



NAMA : HUSNUL HOTIMAH

NIM : 1841720014

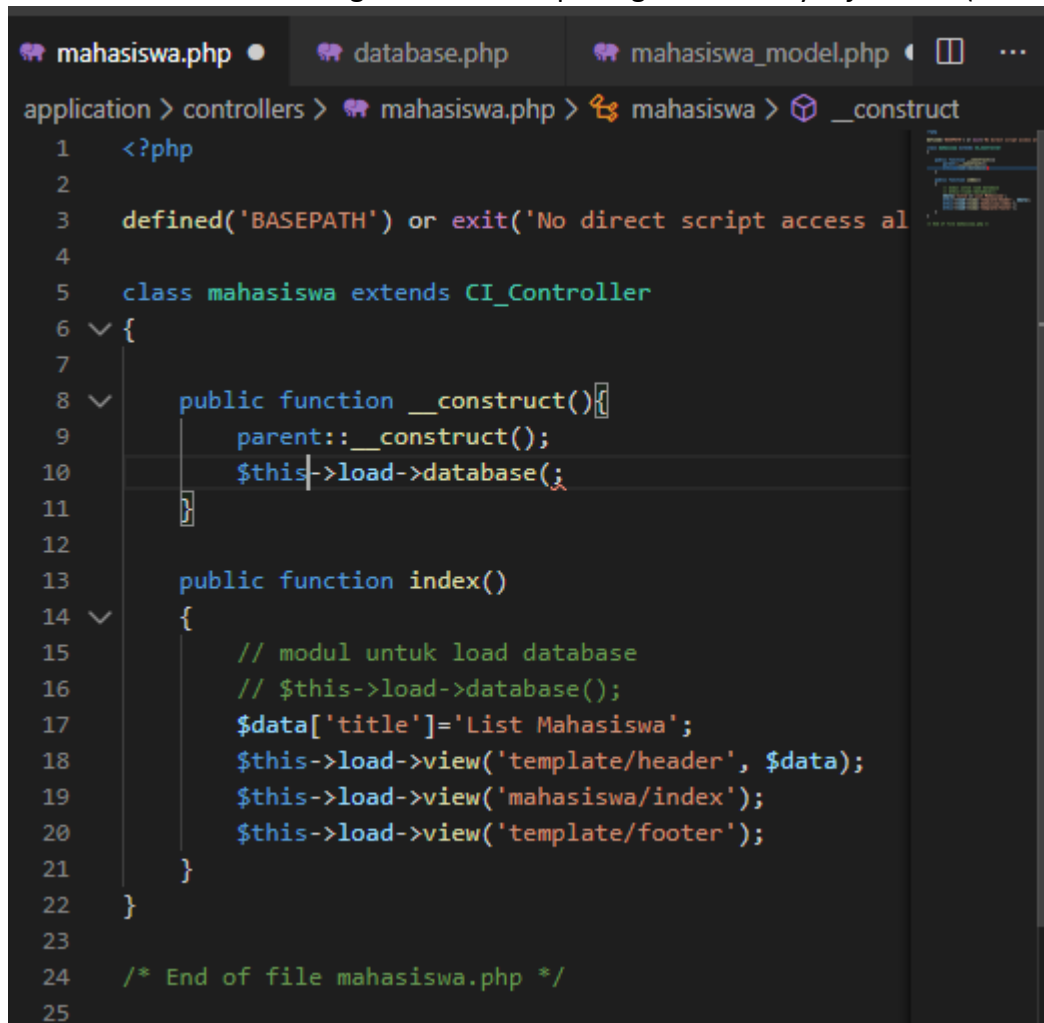
KELAS : TI-2A

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

| Langkah | Keterangan |
|---------|---|
| 1 | Buka kembali folder CodeIgniter yang telah dibuat pada Jobsheet sebelumnya |
| 2 | <p>Buka <i>controller</i> mahasiswa dan tambahkan <i>code</i> berikut (line 9)</p> <pre>application > controllers > mahasiswa.php > mahasiswa > index 1 <?php 2 3 defined('BASEPATH') or exit('No direct script access allowed'); 4 5 class mahasiswa extends CI_Controller 6 { 7 8 public function index() 9 { 10 // modul untuk load database 11 \$this->load->database(); 12 \$data['title']='List Mahasiswa'; 13 \$this->load->view('template/header', \$data); 14 \$this->load->view('mahasiswa/index'); 15 \$this->load->view('template/footer'); 16 } 17 } 18 19 /* End of file mahasiswa.php */ 20</pre> <p>Keterangan :</p> <ul style="list-style-type: none">- Pada line 9 menjelaskan bawa <i>function</i> index ini akan melakukan <i>load</i> database. <i>Load</i> database ini hanya dilakukan di <i>function</i> index saja jika ada <i>function</i> lain maka kita harus melakukan <i>load</i> database lagi. Sehingga hal ini dirasa tidak praktis maka untuk itu kita harus tambahkan <i>code</i> agar <i>load</i> database ini dapat digunakan banyak <i>function</i>. |

3

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



```

1  <?php
2
3  defined('BASEPATH') or exit('No direct script access allowed');
4
5  class mahasiswa extends CI_Controller
6  {
7
8      public function __construct()
9      {
10         parent::__construct();
11         $this->load->database();
12     }
13
14     public function index()
15     {
16         // modul untuk load 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     /* End of file mahasiswa.php */
25

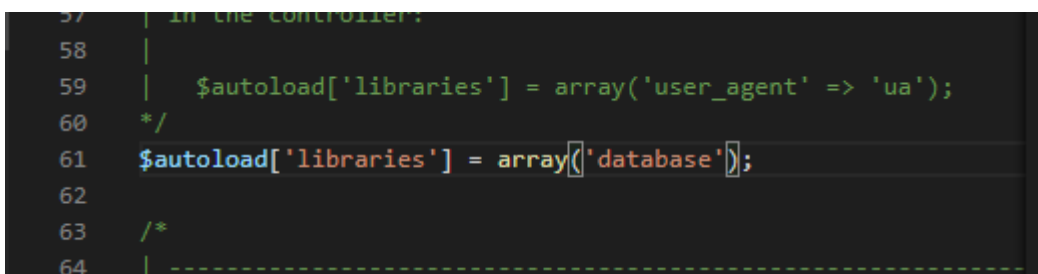
```

