

Aplikasi CRUD (Create Read Update Delete) dengan Codeigniter

Jamal Apriadi

Ghantexs16@gmail.com

<http://commit-tegal.org/>

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Didalam pembuatan suatu aplikasi dengan menggunakan bahasa pemrograman apapun maka hal dasar yang harus kita ketahui adalah CRUD (Create Read Update Delete), sebelum melanjutkan membaca artikel ini pastikan anda sudah mengenal HTML, MySql, PHP (Konsep OOP di PHP) terlebih dahulu dan anda sudah paham apa itu Framework PHP. . . :D

Pendahuluan

Codeigniter adalah framework PHP yang sangat powerful, dibangun dengan PHP dan ditujukan bagi developer yang membutuhkan toolkit sederhana dan elegan untuk membuat fitur website yang komplit (www.codeigniter.com).

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.

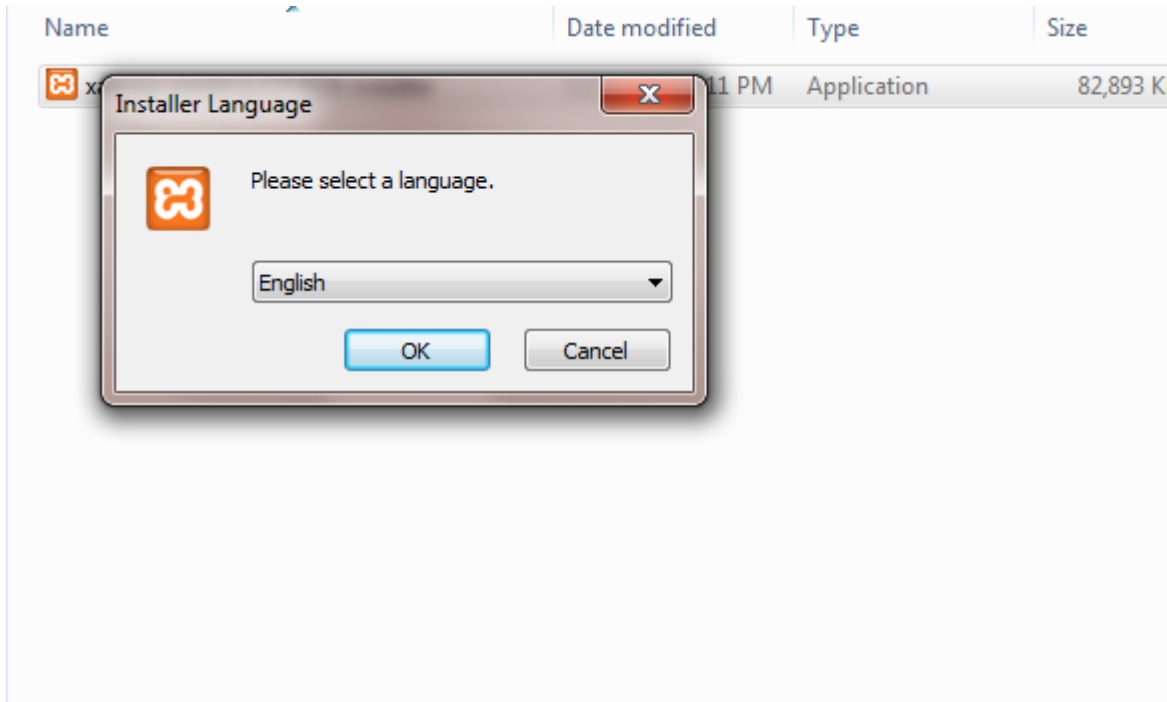
A. Kebutuhan

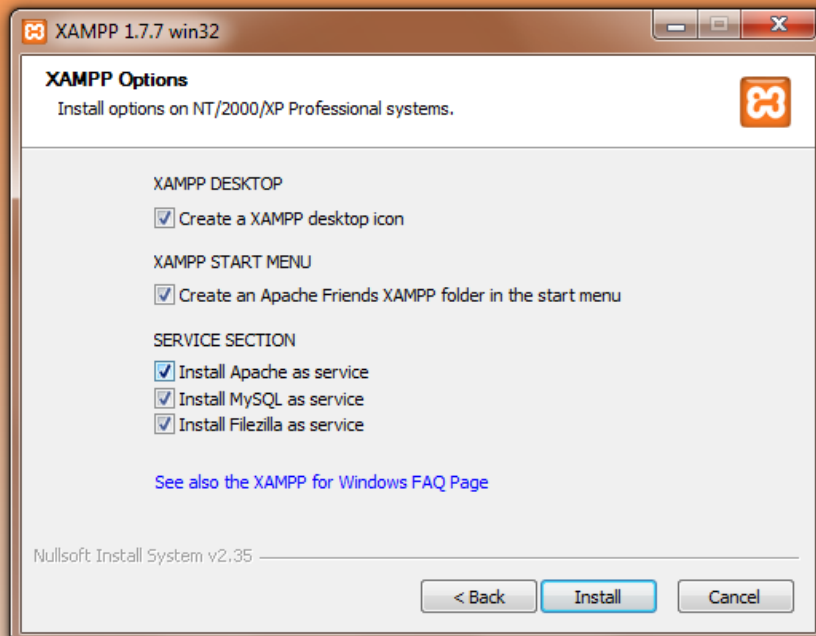
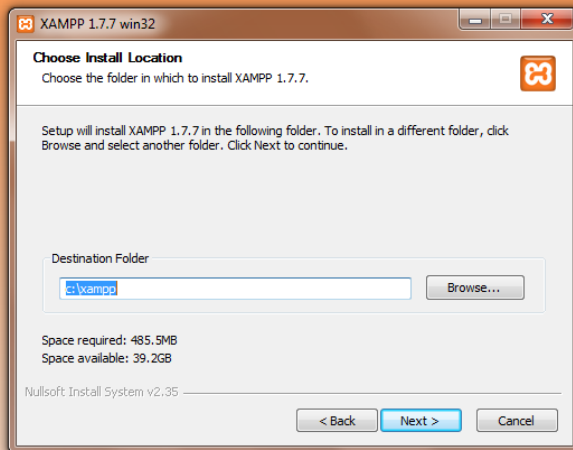
Berikut ini adalah kebutuhan yang diperlukan untuk menggunakan framework Codeigniter :

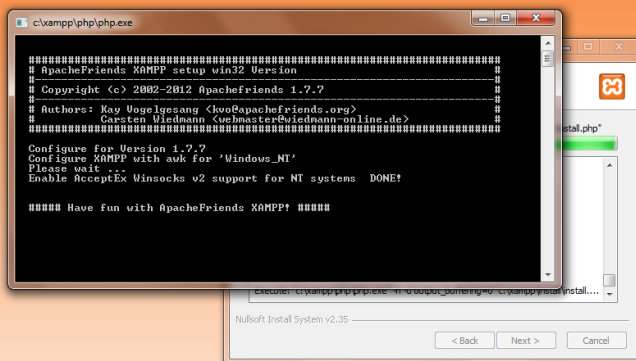
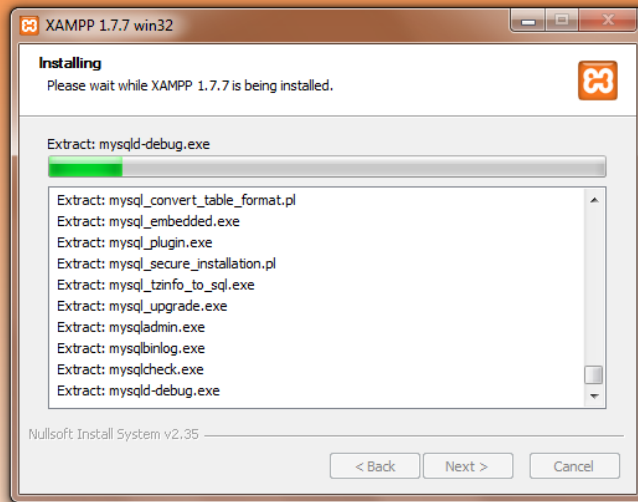
- 1) XAMPP (Apache, MySQL, PHP)
- 2) Framework Codeigniter 2.1.0
- 3) PHP Editor (Notepad ++, PHP Designer, Dreamweaver, dsb)

