

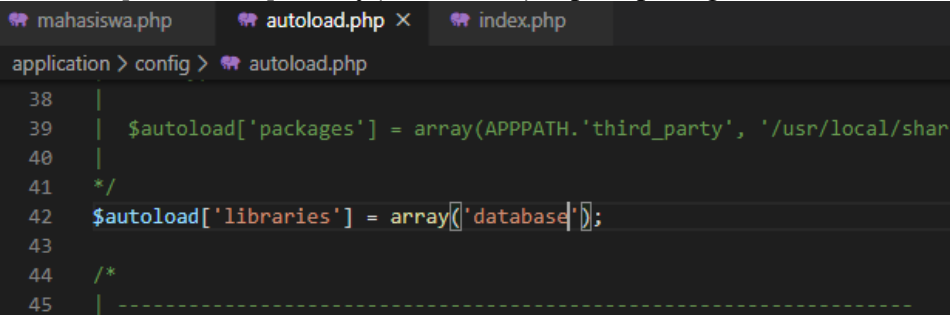


NAMA : QORINDA YULVARISMA
KELAS : TI-2A
NIM : 1841720084

LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT

Praktikum Bagian 1. Menampilkan data dari database menggunakan codeIgniter

Langkah	Keterangan
1	Buka kembali folder CodeIgniter yang telah dibuat pada Jobsheet sebelumnya
2	Buka controller mahasiswa dan tambahkan code berikut (line 9) <pre>mahasiswa.php x index.php application > controllers > mahasiswa.php 1 <?php 2 3 defined('BASEPATH') OR exit('No direct script access allowed'); 4 5 class mahasiswa extends CI_Controller { 6 7 public function index() 8 { 9 //modul untuk load database 10 \$this->load->database(); 11 \$data['title']='List Mahasiswa'; 12 \$this->load->view('template/header',\$data); 13 \$this->load->view('mahasiswa/index'); 14 \$this->load->view('template/footer'); 15 } 16 17 } 18 19 /* End of file Controllername.php */ 20 21 ></pre>
3	Tambahkan code berikut agar database dapat digunakan banyak function (line 7-12)

	 <pre> mahasiswa.php x index.php application > controllers > mahasiswa.php 1 <?php 2 3 defined('BASEPATH') OR exit('No direct script access allowed'); 4 5 class mahasiswa extends CI_Controller { 6 7 //fungsi yang akan dijalankan saat classnya diinstansiasi 8 public function _construct() 9 { 10 //digunakan untuk menjalankan fungsi construct pada class parent 11 parent::_construct(); 12 \$this->load->database(); 13 } 14 15 public function index() 16 { 17 //modul untuk load database 18 // \$this->load->database(); 19 \$data['title']='List Mahasiswa'; 20 \$this->load->view('template/header',\$data); 21 \$this->load->view('mahasiswa/index'); 22 \$this->load->view('template/footer'); 23 } 24 25 } 26 27 /* End of file Controllername.php */ 28 29 ?> </pre>
4	<p>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</p>  <pre> mahasiswa.php x autoload.php x index.php application > config > autoload.php 38 39 \$autoload['packages'] = array(APPPATH.'third_party', '/usr/local/shar 40 41 */ 42 \$autoload['libraries'] = array('database'); 43 44 /* 45 ----- </pre>
5	<p>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</p>

