh :

```
$this->db-> get_where ('mytable',array('id'=>$id),$limit,$offset);
```

// Sama dengan : *"select * from mytable where id='\$id' limit 20, 10";*

```
- $this->db->select();
```

Fungsi : Untuk memilih tabel yang akan di proses dengan perintah select.

Contoh :

```
$this->db->select('nama,nilai,alamat');
```

```
$query=$this->db->get('mahasiswa');
```

// Sama dengan : *select nama,nilai,alamat from mahasiswa;*

```
- $this->db->from();
```

Fungsi : Untuk memilih tabel.

Contoh :

```
$this->db->select('nama,nilai,alamat');  
$this->db->from('mahasiswa');  
$query=$this->db->get();  
// Sama dengan : select nama,nilai,alamat from mahasiswa;
```

- *\$this->db->join();*

Fungsi : Untuk melakukan perintah join terhadap 2 atau lebih tabel.

Contoh :

```
$this->db->select(“*”);  
$this->db->from(“nilai”);  
$this->db->join(“mahasiswa”, “mahasiswa.nim=nilai.nim”);  
$query=$this->db->get();  
// Sama dengan : “select * from nilai join mahasiswa on mahasiswa.nim=nilai.nim”;
```

- *\$this->db->where();*

Fungsi : Untuk menerapkan kondisi where suatu syntax query.

Contoh :

```
$this->db->where('nama','opan');  
$query= $this->db->get('mahasiswa');  
// sama dengan : “select * from mahasiswa where nama='opan'”;
```

- *\$this->db->like();*

Fungsi : Menyatakan syntax like ke dalam query.

Contoh :

```
$this->db->like('nama','sofwan');  
$query=$this->db->get(“mahasiswa”);  
// sama dengan : “select * from mahasiwa where nama like '%sofwan%'”;
```

- *\$this->db->group_by()*

Fungsi : Menambahkan perintah group by pada query.

Contoh :

```
$this->db->group_by(“kota”);  
$query=$this->db->get(“mahasiswa”);  
// Sama dengan : select * from mahasiswa group by kota”;
```

A.5.2 Inserting Data

- `$this->db->insert();`

Fungsi : Untuk menginsert data ke dalam sebuah tabel.

Kita dapat menggunakan data yang akan di insert berupa array atau object.

Contoh, menggunakan array :

```
$data=array(
    'nim'=>'0811500292',
    'nama'=>'ali',
    'kota'=>'jakarta');
$this->db->insert('mahasiswa',$data);
```

Contoh menggunakan Object :

```
class kelasku
{
    var $nim="0811500292",
    var $nama="ali",
    var $kota="jakarta"
}
$obj=new kelasku;
$this->db->insert("mahasiswa",$obj);
Kedua contoh di atas sama dengan perintah :
// insert into mahasiswa (nim,nama,kota) values('0811500292','ali','jakarta');
```

- `$this->db->set();`

Fungsi : Fungsi ini mengambil data untuk di lakukan perintah insert dan update.

Contoh : `$this->db->set('nama',$nama);`

`$this->db->insert('mahasiswa');`

// Sama dengan : `insert into mahasiswa (nama) values ('{$nama}');`

A.5.3 Updating Data

- `$this->db->update();`

Fungsi : Untuk update data

```
$data=array(
    'nim'=>'$nim',
    'nama'=>'$nama',
    'kota'=>'$kota');
$this->db->where('id',$id);
```

```
$this->db->update('mahasiswa',$data);  
// Sama dengan : update mahasiswa set nim="$nim",nama="$nama",kota="$kota"  
where id="$id";
```

Atau dapat juga dengan menggunakan object.

```
Class kelasku {  
    var $nim="$vnim",  
    var $nama="$vnama",  
    var $kota="$vkota";  
}  
$object = new kelasku;  
$this->db->where ('id',$id);  
$this->db->update ("mahasiswa",$object);
```

A.5.4. Deleting Data

- *\$this->db->delete()*

Fungsi : Menghapus data di dalam query

contoh :

```
$this->db->delete("mahasiswa",array('nim'->$nim));  
// Sama dengan : delete mahasiswa where nim="$nim"
```

