

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT

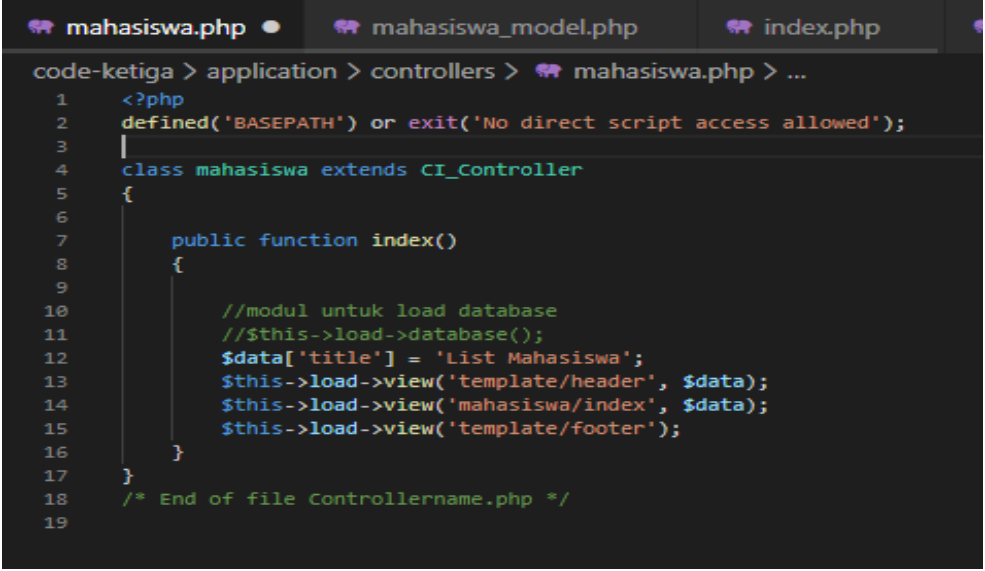
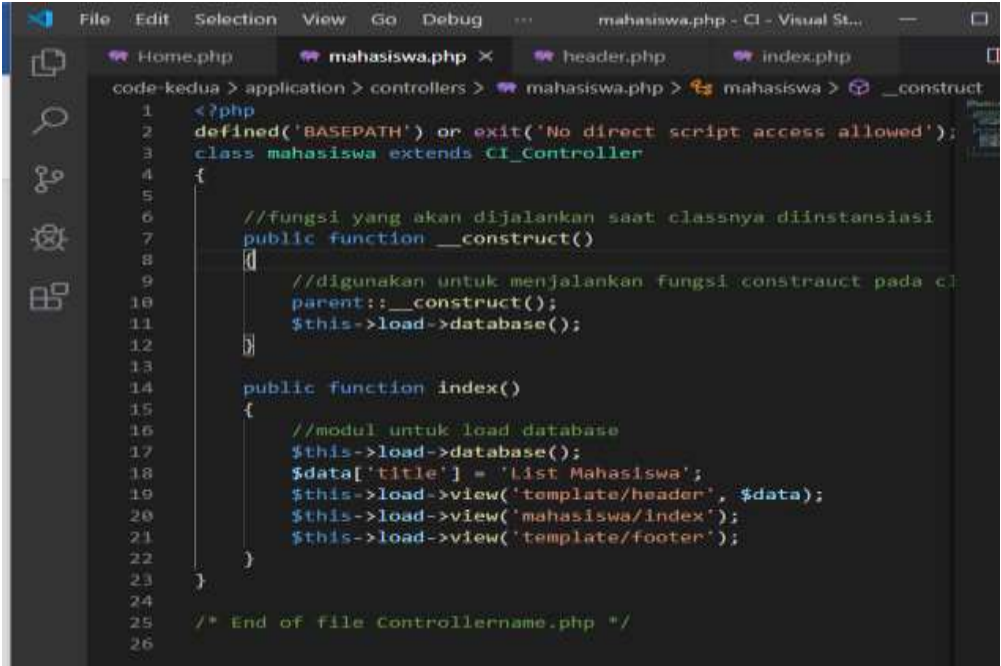
Jobsheet-2
Read dan Input Database serta Form Validation

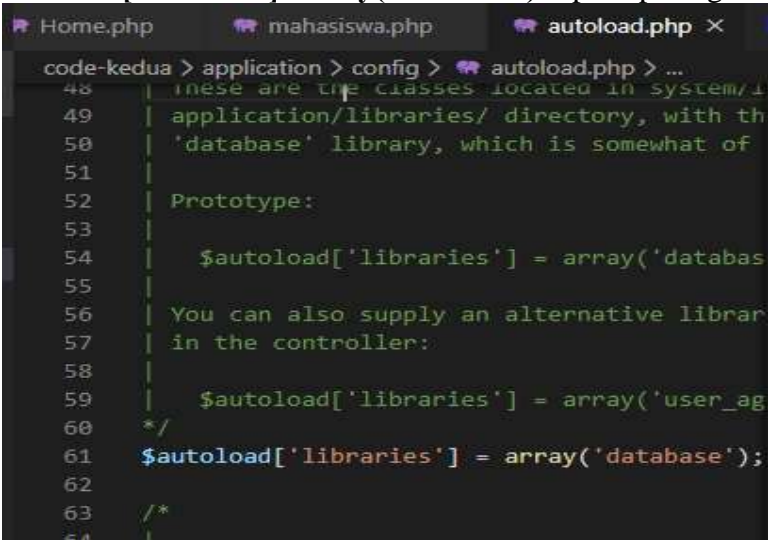
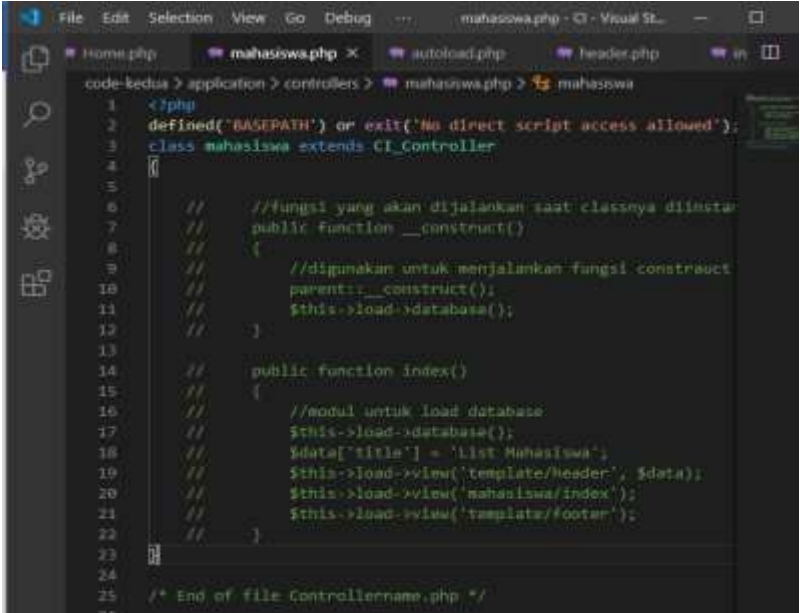


