**Bab I**

**DASAR-DASAR CODEIGNITER**

Pada bagian ini akan dijelaskan apa-apa saja yang diperlukan untuk mengembangkan aplikasi dengan menggunakan framework Codeigniter, setting Codeigniter, mencoba membuat kode sederhana dan studi kasus untuk melatih penggunaan helper dan library.

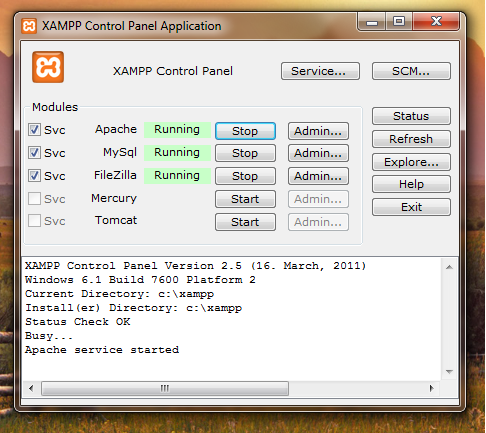
1. **Kebutuhan**

Berikut ini adalah kebutuhan yang diperlukan untuk menggunakan framework Codeigniter :

1) XAMPP (Apache, MySQL, PHP)

2) Framework Codeigniter 3.1.0

3) PHP Editor (Notepad ++, PHP Designer, Dreamweaver, dsb)