	<pre> mahasiswa.php × autoload.php index.php application > controllers > mahasiswa.php 1 <?php 2 3 defined('BASEPATH') OR exit('No direct script access allowed'); 4 5 class mahasiswa extends CI_Controller { 6 7 //fungsi yang akan dijalankan saat classnya diinstansiasi 8 // public function _construct() 9 // { 10 // //digunakan untuk menjalankan fungsi construct pada class parent 11 // parent::_construct(); 12 // \$this->load->database(); 13 // } 14 </pre>
6	<p>Setelah itu kita akan mengatur konfigurasi database. Untuk mengatur konfigurasi database maka buka folder application/config/database.php</p> <pre> mahasiswa.php database.php × autoload.php index.php application > config > database.php 72 */ 73 \$active_group = 'default'; 74 \$query_builder = TRUE; 75 76 \$db['default'] = array(77 'dsn' => '', 78 'hostname' => 'localhost', 79 'username' => '', 80 'password' => '', 81 'database' => '', 82 'dbdriver' => 'mysqli', 83 'dbprefix' => '', 84 'pconnect' => FALSE, 85 'db_debug' => (ENVIRONMENT !== 'production'), 86 'cache_on' => FALSE, 87 'cachedir' => '', 88 'char_set' => 'utf8', 89 'dbcollat' => 'utf8_general_ci', 90 'swap_pre' => '', 91 'encrypt' => FALSE, 92 'compress' => FALSE, 93 'stricton' => FALSE, 94 'failover' => array(), 95 'save_queries' => TRUE 96); 97 </pre>
7	<p>Kita set konfigurasinya seperti pada gambar dibawah ini</p>

```

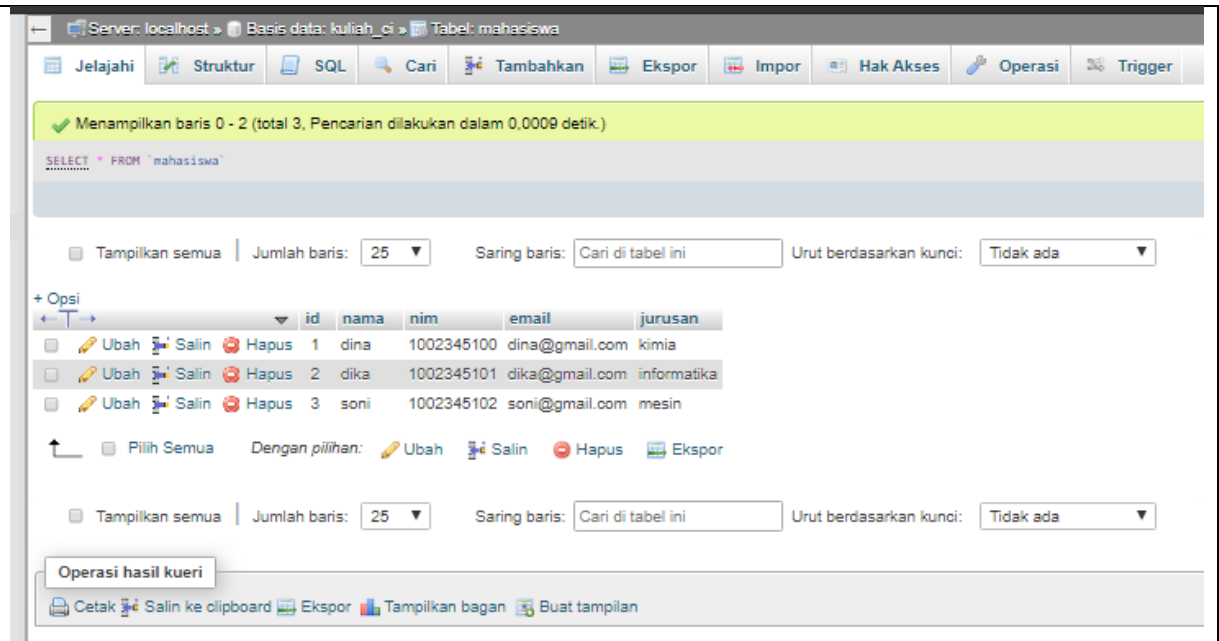
database.php x mahasiswa_model.php autoload.php
application > config > database.php
72 */
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => 'mysql',
81     'database' => '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

```

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)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	nim	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	email	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	jurusan	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

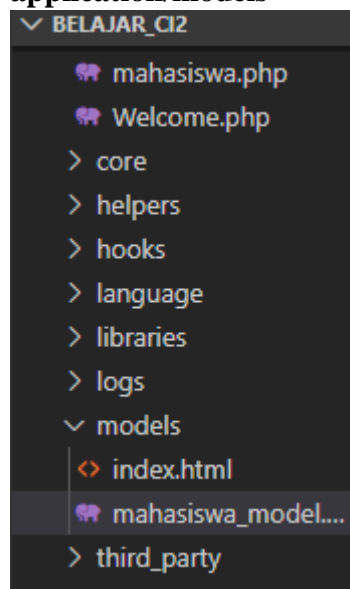
9 Tambahkan beberapa data ke dalam tabel mahasiswa seperti pada gambar (pada contoh terdapat 3 data, jumlah data lebih banyak tidak masalah)



