

Jobsheet-2: Read dan Input Database serta Form Validation

Mata Kuliah Pemrograman Web Lanjut



Oleh :

**ALAN PERDHANA TIMOR / NIM : 1841720187
KELAS TI 2A**

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020**

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

Langkah	Keterangan
1	Buka kembali folder CodeIgniter yang telah dibuat pada Jobsheet sebelumnya
2	<p>Buka <i>controller</i> mahasiswa dan tambahkan <i>code</i> berikut (<i>line 9</i>)</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 /* End of file Controllername.php */ 19 20 21 22 23 ?> </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>)

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed')

class mahasiswa extends CI_Controller {

    //fungsi yang dijalankan saat classnya diinstansiasi
    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function index()
    {
        //modul untuk load database
        // $this->load->database();

        $this->load->database();
        $data['title']='list mahasiswa';
        $this->load->view('template/header',$data);
        $this->load->view('mahasiswa/index');
        $this->load->view('template/footer');
    }
}

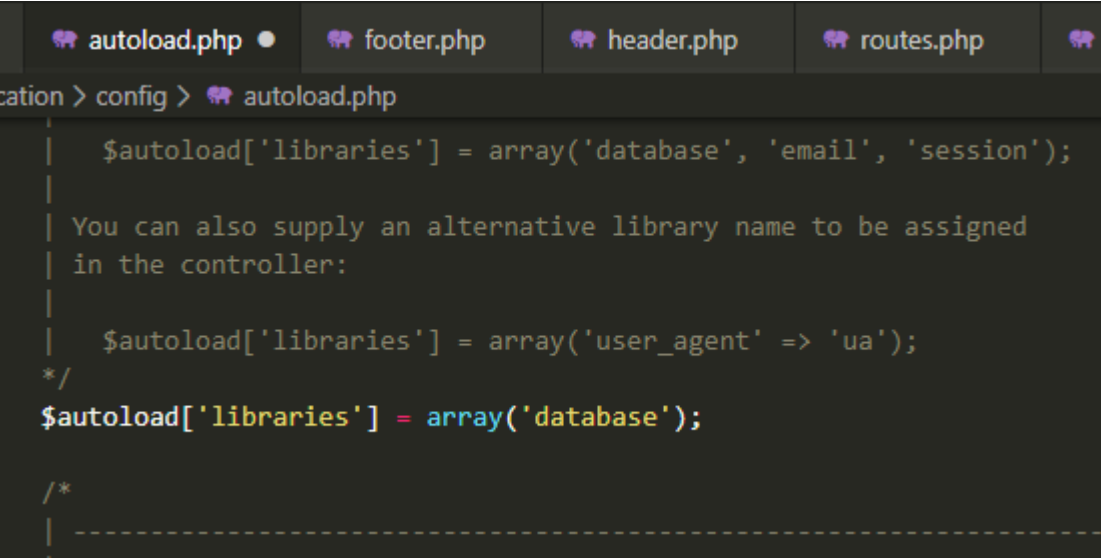
```

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

	 <pre> application > config > autoload.php \$autoload['libraries'] = array('database', 'email', 'session'); You can also supply an alternative library name to be assigned in the controller: \$autoload['libraries'] = array('user_agent' => 'ua'); */ \$autoload['libraries'] = array('database'); /* ----- </pre>
5	<p>Untuk kasus ini kita gunakan cara yang terakhir (no 4) karena kita asumsikan bahwa semua <i>controller</i> membutuhkan database. Maka kita <i>comment code</i> function construct yang ada di <i>controller</i> mahasiswa</p>  <pre> class mahasiswa extends CI_Controller { //fungsi yang dijalankan saat classnya diinstansiasi //public function __construct() //{ // //digunakan untuk menjalankan fungsi construct pada class parent(ci_controller) // parent::__construct(); //\$this->load->database(); //} } </pre>
6	<p>Setelah itu kita akan mengatur konfigurasi database. Untuk mengatur konfigurasi database maka buka folder application/config/database.php</p>

	<div data-bbox="295 100 1388 1064"> <div> > cache ✓ config 🐘 autoload.php 🐘 config.php 🐘 constants.php 🐘 database.php 🐘 doctypes.php 🐘 foreign_chars.php 🐘 hooks.php ☰ index.html 🐘 memcached.php 🐘 migration.php 🐘 mimes.php 🐘 profiler.php 🐘 routes.php 🐘 smileys.php 🐘 user_agents.php ✓ controllers 🐘 home.php ☰ index.html 🐘 mahasiswa.php 🐘 Welcome.php </div> <div> 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 !== 'product 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); </div> </div>
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' => '',
81     'database' => 'Kuliah CI',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'product
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

✓ Table mahasiswa has been altered successfully.