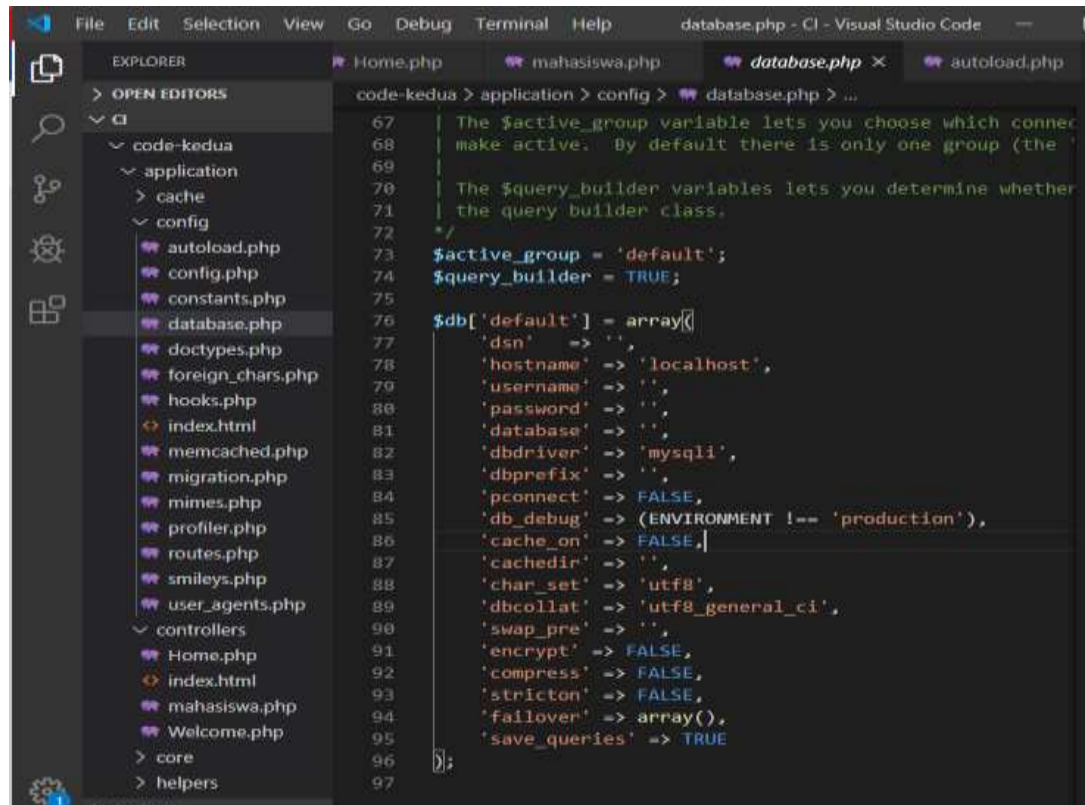
NAMA : DINA RISKY ALIN SAPUTRI
NIM : 1841720016
KELAS : 2B

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020

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) gunanya untuk melakukan <i>load</i> database dan <i>load</i> database ini dilakukan di <i>function</i> <i>index</i> saja.</p>  <pre>code-ketiga > 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 7 public function index() 8 { 9 10 11 //modul untuk load database 12 //\$this->load->database(); 13 \$data['title'] = 'List Mahasiswa'; 14 \$this->load->view('template/header', \$data); 15 \$this->load->view('mahasiswa/index', \$data); 16 \$this->load->view('template/footer'); 17 } 18 /* End of file Controllername.php */ 19 </pre>
3	<p>Tambahkan <i>code</i> berikut agar database dapat digunakan banyak <i>function</i> (line 7-12 merupakan <i>function construct</i>). Yang dijalankan pada class parent dimana parent ini adalah <i>CI_Controller</i>.</p>  <pre>File Edit Selection View Go Debug ... mahasiswa.php - CI - Visual St... Home.php mahasiswa.php x header.php index.php code-kedua > application > controllers > mahasiswa.php > mahasiswa > _construct 1 <?php 2 defined('BASEPATH') or exit('No direct script access allowed'); 3 class mahasiswa extends CI_Controller 4 { 5 6 7 //fungsi yang akan dijalankan saat classnya diinstansiasi 8 public function __construct() 9 { 10 11 //digunakan untuk menjalankan fungsi constrauct pada cl 12 parent::__construct(); 13 \$this->load->database(); 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 /* End of file Controllername.php */ 26 </pre>

4	<p>Selain 2 cara diatas kita juga dapat melakukan proses <i>load</i> database pada application/config/autoload dan tambahkan pada database pada autoload['libraries']=array('database') seperti pada gambar</p>  <p>The screenshot shows the <code>autoload.php</code> file in a code editor. It displays the configuration for loading the database library. The code includes comments in Indonesian explaining the process and the <code>\$autoload['libraries'] = array('database');</code> line.</p>
5	<p>Untuk kasus ini kita gunakan cara yang terakhir (no 4) karena kita asumsikan bahwa semua <i>controller</i> membutuhkan database. Maka kita <i>comment code</i> function construct yang ada di <i>controller mahasiswa</i></p>  <p>The screenshot shows the <code>mahasiswa.php</code> controller file. It includes the <code>__construct()</code> method which is commented out, and the <code>index()</code> method which calls the <code>load->database()</code> function.</p>
6	<p>Setelah itu atur konfigurasi database. Maka buka folder application/config/database.php</p>



7

Set konfigurasinya seperti pada gambar dibawah ini

```

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 ]);
97

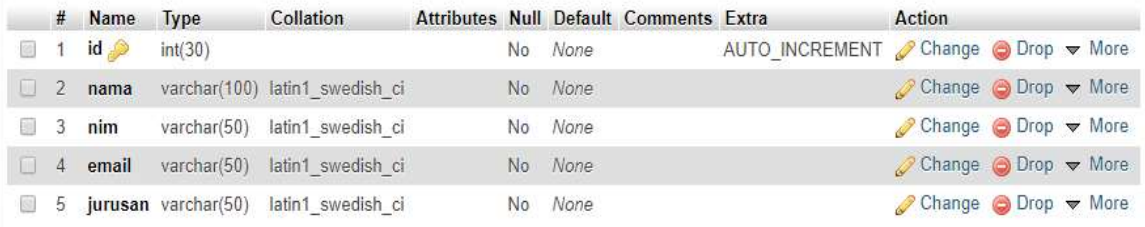
```

Keterangan :

- Line 78 -> hostname yang digunakan adalah localhost
- Line 81 -> nama database yang nanti digunakan (**kuliah_CI**)

8

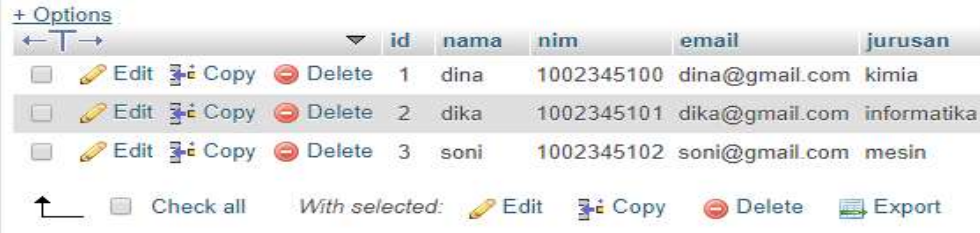
Buat sebuah database dengan nama **kuliah_CI** dan tabel mahasiswa seperti pada gambar dibawah ini



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

9

Tambahkan beberapa data ke dalam tabel mahasiswa seperti pada gambar.

