

LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT

Read dan Input Database serta Form Validation



Oleh :

Rahmad Alfian Maskuri / NIM : 1841720048 / TI – 2A

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**



Topik

Menampilkan dan Menambahkan Data ke *Database* menggunakan CodeIgniter serta Menambahkan Form Validation

Tujuan

Mahasiswa diharapkan dapat:

1. Memahami cara menampilkan data dari *database* menggunakan CodeIgniter
2. Memahami cara menambahkan *form validation* menggunakan CodeIgniter
3. Memahami cara menambahkan data ke *database* menggunakan CodeIgniter

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 (<i>line 9</i>)</p> <pre>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 *code* berikut agar database dapat digunakan banyak *function* (line 7-12)

```

mahasiswa.php x Home.php index.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 parent(ci_controller)
10         parent::__construct();
11         $this->load->database();
12     }
13
14
15     public function index()
16     {
17         // modul untuk load database
18         // $this->load->database();
19         $data['title']='List Mahasiswa';
20         $this->load->view('template/header',$data);
21         $this->load->view('mahasiswa/index');
22         $this->load->view('template/footer');
23     }
24 }
25
26
27
28 /* End of file Controllername.php */
29 ?>

```

```

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

```

Keterangan:

- Line 7 – 12 adalah sebuah *function* construct (**PERHATIKAN ITU MENGGUNAKAN DOUBLE UNDERSCORE `__construct` (tanpa spasi)**)

	<ul style="list-style-type: none"> - <i>Function</i> construct dijalankan pada saat classnya diinstansiasi - <i>Line</i> 10 digunakan untuk menjalankan fungsi construct pada class parent dimana class parent ini adalah CI_Controller - <i>Line</i> 11 digunakan agar untuk mengload database (Semua fungsi di class mahasiswa dapat menggunakannya) - Jika kita sudah melakukan load database pada saat <i>function</i> construct maka pada <i>function</i> index kita comment agar tidak bekerja dua kali
4	Selain 2 cara diatas kita juga dapat melakukan proses <i>load</i> database pada application/config/autoload dan tambahkan pada database pada autoload['libraries']=array('database') seperti pada gambar

```

mahasiswa.php  autoload.php x  index.php
48 | These are the classes located in system/libraries/ or your
49 | application/libraries/ directory, with the addition of the
50 | 'database' library, which is somewhat of a special case.
51 |
52 | Prototype:
53 |
54 | $autoload['libraries'] = array('database', 'email', 'session');
55 |
56 | You can also supply an alternative library name to be assigned
57 | in the controller:
58 |
59 | $autoload['libraries'] = array('user_agent' => 'ua');
60 */
61 $autoload['libraries'] = array('database');
62
63 /*
64

```

➤ Jawaban :

```

-----
Auto-load Libraries
-----

These are the classes located in system/libraries/ or your
application/libraries/ directory, with the addition of the
'database' library, which is somewhat of a special case.

Prototype:

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

You can also supply an alternative library name to be assigned
in the controller:

    $autoload['libraries'] = array('user_agent' => 'ua');
*/
$autoload['libraries'] = array('database');

```

5

Untuk kasus ini kita gunakan cara yang terakhir (no 4) karena kita asumsikan bahwa semua *controller* membutuhkan database. Maka kita *comment code* function construct yang ada di *controller* mahasiswa