```
ALTER TABLE `mahasiswa` CHANGE `id` `id` INT(30) NOT NULL AUTO_INCREMENT;
```

[Edit inline] [Edit]

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

☐ Check all With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns
 Remove from central columns

9

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

	<div> <input type="checkbox"/> Show all Number of rows: 25 Filter rows: <input type="text" value="Search this table"/> </div> <div> + Options <table> <tr> <th></th><th></th><th></th><th></th><th>id</th><th>nama</th><th>nim</th><th>email</th><th>jurusan</th></tr> <tr> <td><input type="checkbox"/></td><td></td><td></td><td></td><td>1</td><td>Dina</td><td>1002345100</td><td>dina@gamil.com</td><td>kimia</td></tr> <tr> <td><input type="checkbox"/></td><td></td><td></td><td></td><td>2</td><td>dika</td><td>1002345101</td><td>dika@gmail.com</td><td>informatika</td></tr> <tr> <td><input type="checkbox"/></td><td></td><td></td><td></td><td>3</td><td>soni</td><td>1002345102</td><td>soni@gmail.com</td><td>mesin</td></tr> </table> <div> <input type="checkbox"/> Check all With selected: Edit Copy Delete Export </div> </div>					id	nama	nim	email	jurusan	<input type="checkbox"/>				1	Dina	1002345100	dina@gamil.com	kimia	<input type="checkbox"/>				2	dika	1002345101	dika@gmail.com	informatika	<input type="checkbox"/>				3	soni	1002345102	soni@gmail.com	mesin
				id	nama	nim	email	jurusan																													
<input type="checkbox"/>				1	Dina	1002345100	dina@gamil.com	kimia																													
<input type="checkbox"/>				2	dika	1002345101	dika@gmail.com	informatika																													
<input type="checkbox"/>				3	soni	1002345102	soni@gmail.com	mesin																													

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

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

```

CODE-KEDUA
├── controllers
│   ├── home.php
│   ├── index.html
│   ├── mahasiswa.php
│   └── Welcome.php
├── core
├── helpers
├── hooks
├── language
├── libraries
├── logs
├── models
│   ├── index.html
│   └── mahasiswa_model.php
└── third_party
        
```

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

```

application > models > mahasiswa_model.php
1  <?php
2
3
4  defined('BASEPATH') OR exit('No direct script access allowed');
5
6  class mahasiswa_model extends CI_Controller {
7
8      public function index()
9      {
10         //- https://www.codeigniter.com/user_guide/database/query_builder.html#selecting
11         //$query digunakan untuk menampung isi dari tabel mahasiswa
12         $query=$this->db->get('mahasiswa');
13
14         //- https://www.codeigniter.com/user_guide/database/results.html
15         return $query->result_array();
16     }
17
18 }
19
20 /* End of file Controllername.php */
21
22

```

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

```

//atau bisa juga meggunakan code berikut
return $this->db->get('mahasiswa')->result_array();
}
}

```

11

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


```

public function index()
{
    //modul untuk load database
    // $this->load->database();

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

```

Keterangan :

- Line 17 digunakan untuk melakukan *load* model dengan nama modelnya adalah **mahasiswa_model**
- Line 22 artinya ada data yang berupa array dengan isi arraynya adalah mahasiswa dimana terdapat isi berupa function **getAllmahasiswa()** yang ada pada model dengan nama **mahasiswa_model**
- Line 24 tambahkan **\$data**

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class mahasiswa_model extends CI_Controller {

    public function index()
    {

        //$this->load->database();
        $data['title']='list mahasiswa';
        $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
    }
}

```

12

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

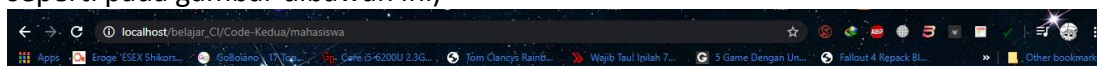
```

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
10         <ul class="list-group">
11             <li class="list-group-item">Cras justo odio</li>
12             <li class="list-group-item">Dapibus ac facilisis</li>
13             <li class="list-group-item">Morbi leo risus</li>
14             <li class="list-group-item">Porta ac consectetur ac</li>
15             <li class="list-group-item">vestibulum at eros</li>
16         </ul>
17     </div>
18 </div>
19 </div>
20 </div>

```

Keterangan

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



A PHP Error was encountered

Severity: Warning
 Message: mysqli::real_connect(): (HY000/1045): Access denied for user 'root'@'localhost' (using password: NO)
 Filename: mysqli_driver.php
 Line Number: 203
 Backtrace:
 File: C:\Program Files\Ampms\www\belajar_CI\Code-Kedua\index.php
 Line: 315
 Function: require_once

