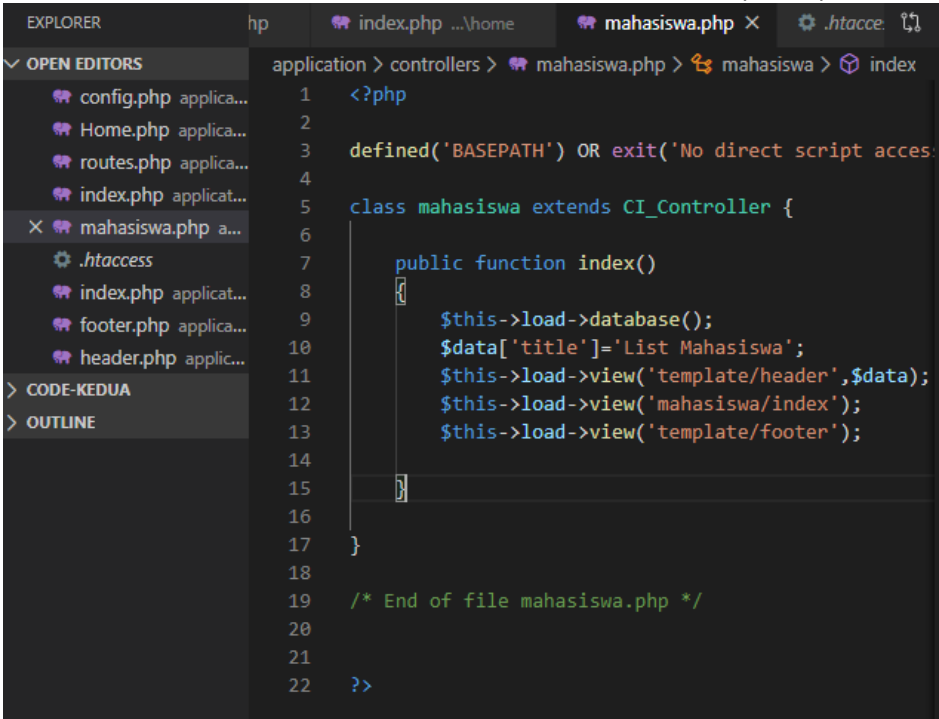




Nama : Agsal Fairrohmad A.P

No : 04/ TI2A

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

Langkah	Keterangan
1	Buka kembali folder CodeIgniter yang telah dibuat pada Jobsheet sebelumnya
2	<p>Buka <i>controller</i> mahasiswa dan tambahkan <i>code</i> berikut (<i>line 9</i>)</p>  <pre>EXPLORER hp index.php ...home mahasiswa.php X .htacce: 1 OPEN EDITORS application > controllers > mahasiswa.php > mahasiswa > index config.php applica... 1 <?php Home.php applica... 2 routes.php applica... 3 defined('BASEPATH') OR exit('No direct script acces index.php applicat... 4 mahasiswa.php a... 5 class mahasiswa extends CI_Controller { .htaccess 6 index.php applicat... 7 public function index() 8 { 9 \$this->load->database(); 10 \$data['title']='List Mahasiswa'; 11 \$this->load->view('template/header',\$data); 12 \$this->load->view('mahasiswa/index'); 13 \$this->load->view('template/footer'); 14 15 16 17 } 18 19 /* End of file mahasiswa.php */ 20 21 22 ?></pre> <p>Keterangan :</p> <ul style="list-style-type: none">- Pada line 9 menjelaskan bawa <i>function</i> index ini akan melakukan <i>load</i> database. <i>Load</i> database ini hanya dilakukan di <i>function</i> index saja jika ada <i>function</i> lain maka kita harus melakukan <i>load</i> database lagi. Sehingga hal ini dirasa tidak praktis maka untuk itu kita harus tambahkan <i>code</i> agar <i>load</i> database ini dapat digunakan banyak <i>function</i>.
3	Tambahkan <i>code</i> berikut agar database dapat digunakan banyak <i>function</i> (<i>line 7-12</i>)

```

index.php ...\home mahasiswa.php .htacce
application > controllers > mahasiswa.php > ...
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access:
4
5  class mahasiswa extends CI_Controller {
6
7      public function __construct()
8      {
9          //digunakan untuk menjalankan fungsi constr
10         parent::__construct()
11         public function __construct()
12         {
13             parent::__construct();
14             $this->load->database();
15         }
16     }
17
18     public function index()
19     {
20         //modul untuk database
21         //$this->load->database();
22         $data['title']='List Mahasiswa';
23         $this->load->view('template/header',$data);
24         $this->load->view('mahasiswa/index');
25         $this->load->view('template/footer');
26     }
27
28
29 }

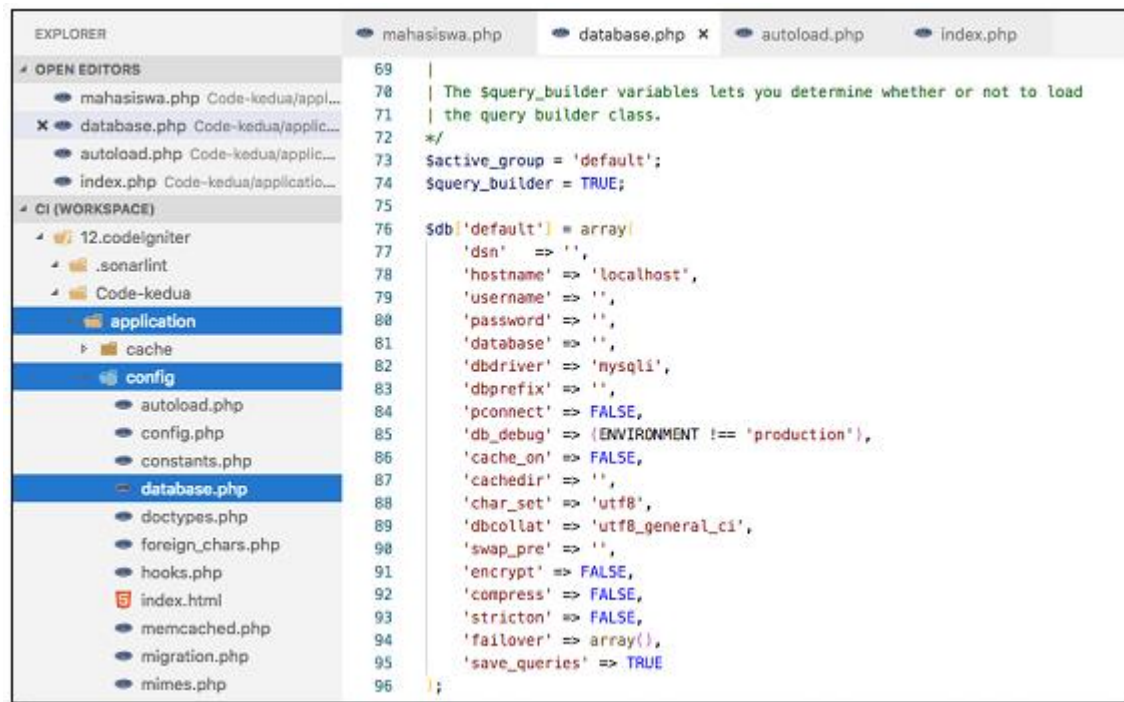
```

