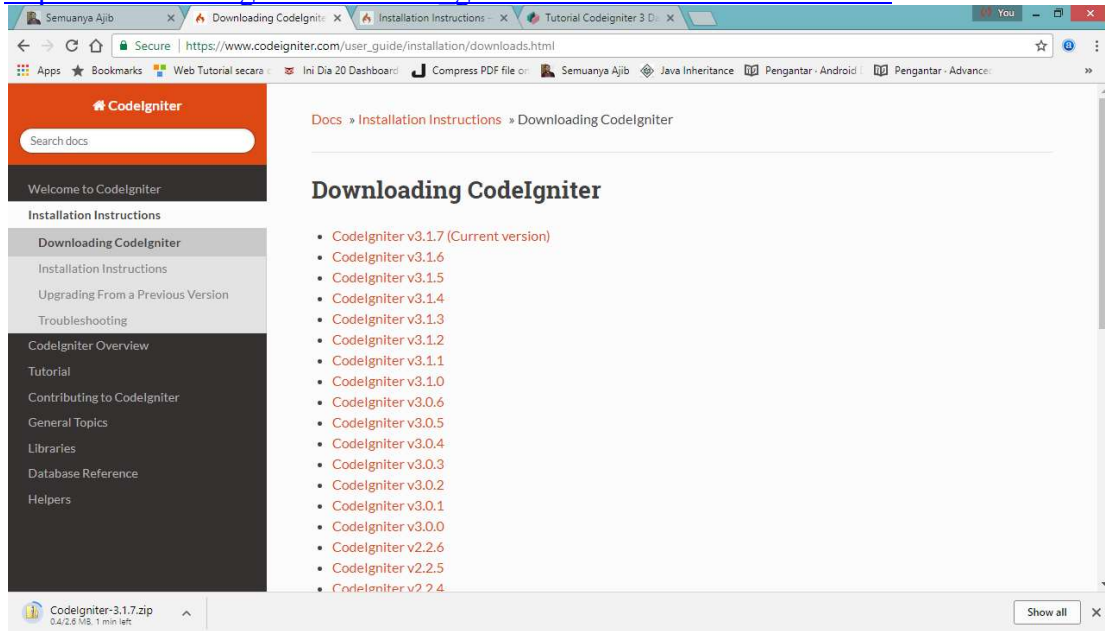


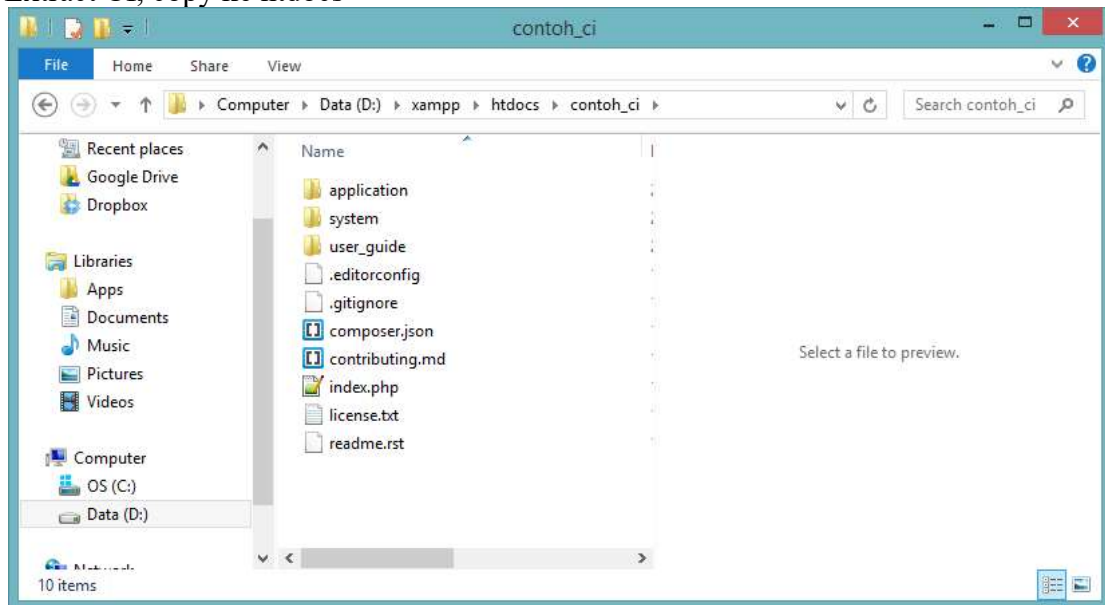
Tutorial CI CRUD with MySQL

Download CI

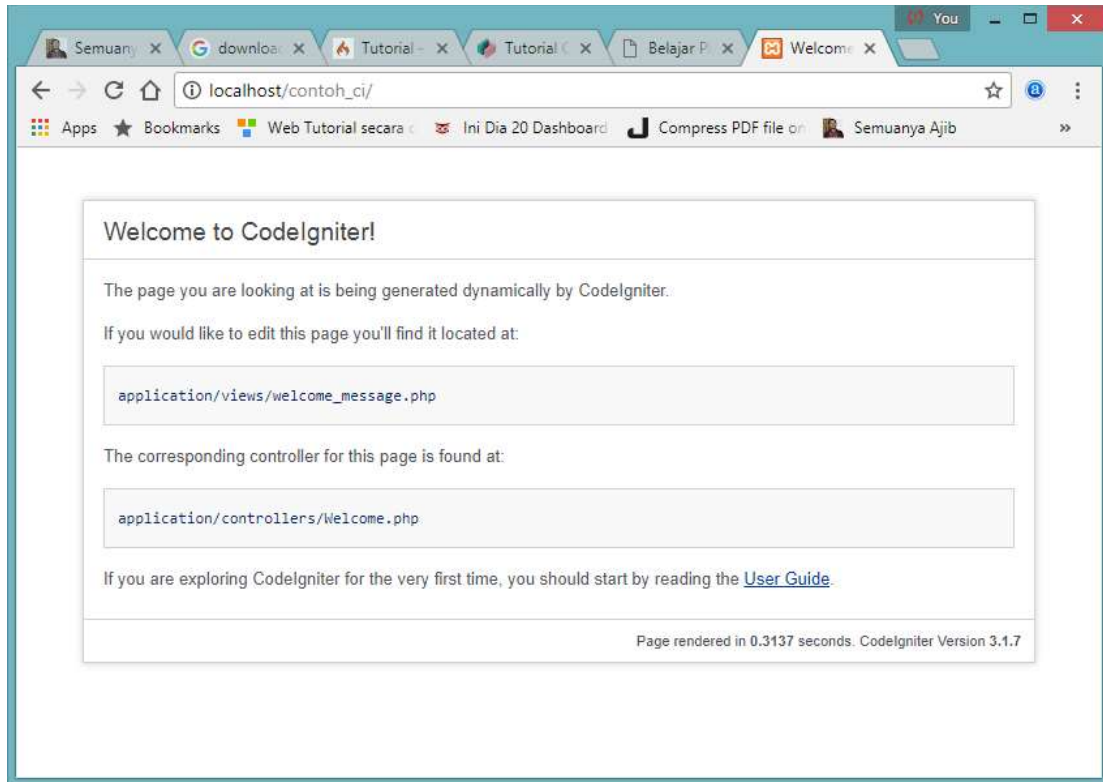
https://www.codeigniter.com/user_guide/installation/downloads.html



