

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT

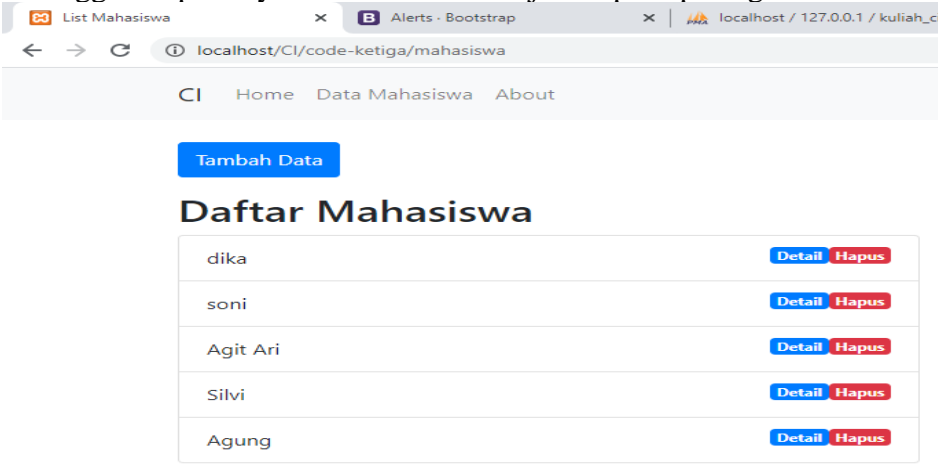
Jobsheet-4
Edit Data dan Detail Data, Searching Data

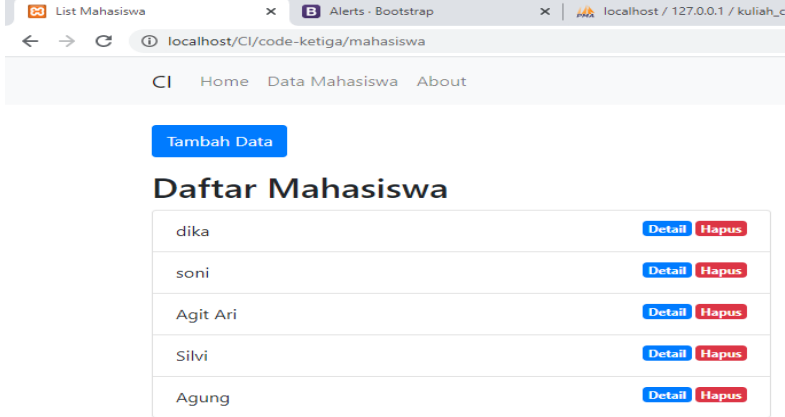


NAMA : DINA RISKY ALIN SAPUTRI
NIM : 1841720016
KELAS : 2B

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020

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

Langka h	Keterangan												
1	<p>Buka halaman views/mahasiswa/index.php tambahkan code berikut ini (line 31)</p> <pre><ul class="list-group"> <?php foreach (\$mahasiswa as \$mhs) : ?> <li class="list-group-item"><?=\$mhs['nama']; ?> <a href="<?=\$base_url(); ?>mahasiswa/hapus/<?=\$mhs['id']; ?>" class="badge badge-danger float-right" onclick="return confirm('Yakin Data ini akan dihapus');">Hapus <a href="<?=\$base_url(); ?>mahasiswa/detail/<?=\$mhs['id']; ?>" class="badge badge-primary float-right">Detail <?php endforeach; ?> </pre>												
2	<p>Sehingga tampilannya akan berubah menjadi seperti pada gambar dibawah ini</p>  <table border="1"><thead><tr><th colspan="2">Daftar Mahasiswa</th></tr></thead><tbody><tr><td>dika</td><td>Detail Hapus</td></tr><tr><td>soni</td><td>Detail Hapus</td></tr><tr><td>Agit Ari</td><td>Detail Hapus</td></tr><tr><td>Silvi</td><td>Detail Hapus</td></tr><tr><td>Agung</td><td>Detail Hapus</td></tr></tbody></table>	Daftar Mahasiswa		dika	Detail Hapus	soni	Detail Hapus	Agit Ari	Detail Hapus	Silvi	Detail Hapus	Agung	Detail Hapus
Daftar Mahasiswa													
dika	Detail Hapus												
soni	Detail Hapus												
Agit Ari	Detail Hapus												
Silvi	Detail Hapus												
Agung	Detail Hapus												
3	<p>Jika kita lihat letak button antara detail dan hapus terlalu berdekatan. Maka kita perlu tambahkan file css agar tampilannya lebih menarik. Ikuti langkah-langkah berikut ini :</p> <ol style="list-style-type: none">1) Tambahkan Code style berikut pada views/template/header.php (line 12-16) <pre><!--style css --> <style> .badge { margin-left: 3px; } </style> <!--<title>Hello, world!</title> --> <title><?=\$title ?></title> </head> <body></pre> <ol style="list-style-type: none">2) Hasilnya												

