

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT

Jobsheet 2

Read dan Input Database serta Form Validation



NAMA : SILVI INDAH NOVITA SARI
NIM : 1841720006
KELAS : 2B – TI

D-IV TEKNIK INFORMATIKA
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2019



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-1: Pengenalan CodeIgniter

Mata Kuliah Pemrograman WEB Lanjut

Pengampu: Tim Ajar Pemrograman Web Lanjut

10 Februari 2020

Praktikum – Bagian 1: Menampilkan data dari *database* menggunakan codeIgniter

Langkah	Keterangan
1	Buka kembali folder CodeIgniter yang telah dibuat pada Jobsheet sebelumnya
2	<p>Buka <i>controller</i> mahasiswa dan tambahkan <i>code</i> berikut (line 9)</p> <pre>application > controllers > mahasiswa.php 1 <?php 2 defined('BASEPATH') OR exit('No direct script access allowed'); 3 4 class mahasiswa extends CI_Controller { 5 6 public function index() 7 { 8 //modul untuk load database 9 \$this->load->database(); 10 \$data['title']='List Mahasiswa'; 11 \$this->load->view('template/header', \$data); 12 \$this->load->view('mahasiswa/index'); 13 \$this->load->view('template/footer'); 14 } 15 16 } 17 18 19 /* End of file Controllername.php */ 20 21 ?></pre> <p>Keterangan :</p> <ul style="list-style-type: none">- Pada line 9 menjelaskan bawa <i>function</i> index ini akan melakukan <i>load</i> database. <i>Load</i> database ini hanya dilakukan di <i>function</i> index saja jika ada <i>function</i> lain maka kita harus melakukan <i>load</i> database lagi. Sehingga hal ini dirasa tidak praktis maka untuk itu kita harus tambahkan <i>code</i> agar <i>load</i> database ini dapat digunakan banyak <i>function</i>.
3	Tambahkan <i>code</i> berikut agar database dapat digunakan banyak <i>function</i> (line 7-12)

```

application > controllers > mahasiswa.php
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class mahasiswa extends CI_Controller {
5      //fungsi yang akan dijalankan saat classnya diinstansiasi
6      public function __construct(){
7          //digunakan untuk menjalankan fungsi construct pada class parent
8          parent::__construct();
9          $this->load->database();
10     }
11
12     public function index()
13     {
14         //modul untuk load database
15         //$this->load->database();
16         $data['title']='List Mahasiswa';
17         $this->load->view('template/header', $data);
18         $this->load->view('mahasiswa/index');
19         $this->load->view('template/footer');
20     }
21 }
22
23 }
24
25 /* End of file Controllername.php */
26
27 ?>

```

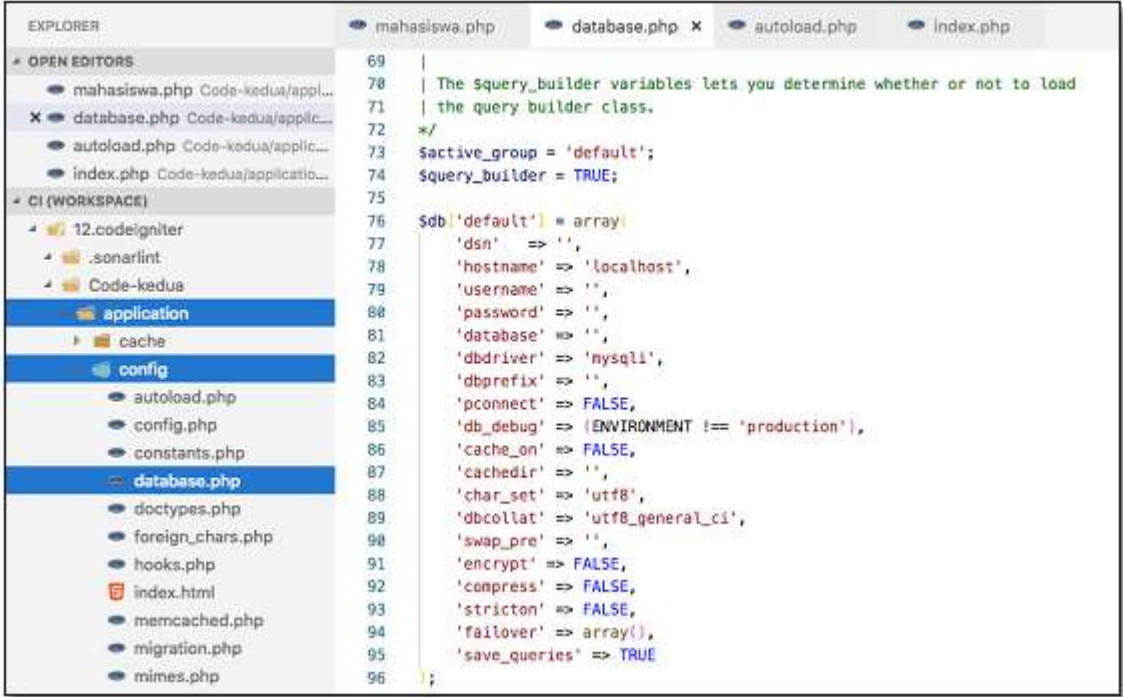
Keterangan:

- Line 7 – 12 adalah sebuah *function* construct (**PERHATIKAN ITU MENGGUNAKAN DOUBLE UNDERSCORE __construct (tanpa spasi)**)
- *Function* construct dijalankan pada saat classnya diinstansiasi
- Line 10 digunakan untuk menjalankan fungsi construct pada class parent dimana class parent ini adalah CI_Controller
- Line 11 digunakan agar untuk memload database (Semua fungsi di class mahasiswa dapat menggunakannya)
- Jika kita sudah melakukan load database pada saat *function* construct maka pada *function* index kita comment agar tidak bekerja dua kali

4 Selain 2 cara diatas kita juga dapat melakukan proses *load* database pada **application/config/autoload** dan tambahkan pada database pada **autoload['libraries']=array('database')** seperti pada gambar