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**



- Tambahkan function getAllmahasiswa pada mahasiswa_model.php

```

mahasiswa.php database.php mahasiswa_model.php x autoload.php
application > models > mahasiswa_model.php
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class mahasiswa_model extends CI_Model {
6
7      public function getAllmahasiswa(){
8          // $query digunakan untuk menampung isi dari dari tabel mah
9          $query=$this->db->get('mahasiswa');
10
11          // untuk menampilkan isi dari query
12          return $query->result_array();
13      }
14  }
15
16
17 }
18
19 /* End of file ModelName.php */
20 ?>

```

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

```

mahasiswa.php database.php mahasiswa_model.php x autoload.php
application > models > mahasiswa_model.php
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class mahasiswa_model extends CI_Model {
6
7      public function getAllmahasiswa(){
8          // $query digunakan untuk menampung isi dari dari tabel mah
9          // $query=$this->db->get('mahasiswa');
10
11          // untuk menampilkan isi dari query
12          // return $query->result_array();
13
14          //atau bisa juga menggunakan code berikut
15          return $this->db->get(['mahasiswa']->result_array();
16      }
17
18
19 }
20
21 /* End of file ModelName.php */
22 ?>

```

11

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

```

mahasiswa.php × mahasiswa_model.php index.php
application > controllers > mahasiswa.php
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 }
29
30 /* End of file Controllername.php */
31
32 ?>

```

12

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

```

database.php mahasiswa_model.php autoload.php index.php ×
application > views > mahasiswa > index.php
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>

```

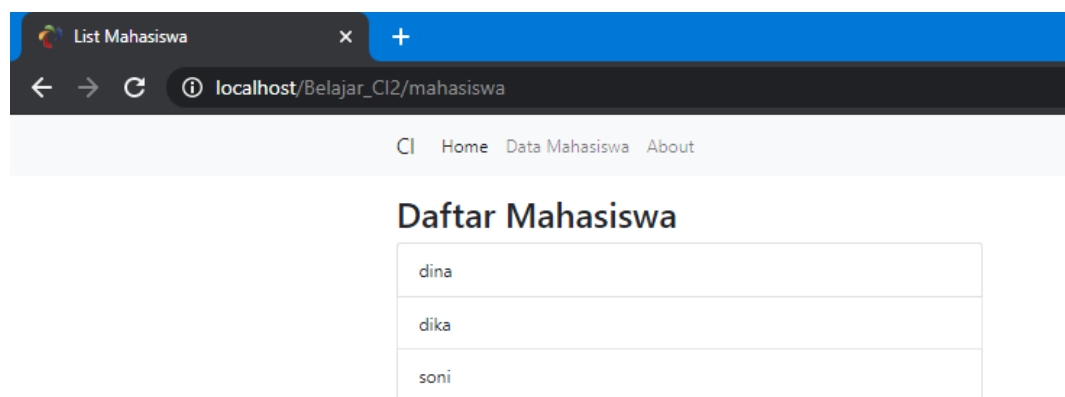
	<div><div>localhost/Belajar_CI2/mahasiswa/</div><div>An uncaught Exception was encountered Type: RuntimeException Message: Class Mahasiswa already exists and doesn't extend CI_Model Filename: C:\Program Files\Ampps\www\Belajar_CI2\system\core\Loader.php Line Number: 353 Backtrace: File: C:\Program Files\Ampps\www\Belajar_CI2\application\controllers\mahasiswa.php Line: 17 Function: model File: C:\Program Files\Ampps\www\Belajar_CI2\index.php Line: 315 Function: require_once</div></div>
13	<div>Jika berhasil maka data akan tampil seperti pada gambar berikut</div> <div><div>List Mahasiswa</div><div>localhost/Belajar_CI2/mahasiswa</div><div>CI Home Data Mahasiswa About</div><div>C:\Program Files\Ampps\www\Belajar_CI2\application\views\mahasiswa\index.php:2: array (size=3) 0 => array (size=5) 'id' => string '1' (Length=1) 'nama' => string 'dina' (Length=4) 'nim' => string '1002345100' (Length=10) 'email' => string 'dina@gmail.com' (Length=14) 'jurusan' => string 'kimia' (Length=5) 1 => array (size=5) 'id' => string '2' (Length=1) 'nama' => string 'dika' (Length=4) 'nim' => string '1002345101' (Length=10) 'email' => string 'dika@gmail.com' (Length=14) 'jurusan' => string 'informatika' (Length=11) 2 => array (size=5) 'id' => string '3' (Length=1) 'nama' => string 'soni' (Length=4) 'nim' => string '1002345102' (Length=10) 'email' => string 'soni@gmail.com' (Length=14) 'jurusan' => string 'mesin' (Length=5)</div><div>Daftar Mahasiswa</div><div><div>Cras justo odio</div><div>Dapibus ac facilisis in</div><div>Morbi leo risus</div><div>Porta ac consectetur ac</div><div>Vestibulum at eros</div></div></div>
14	<div>Agar semua isi dari tabel dapat tampil di halaman index mahasiswa maka kita harus lakukan proses looping. Gunakan code berikut pada views/mahasiswa/index</div>


```

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