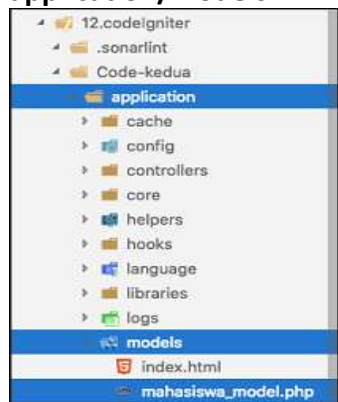


	id	nama	nim	email	jurusan
Edit Copy Delete	1	dina	1002345100	dina@gmail.com	kimia
Edit Copy Delete	2	dika	1002345101	dika@gmail.com	informatika
Edit Copy Delete	3	soni	1002345102	soni@gmail.com	mesin

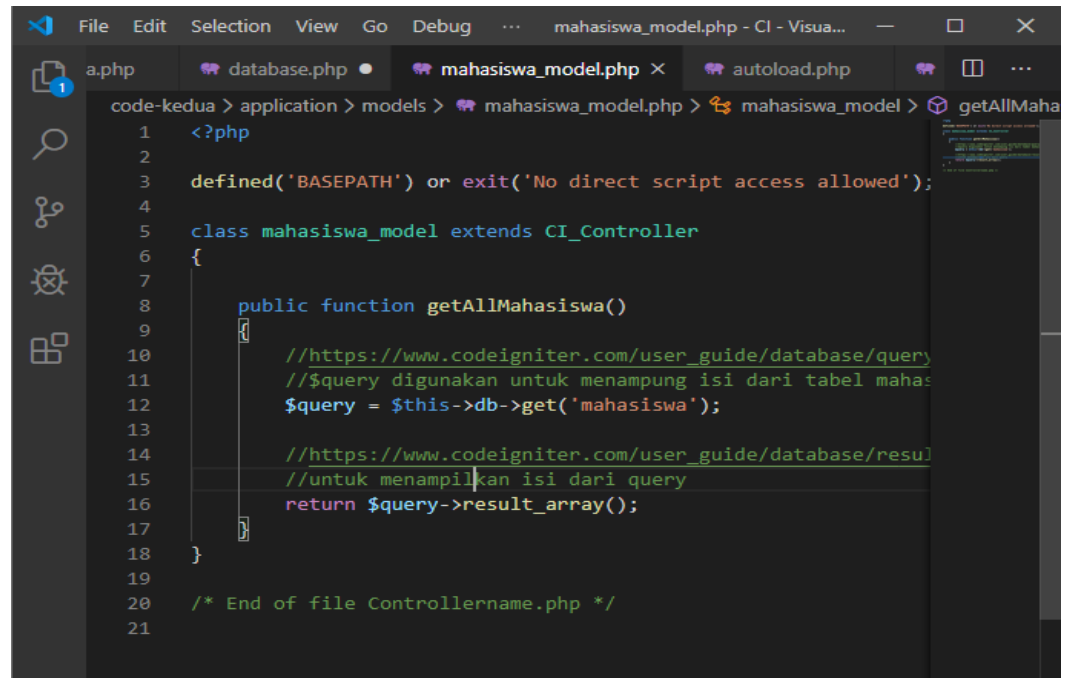
10

Setelah kita membuat sebuah *database*, maka membuat sebuah model untuk melakukan proses *database*. Di dalam model ini kita akan membuat sebuah *function* untuk melakukan perintah ambil data dari *database*.

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



b. Tambahkan *function* getAllmahasiswa pada mahasiswa_model.php

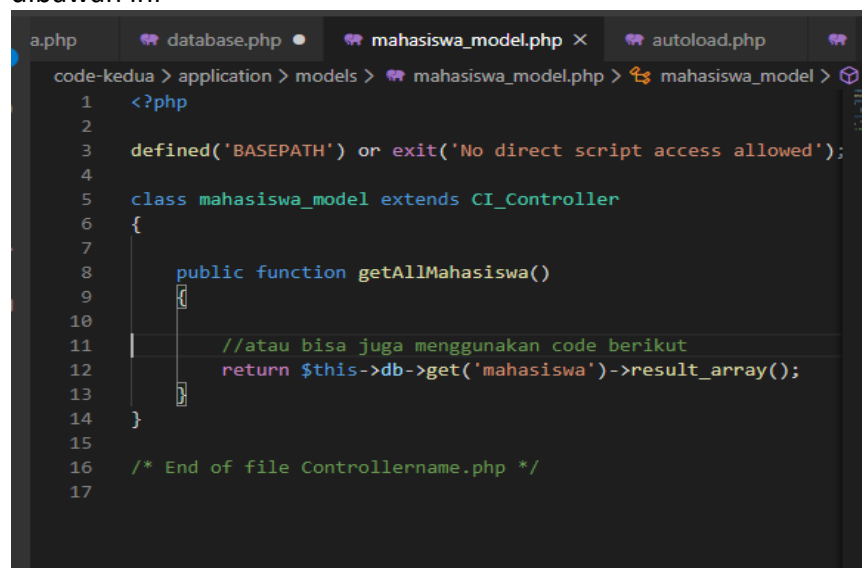


```
1 <?php
2
3 defined('BASEPATH') or exit('No direct script access allowed');
4
5 class mahasiswa_model extends CI_Controller
6 {
7
8     public function getAllMahasiswa()
9     {
10         //https://www.codeigniter.com/user_guide/database/query
11         // $query digunakan untuk menampung isi dari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

Keterangan :

- Pada line 12 membuat sebuah query *builder* yang digunakan untuk mengambil semua data
- https://www.codeigniter.com/user_guide/database/query_builder.html#selecting-data
- Pada line 16 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



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