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



```

57  | In the controller:
58  |
59  | $autoload['libraries'] = array('user_agent' => 'ua');
60  */
61  $autoload['libraries'] = array('database');
62
63  /*
64  | -----

```

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

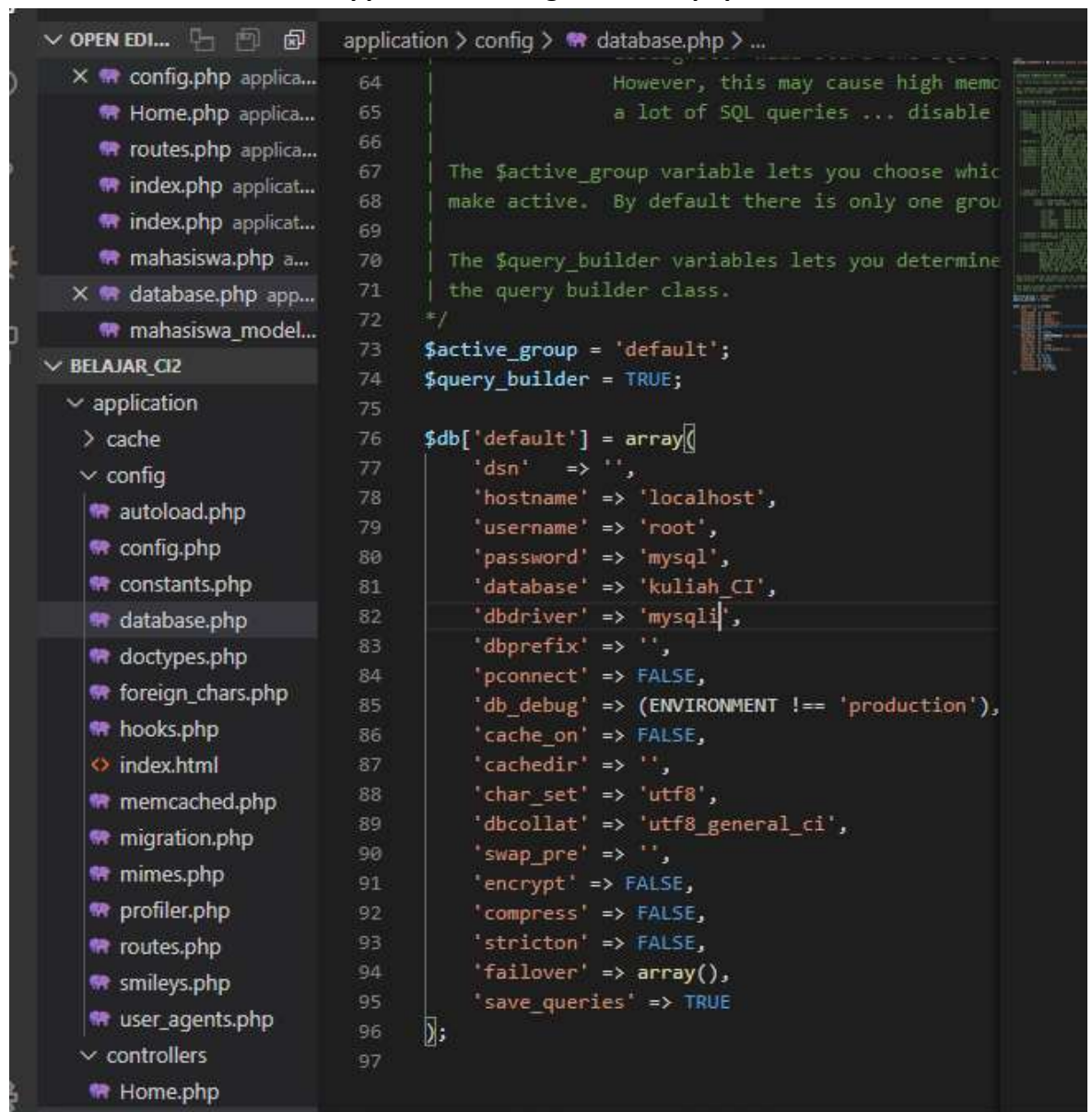
```

mahasiswa.php x 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     //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    // }

```

6

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



```

application > config > database.php > ...
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97

```

However, this may cause high memory usage if you have a lot of SQL queries ... disable

The \$active_group variable lets you choose which database group to make active. By default there is only one group defined.

The \$query_builder variable lets you determine whether to use the query builder class.

```

*/
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => 'mysql',
    'database' => 'kuliah_CI',
    'dbdriver' => 'mysql',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

7

Kita set konfigurasinya seperti pada gambar dibawah ini

```

64 |         However, this may cause high memo
65 |         a lot of SQL queries ... disable
66 |
67 | The $active_group variable lets you choose whic
68 | make active. By default there is only one grou
69 |
70 | The $query_builder variables lets you determine
71 | the query builder class.
72 | */
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array([
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => 'mysql',
81     'database' => 'kuliah_CI',
82     'dbdriver' => 'mysql',
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



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

☐ Pilih Semua Dengan pilihan: ☐ Jelajahi ☐ Ubah ☐ Hapus ☐ Utama ☐ Unik ☐ Indeks ☐ Teks penuh