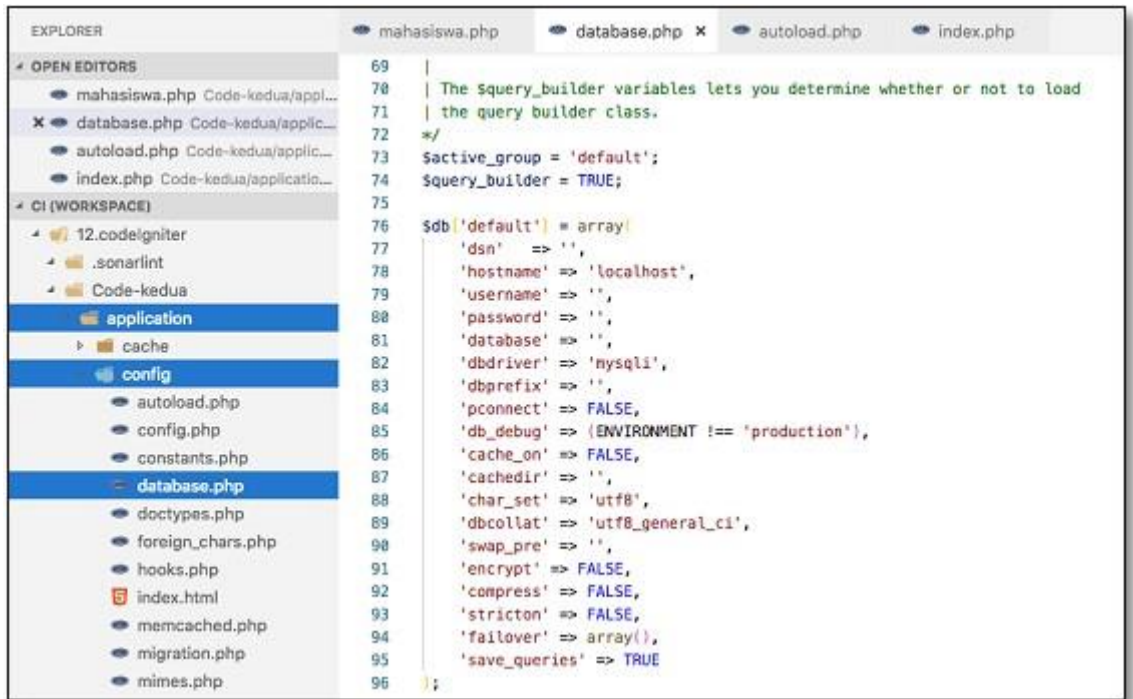
```

mahasiswa.php x  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      //fungsi yang akan dijalankan saat classnya diinstansiasi
7      // public function __construct()
8      // {
9          //     //digunakan untuk menjalankan fungsi construct pada class parent(ci_controller)
10         //     parent::__construct();
11         //     $this->load->database();
12         // }

```

```
code-kedua > application > controllers > mahasiswa.php
1  p
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class mahasiswa extends CI_Controller {
6
7      //fungsi yang akan dijalankan saat classnya diinstansiasi
8      //public function __construct()
9      //{
10         //digunakan menjalankan fungsi construct pada class parent(ci_contro
11         //parent::__construct();
12         //$this->load->database();
13     }
```

- 6 Setelah itu kita akan mengatur konfigurasi database. Untuk mengatur konfigurasi database maka buka folder **application/config/database.php**



```
EXPLORER
mahasiswa.php database.php x autoload.php index.php
OPEN EDITORS
mahasiswa.php Code-kedua/appl...
x database.php Code-kedua/appl...
autoload.php Code-kedua/appl...
index.php Code-kedua/appl...
CI (WORKSPACE)
12.codeigniter
.sonarlint
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
69 |
70 | The $query_builder variables 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' => '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 );
```

Kita set konfigurasinya seperti pada gambar dibawah ini

```

76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'kuliah_CI',
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 );

```

```

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

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => 'mysql',
    '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

➤ Jawaban :

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

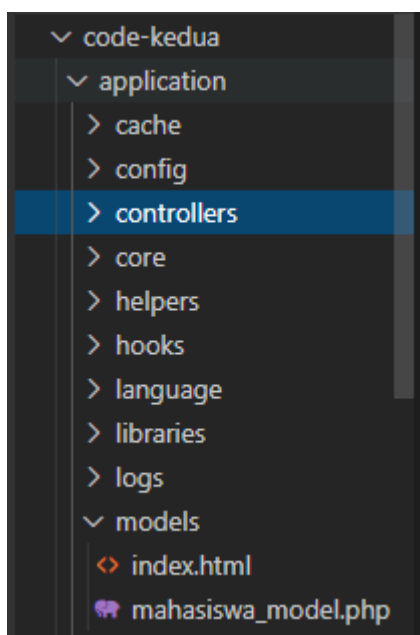
➤ Jawaban :

+ 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
<input type="checkbox"/> Check all With selected: Edit Copy Delete Export						

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**



b. Tambahkan *function* getAllmahasiswa pada mahasiswa_model.php

```

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          // https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=query%20builder
10         // $query digunakan untuk menampung isi dari tabel mahasiswa
11         $query=$this->db->get('mahasiswa');
12
13         // https://www.codeigniter.com/user_guide/database/results.html
14         // untuk menampilkan isi dari query
15         return $query->result_array();
16
17     }
18
19 }
20
21 /* End of file mahasiswa_model.php */
22
23 ?>

```

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



```
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 ?>
```

➤ Jawaban :



```
1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class mahasiswa_model extends CI_Controller {
6
7     public function getAllmahasiswa()
8     {
9         // - https://www.codeigniter.com/user_guide/database/query_builder.ht
10         //$query digunakan untuk menampung isi dari tabel mahasiswa
11         //$query=$this->db->get('mahasiswa');
12
13         // - https://www.codeigniter.com/user_guide/database/results.html
14         // untuk menampilkan isi dari query
15         //return $query->result_array();
16
17         //atau bisa juga menggunakan code berikut
18         return $this->db->get('mahasiswa')->result_array();
19     }
20
21 }
22
23
24 /* End of file mahasiswa_model.php */
25
26 ?>
```

11

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