```

15 Jika berhasil maka tampilan akan seperti pada gambar dibawah ini



16 Jika kita klik Home, halaman Home belum dapat tampil. Maka dari itu kita akan set agar semua dapat saling terhubung. Untuk itu kita membutuhkan yang namanya base_url. untuk lebih jelasnya ikuti langkah-langkah berikut:

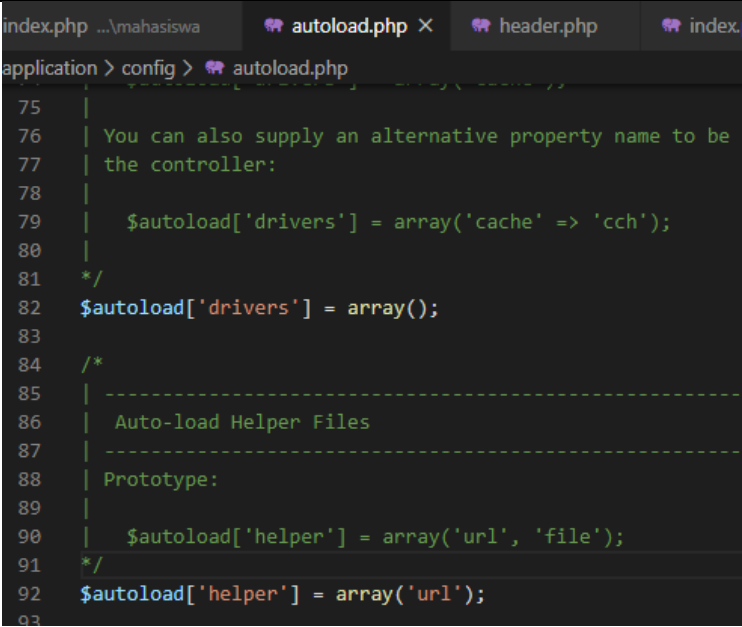
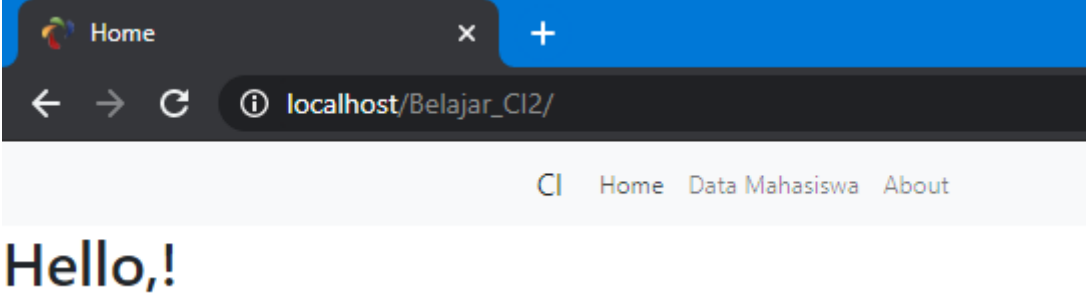
a. Tambahkan code berikut pada views/template/header (line 26,27,28)

```

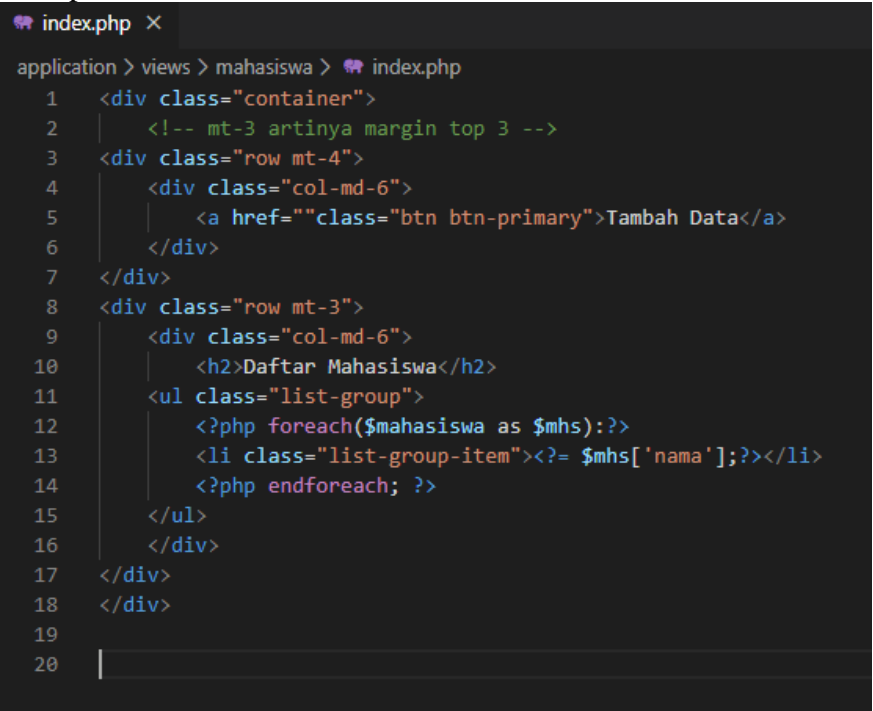
application > views > template > header.php
12  <title><?php echo $title;?></title>
13  </head>
14  <body>
15
16
17  <!-- Navbar -->
18  <nav class="navbar navbar-expand-lg navbar-light bg-light">
19      <div class="container">
20          <a class="navbar-brand" href="#">CI</a>
21          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar"
22              <span class="navbar-toggler-icon"></span>
23          </button>
24          <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
25              <div class="navbar-nav">
26                  <a class="nav-item nav-link" href="<?php echo base_url();?>">Home <span class="sr-only">(c
27                  <a class="nav-item nav-link" href="<?php echo base_url();?>mahasiswa">Data Mahasiswa</a>
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>

```

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

	 <p>c. Jika berhasil maka semua halaman dapat ditampilkan pada saat user melakukan event click.</p> 
--	--

Praktikum Bagian 2. Membuat Form Input dan Menambahkan Form Validation

Langkah	Keterangan
1	<p>Buatlah sebuah button tambah data pada views/mahasiswa/index.php (line 3-7). Tambahkan code pada line 3-7 untuk membuat sebuah button</p> 