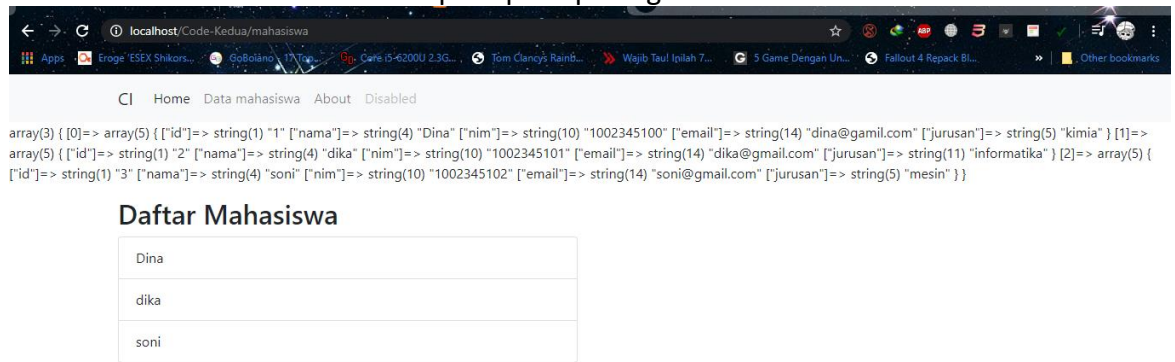
A Database Error Occurred

Unable to connect to your database server using the provided settings.
 Filename: C:\Program Files\Ampms\www\belajar_CI\Code-Kedua\system\database\DB_driver.php
 Line Number: 436

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*.

