

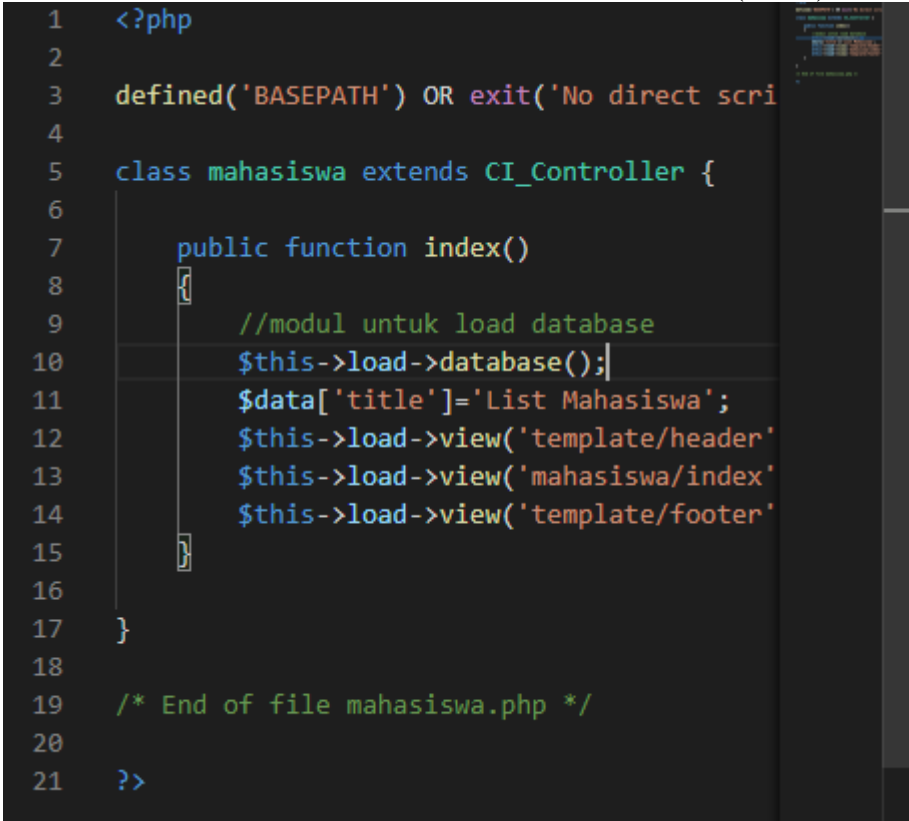
**LAPORAN PRAKTIKUM
DESAIN DAN PEMROGRAMAN WEB
JOBSHEET 2
Read dan Input Database serta Form Validation
Mata Kuliah Pemrograman Web Lanjut**



**Nama: Osa Mahanani S.
NIM: 1841720066
Kelas: TI-2B**

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

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

Langka h	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> 1 <?php 2 3 defined('BASEPATH') OR exit('No direct scri 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' 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)

```

mahasiswa.php x Home.php index.php
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class mahasiswa extends CI_Controller {
5
6      //fungsi yang akan dijalankan saat classnya diinstansiasi
7      public function __construct()
8      {
9          //digunakan untuk menjalankan fungsi construct pada class parent(ci_controller)
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
28 /* End of file Controllername.php */
29 ?>

```

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

```

59  $autoload['libraries'] = array( 'user_agent' => 'ua' );
60  */
61  $autoload['libraries'] = array( 'database' );
62
63  /*

