

# **LAPORAN PEMROGRAMAN WEB LANJUT**

## **JOBSHEET 2**

### **READ DAN INPUT DATABASE SERTA FORM VALIDATION**

**Oleh:**

**KINANTI PERMATA PUTRI**

**NIM. 1841720022**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

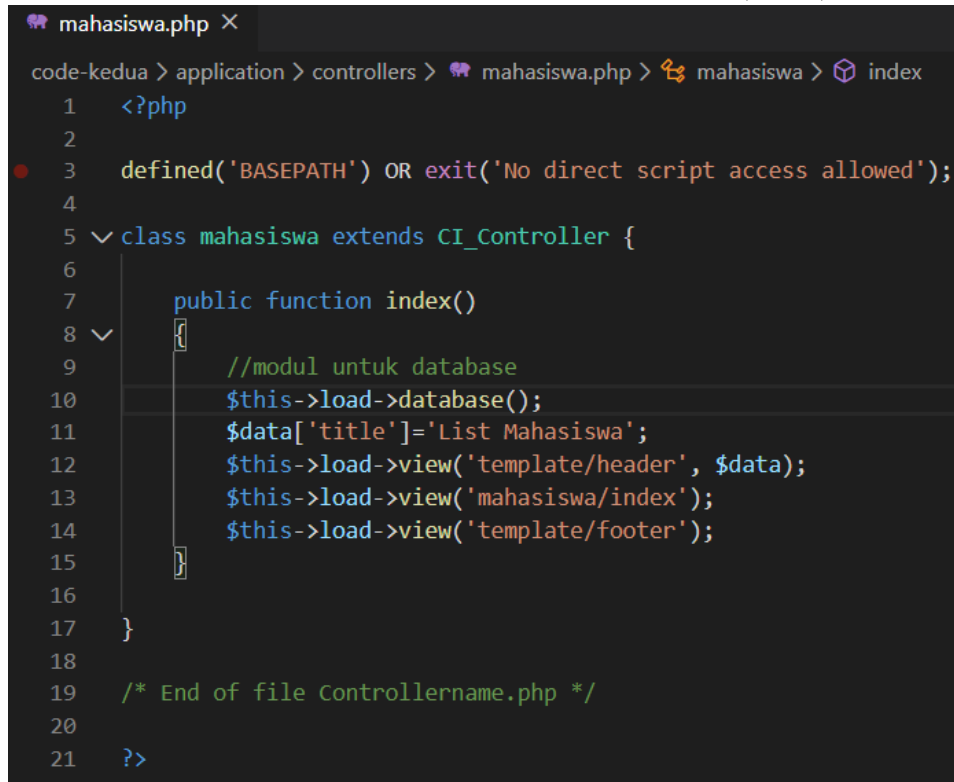
**POLITEKNIK NEGERI MALANG**

**FEBRUARI 2020**

## A. Praktikum

### 1. Praktikum – Bagian 1: Menampilkan data dari database menggunakan codeIgniter

- 1) Buka kembali folder CodeIgniter yang telah dibuat pada Jobsheet sebelumnya
- 2) Buka *controller* mahasiswa dan tambahkan *code* berikut (*line 9*)



```
code-kedua > application > controllers > mahasiswa.php > mahasiswa > index
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 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
21 ?>
```

Keterangan:

- Pada line 9 menjelaskan bawa *function* index ini akan melakukan *load* database. *Load* database ini hanya dilakukan di *function* index saja jika ada *function* lain maka kita harus melakukan *load* database lagi. Sehingga hal ini dirasa tidak praktis maka untuk itu kita harus tambahkan *code* agar *load* database ini dapat digunakan banyak *function*.

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

```
mahasiswa.php x
code-kedua > 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     public function index()
15     {
16         //modul untuk database
17         $this->load->database();
18         $data['title']='List Mahasiswa';
19         $this->load->view('template/header', $data);
20         $this->load->view('mahasiswa/index');
21         $this->load->view('template/footer');
22     }
23 }
24
25
26 /* End of file Controllername.php */
27
28
```

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

```
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 autoload.php
code-kedua > application > controllers > mahasiswa.php > mahasiswa
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     public function index()
15     {
16         //modul untuk database
17         $this->load->database();
18         $data['title']='List Mahasiswa';
19         $this->load->view('template/header', $data);
20         $this->load->view('mahasiswa/index');
21         $this->load->view('template/footer');
22     }
23 }
24
25
26 /* End of file Controllername.php */
27
28 >>
```