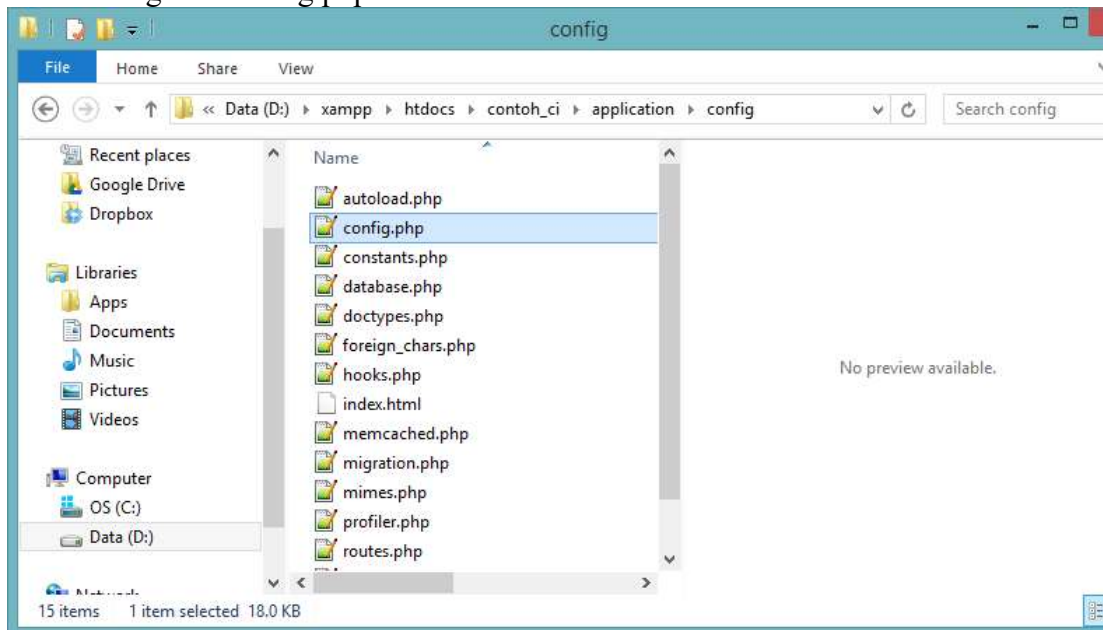
Extract CI, copy ke htdocs

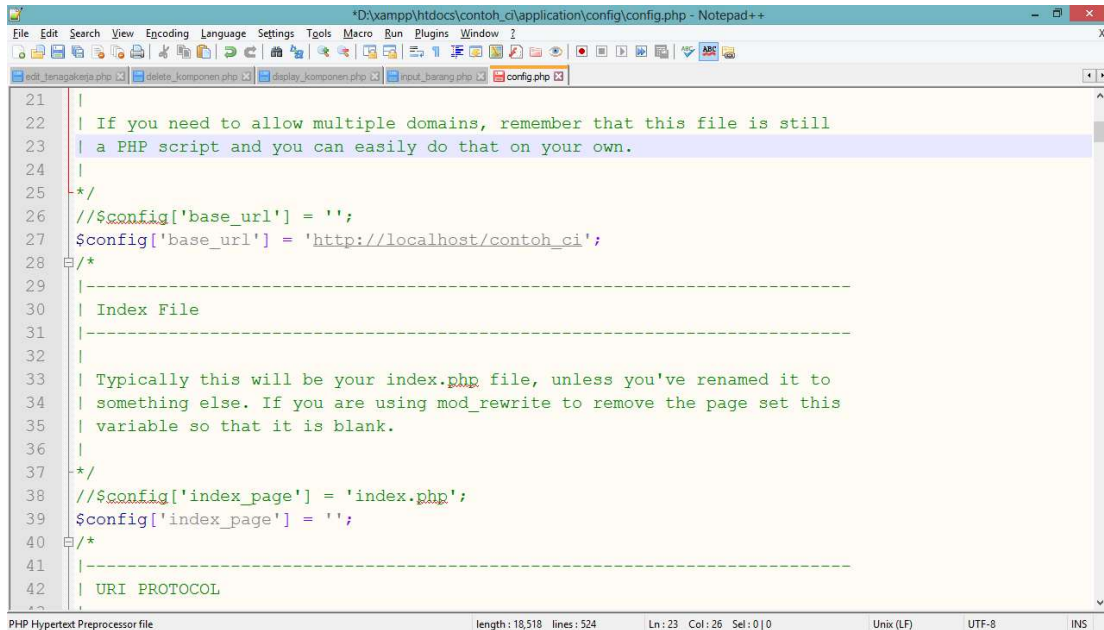


Tes CI di localhost



Ubah konfigurasi config.php



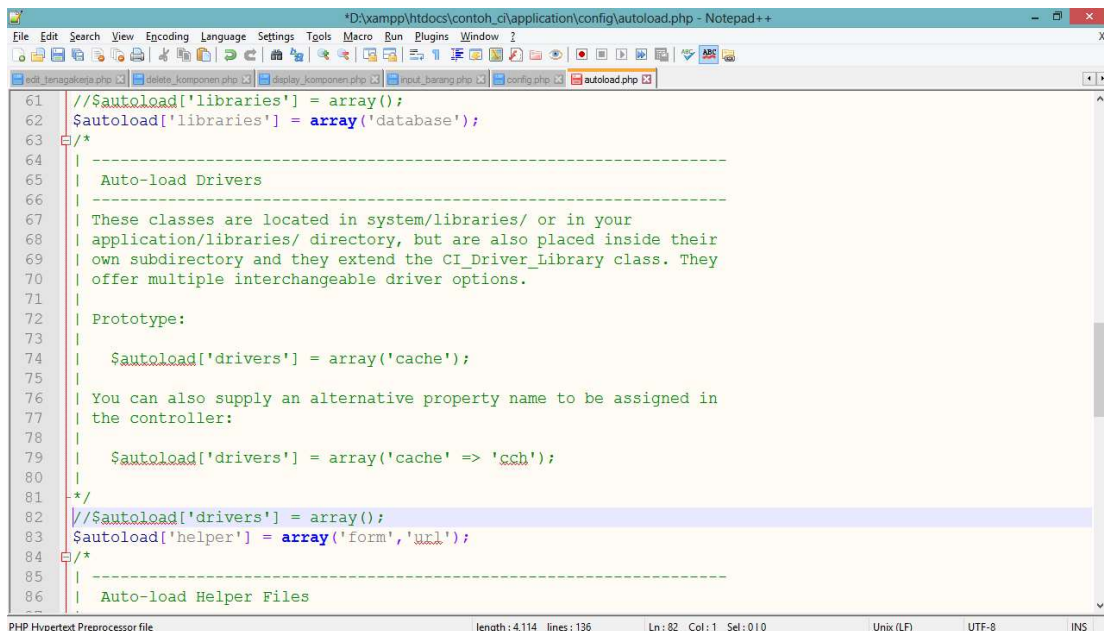


```
21 |
22 | If you need to allow multiple domains, remember that this file is still
23 | a PHP script and you can easily do that on your own.
24 |
25 | */
26 | // $config['base_url'] = '';
27 | $config['base_url'] = 'http://localhost/contoh_ci';
28 | /*
29 | -----
30 | | Index File
31 | -----
32 |
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 |
37 | */
38 | // $config['index_page'] = 'index.php';
39 | $config['index_page'] = '';
40 | /*
41 | -----
42 | | URI PROTOCOL
```

`$config['base_url'] = 'http://localhost/contoh_ci';`

`$config['index_page'] = '';`

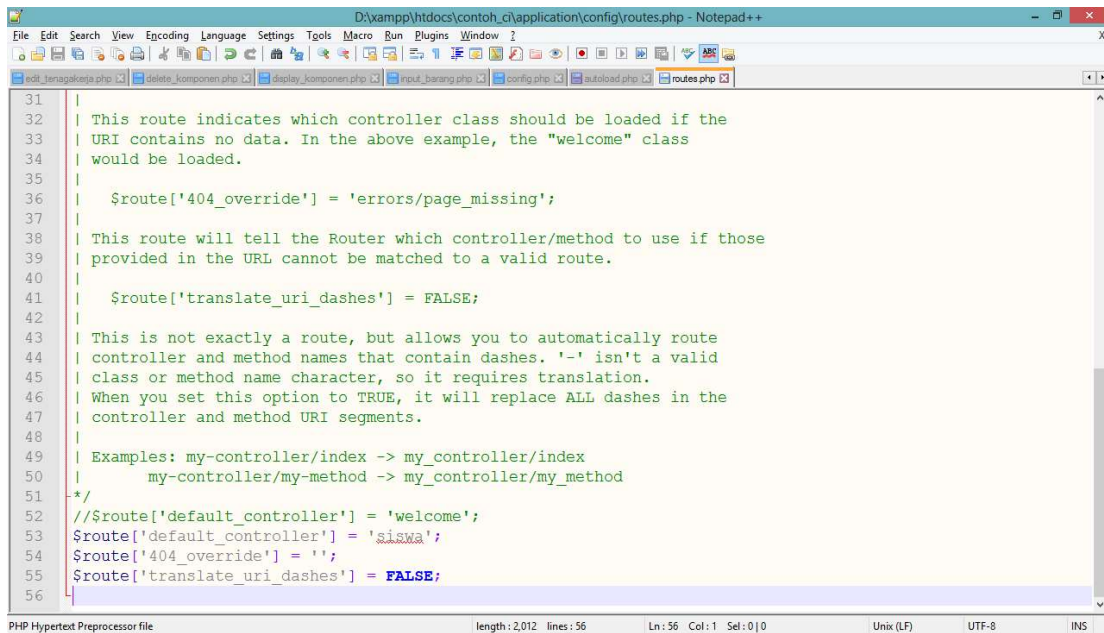
Ubah file autoload.php



```
61 | // $autoload['libraries'] = array();
62 | $autoload['libraries'] = array('database');
63 | /*
64 | -----
65 | | Auto-load Drivers
66 | -----
67 | These classes are located in system/libraries/ or in your
68 | application/libraries/ directory, but are also placed inside their
69 | own subdirectory and they extend the CI_Driver_Library class. They
70 | offer multiple interchangeable driver options.
71 |
72 | Prototype:
73 |
74 | $autoload['drivers'] = array('cache');
75 |
76 | You can also supply an alternative property name to be assigned in
77 | the controller:
78 |
79 | $autoload['drivers'] = array('cache' => 'cch');
80 |
81 | */
82 | // $autoload['drivers'] = array();
83 | $autoload['helper'] = array('form', 'url');
84 | /*
85 | -----
86 | | Auto-load Helper Files
```