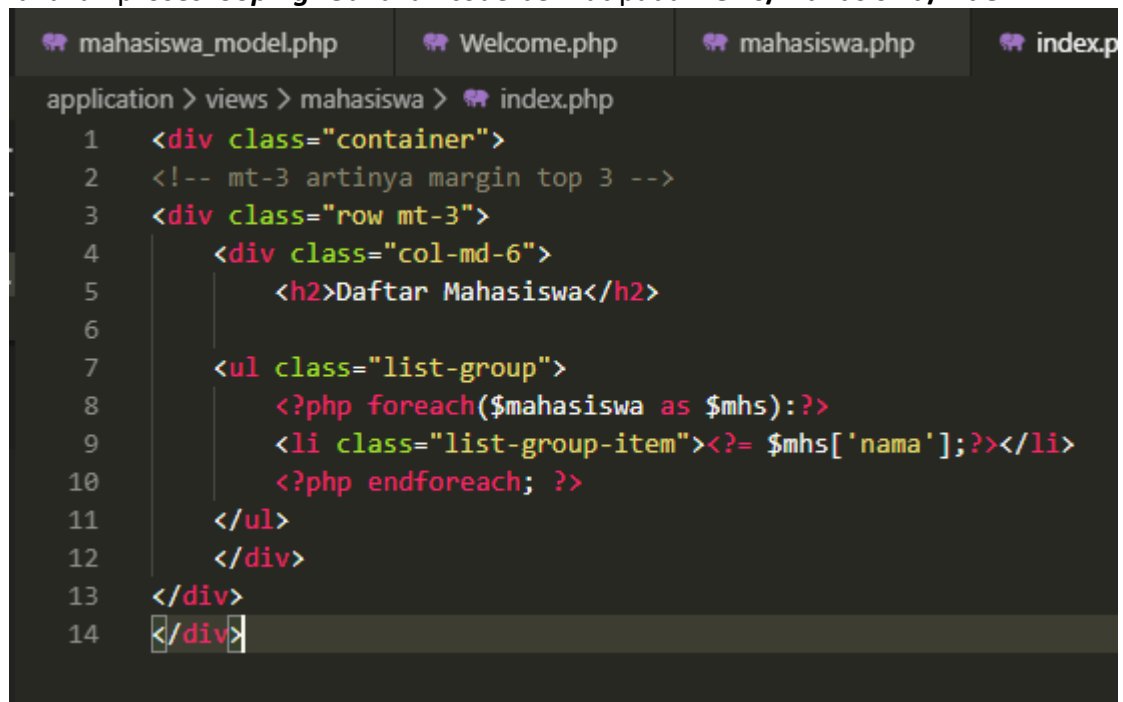
13

Jika berhasil maka data akan tampil seperti pada gambar berikut



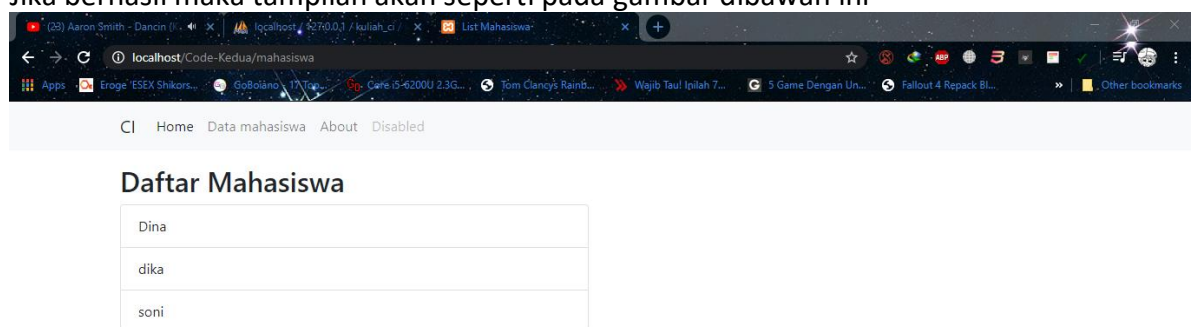
14

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)

```

mahasiswa_model.php Welcome.php mahasiswa.php index.php header.php ● config.php autolo
application > views > template > header.php
15
16 <!-- navbar cuy -->
17 <nav class="navbar navbar-expand-lg navbar-light bg-light">
18   <div class = "container">
19     <a class="navbar-brand" href="#">CI</a>
20     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-
21     <span class="navbar-toggler-icon"></span>
22   </button>
23   <div class="collapse navbar-collapse" id="navbarNav">
24     <ul class="navbar-nav">
25       <li class="nav-item active">
26         <a class="nav-link" href="{?= base_url(); ?}">Home <span class="sr-only">(current)</span></a>
27       </li>
28       <li class="nav-item">
29         <a class="nav-link" href="{?= base_url(); ?}">mahasiswa</a>
30       </li>
31       <li class="nav-item">
32         <a class="nav-link" href="{?= base_url(); ?}">About</a>
33       </li>
34       <li class="nav-item">
35         <!--a class="nav-link disabled" href="#">Disabled</a -->
36       </li>
37     </ul>
38   </div>
39 </div>
40 </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)

```

| a PHP script and you can easily do that on your own.
|
| */
$config['base_url'] = 'http://localhost/Code-Kedua/mahasiswa';
/*

```

- 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')`

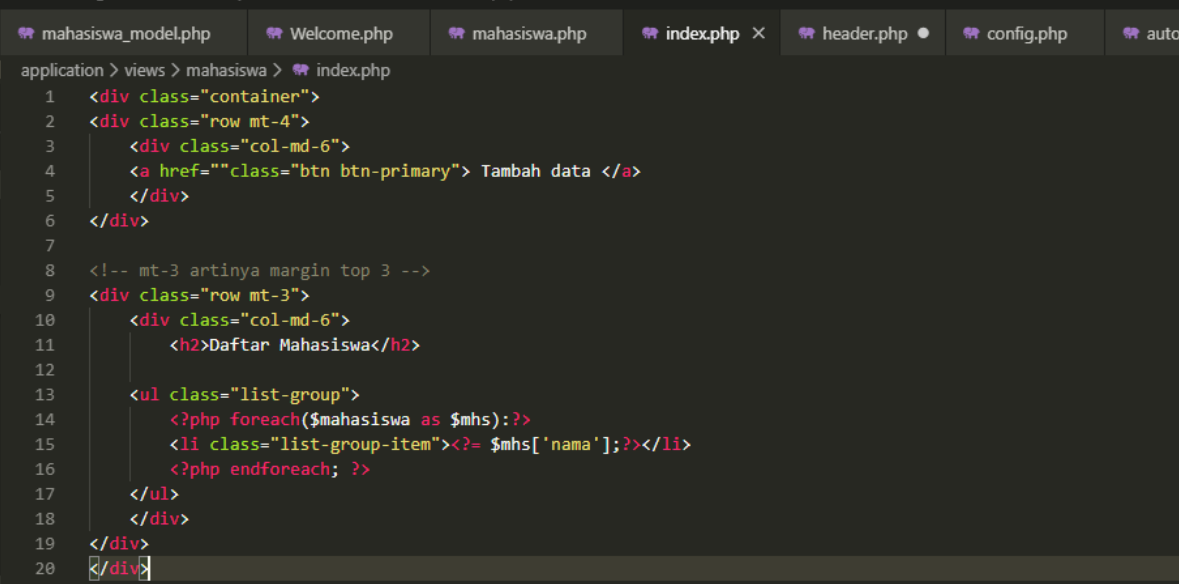
```

