

LAPORAN PRATIUM PEMOGRAMAN WEB LANJUT

JOBSHEET 2 – Read dan Input Database serta Form Validation



DISUSUN OLEH :

NAMA : MARDHIYAH MILLANIA

KELAS : 2A

PRODI : D-4 TEKNIK INFORMATIKA

NIM : 1841720081

ABSEN : 18



Praktikum – Bagian 1: Menampilkan data dari database menggunakan codeIgniter

Langkah	Keterangan
1	Buka kembali folder CodeIgniter yang telah dibuat pada Jobsheet sebelumnya
2	<p>Buka <i>controller</i> mahasiswa dan tambahkan <i>code</i> berikut (line 9)</p> <pre>application > controllers > mahasiswa.php 1 <?php 2 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 mahasiswa.php */ 20 21 ?></pre> <p>Keterangan :</p> <ul style="list-style-type: none">- Pada line 9 menjelaskan bawa <i>function</i> index ini akan melakukan <i>load</i> database. <i>Load</i> database ini hanya dilakukan di <i>function</i> index saja jika ada <i>function</i> lain maka kita harus melakukan <i>load</i> database lagi. Sehingga hal ini dirasa tidak praktis maka untuk itu kita harus tambahkan <i>code</i> agar <i>load</i> database ini dapat digunakan banyak <i>function</i>.
3	Tambahkan <i>code</i> berikut agar database dapat digunakan banyak <i>function</i> (line 7-12)



Jobsheet-2: Read dan Input Database serta Form Validation Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

	<div data-bbox="375 398 1404 1236"><p>mahasiswa.php</p><pre>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 //digunakan untuk menjalankan fungsi construct pada class parent(ci_control) 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 /* End of file mahasiswa.php */ 28 29 ?></pre></div> <p>Keterangan:</p> <ul style="list-style-type: none">- Line 7 – 12 adalah sebuah <i>function</i> construct (PERHATIKAN ITU MENGGUNAKAN DOUBLE UNDERSCORE __construct (tanpa spasi))- <i>Function</i> construct dijalankan pada saat classnya diinstansiasi- Line 10 digunakan untuk menjalankan fungsi construct pada class parent dimana class parent ini adalah CI_Controller- Line 11 digunakan agar untuk memload database (Semua fungsi di class mahasiswa dapat menggunakannya)- Jika kita sudah melakukan load database pada saat <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



	<pre>application > config > autoload.php 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', 'ses 55 56 You can also supply an alternative library name to be assi 57 in the controller: 58 59 \$autoload['libraries'] = array('user_agent' => 'ua'); 60 */ 61 \$autoload['libraries'] = array('database'); 62 63 /* 64 -----</pre>
5	Untuk kasus ini kita gunakan cara yang terakhir (no 4) karena kita asumsikan bahwa semua <i>controller</i> membutuhkan database. Maka kita <i>comment code</i> function construct yang ada di <i>controller</i> mahasiswa



	<pre>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 diinstansi 8 // public function __construct(){ 9 // //digunakan untuk menjalankan fungsi construct 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);</pre>
6	Setelah itu kita akan mengatur konfigurasi database. Untuk mengatur konfigurasi database maka buka folder application/config/database.php



Jobsheet-2: Read dan Input Database serta Form Validation Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