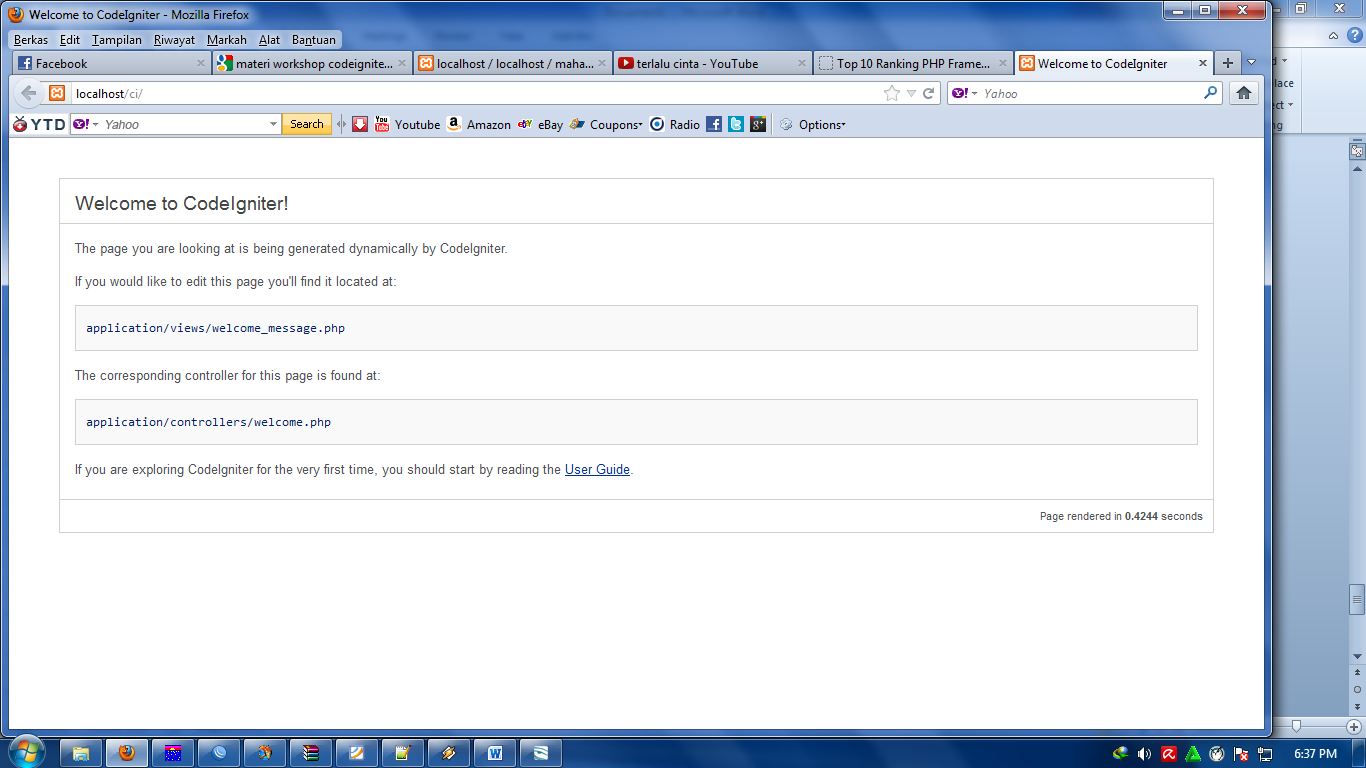
****

1. **Setting Framework Codeigniter**

Untuk melakukan setting framework Codeigniter, lakukan langkah-langkah berikut :

1. Setelah Xampp terinstal dengan baik, Ekstrak file CodeIgniter\_3.1.0.zip ke direktori xampp/htdocs dan remane menjadi Ci agar lebih memudahkan kita.
2. Lakukan testing dengan mengetikan alamat http://localhost/ci pada address bar di browser.

Berikut ini adalah tampilan yang pada browser jika framework Codeigniter telah berjalan dengan baik :



Gambar 2. 1 Tampilan Jika Konfigurasi Berhasil

**Bab II**

**HELLO CODEIGNITER**

Pada bab ini kita akan memulai coding menggunakan Codeigniter. Kita akan membuat program pertama untuk menampilkan tulisan **Hello CodeIgniter** dengan Controller dan View.

1. **Controller**

Dengan menggunakan Text Editor buatlah sebuah file dengan nama Mahasiswa.php dan simpan di dalam folder Application/ Controller. Setelah itu buatlah sebuah kelas Mahasiswa yang merupakan turunan dari kelas CI\_Controller.

1. <?php
2. Class Mahasiswa extends Ci\_Controller
3. {
4. Function \_\_construct()
5. {
6. Parent::\_\_construct();
7. }
8. Function index()
9. {
10. Echo “<h1> Hello CodeIgniter</h1>”;
11. }
12. }

Setelah itu apa yang terjadi? Coba buka situs Anda dengan menggunakan URL seperti ini:

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



Catatan : ketentuan penamaan class (baik controller maupun library) harus dimulai dengan huruf besar. Contoh:

<?php

Class **Mahasiswa** Extends ci\_controller {

}

Berikut contoh yang salah

<?php

Class **mahasiswa** extends ci\_controller{

}

1. **Controller dan View**

Pada contoh program Hello CodeIgniter diatas adalah cara untuk menampilkan tulisan "hello CodeIgniter" secara langsung di controller. Namun sebenarnya hal tersebut bisa dilakukan di view. Sebagai contoh: tuliskan kode program di bawah ini pada file

application/controller/mahasiswa.php :

1. <?php
2. Class Mahasiswa extends Ci\_Controller
3. {
4. Function \_\_construct()
5. {
6. Parent::\_\_construct();
7. }
8. Function index()
9. {
10. $this->load->view(‘hello\_codeigniter’);
11. }
12. }

Selanjutnya buatlah file hello\_CodeIgniter.php di folder application>view

(application/view/hello\_codeIgniter.php) yang berisi tulisan:

<h1>Hello saya adalah view</h1>

<h1>Hello saya adalah Chan</h1>

1. **Membuat Fungsi baru**

Untuk mengakses fungsi – fungsi yang ada pada suatu halaman ( controllers ), kita dapat mengetikkannya di belakang nama url file. Pada contoh diatas kita membentuk function index ( fungsi default yang ditampilkan ketika halaman hello\_codeigniter.php dibuka ), maka kita tidak perlu menuliskan index/ di belakang url nya. Untuk lebih jelasnya mengenai hal ini, mari kita buka kembali file mahasiswa.php pada editor kita. Kemudian buat fungsi baru dengan nama “fungsibaru”. Sehingga scripnya menjadi seperti berikut :

1. <?php
2. Class Mahasiswa extends Ci\_Controller
3. {
4. Function \_\_construct()
5. {
6. Parent::\_\_construct();
7. }
8. Function index()
9. {
10. $this->load->view(‘hello\_codeigniter’);
11. }
13. Function **fungsibaru()**
14. {
15. $this->load->view(‘view\_baru’);
16. }
17. }

Seperti biasa buatlah file untuk menampilkan halaman baru kita, simpan di folder application/views dan beri nama dengan view\_baru.php, berikut script file view\_baru.php

<h1>Ini adalah Fungsi Baru</h1>



1. **Membuat file Model**

Sebelum kita mulai membuat suatu model, ada beberapa hal yang harus kita perhatikan yaitu pada folder application/config, disana kita harus melalukan beberapa konfigurasi.