```

- 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

	<pre> 1 <?php 2 defined('BASEPATH') OR exit('No direct script access allowed'); 3 4 class mahasiswa extends CI_Controller { 5 6 //fungsi yang akan dijalankan saat classnya diinstansiasi 7 //public function __construct() 8 //{ 9 //digunakan untuk menjalankan fungsi constrauct pada class parent(ci_controller) 10 // parent::__construct(); 11 // \$this->load->database(); 12 } 13 </pre>
6	<p>Setelah itu kita akan mengatur konfigurasi database. Untuk mengatur konfigurasi database maka buka folder application/config/database.php</p> <pre> 75 76 \$db['default'] = array(77 'dsn' => '', 78 'hostname' => 'localhost', 79 'username' => 'root', 80 'password' => 'mysql', 81 'database' => 'kuliah_CI', 82 'dbdriver' => 'mysqli', 83 'dbprefix' => '', 84 'pconnect' => FALSE, 85 'db_debug' => (ENVIRONMENT !== 'production'), 86 'cache_on' => FALSE, 87 'cachedir' => '', 88 'char_set' => 'utf8', 89 'dbcollat' => 'utf8_general_ci', 90 'swap_pre' => '', 91 'encrypt' => FALSE, 92 'compress' => FALSE, 93 'stricton' => FALSE, 94 'failover' => array(), 95 'save_queries' => TRUE 96); 97 </pre>
7	<p>Kita set konfigurasinya seperti pada gambar dibawah ini</p>

```

75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => 'mysql',
81     'database' => 'kuliah_CI',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97

```

Keterangan :

- Line 78 artinya *hostname* yang digunakan adalah localhost
- Line 79 artinya *username* (phpmyadmin) yang digunakan apa
- Line 80 artinya *password* (phpmyadmin) yang digunakan apa
- Line 81 artinya nama database yang nanti digunakan

8

Jika kita lihat konfigurasi database yang kita buat maka kita perlu sebuah database dengan nama **kuliah_CI**. Maka buat lah sebuah database dengan nama kuliah_CI dan dengan tabel mahasiswa seperti pada gambar dibawah ini

#	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)

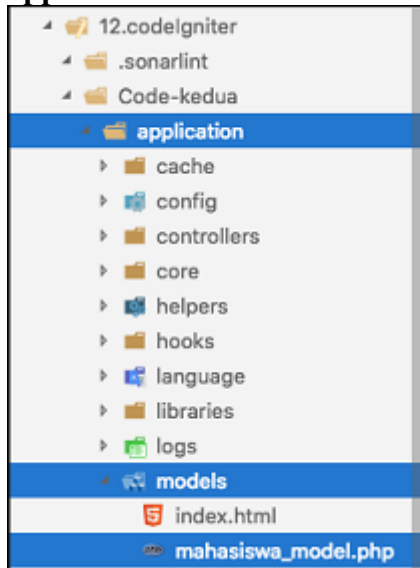
+ Options								
				id	nama	nim	email	jurusan
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	dina	1002345100	dina@gmail.com	kimia
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	dika	1002345101	dika@gmail.com	informatika
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	soni	1002345102	soni@gmail.com	mesin

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`

```
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class mahasiswa_model extends CI_Model {
6
7      public function getAllmahasiswa()
8      {
9          // https://www.codeigniter.com/user_guide/database/query_builder.html?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
18 }
19
20
21 /* End of file mahasiswa_model.php */
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 generate 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

```

mahasiswa.php  mahasiswa_model.php  index.php
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class mahasiswa_model extends CI_Model {
6
7      public function getAllmahasiswa()
8      {
9
10
11      // atau bisa juga menggunakan code berikut
12      return $this->db->get('mahasiswa')->result_array();
13      }
14
15  }
16
17  /* End of file mahasiswa_model.php */
18
19  ?>

```

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

```

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/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
16         //return $query->result_array();
17
18         //atau bisa menggunakan code berikut
19         return $this->db->get('mahasiswa')->result_array();
20     }
21
22 }
23
24 /* End of file mahasiswa_model.php */
25
26 ?>

```

Keterangan :

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

```
mahasiswa.php  mahasiswa_model.php  index.php
1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class mahasiswa_model extends CI_Model {
6
7     public function getAllmahasiswa()
8     {
9
10
11         // atau bisa juga menggunakan code berikut
12         return $this->db->get('mahasiswa')->result_array();
13     }
14
15 }
16
17 /* End of file mahasiswa_model.php */
18
19 ?>
```