	<div> <div> <div>CODE-KEDUA</div> <div> <div>application</div> <div> <div>cache</div> <div>config</div> <div>autoload.php</div> <div>config.php</div> <div>constants.php</div> <div>database.php</div> <div>doctypes.php</div> <div>foreign_chars.php</div> <div>hooks.php</div> </div> </div> </div> <div> <div>34</div> <div>35</div> <div>36</div> <div>37</div> <div>38</div> <div>39</div> <div>40</div> <div>41</div> <div>42</div> <div>43</div> <div>44</div> <div>45</div> </div> <div> <div>-----</div> <div>Auto-load Packages</div> <div>-----</div> <div>Prototype:</div> <div>\$autoload['packages'] = array(APPPATH.'thi</div> <div>-----</div> <div>*/</div> <div>\$autoload['libraries'] = array(['database']);</div> <div>-----</div> <div>/*</div> <div>-----</div> </div> </div>
5	<p>Untuk kasus ini kita gunakan cara yang terakhir (no 4) karena kita asumsikan bahwa semua <i>controller</i> membutuhkan database. Maka kita <i>comment code</i> function construct yang ada di <i>controller</i> mahasiswa</p>

	<pre> application > controllers > mahasiswa.php > ... 1 <?php 2 defined('BASEPATH') or exit('No direct script access allowed'); 3 4 class mahasiswa extends CI_Controller 5 { 6 //fungsi yang akan dijalankan saat classnya diinstansiasi 7 // public function __construct() 8 // { 9 //digunakan untuk menjalankan fungsi construct pada class p 10 // parent::__construct(); 11 // \$this->load->database(); 12 // } 13 14 //public function index() 15 // { 16 // \$this->load->model('mahasiswa_model'); 17 // \$this->load->database('mahasiswa_model'); 18 //modul untuk load database 19 // \$this->load->database(); 20 // \$data['title'] = 'List Mahasiswa'; 21 // \$data['mahasiswa'] = \$this->mahasiswa_model->getAllmaha 22 // \$this->load->view('template/header', \$data); 23 // \$this->load->view('mahasiswa/index', \$data); 24 // \$this->load->view('template/footer'); 25 //} 26 } 27 28 /* End of file Controllername.php */ </pre>
6	<p>Setelah itu kita akan mengatur konfigurasi database. Untuk mengatur konfigurasi database maka buka folder application/config/database.php</p>

	 <pre> 69 70 The \$query_builder variable lets you determine whether or not to load 71 the query builder class. 72 */ 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' => 'mysql', 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); </pre>
7	Kita set konfigurasinya seperti pada gambar dibawah ini

CODE-KEDUA

application

cache

config

autoload.php

config.php

constants.php

database.php

doctypes.php

foreign_chars.php

hooks.php

index.html

memcached.php

migration.php

mimes.php

profiler.php

routes.php

smileys.php

user_agents.php

controllers

Home.php

index.html

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

```

$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'kuliah_CI',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

Keterangan :

- Line 78 artinya *hostname* yang digunakan adalah localhost
- Line 79 artinya *username* (phpmyadmin) yang digunakan apa
- Line 80 artinya *password* (phpmyadmin) yang digunakan apa
- Line 81 artinya nama database yang nanti digunakan

8

Jika kita lihat konfigurasi database yang kita buat maka kita perlu sebuah database dengan nama **kuliah_CI**. Maka buat lah sebuah database dengan nama kuliah_CI dan dengan tabel mahasiswa seperti pada gambar dibawah ini

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(30)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index More
2	nama	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index More
3	nim	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index More
4	email	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index More
5	jurusan	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index More

9

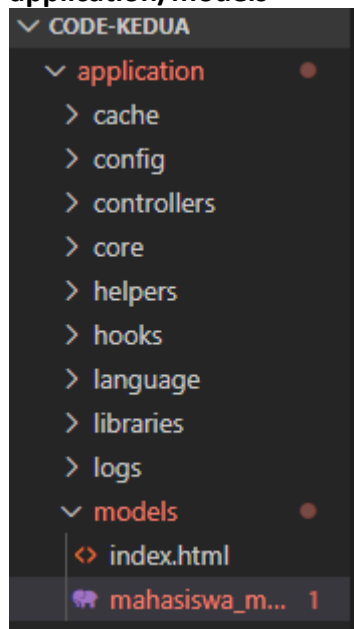
Tambahkan beberapa data ke dalam tabel mahasiswa seperti pada gambar (pada contoh terdapat 3 data, jumlah data lebih banyak tidak masalah)

+ Options								
				id	nama	nim	email	jurusan
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	dina	1002345100	dina@gmail.com	kimia
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	dika	1002345101	dika@gmail.com	informatika
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	soni	1002345102	soni@gmail.com	mesin

10

Setelah kita membuat sebuah *database*, maka langkah berikutnya adalah membuat sebuah model untuk melakukan proses *database*. Di dalam model ini kita akan membuat sebuah *function* untuk melakukan perintah ambil data dari *database*. Untuk lebih jelasnya ikuti langkah-langkah berikut:

- Buatlah sebuah file model dengan nama **mahasiswa_model** pada folder **application/models**



- Tambahkan *function* getAllmahasiswa pada mahasiswa_model.php


```

application > models > mahasiswa_model.php > mahasiswa_model > getAllmahasiswa
1  <?php
2
3  defined('BASEPATH') or exit('No direct script access allowed');
4
5  class mahasiswa_model extends CI_Model {
6
7
8      public function getAllmahasiswa()
9      {
10
11          $query = $this->db->get('mahasiswa');
12
13
14
15          return $query->result_array();
16      }
17
18  }
19
20  /* End of file Controllername.php */
21

```

Keterangan :

- Pada *line* 11 membuat sebuah *query builder* yang digunakan untuk mengambil semua data
 - https://www.codeigniter.com/user_guide/database/query_builder.html#selecting-data
 - Pada *line* 15 digunakan untuk mengenerate hasil query dalam bentuk array
 - https://www.codeigniter.com/user_guide/database/results.html
 - https://www.codeigniter.com/user_guide/database/results.html#result-arrays
- c. Selain menggunakan query builder kita juga bisa menggunakan cara lain seperti gambar dibawah ini