File Edit Selection View Go Debug Terminal Help
autoload.php - Code-Kedua - Visual Studio Code
EXPLORER
mahasiswa_model.php Welcome.php mahasiswa.php index.php header
OPEN EDITO... 1 UNSAVED application > config > autoload.php
mahasiswa_model... 88 | Prototype:
Welcome.php app... 89 |
mahasiswa.php a... 90 | $autoload['helper'] = array('url', 'file');
index.php applicat... 91 | */
header.php applic... 92 | $autoload['helper'] = array('url');
config.php applica... 93 |
autoload.php app... 94 | /*
CODE KEDUA 95 | -----
96 | Auto-load Config files

```

- 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> application > views > mahasiswa > index.php 1 <div class="container"> 2 <div class="row mt-4"> 3 <div class="col-md-6"> 4 Tambah data 5 </div> 6 </div> 7 8 <!-- mt-3 artinya margin top 3 --> 9 <div class="row mt-3"> 10 <div class="col-md-6"> 11 <h2>Daftar Mahasiswa</h2> 12 13 <ul class="list-group"> 14 <?php foreach(\$mahasiswa as \$mhs):?> 15 <li class="list-group-item"><?= \$mhs['nama'];?> 16 <?php endforeach; ?> 17 18 </div> 19 </div> 20 </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>

```
mahasiswa_model.php Welcome.php mahasiswa.php x index.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 __construct()
8      {
9          parent::__construct();
10         $this->load->model('mahasiswa_model');
11     }
12     public function index()
13     {
14
15         //$this->load->database();
16         $data['title']='List Mahasiswa';
17         $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
18         $this->load->view('template/header',$data);
19         $this->load->view('mahasiswa/index',$data);
20         $this->load->view('template/footer');
21     }
22
23 }
24
25 /* End of file Controllername.php */
26
27
28 ?>
```

- 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**)

```

22
23     public function tambah() {
24
25         $data['title']='Form menambahkan data mahasiswa';
26         $this->load->view('template/header',$data);
27         $this->load->view('mahasiswa/tambah',$data);
28         $this->load->view('template/footer');
29     }
30
31
32
33
34     End of file Controllername.php */
35
36