```

13
14     public function index()
15     {
16         $this->load->model('mahasiswa_model');
17
18         //modul untuk load database
19         //$this->load->database();
20         $data['title'] = 'List Mahasiswa';
21         $data['mahasiswa'] = $this->mahasiswa_model->getAllmahasiswa();
22         $this->load->view('template/header', $data);
23         $this->load->view('mahasiswa/index', $data);
24         $this->load->view('template/footer');
25     }
26 }
27
28 /* End of file Controllername.php */
29

```

Keterangan :

- Line 16 digunakan untuk melakukan *load* model dengan nama modelnya adalah **mahasiswa_model**
- Line 21 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 23 tambahkan **\$data**

The image shows two code editors. The top editor displays the `mahasiswa_model.php` file, which contains a `class mahasiswa_model extends CI_Model` with a `getAllmahasiswa()` function. The bottom editor displays the `index.php` file, which contains a `class mahasiswa extends CI_Controller` with an `index()` function. A green arrow points from the `getAllmahasiswa()` function in the model to its usage in the controller. A red arrow points from the `load->model('mahasiswa_model')` call in the controller to the `mahasiswa_model.php` file tab in the editor's tab bar.

12

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

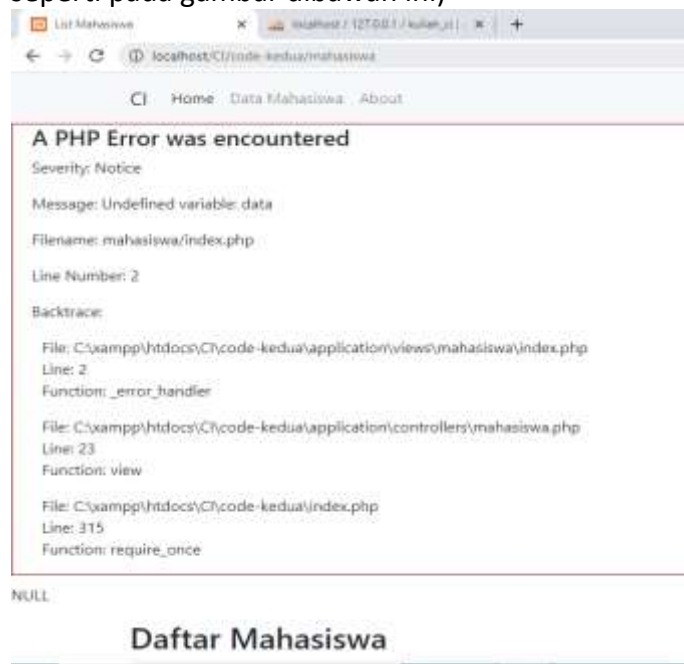
```

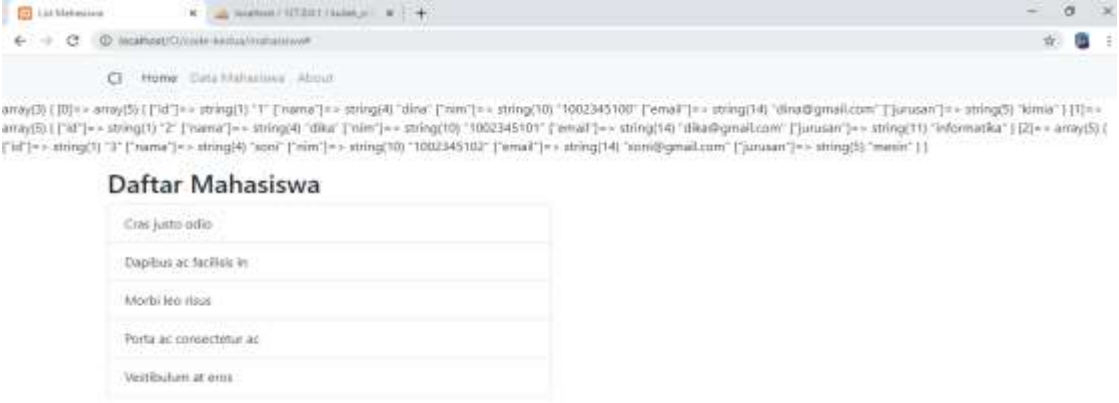
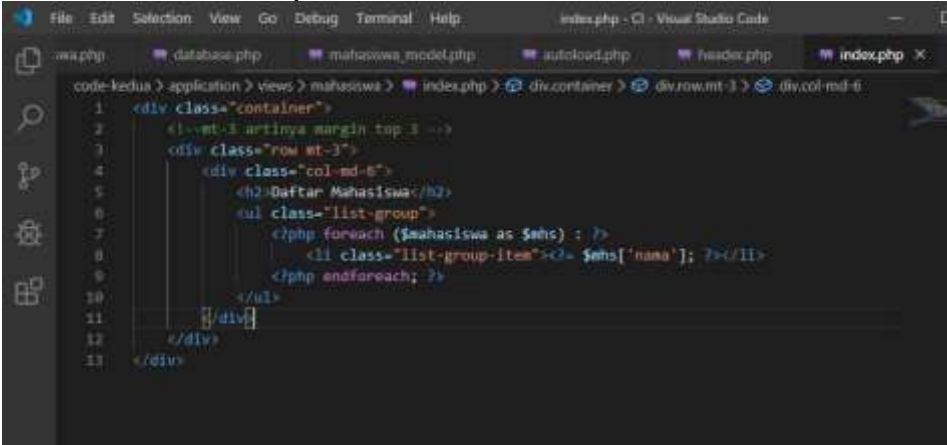