```
mahasiswa_model.php X mahasiswa.php
application > models > mahasiswa_model.php > mahasiswa_model
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class mahasiswa_model extends CI_Model
6  {
7      public function getAllmahasiswa()
8      {
9          return $this->db->get('mahasiswa')->result_array();
10     }
11
12 }
13
14 /* End of file Controllername.php */
15
```

- 11 Setelah kita mempunyai sebuah model dengan nama **mahasiswa_model** maka langkah berikutnya kita akan *load* model ini ke dalam **controller mahasiswa**.

```
public function index()
{
    $this->load->model('mahasiswa_model');

    //modul untuk load database
    //$this->load->database();
    $data['title'] = 'List Mahasiswa';
    $data['mahasiswa'] = $this->mahasiswa_model->getAllmahasiswa();
    $this->load->view('template/header', $data);
    $this->load->view('mahasiswa/index', $data);
    $this->load->view('template/footer');
}

/* End of file Controllername.php */
```

Keterangan :

- Line 17 digunakan untuk melakukan *load* model dengan nama modelnya adalah **mahasiswa_model**
- Line 22 artinya ada data yang berupa array dengan isi arraynya adalah mahasiswa dimana terdapat isi berupa function **getAllmahasiswa()** yang ada pada model dengan

nama

mahasiswa_model

- Line 24 tambahkan **\$data**

```

mahasiswa.php  mahasiswa_model.php  index.php
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class mahasiswa_model extends CI_Model {
6
7      public function getAllmahasiswa()
8      {
9
10
11      // atau bisa juga menggunakan code berikut
12      return $this->db->get('mahasiswa')->result_array();
13      }
14
15  }
16
17  /* End of file mahasiswa_model.php */
18
19  ?>

mahasiswa.php  mahasiswa_model.php  index.php
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class mahasiswa extends CI_Controller {
5
6      public function index()
7      {
8          $this->load->model('mahasiswa_model');
9
10
11      // modul untuk load database
12      // $this->load->database();
13      $data['title']='List Mahasiswa';
14      $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
15      $this->load->view('template/header',$data);
16      $this->load->view('mahasiswa/index',$data);
17      $this->load->view('template/footer');
18
19      }
20
21
22  /* End of file Controllername.php */
23  ?>
24
```

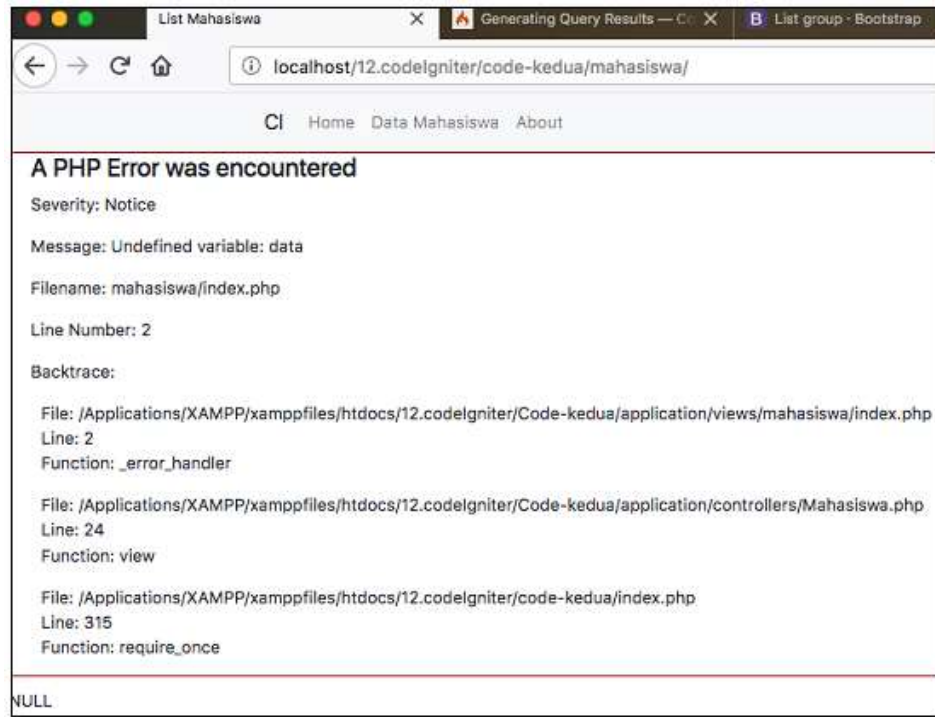
12

Jika kita lakukan `var_dump($mahasiswa)` pada **index.php** pada **views/mahasiswa/index**