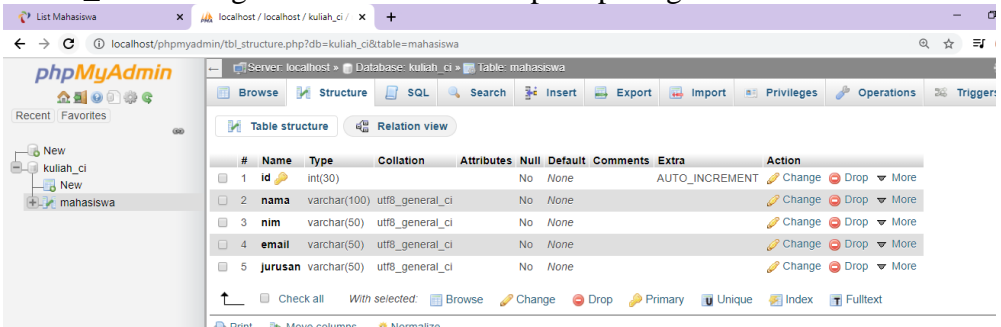
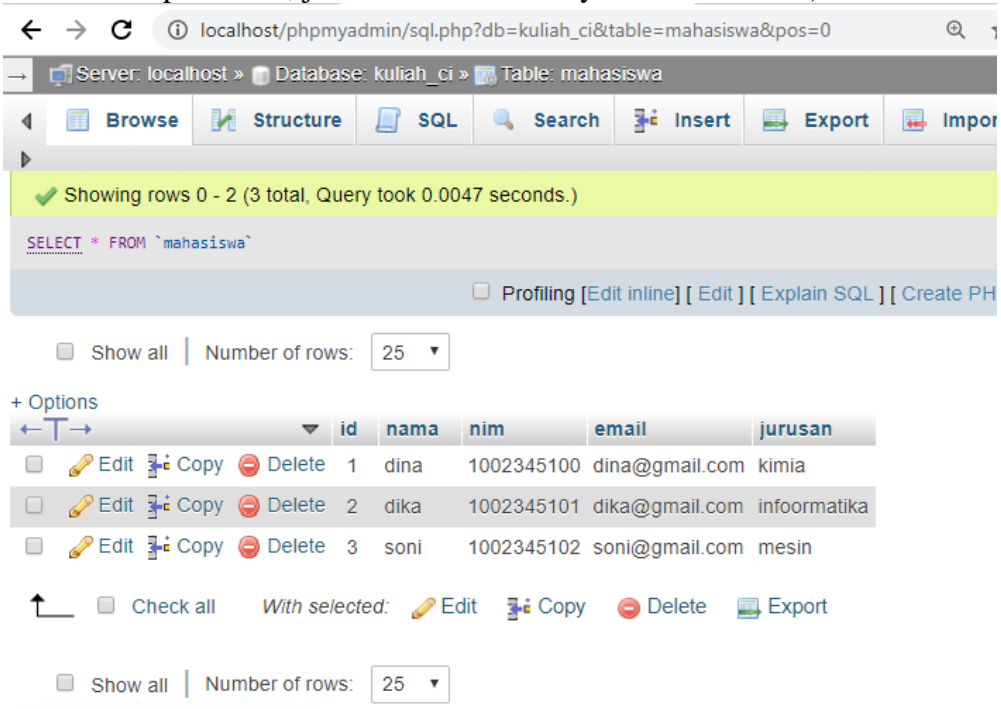
7	<p>Kita set konfigurasinya seperti pada gambar dibawah ini</p> <pre>73 \$active_group = 'default'; 74 \$query_builder = TRUE; 75 76 \$db['default'] = array(77 'dsn' => '', 78 'hostname' => 'localhost', 79 'username' => '', 80 'password' => '', 81 'database' => 'kuliah_CI', 82 'dbdriver' => 'mysqli', 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);</pre>



Jobsheet-2: Read dan Input Database serta Form Validation Mata Kuliah Pemrograman Web Lanjut

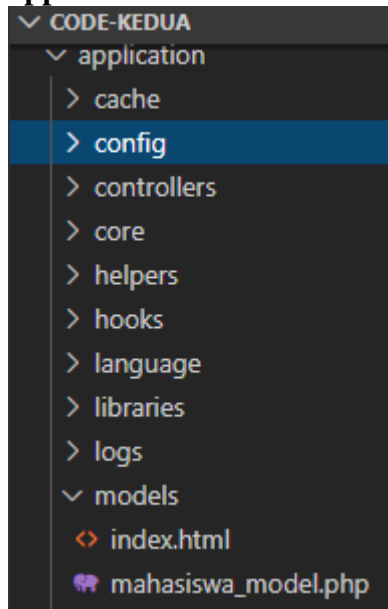
Pengampu: Tim Ajar Web Lanjut

Februari 2020

	<p>Keterangan :</p> <ul style="list-style-type: none">- Line 78 artinya <i>hostname</i> yang digunakan adalah localhost- Line 79 artinya <i>username</i> (phpmyadmin) yang digunakan apa- Line 80 artinya <i>password</i> (phpmyadmin) yang digunakan apa- Line 81 artinya nama database yang nanti digunakan
8	<p>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</p> 
9	<p>Tambahkan beberapa data ke dalam tabel mahasiswa seperti pada gambar (pada contoh terdapat 3 data, jumlah data lebih banyak tidak masalah)</p> 
10	<p>Setelah kita membuat sebuah <i>database</i>, maka langkah berikutnya adalah membuat sebuah model untuk melakukan proses <i>database</i>. Di dalam model ini kita akan membuat sebuah <i>function</i> untuk melakukan perintah ambil data dari <i>database</i>. Untuk lebih jelasnya ikuti langkah-langkah berikut:</p>