1 <?php
2 var_dump($mahasiswa);
3
4 <div class="container">
5 <!-- mt-3 artinya margin top 3 -->
6 <div class="row mt-3">
7 <div class="col-md-6">
8 <h2>Daftar Mahasiswa</h2>
9 <ul class="list-group">
10 <li class="list-group-item">Cras justo odio</li>
11 <li class="list-group-item">Dapibus ac facilisis in</li>
12 <li class="list-group-item">Morbi leo risus</li>
13 <li class="list-group-item">Porta ac consectetur ac</li>
14 <li class="list-group-item">Vestibulum at eros</li>
15 </ul>
16 </div>
17 </div>
18 </div>

```

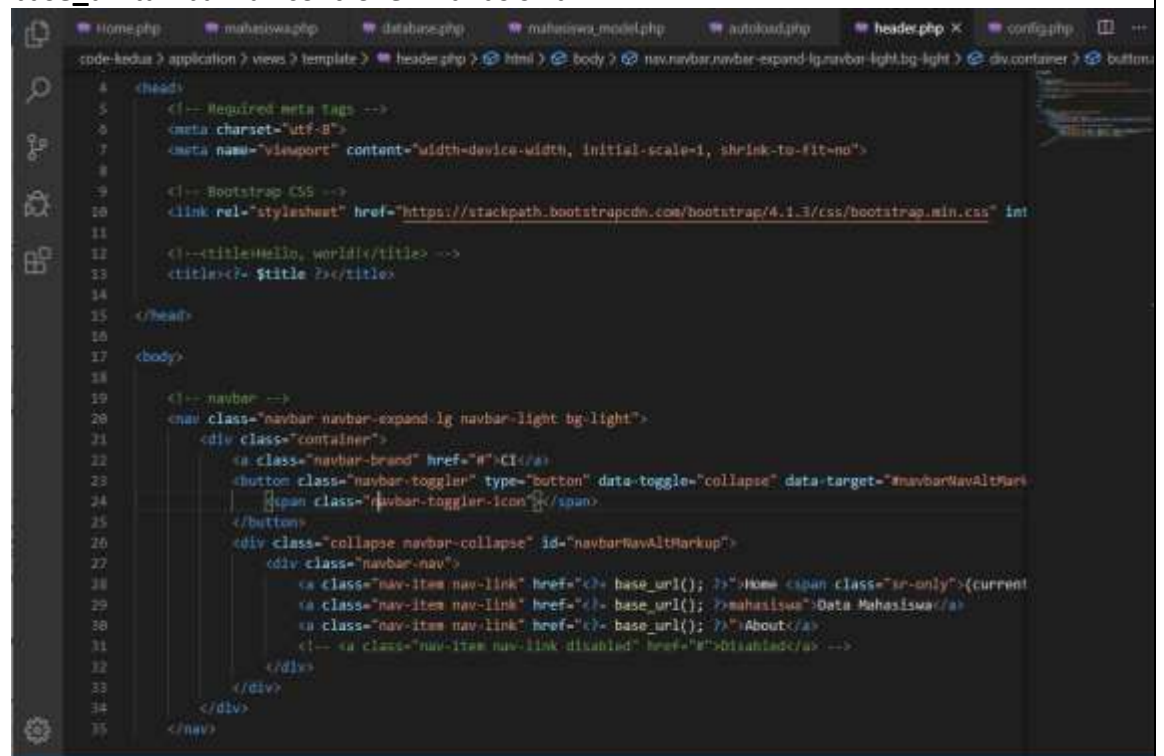
Keterangan:

- \$mahasiswa didapat dari mana ??? -> jika kita lihat *line 21* 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)



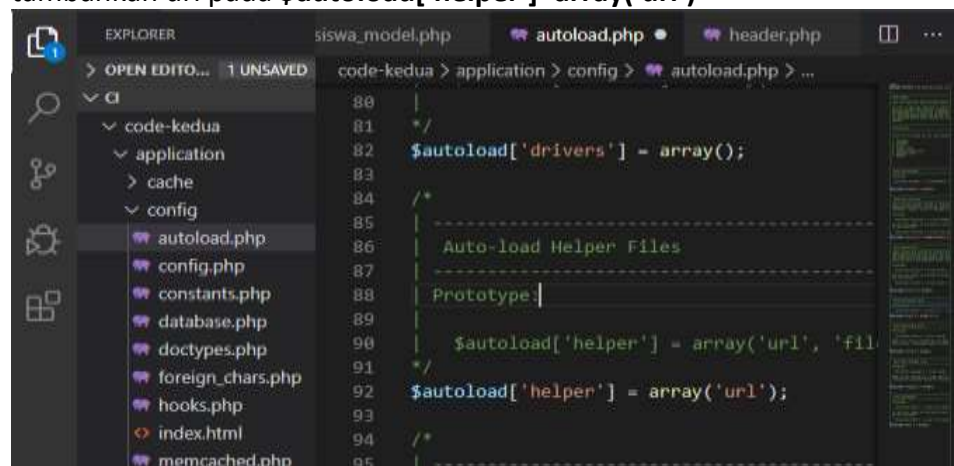
	<p>Hal ini disebabkan karena tidak ketahui data pada mahasiswa/index.php pada <i>line</i> 2 berupa <code>var_dump(\$mahasiswa)</code>, maka pastikan terlebih dahulu pada <i>controller</i> mahasiswa.php sudah terdapat <code>\$data</code> di <i>line</i> 13 dan 15.</p>
13	<p>Jika berhasil maka data akan tampil seperti pada gambar berikut</p> 
14	<p>Lakukan proses looping agar semua isi dari tabel dapat tampil di halaman index mahasiswa. Gunakan <i>code</i> berikut pada views/mahasiswa/index</p> 
15	<p>Jika berhasil maka tampilan akan seperti pada gambar dibawah ini</p> 
16	<p>Jika kita klik Home, halaman Home belum dapat tampil. Set agar semua dapat saling terhubung. Untuk itu kita butuh yang namanya base_url.</p>

- a. Tambahkan *code* berikut pada **views/template/header (line 28, 29 , 30)** Agar halaman web mahasiswa dapat ditampilkan saat *user* mengklik mahasiswa, maka setelah **base_url** tambahkan **controller mahasiswa**



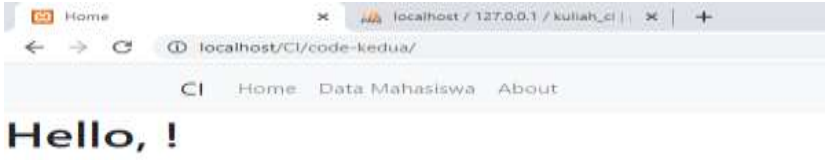


```
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8 <!-- Bootstrap CSS -->
9 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" int
10
11 <!--<title>Hello, world!</title> -->
12 <title><? $title ?></title>
13
14
15 </head>
16
17 <body>
18
19 <!-- navbar -->
20 <nav class="navbar navbar-expand-lg navbar-light bg-light">
21   <div class="container">
22     <a class="navbar-brand" href="#">CI</a>
23     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMark
24       <span class="navbar-toggler-icon"></span>
25   </button>
26   <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
27     <div class="navbar-nav">
28       <a class="nav-item nav-link" href="{<? base_url(); ?>">Home <span class="sr-only">{current
29       <a class="nav-item nav-link" href="{<? base_url(); ?>mahasiswa">Data Mahasiswa</a>
30       <a class="nav-item nav-link" href="{<? base_url(); ?>">About</a>
31     <!-- <a class="nav-item nav-link disabled" href="#">Disabled</a> -->
32     </div>
33   </div>
34 </div>
35 </nav>
```

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



