

**LAPORAN PRAKTIKUM  
PEMROGRAMAN WEB LANJUT**

Jobsheet-2: Read dan Input Database serta Form Validation



Disusun oleh:  
**Livia Yurike Khuril Maula**  
D4 TI – 2A / 15  
NIM : 1841720025

**POLITEKNIK NEGERI MALANG  
JURUSAN TEKNOLOGI INFORMASI  
PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**FEBRUARI 2020**



Jurusan Teknologi Informasi Politeknik Negeri Malang  
**Jobsheet-2: Read dan Input Database serta Form Validation**  
**Mata Kuliah Pemrograman Web Lanjut**  
**Pengampu:** Tim Ajar Pemrograman Web Lanjut  
*Februari 2020*

---

### 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>mahasiswa.php × application &gt; controllers &gt; mahasiswa.php 1  &lt;?php 2 3  defined('BASEPATH') OR exit('No direct script access allowed'); 4 5  class mahasiswa extends CI_Controller { 6 7      public function index() 8      { 9          // modul untuk load database 10         \$this-&gt;load-&gt;database(); 11         \$data['title']='List Mahasiswa'; 12         \$this-&gt;load-&gt;view('template/header',\$data); 13         \$this-&gt;load-&gt;view('mahasiswa/index'); 14         \$this-&gt;load-&gt;view('template/footer'); 15     } 16 17 18 } 19 20 /* End of file Controllername.php */ 21 22 ?&gt;</pre> <p>Keterangan :</p> <ul style="list-style-type: none"><li>- 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>.</li></ul> |

3

Tambahkan *code* berikut agar database dapat digunakan banyak *function* (line 7-12)

```

mahasiswa.php
application > controllers > mahasiswa.php
1  <?php
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 untuk menjalankan fungsi construct pada class parent(ci_controller)
11         parent::__construct();
12         $this->load->database();
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
30 ?>

```

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 mengload 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

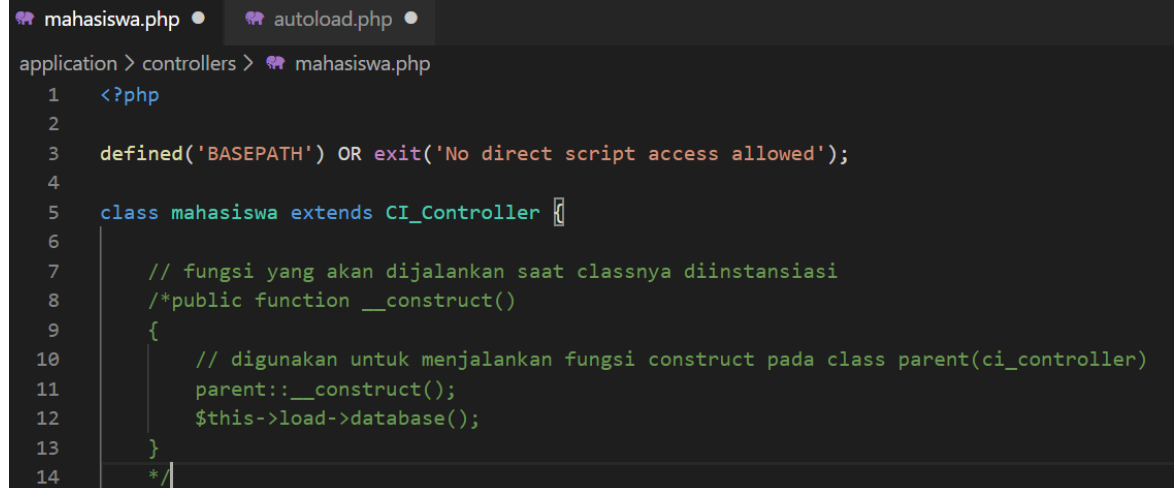
```

mahasiswa.php • autoload.php
application > config > autoload.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

