

PEMROGRAMAN WEB LANJUT

JOBSHEET 2

Oleh:

**MUHAMMAD HIFZHAN SILMI
TI-2A
NIM. 1841720159**



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
FEBRUARI 2020**

PRAKTIKUM 1

1. Buka kembali folder CodeIgniter yang telah dibuat pada Jobsheet sebelumnya
2. Buka *controller* mahasiswa dan tambahkan *code* berikut (line 9)

```
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              $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     /* End of file Controllername.php */
18
19
20  ?>
```

Keterangan :

Pada line 9 menjelaskan bawa *function* index ini akan melakukan *load* database. *Load* database ini hanya dilakukan di *function* index saja jika ada *function* lain maka kita harus melakukan *load* database lagi. Sehingga hal ini dirasa tidak praktis maka untuk itu kita harus tambahkan *code* agar *load* database ini dapat digunakan banyak *function*.

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

```
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 __construct()
8          {
9              parent::__construct();
10             $this->load->database();
11
12             public function index()
13             {
14                 // $this->load->database();
15                 $data['title']='List Mahasiswa';
16                 $this->load->view('template/header',$data);
17                 $this->load->view('mahasiswa/index');
18                 $this->load->view('template/footer');
19             }
20         }
21
22     /* End of file Controllername.php */
23
24  ?>
```

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 memload 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

```
/*
| -----
|   Auto-load Libraries
| -----
| These are the classes located in system/libraries/ or your
| application/libraries/ directory, with the addition of the
| 'database' library, which is somewhat of a special case.
|
| Prototype:
|
|   $autoload['libraries'] = array('database', 'email', 'session');
|
| You can also supply an alternative library name to be assigned
| in the controller:
|
|   $autoload['libraries'] = array('user_agent' => 'ua');
|
| */
$autoload['libraries'] = array('database');
```

5. Untuk kasus ini kita gunakan cara yang terakhir (no 4) karena kita asumsikan bahwa semua *controller* membutuhkan database. Maka kita *comment code* function construct yang ada di *controller* mahasiswa

```
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          // public function __construct(){
7              //     parent::__construct();
8              //     $this->load->database();
9
10         // }
11         // public function index()
12         // {
13             //     // $this->load->database();
```

- Setelah itu kita akan mengatur konfigurasi database. Untuk mengatur konfigurasi database maka buka folder **application/config/database.php**
- Kita set konfigurasinya seperti pada gambar dibawah ini

```
$active_group = 'default';
$query_builder = TRUE;

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

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

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	nama	varchar(100)	utf8_unicode_ci		No	None			Change Drop More
<input type="checkbox"/> 3	nim	varchar(50)	utf8_unicode_ci		No	None			Change Drop More
<input type="checkbox"/> 4	email	varchar(50)	utf8_unicode_ci		No	None			Change Drop More
<input type="checkbox"/> 5	jurusan	varchar(50)	utf8_unicode_ci		No	None			Change Drop More

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

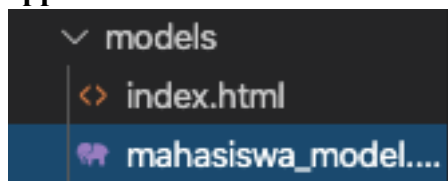
+ Options

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

☐ Check all With selected: Edit Copy Delete Export

10. Setelah kita membuat sebuah *database*, maka langkah berikutnya adalah membuat sebuah model untuk melakukan proses *database*. Di dalam model ini kita akan membuat sebuah *function* untuk melakukan perintah ambil data dari *database*. Untuk lebih jelasnya ikuti langkah-langkah berikut:

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



- b) Tambahkan *function* `getAllMahasiswa` pada `mahasiswa_model.php`

```

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              return $this->db->get('mahasiswa')->result_array();
9          }
10
11      }
12
13      /* End of file ModelName.php */
14
15  ?>
  
```

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