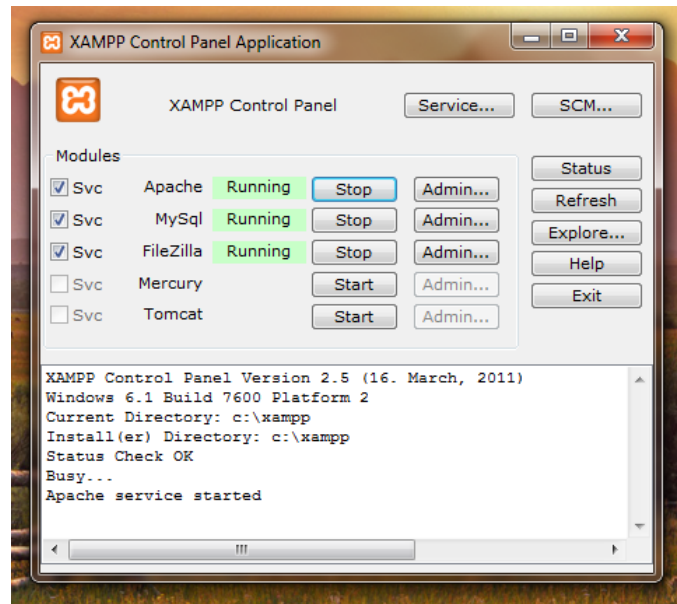
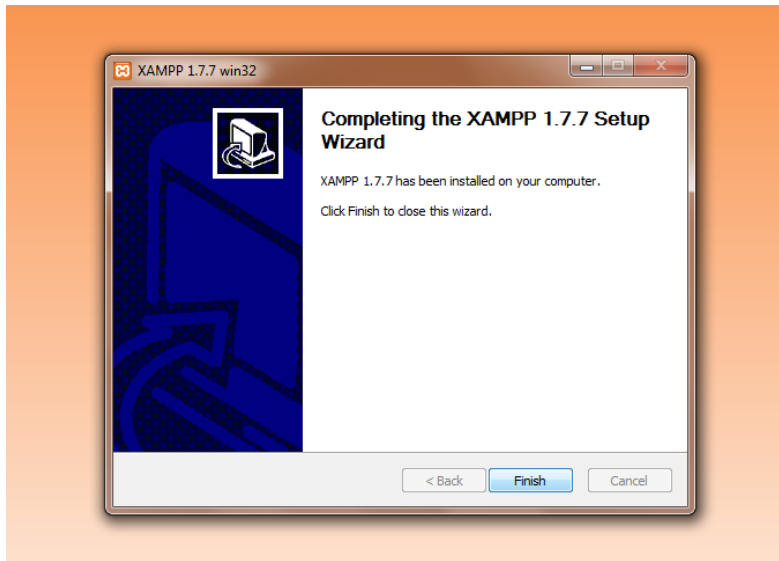
B. Instalasi Xampp

Download xampp di alamat : <http://www.apachefriends.org>







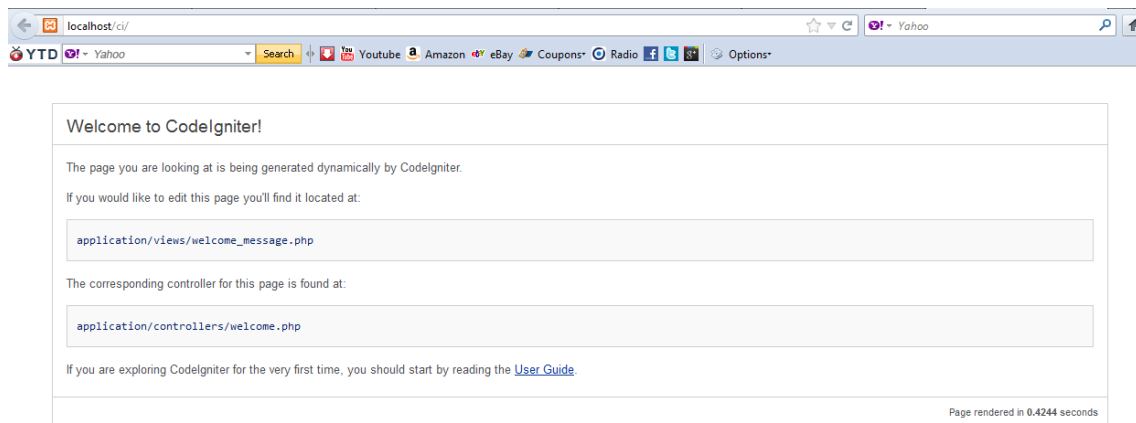


C. Setting Framework Codeigniter

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

- a. Setelah Xampp terinstal dengan baik, Ekstrak file CodeIgniter_2.1.0.zip ke direktori xampp/htdocs dan remane menjadi Ci agar lebih memudahkan kita.
- b. 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/index.php/mahasiswa/>



Hello CodeIgniter

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{  
}
```

2. 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>
```

3. 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.
9.     Function index()
10.    {
11.        $this->load->view('hello_codeigniter');
12.    }
13.
14.    Function fungsibaru()
15.    {
16.        $this->load->view('view_baru');
17.    }
18. }
```

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>



4. Membuat file Model

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

- Pada file autoload.php, cari kode baris 55

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

Ubah menjadi seperti berikut

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

Dan juga pada baris 67

```
$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.