```
80 |
81 | */
82 | $autoload['drivers'] = array();
83 |
84 | /*
85 | -----
86 | Auto-load Helper Files
87 | -----
88 | Prototype:|
89 |
90 | $autoload['helper'] = array('url', 'fil
91 | */
92 | $autoload['helper'] = array('url');
93 |
94 | /*
95 | -----
```

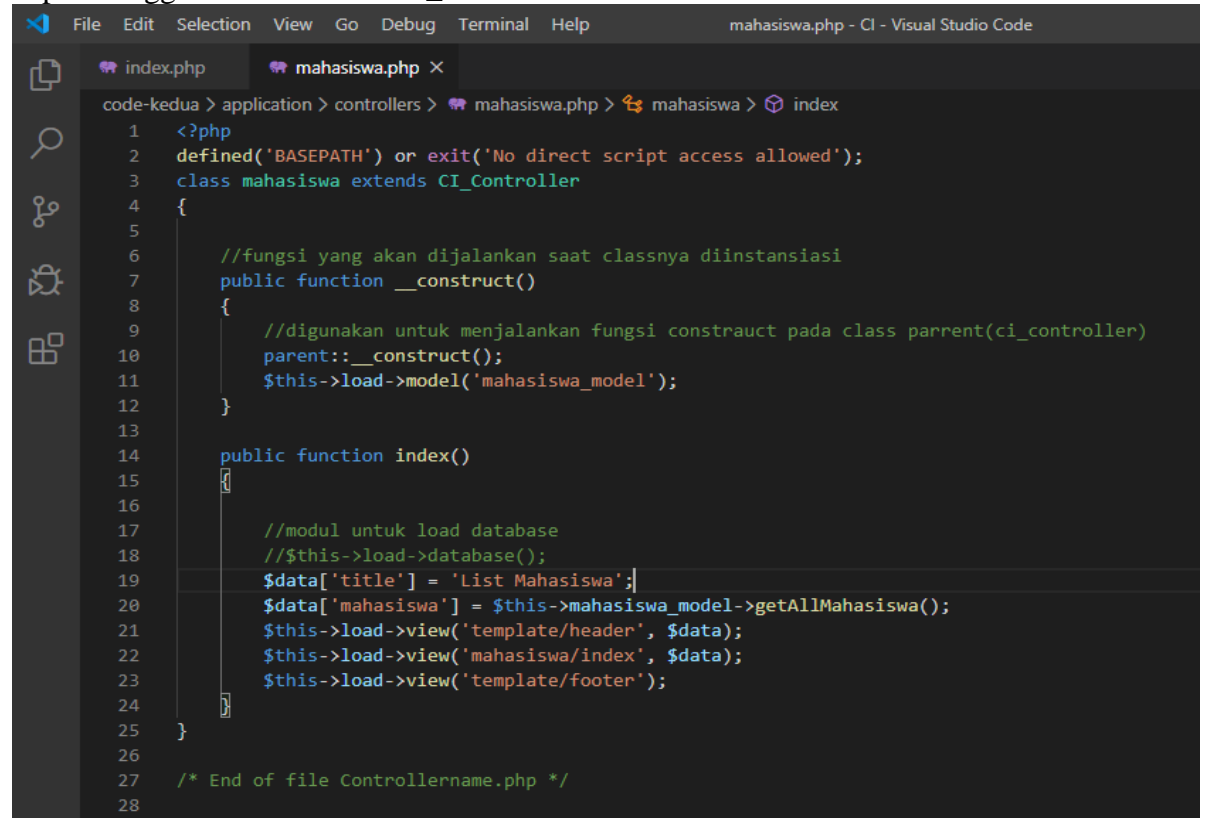
- c. Jika berhasil maka semua halaman dapat ditampilkan pada saat *user* melakukan *event click*.
- Pada Home

	 <p>- Pada Data Mahasiswa</p>  <p>- Pada About</p> 
--	--

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

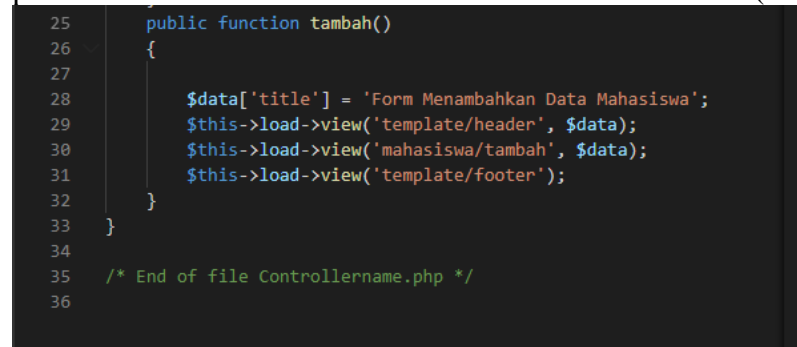
Langkah	Keterangan
1	<p>Buatlah sebuah <i>button</i> tambah data pada <code>views/mahasiswa/index.php</code> (line 3-7).</p> 
2	<p>Dikarenakan mahasiswa_model ini nantinya akan digunakan oleh semua <i>function</i> di <i>controller</i> mahasiswa, maka “ <code>\$this->load->model('mahasiswa_model');</code> “ yang awalnya</p>

hanya berada pada **function index**, kita ubah ke **function construct** agar semua *function* dapat menggunakan **mahasiswa_model**.



```
1 <?php
2 defined('BASEPATH') or exit('No direct script access allowed');
3 class mahasiswa extends CI_Controller
4 {
5
6     //fungsi yang akan dijalankan saat classnya diinstansiasi
7     public function __construct()
8     {
9         //digunakan untuk menjalankan fungsi construct pada class parrent(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    }
25 }
26
27 /* End of file Controllername.php */
28
```

- 3 Setelah itu tambahkan **function tambah** akan melakukan *load view* pada folder mahasiswa pada **controller mahasiswa** dan tambahkan *code* berikut. (line 30)



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

- 4 Buatlah **form tambah** pada **views/mahasiswa/tambah.php** Pada (line 33) terdapat atribut class dengan *value* btn btn-primary float-right yang artinya membuat *button* dan posisinya ada dikanan.

```

code-kedua > application > views > mahasiswa > tambah.php > div.container > div.row.mt-3 > div.col-md-6
1 <div class="container">
2   <div class="row mt-3">
3     <div class="col-md-6">
4       <!-- http://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            <!-- http://getbootstrap.com/docs/4.1/components/form/ -->
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="number" id="jurusan" name="jurusan">
27                <option selected="" value="1">-----</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 :

- Ganti **value** dari *line* 28-31 menjadi seperti di bawah ini (Agar di tabel tidak tersimpan **value** berupa angka)

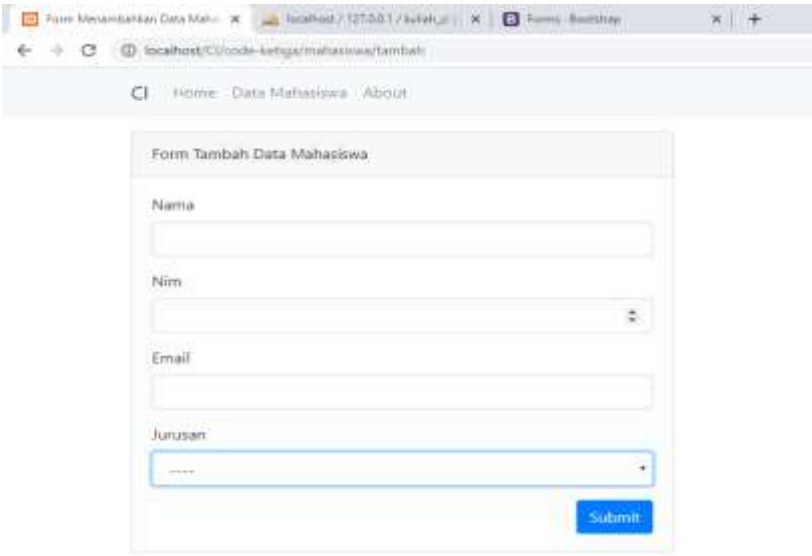
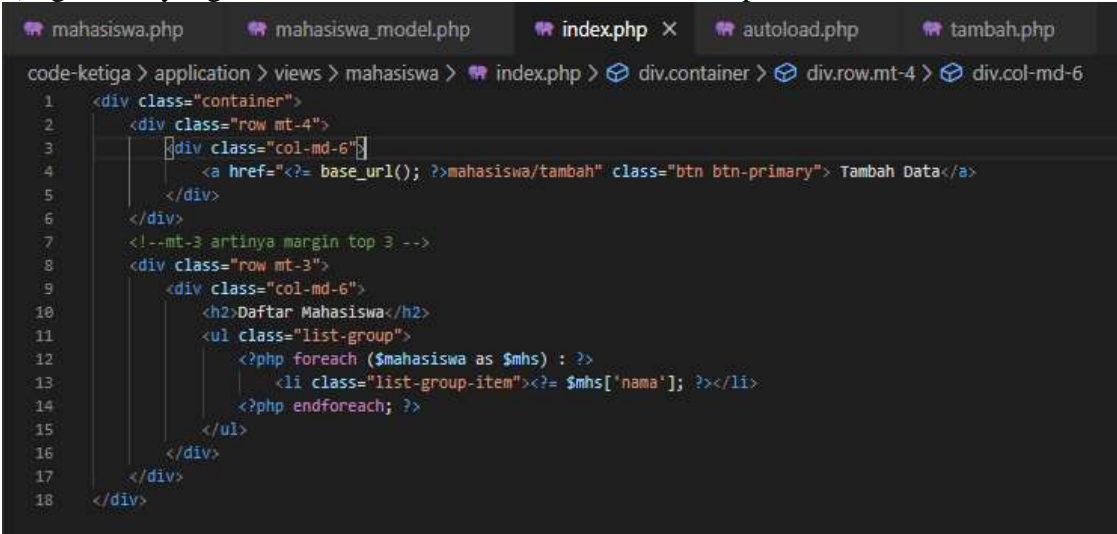
```

<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>Selanjutnya buka file index pada views/mahasiswa/index.php dan kita atur href nya (line 4) agar user yang melakukan <i>event click</i> maka akan ditampilkan di halaman tambah.</p> 
7	<p>Tambahkan validasi menggunakan <code>form_validation</code>. Contoh validasi <i>form</i> adalah seperti <i>field</i> nama harus diisi, nim harus angka dsb.</p> <ol style="list-style-type: none"> Tambahkan <code>form_validation</code> https://www.codeigniter.com/user_guide/libraries/form_validation.html?highlight=form#the-form (searching controller) Buka <i>controller</i> mahasiswa dan tambahkan <code>form_validation</code>