```
mahasiswa.php  mahasiswa_model.php  index.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class mahasiswa extends CI_Controller {
5
6     public function index()
7     {
8         $this->load->model('mahasiswa_model');
9
10         // modul untuk load database
11         // $this->load->database();
12         $data['title']='List Mahasiswa';
13         $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
14         $this->load->view('template/header',$data);
15         $this->load->view('mahasiswa/index',$data);
16         $this->load->view('template/footer');
17     }
18
19 }
20
21
22 /* End of file Controllername.php */
23
24 ?>
```

12

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

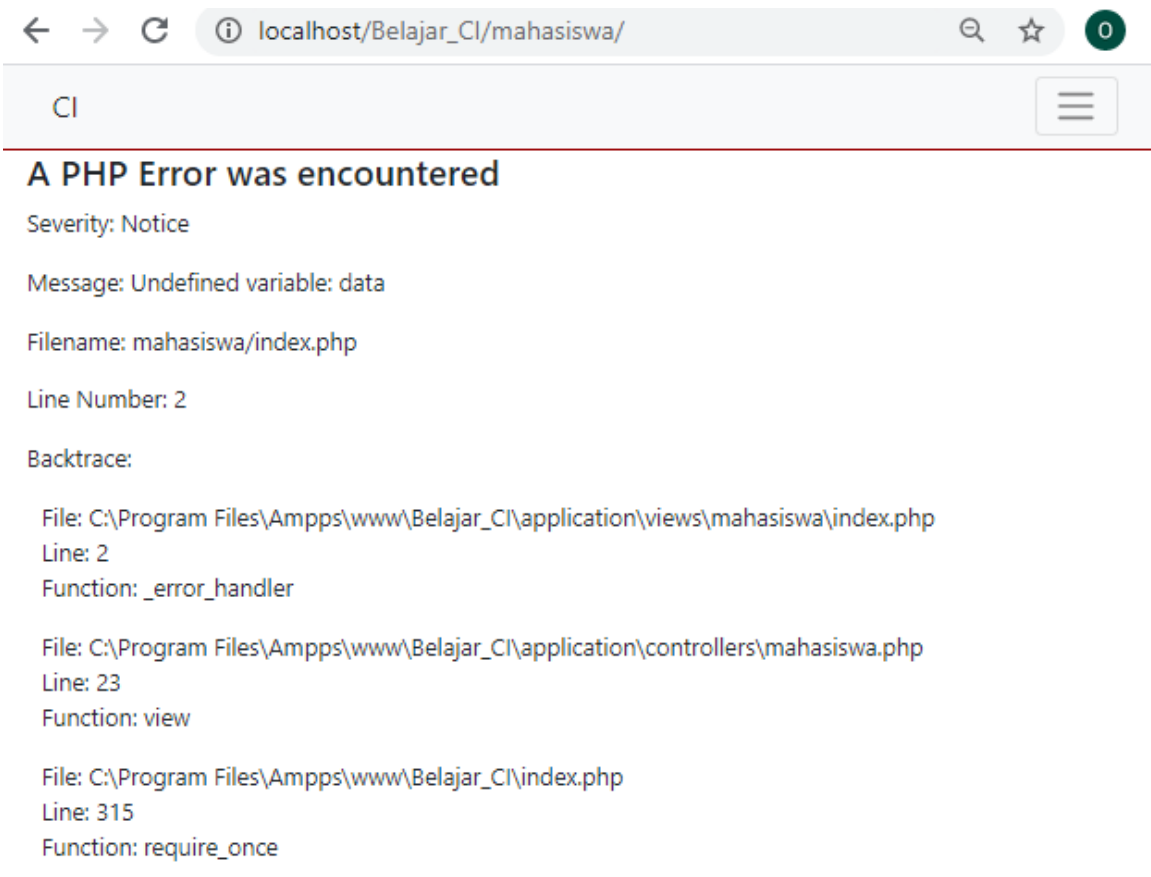