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



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

```
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_Controller {
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
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 /* End of file mahasiswa_model.php */
21
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

```
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          // //- https://www.codeigniter.com/user_guide/database
10         // //$query digunakan untuk menampung isi dari tabel ma
11         // $query=$this->db->get('mahasiswa');
12
13         // //- https://www.codeigniter.com/user_guide/database
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 /* End of file mahasiswa_model.php */
24
25
26 ?>
```

11

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



```
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      //     //digunakan untuk menjalankan fungsi construct pada
10     //     parent::__construct();
11     //     $this->load->database();
12     // }
13
14
15     public function index()
16     {
17         $this->load->model('mahasiswa_model');
18         //modul untuk load database
19         //$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     * End of file mahasiswa.php */
30
31 >
```

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**

12

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



Jobsheet-2: Read dan Input Database serta Form Validation Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

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>
19 </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 $data` maka akan terjadi *error* seperti
- Hal ini disebabkan karena tidak ketahu data pada mahasiswa/index.php pada *line 2* berupa `var_dump($mahasiswa)`, maka pastikan terlebih dahulu pada *controller* mahasiswa.php sudah terdapat \$data di *line 13* dan *15*.

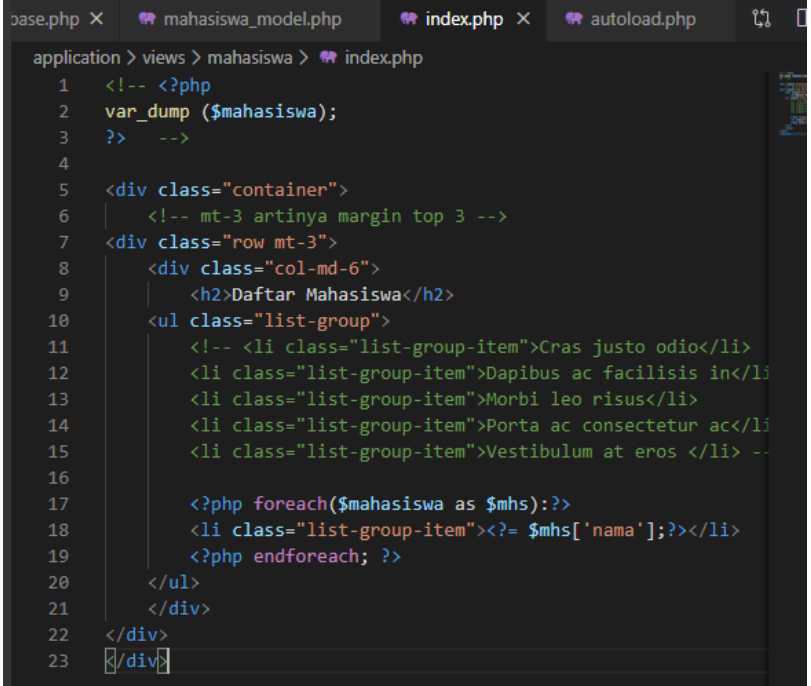
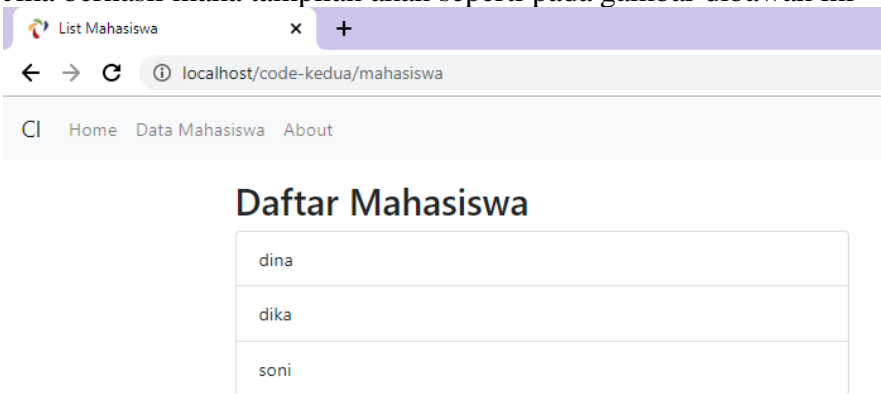
13

Jika berhasil maka data akan tampil seperti pada gambar berikut