☐ Cetak ☐ Move columns ☐ Normalisasi

9

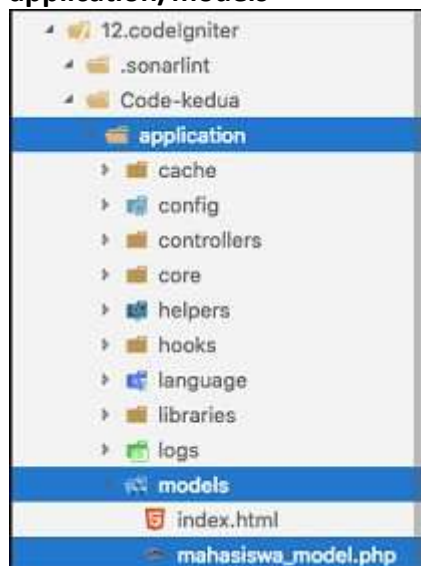
Tambahkan beberapa data ke dalam tabel mahasiswa seperti pada gambar (pada contoh terdapat 3 data, jumlah data lebih banyak tidak masalah)

| | | | | | |
|--|----|------|------------|----------------|-------------|
| <input type="checkbox"/> Tampilkan semua Jumlah baris: 25 Saring baris: Cari di tabel ini Urut berdasarkan kur | | | | | |
| + Opsi | | | | | |
| | id | nama | nim | email | jurusan |
| <input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus | 1 | dina | 1002345100 | dina@gmail.com | kimia |
| <input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus | 2 | dika | 1002345101 | dika@gmail.com | informatika |
| <input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus | 3 | soni | 1002345102 | soni@gmail.com | mesin |
| <input type="checkbox"/> Pilih Semua Dengan pilihan: <input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus <input type="checkbox"/> Ekspor | | | | | |
| <input type="checkbox"/> Tampilkan semua Jumlah baris: 25 Saring baris: Cari di tabel ini Urut berdasarkan kur | | | | | |

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**



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

```
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          $query=$this->db->get('mahasiswa');
10         return $query->result->result_array();
11     }
12
13 }
14
15 /* End of file mahasiswa_model.php */
16
17 ?>
```

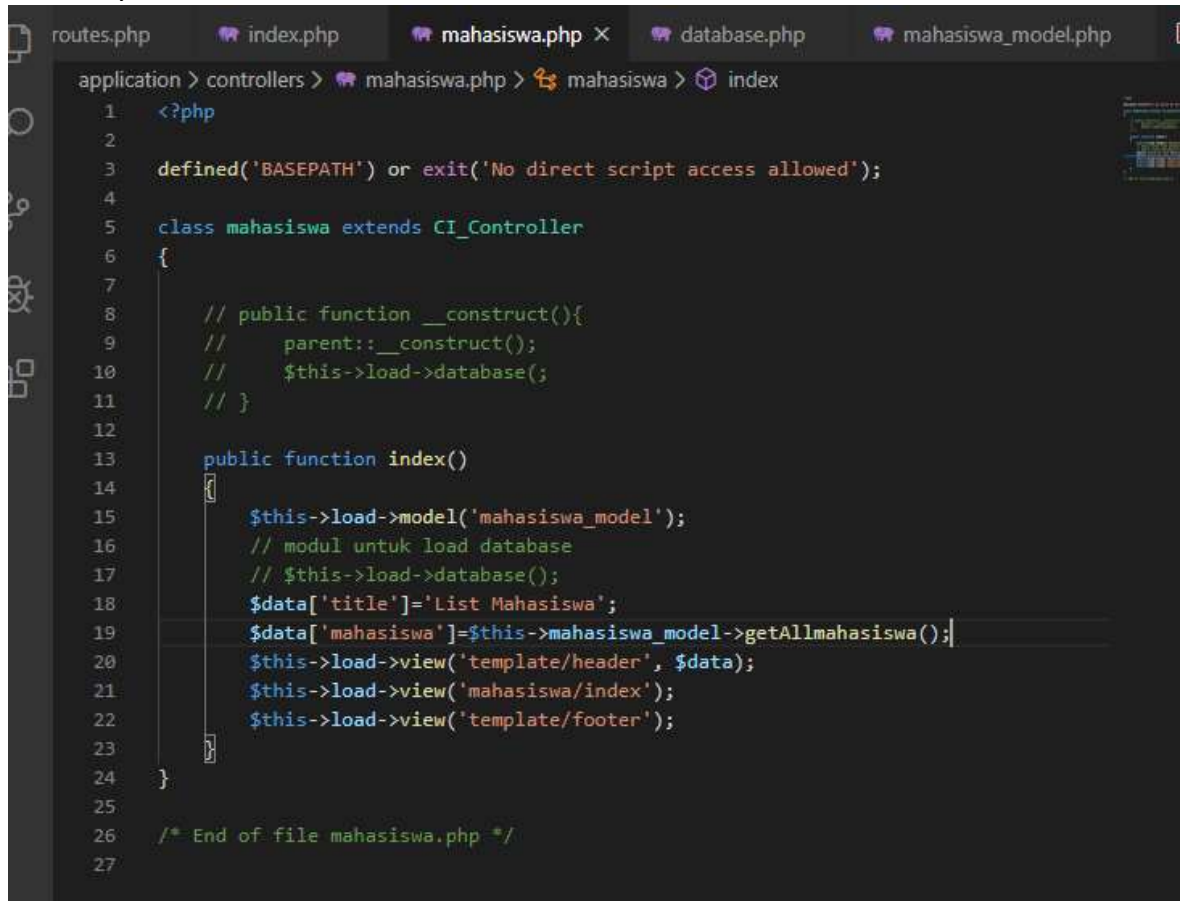
Keterangan :

- 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

```
...mahasiswa  mahasiswa.php  database.php  mahasiswa_model.php x ...
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          // $query=$this->db->get('mahasiswa');
10         return $this->db->get('mahasiswa')->result_array();
11     }
12
13 }
14
15 /* End of file mahasiswa_model.php */
16
17 ?>
```

11

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



```

1  <?php
2
3  defined('BASEPATH') or exit('No direct script access allowed');
4
5  class mahasiswa extends CI_Controller
6  {
7
8      // public function __construct(){
9      //     parent::__construct();
10     //     $this->load->database();
11     // }
12
13     public function index()
14     {
15         $this->load->model('mahasiswa_model');
16         // modul untuk load database
17         // $this->load->database();
18         $data['title']='List Mahasiswa';
19         $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
20         $this->load->view('template/header', $data);
21         $this->load->view('mahasiswa/index');
22         $this->load->view('template/footer');
23     }
24 }
25
26 /* End of file mahasiswa.php */
27

