



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

Nama : Septianda Reza Maulana
Kelas : TI-2A
NIM/No : 1841720135/26

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 > 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 public function index() 8 { 9 //modul untuk load database 10 \$this->load->database(); 11 \$data['title']='List Mahasiswa'; 12 \$this->load->view('template/header',\$data); 13 \$this->load->view('mahasiswa/index'); 14 \$this->load->view('template/footer'); 15 } 16 17 } 18 19 /* End of file Controllername.php */ 20 ?></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> (<i>line 7-12</i>)

```

mahasiswa.php x
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 /* End of file Controllername.php */
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 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

```

autoload.php x
application > config > autoload.php > ...
43
44  /*
45  | -----
46  |  Auto-load Libraries
47  | -----
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 x
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 parrent(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 /* End of file Controllername.php */
27

```

6

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

```

EXPLORER
v OPEN EDITORS
x database.php application\config
v BELAJAR_CI2
v application
  > cache
  v 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
  v controllers
    Home.php
    ...

database.php x
application > config > database.php > ...
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

```

database.php ×
application > config > database.php > ...

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 );

```

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 buatlah sebuah database dengan nama **kuliah_CI** dan dengan tabel **mahasiswa** seperti pada gambar dibawah ini

#	Nama	Jenis	Penyortiran	Atribut	Tak Terbilang	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(30)		Tidak	Tidak ada			AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama	varchar(100)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	nim	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
4	email	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
5	jurusan	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya

9

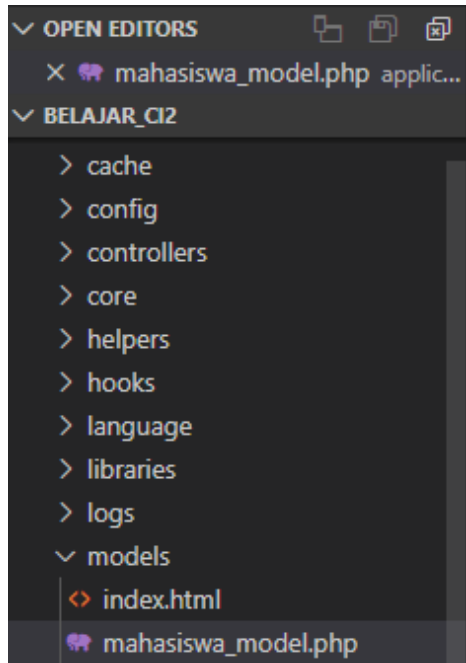
Tambahkan beberapa data ke dalam tabel mahasiswa seperti pada gambar (pada contoh terdapat 3 data, jumlah data lebih banyak tidak masalah)

<input type="checkbox"/>	Ubah	Salin	Hapus	1	Reza	1841720135	septiandareza07@gmail.com	Teknologi Informasi
<input type="checkbox"/>	Ubah	Salin	Hapus	2	Baul	1831210209	mambaul@gmail.com	Teknik Mesin
<input type="checkbox"/>	Ubah	Salin	Hapus	3	Roni	1941150042	ronii12@gmail.com	Teknik Elektro

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.php** pada folder **application/models**