```

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



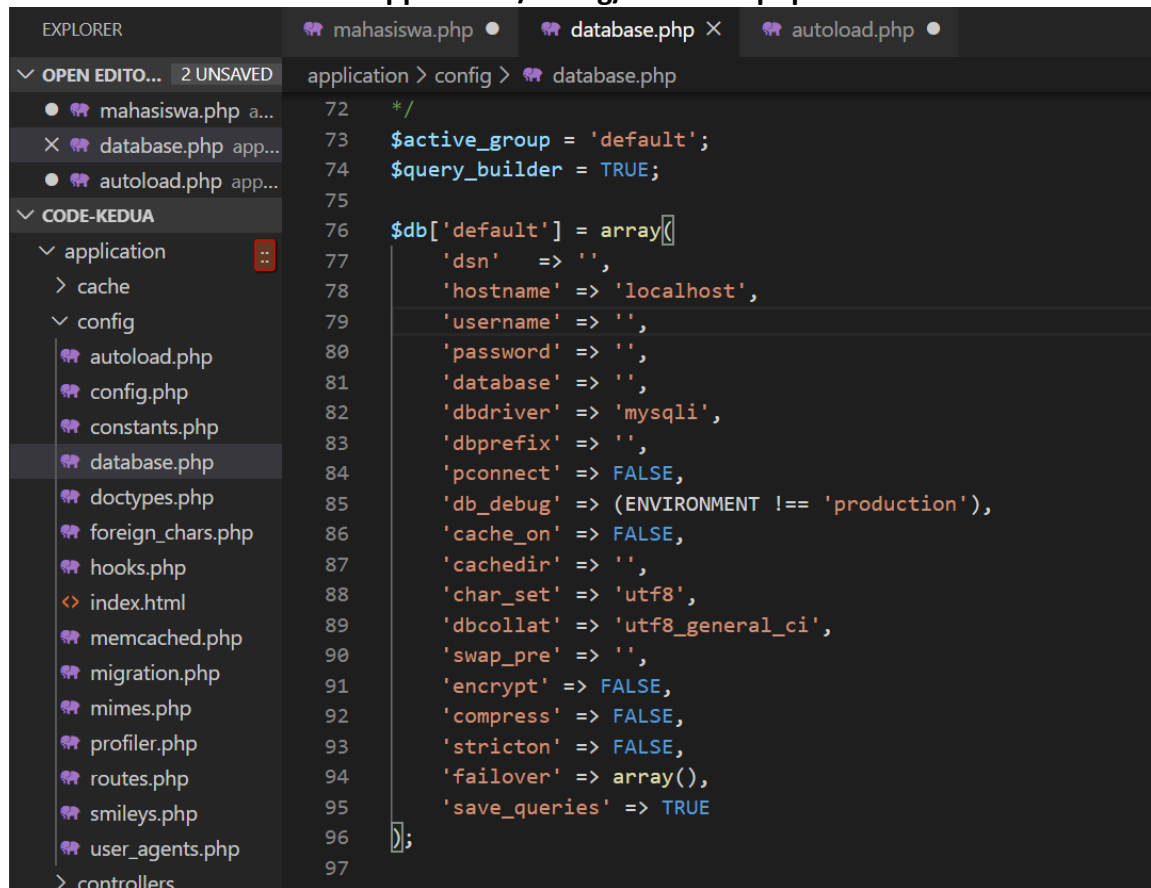
```

mahasiswa.php • autoload.php •
application > controllers > mahasiswa.php
1  <?php
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 untuk menjalankan fungsi construct pada class parent(ci_controller)
11         parent::__construct();
12         $this->load->database();
13     }
14     */

```

6

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



```

EXPLORER
mahasiswa.php • database.php ✕ autoload.php •
OPEN EDITOR... 2 UNSAVED
mahasiswa.php a...
database.php app...
autoload.php app...
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
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  );
97

```

7

Kita set konfigurasinya seperti pada gambar dibawah ini

```

73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => 'mysql',
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 );
97

```

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

| # | Nama    | Jenis        | Penyortiran        | Atribut | Tak Ternilai | Bawaan | Komentar | Ekstra         | Tindakan           |
|---|---------|--------------|--------------------|---------|--------------|--------|----------|----------------|--------------------|
| 1 | id      | int(30)      |                    | Tidak   | Tidak ada    |        |          | AUTO_INCREMENT | Ubah Hapus Lainnya |
| 2 | nama    | varchar(100) | utf8mb4_general_ci | Tidak   | Tidak ada    |        |          |                | Ubah Hapus Lainnya |
| 3 | nim     | varchar(50)  | utf8mb4_general_ci | Tidak   | Tidak ada    |        |          |                | Ubah Hapus Lainnya |
| 4 | email   | varchar(50)  | utf8mb4_general_ci | Tidak   | Tidak ada    |        |          |                | Ubah Hapus Lainnya |
| 5 | jurusan | varchar(50)  | utf8mb4_general_ci | Tidak   | Tidak ada    |        |          |                | Ubah Hapus Lainnya |

9

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

+ Opsi

|                  | id | nama                      | nim        | email                  | jurusan            |
|------------------|----|---------------------------|------------|------------------------|--------------------|
| Ubah Salin Hapus | 1  | Livia Yurike Khuril Maula | 1841720025 | kliviayurike@gmail.com | Teknik Informatika |
| Ubah Salin Hapus | 2  | Darin Zahira              | 1841720033 | darinzahira@gmail.com  | Teknik Informatika |
| Ubah Salin Hapus | 3  | Inayati Machsus           | 1841720202 | inayati@gmail.com      | Teknik Informatika |