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

	<pre> application > config > autoload.php > ... 48 these are the classes located in system/libraries, 49 application/libraries/ directory, with the addition 50 'database' library, which is somewhat of a special 51 52 Prototype: 53 54 \$autoload['libraries'] = array('database', 'email', 'session'); 55 56 You can also supply an alternative library name to 57 in the controller: 58 59 \$autoload['libraries'] = array('user_agent' => 'php_session'); 60 61 \$autoload['libraries'] = array('database'); 62 63 </pre>
5	<p>Untuk kasus ini kita gunakan cara yang terakhir (no 4) karena kita asumsikan bahwa semua <i>controller</i> membutuhkan database. Maka kita <i>comment code</i> function construct yang ada di <i>controller</i> mahasiswa</p> <pre> 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 parrent(ci_controller) 10 // parent::__construct(); 11 // \$this->load->database(); 12 // } </pre>
6	<p>Setelah itu kita akan mengatur konfigurasi database. Untuk mengatur konfigurasi database maka buka folder application/config/database.php</p>



7

Kita set konfigurasinya seperti pada gambar dibawah ini

```

index.php ...\home  mahasiswa.php  database.php X
application > config > database.php > ...
64 |         However, this may cause high memory
65 |         a lot of SQL queries ... disable th
66 |
67 | The $active_group variable lets you choose which
68 | make active. By default there is only one group
69 |
70 | The $query_builder variable lets you determine wh
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' => '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

	<table><tr><th>#</th><th>Nama</th><th>Jenis</th><th>Penyortiran</th><th>Atribut</th><th>Tak Ternilai</th><th>Bawaan</th><th>Komentar</th><th>Ekstra</th><th>Tindakan</th></tr><tr><td><input type="checkbox"/></td><td>1</td><td>id</td><td>int(11)</td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td>AUTO_INCREMENT</td><td> Ubah Hap</td></tr><tr><td><input type="checkbox"/></td><td>2</td><td>nama</td><td>varchar(100)</td><td>utf8mb4_general_ci</td><td>Tidak</td><td>Tidak ada</td><td></td><td></td><td> Ubah Hap</td></tr><tr><td><input type="checkbox"/></td><td>3</td><td>nim</td><td>varchar(50)</td><td>utf8mb4_general_ci</td><td>Tidak</td><td>Tidak ada</td><td></td><td></td><td> Ubah Hap</td></tr><tr><td><input type="checkbox"/></td><td>4</td><td>email</td><td>varchar(50)</td><td>utf8mb4_general_ci</td><td>Ya</td><td>NULL</td><td></td><td></td><td> Ubah Hap</td></tr><tr><td><input type="checkbox"/></td><td>5</td><td>jurusan</td><td>varchar(50)</td><td>utf8mb4_general_ci</td><td>Tidak</td><td>Tidak ada</td><td></td><td></td><td> Ubah Hap</td></tr></table>	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan	<input type="checkbox"/>	1	id	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hap	<input type="checkbox"/>	2	nama	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hap	<input type="checkbox"/>	3	nim	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hap	<input type="checkbox"/>	4	email	varchar(50)	utf8mb4_general_ci	Ya	NULL			Ubah Hap	<input type="checkbox"/>	5	jurusan	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hap
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan																																																				
<input type="checkbox"/>	1	id	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hap																																																				
<input type="checkbox"/>	2	nama	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hap																																																				
<input type="checkbox"/>	3	nim	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hap																																																				
<input type="checkbox"/>	4	email	varchar(50)	utf8mb4_general_ci	Ya	NULL			Ubah Hap																																																				
<input type="checkbox"/>	5	jurusan	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hap																																																				
9	<p>Tambahkan beberapa data ke dalam tabel mahasiswa seperti pada gambar (pada contoh terdapat 3 data, jumlah data lebih banyak tidak masalah)</p> <div><div>+ Options</div><table><tr><th></th><th>id</th><th>nama</th><th>nim</th><th>email</th><th>jurusan</th></tr><tr><td><input type="checkbox"/> Edit Copy Delete</td><td>1</td><td>dina</td><td>1002345100</td><td>dina@gmail.com</td><td>kimia</td></tr><tr><td><input type="checkbox"/> Edit Copy Delete</td><td>2</td><td>dika</td><td>1002345101</td><td>dika@gmail.com</td><td>informatika</td></tr><tr><td><input type="checkbox"/> Edit Copy Delete</td><td>3</td><td>soni</td><td>1002345102</td><td>soni@gmail.com</td><td>mesin</td></tr></table></div>		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																																				
	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	<p>Setelah kita membuat sebuah <i>database</i>, maka langkah berikutnya adalah membuat sebuah model untuk melakukan proses <i>database</i>. Di dalam model ini kita akan membuat sebuah <i>function</i> untuk melakukan perintah ambil data dari <i>database</i>. Untuk lebih jelasnya ikuti langkah-langkah berikut:</p> <p>a. Buatlah sebuah file model dengan nama mahasiswa_model pada folder application/models</p> <div><div>CODE-KEDUA</div><div><div>> controllers</div><div>> core</div><div>> helpers</div><div>> hooks</div><div>> language</div><div>> libraries</div><div>> logs</div><div>> models</div><div>> index.html</div><div> mahasiswa_model....</div><div>> third_party</div><div>> views</div><div> .htaccess</div><div>> index.html</div><div>> assets</div><div>> system</div><div>> user_guide</div><div> .editorconfig</div><div> .gitignore</div></div></div> <p>b. Tambahkan <i>function</i> getAllmahasiswa pada mahasiswa_model.php</p>																																																												