```

14
15     public function index()
16     {
17         $this->load->model('mahasiswa_model');
18
19         // modul untuk load database
20         // $this->load->database();
21         $data['title']='List Mahasiswa';
22         $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
23         $this->load->view('template/header',$data);
24         $this->load->view('mahasiswa/index',$data);
25         $this->load->view('template/footer');
26     }
27
28

```

➤ Jawaban :

```

public function index()
{
    //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');
}

```

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         // modul untuk load database
11         // $this->load->database();
12         $data['title']='List Mahasiswa';
13         $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
14         $this->load->view('template/header',$data);
15         $this->load->view('mahasiswa/index',$data);
16         $this->load->view('template/footer');
17     }
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.php  mahasiswa_model.php  index.php x
1  <?php
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 in</li>
12                 <li class="list-group-item">Morbi leo risus</li>
13                 <li class="list-group-item">Porta ac consectetur ac</li>
14                 <li class="list-group-item">Vestibulum at eros</li>
15             </ul>
16          </div>
17      </div>
18  </div>

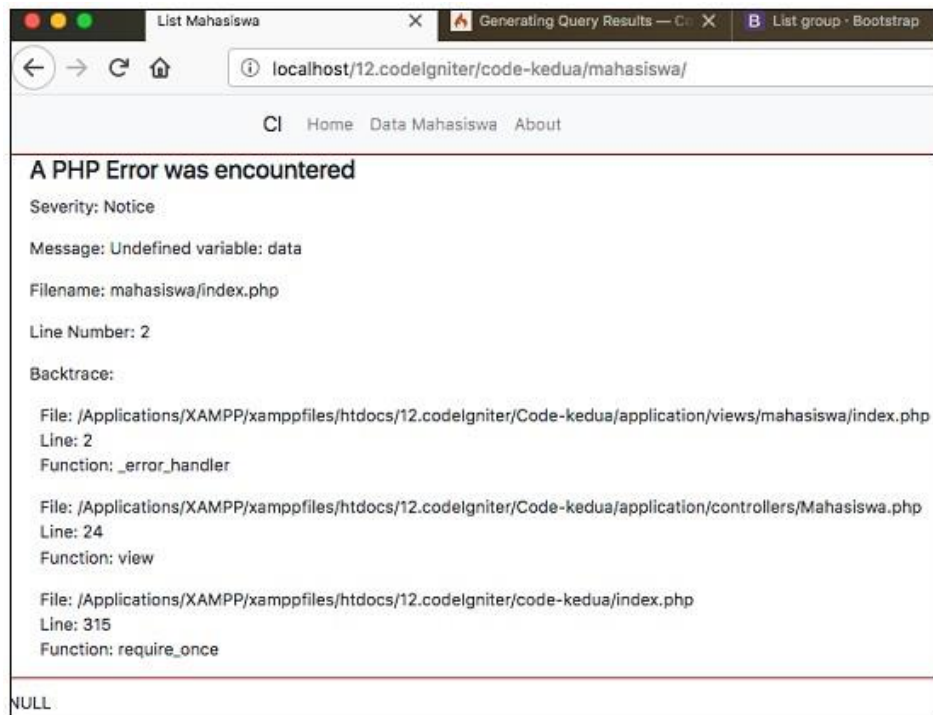
```