```
mahasiswa_model.php  index.php X
application > views > mahasiswa > index.php > ...
1  <?
2  var_dump($mahasiswa);
3  ?>
4  <div class="container">
5      <!--mt-3 artinya margin top 3 -->
6      <div class="row mt-3">
7          <div class="col-md-6">
8              <h2>Daftar Mahasiswa</h2>
9              <ul class="list-group">
10                 <li class="list-group-item">Cras justo odio</li>
11                 <li class="list-group-item">Dapibus ac facilisis
12                 <li class="list-group-item">Morbi leo risus</li>
13                 <li class="list-group-item">Porta ac consectetur
14                 <li class="list-group-item">Vestibulum at eros</li>
15             </ul>
16         </div>
17     </div>
18 </div>
```

Keterangan

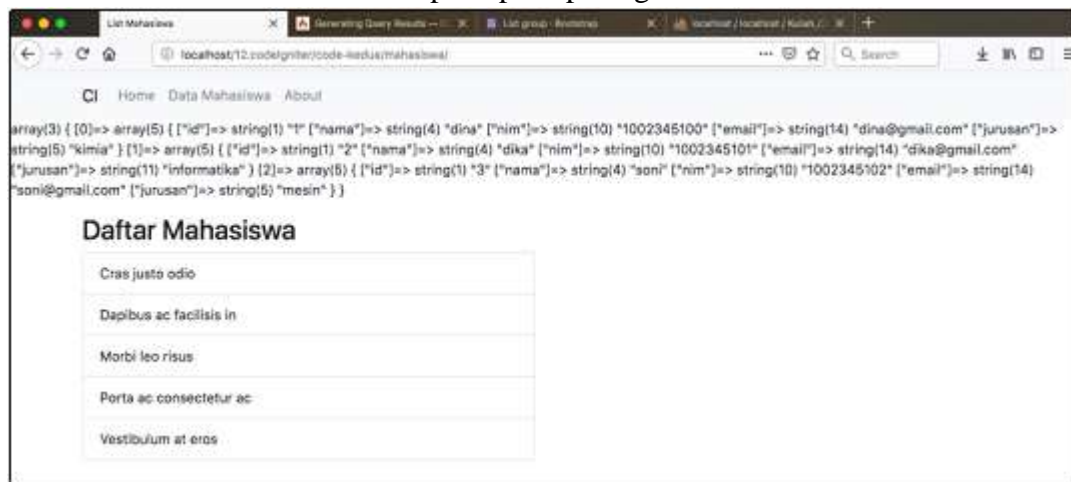
- \$mahasiswa didapat dari mana ??? -> jika kita lihat *line 22* pada *controller* mahasiswa terdapat [mahasiswa]
- Jika kita lihat pada *controller* mahasiswa terdapat perintah/code seperti berikut:
 - `$data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();` (Code pertama)
 - `$this->load->view('mahasiswa/index',$data);` (Code kedua)
- Jika kita lihat *code* kedua, ada \$data yang isi arraynya adalah mahasiswa dan isi mahasiswa tersebut adalah **function** `getAllmahasiswa` pada model dengan nama mahasiswa_model. Dimana nanti isi dari `$data['mahasiswa']` ini dikirimkan ke halaman **view** dengan **2 paramater** yaitu **mahasiswa/index** dan `$data` (yang isinya mahasiswa, sehingga jika mau `var_dump` di halaman mahasiswa/index, kita menggunakan `$mahasiswa`. Jika pada mahasiswa/index ke `var_dump $data` maka akan terjadi *error* seperti pada gambar dibawah ini)



Hal ini disebabkan karena tidak ketahui data pada mahasiswa/index.php pada *line* 2 berupa `var_dump($mahasiswa)`, maka pastikan terlebih dahulu pada *controller* mahasiswa.php sudah terdapat `$data` di *line* 13 dan 15.

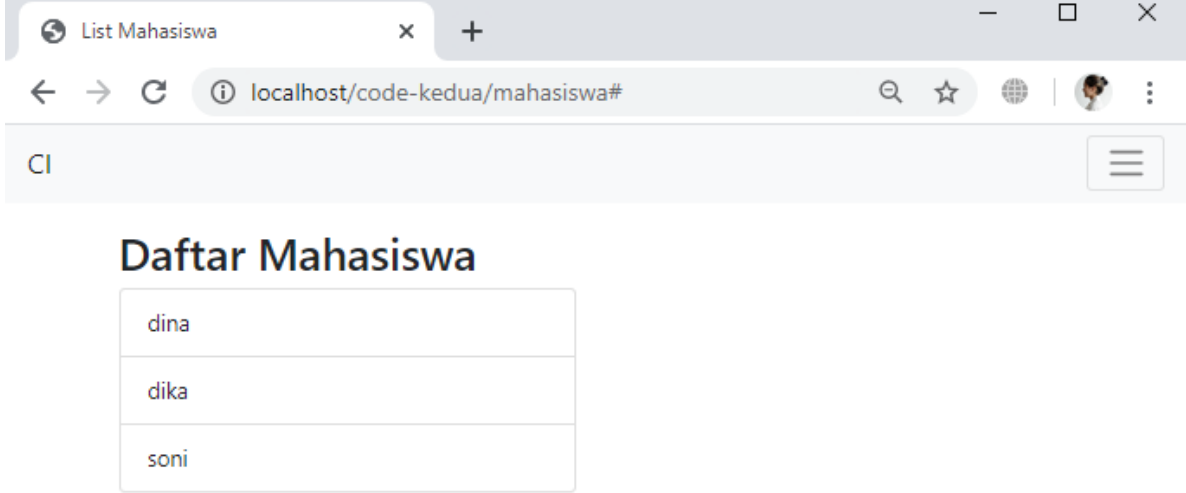
13

Jika berhasil maka data akan tampil seperti pada gambar berikut



14

Agar semua isi dari tabel dapat tampil di halaman index mahasiswa maka kita harus lakukan proses **looping**. Gunakan *code* berikut pada **views/mahasiswa/index**

	<pre> application > views > mahasiswa > index.php > div.container > div.row.mt-3 > c 1 <div class="container"> 2 <!--mt-3 artinya margin top 3 --> 3 <div class="row mt-3"> 4 <div class="col-md-6"> 5 <h2>Daftar Mahasiswa</h2> 6 <ul class="list-group"> 7 <?php foreach (\$mahasiswa as \$mhs) : ?> 8 <li class="list-group-item"><?= \$mhs['nama' 9 <?php endforeach; ?> 10 11 </div> 12 </div> 13 </div> </pre>
15	<p>Jika berhasil maka tampilan akan seperti pada gambar dibawah ini</p> 
16	<p>Jika kita klik Home, halaman Home belum dapat tampil. Maka dari itu kita akan set agar semua dapat saling terhubung. Untuk itu kita membutuhkan yang namanya base_url. untuk lebih jelasnya ikuti langkah-langkah berikut:</p> <p>a. Tambahkan <i>code</i> berikut pada views/template/header (line 26,27,28)</p>