```

application > views > mahasiswa > index.php
1  <?php
2  var_dump($data);
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</li>
12             <li class="list-group-item">Morbi leo risus</li>
13             <li class="list-group-item">Porta ac consectetur ac</li>
14             <li class="list-group-item">Vestibulum at eros</li>
15         </ul>
16     </div>
17 </div>
18 </div>

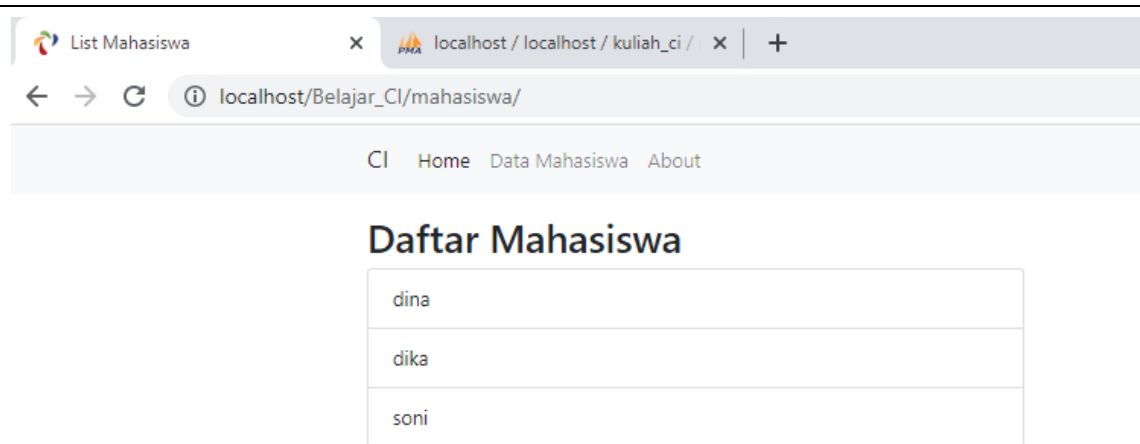
```

Keterangan

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

	 <p>A PHP Error was encountered</p> <p>Severity: Notice</p> <p>Message: Undefined variable: data</p> <p>Filename: mahasiswa/index.php</p> <p>Line Number: 2</p> <p>Backtrace:</p> <p>File: C:\Program Files\Ampps\www\Belajar_CI\application\views\mahasiswa\index.php Line: 2 Function: _error_handler</p> <p>File: C:\Program Files\Ampps\www\Belajar_CI\application\controllers\mahasiswa.php Line: 23 Function: view</p> <p>File: C:\Program Files\Ampps\www\Belajar_CI\index.php Line: 315 Function: require_once</p> <hr/> <p>Hal ini disebabkan karena tidak ketahui data pada mahasiswa/index.php pada <i>line 2</i> berupa <code>var_dump(\$mahasiswa)</code>, maka pastikan terlebih dahulu pada <i>controller</i> mahasiswa.php sudah terdapat <code>\$data</code> di <i>line 13</i> dan <i>15</i>.</p>
13	Jika berhasil maka data akan tampil seperti pada gambar berikut

	<div><div>CIHomeData MahasiswaAbout</div><div>C:\Program Files\Amps\www\Belajar_CI\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)</div><div><div>Daftar Mahasiswa</div><table><tr><td>Cras justo odio</td></tr><tr><td>Dapibus ac facilisis</td></tr><tr><td>Morbi leo risus</td></tr><tr><td>Porta ac consectetur ac</td></tr><tr><td>Vestibulum at eros</td></tr></table></div></div>	Cras justo odio	Dapibus ac facilisis	Morbi leo risus	Porta ac consectetur ac	Vestibulum at eros
Cras justo odio						
Dapibus ac facilisis						
Morbi leo risus						
Porta ac consectetur ac						
Vestibulum at eros						
14	<div><div>Agar semua isi dari tabel dapat tampil di halaman index mahasiswa maka kita harus lakukan proses <i>looping</i>. Gunakan <i>code</i> berikut pada views/mahasiswa/index</div><div><div>mahasiswa.phpmahasiswa_model.phpindex.php x</div><div><div>1</div><div><div><div class="container"></div><div><!-- mt-3 artinya margin top 3 --></div><div><div class="row mt-3"></div><div><div class="col-md-6"></div><div><h2>Daftar Mahasiswa</h2></div><div><ul class="list-group"></div><div><?php foreach(\$mahasiswa as \$mhs):?></div><div><li class="list-group-item"><?= \$mhs['nama'];?></div><div><?php endforeach; ?></div><div></div><div></div></div><div></div></div></div></div></div></div>					
15	<div>Jika berhasil maka tampilan akan seperti pada gambar dibawah ini</div>					



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)