```

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**

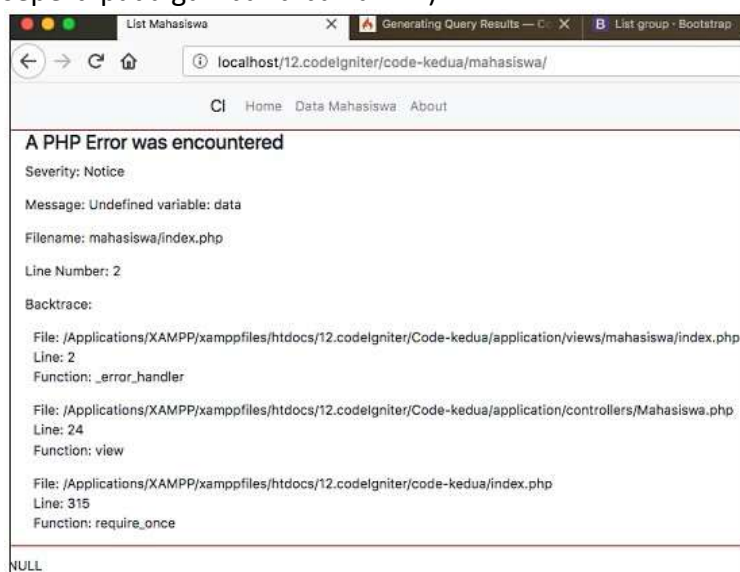
```

1  <?php
2  var_dump($mahasiswa);
3  ?>
4
5  <div class="container">
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">Crass justo odio</li>
11                 <li class="list-group-item">Dapibus ac facilist in</li>
12                 <li class="list-group-item">Morbi leo risus</li>
13                 <li class="list-group-item">Porta ac consectetur</li>
14                 <li class="list-group-item">Vestibulum at eros</li>
15             </ul>
16          </div>
17      </div>
18  </div>

```

Keterangan

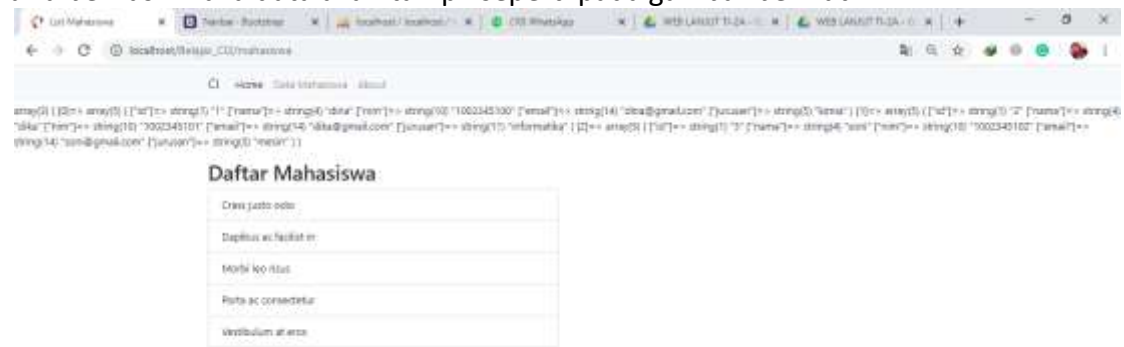
- `$mahasiswa` didapat dari mana ??? -> jika kita lihat *line 22* pada *controller* mahasiswa terdapat `[mahasiswa]`
- Jika kita lihat pada *controller* mahasiswa terdapat perintah/code seperti berikut:
 - `$data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();` (Code pertama)
 - `$this->load->view('mahasiswa/index',$data);` (Code kedua)
- Jika kita lihat *code* kedua, ada `$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)



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



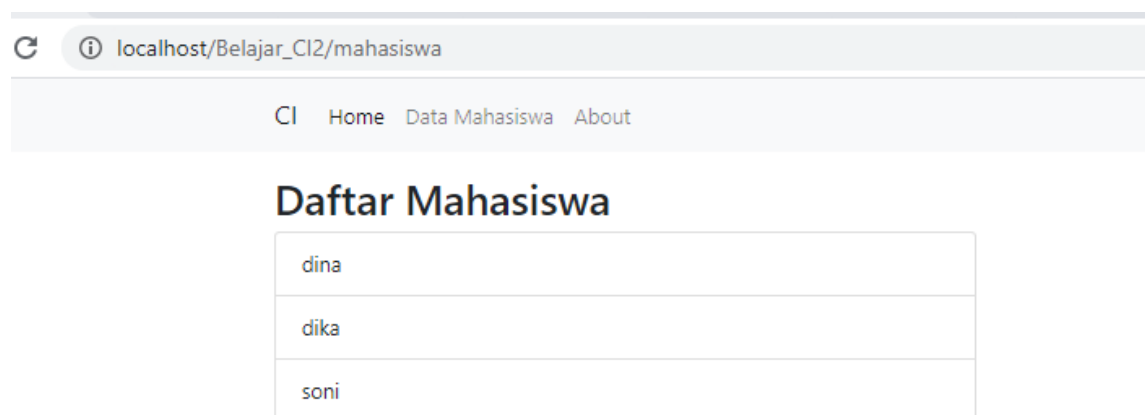
14

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



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)