1. `$autoload['libraries'] = array('database');`
2. `$autoload['helper'] = array('form','url');`

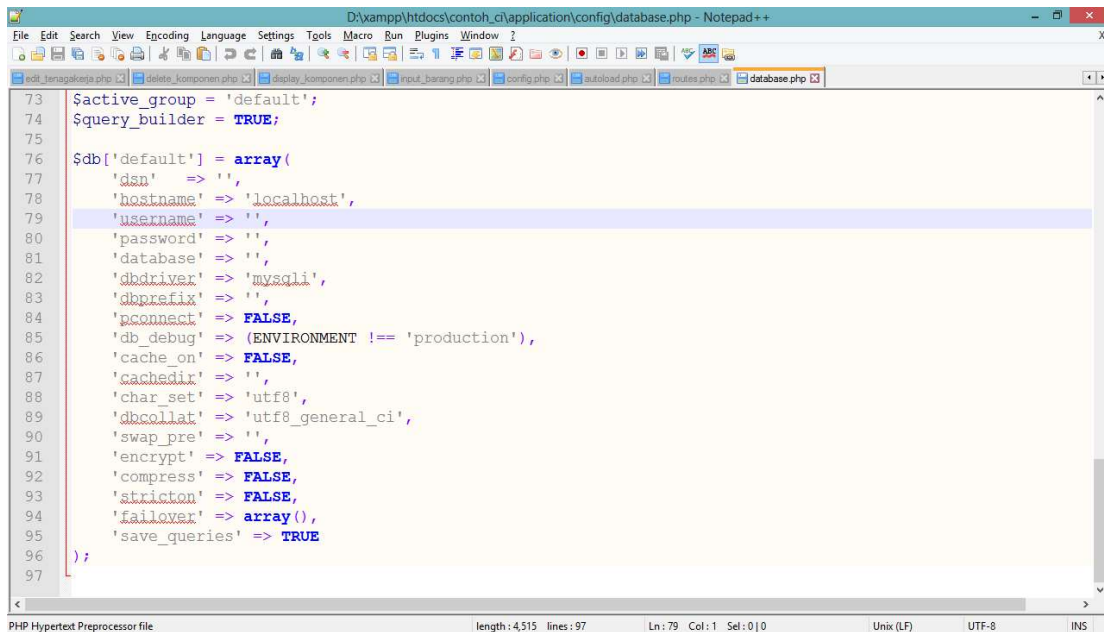
Buka router.php



```
31 |
32 | This route indicates which controller class should be loaded if the
33 | URI contains no data. In the above example, the "welcome" class
34 | would be loaded.
35 |
36 | $route['404_override'] = 'errors/page_missing';
37 |
38 | This route will tell the Router which controller/method to use if those
39 | provided in the URL cannot be matched to a valid route.
40 |
41 | $route['translate_uri_dashes'] = FALSE;
42 |
43 | This is not exactly a route, but allows you to automatically route
44 | controller and method names that contain dashes. '-' isn't a valid
45 | class or method name character, so it requires translation.
46 | When you set this option to TRUE, it will replace ALL dashes in the
47 | controller and method URI segments.
48 |
49 | Examples: my-controller/index -> my_controller/index
50 |           my-controller/my-method -> my_controller/my_method
51 |
52 | //$route['default_controller'] = 'welcome';
53 | $route['default_controller'] = 'siswa';
54 | $route['404_override'] = '';
55 | $route['translate_uri_dashes'] = FALSE;
56 |
```