```

mahasiswa.php database.php mahasiswa_model.php x
application > models > mahasiswa_model.php > mahasiswa_model > g
4  defined('BASEPATH') OR exit('No direct script acces:
5
6  class mahasiswa_model extends CI_Controller {
7
8      public function getAllmahasiswa()
9      {
10         $query=$this->db->get('mahasiswa');
11
12         return $query->result_array();|
13     }
14
15 }
16
17 /* End of file mahasiswa_model.php */
18
19
20 ?>

```

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

```

application > models > mahasiswa_model.php > mahasiswa_model >
4  defined('BASEPATH') OR exit('No direct script acces:
5
6  class mahasiswa_model extends CI_Controller {
7
8      public function getAllmahasiswa()
9      {
10         // $query=$this->db->get('mahasiswa');
11
12         //return $query->result_array();
13
14         //atau bisa juga menggunakan code berikut
15
16         return $this->db->get('mahasiswa')->result_
17     }
18
19 }
20
21 /* End of file mahasiswa_model.php */
22
23
24 ?>

```

11

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

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

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**

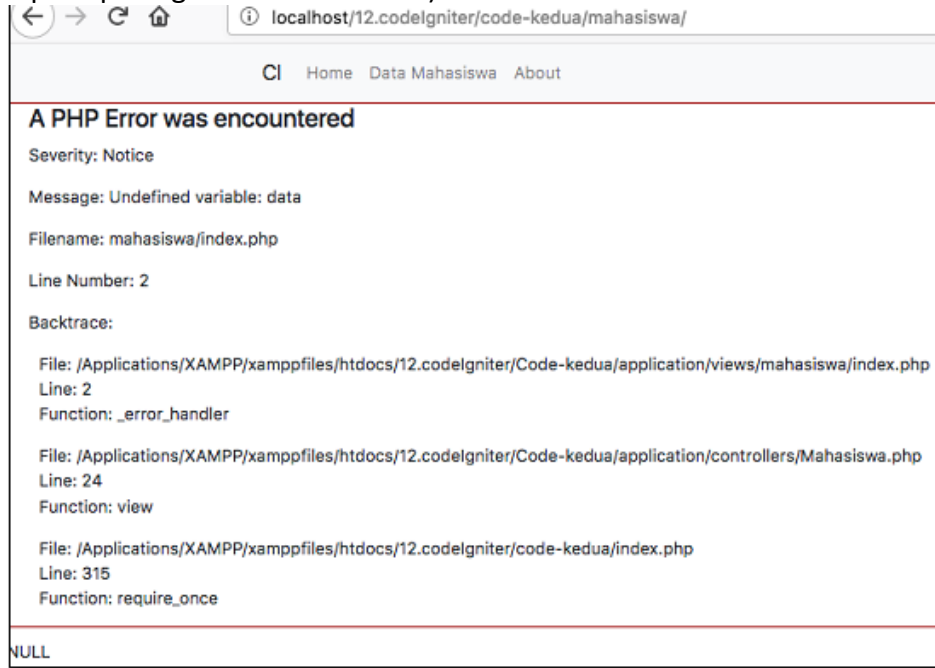