code-kedua > application > views > mahasiswa > index.php

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



A PHP Error was encountered

Severity: Notice

Message: Undefined property: mahasiswa::\$mahasiswa_model

Filename: controllers/mahasiswa.php

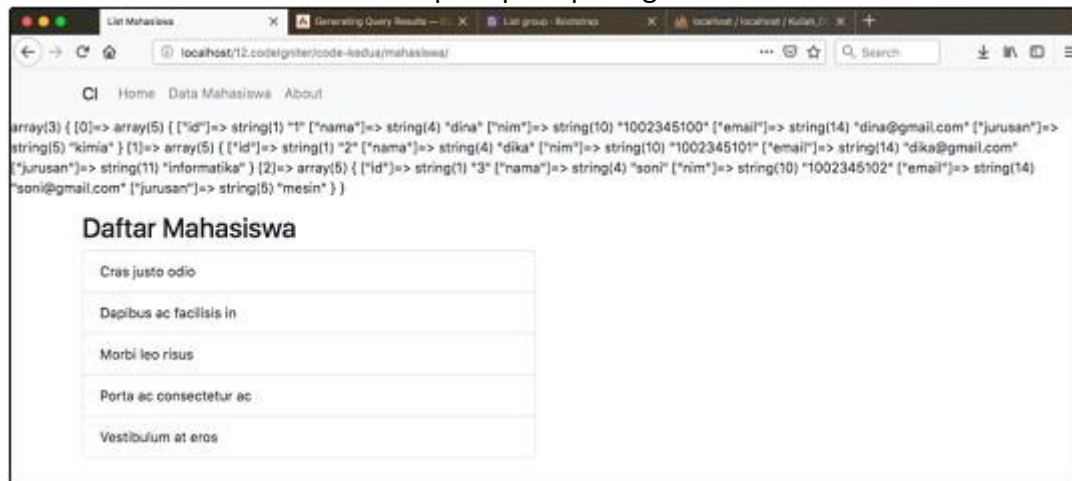
Line Number: 19

Backtrace:

File: C:\Program Files\Ampps\www\12.codeIgniter\code-kedua\application\controllers\mahasiswa.php
Line: 19
Function: _error_handler

File: C:\Program Files\Ampps\www\12.codeIgniter\code-kedua\index.php
Line: 315
Function: require_once

Hal ini disebabkan karena tidak ketahu 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*.



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**



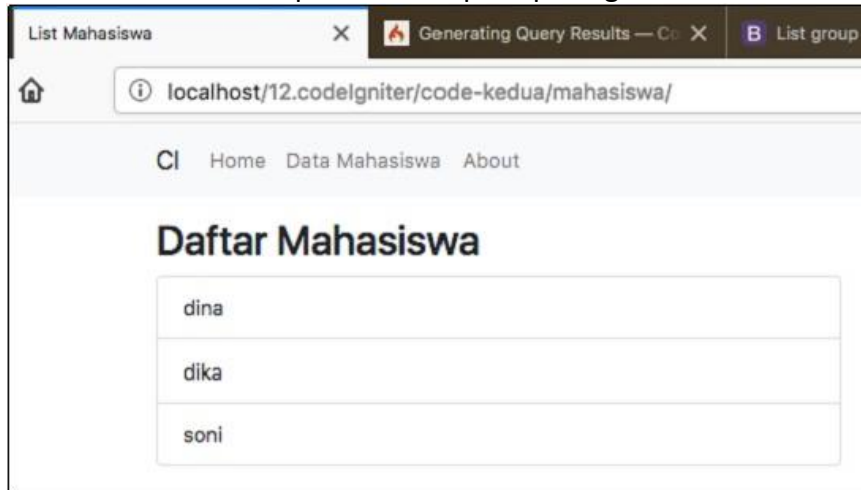
```

1
2 <div class="container">
3   <!-- mt-3 artinya margin top 3 -->
4   <div class="row mt-3">
5     <div class="col-md-6">
6       <h2>Daftar Mahasiswa</h2>
7       <ul class="list-group">
8         <?php foreach($mahasiswa as $mhs):?>
9           <li class="list-group-item"><?= $mhs['nama'];?></li>
10        <?php endforeach; ?>
11      </ul>
12    </div>
13  </div>
14 </div>

```

15

Jika berhasil maka tampilan akan seperti pada gambar dibawah ini



16

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:

a. Tambahkan **code** berikut pada **views/template/header (line 26,27,28)**

```

● autoload.php ● header.php x ● index.php
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8 <!-- Bootstrap CSS -->
9 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
10
11 <!-- <title>Hello, world!</title> -->
12 <title><?= $title ?></title>
13
14 </head>
15 <body>
16
17 <!-- navbar -->
18 <nav class="navbar navbar-expand-lg navbar-light bg-light">
19 | <div class="container">
20 | <a class="navbar-brand" href="#">CI</a>
21 | <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-label="Toggle navigation">
22 | <span class="navbar-toggler-icon"></span>
23 | </button>
24 | <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
25 | | <div class="navbar-nav">
26 | | | <a class="nav-item nav-link" href="<?= base_url(); ?>">Home <span class="sr-only">|current|</span></a>
27 | | | <a class="nav-item nav-link" href="<?= base_url(); ?>mahasiswa">Data Mahasiswa</a>
28 | | | <a class="nav-item nav-link" href="<?= base_url(); ?>">About</a>
29 | | | <!-- <a class="nav-item nav-link disabled" href="#">Disabled</a> -->
30 | | </div>
31 | </div>
32 </div>
33 </nav>