2	<p>Dikarenakan mahasiswa_model ini nantinya akan digunakan oleh semua function di controller mahasiswa, maka “ \$this->load->model('mahasiswa_model'); “ yang awalnya hanya berada pada function index, kita ubah ke function construct agar semua function dapat menggunakan mahasiswa_model.</p> <pre> index.php mahasiswa.php X application > controllers > mahasiswa.php 1 <?php 2 3 defined('BASEPATH') OR exit('No direct script access allowed'); 4 5 class mahasiswa extends CI_Controller { 6 7 //fungsi yang akan dijalankan saat classnya diinstansiasi 8 //public function __construct() 9 //{ 10 //digunakan untuk menjalankan fungsi construct pada class parent 11 // parent::__construct(); 12 // \$this->load->database(); 13 // } 14 public function __construct() 15 { 16 //digunakan untuk menjalankan fungsi construct pada class parent(ci 17 parent::__construct(); 18 \$this->load->model('mahasiswa_model'); 19 } 20 21 22 public function index() 23 { 24 25 //modul untuk load database 26 // \$this->load->database(); 27 \$data['title']='List Mahasiswa'; 28 \$data['mahasiswa']=\$this->mahasiswa_model->getAllmahasiswa(); 29 \$this->load->view('template/header',\$data); 30 \$this->load->view('mahasiswa/index',\$data); 31 \$this->load->view('template/footer'); 32 } </pre>
3	<p>Setelah itu tambahkan function tambah pada controller mahasiswa dan tambahkan code berikut.</p> <p>(Tips : jika menggunakan vsCode untuk menduplikat 1 baris dapat menggunakan alt+shift+panah_bawah)</p> <pre> 34 public function tambah(){ 35 \$data['title']='Form Menambahkan Data Mahasiswa'; 36 \$this->load->view('template/header',\$data); 37 \$this->load->view('mahasiswa/tambah',\$data); 38 \$this->load->view('template/footer'); 39 } 40 41 } 42 43 /* End of file Controllername.php */ 44 45 ?> </pre>
4	<p>Setelah kita membuat function tambah pada controller mahasiswa, selanjutnya buatlah form tambah pada views/mahasiswa/tambah.php</p>

```

index.php mahasiswa.php tambah.php X
application > views > mahasiswa > tambah.php
1 s="container">
2 class="row mt-3">
3 div class="col-md-6">
4 <!-- https://getbootstrap.com/docs/4.4/components/card/ -->
5 <div class="card">
6   <div class="card-header">
7     Form Tambah Data Mahasiswa
8   </div>
9   <div class="card-body">
10    <?= validation_errors() ?>
11    <form action="" method="post">
12      <!-- https://getbootstrap.com/docs/4.4/components/forms/ -->
13      <div class="form-group">
14        <label for="nama">Nama</label>
15        <input type="text" class="form-control" id="nama" name="nama">
16      </div>
17      <div class="form-group">
18        <label for="nim">Nim</label>
19        <input type="number" class="form-control" id="nim" name="nim">
20      </div>
21      <div class="form-group">
22        <label for="email">Email</label>
23        <input type="email" class="form-control" id="email" name="email">
24      </div>
25      <div class="form-group">
26        <label for="jurusan">Jurusan</label>
27        <select class="form-control" id="jurusan" name="jurusan">
28          <option selected>=====</option>
29          <option value="informatika">Informatika</option>
30          <option value="kimia">Kimia</option>
31          <option value="mesin">Mesin</option>
32        </select>

```

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' with the following fields: 'Nama' (text input), 'Nim' (text input), 'Email' (text input), and 'Jurusan' (dropdown menu with a downward arrow). A blue 'Submit' button is located at the bottom right of the form.

6

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

```

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

```

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

```

index.php x      mahasiswa.php x
application > controllers > mahasiswa.php
22 public function index()
23 {
24
25     //modul untuk load database
26     // $this->load->database();
27     $data['title']='List Mahasiswa';
28     $data['mahasiswa']=$this->mahasiswa_model->getAllmahasiswa();
29     $this->load->view('template/header',$data);
30     $this->load->view('mahasiswa/index',$data);
31     $this->load->view('template/footer');
32 }
33
34 public function tambah(){
35     $data['title']='Form Menambahkan Data Mahasiswa';
36
37     if($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 }
49
50
51 /* End of file Controllername.php */
52
53 ?>

```

- c. Setelah kita tambahkan form validation, maka langkah berikutnya menambahkan rulanya