b. Tambahkan *function* getAllmahasiswa pada mahasiswa_model.php

```

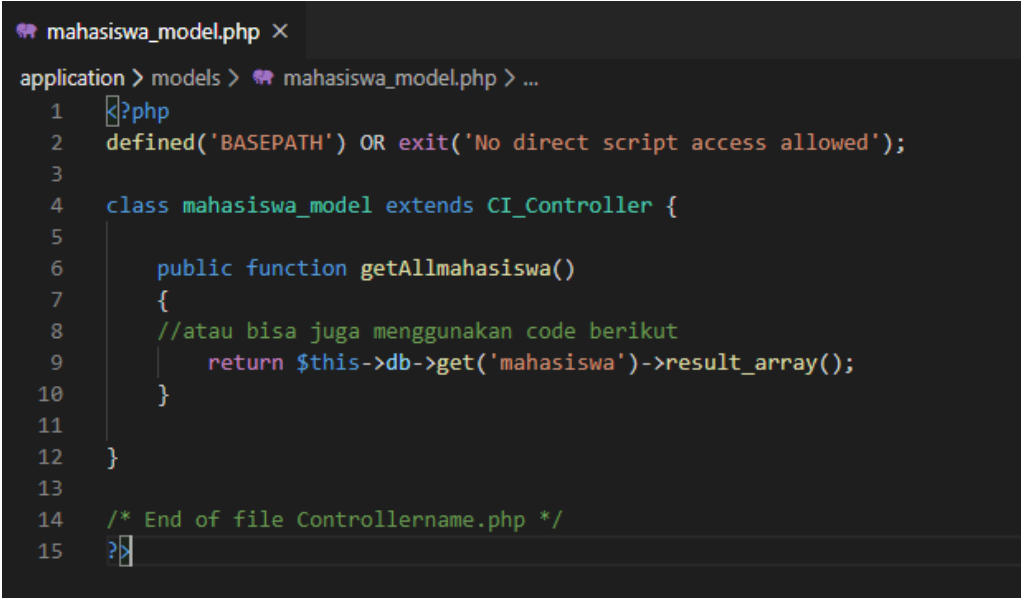
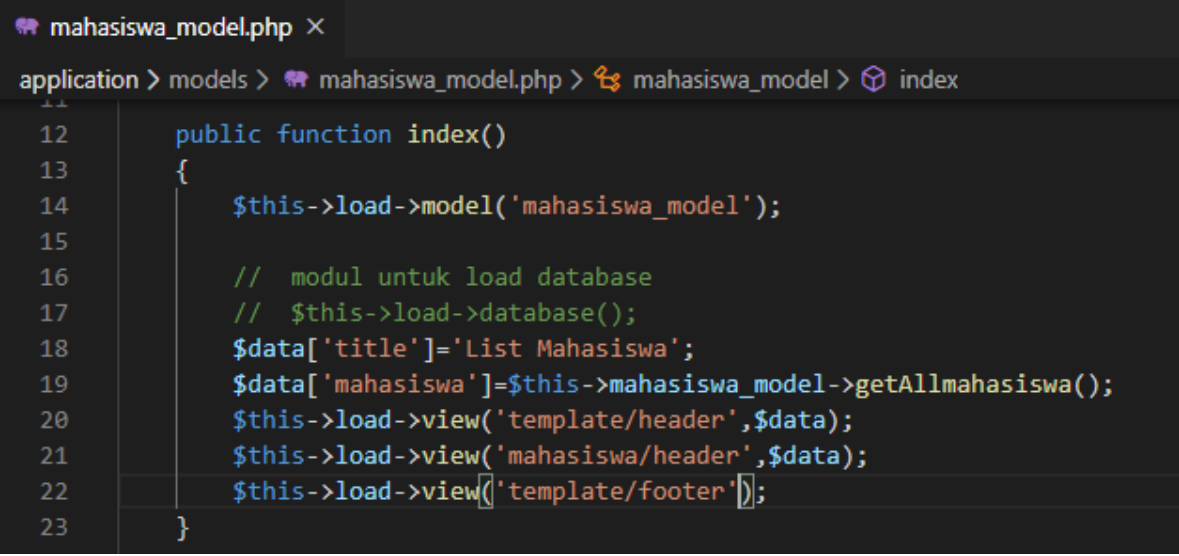
mahasiswa_model.php x
application > models > mahasiswa_model.php > ...
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class mahasiswa_model extends CI_Controller {
5
6      public function getAllmahasiswa()
7      {
8          // https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=query%20builder
9          // $query digunakan untuk menampung isi dari tabel mahasiswa
10         $query = $this->db->get('mahasiswa');
11
12         // https://www.codeigniter.com/user_guide/database/results.fa-html
13         // untuk menampilkan isi dari query
14         return $query->result_array();
15     }
16
17 }
18
19 /* End of file Controllername.php */
20 >

```

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

	 <pre> mahasiswa_model.php × application > models > mahasiswa_model.php > ... 1 <?php 2 defined('BASEPATH') OR exit('No direct script access allowed'); 3 4 class mahasiswa_model extends CI_Controller { 5 6 public function getAllmahasiswa() 7 { 8 //atau bisa juga menggunakan code berikut 9 return \$this->db->get('mahasiswa')->result_array(); 10 } 11 12 } 13 14 /* End of file Controllername.php */ 15 > </pre>
11	<p>Setelah kita mempunyai sebuah model dengan nama mahasiswa_model maka langkah berikutnya kita akan <i>load</i> model ini ke dalam controller mahasiswa.</p>  <pre> mahasiswa_model.php × application > models > mahasiswa_model.php > mahasiswa_model > index 12 public function index() 13 { 14 \$this->load->model('mahasiswa_model'); 15 16 // modul untuk load database 17 // \$this->load->database(); 18 \$data['title']='List Mahasiswa'; 19 \$data['mahasiswa']=\$this->mahasiswa_model->getAllmahasiswa(); 20 \$this->load->view('template/header',\$data); 21 \$this->load->view('mahasiswa/header',\$data); 22 \$this->load->view('template/footer'); 23 } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - <i>Line 17</i> digunakan untuk melakukan <i>load</i> model dengan nama modelnya adalah mahasiswa_model - <i>Line 22</i> 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 - <i>Line 24</i> tambahkan \$data

```

mahasiswa_model.php X
application > models > mahasiswa_model.php > ...

4  class mahasiswa_model extends CI_Controller {
5
6      public function getAllmahasiswa()
7      {
8          //atau bisa juga menggunakan code berikut
9          return $this->db->get('mahasiswa')->result_array();
10     }
11
12 }
13
14 /* End of file Controllername.php */
15
mahasiswa.php X
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      public function index()
8      {
9          //modul untuk load database
10         //$this->load->database();
11         $data['title']='List Mahasiswa';
12         $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
13         $this->load->view('template/header',$data);
14         $this->load->view('mahasiswa/index');
15         $this->load->view('template/footer');
16     }
17
18 }
19 /* End of file Controllername.php */
20