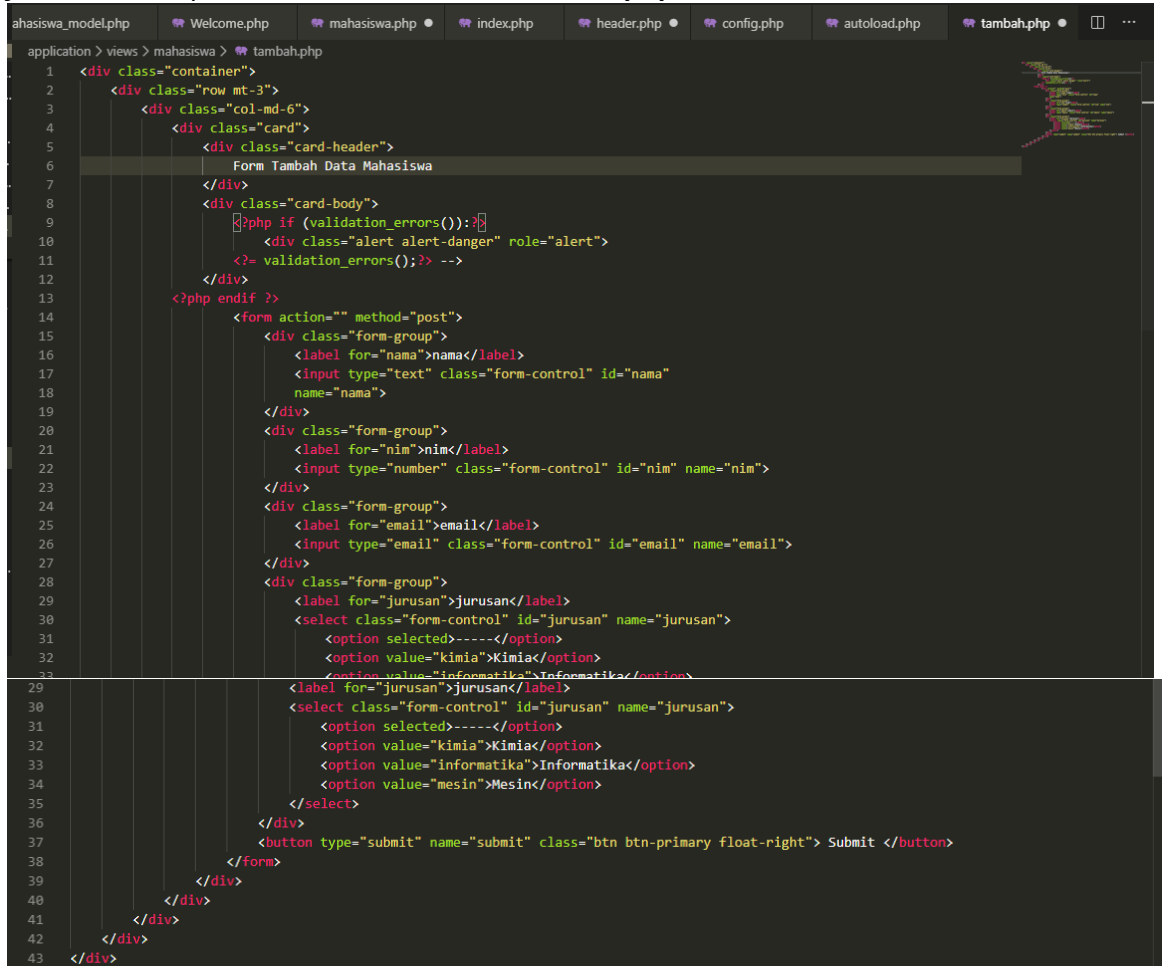
```

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`



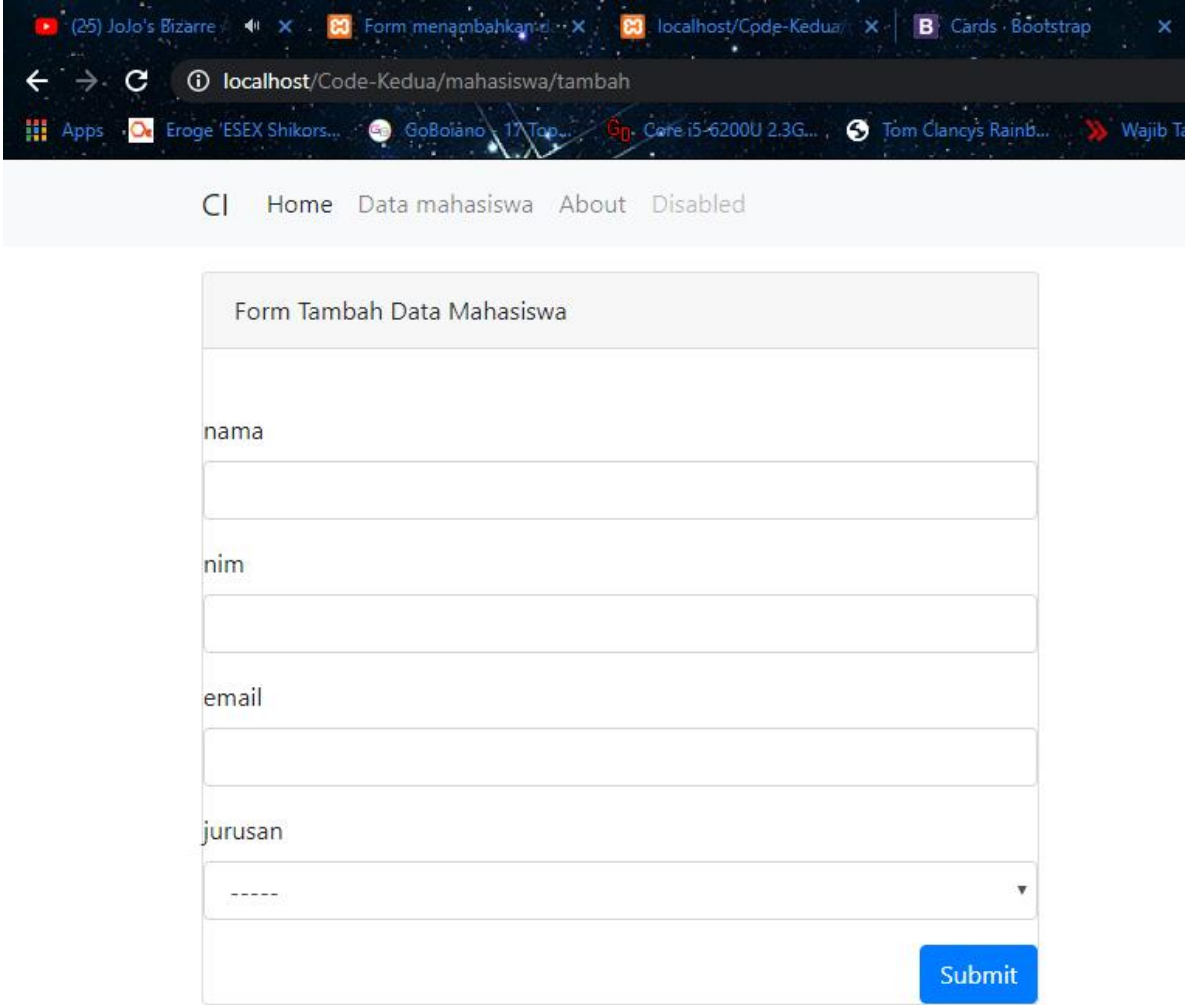
```