```

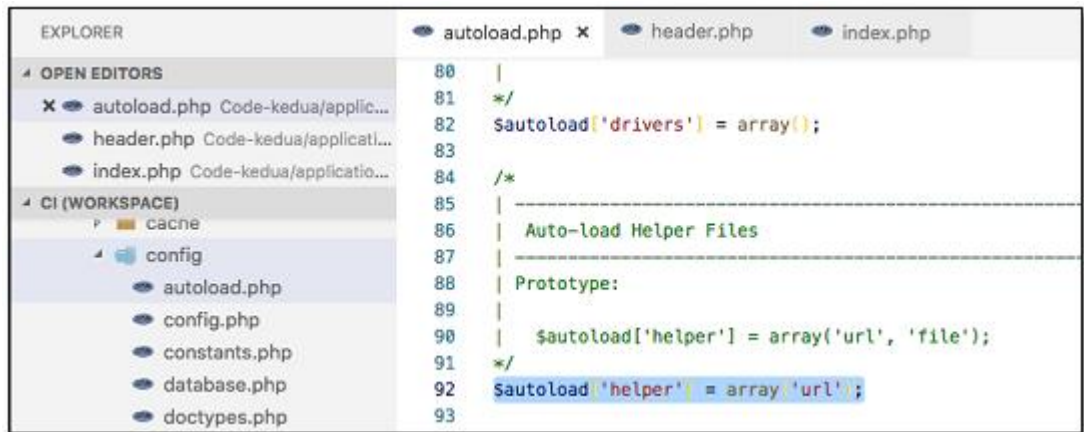
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8 <!-- Bootstrap CSS -->
9 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
10
11 <!-- <title>Hello, world!</title> -->
12 <title><?= $title ?></title>
13
14 </head>
15 <body>
16
17 <!-- navbar -->
18 <nav class="navbar navbar-expand-lg navbar-light bg-light">
19 <div class="container">
20 <a class="navbar-brand" href="#">CI</a>
21 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-expanded="false" aria-controls="navbarNav">
22 <span class="navbar-toggler-icon"></span>
23 </button>
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 <!-- <a class="nav-item nav-link disabled" href="#">Disabled</a> -->
30 </div>
31 </div>
32 </div>
33 </nav>

```

Keterangan :

- Line 26, 27, 28 tambahkan **base_url** (pastikan **config/config.php** sudah diset)
\$config['base_url'] = 'http://localhost/12.codeIgniter/code-kedua/'; (code_kedua adalah nama project CI jadi sesuaikan dengan nama project CI kalian)
- Agar halaman web mahasiswa dapat ditampilkan saat *user* mengklik mahasiswa, maka setelah **base_url** tambahkan **controller mahasiswa**

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



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

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

Langka h	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 2 <div class="container"> 3 <div class="row mt-4"> 4 <div class="col-md-6"> 5 Tambah Data 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'];?> 15 <?php endforeach; ?> 16 17 </div> 18 </div> 19 </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 defined('BASEPATH') OR exit('No direct script access allowed'); 3 4 class mahasiswa extends CI_Controller { 5 6 //fungsi yang akan dijalankan saat classnya diinstansiasi 7 public function __construct() 8 { 9 //digunakan untuk menjalankan fungsi constrauct pada class parent(ci_controller) 10 parent::__construct(); 11 \$this->load->model('mahasiswa_model'); 12 } 13 14 public function index() 15 { 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 } </pre>
3	<p>Setelah itu tambahkan function tambah pada controller mahasiswa dan tambahkan code berikut.</p> <p>(Tips : jika menggunakan vsCode untuk menduplikat 1 baris dapat menggunakan alt+shift+panah bawah)</p>  <pre> mahasiswa.php X tambah.php database.php mahasiswa_m application > controllers > mahasiswa.php 25 public function tambah() { 26 27 \$data['title']= 'Form Menambahkan Data Mahasiswa'; 28 \$this->load->view('template/header', \$data); 29 \$this->load->view('mahasiswa/tambah', \$data); 30 \$this->load->view('template/footer'); 31 } 32 33 } 34 35 36 /* End of file mahasiswa.php */ 37 38 ?> </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Pada line 32, \$this->load->view('mahasiswa/tambah') artinya bahwa function tambah akan melakukan <i>load view</i> pada folder mahasiswa, dengan nama filenya adalah tambah
4	<p>Setelah kita membuat function tambah pada controller mahasiswa, selanjutnya buatlah form tambah pada views/mahasiswa/tambah.php</p>