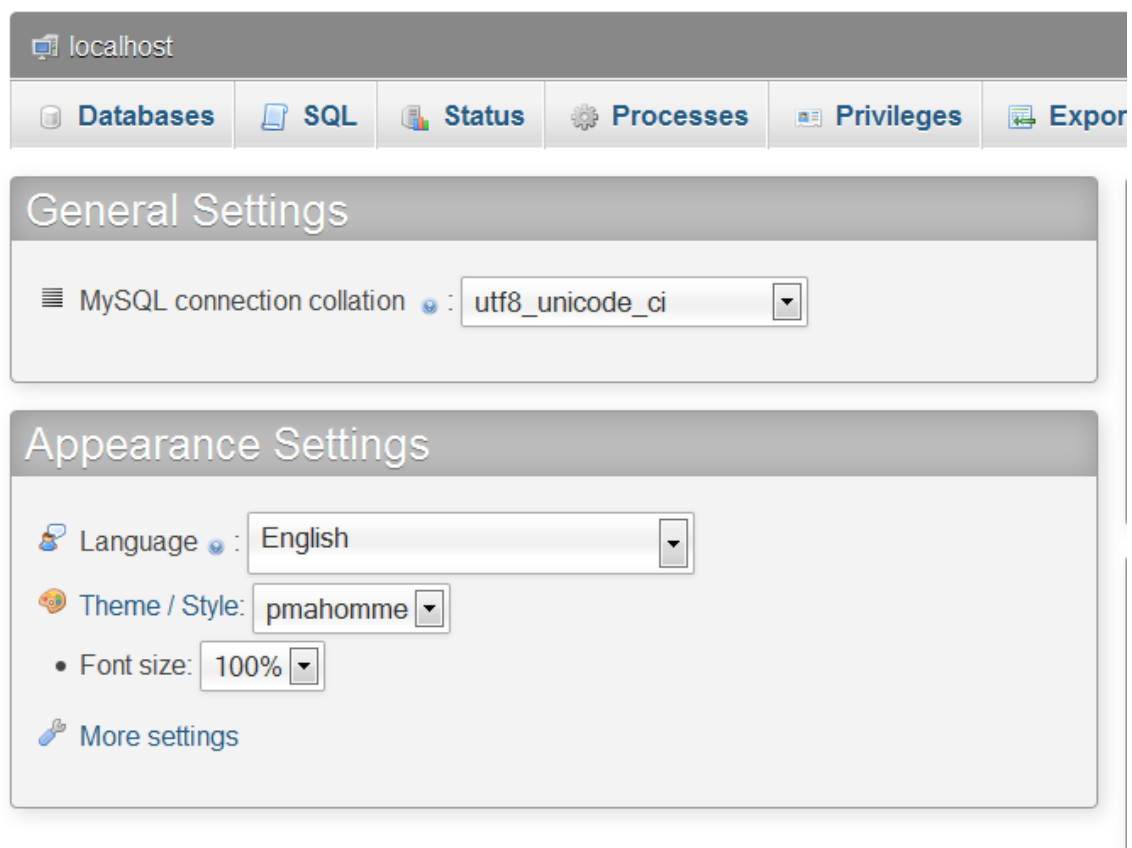
- Pada file database.php

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

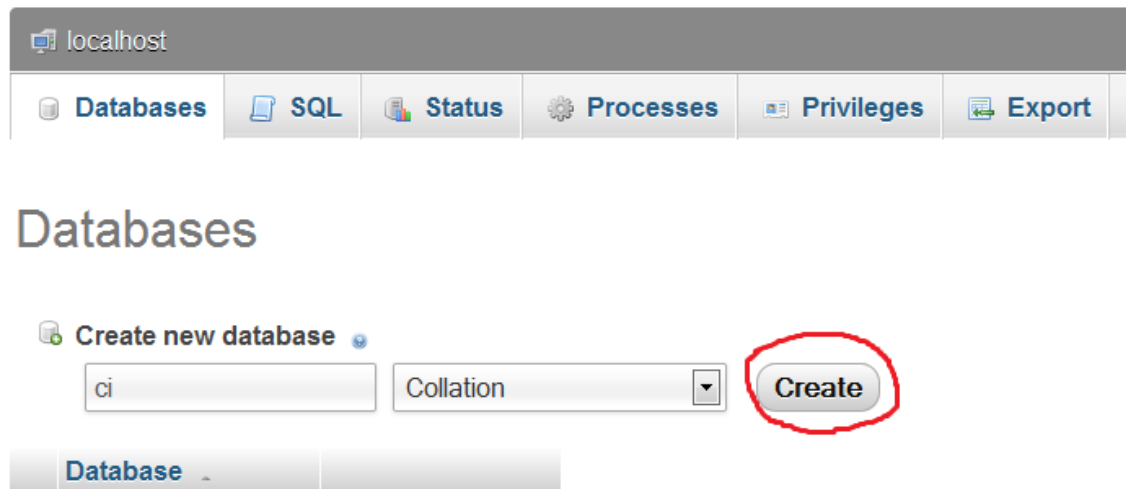
```
$db['default']['hostname'] = 'localhost';  
$db['default']['username'] = ""; // sesuaikan dengan webserver anda  
$db['default']['password'] = "";  
$db['default']['database'] = ""; // nama database yang telah dibuat  
$db['default']['dbdriver'] = 'mysql';
```

```
$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';
```

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



localhost

Databases SQL Status Processes Privileges Export

Databases

Create new database

ci Collation

Create

Database

Setelah buat database, buatlah sebuah table dengan nama mahasiswa

<input type="checkbox"/>	1	<u>nim</u>	varchar(10)	latin1_swedish_ci
<input type="checkbox"/>	2	nama	varchar(20)	latin1_swedish_ci
<input type="checkbox"/>	3	jurusan	varchar(15)	latin1_swedish_ci
<input type="checkbox"/>	4	alamat	varchar(30)	latin1_swedish_ci

Kemudian sisipkan beberapa record seperti berikut :

nim	nama	jurusan	alamat
10185068	Jamal Apriadi	Teknik Informat	Pangkah
10185074	Nazar Zulmi	Teknik Informat	Kalibuntu
10185077	Agus Imam B	Teknik Informat	Trayeman