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

14BELAJARCI	63		CodeIgniter will store the SQL statement
code-kedua	64		However, this may cause high memory usage
application	65		a lot of SQL queries ... disable this to
cache	66		
config	67		The \$active_group variable lets you choose which connect
autoload.php	68		make active. By default there is only one group (the 'default')
config.php	69		
constants.php	70		The \$query_builder variable lets you determine whether
database.php	71		the query builder class.
	72		*/
doctypes.php	73		\$active_group = 'default';
foreign_chars.php	74		\$query_builder = TRUE;
hooks.php	75		
index.html	76		\$db['default'] = array(
memcached.php	77		'dsn' => '',
migration.php	78		'hostname' => 'localhost',
mimes.php	79		'username' => '',
profiler.php	80		'password' => '',
routes.php	81		'database' => '',
smileys.php	82		'dbdriver' => 'mysqli',
user_agents.php	83		'dbprefix' => '',
controllers	84		'pconnect' => FALSE,
core	85		'db_debug' => (ENVIRONMENT !== 'production'),
helpers	86		'cache_on' => FALSE,
hooks	87		'cachedir' => '',
language	88		'char_set' => 'utf8',
libraries	89		'dbcollat' => 'utf8_general_ci',
logs	90		'swap_pre' => '',
models	91		'encrypt' => FALSE,
third_party	92		'compress' => FALSE,
views	93		'stricton' => FALSE,
OUTLINE	94		'failover' => array(),
	95		'save_queries' => TRUE
	96		);
	97		

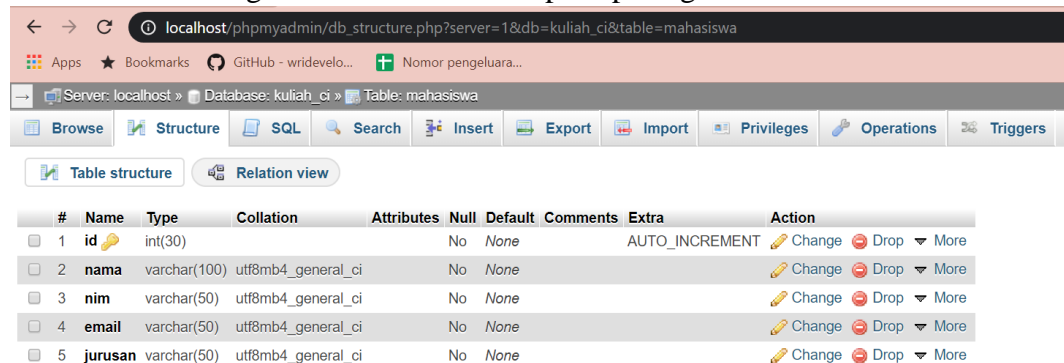
7) Kita set konfigurasinya seperti pada gambar dibawah ini

```
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
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 buat lah sebuah database dengan nama kuliah\_CI dan dengan tabel mahasiswa seperti pada gambar dibawah ini



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
2	nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	nim	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
4	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
5	jurusan	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

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

Showing rows 0 - 2 (3 total, Query took 0.0020 seconds.)

```
SELECT * FROM `mahasiswa`
```

☐ Profiling [Edit inline] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼

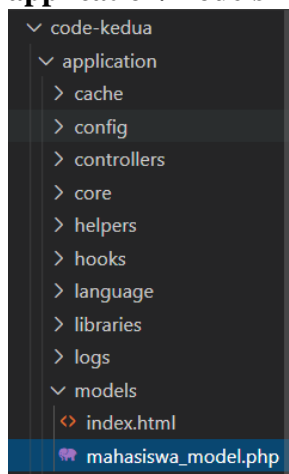
+ Options

				id	nama	nim	email	jurusan
<input type="checkbox"/>	Edit	Copy	Delete	1	dina	1002345100	dina@gmail.com	kimia
<input type="checkbox"/>	Edit	Copy	Delete	2	dika	1002345101	dika@gmail.com	informatika
<input type="checkbox"/>	Edit	Copy	Delete	3	soni	1002345102	soni@gmail.com	mesin

↑ ☐ Check all With selected: Edit Copy Delete Export

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

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



b) Tambahkan *function* getAllmahasiswa pada mahasiswa\_model.php