* Pada file autoload.php, cari kode baris 61

$autoload['libraries'] = array('');

Ubah menjadi seperti berikut

$autoload['libraries'] = array('database');

Dan juga pada baris 92

$autoload['helper'] = array(' ');

Ubah menjadi seperti berikut

$autoload['helper'] = array('url','form');

Pada file autoload.php ini kita melakukan konfigurasi untuk library ataupun helper yang akan kita panggil secara otomatis ketika program codeigniter kita dijalankan.

* Pada file database.php

Di file database.php ini mengatur tentang urusan koneksi framework codeigniter dengan database.

$db['default']['hostname'] = 'localhost';

$db['default']['username'] = ''; // sesuaikan dengan webserver anda

$db['default']['password'] = '';

$db['default']['database'] = ''; // nama database yang telah dibuat

$db['default']['dbdriver'] = 'mysql';

$db['default']['hostname'] = 'localhost';

$db['default']['username'] = 'root'; // sesuaikan dengan webserver anda

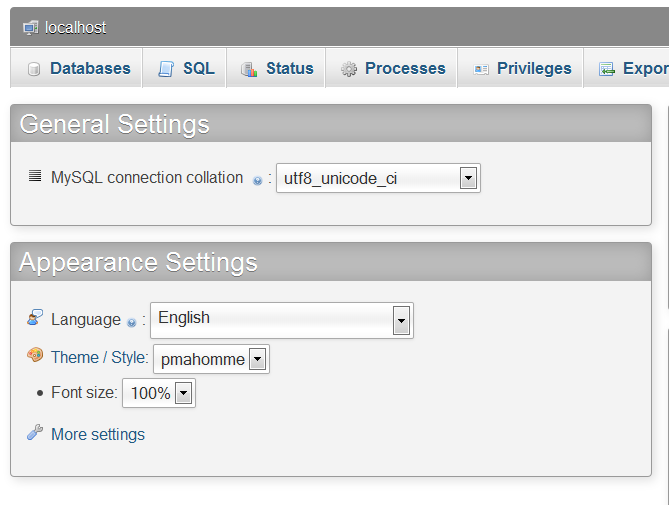
$db['default']['password'] = '';

$db['default']['database'] = 'ci'; // nama database yang telah dibuat

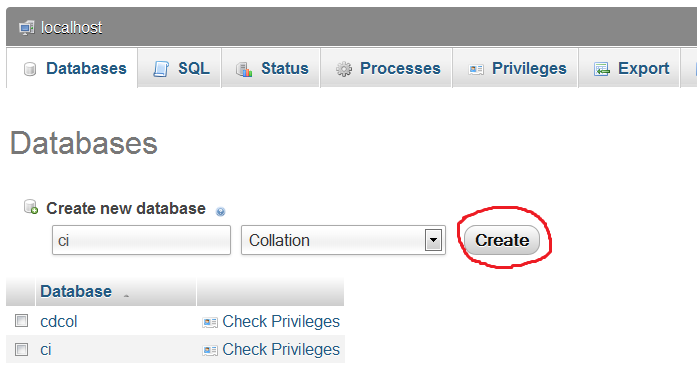
$db['default']['dbdriver'] = 'mysql';

Setelah kita mengkonfigurasi kedua file diatas, sekarang kita buat database

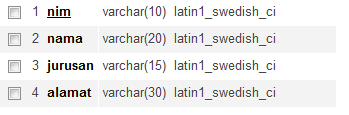
dengan nama “ ci “, karena pada file database.php pada bagian $db['default']['database'] = 'ci'; kita memberi namanya dengan ci, sekarang buka web browser kita, masukkan alamat <http://localhost/phpmyadmin>



Kemudian kita buat databasenya



Setelah buat database, buatlah sebuah table dengan nama mahasiswa



Kemudian sisipkan beberapa record seperti berikut :