- 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**

```

12 <div class="form-group">
13     <label for="nama">Nama</label>
14     <input type="text" class="form-control" id="nama" name="nama">
15 </div>
16 <div class="form-group">
17     <label for="nim">Nim</label>
18     <input type="number" class="form-control" id="nim" name="nim">
19 </div>
20 <div class="form-group">
21     <label for="email">Email</label>
22     <input type="email" class="form-control" id="email" name="email">
23 </div>
24 <div class="form-group">
25     <label for="jurusan">Jurusan</label>
26     <select class="form-control" id="jurusan" name="jurusan">
27         <option selected>-----</option>
28         <option value="informatika">Informatika</option>
29         <option value="kimia">Kimia</option>
30         <option value="mesin">Mesin</option>

```

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

```

34 public function tambah(){
35     $data['title']='Form Menambahkan Data Mahasiswa';
36
37     $this->form_validation->set_rules('nama','Nama','required');
38
39     if($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

```

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

```

9 <div class="card-body">
10     <div class="validation_errors">
11         <div class="form-group">
12             <label for="nama">Nama</label>
13             <input type="text" class="form-control" id="nama" name="nama">
14         </div>

```

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

Form Menambahkan Data Mahasiswa

localhost/Belajar_CI2/mahasiswa/tambah

CI Home Data Mahasiswa About

Form Tambah Data Mahasiswa

Nama

Nim

Email

Jurusan

Submit

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/>

```

9      <div class="card-body">
10         <?php if (validation_errors()): ?>
11             <div class="alert alert-danger" role="alert">
12                 <?= validation_errors() ?>
13             </div>
14         <?php endif ?>
15         <form action="" method="post">
16             <!-- https://getbootstrap.com/docs/4.4/components/forms/ -->
17             <div class="form-group">
18                 <label for="nama">Nama</label>
19                 <input type="text" class="form-control" id="nama" name="nama">
20             </div>

```

- Tambahkan tanda ; pada akhir **validation_errors()** sehingga menjadi seperti berikut <?= validation_errors(); ?>
- i. Akan dihasilkan seperti gambar di bawah ini

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

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

```

34     public function tambah(){
35         $data['title']='Form Menambahkan Data Mahasiswa';
36
37         $this->form_validation->set_rules('nama','Nama','required');
38         $this->form_validation->set_rules('nim','Nim','required');
39         $this->form_validation->set_rules('email','Email','required');
40
41
42
43         if($this->form_validation->run() == FALSE){
44             #code
45             $this->load->view('template/header',$data);
46             $this->load->view('mahasiswa/tambah',$data);
47             $this->load->view('template/footer');
48         } else{
49             #code
50             echo "data berhasil ditambah";
51         }
52     }
53 }
54
55 }
56
57 /* End of file Controllername.php */
58
59 ?>

```

- 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)**

```

37 $this->form_validation->set_rules('nama','Nama','required');
38 $this->form_validation->set_rules('nim','Nim','required');
39 $this->form_validation->set_rules('email','Email','required|valid_email');
40

```

- m. Hasilnya sebagai berikut.