```
mahasiswa.php database.php mahasiswa_model.php × autoload.php
code-kedua > application > models > mahasiswa_model.php > mahasiswa_model
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          // https://www.codeigniter.com/user_guide/database/query_builder.html#selecting-data
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.html
13         // untuk menampilkan isi dari query
14         return $query->result_array();
15     }
16
17 }
18
19 /* End of file of mahasiswa_model.php */
20
21 ?>
```

Keterangan:

- Pada *line* 11 membuat sebuah *query builder* yang digunakan untuk mengambil semua data  
[https://www.codeigniter.com/user\\_guide/database/query\\_builder.html#selecting-data](https://www.codeigniter.com/user_guide/database/query_builder.html#selecting-data)
- Pada *line* 15 digunakan untuk mengenerate hasil query dalam bentuk array  
[https://www.codeigniter.com/user\\_guide/database/results.html](https://www.codeigniter.com/user_guide/database/results.html)  
[https://www.codeigniter.com/user\\_guide/database/results.html#result-arrays](https://www.codeigniter.com/user_guide/database/results.html#result-arrays)



- c) Selain menggunakan query builder kita juga bisa menggunakan cara lain seperti gambar dibawah ini

```
mahasiswa.php × database.php mahasiswa_model.php × autoload.php
code-kedua > application > models > mahasiswa_model.php > mahasiswa_model > getAllmahasiswa

1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class mahasiswa_model extends CI_Model{
6
7      public function getAllmahasiswa(){
8
9          |
10         // atau bisa juga menggunakan code berikut
11         return $this->db->get('mahasiswa')->result_array();
12     }
13 }
14
15
16 /* End of file of mahasiswa_model.php */
17
18 ?>
```

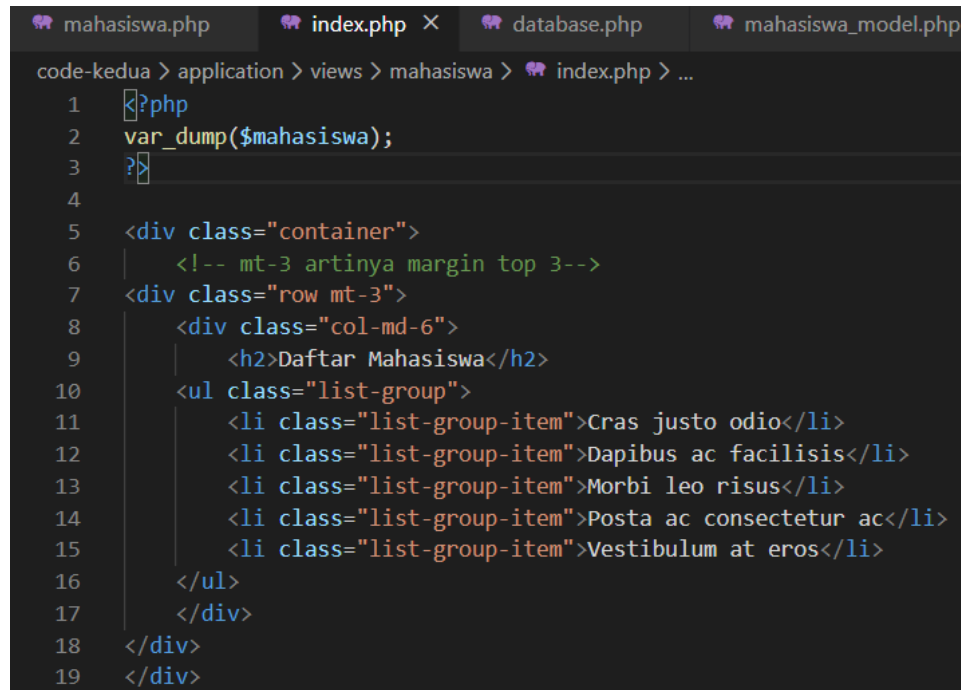
- 11) Setelah kita mempunyai sebuah model dengan nama **mahasiswa\_model** maka langkah berikutnya kita akan *load* model ini ke dalam *controller* **mahasiswa**.