Sekarang semua telah diatur dengan baik, sekarang kita buat file model kita, beri nama model\_mahasiswa.php dan simpan di application/model

Script **Model\_mahasiswa.php**

1. <?php
2. Class Model\_mahasiswa extends ci\_model
3. {
4. Function tampilData()
5. {
6. $query=$this->db->get(‘mahasiswa’);
7. If ($query->num\_rows()>0)
8. {
9. Return $query->result();
10. }
11. Else
12. {
13. Return array();
14. }
15. }

Setelah itu kita kembali ke file mahasiswa.php ( controller ). Kita akan meletakkan kode agar controller dapat mengambil data dari database yang sudah di deklerasikan pada file model.hapus beberapa baris kode yang telah kita buat sebelumnya sehingga skrip mahasiswa.php akan seperti berikut :

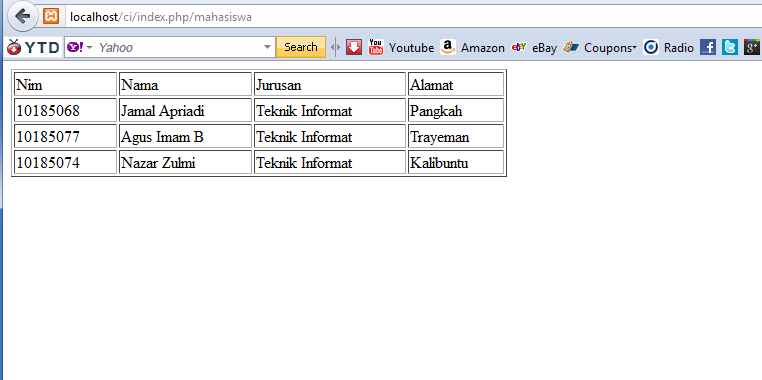
1. <?php
2. class Mahasiswa extends ci\_controller
3. {
4. function \_\_construct()
5. {
6. parent::\_\_construct();
7. $this->load->model('model\_mahasiswa');
8. // adalah sebagai konstruktor berbagai perintah yang ada di
9. // dalam fungsi ini akan selalu dipanggil di saat objek Data\_mahasiswa dibuat
10. }
11. function index()
12. {
13. $data['data']=$this->model\_mahasiswa->tampilData();
14. $this->load->view('mahasiswa',$data);
15. }
16. }

Selanjutnya, kita akan membuat file view. Kita akan meletakkan kode pada file view ini untuk menampilkan setiap data yang masuk, yaitu data yang ada di database ci table mahasiswa, simpan di folder application/views dengan nama mahasiswa.php

**Berikut script mahasiswa.php**

1. <!DOCTYPE HTML>
2. <html>
3. <head>
4. <title>Data Mahasiswa</title>
5. </head>
6. <body>
7. <table width="40%" border="1">
8. <tr>
9. <td>Nim</td>
10. <td>Nama</td>
11. <td>Jurusan</td>
12. <td>Alamat</td>
13. </tr>
14. <tr>
15. <?php foreach ($data as $row):?>
16. <td><?php echo $row->nim;?></td>
17. <td><?php echo $row->nama;?></td>
18. <td><?php echo $row->jurusan;?></td>
19. <td><?php echo $row->alamat;?></td>
20. </tr>
21. <?php endforeach;?>
22. </table>
23. </body>
24. </html>

Sekarang coba jalankan di web browser anda dengan mengetikkan alamat url <http://localhost/ci/index.php/mahasiswa>



**Bagian III**

**Operasi CRUD**

**( CREATE, READ, UPDATE, DELETE )**

CRUD merupakan singkatan dari **C**reate, **R**ead, **U**pdate dan **D**elete. Operasi CRUD merupakan operasi yang total membutuhkan sebuah database. Kali ini kita akan membahas bagaimana cara melakukan operasi CRUD database menggunakan CodeIgniter.