	<div><div>List Mahasiswa</div><div>localhost/code-kedua/mahasiswa</div><div>CI Home Data Mahasiswa About</div><div>C:\Program Files\Amps\www\code-kedua\application\views\mahasiswa\index.php:2</div><div>array (size=3) 0 => array (size=5) 'id' => string '1' (length=1) 'nama' => string 'dina' (length=4) 'nim' => string '1002345100' (length=10) 'email' => string 'dina@gmail.com' (length=14) 'jurusan' => string 'kimia' (length=5) 1 => array (size=5) 'id' => string '2' (length=1) 'nama' => string 'dika' (length=4) 'nim' => string '1002345101' (length=10) 'email' => string 'dika@gmail.com' (length=14) 'jurusan' => string 'informatika' (length=12) 2 => array (size=5) 'id' => string '3' (length=1) 'nama' => string 'soni' (length=4) 'nim' => string '1002345102' (length=10) 'email' => string 'soni@gmail.com' (length=14) 'jurusan' => string 'mesin' (length=5)</div><div><h3>Daftar Mahasiswa</h3><table><tr><td>dina</td></tr><tr><td>dika</td></tr><tr><td>soni</td></tr></table></div></div>	dina	dika	soni
dina				
dika				
soni				
14	Agar semua isi dari tabel dapat tampil di halaman index mahasiswa maka kita harus lakukan proses looping . Gunakan <i>code</i> berikut pada views/mahasiswa/index			



	 <pre>base.php x mahasiswa_model.php index.php x autoload.php application > views > mahasiswa > index.php 1 <!-- <?php 2 var_dump (\$mahasiswa); 3 ?> --> 4 5 <div class="container"> 6 <!-- mt-3 artinya margin top 3 --> 7 <div class="row mt-3"> 8 <div class="col-md-6"> 9 <h2>Daftar Mahasiswa</h2> 10 <ul class="list-group"> 11 <!-- <li class="list-group-item">Cras justo odio 12 <li class="list-group-item">Dapibus ac facilisis in 13 <li class="list-group-item">Morbi leo risus 14 <li class="list-group-item">Porta ac consectetur ac 15 <li class="list-group-item">Vestibulum at eros --> 16 17 <?php foreach(\$mahasiswa as \$mhs):?> 18 <li class="list-group-item"><?= \$mhs['nama'];?> 19 <?php endforeach; ?> 20 21 </div> 22 </div> 23 </div></pre>
15	<p>Jika berhasil maka tampilan akan seperti pada gambar dibawah ini</p> 
16	<p>Jika kita klik Home, halaman Home belum dapat tampil. Maka dari itu kita akan set agar semua dapat saling terhubung. Untuk itu kita membutuhkan yang namanya base_url. untuk lebih jelasnya ikuti langkah-langkah berikut:</p> <p>a. Tambahkan <i>code</i> berikut pada views/template/header (line 26,27,28)</p>



Jobsheet-2: Read dan Input Database serta Form Validation Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