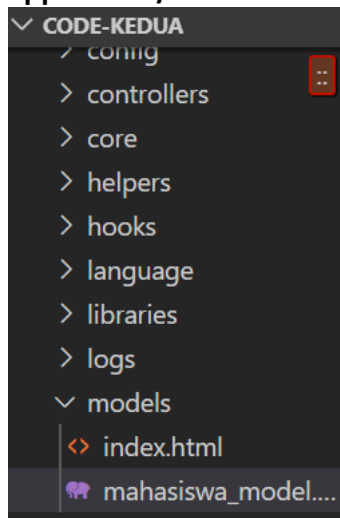
Pilih Semua Dengan pilihan: Ubah Salin Hapus Ekspor

10

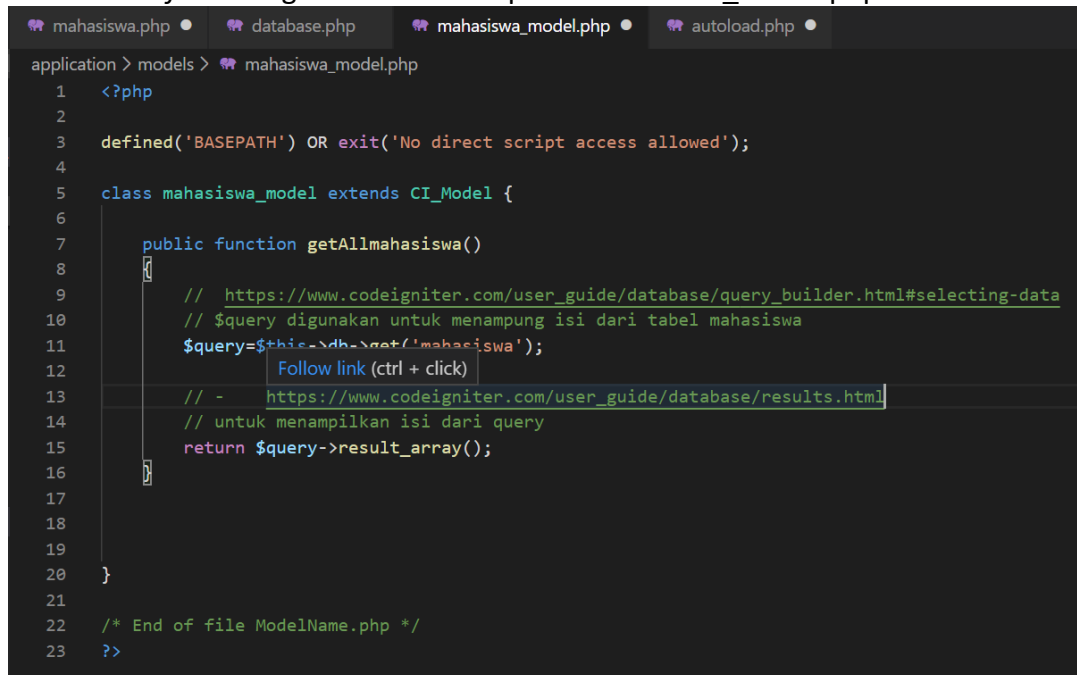
Setelah kita membuat sebuah *database*, maka langkah berikutnya adalah membuat sebuah model untuk melakukan proses *database*. Di dalam model ini kita akan membuat

sebuah *function* untuk melakukan perintah ambil data dari *database*. Untuk lebih jelasnya ikuti langkah-langkah berikut:

- a. Buatlah sebuah file model dengan nama **mahasiswa\_model** pada folder **application/models**



- b. Tambahkan *function* `getAllmahasiswa` pada `mahasiswa_model.php`

A screenshot of a code editor with a dark theme. The file 'mahasiswa\_model.php' is open. The code defines a class 'mahasiswa\_model' extending 'CI\_Model'. It includes a public function 'getAllmahasiswa()' which uses the CodeIgniter Query Builder to fetch all data from the 'mahasiswa' table and returns the result as an array. Comments in Indonesian explain the steps: connecting to the database, selecting data, and displaying the results. Line numbers 1 through 23 are visible on the left.

```
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#selecting-data
10        // $query digunakan untuk menampung isi dari tabel mahasiswa
11        $query=$this->db->get('mahasiswa');
12        // - https://www.codeigniter.com/user_guide/database/results.html
13        // untuk menampilkan isi dari query
14        return $query->result_array();
15    }
16
17
18
19
20 }
21
22 /* End of file ModelName.php */
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](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)
- [https://www.codeigniter.com/user\\_guide/database/results.html#result-arrays](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.php database.php mahasiswa_model.php autoload.p
application > models > mahasiswa_model.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          // atau bisa juga menggunakan code berikut
10         return $this->db->get('mahasiswa')->result_array();
11     }
12 }
13
14 /* End of file ModelName.php */
15 ?>

```

11

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

```

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
29  }
30
31  /* End of file Controllername.php */
32
33  ?>

```

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 • index.php • database.php • mahasiswa_model.php
application > views > mahasiswa > index.php
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>