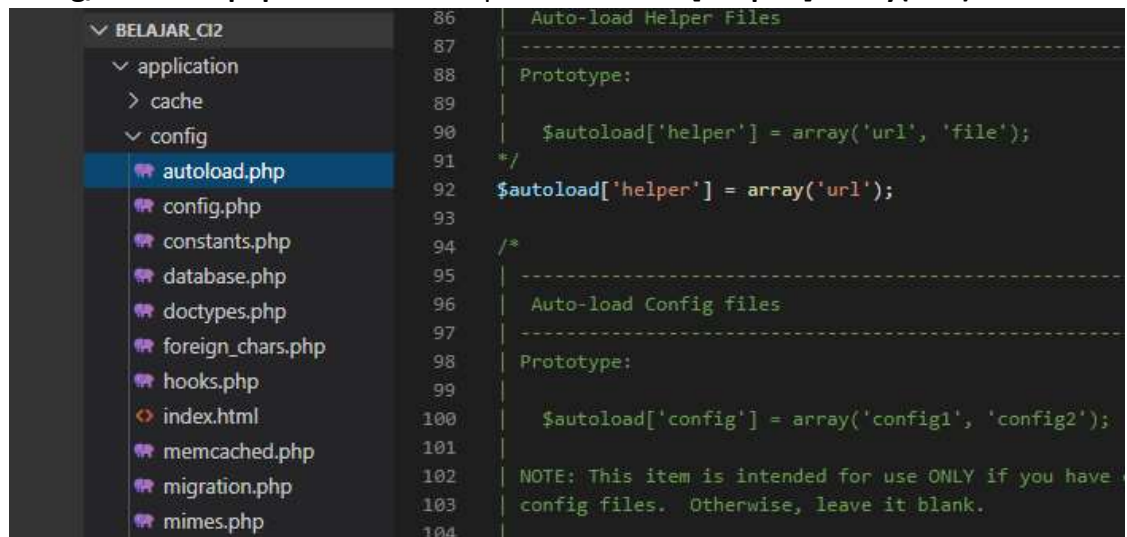
```

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><?php echo $title ?></title>
13 </head>
14 <body>
15
16 <!-- navbar -->
17
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="#navbarSupportedContent"
22 <span class="navbar-toggler-icon"></span>
23 </button>
24 <div class="collapse navbar-collapse" id="navbarSupportedContent">
25 <div class="navbar-nav">
26 <a class="nav-item nav-link" href="<?php echo base_url(); ?>">Home <span class="sr-only">Home</span>
27 <a class="nav-item nav-link" href="<?php echo base_url(); ?>mahasiswa">Data Mahasiswa
28 <a class="nav-item nav-link" href="<?php echo base_url(); ?>">About</a>
29 <!-- <a class="nav-item nav-link disabled" href="#">Disabled</a> -->
30 </div>
31 </div>
32 </div>
33 </nav>
  
```

Keterangan :

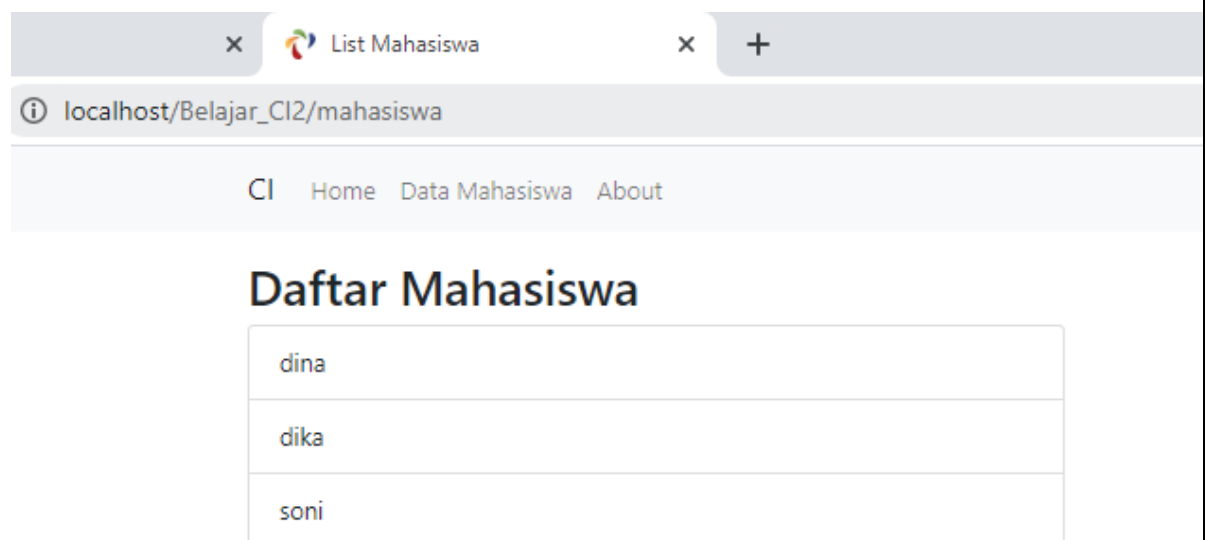
- Line 26, 27, 28 tambahkan **base_url** (pastikan **config/config.php** sudah diset)
`$config['base_url'] = 'http://localhost/12.codeigniter/code-kedua/';` (code_kedua adalah nama project CI jadi sesuaikan dengan nama project CI kalian)
- Agar halaman web mahasiswa dapat ditampilkan saat *user* mengklik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



```
86 | Auto-load Helper Files
87 | -----
88 | Prototype:
89 |
90 |     $autoload['helper'] = array('url', 'file');
91 | */
92 | $autoload['helper'] = array('url');
93 |
94 | /*
95 | -----
96 | Auto-load Config files
97 | -----
98 | Prototype:
99 |
100 |     $autoload['config'] = array('config1', 'config2');
101 |
102 | NOTE: This item is intended for use ONLY if you have
103 | config files. Otherwise, leave it blank.
104 |
```

- 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 data-bbox="303 309 1460 952"> <div class="container"> <div class="row mt-4"> <div class="col-md-6"> Tambah Data </div> </div> <div class="row mt-3"> <div class="col-md-6"> <h2>Daftar Mahasiswa</h2> <ul class="list-group"> <?php foreach(\$mahasiswa as \$mhs):?> <li class="list-group-item"><?= \$mhs['nama'];?> <?php endforeach; ?> </div> </div> </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 data-bbox="303 1176 1388 1989"> <?php defined('BASEPATH') or exit('No direct script access allowed'); class mahasiswa extends CI_Controller { public function __construct(){ parent::__construct(); \$this->load->model('mahasiswa_model'); } public function index() { // \$this->load->model('mahasiswa_model'); // modul untuk 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'); } } </pre> |

3

Setelah itu tambahkan **function** **tambah** pada **controller** **mahasiswa** dan tambahkan **code** berikut.