B. Library E-mail

Berikut adalah contoh setting e-mail :

```
$this->load->library("email");  
// Kita perlu meload library e-mail di dalam controller atau di taruh di autoload.php  
$this->email->from("sofwan@sofwan.net","Akhmad Sofwan");  
$this->email->to("sofwan@sofwan.net");  
$this->email->cc("wansofwan@yahoo.co.id");  
$this->email->bcc("wansofwan@gmail.com");  
$this->email->subject("Test Email");  
$this->email->message("Hallo apa kabar nich ... ?");  
$this->email->send();
```

Setting e-mail di set secara default di dalam file config.php. Contoh melakukan setting Manual :

```
$config["protocol"]="sendmail";  
$config["mailpath"]="/usr/sbin/sendmail";  
$config["charset"]="iso-8859-1";  
$config["wordwrap"]=TRUE;
```

```
$this->email->initialize($config);
```

C.Library Uploading File

Adalah library untuk upload file ke server dan di simpan di folder tertentu. Folder tempat penyimpanan file di buat write accessible. Ukuran max di php.ini juga di atur agar sesuai dengan yang kita inginkan. Perhatikan baris berikut pada php.

```
file_uploads=on // Buat menjadi on
```

```
post_max_size = 2M // Edit ukuran file nya sesuai dengan yang anda inginkan.
```

```
upload_tmp_dir="/opt/lampp/htdocs" // Folder ini harus write accessible
```

Setting upload nya adalah sbb :

```
$config['upload_path']='./uploads/'; // Folder penyimpanan file. Musti Write accessible
```

```
$config['allowed_types']='gif|jpg|png';
```

```
$config['max_size']='100';
```

```
$config['max_width']='1024';
```

```
$config['max_height']='768';
```

```
$this->load->library('upload',$config);
```

// Jika kita meletakkan library uploading file di dalam file autoload.php, maka syntax di bawah ini yang perlu kita taruh :

```
$this->upload->initialize($config);
```

```
$this->upload->do_upload()
```

Fungsi : Untuk melakukan eksekusi uploading file Contoh :

```
<?php
```

```
class Upload extends Controller {
```

```
function Upload()
```

```
{
```

```
    parent::Controller();
```

```
    $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);
}
}
}
?>

```

Agar form dapat melakukan uploading file, maka tipe enctype suatu form haruslah berbentuk multipart/form-data.

```
<form method="post" enctype="multipart/form-data">
```

D.Pagination Library

Pagination umum nya di gunakan di dalam penampilan suatu database yang berbentuk tabel.

Contoh penggunaan Pagination :

```

$this->load->library("pagination"); // Loading library pagination
$config["base_url"] = "http://localhost/dkpnew"; // base_url
$config["total_rows"] = "100"; // Total baris
$config["per_page"] = "20"; // per page
$this->pagination->initialize($config);
echo $this->pagination->create_links();

```

E. HTML Table Library

Library ini membuat tabel secara otomatis dari data yang anda retrieve.

Contoh :

```
$this->load->library("table");  
$data=array(  
    array("Nim","Nama","Kota"),  
    array("090012","Andi","Jakarta"),  
    array("090014","Umar","Jakarta"),  
    array("090013","Udin","Semarang"));  
echo $this->table->generate($data);
```

Menampilkan tabel dari data yang di peroleh dari database. Contoh nya adalah sbb :

```
$this->load->library("table");  
$query=$this->db->query("select * from mahasiswa");  
echo $this->table->generate($query);
```

F. Session Library

Session berguna untuk menyimpan data pengunjung situs ke dalam cookie secara otomatis.

Contoh Session :

```
$data=array(  
    'nim'=>'0900189',  
    'nama'=>'Andi Hasan',  
    'logged_in'=>TRUE);  
$this->session->set_userdata($data);
```

G. Menghapus session

```
$this->session->unset_userdata('some_name');
```

- Flash data

Flashdata atau session data hanya tersedia untuk request server selanjutnya dan kemudian di hapus secara otomatis. Hal ini sangat berguna dan di gunakan untuk informasi atau status pesan.