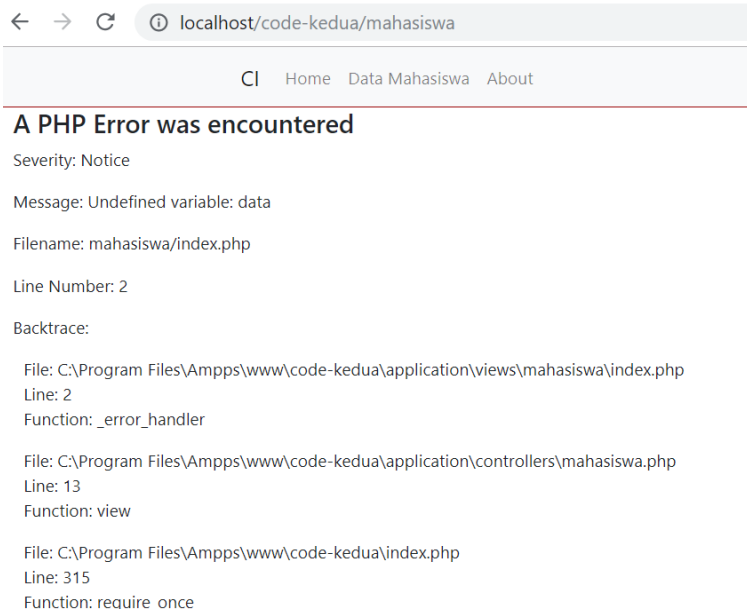
```



## 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)

```
1 <?php
2 var_dump($data);
3 ?>
```



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

← → ↻ localhost/code-kedua/mahasiswa

CI Home Data Mahasiswa About

2 =>

```
array (size=5)
  'id' => string '3' (Length=1)
  'nama' => string 'Inayati Machsus' (Length=15)
  'nim' => string '1841720202' (Length=10)
  'email' => string 'inayati@gmail.com' (Length=17)
  'jurusan' => string 'Teknik Informatika' (Length=18)
```

2 =>

```
array (size=5)
  'id' => string '1' (Length=1)
  'nama' => string 'Livia Yurike Khuril Maula' (Length=25)
  'nim' => string '1841720025' (Length=10)
  'email' => string 'kliviayurike@gmail.com' (Length=22)
  'jurusan' => string 'Teknik Informatika' (Length=18)
```

1 =>

```
array (size=5)
  'id' => string '2' (Length=1)
  'nama' => string 'Darin Zahira' (Length=12)
  'nim' => string '1841720033' (Length=10)
  'email' => string 'darinzahira@gmail.com' (Length=21)
  'jurusan' => string 'Teknik Informatika' (Length=18)
```

### Daftar Mahasiswa

|                         |
|-------------------------|
| Cras justo odio         |
| Dapibus ac facilisis in |
| Morbi leo risus         |
| Porta ac consectetur ac |
| Vestibulum at eros      |

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**

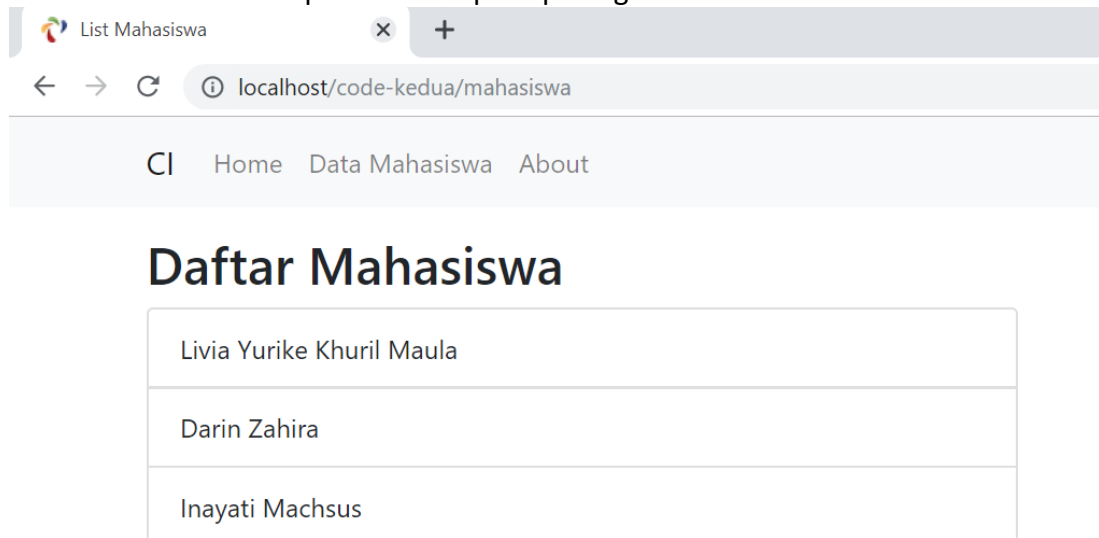
```