```

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              $this->load->model('mahasiswa_model');
10             $data['title']='List Mahasiswa';
11             $data['mahasiswa']=$this->mahasiswa_model->getAllMahasiswa();
12             $this->load->view('template/header',$data);
13             $this->load->view('mahasiswa/index',$data);
14             $this->load->view('template/footer');
15         }
16     }
17
18     /* End of file Controllername.php */
19
20
21  ?>

```

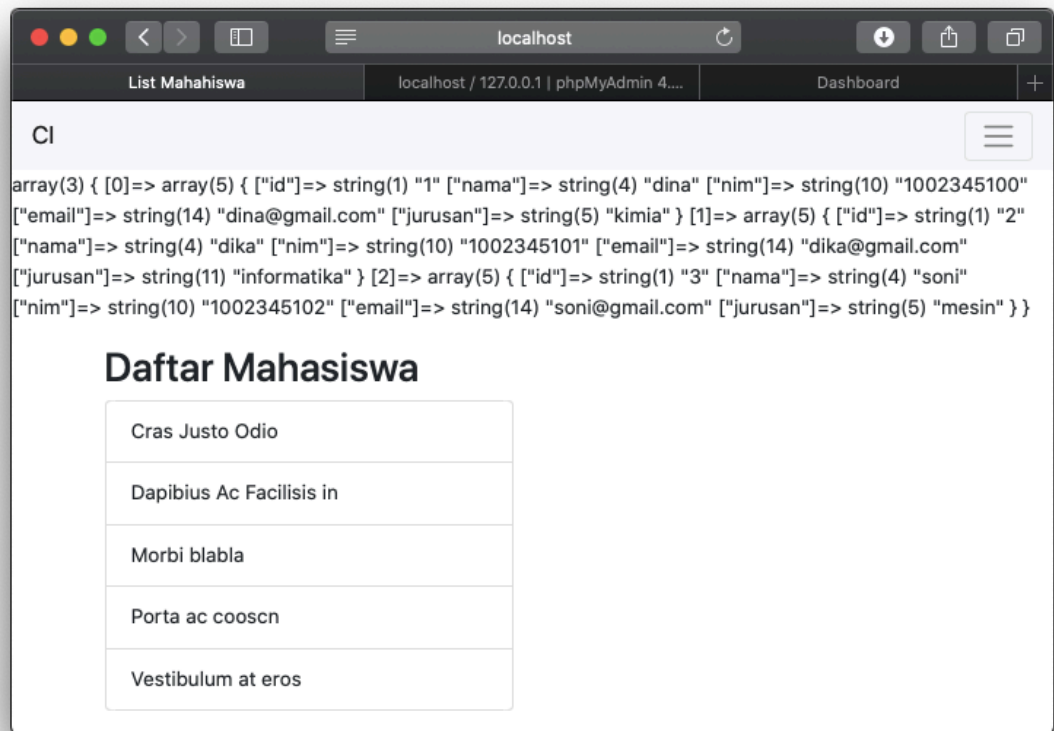
12. Jika kita lakukan `var_dump($mahasiswa)` pada **index.php** pada **views/mahasiswa/index**

```

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 Facilis in</li>
12                 <li class="list-group-item">Morbi blabla</li>
13                 <li class="list-group-item">Porta ac cooscn</li>
14                 <li class="list-group-item">Vestibulum at eros</li>
15             </ul>
16         </div>
17     </div>
18 </div>

```

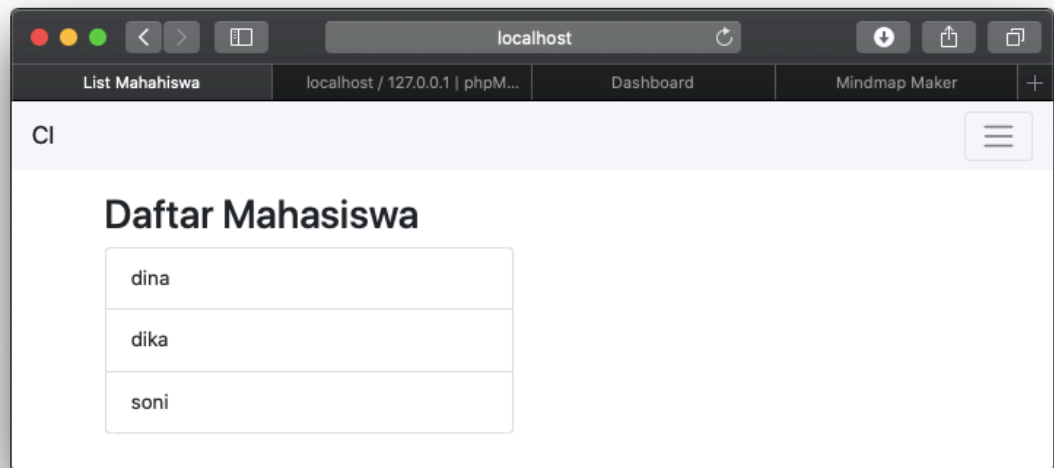
13. Jika berhasil maka data akan tampil seperti pada gambar berikut



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  |   <!-- 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 $mhs):?>
8  |               <li class="list-group-item"><?=$mhs['nama'];?></li>
9  |           <?php endforeach;?>
10 |       </ul>
11 |   </div>
12 </div>
13 </div>
```