```
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="sha384-NCw98/SFnGE8fJT3GXw
10
11 <!-- <title>Hello, world!</title> -->
12 <title> <? $title> </title>
13 </head>
14 <body>
15
16
17 <!-- navbar -->
18 <nav class="navbar navbar-expand-lg navbar-light bg-light">
19 <a class="navbar-brand" href="#CI">CI</a>
20 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
21 <span class="navbar-toggler-icon"></span>
22 </button>
23
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 </div>
30 </div>
31 </nav>
```

Keterangan :

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

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

```
application > config > autoload.php
78 |
79 | $autoload['drivers'] = array('cache' => 'cch');
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');
```

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



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-2: Read dan Input Database serta Form Validation Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

The screenshot displays two browser windows. The top window, titled 'Home', shows the URL 'localhost/code-kedua/' and a navigation menu with links 'CI', 'Home', 'Data Mahasiswa', and 'About'. The main content area displays 'Hello, !'. The bottom window, titled 'List Mahasiswa', shows the URL 'localhost/code-kedua/mahasiswa' and the same navigation menu. Below the navigation bar, the title 'Daftar Mahasiswa' is followed by a table with three rows of student names: 'dina', 'dika', and 'soni'.

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

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



	<pre>application > views > mahasiswa > index.php 1 <!-- <?php 2 var_dump (\$mahasiswa); 3 ?> --> 4 5 <div class="container"> 6 <div class="row mt-4"> 7 <div class="col-md-6"> 8 Tambah Data 9 </div> 10 </div> 11 <!-- mt-3 artinya margin top 3 --> 12 <div class="row mt-3"> 13 <div class="col-md-6"> 14 <h2>Daftar Mahasiswa</h2> 15 <ul class="list-group"> 16 <!-- <li class="list-group-item">Cras justo odio 17 <li class="list-group-item">Dapibus ac facilisis in 18 <li class="list-group-item">Morbi leo risus 19 <li class="list-group-item">Porta ac consectetur ac 20 <li class="list-group-item">Vestibulum at eros --> 21 22 <?php foreach(\$mahasiswa as \$mhs):?> 23 <li class="list-group-item"><?= \$mhs['nama'];?> 24 <?php endforeach; ?> 25 26 </div> 27 </div> 28 </div></pre>
2	<p>Dikarenakan mahasiswa_model ini nantinya akan digunakan oleh semua <i>function</i> di <i>controller</i> mahasiswa, maka “ \$this->load->model('mahasiswa_model');”; yang awalnya hanya berada pada function index, kita ubah ke function construct agar semua <i>function</i> dapat menggunakan mahasiswa_model.</p>



	<pre>application > controllers > mahasiswa.php 1 <?php 2 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 //digunakan untuk menjalankan fungsi construct pada class parent(c 10 parent::__construct(); 11 \$this->load->model('mahasiswa_model'); 12 } 13 14 15 public function index() 16 { 17 \$this->load->model('mahasiswa_model'); 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 /* End of file mahasiswa.php */ 30 31 ?></pre>
3	<p>Setelah itu tambahkan <i>function</i> tambah pada <i>controller</i> mahasiswa dan tambahkan <i>code</i> berikut.</p> <p>(Tips : jika menggunakan vsCode untuk menduplikat 1 baris dapat menggunakan alt+shift+panah_bawah)</p>



```
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          //digunakan untuk menjalankan fungsi constrauct pada class parrent(ci
10         parent::__construct();
11         $this->load->model('mahasiswa_model');
12     }
13
14
15     public function index()
16     {
17         $this->load->model('mahasiswa_model');
18         //modul untuk load database
19         //$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     public function tambah(){
28         $data['title']='Form Menambahkan Data Mahasiswa';
29         $this->load->view('template/header',$data);
30         $this->load->view('mahasiswa/index',$data);
31         $this->load->view('template/footer');
32     }
33
34 }
35
36 /* End of file mahasiswa.php */
37
38 >>
```

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 | index.php
application > 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="text" class="form-control" id="nim">
18            </div>
19
20            <div class="form-group">
21              <label for="email">Email</label>
22              <input type="text" 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>
33            </div>
34            <button type="submit" name="submit" class="btn btn-primary float-right">Submit</button>
35          </form>
36        </div>
37      </div>
38    </div>
39  </div>
40</div>
```

Keterangan :