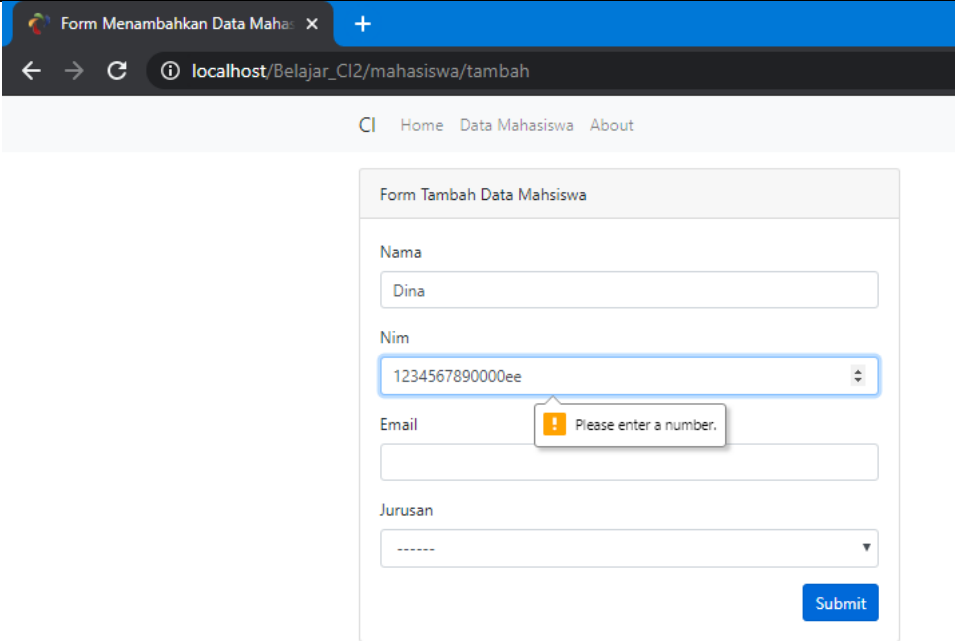
- n. Selain itu kita juga dapat memberikan rules harus diisi dengan angka menggunakan rules **numeric**

```

37 $this->form_validation->set_rules('nama','Nama','required');
38 $this->form_validation->set_rules('nim','Nim','required|numeric');
39 $this->form_validation->set_rules('email','Email','required|valid_email');
40

```

- o. Akan dihasilkan alert seperti berikut ini

	
--	---

Praktikum Bagian 3. Menambahkan Data ke Database / Insert Data

Langkah	Keterangan
1	<p>Buka controller/mahasiswa dan tambahkan code berikut pada function tambah agar pada saat form_validation bernilai true, maka akan ditambahkan data dengan nama modelnya > model_mahasiswa dengan nama function tambahdatamhs() dan jika sudah ditambah ke database maka akan redirect ke controller mahasiswa (line 46 - 47)</p> <pre> 34 public function tambah(){ 35 \$data['title']='Form Menambahkan Data Mahasiswa'; 36 37 \$this->form_validation->set_rules('nama','Nama','required'); 38 \$this->form_validation->set_rules('nim','Nim','required numeric'); 39 \$this->form_validation->set_rules('email','Email','required valid_email'); 40 41 42 43 if(\$this->form_validation->run() == FALSE){ 44 #code 45 \$this->load->view('template/header',\$data); 46 \$this->load->view('mahasiswa/tambah',\$data); 47 \$this->load->view('template/footer'); 48 } else{ 49 #code 50 \$this->mahasiswa_model->tambahdatamhs(); 51 redirect('mahasiswa','refresh'); 52 // echo "data berhasil ditambah"; 53 } 54 } 55 </pre>
2	<p>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()</p>
3	<p>Buatlah sebuah function 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 code seperti pada gambar dibawah ini <p>Keterangan:</p>

```

20 public function tambahdatamhs(){
21     $data=[
22         'nama' => $this->input->post('nama', true), // ini sama dengan htmlspecialchars
23         'nim' => $this->input->post('nim', true),
24         'email' => $this->input->post('email', true),
25         'jurusan' => $this->input->post('jurusan', true)
26     ];
27     $this->db->insert('mahasiswa',$data);
28 }
29

```

- Line 22-27 adalah sebuah array dimana nantinya isi array adalah data-data yang akan dikirim (POST) ke database
- Line 23 “nama” => \$this->input->post(‘nama’,true)
 - nama->alias
 - Parameter nama pada post adalah name pada html seperti pada gambar dibawah ini
 - Parameter true digunakan untuk mengaktifkan htmlspecialchars
- Line 29 adalah proses insert data ke database dengan **2 parameter** yaitu **nama tabelnya** dan **objectnya** (dalam hal ini data yang akan dimasukkan ke database)

4

Langkah berikutnya kita coba untuk tambahkan data. Jika berhasil maka data akan tampil di halaman **index.php**

- a. redirect('mahasiswa',refresh'); (line 47 gambar controller mahasiswa)
- b. Hasil

data berhasil ditambah

List Mahasiswa

localhost/Belajar_CI2/mahasiswa/

CI Home Data Mahasiswa About

Tambah Data

Daftar Mahasiswa

dina
dika
soni
Qorinda