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)

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPM0"
crossorigin="anonymous">

    <!-- <title>Hello, world!</title> -->
    <title><?= $title ?></title>
  </head>
  <body>
    <!-- navbar -->
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <a class="navbar-brand" href="#">CI</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarNavAltMakeUp">
        <div class="navbar-nav">
```



```

        <a class="nav-item nav-link" href="<?= base_url(); ?>">Home <span
class="sr-only">[current]</span></a>
        <a class="nav-item nav-link" href="<?= base_url(); ?>mahasiswa">Data
Mahasiswa</a>
        <a class="nav-item nav-link" href="<?= base_url(); ?>">About</a>
    </div>
</div>
</nav>

```

Keterangan :

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

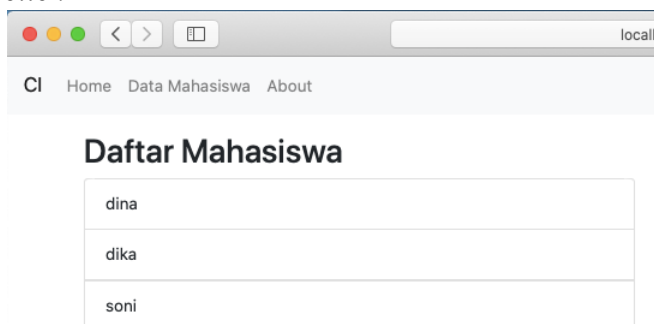
- b. Agar `base_url` dikenali oleh sistem maka tambahkan konfigurasi pada **config/autoload.php** tambahkan url pada `$autoload['helper']=array('url')`

```

/*
| -----
|  Auto-load Helper Files
| -----
|  Prototype:
|
|  $autoload['helper'] = array('url', 'file');
*/
$autoload['helper'] = array('url');

```

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



PRAKTIKUM 2

1. Buatlah sebuah *button* tambah data pada `views/mahasiswa/index.php` (line 3-7).
 Tambahkan *code* pada line 3-7 untuk membuat sebuah *button*

```

<div class="container">
    <div class="row mt-4">
        <div class="col-md-6">
            <a href="" class="btn btn-primary">Tambah Data</a>
        </div>
    </div>
    <!-- mt-3 artinya margin top 3 -->

```

```

<div class="row mt-3">
  <div class="col-md-6">
    <h2>Daftar Mahasiswa</h2>
    <ul class="list-group">
      <?php foreach($mahasiswa as $mhs):?>
        <li class="list-group-item"><?=$mhs['nama'];?></li>
      <?php endforeach;?>
    </ul>
  </div>
</div>
</div>

```

2. 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**.

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class mahasiswa extends CI_Controller {

    //fungsi yang akan dijalankan saat classnya diinstansiasi
    public function __construct(){
        //digunakan untuk menjalankan fungsi construct pada class
        parent(c_i_controller)
        parent::__construct();
        $this->load->model('mahasiswa_model');
    }

    public function index()
    {
        $this->load->model('mahasiswa_model');
        $data['title']='List Mahahiswa';
        $data['mahasiswa']=$this->mahasiswa_model->getAllMahasiswa();
        $this->load->view('template/header',$data);
        $this->load->view('mahasiswa/index',$data);
        $this->load->view('template/footer');
    }

}

/* End of file Controllername.php */

?>

```

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

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

```

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

```

```

        $this->load->view('template/header',$data);
        $this->load->view('mahasiswa/tambah',$data);
        $this->load->view('template/footer');
    }

```

Keterangan :