(Tips : jika menggunakan **vsCode** untuk menduplikat 1 baris dapat menggunakan **alt+shift+panah_bawah**)

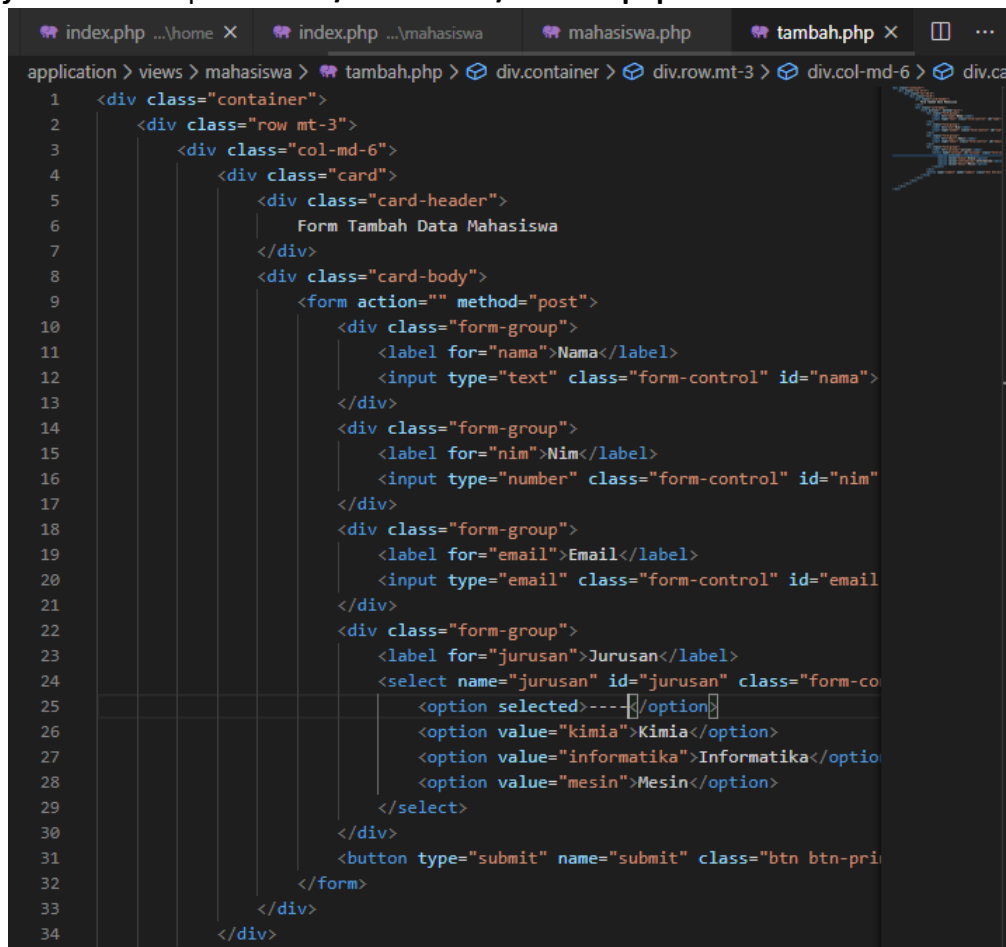
```
public function tambah()
{
    $data['title']='Form Menambahkan Data Mahasiswa';
    $this->load->view('template/header', $data);
    $this->load->view('mahasiswa/index', $data);
}
```

Keterangan :

- Pada **line 32**, **\$this->load->view('mahasiswa/tambah')** artinya bahwa **function** **tambah** akan melakukan **load view** pada folder **mahasiswa**, dengan nama filenya adalah **tambah**

4

Setelah kita membuat **function** **tambah** pada **controller** **mahasiswa**, selanjutnya buatlah **form** **tambah** pada **views/mahasiswa/tambah.php**



```
1 <div class="container">
2 <div class="row mt-3">
3 <div class="col-md-6">
4 <div class="card">
5 <div class="card-header">
6 Form Tambah Data Mahasiswa
7 </div>
8 <div class="card-body">
9 <form action="" method="post">
10 <div class="form-group">
11 <label for="nama">Nama</label>
12 <input type="text" class="form-control" id="nama">
13 </div>
14 <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 name="jurusan" id="jurusan" class="form-control">
25 <option selected="" value="">Pilih Jurusan</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">
32 </button>
33 </form>
34 </div>
35 </div>
36 </div>
```

Keterangan :

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

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


5

Maka akan dihasilkan halaman seperti di bawah ini.

The screenshot shows a web browser window with the address bar displaying 'localhost/Belajar_CI2/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', 'Nim', 'Email', and 'Jurusan' (a dropdown menu). 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

```
<div class="container">
  <div class="row mt-4">
    <div class="col-md-6">
      <a href="<?= base_url();?>mahasiswa/tambah" class="btn btn-primary">Tambah Data</a>
    </div>
  </div>
  <div class="row mt-3">
    <div class="col-md-6">
      <h2>Daftar Mahasiswa</h2>
      <ul class="list-group">
        <?php foreach($mahasiswa as $mhs):?>
          <li class="list-group-item">
            <?= $mhs['nama'];?>
          </li>
        <?php endforeach; ?>
      </ul>
    </div>
  </div>
</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()
{
    $data['title']='Form Menambahkan Data Mahasiswa';

    if($this->form_validation->run() == FALSE){
        $this->load->view('template/header', $data);
        $this->load->view('mahasiswa/tambah', $data);
        $this->load->view('template/footer');
    }
    else{
        echo "data berhasil ditambah";
    }
}
```