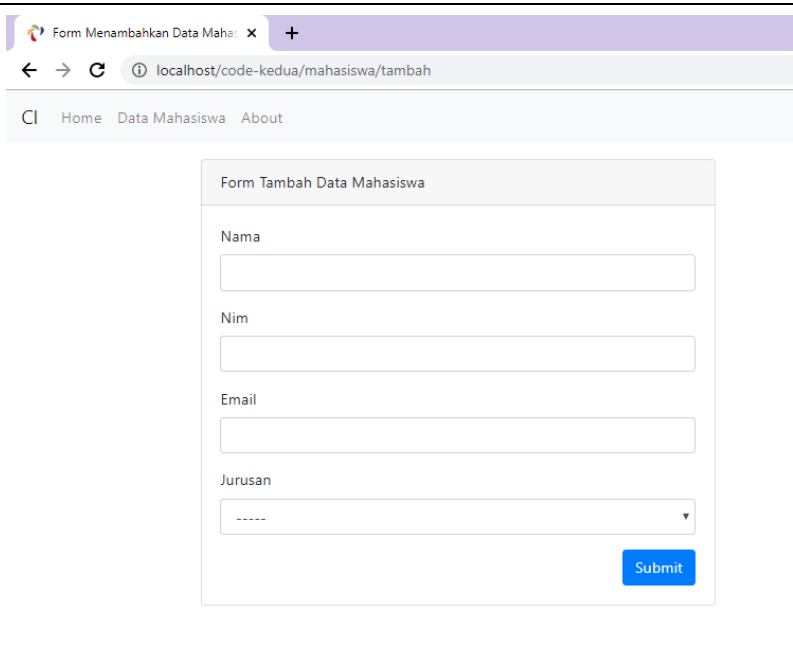
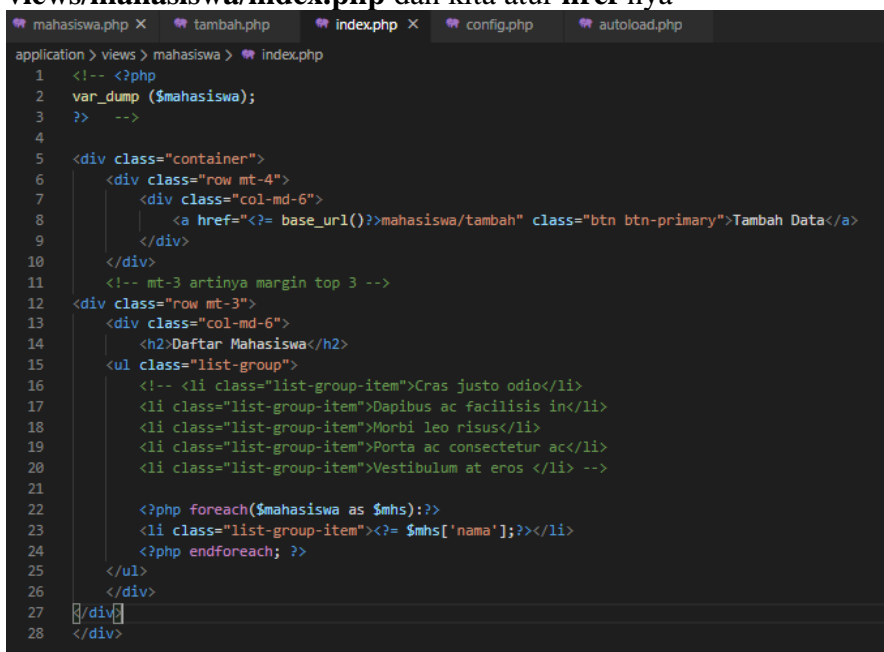
- Pada *line 33*, terdapat atribut class dengan *value* btn btn-primary float-right yang artinya kita membuat sebuah *button* dan posisi buttonnya ada di kanan
- Ganti *value* dari *line 28-31* menjadi seperti di bawah ini (**Agar di tabel tidak tersimpan *value* berupa angka**)

```
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="kimia">Kimia</option>
30     <option value="informatika">Informatika</option>
31     <option value="mesin">Mesin</option>
32   </select>
33 </div>
34 <button type="submit" name="submit" class="btn btn-primary">Submit</button>
35 </form>
```

5

Maka akan dihasilkan halaman seperti di bawah ini.



	
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>  <p>Keterangan :</p> <p>Tambahkan <i>code</i> pada line 8 agar pada saat <i>user</i> melakukan <i>event click</i> maka <i>user</i> akan ditampilkan halaman tambah.</p>
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>



Jobsheet-2: Read dan Input Database serta Form Validation

Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

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

```
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          //digunakan untuk menjalankan fungsi constrauct pada class parrent(c
10         parent::__construct();
11         $this->load->model('mahasiswa_model');
12     }
13
14
15     public function index()
16     {
17         //modul untuk load database
18         //$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/index', $data);
23         $this->load->view('template/footer');
24     }
25
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 ditambahkan";
37         }
38     }
39
40 }
41
42 /* End of file mahasiswa.php */
43
```

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**

```
application > views > mahasiswa > tambah.php
1  ="container">
2  lass="row mt-3">
3  iv class="col-md-6">
4  iv 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" 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="text" 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="kimia">Kimia</option>
30         <option value="informatika">Informatika</option>
31         <option value="mesin">Mesin</option>
32       </select>
33     </div>
34     <button type="submit" name="submit" class="btn btn-primary float-right">
35   </form>
36 </div>
37 </div>
38 </div>
```