```

application > controllers > mahasiswa.php > mahasiswa > index
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access');
4
5  class mahasiswa extends CI_Controller {
6
7      //public function __construct()
8      //{
9          //digunakan untuk menjalankan fungsi constr
10         //    parent::__construct();
11         //    $this->load->database();
12         //}
13
14
15     public function index()
16     {
17         //modul untuk database
18         //$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         $this->load->model('mahasiswa_model');
25
26         $data['title']='List Mahasiswa';
27         $data['mahasiswa']=$this->mahasiswa_model->getAllMahasiswa();
28         $this->load->view('template/header',$data);
29         $this->load->view('mahasiswa/index',$data);
30         $this->load->view('template/footer');
31
32
33     }
34

```

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

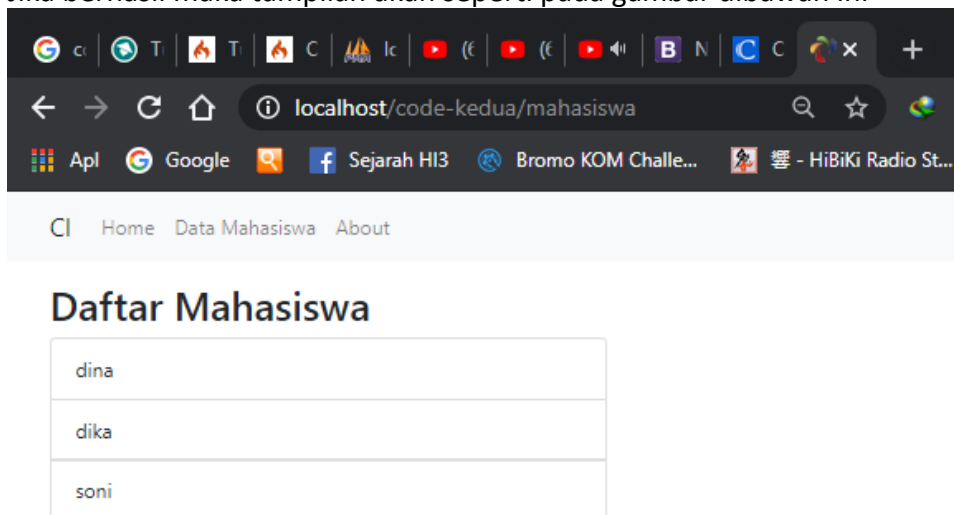


14

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

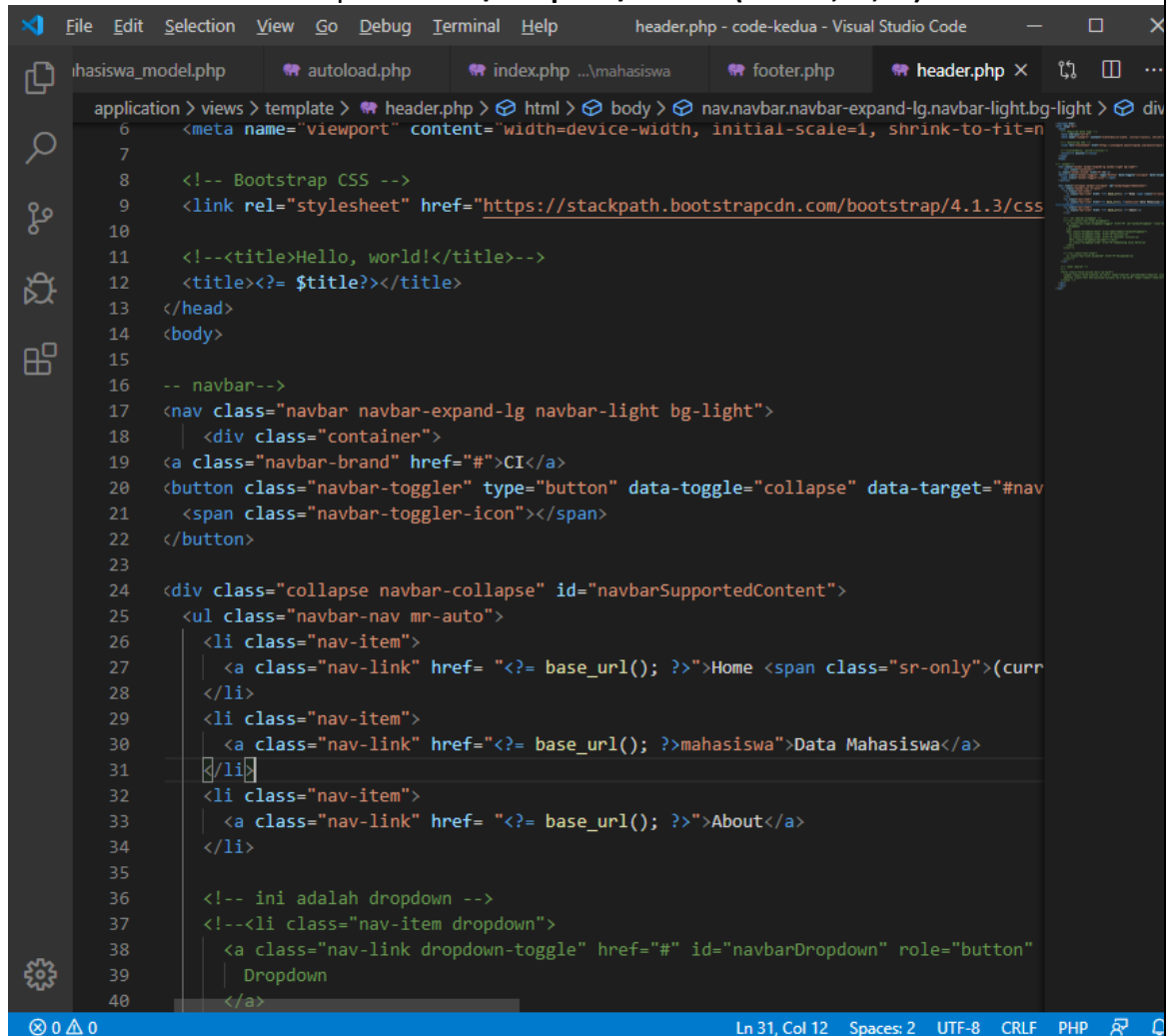
```
application > views > mahasiswa > index.php > ...
1 <div class="container">
2   <div class="row mt-3">
3     <div class="col-md-6">
4       <h2>Daftar Mahasiswa</h2>
5       <ul class="list-group">
6         <?php foreach($mahasiswa as $mhs):?>
7           <li class="list-group-item"><?php echo $mhs['nama'];>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