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**

```
<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 name="jurusan" id="jurusan" class="form-control">
        <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()
{
    $data['title']='Form Menambahkan Data Mahasiswa';

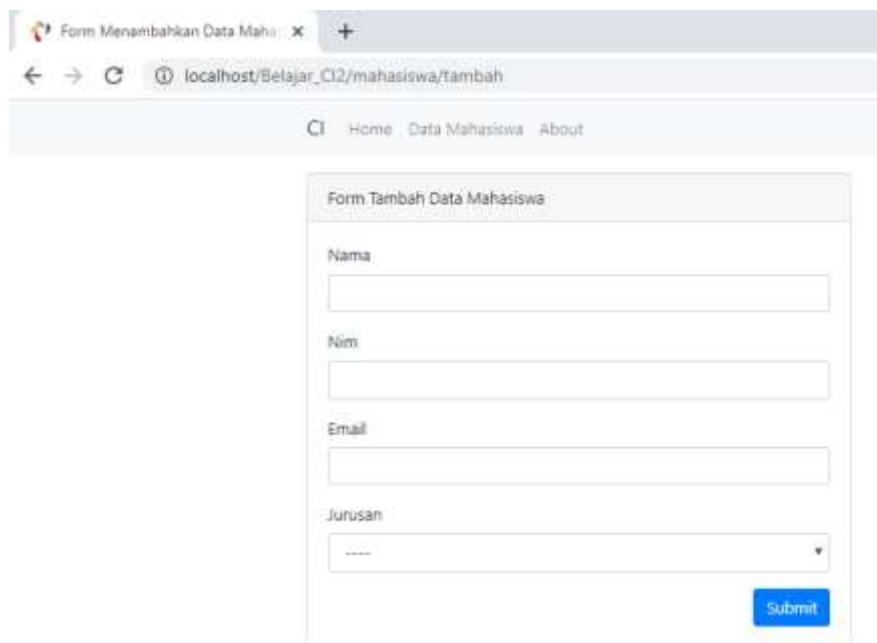
    $this->load->library('form_validation');
    $this->form_validation->set_rules('nama', 'Nama', 'required');

    if($this->form_validation->run() == FALSE){
        $this->load->view('template/header', $data);
        $this->load->view('mahasiswa/tambah', $data);
        $this->load->view('template/footer');
    }
    else{
        echo "data berhasil ditambah";
    }
}
```

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

```
<div class="card-body">
    <?= validation_errors() ?>
    <form action="" method="post">
        <div class="form-group">
            <label for="nama">Nama</label>
            <input type="text" class="form-control" id="nama" name="nama">
        </div>
```

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



The screenshot shows a web browser window with the title 'Form Menambahkan Data Mahasiswa'. The address bar shows 'localhost/Belajar_CI2/mahasiswa/tambah'. The browser's navigation bar includes 'Home', 'Data Mahasiswa', and 'About'. The form itself is titled 'Form Tambah Data Mahasiswa' and contains four input fields: 'Nama', 'Nim', 'Email', and 'Jurusan'. The 'Nama' field is currently empty, and a red error message 'the Nama Field is required' is displayed above it. A blue 'Submit' button is located at the bottom right of the form.

Form Menambahkan Data Mahasiswa

localhost/Belajar_CI2/mahasiswa/tambah

CI Home Data Mahasiswa About

Form Tambah Data Mahasiswa

The Nama field is required.

Nama

Nim

Email

Jurusan

Submit

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">
    <div class="form-group">
      <label for="nama">Nama</label>
      <input type="text" class="form-control" id="nama" name="nama">
    </div>
  </form>
</div>
```

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

i. Akan dihasilkan seperti gambar di bawah ini

The screenshot shows a web browser window with the address bar displaying 'http://Belajar_CI2/mahasiswa/tambah'. The browser's tab is titled 'Form Menambahkan Data Mahasiswa'. The page has a navigation bar with links for 'Home', 'Data Mahasiswa', and 'About'. The main content area is titled 'Form Tambah Data Mahasiswa'. It contains a form with four input fields: 'Nama', 'Nim', 'Email', and 'Jurusan'. A red error message 'The Nama field is required.' is displayed above the 'Nama' field. A blue 'Submit' button is located at the bottom right of the form.

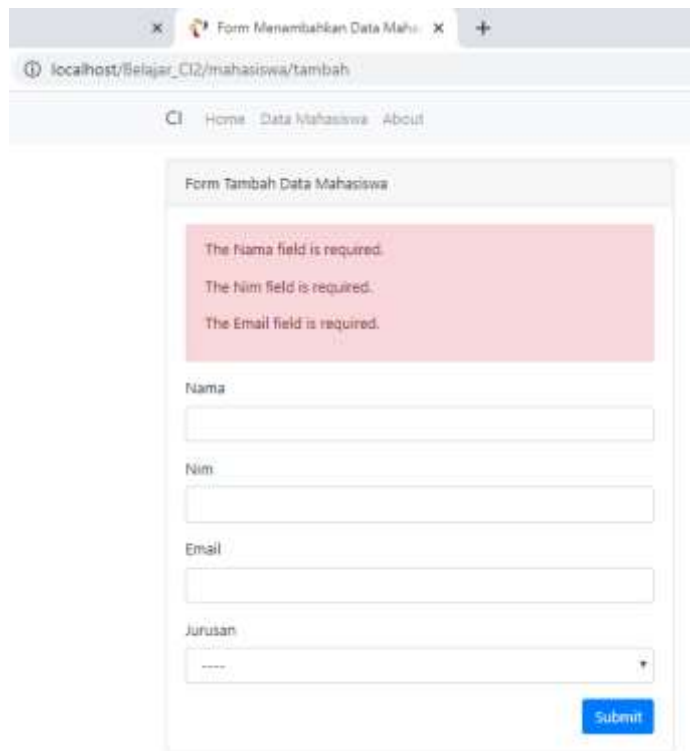
i. Setelah kita berhasil memberikan **rules validation** pada **nama**, maka tambahkan juga untuk **nim** dan **email**.