```

12

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

```

index.php X
application > views > mahasiswa > index.php > div.container > div.row.mt-3 > div.col-m

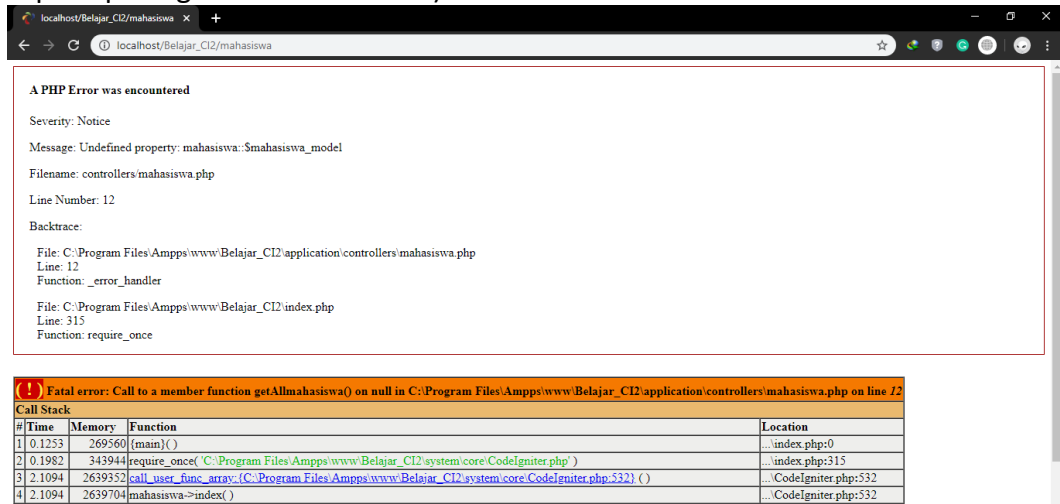
1  <?php
2  var_dump($mahasiswa);
3  >
4  <div class="container">
5      <!-- mt-3 artinya margin top 3 -->
6  <div class="row mt-3">
7      <div class="col-md-6">
8          <h2>Daftar Mahasiswa</h2>
9          <ul class="list-group">
10             <li class="list-group-item">Septianda Reza Maulana</li>
11             <li class="list-group-item">Septianda Reza Maulana</li>
12             <li class="list-group-item">Septianda Reza Maulana</li>
13             <li class="list-group-item">Agsal Fairrohmad</li>
14             <li class="list-group-item">Dhito Mandala Sakti</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)

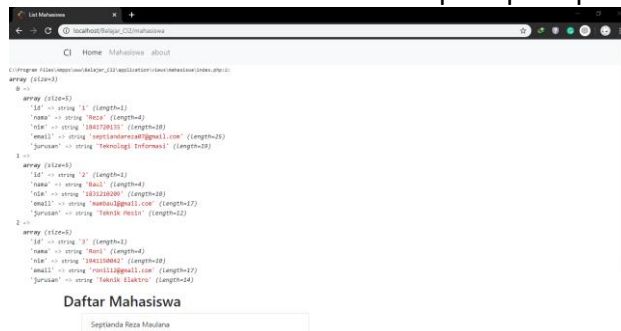
- Jika kita lihat *code* kedua, ada `$data` yang isi arraynya adalah mahasiswa dan isi mahasiswa tersebut adalah **function** `getAllmahasiswa` pada model dengan nama mahasiswa_model. Dimana nanti isi dari `$data['mahasiswa']` ini dikirimkan ke halaman **view** dengan **2 paramater** yaitu `mahasiswa/index` 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 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.

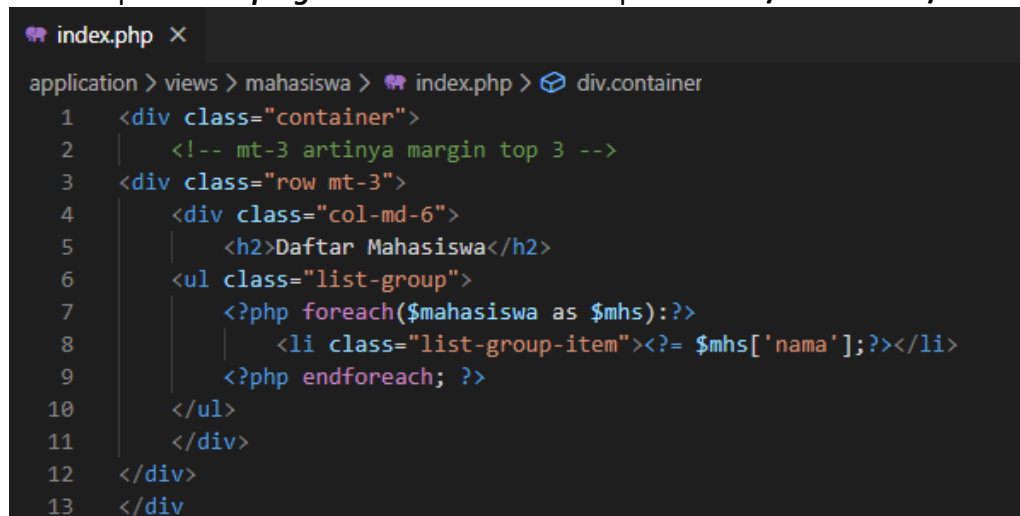
13

Jika berhasil maka data akan tampil seperti pada gambar berikut



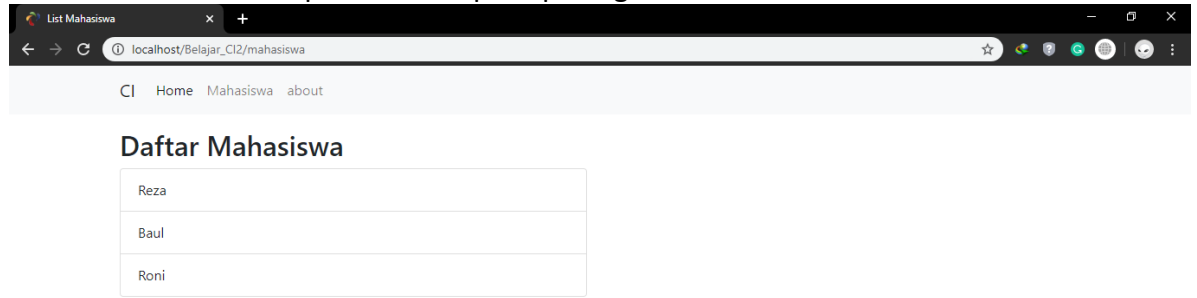
14

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



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)

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