```

application > views > template > header.php > hint > body > nav.navbar.navbar-expand-lg.navbar-light.bg-light
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5 <!-- Required meta tags -->
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8
9 <!-- Bootstrap CSS -->
10 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-C6hZ186VtSzSNaRmKGuhdr5u17V74bnBivB/kpzf4m5/rfdUva2JAk+9eu8xfzZwaq=">
11
12 <!-- Title and meta, world/1/1/1/1 -->
13 <title><? $title </?></title>
14
15 </head>
16
17 <body>
18
19 <!-- navbar -->
20 <nav class="navbar navbar-expand-lg navbar-light bg-light">
21 <a class="navbar-brand" href="#">CI</a>
22 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" data-keyboard="true">
23 <span class="navbar-toggler-icon"></span>
24 </button>
25 <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
26 <div class="navbar-nav">
27 <a class="nav-item nav-link active" href="#"><? $base_url()> /home <span class="sr-only">(current)</span></a>
28 <a class="nav-item nav-link" href="#"><? $base_url()> /mahasiswa/Duta Mahasiswa</a>
29 <a class="nav-item nav-link" href="#"><? $base_url()> /</a>
30 <a class="nav-item nav-link disabled" href="#">About</a>
31 </div>
32 </div>
33

```

Keterangan :

- Line 26, 27, 28 tambahkan `base_url` (pastikan **config/config.php** sudah diset)
`$config['base_url'] = 'http://localhost/12.codeigniter/code-kedua/';` (code_kedua adalah nama project CI jadi sesuaikan dengan nama project CI kalian)
- Agar halaman web mahasiswa dapat ditampilkan saat *user* mengklik mahasiswa, maka setelah **base_url** tambahkan **controller mahasiswa**

- b. Agar `base_url` dikenali oleh sistem maka tambahkan konfigurasi pada **config/autoload.php** tambahkan url pada `$autoload['helper']=array('url')`

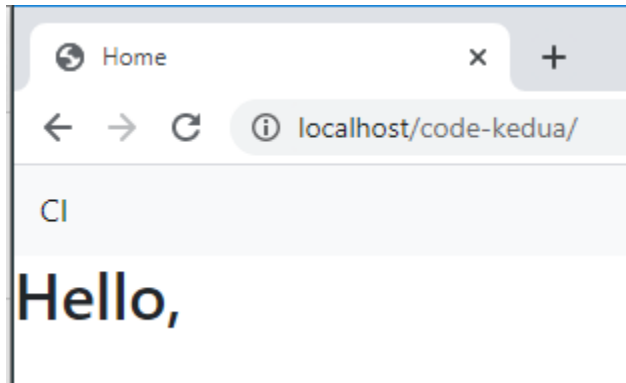
```

application > config > autoload.php > ...
89 |
90 | $autoload['helper'] = array('url', 'file');
91 */
92 $autoload['helper'] = array('url');
93
94

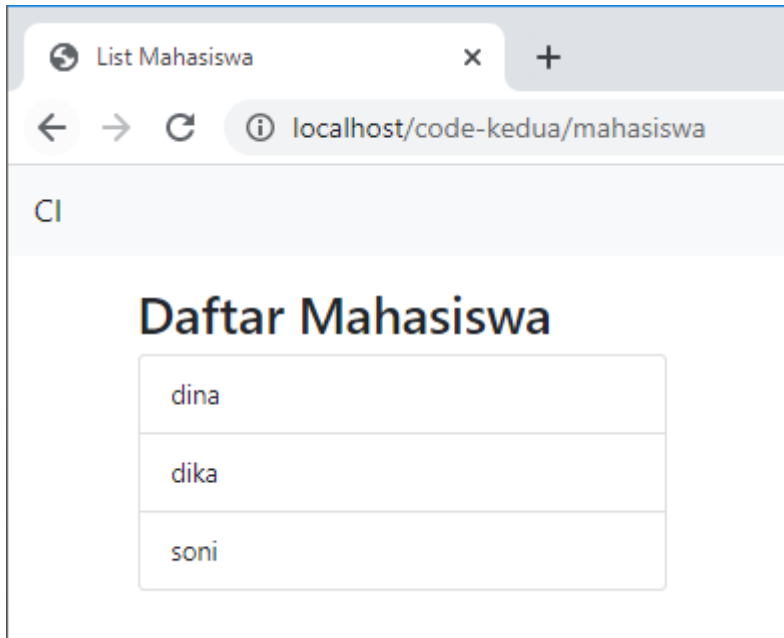
```

c. Jika berhasil maka semua halaman dapat ditampilkan pada saat *user* melakukan *event click*.

a. Home



b. Data Mahasiswa



Bagian 2: Membuat Form Input dan Menambahkan Form Validation

1.	Buatlah sebuah <i>button</i> tambah data pada <code>views/mahasiswa/index.php</code> (<i>line 3-7</i>). Tambahkan <i>code</i> pada <i>line 3-7</i> untuk membuat sebuah <i>button</i>
----	--

	<pre> application > views > mahasiswa > index.php > div.container 1 <div class="container"> 2 <div class="row mt-4"> 3 <div class="col-md-6"> 4 Tambah Data 5 </div> 6 </div> 7 <!--mt-3 artinya margin top 3 --> 8 <div class="row mt-3"> 9 <div class="col-md-6"> 10 <h2>Daftar Mahasiswa</h2> 11 <ul class="list-group"> 12 <?php foreach (\$mahasiswa as \$mhs) : ?> 13 <li class="list-group-item"><?php echo \$mhs['nama'] 14 <?php endforeach; ?> 15 16 </div> 17 </div> 18 </div> </pre>
2.	<p>Dikarenakan mahasiswa_model ini nantinya akan digunakan oleh semua <i>function</i> di <i>controller</i> mahasiswa, maka “ <code>\$this->load->model('mahasiswa_model');</code> “ yang awalnya hanya berada pada function index, kita ubah ke function construct agar semua <i>function</i> dapat menggunakan mahasiswa_model.</p> <pre> application > controllers > mahasiswa.php > ... 1 <?php 2 defined('BASEPATH') or exit('No direct script access allowed'); 3 4 class mahasiswa extends CI_Controller 5 { 6 7 function __construct() 8 { 9 parent::__construct(); 10 \$this->load->model('mahasiswa_model'); 11 } 12 13 14 public function index() 15 { 16 //\$this->load->database('mahasiswa_model'); 17 //modul untuk load database 18 //\$this->load->database(); 19 \$data['title'] = 'List Mahasiswa'; 20 \$data['mahasiswa'] = \$this->mahasiswa_model->getAllmahas 21 \$this->load->view('template/header', \$data); 22 \$this->load->view('mahasiswa/index', \$data); 23 \$this->load->view('template/footer'); 24 } 25 } 26 27 /* End of file Controllername.php */ </pre>

3. Setelah itu tambahkan **function tambah** pada **controller mahasiswa** dan tambahkan **code** berikut.

(Tips : jika menggunakan vsCode untuk menduplikat 1 baris dapat menggunakan **alt+shift+panah_bawah**)