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)**



```

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 </head>
14 <body>
15
16 -- navbar-->
17 <nav class="navbar navbar-expand-lg navbar-light bg-light">
18   <div class="container">
19     <a class="navbar-brand" href="#">CI</a>
20     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar"
21       <span class="navbar-toggler-icon"></span>
22   </button>
23
24   <div class="collapse navbar-collapse" id="navbarSupportContent">
25     <ul class="navbar-nav mr-auto">
26       <li class="nav-item">
27         <a class="nav-link" href="<?= base_url(); ?>">Home <span class="sr-only">(current)
28       </li>
29       <li class="nav-item">
30         <a class="nav-link" href="<?= base_url(); ?>mahasiswa">Data Mahasiswa</a>
31       </li>
32       <li class="nav-item">
33         <a class="nav-link" href="<?= base_url(); ?>">About</a>
34       </li>
35
36       <!-- ini adalah dropdown -->
37       <li class="nav-item dropdown">
38         <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"
39           Dropdown
40       </a>

```

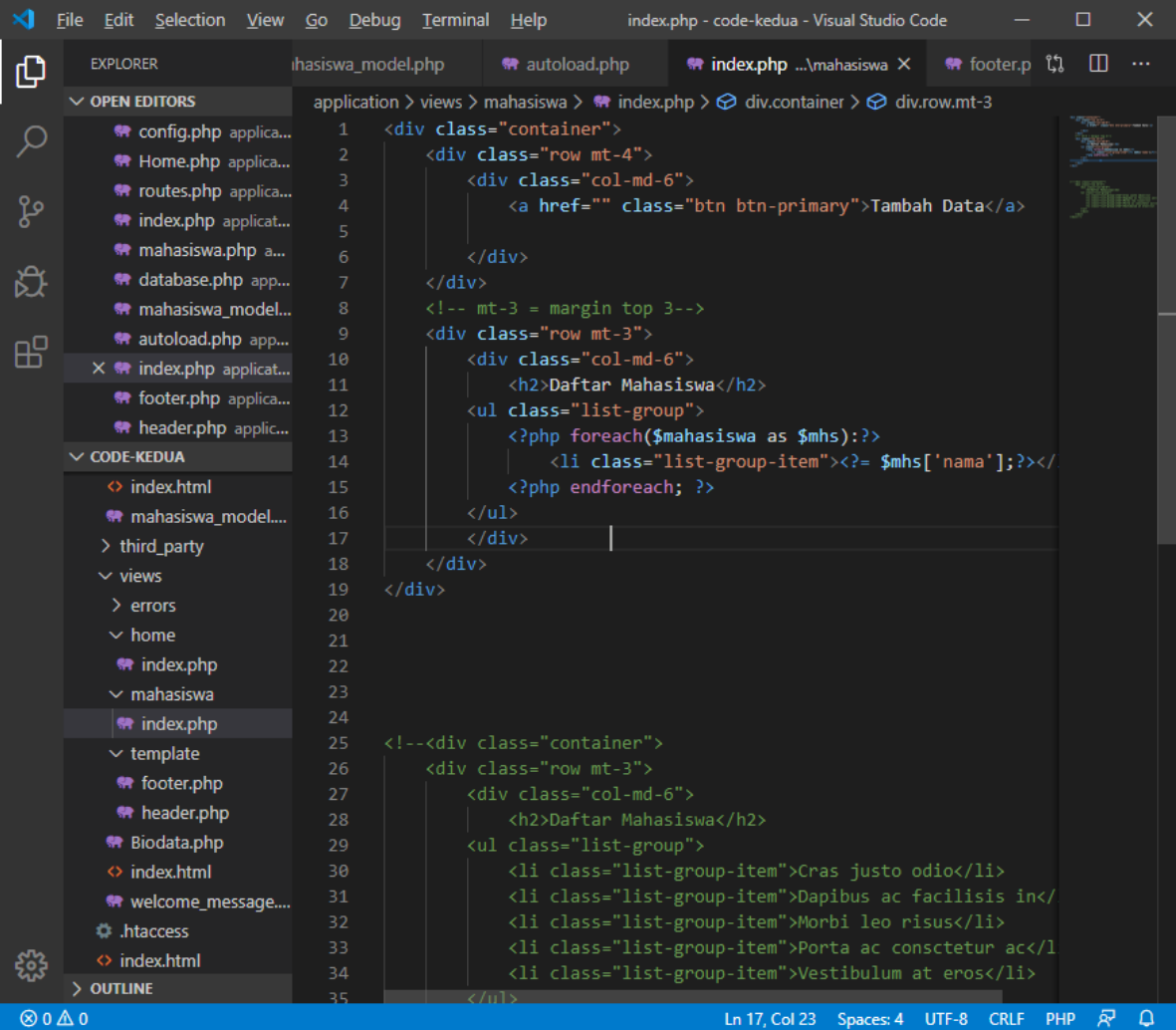
Keterangan :

- 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

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> 1 <div class="container"> 2 <div class="row mt-4"> 3 <div class="col-md-6"> 4 Tambah Data 5 </div> 6 </div> 7 </div> 8 <!-- mt-3 = 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> 20 21 22 23 24 25 <!--<div class="container"> 26 <div class="row mt-3"> 27 <div class="col-md-6"> 28 <h2>Daftar Mahasiswa</h2> 29 <ul class="list-group"> 30 <li class="list-group-item">Cras justo odio 31 <li class="list-group-item">Dapibus ac facilisis in 32 <li class="list-group-item">Morbi leo risus 33 <li class="list-group-item">Porta ac consectetur ac 34 <li class="list-group-item">Vestibulum at eros 35 36 </div> 37 </div> 38 </div> </pre>
2	<p>Dikarenakan mahasiswa_model ini nantinya akan digunakan oleh semua <i>function</i> di <i>controller</i> mahasiswa, maka “ <i>\$this->load->model('mahasiswa_model');</i> ” yang awalnya hanya berada pada <i>function index</i>, kita ubah ke <i>function construct</i> agar semua <i>function</i> dapat menggunakan mahasiswa_model.</p>

```

1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class mahasiswa extends CI_Controller {
6
7      //public function __construct()
8      //{
9          //digunakan untuk menjalankan fungsi construct pada class
10         // parent::__construct();
11         // $this->load->database();
12         //}
13
14
15     public function index()
16     {
17         //modul untuk database
18         //$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         $this->load->model('mahasiswa_model');
25
26         $data['title']='List Mahasiswa';
27         $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
28         $this->load->view('template/header',$data);
29         $this->load->view('mahasiswa/index',$data);
30         $this->load->view('template/footer');
31     }
32 }