```
autoload.php X
application > config > autoload.php > ...
84 /*
85 | -----
86 | Auto-load Helper Files
87 | -----
88 | Prototype:
89 |
90 | $autoload['helper'] = array('url', 'file');
91 */
92 $autoload['helper'] = array('url');
```

	c. Jika berhasil maka semua halaman dapat ditampilkan pada saat <i>user</i> melakukan <i>event click</i> .
--	--

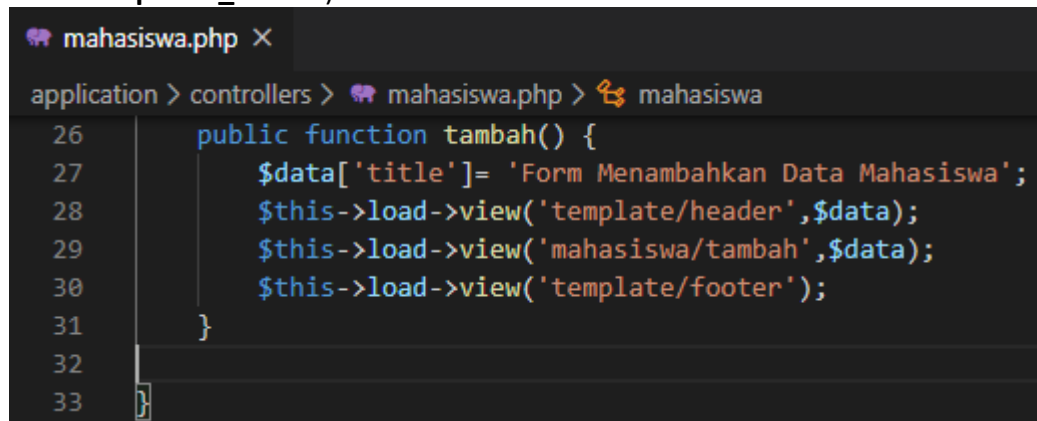
Praktikum – Bagian 2: Membuat Form Input dan Menambahkan Form Validation

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