```
application > controllers > mahasiswa.php > mahasiswa > tambah

25
26     public function tambah()
27     {
28         $data['title'] = 'Form Menambahkan Data Mahasiswa';
29         $this->load->view('template/header', $data);
30         $this->load->view('mahasiswa/tambah', $data);
31         $this->load->view('template/footer');
32     }
33 }
34
35
36
37 /* End of file Controllername.php */
38
```

4. Setelah kita membuat **function tambah** pada **controller mahasiswa**, selanjutnya buatlah **form tambah** pada **views/mahasiswa/tambah.php**

```
application > views > mahasiswa > tambah.php > div.container > div.row > div.col-md-6 > div.card > div.card-body > form > div.form-group

1 <div class="container">
2     <div class="row">
3         <div class="col-md-6">
4             <!-- http://getbootstrap.com/docs/4.1/components/card/ -->
5             <div class="card">
6                 <div class="card-header">
7                     Form Tambah Mahasiswa
8                 </div>
9                 <div class="card-body">
10                     <form action="" method="post">
11                         <!-- http://getbootstrap.com/docs/4.1/components/card/ -->
12                         <div class="form-group">
13                             <label for="nama">Nama </label>
14                             <input type="text" class="form-control" id="nama">
15                         </div>
16                         <div class="form-group">
17                             <label for="nim">Nim </label>
18                             <input type="number" class="form-control" id="nim">
19                         </div>
20                         <div class="form-group">
21                             <label for="email">Email </label>
22                             <input type="email" class="form-control" id="email">
23                         </div>
24                         <div class="form-group">
25                             <label for="jurusan">Jurusan </label>
26                             <select class="form-control" id="jurusan" name="jurusan">
27                                 <option selected="">-----</option>
28                                 <option value="1">Kimia</option>
29                                 <option value="2">Informatika</option>
30                                 <option value="3">Mesin</option>
31                             </select>
32                         </div>
33                         <button type="submit" name="submit" class="btn btn-primary float-right">Submit</button>
34                     </form>
35                 </div>
36             </div>
37         </div>
38     </div>
39 </div>
```

- Ganti **value** dari **line 28-31** menjadi seperti di bawah ini (**Agar di tabel tidak tersimpan value berupa angka**)

```

24     <div class="form-group">
25         <label for="jurusan">Jurusan </label>
26         <select class="form-control" id="jurusan" name="jurusan">
27             <option selected>-----</option>
28             <option value="Informatika"></option>
29             <option value="Kimia"></option>
30             <option value="Mesin"></option>
31         </select>
32     </div>
33     <button type="submit" name="submit" class="btn btn-primary">
34 </form>
35 </>

```

5. Maka akan dihasilkan halaman seperti di bawah ini.

The screenshot shows a web browser window with the following elements:

- Browser Tab:** Form Menambahkan Data Mahasiswa x
- Address Bar:** localhost/code-kedua/mahasiswa/tambah
- Page Header:** CI
- Navigation Menu:** Home, Data Mahasiswa, About
- Form Title:** Form Tambah Mahasiswa
- Form Fields:**
 - Nama:** Text input field
 - Nim:** Text input field
 - Email:** Text input field
 - Jurusan:** Dropdown menu with '-----' selected
- Submit Button:** Blue button labeled 'Submit'

6. Setelah berhasil membuat *form* tambah data, selanjutnya buka file **index** pada **views/mahasiswa/index.php** dan kita atur **href** nya

```
ation > views > mahasiswa > index.php > div.container > div.row.mt-4 > div.col-md-6
1  <div class="container">
2      <div class="row mt-4">
3          <div class="col-md-6">
4              <a href="php base_url(); ?mahasiswa/tambah" class="btn btn-primary">Tambah</a>
5          </div>
6      </div>
7      <!--mt-3 artinya margin top 3 -->
8      <div class="row mt-3">
9          <div class="col-md-6">
10             <h2>Daftar Mahasiswa</h2>
11             <ul class="list-group">
12                 <?php foreach ($mahasiswa as $mhs) : ?>
13                     <li class="list-group-item"><?php echo $mhs['nama'];>
14                 <?php endforeach; ?>
15             </ul>
16          </div>
17      </div>
18 </div>
```

7. Tambahkan validasi menggunakan `form_validation`. Contoh validasi *form* adalah seperti *field* nama harus diisi, nim harus angka dsb. Untuk lebih jelasnya ikuti langkah berikut ini :

- a. Tambahkan `form_validation`

https://www.codeigniter.com/user_guide/libraries/form_validation.html?highlight=form#the-form (searching controller)

- b. Buka *controller* mahasiswa dan tambahkan `form_validation`