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 deklarasikan pada file model.hapus beberapa baris kode yang telah kita buat sebelumnya sehingga skrip mahasiswa.php akan seperti berikut :

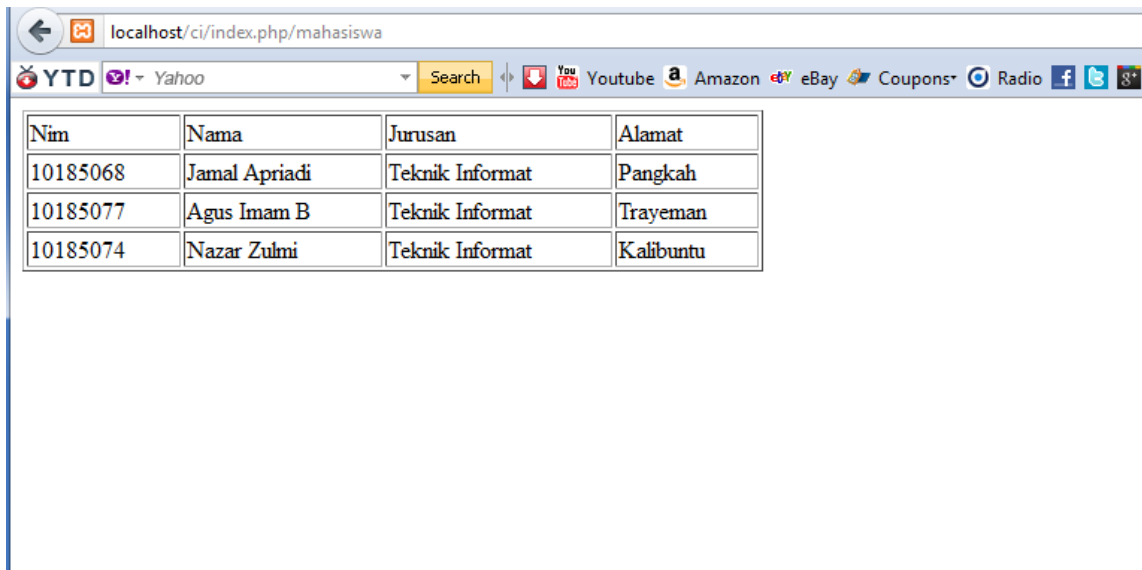
```
1.  <?php
2.  class Mahasiswa extends ci_controller
3.  {
4.      function __construct()
5.      {
6.          parent::__construct();
7.          $this->load->model('model_mahasiswa');
8.          // adalah sebagai konstruktor berbagai perintah yang ada di
9.          // dalam fungsi ini akan selalu 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>



The screenshot shows a web browser window with the address bar displaying 'localhost/ci/index.php/mahasiswa'. The browser's toolbar includes a search bar with 'YTD' and 'Yahoo' suggestions, and various social media and service icons like YouTube, Amazon, eBay, and Radio. Below the browser window, a table displays student information with four columns: Nim, Nama, Jurusan, and Alamat. The table contains three rows of data.

Nim	Nama	Jurusan	Alamat
10185068	Jamal Apriadi	Teknik Informat	Pangkah
10185077	Agus Imam B	Teknik Informat	Trayeman
10185074	Nazar Zulmi	Teknik Informat	Kalibuntu

Bagian III

Operasi CRUD

(CREATE, READ, UPDATE, DELETE)

CRUD merupakan singkatan dari **Create**, **Read**, **Update** dan **Delete**. 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. }
```