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

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

```

<div class="container">
    <div class="row mt-3">
        <div class="col-md-6">
            <div class="card">
                <div class="card-header">
                    Form Tambah Data Mahasiswa
                </div>
                <div class="card-body">
                    <form action="" method="post">
                        <div class="form-group">
                            <label for="nama">Nama</label>
                            <input type="text" class="form-control" id="nama">
                        </div>
                        <div class="form-group">
                            <label for="nim">NIM</label>
                            <input type="text" class="form-control" id="nim">
                        </div>
                        <div class="form-group">
                            <label for="email">Email</label>
                            <input type="text" class="form-control" id="email">
                        </div>
                        <div class="form-group">
                            <label for="jurusan">Jurusan</label>
                            <select class="form-control" name="jurusan"
id="jurusan">
                                <option selected>-----</option>
                                <option value="1">Kimia</option>
                                <option value="2">Informatika</option>
                                <option value="3">Mesin</option>
                            </select>
                        </div>
                        <button type="submit" name="submit" class="btn btn-
primary float-right">Submit</button>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>

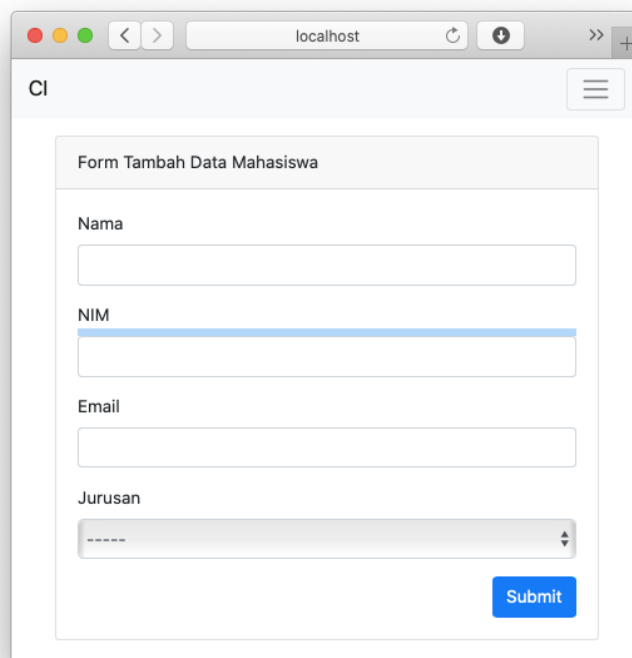
```

Keterangan :

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

```
<select class="form-control" name="jurusan" id="jurusan">
  <option selected>-----</option>
  <option value="kimia">Kimia</option>
  <option value="informatika">Informatika</option>
  <option value="mesin">Mesin</option>
</select>
```

5. maka hasil yang diharapkan akan seperti ini



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

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

```

        </ul>
    </div>
</div>
</div>

```

Keterangan :

Tambahkan *code* pada line 5 agar pada saat *user* melakukan *event click* maka *user* akan ditampilkan halaman tambah.

7. Tambahkan validasi menggunakan *form_validation*. Contoh validasi *form* adalah seperti *field* nama harus diisi, *nim* harus angka dsb. Untuk lebih jelasnya ikuti langkah berikut ini :

- a. Tambahkan *form_validation*

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

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

```

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

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

```

- c. Setelah kita tambahkan *form validation*, maka langkah berikutnya menambahkan *rule*-nya

- https://www.codeigniter.com/user_guide/libraries/form_validation.html?highlight=form#the-form (*searching pipe data*)
- Untuk melihat jenis-jenis *rule* bisa dilihat pada *rule reference*
- `$this->form_validation->set_rules('fieldname','fieldlabel','trim|required|min_length[5]|max_length[12]');`
- Jika kita lihat untuk menambahkan *rules* ada 3 parameter yaitu
 1. *fieldname* = nama dari *html*-nya, 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="text" class="form-control" id="nim"
name="nim">
    </div>
    <div class="form-group">
        <label for="email">Email</label>
        <input type="text" class="form-control"
id="email" name="email">
    </div>
    <div class="form-group">
        <label for="jurusan">Jurusan</label>
        <select class="form-control" name="jurusan"
id="jurusan">
            <option selected>-----</option>
            <option value="kimia">Kimia</option>
            <option
value="informatika">Informatika</option>
            <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*)

```

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

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