```
26     public function tambah()
27     {
28         $data['title'] = 'Form Menambahkan Data Mahasiswa';
29         if ($this->form_validation->run() == FALSE) {
30             #code
31             $this->load->view('template/header', $data);
32             $this->load->view('mahasiswa/tambah', $data);
33             $this->load->view('template/footer');
34         } else {
35             #code
36             echo "data berhasil ditambah";
37         }
38     }
39 }
```

- c. Setelah kita tambahkan *form validation*, maka langkah berikutnya menambahkan *rule*-nya

- https://www.codeigniter.com/user_guide/libraries/form_validation.html?highlight=form#the-form (searching pipe data)
- Untuk melihat jenis-jenis *rule* bisa dilihat pada **rule reference**
- **\$this->form_validation->set_rules('fieldname','fieldlabel','trim|required|min_length[5]|max_length[12]');**
- Jika kita lihat untuk menambahkan *rules* ada 3 parameter yaitu
 1. **fieldname** = nama dari htmlnya, semisal `<input type="text" class="form-control" id="nama" name="nama">` maka yang disebut dengan **fieldname** adalah **nama**
 2. **fieldlabel** = ini adalah alias digunakan untuk menampilkan kesalahannya (jika terjadi *error*)
 3. **Rule** yang digunakan
- d. Sebelum kita tambahkan *rule*-nya maka terlebih dahulu tambahkan **atribut name** pada **views/mahasiswa/tambah**

```

iv.col-md-6 > div.card > div.card-body > form > div.form-group > select#jurusan
4  .com/dosc/4.1/components/card/-->
5
6  der">
7  siswa
8
9  y">
10 method="post">
11 getbootstrap.com/dosc/4.1/components/card/-->
12 form-group">
13 or="nama">Nama </label>
14 ype="text" class="form-control" id="nama" name="nama">
15
16 form-group">
17 or="nim">Nim </label>
18 ype="number" class="form-control" id="nim" name="nim">
19
20 form-group">
21 or="email">Email </label>
22 ype="email" class="form-control" id="email" name="email">
23
24 form-group">
25 or="jurusan">Jurusan </label>
26 class="form-control" id="jurusan" name="jurusan" name="jurusan">
27 ion selected>-----</option>
28 ion value="Informatika">Informatika</option>
29 ion value="Kimia">Kimia</option>
30 ion value="Mesin">Mesin</option>
31 >
32
33 ="submit" name="submit" class="btn btn-primary float-right">Subm

```

- e. Setelah kita tambahkan **atribut name** maka tambahkan **rule** pada **controller/mahasiswa** seperti pada gambar dibawah ini (line 35-46)

application > controllers > mahasiswa.php > ...

```
25
26 public function tambah()
27 {
28     $data['title'] = 'Form Menambahkan Data Mahasiswa';
29     //$this->form_validation->set_rules('fieldname', 'fieldl
30     $this->form_validation->set_rules('nama', 'Nama', 'requir
31
32     if ($this->form_validation->run() == FALSE) {
33         #code
34         $this->load->view('template/header', $data);
35         $this->load->view('mahasiswa/tambah', $data);
36         $this->load->view('template/footer');
37     } else {
38         #code
39         echo "data berhasil ditambah";
40     }
41 }
42
```

f. Tambahkan pesan *error* pada **views/mahasiswa/tambah** (line 11)

```
application > views > mahasiswa > tambah.php > div.container > div.row.mt-3 > <div class="col-md-6">
1  <div class="container">
2  <div class="row mt-3">
3  <div class="col-md-6">
4  <!-- http://getbootstrap.com/docs/4.1/components/card/-->
5  <div class="card">
6  <div class="card-header">
7  <div class="card-body">
8  <div class="card-body">
9  <div class="card-body">
10 <div class="card-body">
11 <div class="card-body">
12 <div class="card-body">
13 <div class="card-body">
```

g. Hasilnya saat *field* nama tidak diisi maka akan muncul keterangan the Nama *Field* is required

Form Menamba x

Form Validation x

Pemrogramar

localhost/code-kedua/mahasiswa/tambah

CI

Home

Data Mahasiswa

About

Form Tambah Mahasiswa

The Nama field is required.

Nama

Nim

Email

Jurusan

----- ▾

Submit

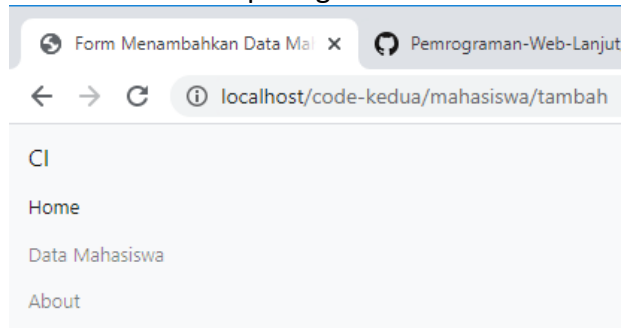
h. Tambahkan *alert* agar pesan *error* lebih menarik setelah *code validation error*

```

tambah.php > div.container > div.row.mt-3 > div.col-md-6 > div.card > div.
1   class="container">
2   <div class="row mt-3">
3     <div class="col-md-6">
4       <!-- http://getbootstrap.com/docs/4.1/components/card/ -->
5       <div class="card">
6         <div class="card-header">
7           Form Tambah Mahasiswa
8         </div>
9         <div class="card-body">
10          <?= validation_errors() ?>
11          <!--menampilkan pesan error-->
12          <?php if (validation_errors()) : ?>
13            <div class="alert alert-danger" role="alert">
14              <?= validation_errors() ?>
15            </div>
16          <?php endif ?>
17          <form action="" method="post">

```

i. Akan dihasilkan seperti gambar di bawah ini



Form Tambah Mahasiswa

The Nama field is required.

Nama

Nim

Email

Jurusan

Submit

- j. Setelah kita berhasil memberikan **rules validation** pada **nama**, maka tambahkan juga untuk **nim** dan **email**.