mahasiswa.php  mahasiswa_model.php  index.php X
application > views > mahasiswa > index.php
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)**

```

mahasiswa.php  mahasiswa_model.php  index.php  header.php X
application > views > template > header.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" integrity=
10
11      <!--<title>Hello, world!</title>-->
12      <title>= $title ?&gt;&lt;/title&gt;
13  &lt;/head&gt;
14  &lt;body&gt;
15
16  &lt;!-- navbar --&gt;
17  &lt;nav class="navbar navbar-expand-lg navbar-light bg-light"&gt;
18      &lt;div class="container"&gt;
19          &lt;a class="navbar-brand" href="#"&gt;CI&lt;/a&gt;
20          &lt;button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls
21              &lt;span class="navbar-toggler-icon"&gt;&lt;/span&gt;
22          &lt;/button&gt;
23          &lt;div class="collapse navbar-collapse" id="navbarNavAltMarkup"&gt;
24              &lt;div class="navbar-nav"&gt;
25                  &lt;a class="nav-item nav-link" href="<?= base_url(); ?&gt;"&gt;Home &lt;span class="sr-only"&gt;(current)&lt;/span&gt;&lt;/a&gt;
26                  &lt;a class="nav-item nav-link" href="<?= base_url(); ?&gt;mahasiswa"&gt;Data Mahasiswa&lt;/a&gt;
27                  &lt;a class="nav-item nav-link" href="<?= base_url(); ?&gt;"&gt;About&lt;/a&gt;
28                  &lt;!-- &lt;a class="nav-item nav-link disabled" href="#"&gt;Disabled&lt;/a&gt; --&gt;
29              &lt;/div&gt;
30          &lt;/div&gt;
</pre

```

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')`

| FILE              | LINE | CODE   |
|-------------------|------|--|
| CODE-KEDUA        | 85   | -----  |
| application       | 86   | Auto-load Helper Files                       |
| > cache           | 87   | -----  |
| > config          | 88   | Prototype:                                   |
| autoload.php      | 89   |  |
| config.php        | 90   | \$autoload['helper'] = array('url', 'file'); |
| constants.php     | 91   | */   |
| database.php      | 92   | \$autoload['helper'] = array('url');         |
| doctypes.php      | 93   |  |
| foreign_chars.php | 94   | /*   |
|                   | 95   | -----  |

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> 1 2 &lt;div class="container"&gt; 3   &lt;div class="row mt-4"&gt; 4     &lt;div class="col-md-6"&gt; 5       &lt;a href="" class="btn btn-primary"&gt; Tambah Data&lt;/a&gt; 6     &lt;/div&gt; 7   &lt;/div&gt; 8   &lt;!-- mt-3 artinya margin top 3 --&gt; 9 &lt;div class="row mt-3"&gt; 10  &lt;div class="col-md-6"&gt; 11    &lt;h2&gt;Daftar Mahasiswa&lt;/h2&gt; 12    &lt;ul class="list-group"&gt; 13      &lt;?php foreach(\$mahasiswa as \$mhs):?&gt; 14        &lt;li class="list-group-item"&gt;&lt;?= \$mhs['nama'];?&gt;&lt;/li&gt; 15      &lt;?php endforeach; ?&gt; 16    &lt;/ul&gt; 17  &lt;/div&gt; 18 &lt;/div&gt; 19 &lt;/div&gt; </pre>   |
| 2       | <p>Dikarenakan <b>mahasiswa_model</b> ini nantinya akan digunakan oleh semua <i>function</i> di <i>controller</i> mahasiswa, maka “ <i>\$this-&gt;load-&gt;model('mahasiswa_model');</i> ” yang awalnya hanya berada pada <b>function index</b>, kita ubah ke <b>function construct</b> agar semua <i>function</i> dapat menggunakan <b>mahasiswa_model</b>.</p>  <pre> 1 &lt;?php 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 untuk menjalankan fungsi construct pada class parent(ci_controller) 11         parent::__construct(); 12         \$this-&gt;load-&gt;model('mahasiswa_model'); 13     } 14 15     public function index() 16     { 17 18         // modul untuk load database 19         // \$this-&gt;load-&gt;database(); 20         \$data['title']='List Mahasiswa'; 21         \$data['mahasiswa']=\$this-&gt;mahasiswa_model-&gt;getAllmahasiswa(); 22         \$this-&gt;load-&gt;view('template/header',\$data); 23         \$this-&gt;load-&gt;view('mahasiswa/index',\$data); 24         \$this-&gt;load-&gt;view('template/footer'); 25     } 26 27 } 28 29 30 /* 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**)

```

mahasiswa.php
application > controllers > mahasiswa.php

27
28     public function tambah() {
29         $data('title')= 'Form Menambahkan Data Mahasiswa':
30         $this->load->view('template/header', $data);
31         $this->load->view('mahasiswa/tambah', $data);
32         $this->load->view('template/footer');
33     }
34
35