```
14  public function index()
15  {
16      $this->load->model('mahasiswa_model');
17
18      // modul untuk 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  }
```

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**

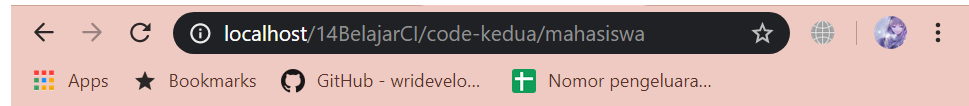
12) Jika kita lakukan `var_dump($mahasiswa)` pada **index.php** pada **views/mahasiswa/index**



```
code-kedua > 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</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">Posta ac consectetur ac</li>
15             <li class="list-group-item">Vestibulum at eros</li>
16         </ul>
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 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/mysqli\_driver.php

Line Number: 203

Backtrace:

File: C:\Program Files\Ampps\www\14BelajarCI\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/Ampps/www/14BelajarCI/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

←

→

↻

📄

localhost/14BelajarCI/code-kedua/mahasiswa

Apps

★

Bookmarks

GitHub - wridevelo...

+

Nomor pengeluaran...

CI

Home

Data Mahasiswa

About

C:\Program Files\Amps\www\14BelajarCI\code-kedua\application\views\mahasiswa\index.php:2:

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=11)

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)

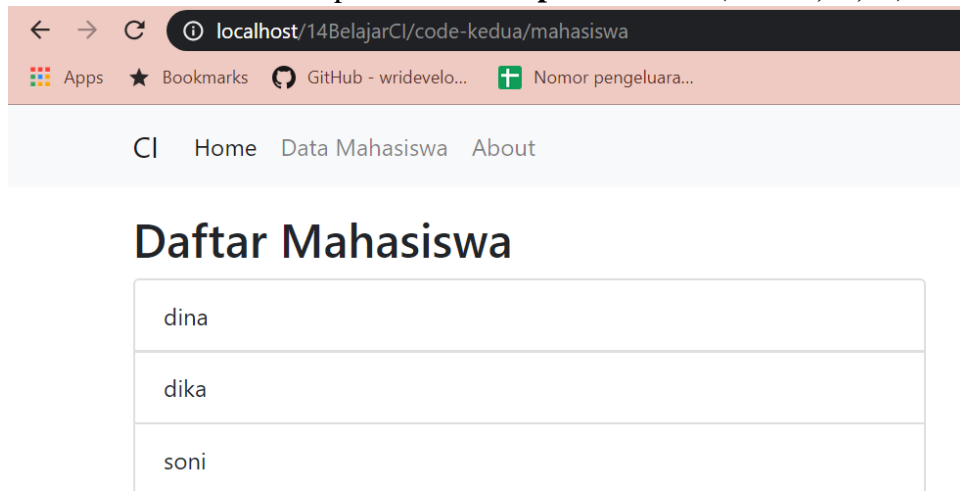
Daftar Mahasiswa

Cras justo odio
Dapibus ac facilisis
Morbi leo risus
Posta ac consectetur ac
Vestibulum at eros

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

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

- 15) Jika berhasil maka tampilan akan seperti pada gambar dibawah ini
- 16) Jika kita klik Home, halaman Home belum dapat tampil. Maka dari itu kita akan set agar semua dapat saling terhubung. Untuk itu kita membutuhkan yang namanya **base\_url**. untuk lebih jelasnya ikuti langkah-langkah berikut:
- a) Tambahkan *code* berikut pada **views/template/header (line 26,27,28)**



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')`

```
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*.

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

- 1) Buatlah sebuah *button* tambah data pada `views/mahasiswa/index.php` (line 3-7).

Tambahkan *code* pada line 3-7 untuk membuat sebuah *button*

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

- 2) Dikarenakan **mahasiswa\_model** ini nantinya akan digunakan oleh semua *function* di *controller* mahasiswa, maka “ `$this->load->model('mahasiswa_model');` “ yang awalnya hanya berada pada **function index**, kita ubah ke **function construct** agar semua *function* dapat menggunakan **mahasiswa\_model**.