Untuk menambah flash data :

```
$this->session->set_flashdata("item","value");
```

Untuk membaca flashdata variabel, dapat di gunakan :

```
$this->session->flashdata("item");
```

Untuk menghancurkan session.

```
$this->session->sess_destroy();
```

2.8.4. Menggunakan Helper CI

A. E-mail Helper

Helper E-mail ini di sediakan untuk membantu developer bekerja dengan E-mail.

Syntax yang ada pada helper ini adalah :

1. `$this->load->helper("email");`

Untuk loading helper email.

2. `valid_email('email');`

Untuk mengecek format e-mail yang benar.

Contoh :

```
if (valid_email("sofwan@sofwan.net"))
    { echo "Mail is valid"; }
else
    { echo "Mail is invalid"; }
```

3. `send_email("recipient","subject","message")`

Syntax untuk Mengirim e-mail.

B. Form Helper

Helper Form ini di sediakan untuk membantu developer bekerja dengan form.

Syntax yang ada pada helper ini adalah :

1. `form_open()`

Fungsi : Untuk membuat form pada awal form.

Sama dengan `<form method="post" action="<?=base_url();?>index.php/input">`

2. `form_open_multipart()`

Fungsi : Sama dengan `form_open()`.

3. `form_input()`

Fungsi : Untuk membentuk fungsi input text

Contoh : `$data=array("name" => "username", "size" => "10");`

`echo form_input ($data)`

Sama dengan : `<input type="text" name="username" size="10">`

4. `form_password()`

Sama dengan `form_input`. Hanya saja form ini masukkan nya berupa password.

Sama dengan : `<input type="password">`

C.Cookie Helper ()

Cookie Helper ini di sediakan untuk membantu developer bekerja dengan cookie.

Syntax yang ada pada helper ini adalah :

`$this->load->helper("cookie");`

1. `set_cookie()`

Fungsi : Setting cookie

2. `get_cookie()`

Fungsi : Fetch Cookie

3. `delete_cookie()`

Fungsi : Menghapus Cookie

D. URL Helper()

URL Helper ini berguna untuk membantu Developer bekerja dengan URL.

Syntax yang ada di dalam URL Helper, yaitu :

1. `base_url()`

Fungsi : Mengembalikan base_url, seperti yang tertulis di dalam file config.php

2. `uri_string()`

Fungsi : Mengembalikan URI Segment di halaman yang berisi fungsi ini. Jika URL anda : <http://sofwan.net/blog/comments/123> ,

Fungsi akan mengembalikan nilai : blog/comments/123

3. `anchor()`

Fungsi : membuat standard html anchor link berdasarkan url local anda.

Contoh :

```
echo anchor ("berita/local/123","berita");
```

```
<a href="http://localhost/local/123" >Click Here</a>
```

4. `redirect()`

Fungsi : Untuk redirect ke function tertentu.

```
redirect("form/index");
```

Maka akan menuju ke syntax tertentu.

E. Security Helper()

1. `xss_clean()`

Menyediakan Cross Site Script Hack filtering. Function ini termasuk juga ke dalam input class.

2. `dohash()`

Akan membentuk SH1 atau MD5 encrypting password.

Contoh : `$str =dohash($str) // SHA1`

`$str =dohash($str,'md5') // MD5`

2.8.5. Insert CSS dan JavaScript di CI.

Insert CSS dan Java Script di CI di letakkan di view, yaitu dengan cara :

- **Insert CSS :**

```
<link rel="stylesheet" href="<?=base_url();?>css/mainutama.css" type="text/css"
media="screen, projection" />
```

Pada script di atas, file css dengan nama [file:mainutama.css](#) yang berada di folder css di insert.

Letak folder css sendiri ada nya di root.

- **Insert JavaScript :**

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

Pada script di atas, file javascript dengan nama [file:addhtmlcontrols.js](#) yang berada di folder javascript di insert. Letak folder javascript sendiri ada nya di root.

2.8.6. Membuat Tampilan Design.

Tampilan design di biasanya di buat dengan menggunakan CSS dan html/xhtml. File Css dan html/xhtml sendiri di letakkan atau di insert pada view. Image juga dapat di letakkan pada folder terpisah dan di letakkan di hierarki utama CI, sama level nya dengan system, namun dapat juga di letakkan di mana saja, asalkan penunjukkan url nya sesuai.