Hal pertama yang akan kita lakukan adalah **memastikan bahwa library database sudah otomatis di – load ketika aplikasi yang akan kita buat dijalankan.** Untuk itu buka file **autoload.php**  yang terdapat di folder application/config. Pastikan library database dan helper sudah di-load secara otomatis. Perhatikan dua baris berikut :

1. **Model CRUD**

Pertama – tama, kita akan membuat sebuah model. Buat sebuah file baru dengan nama **Model\_crud.php,** tuliskan beberapa fungsi CRUD, lengkapnya perhatikan script berikut :

**Script Model\_crud.php**

1. **<?php**
2. **class Model\_crud extends ci\_model**
3. **{**
4. **function tampilData()**
5. **{**
6. **$query=$this->db->get('mahasiswa');**
7. **if($query->num\_rows()>0)**
8. **{**
9. **return $query->result();**
10. **}**
11. **else**
12. **{**
13. **return array();**
14. **}**
15. **}**
16. **function tambah($data)**
17. **{**
18. **$tambah=$this->db->insert('mahasiswa',$data);**
19. **return $tambah;**
20. **}**
21. **function per\_id($id)**
22. **{**
23. **$this->db->where('nim',$id);**
24. **$query=$this->db->get('mahasiswa');**
25. **return $query->result();**
26. **}**
27. **function hapus($id)**
28. **{**
29. **$this->db->where('nim',$id);**
30. **$hapus=$this->db->delete('mahasiswa');**
31. **return $hapus;**
32. **}**
33. **function update($id,$data)**
34. **{**
35. **$this->db->where('nim',$id);**
36. **$update=$this->db->update('mahasiswa',$data);**
37. **return $update;**
38. **}**
39. **}**

1. **Controller CRUD**

Setelah membuat model, selanjutnya controller yang mengatur jalannya proses tampilan dan mem-passing nilai – nilai yang nantinya dimasukkan user untuk dimasukkan kembali ke model. Untuk itu, mari kita uat sebuah controller dengan nama **Crud.php**, pada file controller ini akan menampilkan file view dengan nama **view\_crud.php**, scripnya sebagai berikut :

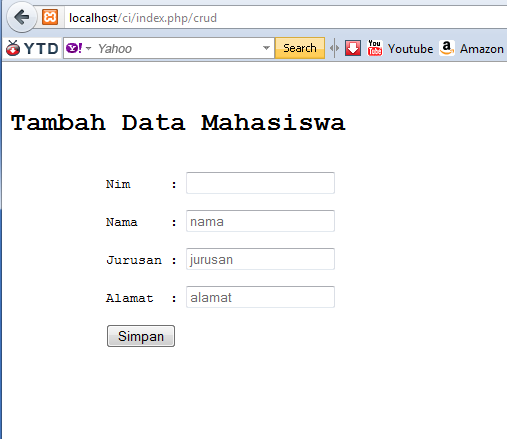
**Script Crud.php**

1. **<?php**
2. **Class Crud extends ci\_controller**
3. **{**
4. **Function \_\_construrct()**
5. **{**
6. **Parent::\_\_construct();**
7. **$this->load->model(‘model\_crud’);**
8. **}**
9. **Function index()**
10. **{**
11. **$this->load->view(‘view\_crud’,$data);**
12. **}**
13. **}**
14. **View CRUD**

Kemudian kita buat halaman view dengan nama view\_crud.php , sekarang untuk pertama kali kita akan membuat sebuah form untuk memasukkan data baru pada table mahasiswa ( Insert data ). Perhatikan kode pada file view berikut :

**Script view\_crud.php**

1. <!DOCTYPE HTML>
2. <html>
3. <head>
4. <title>Aplikasi CodeIgniter Pertama</title>
5. </head>
6. <body>
7. <?php echo form\_open('mahasiswa/tambah');?>
8. <pre>
9. <h1>Tambah Data Mahasiswa</h1>
10. Nim : <input type="text" name="nim" placeholder="nim" required autofocus><br>
11. Nama : <input type="text" name="nama" placeholder="nama" required><br>
12. Jurusan : <input type="text" name="jurusan" placeholder="jurusan" required><br>
13. Alamat : <input type="text" name="alamat" placeholder="alamat" required><br>
14. <input type="submit" value="Simpan">
15. </pre>
16. <?php form\_close();?>
17. </body>
18. </html>