```

- 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**)

```

32     public function tambah(){
33         $data['title']='Form Menambahkan Data Mahasiswa';
34         $this->load->view('template/header',$data);
35         $this->load->view('mahasiswa/tambah',$data);
36         $this->load->view('template/footer');
37     }
38
39
40 }

```

Keterangan :

- Pada line 35, `$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**

```

File Edit Selection View Go Debug Terminal Help tambah.php - code-kedua - Visual Studio Code
EXPLORER
tambah.php X database.php mahasiswa_model.php autoload.f
OPEN EDITORS
container > div.row.mt-3 > div.col-md-6 > div.card > div.card-body > form > div.form-group
Home.php applica... 4 class="card">
routes.php applica... 5 <div class="card-header">
index.php applicat... 6 form tambah data mahasiswa
mahasiswa.php a... 7 </div>
tambah.php appli... 8 <div class="card-body">
database.php app... 9 <form action="" method="POST">
mahasiswa_model... 10 <div class="form-group">
autoload.php app... 11 <label for="nama">Nama</label>
index.php applicat... 12 <input type="text" class="form-control" id="nama">
footer.php applica... 13 </div>
header.php applic... 14 <div class="form-group">
mahasiswa_model... 15 <label for="nim">Nim</label>
third_party 16 <input type="number" class="form-control" id="nim">
views 17 </div>
errors 18 <div class="form-group">
home 19 <label for="email">Email</label>
index.php 20 <input type="email" class="form-control" id="email">
mahasiswa 21 </div>
index.php 22 <div class="form-group">
tambah.php 23 <label for="jurusan">Jurusan</label>
template 24 <select class="form-control" id="jurusan" name="jurusan">
footer.php 25 <option selected>-----</option>
header.php 26 <option value="kimia">Kimia</option>
Biodata.php 27 <option value="informatika">Informatika</option>
index.html 28 <option value="mesin">Mesin</option>
welcome_message.... 29 </select>
.htaccess 30 </div>
index.html 31 <button type="submit" class="btn btn-primary">Submit</but
OUTLINE 32 </form>
33 </div>
34
35
36
37
Ln 16, Col 57 Spaces: 4 UTF-8 CRLF PHP

```

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

5

Maka akan dihasilkan halaman seperti di bawah ini.

Course: Komputas x (41) Karakai Jc x Form Menambah x

localhost/code-kedua/mahasiswa/tambah

CI Home Data Mahasiswa About

form tambah data mahasiswa

Nama

Nim

Email

Jurusan

Submit

6

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

```

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

```

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*

```

31 }
32 public function tambah() {
33     $data['title'] = 'Form Menambahkan Data Mahasiswa';
34     if($this->$this->form_validation->run() == FALSE){
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

```

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**

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

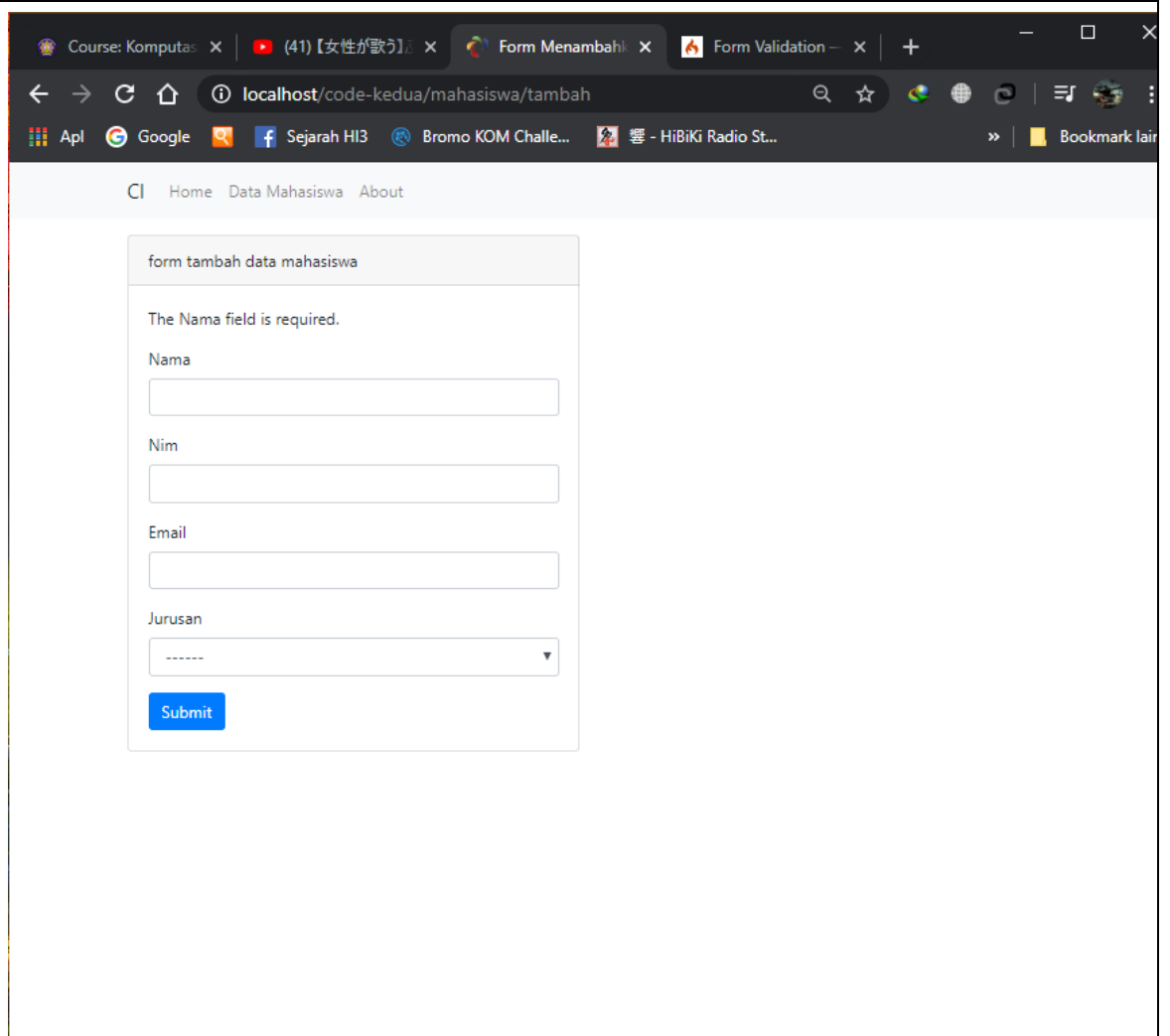
e. Setelah kita tambahkan **atribut name** maka tambahkan **rule** pada **controller/mahasiswa** seperti pada gambar dibawah ini (line 35-46)

```
public function tambah(){
    $data['title']='Form Menambahkan Data Mahasiswa';
    $this->load->library('form_validation');

    //$this->$this->form_validation->set_rules('fieldname', '
    $this->form_validation->set_rules('nama','Nama','required
    if($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)

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



h. 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">
    <!-- https://getbootstrap.com/docs/4.1/components/forms/ -->
    <div class="form-group">
      <label for="nama">Nama</label>
      <input type="text" class="form-control" id="nama" name="nama">
    </div>
```