```

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

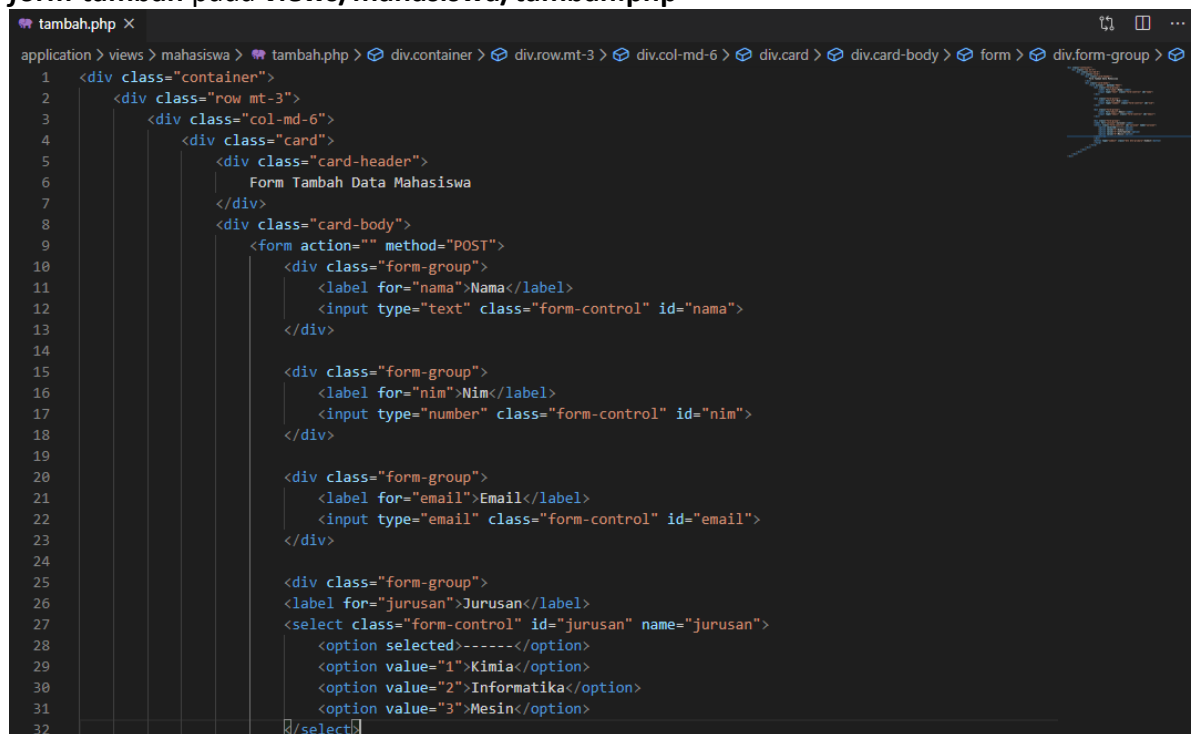
```

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**



```

1 <div class="container">
2   <div class="row mt-3">
3     <div class="col-md-6">
4       <div class="card">
5         <div class="card-header">
6           Form Tambah Data Mahasiswa
7         </div>
8         <div class="card-body">
9           <form action="" method="POST">
10            <div class="form-group">
11              <label for="nama">Nama</label>
12              <input type="text" class="form-control" id="nama">
13            </div>
14
15            <div class="form-group">
16              <label for="nim">Nim</label>
17              <input type="number" class="form-control" id="nim">
18            </div>
19
20            <div class="form-group">
21              <label for="email">Email</label>
22              <input type="email" class="form-control" id="email">
23            </div>
24
25            <div class="form-group">
26              <label for="jurusan">Jurusan</label>
27              <select class="form-control" id="jurusan" name="jurusan">
28                <option selected>-----</option>
29                <option value="1">Kimia</option>
30                <option value="2">Informatika</option>
31                <option value="3">Mesin</option>
32              </select>

```

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

```

<option value="informatika">Informatika</option>
<option value="Kimia">Kimia</option>
<option value="Mesin">Mesin</option>

```

5

Maka akan dihasilkan halaman seperti di bawah ini.

The screenshot shows a web browser window with the title 'Form Menambahkan Data Mahasiswa'. The address bar shows 'localhost/Belajar_CI2/mahasiswa/tambah'. The browser's navigation bar includes 'CI', 'Home', 'Data Mahasiswa', and 'About'. The main content area displays a form titled 'Form Tambah Data Mahasiswa'. The form contains four input fields: 'Nama', 'Nim', 'Email', and 'Jurusan' (a dropdown menu). Below these fields is a blue 'Submit' button.

6

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