```

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')`

EXPLORER	● autoload.php x ● header.php ● index.php
OPEN EDITORS x ● autoload.php Code-kedua/applic... ● header.php Code-kedua/applicati... ● index.php Code-kedua/applicatio...	80 81 */ 82 \$autoload['drivers'] = array(); 83 84 /* 85 ----- 86 Auto-load Helper Files 87 ----- 88 Prototype: 89 90 \$autoload['helper'] = array('url', 'file'); 91 */ 92 \$autoload['helper'] = array('url'); 93
CI (WORKSPACE) P cache ▲ config ● autoload.php ● config.php ● constants.php ● database.php ● doctypes.php	

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

Praktikum – Bagian 2: Membuat Form Input dan Menambahkan Form Validation

Langkah	Keterangan
1	<p>Buatlah sebuah <i>button</i> tambah data pada views/mahasiswa/index.php (line 3-7). Tambahkan <i>code</i> pada line 3-7 untuk membuat sebuah <i>button</i></p> <pre> index.php x config.php 1 2 <div class="container"> 3 <div class="row mt-4"> 4 <div class="col-md-6"> 5 Tambah Data 6 </div> 7 </div> 8 <!-- mt-3 artinya margin top 3 --> 9 <div class="row mt-3"> 10 <div class="col-md-6"> 11 <h2>Daftar Mahasiswa</h2> 12 <ul class="list-group"> 13 <?php foreach(\$mahasiswa as \$mhs):?> 14 <li class="list-group-item"><?= \$mhs['nama'];?> 15 <?php endforeach; ?> 16 17 </div> 18 </div> 19 </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> mahasiswa.php x 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 constrauct pada class parrent(ci_controller) 10 parent::__construct(); 11 \$this->load->model('mahasiswa_model'); 12 } 13 14 15 public function index() 16 { 17 18 // modul untuk load database 19 // \$this->load->database(); 20 \$data['title']='List Mahasiswa'; 21 \$data['mahasiswa']=\$this->mahasiswa_model->getAllmahasiswa(); 22 \$this->load->view('template/header',\$data); 23 \$this->load->view('mahasiswa/index',\$data); 24 \$this->load->view('template/footer'); 25 } 26 27 28 } 29 30 /* End of file Controllername.php */ 31 ?> </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**)

```

mahasiswa.php x
26     }
27
28     public function tambah() {
29
30         $data['title'] = 'Form Menambahkan Data Mahasiswa';
31         $this->load->view('template/header', $data);
32         $this->load->view('mahasiswa/tambah', $data);
33         $this->load->view('template/footer');
34     }
35
36 }
37
38
39 /* End of file Controllername.php */
40 ?>

```

Keterangan :

- Pada *line* 32, `$this->load->view('mahasiswa/tambah')` artinya bahwa **function** tambah akan melakukan *load view* pada folder mahasiswa, dengan nama filenya adalah tambah