```

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

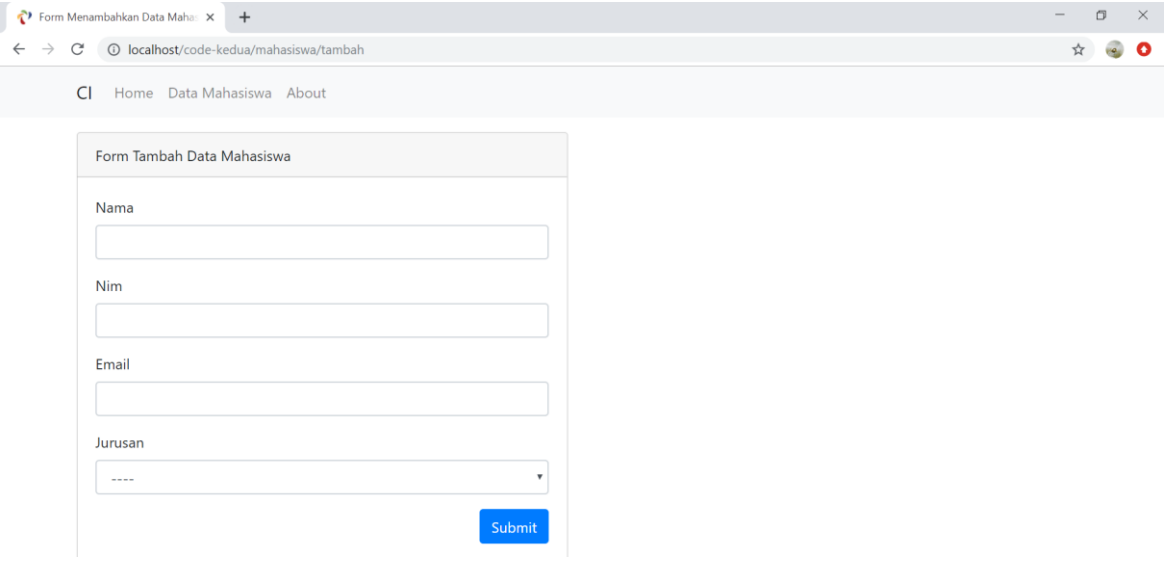
Setelah kita membuat **function** **tambah** pada **controller** mahasiswa, selanjutnya buatlah **form** **tambah** pada **views/mahasiswa/tambah.php**

```

mahasiswa.php • tambah.php
application > views > mahasiswa > tambah.php

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                      Form Tambah Data Mahasiswa
8                  </div>
9                  <div class="card-body">
10                     <form action="" method="post">
11                         <!-- http://getbootstrap.com/docs/4.1/components/card/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>

```

|   |  |
|---|--|
|   | <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Pada <i>line 33</i>, terdapat atribut class dengan <i>value</i> btn btn-primary float-right yang artinya kita membuat sebuah <i>button</i> dan posisi buttonnya ada di kanan</li> <li>- Ganti <i>value</i> dari <i>line 28-31</i> menjadi seperti di bawah ini (<b>Agar di tabel tidak tersimpan <i>value</i> berupa angka</b>)</li> </ul> <pre>&lt;option value="informatika"&gt;Informatika&lt;/option&gt; &lt;option value="kimia"&gt;Kimia&lt;/option&gt; &lt;option value="Mesin"&gt;Mesin&lt;/option&gt;</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 <i>index</i> pada <i>views/mahasiswa/index.php</i> dan kita atur <i>href</i> nya</p> <pre>application &gt; views &gt; mahasiswa &gt; index.php 1 2 &lt;div class="container"&gt; 3   &lt;div class="row mt-4"&gt; 4     &lt;div class="col-md-6"&gt; 5       &lt;a href="&lt;?= base_url();?&gt;mahasiswa/tambah" class="btn btn-primary"&gt; Tambah Data&lt;/a&gt; 6     &lt;/div&gt; 7   &lt;/div&gt; 8   &lt;!-- mt-3 artinya margin top 3 --&gt; 9   &lt;div class="row mt-3"&gt; 10    &lt;div class="col-md-6"&gt; 11      &lt;h2&gt;Daftar Mahasiswa&lt;/h2&gt; 12      &lt;ul class="list-group"&gt; 13        &lt;?php foreach(\$mahasiswa as \$mhs):?&gt; 14          &lt;li class="list-group-item"&gt;&lt;?= \$mhs['nama'];?&gt;&lt;/li&gt; 15        &lt;?php endforeach; ?&gt; 16      &lt;/ul&gt; 17    &lt;/div&gt; 18  &lt;/div&gt; 19 &lt;/div&gt;</pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Tambahkan <i>code</i> pada <i>line 5</i> agar pada saat <i>user</i> melakukan <i>event click</i> maka <i>user</i> akan ditampilkan halaman tambah.</li> </ul> |
| 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>   |