```
index.php X
application > views > mahasiswa > index.php > div.container
1 <div class="container">
2   <div class="row mt-4">
3     <div class="col-md-6">
4       <a href="<?= base_url();?>mahasiswa/tambah" class="btn btn-primary"> Tambah Data</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"><?= $mhs['nama'];?></li>
14      <?php endforeach; ?>
15    </ul>
16  </div>
17 </div>
18 </div>
```

Keterangan :

- Tambahkan *code* pada line 5 agar pada saat *user* melakukan *event click* maka *user* akan ditampilkan halaman tambah.

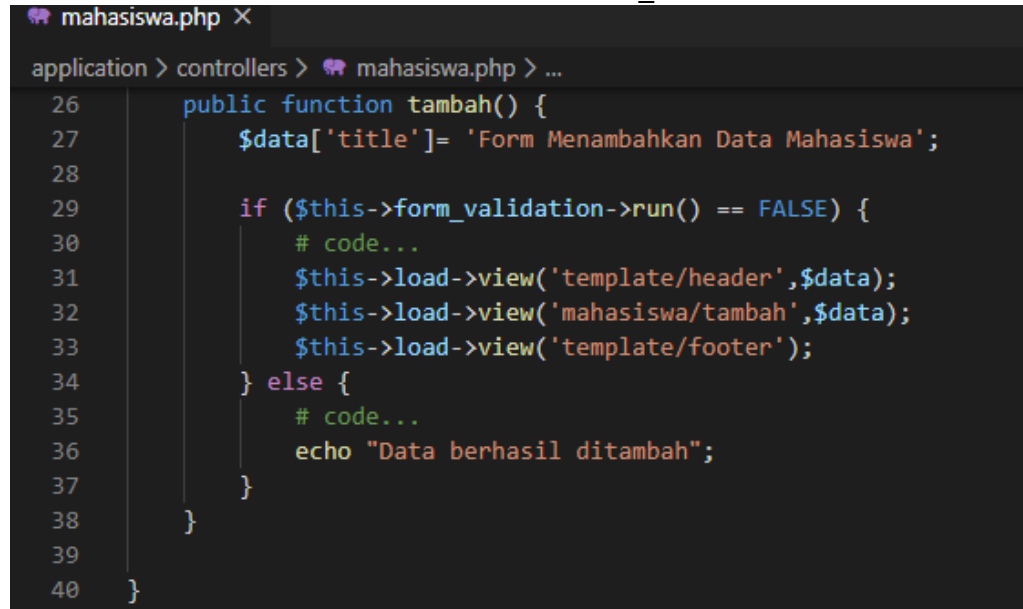
7

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`



```

mahasiswa.php X
application > controllers > mahasiswa.php > ...
26     public function tambah() {
27         $data['title'] = 'Form Menambahkan Data Mahasiswa';
28
29         if ($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
40 }
  
```

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**

```
tambah.php X
application > views > mahasiswa > tambah.php > div.container > div.row.mt-3 > div.col-md-6 > div.card > div.card-body > form >
10 <div class="form-group">
11 <label for="nama">Nama</label>
12 <input type="text" class="form-control" id="nama" name="nama">
13 </div>
14
15 <div class="form-group">
16 <label for="nim">Nim</label>
17 <input type="number" class="form-control" id="nim" name="nim">
18 </div>
19
20 <div class="form-group">
21 <label for="email">Email</label>
22 <input type="email" class="form-control" id="email" name="email">
23 </div>
24
25 <div class="form-group">
26 <label for="jurusan">Jurusan</label>
27 <select class="form-control" id="jurusan" name="jurusan">
28 <option selected>-----</option>
29 <option value="informatika">Informatika</option>
30 <option value="kimia">Kimia</option>
31 <option value="mesin">Mesin</option>
32 </select>
33 </div>
```

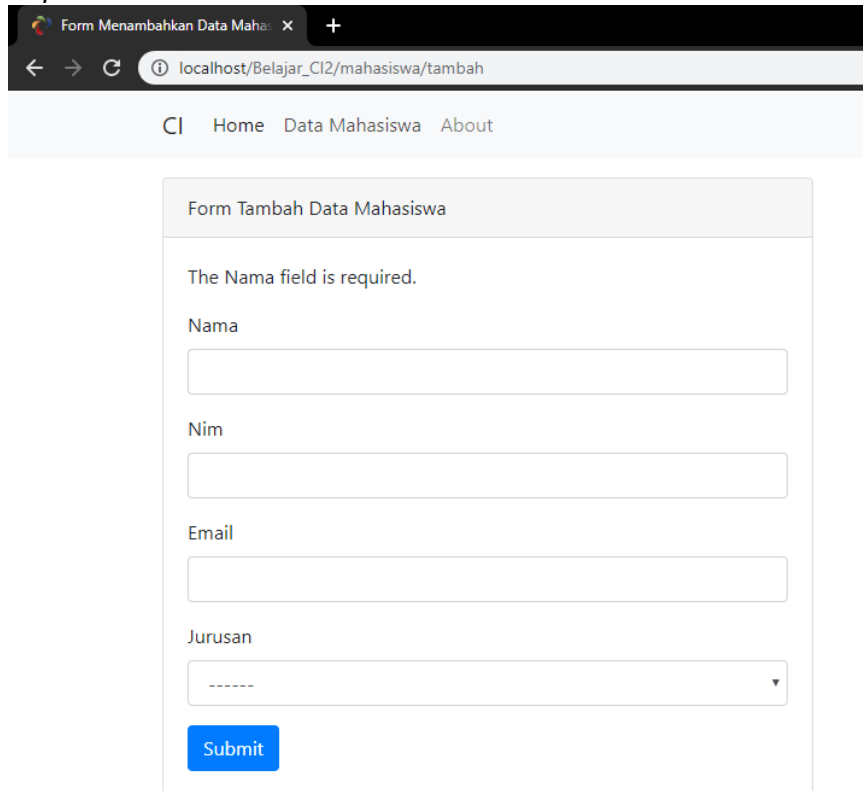
- e. Setelah kita tambahkan atribut **name** maka tambahkan **rule** pada **controller/mahasiswa** seperti pada gambar dibawah ini (line 35-46)