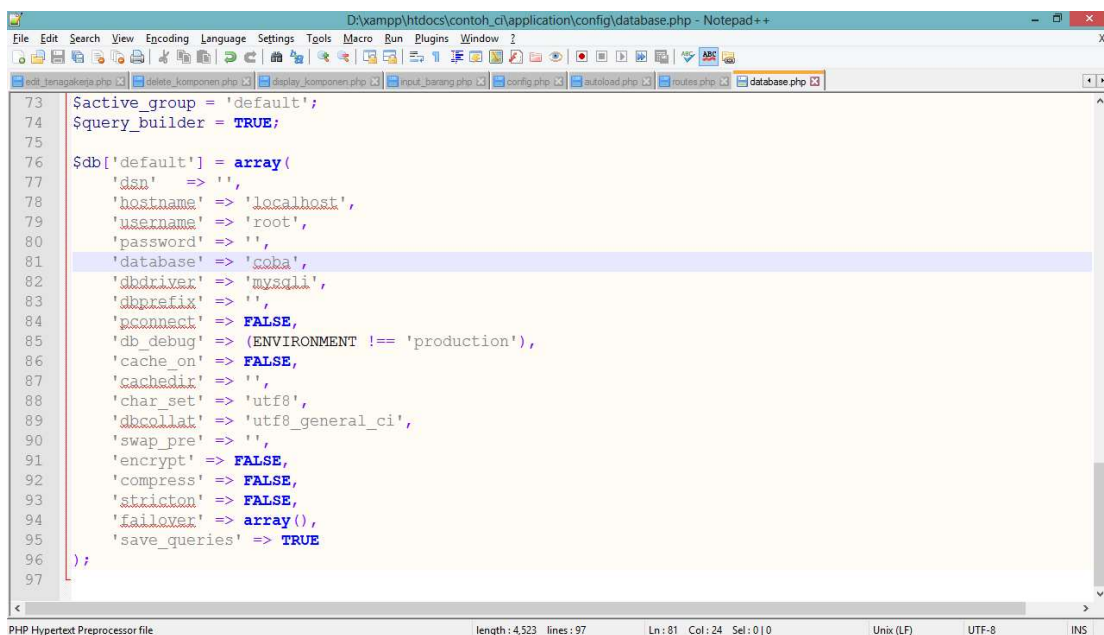
```
// $route['default_controller'] = 'welcome';
$route['default_controller'] = 'siswa';
```

Buka database.php



```
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => '',
80     'password' => '',
81     'database' => '',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97
```

Ubah menjadi



```
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'coba',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97
```

Membuat Model

Sekarang kita akan membuat modelnya. berisi sebuah fungsi untuk menampilkan semua data siswa pada tabel **siswa**. Buat sebuah file dengan

nama **SiswaModel.php**, Lalu simpan pada folder **xampp/htdocs/crud_ci/application/models/**. Berikut kodenya :

```
1. <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2.
3. class SiswaModel extends CI_Model {
4. // Fungsi untuk menampilkan semua data siswa
5. public function view(){
6. return $this->db->get('siswa')->result();
7. }
8.
9. // Fungsi untuk menampilkan data siswa berdasarkan NIS nya
10. public function view_by($nis){
11. $this->db->where('nis', $nis);
12. return $this->db->get('siswa')->row();
13. }
14.
15. // Fungsi untuk validasi form tambah dan ubah
16. public function validation($mode){
17. $this->load->library('form_validation'); // Load library form_validation untuk proses validasinya
18.
19. // Tambahkan if apakah $mode save atau update
20. // Karena ketika update, NIS tidak harus divalidasi
21. // Jadi NIS di validasi hanya ketika menambah data siswa saja
22. if($mode == "save")
23. $this->form_validation->set_rules('input_nis', 'NIS', 'required|numeric|max_length[11]');
24.
25. $this->form_validation->set_rules('input_nama', 'Nama', 'required|max_length[50]');
26. $this->form_validation->set_rules('input_jeniskelamin', 'Jenis Kelamin', 'required');
27. $this->form_validation->set_rules('input_telp', 'telp', 'required|numeric|max_length[15]');
28. $this->form_validation->set_rules('input_alamat', 'Alamat', 'required');
29.
30. if($this->form_validation->run()) // Jika validasi benar
31. return TRUE; // Maka kembalikan hasilnya dengan TRUE
32. else // Jika ada data yang tidak sesuai validasi
33. return FALSE; // Maka kembalikan hasilnya dengan FALSE
34. }
35.
36. // Fungsi untuk melakukan simpan data ke tabel siswa
37. public function save(){
38. $data = array(
39. "nis" => $this->input->post('input_nis'),
40. "nama" => $this->input->post('input_nama'),
41. "jenis_kelamin" => $this->input->post('input_jeniskelamin'),
42. "telp" => $this->input->post('input_telp'),
43. "alamat" => $this->input->post('input_alamat')
44. );
45.
```



```

46. $this->db->insert('siswa', $data); // Untuk mengeksekusi perintah insert data
47. }
48.
49. // Fungsi untuk melakukan ubah data siswa berdasarkan NIS siswa
50. public function edit($nis){
51. $data = array(
52. "nama" => $this->input->post('input_nama'),
53. "jenis_kelamin" => $this->input->post('input_jeniskelamin'),
54. "telp" => $this->input->post('input_telp'),
55. "alamat" => $this->input->post('input_alamat')
56. );
57.
58. $this->db->where('nis', $nis);
59. $this->db->update('siswa', $data); // Untuk mengeksekusi perintah update data
60. }
61.
62. // Fungsi untuk melakukan menghapus data siswa berdasarkan NIS siswa
63. public function delete($nis){
64. $this->db->where('nis', $nis);
65. $this->db->delete('siswa'); // Untuk mengeksekusi perintah delete data
66. }
67. }

```

Pada kode diatas, kita membuat sebuah fungsi view(). dimana didalamnya terdapat kode return `$this->db->get('siswa')->result()`. Kode tersebut berfungsi untuk menampilkan semua data pada tabel `siswa`. Struktur dasar penulisannya seperti ini : `return $this->db->get('nama_tabel')->result()`. Lalu pada kode diatas juga ada kode `return`, kode tersebut digunakan untuk mengeluarkan hasil dari sebuah fungsi. Pada kasus diatas, hasil yang dikeluarkan oleh `return` yaitu data-data siswa.

Lalu pada fungsi `view_by($nis)`. Fungsi ini digunakan untuk menampilkan data siswa berdasarkan nis yang kita tentukan. Pada fungsi ini terdapat skrip `$this->db->where('nis', $nis)`; yang berfungsi untuk menambahkan “where clause” pada query yang kita deklarasikan. Kode ini sama saja dengan : `WHERE nis='$nis'`.