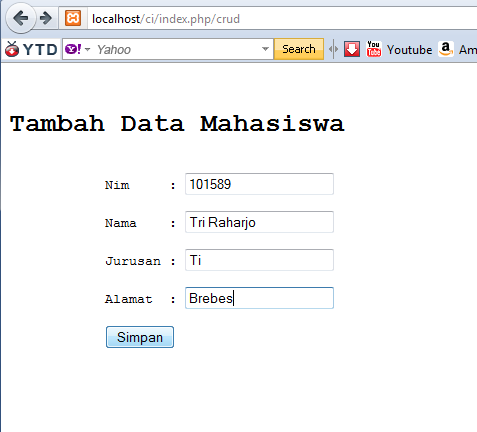
1. **Operas Create ( Menambah Data )**

Kemudian karena kita belum membuat fungsi tambah, maka data yang dimasukkan oleh user tidak akan bisa masuk pada database. Oleh karena itu, kita akan membuat fungsi ini pada Controller. Buka lagi Crud.php, kemudian lengkapi kodenya menjadi sebagai berikut :

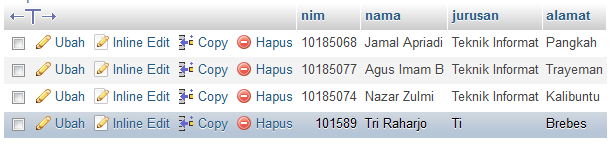
**Skrip Crud.php ( fungsi tambah )**

1. <?php
2. class Model\_crud extends ci\_controller
3. {
4. function \_\_construct()
5. {
6. parent::\_\_construct();
7. $this->load->model('model\_crud');
8. }
9. function index()
10. {
11. $this->load->view('view\_crud',$data);
12. }
13. **function tambah()**
14. **{**
15. **$data=array(**
16. **‘nim'=>$this->input->post('nim'),**
17. **'nama'=>$this->input->post('nama'),**
18. **'jurusan'=>$this->input->post('jurusan'),**
19. **'alamat'=>$this->input->post('alamat')**
20. **);**
21. **$this->model\_crud->tambah($data);**
22. **redirect('crud');**
23. **}**
24. **}**

Sekarang coba kita buka web browser kita, masukkan alamat <http://localhost/ci/index.php/crud>



Ketika tombol simpan maka data yang telah kita input telah masuk kedalam database, sementara untuk mengeceknya bisa melalui phpmyadmin.



1. **Operasi Read ( Menampilkan Data )**

Selanjutnya, kita akan menampilkan data pada database ( Read ). Buka lagi file controller **Crud.php**. kita akan menambahkan kode untuk membuat variable baru untuk data yang nantinya akan kita ambil pada database, sehingga nantinya akan mudah digunakan oleh file view.

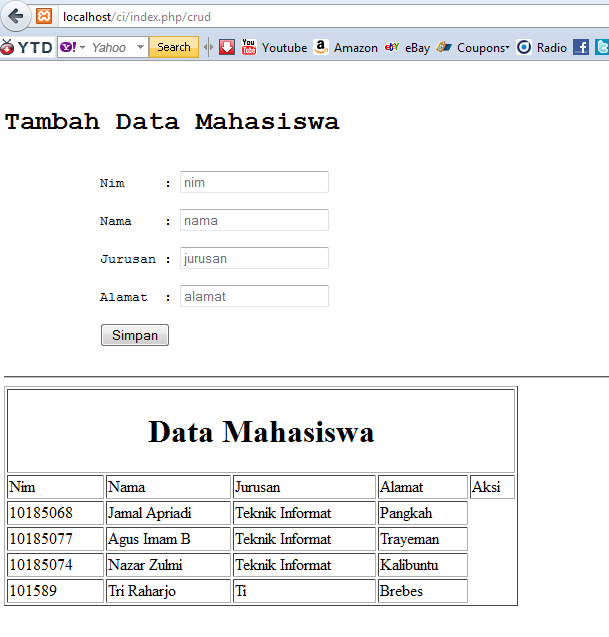
1. <?php
2. class Model\_crud extends ci\_controller
3. {
4. function \_\_construct()
5. {
6. parent::\_\_construct();
7. $this->load->model('model\_crud');
8. }
9. **function index()**
10. {
11. **$data['data']=$this->model\_crud->tampilData();**
12. **$this->load->view('view\_crud',$data);**
13. }
14. // kode selanjutnya