```
index.php  mahasiswa.php x  mahasiswa_model.php
code-kedua > application > controllers > mahasiswa.php > mahasiswa > index
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
18         // modul untuk 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 Controllername.php */
30
31 ?>
```

- 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**)

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

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*

```
index.php  mahasiswa.php  tambah.php  mahasiswa_model.php
code-kedua > application > views > mahasiswa > tambah.php > div.container > div.row.mt-3 > div.col-md-6 > div.card > div.card-body >
1  <div class="container">
2  <div class="row mt-3">
3  <div class="col-md-6">
4      <!-- https://getbootstrap.com/docs/4.1/components/card/ -->
5      <div class="card">
6          <div class="card-header">
7              Form Tambah Data Mahasiswa
8          </div>
9          <div class="card-body">
10             <form action="" method="post">
11                 <!-- https://getbootstrap.com/docs/4.1/components/forms/ -->
12                 <div class="form-group">
13                     <label for="nama">Nama</label>
14                     <input type="text" class="form-control" id="nama">
15                 </div>
16                 <div class="form-group">
17                     <label for="nim">Nim</label>
18                     <input type="number" class="form-control" id="nim">
19                 </div>
20                 <div class="form-group">
21                     <label for="email">email</label>
22                     <input type="email" class="form-control" id="email">
23                 </div>
24                 <div class="form-group">
25                     <label for="jurusan">Jurusan</label>
26                     <select class="form-control" id="jurusan" name="jurusan">
27                         <option selected>----</option>
28                         <option value="1">Kimia</option>
29                         <option value="2">Informatika</option>
30                         <option value="3">Mesin</option>
31                     </select>
32                 </div>
33                 <button type="submit" name="submit" class="btn btn-primary float-right">Submit</button>
34             </form>
35         </div>
36     </div>
37 </div>
38 </div>
39 </div>
```

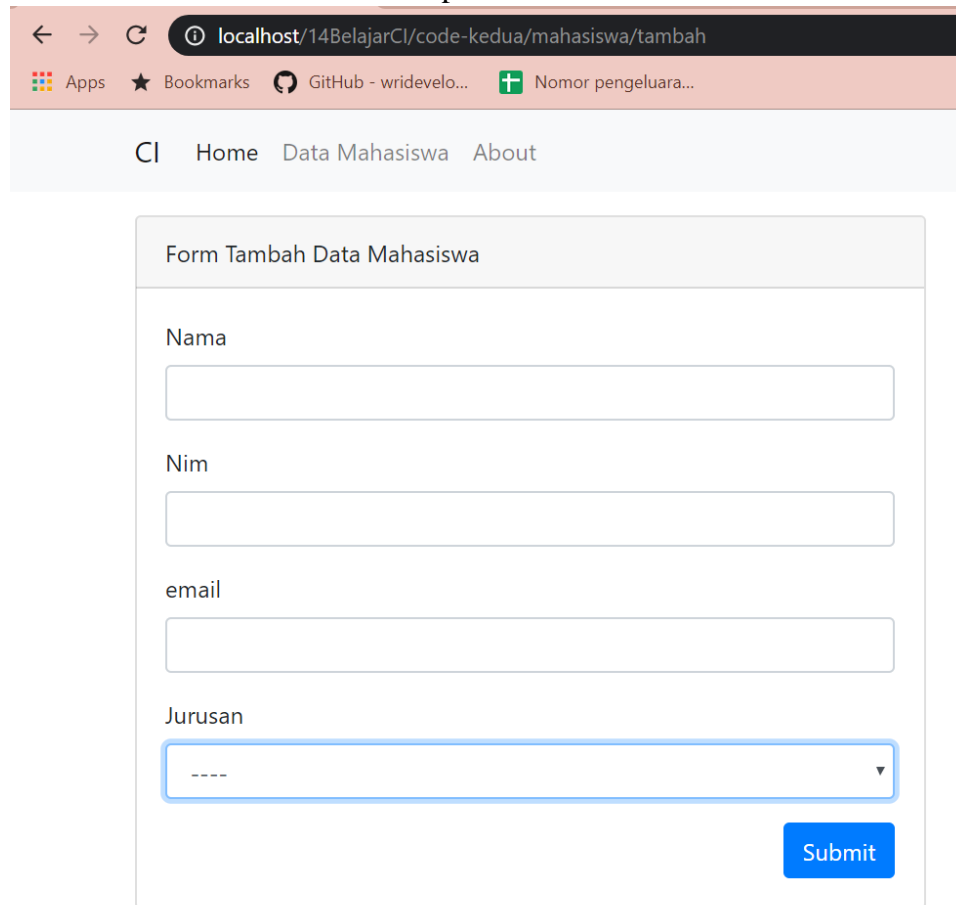
Keterangan:

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