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



Jobsheet-2: Read dan Input Database serta Form Validation Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

```
public function tambah() {  
    $data['title']='Form Menambahkan Data Mahasiswa';  
    $this->load->library('form_validation');  
    $this->form_validation->set_rules('nama', 'Nama', 'required');  
  
    if ($this->form_validation->run() == FALSE) {  
        #code..  
        $this->load->view('template/header', $data);  
        $this->load->view('mahasiswa/tambah', $data);  
        $this->load->view('template/footer');  
    }else{  
        #code..  
        echo"data berhasil ditambahkan";  
    }  
}
```

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

```
application > views > mahasiswa > tambah.php  
1  ss="container">  
2  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          <?= validation_errors() ?>  
10         <form action="" method="post">  
11             <div class="form-group">  
12                 <label for="nama">Nama</label>
```

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



Form Validation — CodeIgniter 3 x Form Menambahkan Data Mahasiswa x +

localhost/code-kedua/mahasiswa/tambah

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

```
application > 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           <?php if (validation_errors());?>
10          <div class="alert alert-danger" role="alert">
11            <?= validation_errors() ?>
12          </div>
13          <?php endif ?>
14          <form action="" method="post">
15            <div class="form-group">
```

i. Akan dihasilkan seperti gambar di bawah ini



Form Menambahkan Data Mahasiswa

← → ↻ localhost/code-kedua/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**.

```
23     $this->load->view('template/footer');  
24 }  
25  
26 public function tambah() {  
27     $data['title'] = 'Form Menambahkan Data Mahasiswa';  
28     $this->load->library('form_validation');  
29     $this->form_validation->set_rules('nama', 'Nama', 'required');  
30     $this->form_validation->set_rules('nim', 'Nim', 'required');  
31     $this->form_validation->set_rules('email', 'Email', 'required');  
32  
33     if ($this->form_validation->run() == FALSE) {  
34         #code..  
35         $this->load->view('template/header', $data);  
36         $this->load->view('mahasiswa/tambah', $data);  
37         $this->load->view('template/footer');  
38     } else {  
39         #code..  
40         echo "data berhasil ditambahkan";  
    }
```

- k. Hasilnya seperti berikut ini.



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-2: Read dan Input Database serta Form Validation

Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

Form Menambahkan 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)

```
public function tambah() {  
    $data['title']='Form Menambahkan Data Mahasiswa';  
    $this->load->library('form_validation');  
    $this->form_validation->set_rules('nama', 'Nama', 'required');  
    $this->form_validation->set_rules('nim', 'Nim', 'required');  
    $this->form_validation->set_rules('email', 'Email', 'required|valid_email');
```

m. Hasilnya sebagai berikut.



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-2: Read dan Input Database serta Form Validation

Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

Form Tambah Data Mahasiswa

The Email field must contain a valid email address.

Nama

Nim

Email

Jurusan

Submit

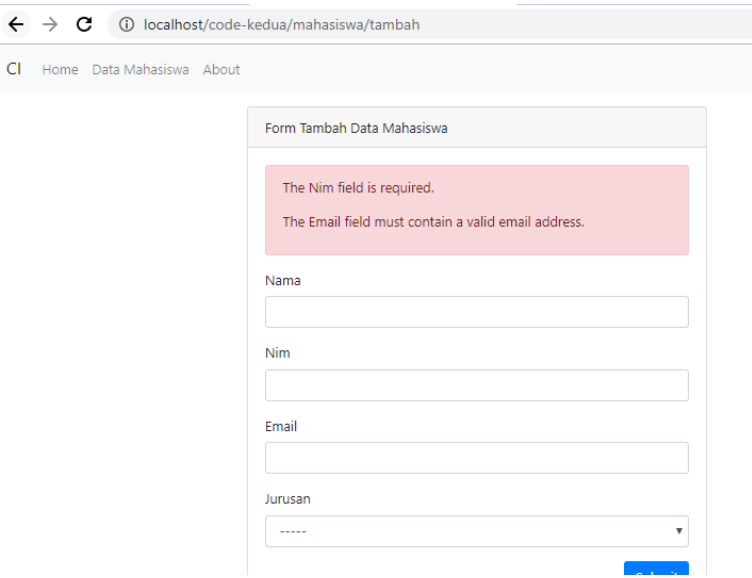
n. Selain itu kita juga dapat memberikan *rules* harus diisi dengan angka menggunakan *rules numeric*

```
public function tambah() {  
    $data['title']='Form Menambahkan Data Mahasiswa';  
    $this->load->library('form_validation');  
    $this->form_validation->set_rules('nama', 'Nama', 'required');  
    $this->form_validation->set_rules('nim', 'Nim', 'required|numeric');  
    $this->form_validation->set_rules('email', 'Email', 'required|valid_email');  
  
    if ($this->form_validation->run() == FALSE) {  
        #code..  
        $this->load->view('template/header', $data);  
        $this->load->view('mahasiswa/tambah', $data);  
        $this->load->view('template/footer');  
    }else{  