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                      <?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">
16                             <label for="nama">nama</label>
17                             <input type="text" class="form-control" id="nama"
18                                 name="nama">
19                         </div>
20                         <div class="form-group">
21                             <label for="nim">nim</label>
22                             <input type="number" class="form-control" id="nim" name="nim">
23                         </div>
24                         <div class="form-group">
25                             <label for="email">email</label>
26                             <input type="email" class="form-control" id="email" name="email">
27                         </div>
28                         <div class="form-group">
29                             <label for="jurusan">jurusan</label>
30                             <select class="form-control" id="jurusan" name="jurusan">
31                                 <option selected>-----</option>
32                                 <option value="kimia">Kimia</option>
33                                 <option value="informatika">Informatika</option>
34                                 <option value="mesin">Mesin</option>
35                             </select>
36                         </div>
37                         <button type="submit" name="submit" class="btn btn-primary float-right"> Submit </button>
38                     </form>
39                 </div>
40             </div>
41         </div>
42     </div>
43 </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**)

	<pre> <label for="jurusan">jurusan</label> <select class="form-control" id="jurusan" name="jurusan"> <option selected>-----</option> <option value="kimia">Kimia</option> <option value="informatika">Informatika</option> <option value="mesin">Mesin</option> </select> </pre>
5	<p>Maka akan dihasilkan halaman seperti di bawah ini.</p> 
6	<p>Setelah berhasil membuat <i>form</i> tambah data, selanjutnya buka file index pada views/mahasiswa/index.php dan kita atur href nya</p>


```

Welcome.php  mahasiswa.php  index.php x  index.html  header.php  config
application > views > mahasiswa > index.php
1  <div class="container">
2  <div class="row mt-4">
3      <div class="col-md-6">
4          <a href="php= base_url(); ?/tambah" class="btn btn-primary"> Tambah data </a>
5      </div>
6  </div>
7
8  <!-- mt-3 artinya margin top 3 -->
9  <div class="row mt-3">
10     <div class="col-md-6">
11         <h2>Daftar Mahasiswa</h2>
12
13         <ul class="list-group">
14             <?php foreach($mahasiswa as $mhs):?>
15                 <li class="list-group-item"><?php= $mhs['nama'];?></li>
16             <?php endforeach; ?>
17         </ul>
18     </div>
19 </div>
20 </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*

```

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

```

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**

```
<form action="" method="post">
<!-- https://getbootstrap.com/docs/4.0/components/card/ -->
<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="kimia">Kimia</option>
    <option value="informatika">Informatika</option>
    <option value="mesin">Mesin</option>
  </select>
</div>
```

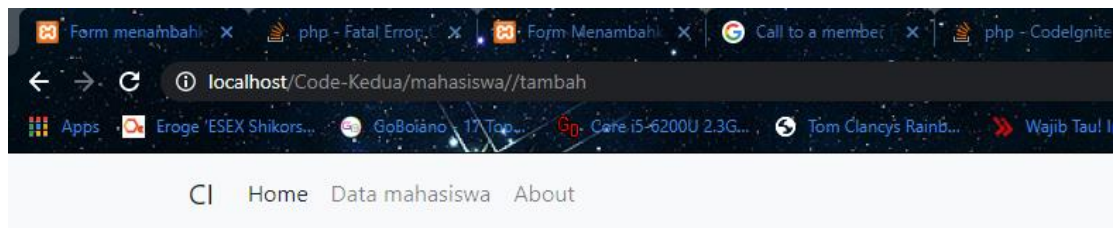
e. Setelah kita tambahkan **atribut name** maka tambahkan **rule** pada **controller/mahasiswa** seperti pada gambar dibawah ini (line 35-46)

```
public function tambah() {
  if ($this->form_validation->run() == FALSE) {
    $this->form_validation->set_rules('nama', 'nama', 'required');
    # code...
```

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

```
<div class="alert alert-danger role="alert">
  <?= validation_errors();?>
</div>
<?php endif ?>
<form action="" method="post">
```

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



Form Tambah Data Mahasiswa

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/>

```

<div class="card-header">
    Form Tambah Data Mahasiswa
</div>
<div class="card-body">
    <?php if (validation_errors()):?>
        <div class="alert alert-danger" role="alert">
            <?= validation_errors();?>
        </div>
    </if>

```

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

i. Akan dihasilkan seperti gambar di bawah ini

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**.