Kemudian buka kembali file **view\_crud.php**, kita akan menambahkan kode untuk menampilkan data – data dan array rowrecord, letakkan kodenya setelah ***<?php form\_close(); ?>***

**Skrip view\_crud.php ( untuk membaca data )**

1. ***//kode sebelumnya***
2. <?php form\_close();?>
3. <hr>
4. <table width="40%" border="1">
5. <tr>
6. <td colspan="6"><h1>Data Mahasiswa</h1></1></td>
7. </tr>
8. <tr>
9. <td>Nim</td>
10. <td>Nama</td>
11. <td>Jurusan</td>
12. <td>Alamat</td>
13. <td colspan=”2”>Aksi</td>
14. </tr>
15. <tr>
16. <?php foreach ($data as $row):?>
17. <td><?php echo $row->nim;?></td>
18. <td><?php echo $row->nama;?></td>
19. <td><?php echo $row->jurusan;?></td>
20. <td><?php echo $row->alamat;?></td>
21. </tr>
22. <?php endforeach;?>
23. </table>
24. </body>
25. </html>

Sekarang coba jalankan lagi pada browser dengan mengetikkan <http://localhost/ci/index.php/crud> , maka akan tampil data – data dari database



1. **Operasi Update ( Mengubah Data )**

Selanjutnya kita akan mencoba membuat file operasi untuk mengubah data yang ada di table mahasiswa Buka lagi file controller **Crud.php**. kita akan menambahkan kode untuk menangkap url data yang akan kita ubah.

Script Crud.php

**// kode sebelumnya**

1. **function edit()**
2. **{**
3. **$nim=$this->uri->segment(3);**
4. **$data['data']=$this->model\_crud->per\_nim($nim);**
5. **$this->load->view('update\_crud',$data);**
6. **}**

Pada baris ke 5 dijelaskan bahwa **$this->load->view(‘update\_crud’,$data);** yang artinya file view yang akan kita gunakan adalah Update\_crud.php, maka nantinya kita harus membuat file pada folder application/view dengan nama **Update\_crud.php.**

Sebelum itu buka lagi file view kita, kita akan menambahkan sebuah link aksi untuk edit kita, buka file **Crud\_view.php** kemudian tambahkan source code berikut :

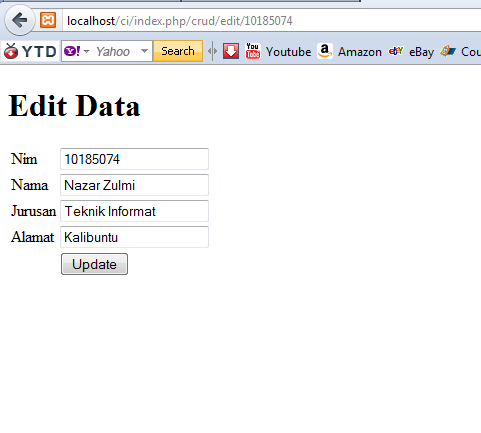
1. <tr>
2. <?php foreach ($data as $row):?>
3. <td><?php echo $row->nim;?></td>
4. <td><?php echo $row->nama;?></td>
5. <td><?php echo $row->jurusan;?></td>
6. <td><?php echo $row->alamat;?></td>
7. **<td><a href= ‘<?php base\_url();?>crud/edit/<?php echo $row->nim;?>’>Edit </a></td>**
8. **<td><a href=’<?php base\_url();?>crud/hapus/<?php echo $row->nim;?>’>Hapus</a></td>**
9. </tr>
10. <?php endforeach;?>
11. </table>
12. </body>
13. </html>

Setelah kita membuat link di file **crud\_view.php**, kita buat lagi sebuah file dengan nama **update\_crud.php** dan simpan di folder application/view, file ini nantinya digunakan sebagai form untuk mengubah data kita.