Mungkin Anda bertanya-tanya, pada kedua fungsi tersebut, ada perbedaan yakni `->result()` dan `->row()`. Perbedaannya yakni kalau `result()` digunakan apabila data yang dihasilkan lebih dari 1 data, dalam kasus ini fungsi `view()` berguna untuk menampilkan semua data siswa. Sedangkan `row()` digunakan apabila data yang dihasilkan hanya 1 data, dalam kasus ini fungsi `view_by($nis)` memang hanya menghasilkan 1 data siswa dengan nis yang sudah ditentukan.

Kemudian masuk ke fungsi `validation($mode)`. Fungsi ini mungkin sudah jelas yakni untuk memvalidasi data yang berasal dari input form. Inilah istimewanya kalau kita menggunakan sebuah framework, pasti sudah disediakan sebuah modul / librari untuk kemudahan kita para developer. Untuk validasi di Codeigniter, kita bisa menggunakan librari `form_validation`. Untuk menset validasi di setiap inputan form, gunakan kode ini : `$this->form_validation->set_rules('nama_form_input', 'label_untuk_pesan_error', 'validasi_yang_kita_inginkan')`. untuk validasi-validasi apa saja yang telah disediakan codeigniter, silahkan buka link berikut : **Referensi Validasi**. Lalu untuk penjelasan lebih detail mengenai validasi form di codeigniter, buka link berikut ini : **Dokumentasi Validasi Form**. (Ketika membuka kedua link tadi, Anda akan dibawa ke halaman iklan terlebih dahulu, Lihat di pojok kanan atas, tunggu 5 detik. lalu klik tombol SKIP AD).

```
$data = array(
    "nis" => $this->input->post('input_nis'),
    "nama" => $this->input->post('input_nama'),
    "jenis_kelamin" => $this->input->post('input_jeniskelamin'),
    "telp" => $this->input->post('input_telp'),
    "alamat" => $this->input->post('input_alamat')
);
```

Kemudian kita masuk ke fungsi `save()`. Dalam fungsi tersebut terdapat kode seperti diatas. Kita buat sebuah array untuk menampung data apa saja yang akan kita simpa ke tabel di database. Lihat pada kode yang saya beri warna merah, kode tersebut harus sama dengan nama-nama field/kolom pada tabel siswa di databasenya. Lalu array `$data` tersebut akan kita simpan ke database dengan kode ini `$this->db->insert('siswa', $data)`. Struktur dari kode ini adalah `$this->db->insert(nama_tabel, array_data)`;

Untuk fungsi `update($nis)` pun hampir sama seperti fungsi `save()`. hanya disini kita tambahkan where nis `$this->db->where('nis', $nis)`.

BUAT CONTROLLER

Pada tahap ini, kita akan membuat controllernya. Buat sebuah file dengan nama `Siswa.php`, lalu simpan pada folder `xampp/htdocs/crud_ci/application/controllers/`. Berikut ini kodenya :

```
1. <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2.
3. class Siswa extends CI_Controller {
4.
5.     public function __construct(){
6.         parent::__construct();
7.     }
```



```

8.  $this->load->model('SiswaModel'); // Load SiswaModel ke controller ini
9.  }
10.
11. public function index(){
12.  $data['siswa'] = $this->SiswaModel->view();
13.  $this->load->view('siswa/index', $data);
14.  }
15.
16. public function tambah(){
17.  if($this->input->post('submit')){ // Jika user mengklik tombol submit yang ada di form
18.  if($this->SiswaModel->validation("save")){ // Jika validasi sukses atau hasil validasi adalah TRUE
19.  $this->SiswaModel->save(); // Panggil fungsi save() yang ada di SiswaModel.php
20.  redirect('siswa');
21.  }
22.  }
23.
24.  $this->load->view('siswa/form_tambah');
25.  }
26.
27. public function ubah($nis){
28.  if($this->input->post('submit')){ // Jika user mengklik tombol submit yang ada di form
29.  if($this->SiswaModel->validation("update")){ // Jika validasi sukses atau hasil validasi adalah TRUE
30.  $this->SiswaModel->edit($nis); // Panggil fungsi edit() yang ada di SiswaModel.php
31.  redirect('siswa');
32.  }
33.  }
34.
35.  $data['siswa'] = $this->SiswaModel->view_by($nis);
36.  $this->load->view('siswa/form_ubah', $data);
37.  }
38.
39. public function hapus($nis){
40.  $this->SiswaModel->delete($nis); // Panggil fungsi delete() yang ada di SiswaModel.php
41.  redirect('siswa');
42.  }
43. }