```
<option value="1">Kimia</option>
<option value="2">Informatika</option>
<option value="3">Mesin</option>
```

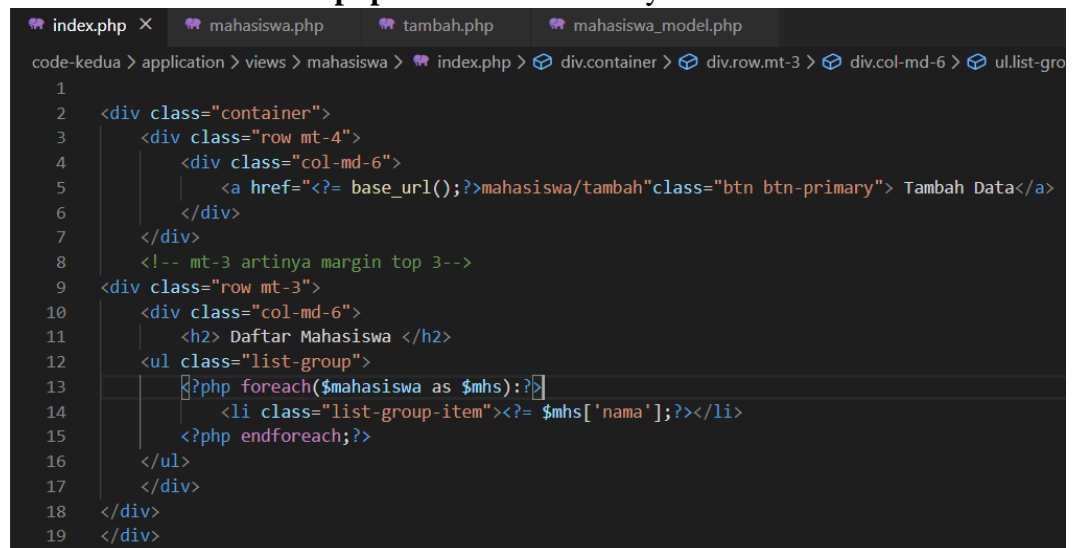


5) Maka akan dihasilkan halaman seperti di bawah ini.



The screenshot shows a web browser at the URL `localhost/14BelajarCI/code-kedua/mahasiswa/tambah`. The browser's address bar and tabs are visible. Below the browser, a navigation bar contains links: `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". The "Jurusan" field is a dropdown menu with "----" selected. A blue "Submit" button is located at the bottom right of the form.

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



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

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](https://www.codeigniter.com/user_guide/libraries/form_validation.html?highlight=form#the-form) (*searching controller*)

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

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

c) Setelah kita tambahkan *form validation*, maka langkah berikutnya menambahkan *rule*-nya

- [https://www.codeigniter.com/user\\_guide/libraries/form\\_validation.html?highlight=form#the-form](https://www.codeigniter.com/user_guide/libraries/form_validation.html?highlight=form#the-form) (*searching pipe data*)
- Untuk melihat jenis-jenis *rule* bisa dilihat pada ***rule reference***
- `$this->form_validation->set_rules('fieldname','fieldlabel','trim|required|min_length[5]|max_length[12]');`
- Jika kita lihat untuk menambahkan *rules* ada 3 parameter yaitu
  - 1) `fieldname` = nama dari htmlnya, semisal `<input type="text" class="form-control" id="nama" name="nama">` maka yang disebut dengan `fieldname` adalah `nama`
  - 2) `fieldlabel` = ini adalah alias digunakan untuk menampilkan kesalahannya (jika terjadi *error*)
  - 3) *Rule* yang digunakan

- d) Sebelum kita tambahkan *rule*-nya maka terlebih dahulu tambahkan **atribut name** pada **views/mahasiswa/tambah**

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

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

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

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

```
7 Form Tambah Data Mahasiswa
8 </div>
9 <div class="card-body">
10 <?php= validation_errors() ?>
11 <form action="" method="post">
```

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

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(); ?>`

```
</div>
<div class="card-body">
  <?php if (validation_errors()):?>
    <div class="alert alert-danger" role="alert">
      <?= validation_errors(); ?>
    </div>
  <?php endif ?>
</div>
```

- i) Akan dihasilkan seperti gambar di bawah ini

Form Tambah Data Mahasiswa

The Nama field is required.

Nama

Nim

email

Jurusan

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