```

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                 <div class="form-group">
15                     <label for="nim">Nim</label>
16                     <input type="number" class="form-control" id="nim">
17                 </div>
18                 <div class="form-group">
19                     <label for="email">Email</label>
20                     <input type="email" class="form-control" id="email">
21                 </div>
22                 <div class="form-group">
23                     <label for="jurusan">Jurusan</label>
24                     <select class="form-control" name="jurusan" id="jurusan">
25                         <option selected>----</option>
26                         <option value="kimia">Kimia</option>
27                         <option value="informatika">Informatika</option>
28                         <option value="mesin">Mesin</option>
29                     </select>
30                 </div>
31                 <button type="submit" name="submit" class="btn btn-primary float-right">Submit</button>
32             </form>
33         </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)

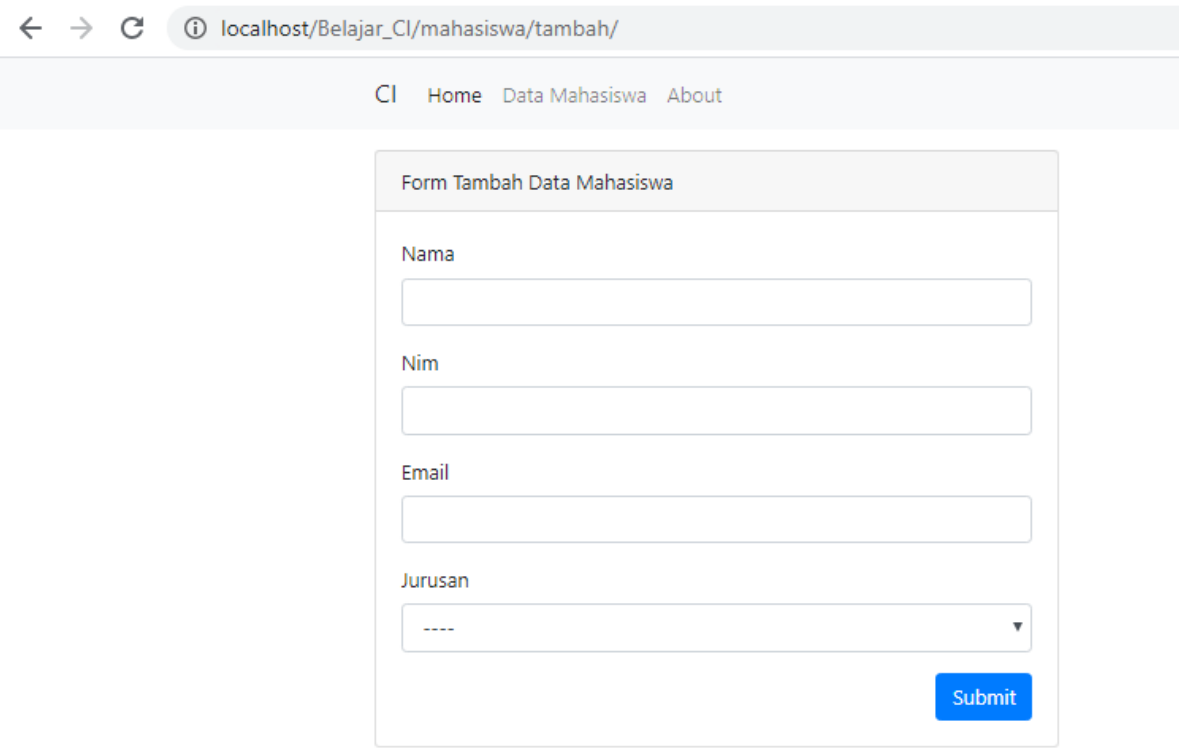
```

<option value="informatika">Informatika</option>
<option value="kimia">kimia</option>
<option value="mesin">Mesin</option>