```
public function tambah()
{
    $data['title']='Form Menambahkan Data Mahasiswa';

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

    if($this->form_validation->run() == FALSE){
        $this->load->view('template/header', $data);
        $this->load->view('mahasiswa/tambah', $data);
        $this->load->view('template/footer');
    }
    else{
        echo "data berhasil ditambah";
    }
}
```

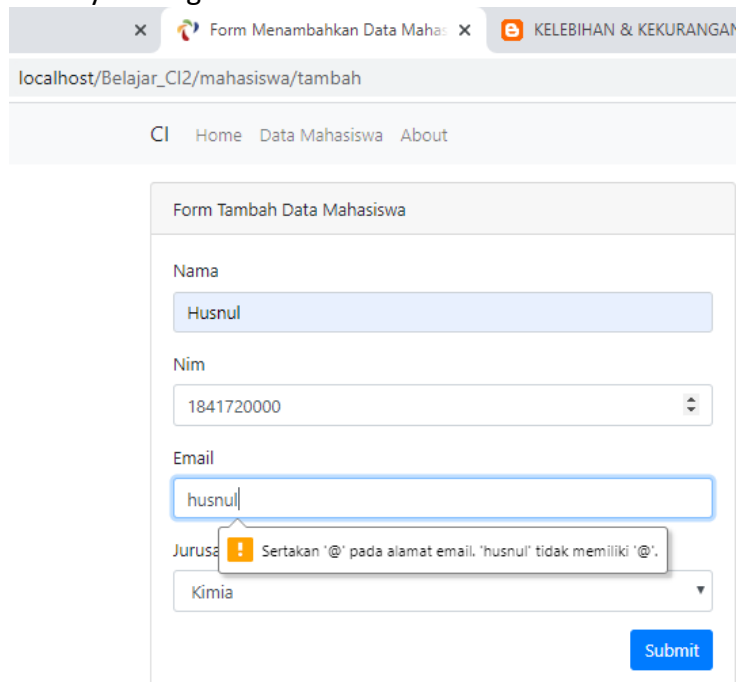

k. Hasilnya seperti berikut ini.



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

```
$this->load->library('form_validation');  
$this->form_validation->set_rules('nama', 'Nama', 'required');  
$this->form_validation->set_rules('nim', 'Nim', 'required');  
$this->form_validation->set_rules('email', 'Email', 'required[valid_email]');
```

m. Hasilnya sebagai berikut.



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

```
$this->load->library('form_validation');  
$this->form_validation->set_rules('nama', 'Nama', 'required');  
$this->form_validation->set_rules('nim', 'Nim', 'required|numeric');  
$this->form_validation->set_rules('email', 'Email', 'required|valid_email');
```

- o. Akan dihasilkan alert seperti berikut ini.

The screenshot shows a web browser window with the address bar displaying 'localhost/Belajar_CI2/mahasiswa/tambah'. The page has a navigation bar with links: 'CI', 'Home', 'Data Mahasiswa', and 'About'. The main content area contains a form titled 'Form Tambah Data Mahasiswa' with the following fields:

- Nama:** A text input field containing 'Husnulccccc'.
- Nim:** A text input field containing '184172001ee'. A validation error alert is displayed above this field with the message 'Masukkan nomor.' (Enter a number).
- Email:** A text input field containing 'husnulhotimah9c@gmail.com'.
- Jurusan:** A dropdown menu with '----' selected.

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)

| Langkah | Keterangan |
|---------|--|
| 1 | <p>Buka controller/mahasiswa dan tambahkan <i>code</i> berikut pada function tambah agar pada saat form_validation bernilai true, maka akan ditambahkan data dengan nama modelnya - > model_mahasiswa dengan nama <i>function</i> tambahdatamhs() dan jika sudah ditambah ke database maka akan <i>redirect</i> ke <i>controller</i> mahasiswa (line 46 - 47)</p> <pre>public function tambah() { \$data['title']='Form Menambahkan Data Mahasiswa'; \$this->load->library('form_validation'); \$this->form_validation->set_rules(['nama', 'Nama', 'required']); \$this->form_validation->set_rules('nim', 'Nim', 'required numeric'); \$this->form_validation->set_rules('email', 'Email', 'required valid_email'); if(\$this->form_validation->run() == FALSE){ \$this->load->view('template/header', \$data); \$this->load->view('mahasiswa/tambah', \$data); \$this->load->view('template/footer'); } else{ \$this->mahasiswa_model->tambahdatamhs(); echo "data berhasil ditambah"; } }</pre> |
| 2 | <p>Jika kita lihat <i>code</i> pada controller mahasiswa(line 46) terdapat model dengan nama mahasiswa_model dan <i>function</i>-nya tambahdatamhs(). Untuk itu kita harus membuat sebuah <i>function</i> dengan nama tambahdatamhs()</p> |
| 3 | <p>Buatlah sebuah <i>function</i> tambahdatamhs() pada models/mahasiswa_model</p> <ul style="list-style-type: none">• https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=insert#inserting-data• Tambahkan <i>code</i> seperti pada gambar dibawah ini  <pre>index.php X mahasiswa.php mahasiswa_model.php X tambah.php application > models > mahasiswa_model.php > mahasiswa_model > tambahdatamhs 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 // \$query=\$this->db->get('mahasiswa'); 10 return \$this->db->get('mahasiswa')->result_array(); 11 } 12 13 public function tambahdatamhs() 14 { 15 \$data = [16 "nama" => \$this->input->post('nama', true), 17 "nim" => \$this->input->post('nim', true), 18 "email" => \$this->input->post('email', true), 19 "jurusan" => \$this->input->post('jurusan', true) 20]; 21 \$this->db->insert('mahasiswa', \$data); 22 } 23 } 24 25 /* End of file mahasiswa_model.php */ 26 27 ?></pre> |

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

The image shows two screenshots of a web application. The first screenshot displays the 'Form Tambah Data Mahasiswa' (Add Student Data Form) at the URL `localhost/Belajar_CI2/mahasiswa/tambah`. The form contains the following fields:

- Nama**: Input field with the value 'Husnul'.
- Nim**: Input field with the value '1841720014'.
- Email**: Input field with the value 'husnulhotimah9c@gmail.com'.
- Jurusan**: Dropdown menu with the selected value 'Informatika'.
- Submit**: A blue button to submit the form.

Below the form, a message states 'data berhasil ditambah' (data successfully added).

The second screenshot shows the 'List Mahasiswa' (List Students) page at the URL `localhost/Belajar_CI2/mahasiswa`. It features a 'Tambah Data' (Add Data) button and a table titled 'Daftar Mahasiswa' (Student List):

| |
|--------|
| dina |
| dika |
| soni |
| Husnul |
| Husnul |