```
mahasiswa.php X
application > controllers > mahasiswa.php > ...
26 public function tambah() {
27     $data['title'] = 'Form Menambahkan Data Mahasiswa';
28
29     $this->form_validation->set_rules('nama', 'Nama', 'required');
30
31     if ($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 }
```

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

```
tambah.php X
application > views > mahasiswa > tambah.php > div.container > div.row.mt-3 > div.col-md-6 > div.card > div.card-body
9 <?php validation_errors() ?>
10 <form action="" method="POST">
11 <div class="form-group">
12 <label for="nama">Nama</label>
13 <input type="text" class="form-control" id="nama" name="nama">
14 </div>
```

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



Form Menambahkan Data Mahasiswa

localhost/Belajar_CI2/mahasiswa/tambah

Home Data Mahasiswa About

Form Tambah Data Mahasiswa

The Nama field is required.

Nama

Nim

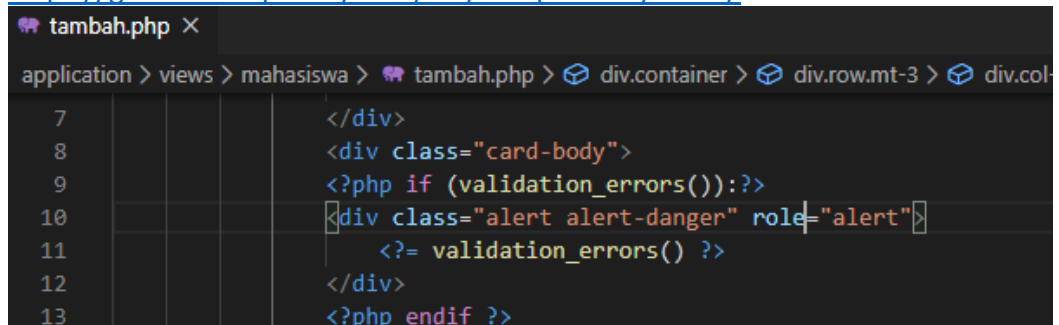
Email

Jurusan

Submit

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

- <https://getbootstrap.com/docs/4.1/components/alerts/>



```
tambah.php X
application > views > mahasiswa > tambah.php > div.container > div.row.mt-3 > div.col-
7      </div>
8      <div class="card-body">
9      <?php if (validation_errors()):?>
10     <div class="alert alert-danger" role="alert">
11         <?= validation_errors() ?>
12     </div>
13     <?php endif ?>
```

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

- i. Akan dihasilkan seperti gambar di bawah ini

Form Menambahkan Data Mahasiswa

← → ↻ localhost/Belajar_CI2/mahasiswa/tambah

CI Home Data Mahasiswa About

Form Tambah Data Mahasiswa

The Nama field is required.

Nama

Nim

Email

Jurusan

Submit

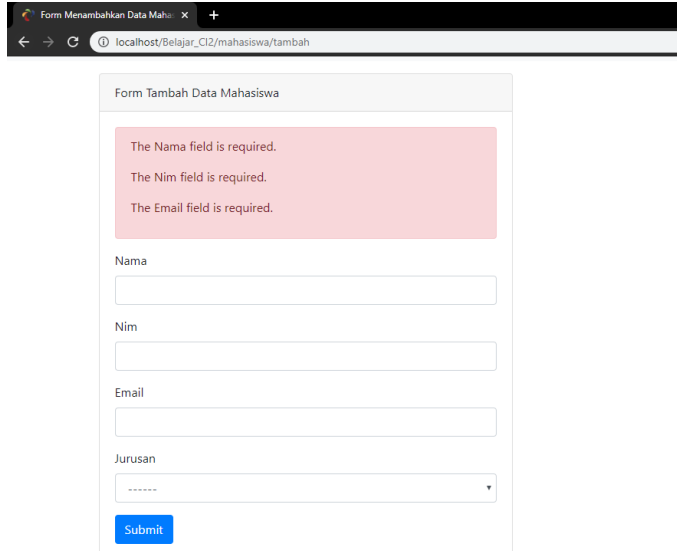
- j. Setelah kita berhasil memberikan **rules validation** pada **nama**, maka tambahkan juga untuk **nim** dan **email**.