```

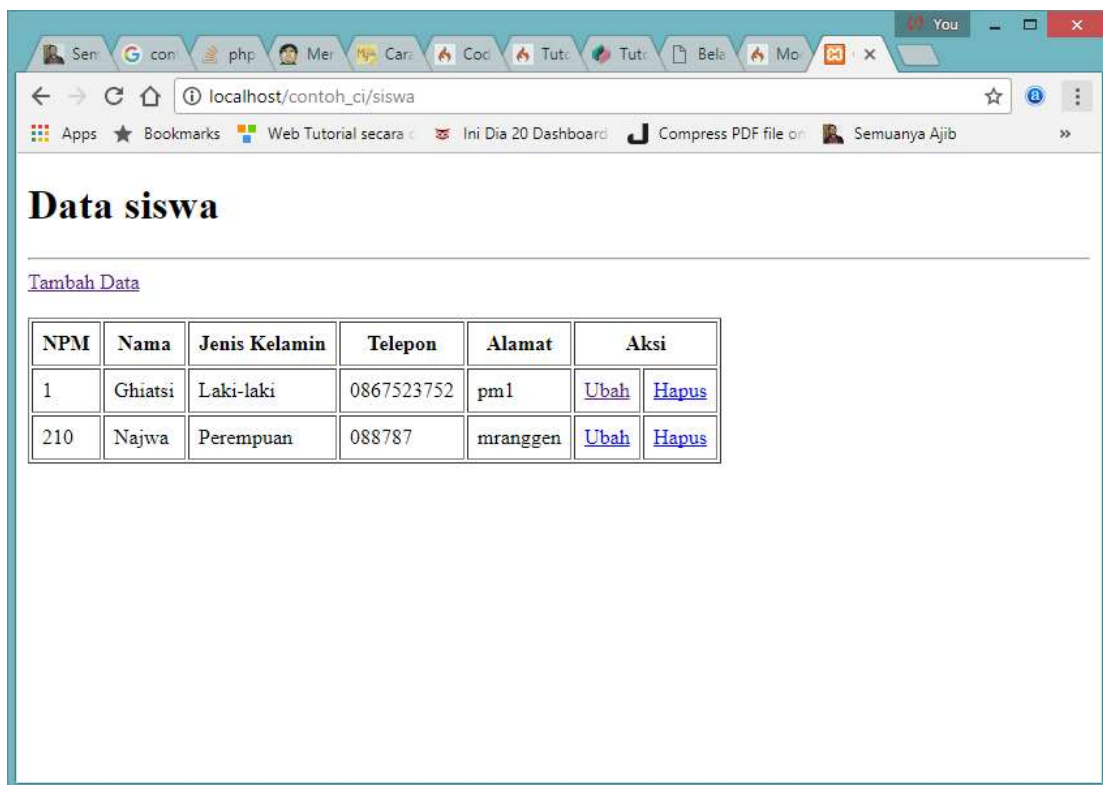
Pada controller ini, kita membuat 3 fungsi. fungsi pertama yaitu `public function __construct(){`, fungsi ini berfungsi untuk menjalankan suatu aksi ketika controller main diload. Didalm fungsi ini, terdapat kode `$this->load->model('SiswaModel');`, kode tersebut berfungsi untuk memuat (meload) model siswa (yang pada step sebelumnya kita buat) agar kita bisa mengakses fungsi-fungsi yang ada didalam model tersebut.

Fungsi yang kedua yaitu `public function index()`. Dalam fungsi ini ada kode `$data['siswa'] = $this->SiswaModel->view();`, kode tersebut digunakan untuk mengambil hasil query sql dari fungsi `view()` yang ada pada model siswa (`SiswaModel.php`) lalu memasukannya ke dalam array data dengan index siswa (`$data['siswa']`). Pada fungsi `index()` ini juga, terdapat kode `$this->load-`

>`view('siswa/index', $data);` Kode tersebut berfungsi untuk memuat (meload) file `index.php` (file ini akan kita buat pada step selanjutnya) dan mengirimkan array data (`$data`) tadi ke file tersebut.

BUAT VIEW

Selanjutnya kita akan membuat file view untuk menampilkan data siswanya. Buat sebuah folder dengan nama `siswa` pada folder `xampp/htdocs/crud_ci/application/views/`. Lalu buat sebuah file dengan nama `index.php`, lalu simpan pada folder siswa yang tadi kita buat (`xampp/htdocs/crud_ci/application/views/siswa/`). Berikut ini tampilan dan kodenya :



1. `<html>`
2. `<head>`
3. `<title>CRUD Codeigniter</title>`
4. `</head>`
5. `<body>`
6. `<h1>Data siswa</h1>`

```

7. <hr>
8.
9. <a href='<?php echo base_url("siswa/tambah"); ?>'>Tambah Data</a><br><br>
10.
11. <table border="1" cellpadding="7">
12. <tr>
13. <th>NPM</th>
14. <th>Nama</th>
15. <th>Jenis Kelamin</th>
16. <th>Telepon</th>
17. <th>Alamat</th>
18. <th colspan="2">Aksi</th>
19. </tr>
20.
21. <?php
22. if( ! empty($siswa)){ // Jika data siswa tidak sama dengan kosong, artinya jika data siswa ada
23. foreach($siswa as $data){
24. echo "<tr>
25. <td>".$data->nis."</td>
26. <td>".$data->nama."</td>
27. <td>".$data->jenis_kelamin."</td>
28. <td>".$data->telp."</td>
29. <td>".$data->alamat."</td>
30. <td><a href='".base_url("siswa/ubah/".$data->nis)."'>Ubah</a></td>
31. <td><a href='".base_url("siswa/hapus/".$data->nis)."'>Hapus</a></td>
32. </tr>";
33. }
34. }else{ // Jika data siswa kosong
35. echo "<tr><td align='center' colspan='7'>Data Tidak Ada</td></tr>";
36. }
37. ?>
38. </table>
39. </body>
40. </html>

```

Pada kode diatas terdapat kode `foreach($siswa as $data){`, kode tersebut akan menampilkan satu per satu data siswa sampai data siswa yang terakhir.