```

o. Akan dihasilkan alert seperti berikut ini.




<p>Saat nim diisi dengan huruf tidak bisa karena telah diset menggunakan angka, sehingga ketika diinput oleh huruf dalam kolom nim tidak bisa digunakan.</p>

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 code berikut pada function tambah agar pada saat form_validation bernilai true, maka akan ditambahkan data dengan nama modelnya -> model_mahasiswa dengan nama function tambahdatamhs() dan jika sudah ditambahkan ke database maka akan redirect ke controller mahasiswa (<i>line 46 - 47</i>)</p> <pre>26 public function tambah() { 27 \$data['title']='Form Menambahkan Data Mahasiswa'; 28 \$this->load->library('form_validation'); 29 \$this->form_validation->set_rules('nama', 'Nama', 'required'); 30 \$this->form_validation->set_rules('nim', 'Nim', 'required numeric'); 31 \$this->form_validation->set_rules('email', 'Email', 'required valid_email'); 32 33 if (\$this->form_validation->run() == FALSE) { 34 #code.. 35 \$this->load->view('template/header', \$data); 36 \$this->load->view('mahasiswa/tambah', \$data); 37 \$this->load->view('template/footer'); 38 }else{ 39 #code.. 40 \$this->mahasiswa_model->tambahdatamhs(); 41 redirect('mahasiswa', 'refresh'); 42 } 43 } 44 45 }</pre>



Jobsheet-2: Read dan Input Database serta Form Validation Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

2	Jika kita lihat <i>code</i> pada controller mahasiswa (line 46) terdapat model dengan nama mahasiswa_model dan <i>function</i> -nya tambahdatamhs() . Untuk itu kita harus membuat sebuah <i>function</i> dengan nama tambahdatamhs()
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 <p>Keterangan:</p> <ul style="list-style-type: none">- 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)<ul style="list-style-type: none">• nama->alias• Parameter nama pada post adalah name pada html seperti pada gambar dibawah ini• Parameter <i>true</i> digunakan untuk mengaktifkan htmlspecialchars• Line 29 adalah proses <i>insert</i> data ke database dengan 2 parameter yaitu nama tablenya dan objectnya (dalam hal ini data yang akan dimasukkan ke database <pre>public function tambahdatamhs(){ \$data=["nama" => \$this->input->post('nama',true), //ini sama dengan html "nim" => \$this->input->post('nim',true), "email" => \$this->input->post('email',true), "jurusan" => \$this->input->post('jurusan',true)]; //\$this->db->insert('Table', \$object); \$this->db->insert('mahasiswa', \$data); } /* End of file mahasiswa_model.php */</pre>
4	<p>Langkah berikutnya kita coba untuk tambahkan data.</p> <p>Jika berhasil maka data akan tampil di halaman index.php</p> <p>a. <code>redirect('mahasiswa','refresh');</code>(line 47 gambar <i>controller mahasiswa</i>)</p> <p>b. Hasil</p>



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-2: Read dan Input Database serta Form Validation

Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Web Lanjut

Februari 2020

Form Menambahkan Data Mahas x +

localhost/code-kedua/mahasiswa/tambah

CI Home Data Mahasiswa About

Form Tambah Data Mahasiswa

Nama

millan

Nim

1841720081

Email

millania@gmail.com

Jurusan

Informatika

Submit

List Mahasiswa x +

localhost/code-kedua/mahasiswa

CI Home Data Mahasiswa About

Tambah Data

Daftar Mahasiswa

dina

dika

soni

millan