```

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> <pre> 1 2 <div class="container"> 3 <div class="row mt-4"> 4 <div class="col-md-6"> 5 <a href="<?=base_url();?>mahasiswa/tambah" class="btn btn-primary">Tambah Mahasiswa 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"> 15 <?= \$mhs['nama'];?> 16 17 <?php endforeach; ?> 18 19 </div> 20 </div> 21 </div> </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Tambahkan <i>code</i> pada line 5 agar pada saat <i>user</i> melakukan <i>event click</i> maka <i>user</i> akan ditampilkan halaman tambah.
7	<p>Tambahkan validasi menggunakan <i>form_validation</i>. Contoh validasi <i>form</i> adalah seperti <i>field</i> nama harus diisi, nim harus angka dsb. Untuk lebih jelasnya ikuti langkah berikut ini :</p> <p>a. Tambahkan <i>form_validation</i></p>

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

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

```
25 public function tambah() {
26
27     $data['title'] = 'Form Menambahkan Data Mahasiswa';
28
29     $this->$this->form_validation->set_rules('nama', 'Nama', 'req
30
31     if ($this->form_validation->run() == FALSE) {
32
33         $this->load->view('template/header', $data);
34         $this->load->view('mahasiswa/tambah', $data);
35         $this->load->view('template/footer');
36     } else {
37         echo "data berhasil ditambah";
38     }
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 (*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**

```

        <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" name="jurusan" 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)

```

27     $data['title']= 'Form Menambahkan Data Mahasiswa';
28
29     $this->$this->form_validation->set_rules('nama', 'Nama', 'required');
30
31     if ($this->form_validation->run() == FALSE) {
32
33         $this->load->view('template/header', $data);
34         $this->load->view('mahasiswa/tambah', $data);
35         $this->load->view('template/footer');
36     } else {
37         echo "data berhasil ditambah";
38     }
39
40 }
41
42 }
43
44 /* End of file mahasiswa.php */
45
46 ?>

```

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

```

8         <div class="card-body">
9             <?php= validation_errors() ?>
10             <form action="" method="POST">
11                 <div class="form-group">

```

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

← → ↻ ⓘ localhost/Belajar_CI/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/>

```

<div class="card-body">

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

    <form action="" method="POST">

```

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

i. Akan dihasilkan seperti gambar di bawah ini

← → ↻ ⓘ localhost/Belajar_CI/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**.

```

25 public function tambah() {
26
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
35         $this->load->view('template/header', $data);
36         $this->load->view('mahasiswa/tambah', $data);
37         $this->load->view('template/footer');
38     } else {
39         echo "data berhasil ditambah";
40     }
41
42 }
43
44 }
45
46 /* End of file mahasiswa.php */
47
48

```

- k. Hasilnya seperti berikut ini.