FORM TAMBAH

Selanjutnya kita akan coba membuat form untuk menambah datanya. Buat sebuah file dengan nama `form_tambah.php`, lalu simpan pada folder `xampp/htdocs/crud_ci/application/views/siswa/`. Berikut tampilan dan kodenya :

The screenshot shows a web browser window with the address bar displaying 'localhost/contoh_ci/siswa/tambah'. The page title is 'Form Tambah Data Mahasiswa'. The form contains the following fields:

- NIS:
- Nama:
- Jenis Kelamin: ☐ Laki-laki ☐ Perempuan
- Telepon:
- Alamat:

At the bottom of the form are two buttons: 'Simpan' and 'Batal'.

```

1. <html>
2. <head>
3. <title>Form Tambah - CRUD Codeigniter</title>
4. </head>
5. <body>
6. <h1>Form Tambah Data Mahasiswa</h1>
7. <hr>
8.
9. <!-- Menampilkan Error jika validasi tidak valid -->
10. <div style="color: red;"><?php echo validation_errors(); ?></div>
11.
12. <?php echo form_open("siswa/tambah"); ?>
13. <table cellpadding="8">
14. <tr>
15. <td>NIS</td>
16. <td><input type="text" name="input_nis" value="<?php echo set_value('input_nis'); ?>"></td>
17. </tr>
18. <tr>
19. <td>Nama</td>
20. <td><input type="text" name="input_nama" value="<?php echo set_value('input_nama'); ?>"></td>
21. </tr>
22. <tr>
23. <td>Jenis Kelamin</td>

```

```

24. <td>
25. <input type="radio" name="input_jeniskelamin" value="Laki-laki" <?php echo set_radio('jeniskelamin',
    'Laki-laki'); ?>> Laki-laki
26. <input type="radio" name="input_jeniskelamin" value="Perempuan" <?php echo
    set_radio('jeniskelamin', 'Perempuan'); ?>> Perempuan
27. </td>
28. </tr>
29. <tr>
30. <td>Telepon</td>
31. <td><input type="text" name="input_telp" value="<?php echo set_value('input_telp'); ?>"></td>
32. </tr>
33. <tr>
34. <td>Alamat</td>
35. <td><textarea name="input_alamat"><?php echo set_value('input_alamat'); ?></textarea></td>
36. </tr>
37. </table>
38.
39. <hr>
40. <input type="submit" name="submit" value="Simpan">
41. <a href="<?php echo base_url(); ?>"><input type="button" value="Batal"></a>
42. <?php echo form_close(); ?>
43. </body>
44. </html>

```

form_open("siswa/tambah")

Kode ini sama saja dengan kita membuat tag form. Kode tersebut akan menghasilkan tag form seperti ini `<form method="post" action="http://localhost/crud_ci/siswa/tambah">`

set_value('input_nis')

Ini berguna untuk men-set value dari textbox input_nis agar ketika terjadi error dan halaman kembali ke form, data yang di input sebelumnya tidak hilang. input_nis disitu adalah name dari textbox nya.

form_close()

Ini akan menghasilkan penutup tag form : `</form>`

FORM UBAH

Selanjutnya kita akan coba membuat form untuk menambah datanya. Buat sebuah file dengan nama `form_ubah.php`, lalu simpan pada folder `xampp/htdocs/crud_ci/application/views/siswa/`. Berikut tampilan dan kodenya :

The screenshot shows a web browser window with the address bar displaying 'localhost/contoh_ci/siswa/ubah/1'. The page title is 'Form Ubah Data Mahasiswa'. The form contains the following fields and values:

- NIS: 1
- Nama: Ghiatsi
- Jenis Kelamin: ☒ Laki-laki ☐ Perempuan
- Telepon: 0867523752
- Alamat: pm1

At the bottom of the form, there are two buttons: 'Ubah' and 'Batal'.

```

1. <html>
2. <head>
3. <title>Form Ubah - CRUD Codeigniter</title>
4. </head>
5. <body>
6. <h1>Form Ubah Data Mahasiswa</h1>
7. <hr>
8.
9. <!-- Menampilkan Error jika validasi tidak valid -->
10. <div style="color: red;"><?php echo validation_errors(); ?></div>
11.
12. <?php echo form_open("siswa/ubah/".$siswa->nis); ?>
13. <table cellpadding="8">
14. <tr>
15. <td>NIS</td>
16. <td><input type="text" name="input_nis" value="<?php echo set_value('input_nis', $siswa->nis); ?>"
    readonly></td>
17. </tr>
18. <tr>
19. <td>Nama</td>
20. <td><input type="text" name="input_nama" value="<?php echo set_value('input_nama', $siswa-
    >nama); ?>"></td>
21. </tr>

```



```

22. <tr>
23. <td>Jenis Kelamin</td>
24. <td>
25. <input type="radio" name="input_jeniskelamin" value="Laki-laki" <?php echo set_radio('jeniskelamin',
    'Laki-laki', ($siswa->jenis_kelamin == "Laki-laki")? true : false); ?>> Laki-laki
26. <input type="radio" name="input_jeniskelamin" value="Perempuan" <?php echo
    set_radio('jeniskelamin', 'Perempuan', ($siswa->jenis_kelamin == "Perempuan")? true : false); ?>>
    Perempuan
27. </td>
28. </tr>
29. <tr>
30. <td>Telepon</td>
31. <td><input type="text" name="input_telp" value="<?php echo set_value('input_telp', $siswa->telp);
    ?>"></td>
32. </tr>
33. <tr>
34. <td>Alamat</td>
35. <td><textarea name="input_alamat"><?php echo set_value('input_alamat', $siswa->alamat);
    ?></textarea></td>
36. </tr>
37. </table>
38.
39. <hr>
40. <input type="submit" name="submit" value="Ubah">
41. <a href="<?php echo base_url(); ?>"><input type="button" value="Batal"></a>
42. <?php echo form_close(); ?>
43. </body>
44. </html>

```

Variabel `$siswa` ini adalah hasil dari query pada controller siswa di function ubah. Lalu kita set terhadap masing-masing inputan pada form agar bisa menampilkan data sebelumnya. disini kita memanfaatkan fungsi buatan codeigniter yaitu `set_value(name_textbox, valuenya)`.

Referensi :

<http://www.mynotescode.com/crud-dengan-codeigniter-dan-mysql/>