```

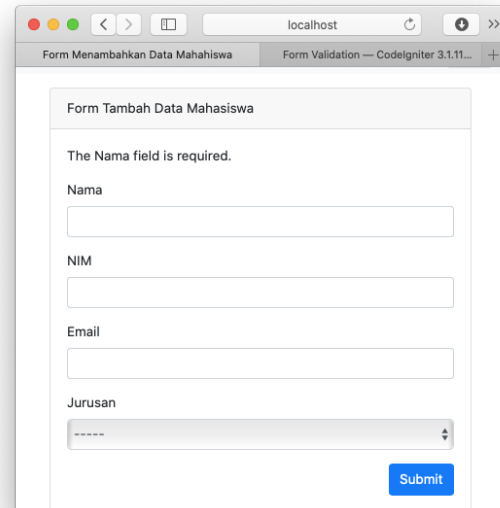
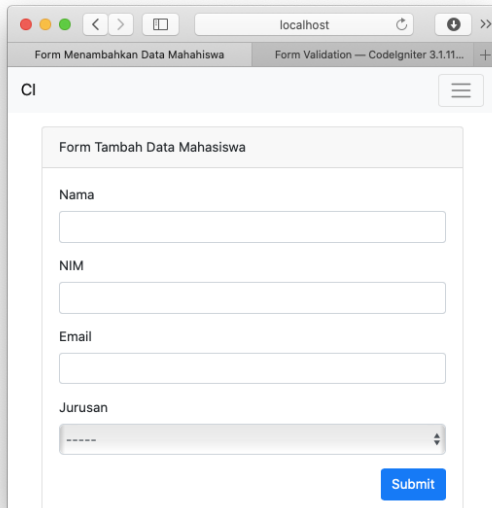
- f. Tambahkan pesan *error* pada **views/mahasiswa/tambah** (*line 11*)

```

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

```

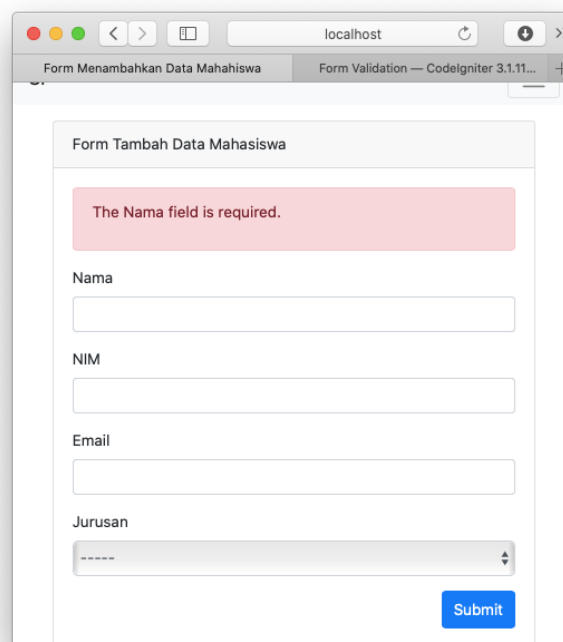
- g. Hasilnya saat *field* nama tidak diisi maka akan muncul keterangan the Nama Field is required



- h. Tambahkan *alert* agar pesan *error* lebih menarik setelah *code validation error*
- <https://getbootstrap.com/docs/4.1/components/alerts/>

```
<div class="card-body">
    <?= validation_errors() ?>
    <?php if(validation_errors()):?>
    <div class="alert alert-danger" role="alert">
        <?= validation_errors();?>
    </div>
    <?php endif ?>
    <form action="" method="post">
```

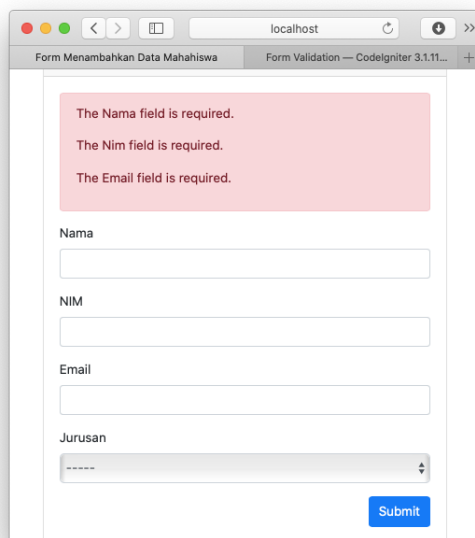
- i. Akan dihasilkan seperti gambar di bawah ini