← → ↻ ⓘ localhost/Belajar_CI/mahasiswa/tambah

CI Home Data Mahasiswa About

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)**

```
$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');  
  
if ($this->form_validation->run() == FALSE) {
```

m. Hasilnya sebagai berikut.

← → ↻ ⓘ localhost/Belajar_CI/mahasiswa/tambah

CI Home Data Mahasiswa About

Form Tambah Data Mahasiswa

Nama

Nim

Email

Jurusan

Submit

← → ↻ ⓘ localhost/Belajar_CI/mahasiswa/tambah

data berhasil ditambah

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

```
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
```

- o. Akan dihasilkan alert seperti berikut ini.

The screenshot shows a web browser window with the address bar displaying 'localhost/Belajar_CI/mahasiswa/tambah'. The page has a navigation bar with links 'CI', 'Home', 'Data Mahasiswa', and 'About'. The main content area features a form titled 'Form Tambah Data Mahasiswa'. The form contains four input fields: 'Nama' with the value 'osaaaaaaaaa', 'Nim' with the value '12345eeeeeeee', 'Email' with the value 'osa@gmail.com', and 'Jurusan' with a dropdown menu showing 'Informatika'. A blue 'Submit' button is located at the bottom right of the form.

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

Setelah kita berhasil membuat **form_validation** langkah berikutnya adalah memasukkan data ke database (**insert database**)

Langka h	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>)

```

25 public function tambah() {
26
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
35         $this->load->view('template/header', $data);
36         $this->load->view('mahasiswa/tambah', $data);
37         $this->load->view('template/footer');
38     } else {
39         $this->mahasiswa_model->tambahdatamhs();
40         redirect('mahasiswa','refresh');
41     }
42
43 }
44
45 }
46
47 /* End of file mahasiswa.php */
48
49 ?>

```

- 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

```

1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class mahasiswa_model extends CI_Model {
6
7     public function getAllmahasiswa()
8     {
9         //https://www.codeigniter.com/user_guide/database/query_builder.html#selecting-data
10        //$query digunakan untuk menampung isi dari tabel mahasiswa
11        $query=$this->db->get('mahasiswa');
12        return $query->result_array();
13
14        //https://www.codeigniter.com/user_guide/database/results.html
15        //untuk menampilkan isi dari query
16
17        //return $query->result_array();
18
19        //atau bisa menggunakan code berikut
20        //return $this->db->get('mahasiswa')->result_array();
21    }
22    public function tambahdatamhs() {
23        $data=[
24            "nama" => $this->input->post('nama',true),
25            "nim" => $this->input->post('nim',true),
26            "email" => $this->input->post('email',true),
27            "jurusan" => $this->input->post('jurusan',true)
28        ];
29
30        $this->db->insert('mahasiswa',$data);
31    }
32 }

```

Keterangan:

- Line 22-27 adalah sebuah array dimana nantinya isi array adalah data-data yang akan dikirim (POST) ke database

	<ul style="list-style-type: none"> - 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 <div data-bbox="343 264 1476 488"> </div> <ul style="list-style-type: none"> • Line 29 adalah proses <i>insert</i> data ke database dengan 2 parameter yaitu nama tablenameya dan objectnya (dalam hal ini data yang akan dimasukkan ke database)
4	<p>Langkah berikutnya kita coba untuk tambahkan data. Jika berhasil maka data akan tampil di halaman index.php</p> <p>a. <code>redirect('mahasiswa','refresh');</code>(line 47 gambar <i>controller</i> mahasiswa)</p> <p>b. Hasil</p> <div data-bbox="323 779 1241 1500"> </div>

Tambah Data

Daftar Mahasiswa

dina

dika

soni

mahanani

osa