4

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

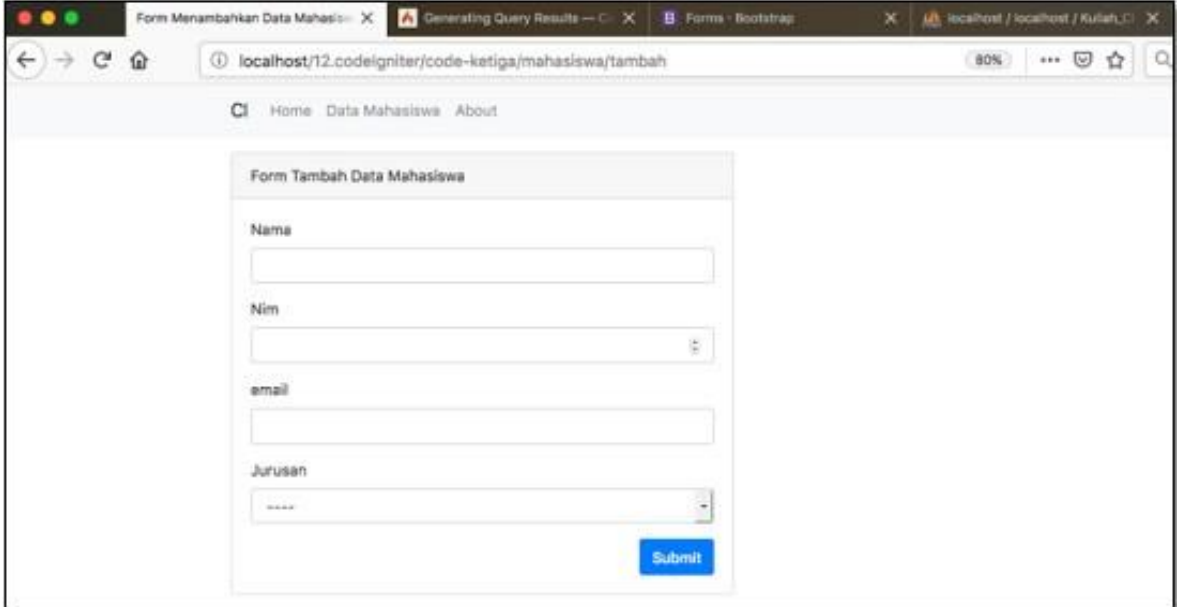
```

mahasiswa.php x tambah.php x
1  <div class="container">
2      <div class="row mt-3">
3          <div class="col-md-6">
4              <!-- https://getbootstrap.com/docs/4.1/components/card/ -->
5              <div class="card">
6                  <div class="card-header">
7                      Form Tambah Data Mahasiswa
8                  </div>
9                  <div class="card-body">
10                     <form action="" method="post">
11                         <!-- https://getbootstrap.com/docs/4.1/components/forms/ -->
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>

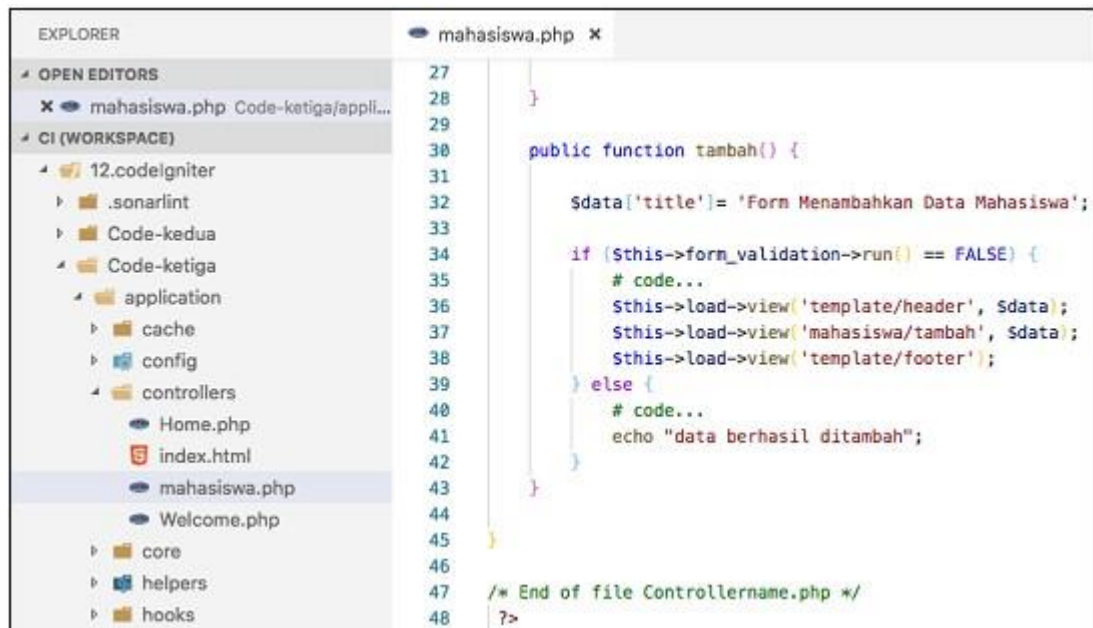
```

Keterangan :

- Pada *line* 33, terdapat atribut class dengan *value* btn btn-primary float-right yang artinya kita membuat sebuah **button** dan posisi buttonnya ada di kanan

	<p>- Ganti value dari <i>line</i> 28-31 menjadi seperti di bawah ini (Agar di tabel tidak tersimpan value berupa angka)</p> <pre><option value="informatika">Informatika</option> <option value="kimia">kimia</option> <option value="mesin">Mesin</option></pre>
5	<p>Maka akan dihasilkan halaman seperti di bawah ini.</p> 
6	<p>Setelah berhasil membuat <i>form</i> tambah data, selanjutnya buka file index pada views/mahasiswa/index.php dan kita atur href nya</p> <pre> 2 <div class="container"> 3 <div class="row mt-4"> 4 <div class="col-md-6"> 5 <a href="<?= base_url();?>mahasiswa/tambah" class="btn btn-primary"> Tambah Data 6 </div> 7 </div> 8 <!-- mt-3 artinya margin top 3 --> 9 <div class="row mt-3"> 10 <div class="col-md-6"> 11 <h2>Daftar Mahasiswa</h2> 12 <ul class="list-group"> 13 <?php foreach(\$mahasiswa as \$mhs):?> 14 <li class="list-group-item"> 15 <?= \$mhs['nama'];?> 16 17 <?php endforeach; ?> 18 19 </div> 20 </div> 21 </div></pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Tambahkan <i>code</i> pada line 5 agar pada saat <i>user</i> melakukan <i>event click</i> maka <i>user</i> akan ditampilkan halaman tambah.
7	<p>Tambahkan validasi menggunakan <i>form_validation</i>. Contoh validasi <i>form</i> adalah seperti <i>field</i> nama harus diisi, nim harus angka dsb. Untuk lebih jelasnya ikuti langkah berikut ini :</p> <p>a. Tambahkan <i>form_validation</i></p> <p>https://www.codeigniter.com/user_guide/libraries/form_validation.html?highlight=form#the-form (<i>searching controller</i>)</p>