```
$this->form_validation->set_rules('nama', 'Nama', 'required');
$this->form_validation->set_rules('nim', 'Nim', 'required');
$this->form_validation->set_rules('email', 'Email', 'required');
```

- 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

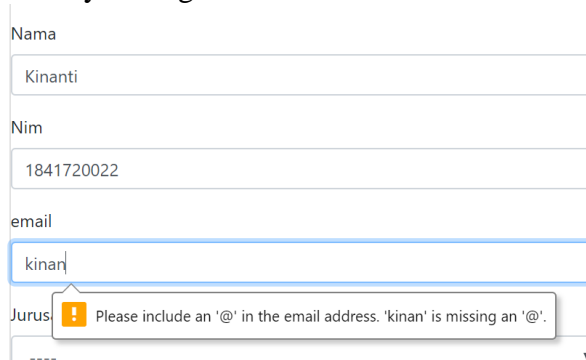
Nim

email

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

```
$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.



Nama  
Kinanti

Nim  
1841720022

email  
kinan

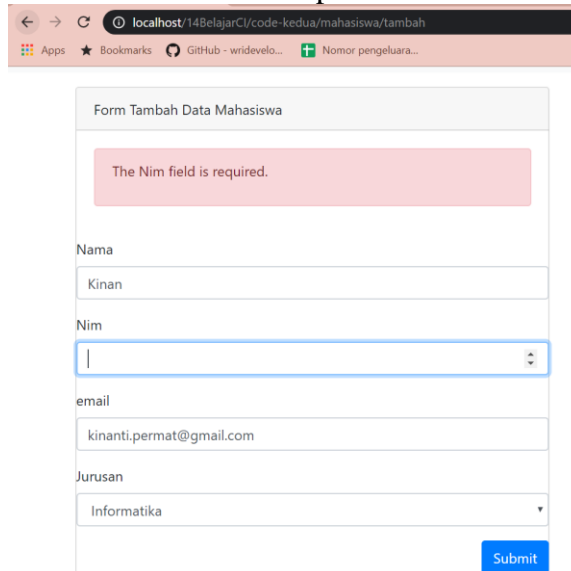
Jurusan  
Jurusan

! Please include an '@' in the email address. 'kinan' is missing an '@'.

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

```
$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');
```

- o) Akan dihasilkan alert seperti berikut ini.



Form Tambah Data Mahasiswa

The Nim field is required.

Nama  
Kinan

Nim  
|

email  
kinanti.permat@gmail.com

Jurusan  
Informatika

Submit

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

- 1) 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 ditambah ke database maka akan *redirect* ke *controller* mahasiswa (line 46 - 47)

```
26 public function tambah(){
27     $data['title']='Form Menambahkan Data Mahasiswa';
28
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('data berhasil ditambah');
42     }
43
44 }
```

- 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](https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=insert#inserting-data)
  - Tambahkan *code* seperti pada gambar dibawah ini

```
7 public function getAllmahasiswa(){
8     // https://www.codeigniter.com/user_guide/database/query_builder.html#selecting-data
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.html
13    // untuk menampilkan isi dari query
14    return $query->result_array();
15
16    // atau bisa juga menggunakan code berikut
17    // return $this->db->get('mahasiswa')->result_array();
18 }
19
20 public function tambahdatamhs(){
21     $data=[
22         "nama" => $this->input->post('nama',true),
23         "nim" => $this->input->post('nim',true),
24         "email" => $this->input->post('email',true),
25         "jurusan" => $this->input->post('jurusan',true)
26     ];
27
28     $this->db->insert('mahasiswa', $data);
29 }
```

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*

- *Line 29* adalah proses *insert* data ke database dengan **2 parameter** yaitu **nama tablenya** dan **objectnya** (dalam hal ini data yang akan dimasukkan ke database)

4) Langkah berikutnya kita coba untuk tambahkan data.

- a) Jika berhasil maka data akan tampil di halaman **index.php**  
`redirect('mahasiswa','refresh');`(*line 47 gambar controller mahasiswa*)

b) Hasil

Form Tambah Data Mahasiswa

Nama  
Kinanti

Nim  
1841720022

email  
kinantipermata@gmail.com

Jurusan  
Informatika

Submit

Tambah Data

Daftar Mahasiswa

dina
dika
soni
Kinanti