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

i. Akan dihasilkan seperti gambar di bawah ini

Course: Komputas x (41) 【女性が歌う】 x Form Menambah x Form Validation x +

localhost/code-kedua/mahasiswa/tambah

Apl Google Sejarah HI3 Bromo KOM Challe... HiBiKi Radio St... Bookmark lair

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**.

```

30 public function tambah() {
31
32     $data['title'] = 'Form Menambahkan Data Mahasiswa';
33
34     // $this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_length[5]|max_length[12]');
35     $this->form_validation->set_rules('nama', 'Nama', 'required');
36     $this->form_validation->set_rules('nim', 'Nim', 'required');
37     $this->form_validation->set_rules('email', 'Email', 'required');
38
39     if ($this->form_validation->run() == FALSE) {
40         # code...
41         $this->load->view('template/header', $data);
42         $this->load->view('mahasiswa/tambah', $data);
43         $this->load->view('template/footer');
44     } else {
45         # code...
46         echo "data berhasil ditambah";
47     }
48 }
49
50
51
52 /* End of file Controllername.php */
53 ?>

```

- k. Hasilnya seperti berikut ini.

form tambah data mahasiswa

The Nama field is required.
The Nim field is required.
The Email field is required.

Nama

Nim

Email

Jurusan

Submit

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

```

CODE-KEDUA
40 >form_validation->set_rules('nama','Nama','required');
41 >form_validation->set_rules('nim','Nim','required|numeric');
42 >form_validation->set_rules('email','Email','required|valid_email
43

```

- m. Hasilnya sebagai berikut.

Nama

Nim

Email

Sertakan '@' pada alamat email. 'aaaa' tidak memiliki '@'.

Jurusan

Submit

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

```
40 $this->form_validation->set_rules('nama','Nama','required');  
41 $this->form_validation->set_rules(['nim','Nim','required|numeric']);  
42 $this->form_validation->set_rules('email','Email','required');
```

- o. Akan dihasilkan alert seperti berikut ini.

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

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

Langkah	Keterangan
1	Buka controller/mahasiswa dan tambahkan <i>code</i> berikut pada function tambah agar pada saat form_validation bernilai true , maka akan ditambahkan data dengan nama modelnya -> model_mahasiswa dengan nama <i>function</i> tambahdatamhs() dan jika sudah ditambah ke database maka akan <i>redirect</i> ke <i>controller</i> mahasiswa (<i>line</i> 46 - 47)

```

File Edit Selection View Go Debug Terminal Help mahasiswa.php - code-kedua - Visual Studio Code
EXPLORER
OPEN EDITORS
  config.php applica...
  Home.php applica...
  routes.php applica...
  index.php applicat...
  X mahasiswa.php a...
  tambah.php appli...
  database.php app...
  mahasiswa_model...
  autoload.php app...
  index.php applicat...
  footer.php applica...
CODE-KEDUA
  mimes.php
  profiler.php
  routes.php
  smileys.php
  user_agents.php
  controllers
    Home.php
    index.html
  mahasiswa.php
  Welcome.php
  core
  helpers
  hooks
  language
  libraries
  logs
  models
  OUTLINE
mahasiswa.php X tambah.php database.php mahasiswa_model
application > controllers > mahasiswa.php > mahasiswa > tambah
32 $this->load->view('template/footer');
33
34 ic function tambah(){
35 $data['title']='Form Menambahkan Data Mahasiswa';
36 $this->load->library('form_validation');
37
38
39 //$this->$this->form_validation->set_rules('fieldname', 'fieldlab
40
41 $this->form_validation->set_rules('nama','Nama','required');
42 $this->form_validation->set_rules('nim','Nim','required|numeric')
43 $this->form_validation->set_rules('email','Email','required|valid
44
45 if($this->form_validation->run() == FALSE){
46 $this->load->view('template/header',$data);
47 $this->load->view('mahasiswa/tambah',$data);
48 $this->load->view('template/footer');
49 } else {
50 $this->load->model(['mahasiswa_model']);
51 $this->mahasiswa_model->tambahdatamhs();
52 redirect('mahasiswa','refresh');
53 }
54
55
56
57
58
59 f file mahasiswa.php */
60
61
62
Ln 50, Col 31 Spaces: 4 UTF-8 CRLF PHP