2.9. Setting di CI dan lainnya di CI

2.9.1 Menggunakan library Third party

CI dapat di extend, yang salah satu nya dengan menggunakan library dari luar. Cara nya yaitu dengan mengcopy file library ke dalam folder system/libraries.

Berikut adalah contoh penerapan penggunaan library report pdf. Aplikasi ini adalah library php yang berfungsi untuk membentuk file pdf, termasuk penggunaan report pdf yang dapat di cetak.

Cara-cara nya adalah sbb, yaitu :

- Download aplikasi Cezpdf dari <http://www.ros.co.nz/pdf>
- Rename file : class.ezpdf.php menjadi cezpdf.php. Copy file cezpdf.php dan class.pdf.php ke dalam folder system/libraries.
- Load library cezpdf, dengan syntax : `$this->load->library('cezpdf.php');`
- Library pdf dapat di gunakan. Cara penggunaan nya dengan membaca readme.pdf yang di sertakan di aplikasi hasil download nya.

2.9.2. Upload CI di Web Hosting

Setelah suatu project web sudah kita selesaikan di localhost atau web server local. Kita akan menaikkan/upload aplikasi CI tsb ke Web Hosting kita.

Cara untuk upload ke web hosting adalah sbb :

- a. Copy kan keseluruhan folder CI ke dalam folder directory web hosting kita.
- b. Edit file config.php dan database.php. Di sesuaikan dengan konfigurasi web hoting dan database nya. Misal url kita adalah : <http://www.budi.com>, maka di base_url yang ada di file **config.php** nilai nya adalah <http://www.budi.com>
- c. Edit file routes.php yang ada di folder /system/application/config.

Ubah nilai \$route['default_controller'] nya menjadi controller default yang kita miliki. Hal ini akan berdampak, jika kita mengetik url : <http://www.budi.com>, maka akan langsung tampil page utama dari web kita.

2.9.3. URI Routing

Reserved Routes. Terdapat 2 reserved routes, yaitu :

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

Pada contoh di atas, default controller yang di gunakan adalah class welcome. Jika anda menggunakan routes yang berbeda, maka hendak nya nilai 'default_controller' di atas dapat segera di ubah, agar jika anda memanggil url nya saja, maka akan langsung mengarah ke default controller anda.

2.9.4. Membuat Library sendiri

Library di simpan di dalam folder system/libraries. Di dalam hal mengubah library, kita dapat melakukan hal-hal berikut, yaitu :

- a. Membuat Library baru.
- b. Menambah/extends native library.
- c. Merubah Native Libraries.

2.9.5. Loading Plugin

Plugin hampir sama dengan Helper, bedanya Plugin memiliki 1 function, sedangkan Helper terdiri dari kumpulan function. Cara loading plugin, yaitu :

```
$this->load->plugin('name').
```

Plugin juga dapat di load secara otomatis dengan cara memasukkan array plugin di file autoload.php di folder system/application/config.

2.9.6. Scaffolding

Scaffolding di gunakan untuk bekerja dengan sebuah tabel dan dapat di add, edit dan delete. Penggunaan Scaffolding untuk mempermudah proses pembuatan aplikasi tsb.

Penggunaan Scaffolding di letakkan di Controller. Berikut adalah contoh penerapan Scaffolding di dalam CI, yaitu :

```
<?
class Blog extends Controller
{
    function Blog()
    {
        parent::Controller();
        $this->load->scaffolding('table_name');
    }
}
?>
```

Sumber

- Belajar PHP dengan Framework
<http://ilmukomputer.org/wp-content/uploads/2010/05/belajar-php-dengan-framework-code-igniter.pdf>
- SEKAPUR SIRIH TENTANG FRAMEWORK CI (CodeIgniter)
<http://oke.or.id/wp-content/plugins/downloads-manager/upload/file47dc937797808.pdf>
- Griffiths, Adam. 2010. CodeIgniter 1.7 Professional Development. Birmingham : Packt Publishing Ltd.