```

mahasiswa.php X tambah.php code-ketiga... index.php code-ketiga...
code-kedua > application > controllers > mahasiswa.php > mahasiswa > tambah
24 }
25
26 public function tambah()
27 {
28
29     $data['title'] = 'Form Menambahkan Data Mahasiswa';
30
31     if ($this->$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
42
43 /* End of file Controllername.php */
44

```

- 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 class="form-control" id="jurusan" name="jurusan">
        <option selected>----</option>
        <option value="Informatika">Informatika</option>
        <option value="Kimia">Kimia</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. (berwarna kuning)

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

    $this->load->library('form_validation');

    //$this->form_validation->set_rules('fielname', 'fieldlabel', 'trim');
    $this->form_validation->set_rules('nama', 'Nama', 'required');

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

/* End of file Controllernama.php */
```

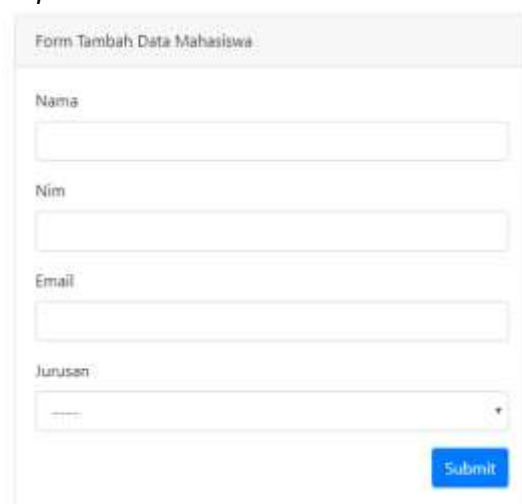
- f. Edit autoload.php dan tambahkan '**form_validation**' untuk memanggil libraries pada **\$autoload['libraries']**

```
9 | $autoload['libraries'] = array('user_agent' => 'ua');
10 */
11 $autoload['libraries'] = array('database', 'form_validation');
12
13 /*
14 | -----
```

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

```
<div class="card-body">
    <?= validation_errors() ?>
    <form action="" method="post">
        <!-- http://getbootstrap.com/docs/4.1/components/form/ -->
        <div class="form-group">
            <label for="Nama">Nama</label>
            <input type="text" class="form-control" id="nama" name="nama">
        </div>
    </form>
</div>
```

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



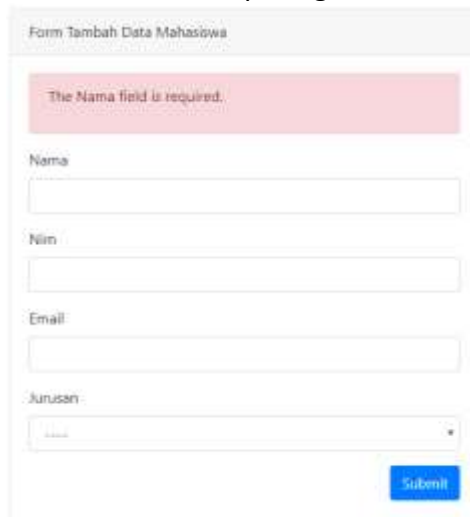
i. Tambahkan *alert* agar pesan *error* lebih menarik setelah *code validation error*

- <https://getbootstrap.com/docs/4.1/components/alerts/>

```
<div class="card-body">
    <!-- untuk menampilkan pesan error -->
    <?php if (validation_errors()) : ?>
        <div class="alert alert-danger" role="alert">
            <?= validation_errors() ?>
        </div>
    <?php endif ?>
    <form action="" method="post">
        <!-- http://getbootstrap.com/docs/4.1/components/form/ -->
        <div class="form-group">
            <label for="Nama">Nama</label>
            <input type="text" class="form-control" id="nama" -->
        </div>
    </form>
</div>
```

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

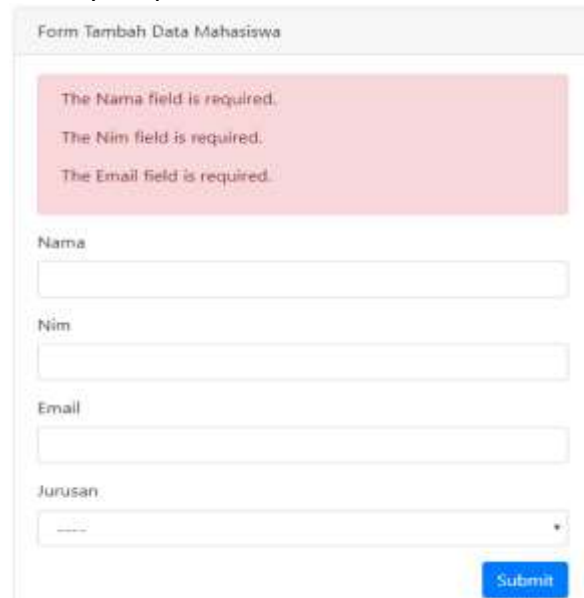
j. Akan dihasilkan seperti gambar di bawah ini



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

```
25 public function tambah()
26 {
27
28     $data['title'] = 'Form Menambahkan Data Mahasiswa';
29
30     $this->load->library('form_validation');
31
32     // $this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_length[5]|max_length[12]');
33     $this->form_validation->set_rules('nama', 'Nama', 'required');
34     $this->form_validation->set_rules('nim', 'Nim', 'required');
35     $this->form_validation->set_rules('email', 'Email', 'required');
36
37     if ($this->form_validation->run() == FALSE) {
38         # code...
39         $this->load->view('template/header', $data);
40         $this->load->view('mahasiswa/tambah', $data);
41         $this->load->view('template/footer');
42     } else {
43         # code...
44         echo "data berhasil ditambah";
45     }
46 }
47
48 /* End of file controllersname.php */
49
```