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



```
27
28
29
30 public function tambah() {
31
32     $data['title'] = 'Form Menambahkan Data Mahasiswa';
33
34     if ($this->form_validation->run() == FALSE) {
35         # code...
36         $this->load->view('template/header', $data);
37         $this->load->view('mahasiswa/tambah', $data);
38         $this->load->view('template/footer');
39     } else {
40         # code...
41         echo "data berhasil ditambah";
42     }
43 }
44
45
46
47 /* End of file Controllername.php */
48 ?>
```

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**

```
<div class="form-group">
    <label for="nama">Nama</label>
    <input type="text" class="form-control" id="nama" name="nama">
</div>
<div class="form-group">
    <label for="nim">Nim</label>
    <input type="number" class="form-control" id="nim" name="nim">
</div>
<div class="form-group">
    <label for="email">email</label>
    <input type="email" class="form-control" id="email" name="email">
</div>
<div class="form-group">
    <label for="jurusan">Jurusan</label>
    <select class="form-control" id="jurusan" name="jurusan">
        <option selected>----</option>
        <option value="1">Kimia</option>
        <option value="2">Informatika</option>
        <option value="3">Mesin</option>
    </select>
</div>
```

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

```

30 public function tambah() {
31
32     $data['title'] = 'Form Menambahkan Data Mahasiswa';
33
34     // $this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_length[5]|max_length[12]');
35     $this->form_validation->set_rules('nama', 'Nama', 'required');
36
37     if ($this->form_validation->run() == FALSE) {
38         # code...
39         $this->load->view('template/header', $data);
40         $this->load->view('mahasiswa/tambah', $data);
41         $this->load->view('template/footer');
42     } else {
43         # code...
44         echo "data berhasil ditambah";
45     }
46 }

```

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

```

10
11 <div class="card-body">
12     <? validation_errors() ?>
13     <form action="" method="post">
14         <!-- https://getbootstrap.com/docs/4.1/components/forms/ -->
15         <div class="form-group">
16             <label for="nama">Nama</label>
17             <input type="text" class="form-control" id="nama" name="nama">

```

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

- h. Tambahkan *alert* agar pesan *error* lebih menarik setelah *code validation error*
- <https://getbootstrap.com/docs/4.1/components/alerts/>


```

<div class="card-body">
  <!-- untuk menampilkan pesan error -->
  <?php if (validation_errors()):?>
  <div class="alert alert-danger" role="alert">
    <? validation_errors() ?>
  </div>
  <?php endif ?>
  <form action="" method="post">
    <!-- https://getbootstrap.com/docs/4.1/components/forms/ -->
    <div class="form-group">
      <label for="nama">Nama</label>
      <input type="text" class="form-control" id="nama" name="nama">
    </div>

```

- Tambahkan tanda ; pada akhir **validation_errors()** sehingga menjadi seperti berikut
 <? validation_errors(); ?>

i. Akan dihasilkan seperti gambar di bawah ini

The screenshot shows a web form titled "Form Tambah Data Mahasiswa". At the top, there is a red alert box with the message "The Nama field is required." Below the alert, the form contains four input fields: "Nama" (a text input), "Nim" (a text input with a clear button), "email" (a text input), and "Jurusan" (a dropdown menu). A blue "Submit" button is located at the bottom right of the form.

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

```

30     public function tambah() {
31
32         $data['title'] = 'Form Menambahkan Data Mahasiswa';
33
34         // $this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_length[5]|max_length[12]');
35         $this->form_validation->set_rules('nama', 'Nama', 'required');
36         $this->form_validation->set_rules('nim', 'Nim', 'required');
37         $this->form_validation->set_rules('email', 'Email', 'required');
38
39         if ($this->form_validation->run() == FALSE) {
40             # code...
41             $this->load->view('template/header', $data);
42             $this->load->view('mahasiswa/tambah', $data);
43             $this->load->view('template/footer');
44         } else {
45             # code...
46             echo "data berhasil ditambah";
47         }
48     }
49
50
51
52     /* End of file Controllername.php */
53     ?>

```

k. Hasilnya seperti berikut ini.