2. 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 buat 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 __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. }
```

3. 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>
```

localhost/ci/index.php/crud

YTD Yahoo Search Youtube Amazon

Tambah Data Mahasiswa

Nim :

Nama :

Jurusan :

Alamat :

a. 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>

Tambah Data Mahasiswa

Nim : 101589

Nama : Tri Raharjo

Jurusan : Ti

Alamat : Brebes

Simpan

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

					nim	nama	jurusan	alamat
	Ubah	Inline Edit	Copy	Hapus	10185068	Jamal Apriadi	Teknik Informat	Pangkah
	Ubah	Inline Edit	Copy	Hapus	10185077	Agus Imam B	Teknik Informat	Trayeman
	Ubah	Inline Edit	Copy	Hapus	10185074	Nazar Zulmi	Teknik Informat	Kalibuntu
	Ubah	Inline Edit	Copy	Hapus	101589	Tri Raharjo	Ti	Brebes

b. 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
```

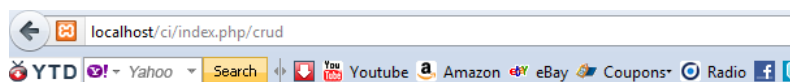
Kemudian buka kembali file **view_crud.php**, kita akan menambahkan kode untuk menampilkan data – data dan array rowrecord, letakkan kodenya setelah **<?php form_close(); ?>**

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></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



Tambah Data Mahasiswa

Nim :

Nama :

Jurusan :

Alamat :

Data Mahasiswa				
Nim	Nama	Jurusan	Alamat	Aksi
10185068	Jamal Apriadi	Teknik Informat	Pangkajene	
10185077	Agus Imam B	Teknik Informat	Trayem	
10185074	Nazar Zulmi	Teknik Informat	Kalibuntu	
101589	Tri Raharjo	Ti	Brebes	

c. 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 </td>**

8. <td><a href='<?php base_url();?>crud/hapus/<?php echo \$row->nim;?>'>Hapus</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>Udata 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 :



Edit Data

Nim	<input type="text" value="10185074"/>
Nama	<input type="text" value="Nazar Zulmi"/>
Jurusan	<input type="text" value="Teknik Informat"/>
Alamat	<input type="text" value="Kalibuntu"/>
<input type="button" value="Update"/>	

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. }
```

d. 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/

Biografi Penulis



Jamal Apriadi. Mahasiswa program studi Teknik Informatika di STMIK TEGAL, saat ini aktif di UKM Communitas Mahasiswa IT Tegal dan sedang mendalami beberapa bidang IT diantaranya Pemrograman Web, Java, Basis Data dan Jaringan Komputer.

Anda dapat menghubungi penulis lewat E – Mail :
Ghantexs16@gmail.com