[https://www.codeigniter.com/user\\_guide/libraries/form\\_validation.html?highlight=form#the-form](https://www.codeigniter.com/user_guide/libraries/form_validation.html?highlight=form#the-form) (searching controller)

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

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

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](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)

```

28     public function tambah() {
29         $data['title'] = 'Form Menambahkan Data Mahasiswa';
30
31         // $this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required');
32         $this->load->library('form_validation');
33         $this->form_validation->set_rules('nama', 'Nama', 'required');
34
35
36         if ($this->form_validation->run() == FALSE) {
37             # code...
38             $this->load->view('template/header', $data);
39             $this->load->view('mahasiswa/tambah', $data);
40             $this->load->view('template/footer');
41         }
42         else {
43             # code...
44             echo " data berhasil ditambah";
45         }
46
47     }
48 }

```

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

```

<div class="card-body">
  <?php = validation_errors() ?>
  <form action="" method="post">
  <!-- http://getbootstrap.com/docs/4.1/components/card/forms/ -->
  <div class="form-group">
    <label for="nama">Nama</label>
    <input type="text" class="form-control" id="nama">
  </div>
  <div class="form-group">

```



- 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">
      <?php= validation_errors(); ?>
    </div>
  <?php endif ?>
```

- Tambahkan tanda ; pada akhir **validation\_errors()** sehingga menjadi seperti berikut  
<?php= validation\_errors(); ?>

- i. Akan dihasilkan seperti gambar di bawah ini

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

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

- k. Hasilnya seperti berikut ini.

Form Tambah Data 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)**