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

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

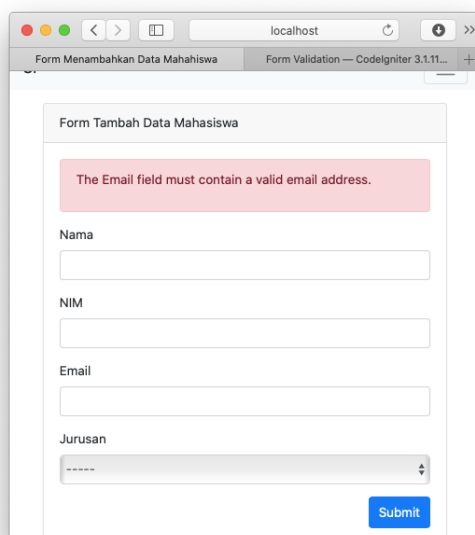
- k. Hasilnya seperti berikut ini.



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

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

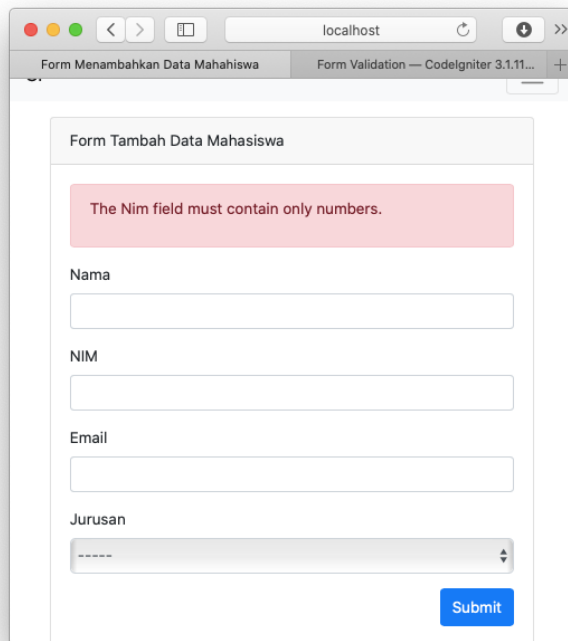
- m. Hasilnya sebagai berikut.



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

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

- o. Akan dihasilkan alert seperti berikut ini.



PRAKTIKUM 3

1. 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 ditambahkan ke database maka akan **redirect** ke **controller mahasiswa** (*line 46 - 47*)

```
public function tambah()  
{  
    $data['title']='Form Menambahkan Data Mahahiswa';  
  
    $this->form_validation->set_rules('nama', 'Nama', 'required');  
    $this->form_validation->set_rules('nim', 'Nim',  
'required|numeric');  
    $this->form_validation->set_rules('email', 'Email',  
'required|valid_email');  
  
    if($this->form_validation->run()==FALSE){  
        $this->load->view('template/header',$data);  
        $this->load->view('mahasiswa/tambah',$data);  
        $this->load->view('template/footer');  
    }else{
```

```

        $this->mahasiswa_model->tambahdatamhs();
        redirect('mahasiswa','refresh');
    }
}

```

2. Jika kita lihat *code* pada **controller mahasiswa**(line 46) terdapat **model** dengan nama **mahasiswa_model** dan *function*-nya **tambahdatamhs()**. Untuk itu kita harus membuat sebuah *function* dengan nama **tambahdatamhs()**
 3. Buatlah sebuah *function* **tambahdatamhs()** pada **models/mahasiswa_model**
- https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=insert#inserting-data
 - Tambahkan *code* seperti pada gambar dibawah ini

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class mahasiswa_model extends CI_Model {

    public function getAllMahasiswa(){
        return $this->db->get('mahasiswa')->result_array();
    }

    public function tambahdatamhs() {
        $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)
        ];
        $this->db->insert('mahasiswa',$data);
    }
}

/* End of file ModelName.php */

?>

```

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
 - 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

localhost

Form Menambahkan Data Mahasiswa

Form Validation — CodeIgniter 3.1.11...

CI

Form Tambah Data Mahasiswa

Nama

akhdan

NIM

1841720159

Email

akhdan@gmail.com

Jurusan

Informatika

Submit

localhost

List Mahasiswa

Form Validation — CodeIgniter 3.1.11...

CI

Tambah Data

Daftar Mahasiswa

dina
dika
soni
akhdan