```

mahasiswa.php x
application > controllers > mahasiswa.php > mahasiswa > tambah
28 public function tambah() {
29     $data['title'] = 'Form Menambahkan Data Mahasiswa';
30     $this->load->library('form_validation');
31
32     $this->form_validation->set_rules('nama', 'Nama', 'required');
33     $this->form_validation->set_rules('nim', 'Nim', 'required');
34     $this->form_validation->set_rules('email', 'Email', 'required');
35
36     if ($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     } else {
42         # code...
43         echo "Data berhasil ditambah";
44     }
45 }
46
47 }

```

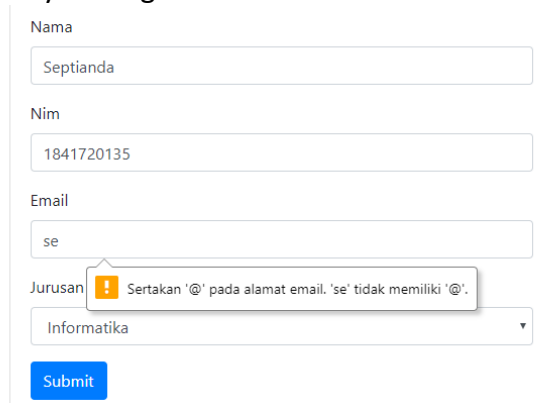
k. Hasilnya seperti berikut ini.



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

```
mahasiswa.php x
application > controllers > mahasiswa.php > mahasiswa > tambah
34 | $this->form_validation->set_rules('email', 'Email', 'required|valid_email');
```

m. Hasilnya sebagai berikut.



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

```
mahasiswa.php x
application > controllers > mahasiswa.php > mahasiswa > tambah
33 | $this->form_validation->set_rules('nim', 'Nim', 'required|numeric');
34 | $this->form_validation->set_rules('email', 'Email', 'required|valid_email');
```

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>mahasiswa.php X application > controllers > mahasiswa.php > ... 28 public function tambah() { 29 \$data['title'] = 'Form Menambahkan Data Mahasiswa'; 30 \$this->load->library('form_validation'); 31 32 \$this->form_validation->set_rules('nama', 'Nama', 'required'); 33 \$this->form_validation->set_rules('nim', 'Nim', 'required numeric'); 34 \$this->form_validation->set_rules('email', 'Email', 'required valid_email'); 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 } else { 42 # code... 43 \$this->mahasiswa_model->tambahdatamhs(); 44 redirect('mahasiswa', 'refresh'); 45 } 46 } 47 48 }</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 <pre>mahasiswa_model.php X application > models > mahasiswa_model.php > ... 1 <?php 2 defined('BASEPATH') OR exit('No direct script access allowed'); 3 4 class mahasiswa_model extends CI_Controller { 5 6 public function getAllmahasiswa() 7 { 8 \$query=\$this->db->get('mahasiswa'); 9 return \$query->result_array(); 10 11 //atau bisa juga menggunakan code berikut 12 // return \$this->db->get('mahasiswa')->result_array(); 13 } 14 public function tambahdatamhs() { 15 \$data=[16 "nama" => \$this->input->post('nama',true), //ini sama dengan htmlspecialchars(\$_POST['nama']) 17 "nim" => \$this->input->post('nim',true), 18 "email" => \$this->input->post('email',true), 19 "jurusan" => \$this->input->post('jurusan',true) 20]; 21 // \$this->db->insert('Table', \$object) 22 \$this->db->insert('mahasiswa', \$data); 23 } 24 } 25 26 27 /* End of file Controllername.php */ 28 </pre><p>Keterangan:</p><ul style="list-style-type: none">- <i>Line 22-27</i> adalah sebuah array dimana nantinya isi array adalah data-data yang akan dikirim (POST) ke database- <i>Line 23</i> "nama" => \$this->input->post('nama',true)<ul style="list-style-type: none">• nama->alias• Parameter nama pada post adalah name pada html seperti pada gambar dibawah ini</pre>

- 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 menambahkan data.

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

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

b. Hasil

Daftar Mahasiswa	
Reza	
Baul	
Roni	
Septianda Reza Maulana	
Septianda Reza Maulana	
Septianda Reza Maulana	

-- Selamat Mengerjakan --