```
31 // $this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_length[5]|max_length[128]');
32 $this->load->library('form_validation');
33 $this->form_validation->set_rules('nama', 'Nama', 'required');
34 $this->form_validation->set_rules('nim', 'Nim', 'required');
35 $this->form_validation->set_rules('email', 'Email', 'required|valid_email');
```

- m. Hasilnya sebagai berikut.


Form Tambah Data Mahasiswa


Nama

Nim

Email

Jurusan





- 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

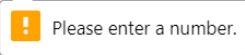
Nama

Nim

Email

Jurusan

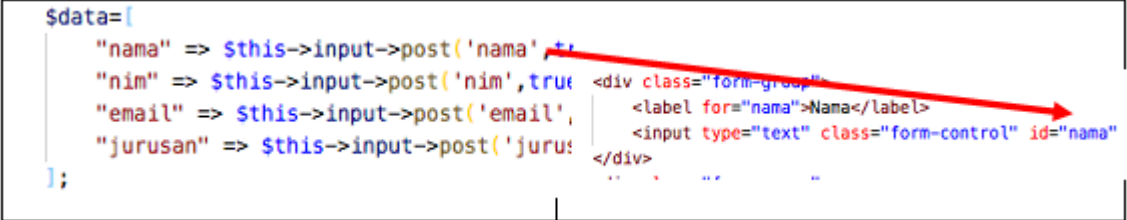
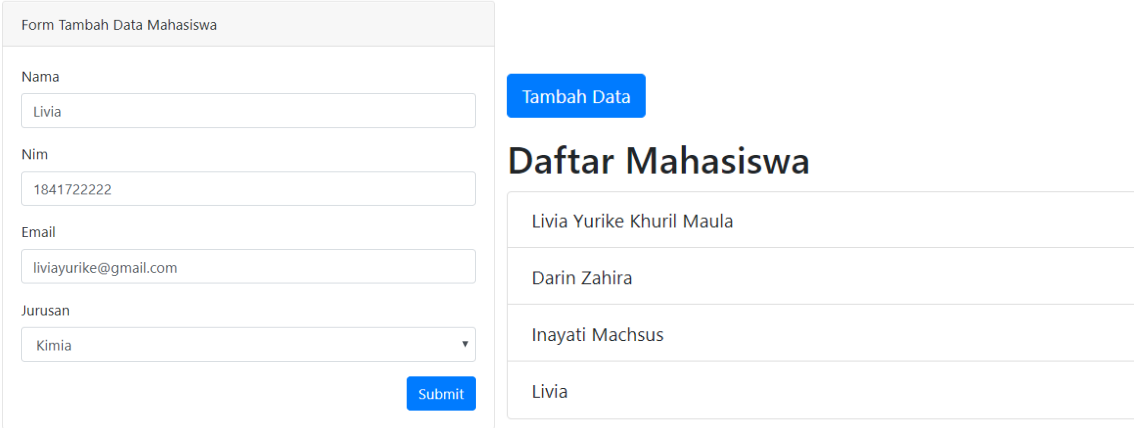




## Praktikum – Bagian 3: Menambahkan Data ke Database / Insert Data

Setelah kita berhasil membuat **form\_validation** langkah berikutnya adalah memasukkan data ke database (insert database)

| Langkah | Keterangan   |
|---------|--|
| 1       | <p>Buka <b>controller/mahasiswa</b> dan tambahkan <i>code</i> berikut pada <b>function tambah</b> agar pada saat <b>form_validation</b> bernilai <b>true</b>, maka akan ditambahkan data dengan nama modelnya - &gt; <b>model_mahasiswa</b> dengan nama <b>function tambahdatamhs()</b> dan jika sudah ditambah ke database maka akan <b>redirect</b> ke <b>controller mahasiswa</b> (line 46 - 47)</p> <pre> 28 public function tambah() { 29     \$data['title'] = 'Form Menambahkan Data Mahasiswa'; 30 31     \$this-&gt;form_validation-&gt;set_rules('fieldname', 'fieldlabel', 'trim required min_leng 32     \$this-&gt;load-&gt;library('form_validation'); 33     \$this-&gt;form_validation-&gt;set_rules('nama', 'Nama', 'required'); 34     \$this-&gt;form_validation-&gt;set_rules('nim', 'Nim', 'required numeric'); 35     \$this-&gt;form_validation-&gt;set_rules('email', 'Email', 'required valid_email'); 36 37 38     if (\$this-&gt;form_validation-&gt;run() == FALSE) { 39         # code... 40         \$this-&gt;load-&gt;view('template/header', \$data); 41         \$this-&gt;load-&gt;view('mahasiswa/tambah', \$data); 42         \$this-&gt;load-&gt;view('template/footer'); 43     } 44     else { 45         # code... 46         \$this-&gt;mahasiswa_model-&gt;tambahdatamhs(); 47         redirect('mahasiswa','refresh'); 48     } 49 } </pre>  |
| 2       | <p>Jika kita lihat <i>code</i> pada <b>controller mahasiswa</b>(line 46) terdapat <b>model</b> dengan nama <b>mahasiswa_model</b> dan <b>function</b>-nya <b>tambahdatamhs()</b>. Untuk itu kita harus membuat sebuah <b>function</b> dengan nama <b>tambahdatamhs()</b></p>   |
| 3       | <p>Buatlah sebuah <b>function tambahdatamhs()</b> pada <b>models/mahasiswa_model</b></p> <ul style="list-style-type: none"> <li><a href="https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=insert#inserting-data">https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=insert#inserting-data</a></li> <li>Tambahkan <i>code</i> seperti pada gambar dibawah ini</li> </ul> <pre> 5 class mahasiswa_model extends CI_Model { 6 7     public function getAllmahasiswa() 8     { 9         // \$query digunakan untuk menampung isi dari tabel mahasiswa 10        \$query=\$this-&gt;db-&gt;get('mahasiswa'); 11 12        //untuk menampilkan isi dari query 13        return \$query-&gt;result_array(); 14 15        // atau bisa juga menggunakan code berikut 16        return \$this-&gt;db-&gt;get('mahasiswa')-&gt;result_array(); 17    } 18 19    public function tambahdatamhs() { 20        \$data=[ 21            "nama" =&gt; \$this-&gt;input-&gt;post('nama',true), // ini sama dengan htmlspecialchars(\$_POST['nama']) 22            "nim" =&gt; \$this-&gt;input-&gt;post('nim',true), 23            "email" =&gt; \$this-&gt;input-&gt;post('email',true), 24            "jurusan" =&gt; \$this-&gt;input-&gt;post('jurusan',true) 25        ]; 26        // \$this-&gt;db-&gt;insert('Table', \$object); 27        \$this-&gt;db-&gt;insert('mahasiswa', \$data); 28    } 29 } 30 31 /* End of file ModelName.php */ 32 ?&gt; </pre> |

|   |  |
|---|--|
|   | <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- <i>Line 22-27</i> adalah sebuah array dimana nantinya isi array adalah data-data yang akan dikirim (POST) ke database</li> <li>- <i>Line 23</i> "nama" =&gt; \$this-&gt;input-&gt;post('nama',true) <ul style="list-style-type: none"> <li>• nama-&gt;alias</li> <li>• Parameter nama pada post adalah name pada html seperti pada gambar dibawah ini</li> <li>• Parameter <i>true</i> digunakan untuk mengaktifkan htmlspecialchars</li> </ul> </li> </ul>  <ul style="list-style-type: none"> <li>• <i>Line 29</i> adalah proses <i>insert</i> data ke database dengan <b>2 parameter</b> yaitu <b>nama</b> <b>tablenya</b> dan <b>objectnya</b> (dalam hal ini data yang akan dimasukkan ke database)</li> </ul> |
| 4 | <p>Langkah berikutnya kita coba untuk tambahkan data.<br/>Jika berhasil maka data akan tampil di halaman <b>index.php</b></p> <p>a. <code>redirect('mahasiswa','refresh');</code>(<i>line 47</i> gambar <i>controller mahasiswa</i>)</p> <p>b. Hasil</p>    |

-- Selamat Mengerjakan --