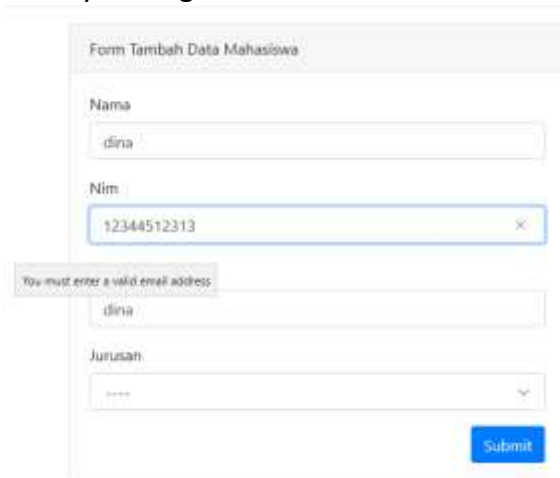
l. Hasilnya seperti berikut ini.



m. Kita juga dapat mengecek ke *valid* an email dengan menambahkan rule pada email seperti pada gambar dibawah ini menggunakan **valid_email**.

```
// $this->form_validation->set_rules('fielname', 'fieldlabel', 'trim|required|min_length[5]|max_length[12]')  
$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');
```

n. Hasilnya sebagai berikut.



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

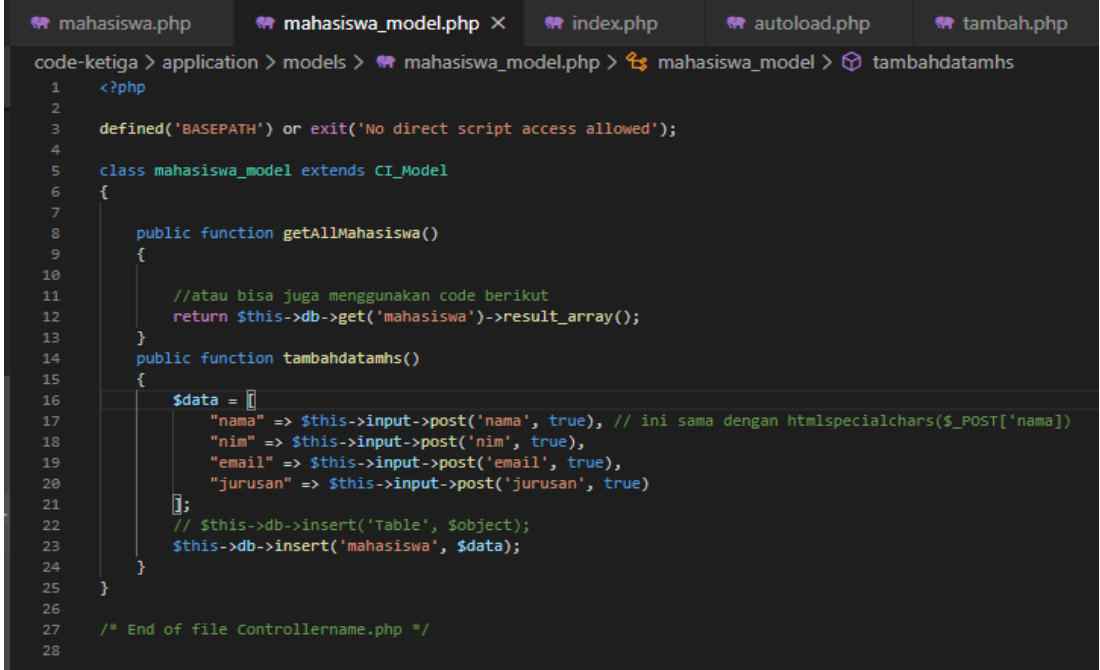
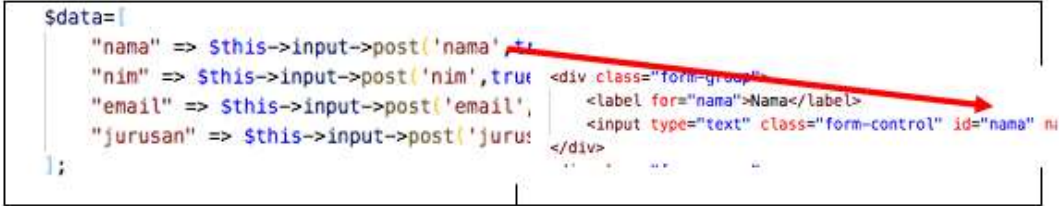
```
31  
32 // $this->form_validation->set_rules('fielname', 'fieldlabel', 'trim|required|min_length[5]|max_length[12]')  
33 $this->form_validation->set_rules('nama', 'Nama', 'required');  
34 $this->form_validation->set_rules('nim', 'Nim', 'required|numeric');  
35 $this->form_validation->set_rules('email', 'Email', 'required|valid_email');
```

p. Akan dihasilkan alert seperti berikut ini.

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

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

Langka h	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 44 - 45)</p> <pre> 25 public function tambah() 26 { 27 28 \$data['title'] = 'Form Menambahkan Data Mahasiswa'; 29 30 \$this->load->library('form_validation'); 31 32 // \$this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim required min_length[5] max_length[33 \$this->form_validation->set_rules('nama', 'Nama', 'required'); 34 \$this->form_validation->set_rules('nim', 'Nim', 'required numeric'); 35 \$this->form_validation->set_rules('email', 'Email', 'required valid_email'); 36 37 if (\$this->form_validation->run() == FALSE) { 38 # code... 39 \$this->load->view('template/header', \$data); 40 \$this->load->view('mahasiswa/tambah', \$data); 41 \$this->load->view('template/footer'); 42 } else { 43 # code... 44 \$this->mahasiswa_model->tambahdatamhs(); 45 redirect('mahasiswa', 'refresh'); 46 } 47 } 48 } 49 /* End of file Controllername.php */ 50 </pre>
2	<p>Jika kita lihat <i>code</i> pada controller mahasiswa(line 46) terdapat model dengan nama</p>

	<p>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  <p>Keterangan:</p> <ul style="list-style-type: none"> - Line 16-21 adalah sebuah array dimana nantinya isi array adalah data-data yang akan dikirim (POST) ke database - Line 17 "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  <ul style="list-style-type: none"> • Line 29 adalah proses <i>insert</i> data ke database dengan 2 parameter yaitu nama tabelnya 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>

b. Hasil

CI Home Data Mahasiswa About

Form Tambah Data Mahasiswa

Nama

Elsa

Nim

184170090

Email

elsa25@gmail.com

Jurusan

Informatika

Submit

CI Home Data Mahasiswa About

Tambah Data

Daftar Mahasiswa

dina

dika

soni

dina

Agit Ari

Silvi

Elsa