```
$this->load->view( 'template/footer' );
}
public function tambah() {
    $data['title']='Form Menambahkan Data Mahasiswa';
    $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');
```

- 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)**

```

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

```

- m. Hasilnya sebagai berikut.

Form Tambah Data Mahasiswa

Nama

Nim

email

Please enter an email address.

Submit

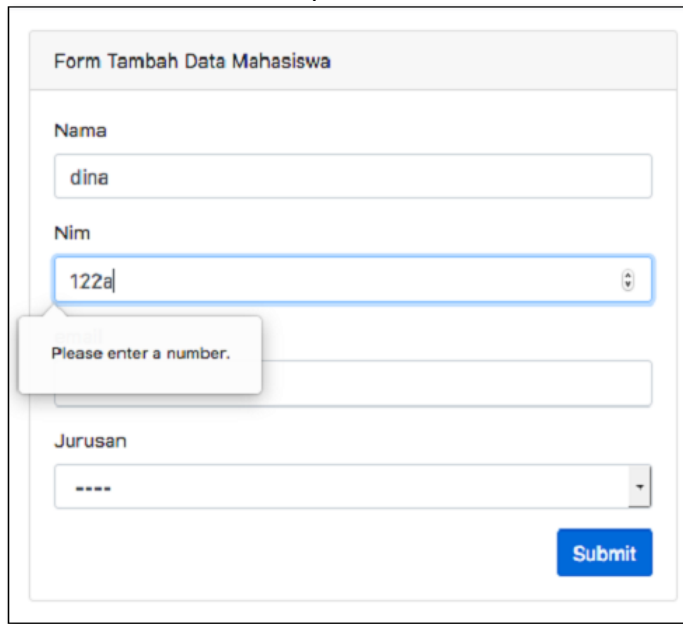
- n. Selain itu kita juga dapat memberikan *rules* harus diisi dengan angka menggunakan **rules numeric**

```

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

```

o. Akan dihasilkan alert seperti berikut ini.



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	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>)

```

30     public function tambah() {
31
32         $data['title'] = 'Form Menambahkan Data Mahasiswa';
33
34         // $this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_length[5]|max_length[12]
35         $this->form_validation->set_rules('nama', 'Nama', 'required');
36         $this->form_validation->set_rules('nim', 'Nim', 'required|numeric');
37         $this->form_validation->set_rules('email', 'Email', 'required|valid_email');
38
39         if ($this->form_validation->run() == FALSE) {
40             # code...
41             $this->load->view('template/header', $data);
42             $this->load->view('mahasiswa/tambah', $data);
43             $this->load->view('template/footer');
44         } else {
45             # code...
46             $this->mahasiswa_model->tambahdatamhs();
47             redirect('mahasiswa','refresh');
48         }
49     }
50 }

```

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

- 3 Buatlah sebuah *function* **tambahdatamhs()** pada **models/mahasiswa_model**
- https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=insert#inserting-data
 - Tambahkan *code* seperti pada gambar dibawah ini

```

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

```

Keterangan:

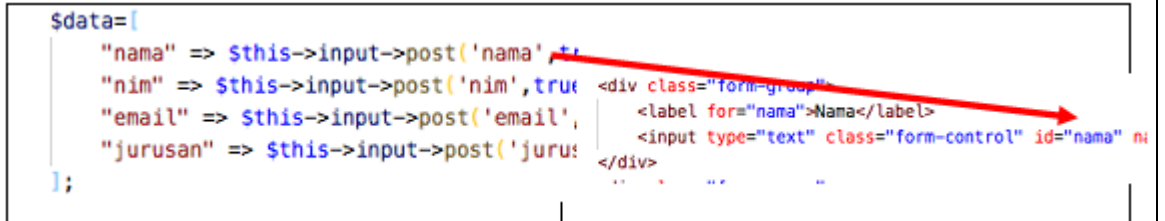
- Line 22-27 adalah sebuah array dimana nantinya isi array adalah data-data yang akan dikirim (POST) ke database
- Line 23 "nama" => \$this->input->post('nama',true)
 - nama->alias
 - Parameter nama pada post adalah name pada html seperti pada gambar dibawah ini

- Parameter *true* digunakan untuk mengaktifkan htmlspecialchars

```

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

```



The screenshot shows a PHP array \$data being populated with POST data. A red arrow points from the 'true' parameter in the 'nama' field to the HTML form input field, indicating that the 'true' parameter is used to enable htmlspecialchars for the 'nama' field.

- 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

[Tambah Data](#)

Daftar Mahasiswa

dina
dika
soni
dino
ani

-- Selamat Mengerjakan --