**Script Update\_crud.php**

1. **<!DOCTYPE HTML>**
2. **<html>**
3. **<head>**
4. **<title>Updata Data</title>**
5. **</head>**
6. **<body>**
7. **<?php foreach ($data as $row):?>**
8. **<?php echo form\_open('crud/update');?>**
9. **<h1>Edit Data</h1>**
10. **<table>**
11. **<tr>**
12. **<td>Nim</td>**
13. **<td><input type="text" name="nim" value="<?php echo $row->nim;?>"></td>**
14. **</tr>**
15. **<tr>**
16. **<td>Nama</td>**
17. **<td><input type="text" name="nama" value="<?php echo $row->nama;?>"></td>**
18. **</tr>**
19. **<tr>**
20. **<td>Jurusan</td>**
21. **<td><input type="text" name="jurusan" value="<?php echo $row->jurusan;?>"></td>**
22. **</tr>**
23. **<tr>**
24. **<td>Alamat</td>**
25. **<td><input type="text" name="alamat" value="<?php echo $row->alamat;?>"></td>**
26. **</tr>**
27. **<tr>**
28. **<td></td>**
29. **<td><input type="Submit" value="Update"></td>**
30. **</tr>**
31. **</table>**
32. **<?php endforeach;?>**
33. **<?php form\_close();?>**
34. **</body>**
35. **</html>**

Sekarang, coba kita klik salah satu tombol edit pada table crud\_view.php, kemudian akan muncul tampilan seperti berikut :



Tentuna ketika tombol Update di klik kita belum melakukan suatu operasi apapun, karena nya coba kita perhatikan file update\_crud.php dan perhatikan pada baris ke 8 **<?php echo form\_open('crud/update');?>** yang artinya bahwa file pada form ini akan di kirim ke fungsi Update pada controller **crud.php**, maka dari itu kita buka lagi file controller kita **crud.php** dan kita tambahkan source code berikut :

**Script crud.php**

**//kode sebelumnya**

1. **function update()**
2. **{**
3. **$nim=$this->input->post('nim');**
4. **$data=array(**
5. **'nama'=>$this->input->post('nama'),**
6. **'jurusan'=>$this->input->post('jurusan'),**
7. **'alamat'=>$this->input->post('alamat')**
8. **);**
9. **$this->model\_crud->update($nim,$data);**
10. **redirect ('crud');**
11. **}**
12. **Operasi Delete ( Menghapus Data )**

Pada operasi yang terakhir kita akan menambahkan fungsi hapus pada file controller **crud.php**, adapun kodenya adalah sebagai berikut :

**Script Crud.php**

**//kode selanjutnya**

1. **function hapus()**
2. **{**
3. **$nim=$this->uri->segment(3);**
4. **$this->model\_crud->hapus($nim);**
5. **redirect ('crud');**
6. **}**

**Full source Code Crud.php**

1. **<?php**
2. **class Model\_crud extends ci\_controller**
3. **{**
4. **function \_\_construct()**
5. **{**
6. **parent::\_\_construct();**
7. **$this->load->model('model\_crud');**
8. **}**
9. **function index()**
10. **{**
11. **$data['data']=$this->model\_crud->tampilData();**
12. **$this->load->view('view\_crud',$data);**
13. **}**
14. **function tambah()**
15. **{**
16. **$data=array(**
17. **‘nim'=>$this->input->post('nim'),**
18. **'nama'=>$this->input->post('nama'),**
19. **'jurusan'=>$this->input->post('jurusan'),**
20. **'alamat'=>$this->input->post('alamat')**
21. **);**
22. **$this->model\_crud->tambah($data);**
23. **redirect('crud');**
24. **}**
25. **function edit()**
26. **{**
27. **$nim=$this->uri->segment(3);**
28. **$data['data']=$this->model\_crud->per\_id($nim);**
29. **$this->load->view('update\_crud',$data);**
30. **}**
31. **function update()**
32. **{**
33. **$nim=$this->input->post('nim');**
34. **$data=array(**
35. **'nama'=>$this->input->post('nama'),**
36. **'jurusan'=>$this->input->post('jurusan'),**
37. **'alamat'=>$this->input->post('alamat')**
38. **);**
39. **$this->model\_crud->update($nim,$data);**
40. **redirect ('crud');**
41. **}**
42. **function hapus()**
43. **{**
44. **$nim=$this->uri->segment(3);**
45. **$this->model\_crud->hapus($nim);**
46. **redirect ('crud');**
47. **}**
48. **}**

**Penutup**

Demikian tutorial singkat ini, mohon maaf jika ada salah kata dalam penulisan dan semoga artikel ini bermanfaat.

**Referensi**

* Codeigniter user guide : <http://codeigniter.com/user_guide/>