```

- 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=insertrt#inserting-data
 - Tambahkan *code* seperti pada gambar dibawah ini

```

33  /* End of file mahasiswa_model.php */
34
35  defined('BASEPATH') OR exit('No direct script access allowed');
36
37  class Mahasiswa_model extends CI_Model {
38
39      public function getAllMahasiswa()
40      {
41
42          return $this->db->get('mahasiswa')->result_array();
43      }
44
45      function tambahDataMhs(){
46          $data=[
47              "nama"=> $this->input->post('nama',true),
48              "nim"=> $this->input->post('nim',true),
49              "email"=> $this->input->post('email',true),
50              "jurusan"=> $this->input->post('jurusan',true)
51          ];
52          $this->db->insert('mahasiswa', $data);
53      }
54
55      /* End of file mahasiswa_model.php */
56      /* Location: ./application/models/mahasiswa_model.php */
57
58      ?>

```

Keterangan:

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

```

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

```

```

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

```

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

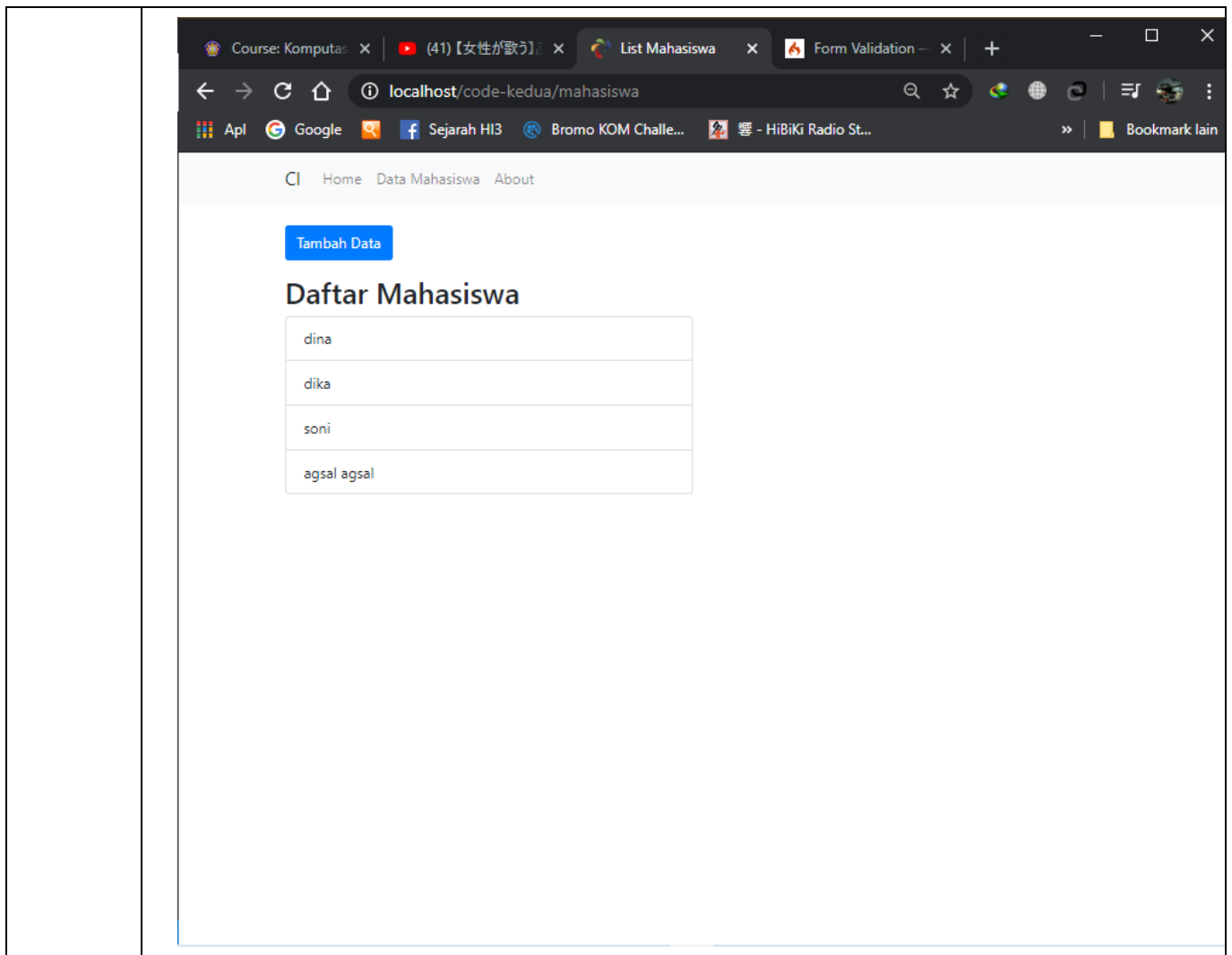
4

Langkah berikutnya kita coba untuk menambahkan data.

Jika berhasil maka data akan tampil di halaman **index.php**

a. `redirect('mahasiswa','refresh');`(line 47 gambar *controller* mahasiswa)

b. Hasil



-- Selamat Mengerjakan --