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

```

34     // $this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_length[5]|max_length[12]');
35     $this->form_validation->set_rules('nama', 'Nama', 'required');
36     $this->form_validation->set_rules('nim', 'Nim', 'required');
37     $this->form_validation->set_rules('email', 'Email', 'required|valid_email');

```

m. Hasilnya sebagai berikut.

Form Tambah Data Mahasiswa

Nama
dina

Nim
12344512313

email
dina

Please enter an email address.

Submit

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

```

34 // $this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_length(5)|max_length(12)');
35     $this->form_validation->set_rules('nama', 'Nama', 'required');
36     $this->form_validation->set_rules('nim', 'Nim', 'required|numeric');
37     $this->form_validation->set_rules('email', 'Email', 'required|valid_email');

```

- o. Akan dihasilkan alert seperti berikut ini.

Form Tambah Data Mahasiswa

Nama
dina

Nim
122a

Please enter a number.

Jurusan

Submit

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

Setelah kita berhasil membuat **form_validation** langkah berikutnya adalah memasukkan data ke database (**insert database**)

Langkah	Keterangan
1	<p>Buka controller/mahasiswa dan tambahkan <i>code</i> berikut pada function tambah agar pada saat form_validation bernilai true, maka akan ditambahkan data dengan nama modelnya - > model_mahasiswa dengan nama <i>function</i> tambahdatamhs() dan jika sudah ditambah ke database maka akan <i>redirect</i> ke <i>controller</i> mahasiswa (<i>line 46 - 47</i>)</p> <pre>30 public function tambah() { 31 32 \$data['title'] = 'Form Menambahkan Data Mahasiswa'; 33 34 // \$this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim required min_length[5] max_length[12] 35 \$this->form_validation->set_rules('nama', 'Nama', 'required'); 36 \$this->form_validation->set_rules('nim', 'Nim', 'required numeric'); 37 \$this->form_validation->set_rules('email', 'Email', 'required valid_email'); 38 39 if (\$this->form_validation->run() == FALSE) { 40 # code... 41 \$this->load->view('template/header', \$data); 42 \$this->load->view('mahasiswa/tambah', \$data); 43 \$this->load->view('template/footer'); 44 } else { 45 # code... 46 \$this->mahasiswa_model->tambahdatamhs(); 47 redirect('mahasiswa','refresh'); 48 } 49 } 50 }</pre>
2	<p>Jika kita lihat <i>code</i> pada controller mahasiswa(<i>line 46</i>) terdapat model dengan nama mahasiswa_model dan <i>function</i>-nya tambahdatamhs(). Untuk itu kita harus membuat sebuah <i>function</i> dengan nama tambahdatamhs()</p>
3	<p>Buatlah sebuah <i>function</i> tambahdatamhs() pada models/mahasiswa_model</p> <ul style="list-style-type: none">• https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=insert#inserting-data• Tambahkan <i>code</i> seperti pada gambar dibawah ini

```

5 class mahasiswa_model extends CI_Model {
6
7     public function getAllmahasiswa()
8     {
9         // https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=query%20builder
10        // $query digunakan untuk menampung isi dari tabel mahasiswa
11        $query=$this->db->get('mahasiswa');
12
13        // https://www.codeigniter.com/user_guide/database/results.html
14        // untuk menampilkan isi dari query
15        return $query->result_array();
16
17        // atau bisa juga menggunakan code berikut
18        // return $this->db->get('mahasiswa')->result_array();
19    }
20
21    public function tambahdatamhs() {
22        $data=[
23            "nama" => $this->input->post('nama',true), // ini sama dengan htmlspecialchars($POST['nama'])
24            "nim" => $this->input->post('nim',true),
25            "email" => $this->input->post('email',true),
26            "jurusan" => $this->input->post('jurusan',true)
27        ];
28        // $this->db->insert('Table', $object);
29        $this->db->insert('mahasiswa', $data);
30    }
31
32
33
34

```

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

```

$data=[
    "nama" => $this->input->post('nama',true),
    "nim" => $this->input->post('nim',true),
    "email" => $this->input->post('email',true),
    "jurusan" => $this->input->post('jurusan',true)
];

```

```

<div class="form-group">
    <label for="nama">Nama</label>
    <input type="text" class="form-control" id="nama" name="nama">
</div>

```

- 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

Form Tambah Data Mahasiswa

Nama

ani

Nim

12349888

email

ani@gmail.com

Jurusan

Mesin

Submit

Tambah Data

Daftar Mahasiswa

dina

dika

soni

dino

ani

-- Selamat Mengerjakan --