```
26
27 public function tambah()
28 {
29     $data['title'] = 'Form Menambahkan Data Mahasiswa';
30     //$this->form_validation->set_rules('fieldname','fieldlabel
31     $this->form_validation->set_rules('nama', 'Nama', 'required
32     $this->form_validation->set_rules('nim', 'Nim', 'required')
33     $this->form_validation->set_rules('email', 'Email', 'required
34
35     if ($this->form_validation->run() == FALSE) {
36         #code
```

- k. Hasilnya seperti berikut ini.

Form Tambah Mahasiswa

The Nama field is required.

The Nim field is required.

The Email field is required.

Nama

Nim

Email

Jurusan

▼

Submit

- l. Kita juga dapat mengecek ke *valid* an email dengan menambahkan rule pada email seperti pada gambar dibawah ini menggunakan **valid_email** (*line 37*)

```

27   tion tambah()
28
29   title'] = 'Form Menambahkan Data Mahasiswa';
30   ->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_le.
31   form_validation->set_rules('nama', 'Nama', 'required');
32   form_validation->set_rules('nim', 'Nim', 'required');
33   form_validation->set_rules(['email', 'Email', 'required|valid_email']);
34

```

m. Hasilnya sebagai berikut

Form Tambah Mahasiswa

Nama

Silvi Indah Novita Sari

Nim

1841720006

Email

silvi

Jurusan

Submit

Sertakan '@' pada alamat email. 'silvi' tidak memiliki '@'.

n. Selain itu kita juga dapat memberikan *rules* harus diisi dengan angka menggunakan **rules numeric**

```

public function tambah()
{
    $data['title'] = 'Form Menambahkan Data Mahasiswa';
    //$this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|m
    $this->form_validation->set_rules('nama', 'Nama', 'required');
    $this->form_validation->set_rules(['nim', 'Nim', 'required|numeric']);
    $this->form_validation->set_rules('email', 'Email', 'required|valid_email');
}

```

o. Akan dihasilkan alert seperti berikut ini.

	<div> <div>Form Tambah Mahasiswa</div> <div> <div>Nama</div> <div>Silvi Indah Novita Sari</div> </div> <div> <div>Nim</div> <div>184172</div> </div> <div> <div>Email</div> <div></div> </div> <div> <div>Jurusan</div> <div>----- ▼</div> </div> <div>Submit</div> </div>
--	--

Praktikum – Bagian 3: Menambahkan Data ke Database / Insert Data

1. Buka **controller/mahasiswa** dan tambahkan *code* berikut pada **function tambah** agar pada saat **form_validation** bernilai **true**, maka akan ditambahkan data dengan nama modelnya - > **model_mahasiswa** dengan nama *function* **tambahdatamhs()** dan jika sudah ditambah ke database maka akan *redirect* ke *controller* mahasiswa (line 46 - 47)

```

public function tambah()
{
    $data['title'] = 'Form Menambahkan Data Mahasiswa';
    //$this->form_validation->set_rules('fieldname', 'fieldlabel', 'required');
    $this->form_validation->set_rules('nama', 'Nama', 'required');
    $this->form_validation->set_rules('nim', 'Nim', 'required');
    $this->form_validation->set_rules('email', 'Email', 'required');

    if ($this->form_validation->run() == FALSE) {
        #code
        $this->load->view('template/header', $data);
        $this->load->view('mahasiswa/tambah', $data);
        $this->load->view('template/footer');
    } else {
        #code
        $this->mahasiswa_model->tambahdatamhs();
        redirect('mahasiswa', 'refresh');
    }
}

```

- 2 . Jika kita lihat *code* pada **controller mahasiswa**(line 46) terdapat **model** dengan nama **mahasiswa_model** dan *function*-nya **tambahdatamhs()**. Untuk itu kita harus membuat sebuah *function* dengan nama **tambahdatamhs()**

```
application > models > mahasiswa_model.php > mahasiswa_model > tambahdatamhs
1  <?php
2
3  defined('BASEPATH') or exit('No direct script access allowed');
4
5  class mahasiswa_model extends CI_Model
6  {
7
8      public function getAllmahasiswa()
9      {
10
11          $query = $this->db->get('mahasiswa');
12
13
14
15          return $query->result_array();
16          //atau bisa juga dengan menggunakan code berikut:
17          // return $this->db->get('mahasiswa')->result_array();
18      }
19      public function tambahdatamhs()
20      {
21          $data=[
22              "nama" => $this->input->post('nama',true),
23              "nim" => $this->input->post('nim',true),
24              "email"=>$this->input->post('email',true),
25              "jurusan"=>$this->input->post('jurusan',true)
26          ];
27
28          // $this->db->insert('Table', $object);
29          $this->db->insert('mahasiswa',$data);
30      }
31  }
```

- 3 . Buatlah sebuah *function* **tambahdatamhs()** pada **models/mahasiswa_model**

- https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=insert#inserting-data
- Tambahkan *code* seperti pada gambar dibawah ini

Keterangan:

- Line 22-27 adalah sebuah array dimana nantinya isi array adalah data-data yang akan dikirim (POST) ke database
- Line 23 "nama" => \$this->input->post('nama',true)

- nama->alias
- Parameter nama pada post adalah name pada html seperti pada gambar dibawah ini
- Parameter *true* digunakan untuk mengaktifkan htmlspecialchars

Line 29 adalah proses *insert* data ke database dengan **2 parameter** yaitu **nama tablenya** dan **objectnya** (dalam hal ini data yang akan dimasukkan ke database)

4 Langkah berikutnya kita coba untuk tambahkan data.

· Jika berhasil maka data akan tampil di halaman **index.php**

a. `redirect('mahasiswa','refresh');`(*line 47* gambar *controller* mahasiswa)

b. Hasil

CI

Form Tambah Mahasiswa

Nama

dino

Nim

18652353

Email

dino@gmail.com

Jurusan

Kimia ▼

Submit