	
4	<p>Untuk selanjutnya kita akan membuat sebuah function dengan nama detail pada controller/mahasiswa. Untuk lebih jelasnya ikuti langkah berikut:</p> <ol style="list-style-type: none"> 1) Buat function dengan nama detail dan tambahkan code berikut <pre data-bbox="446 766 1312 1092"> public function detail(\$id) { \$data['title'] = 'Detail Mahasiswa'; \$data['mahasiswa'] = \$this->mahasiswa_model->getMahasiswaByID(\$id); \$this->load->view('template/header', \$data); \$this->load->view('mahasiswa/detail', \$data); \$this->load->view('template/footer'); } /* End of file Controllername.php */ </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Line 61 merupakan sebuah function dengan parameter id - \$data pada line 65 digunakan untuk menyimpan data yang diambil dari model (mahasiswa_model) dan dengan functionnya (getmahasiswaByID) yang nanti akan digunakan sebagai salah satu parameter pada line 67 - Line 67 nanti kita akan membuat sebuah view dengan nama detail
5	<p>Setelah kita membuat function detail pada controller mahasiswa maka selanjutnya kita akan membuat sebuah function di model dengan nama getmahasiswaByID(). Untuk lebih jelasnya ikuti langkah berikut</p> <ol style="list-style-type: none"> 1) Buka file models/mahasiswa_model buat sebuah function dengan nama getmahasiswaByID() dan tambahkan code berikut: <pre data-bbox="446 1522 1339 1711"> public function getMahasiswaByID(\$id) { return \$this->db->get_where('mahasiswa', ['id' => \$id])->row_array(); } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> • https://www.codeigniter.com/user_guide/database/query_builder.html?highlight=delete#selecting-data cari yang \$this->db>get_where() • \$query = \$this->db->get_where('mytable', array('id' => \$id), \$limit, \$offset);

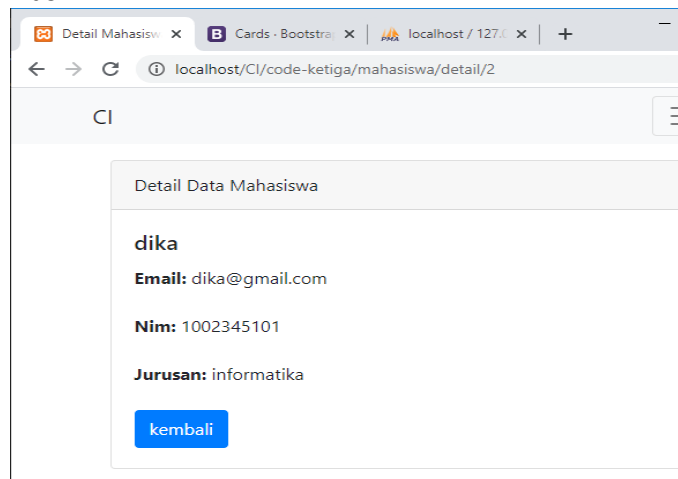
	<ul style="list-style-type: none"> • Pada <code>get_where</code> terdapat 2 parameter yaitu <ul style="list-style-type: none"> • mahasiswa = nama dari tabelnya • <code>['id'=>\$id]</code> = field id didalam tabel sama dengan id yang dikirim kedalam parameter • <code>row_array</code> = result 1 baris dalam bentuk array
6	<p>Setelah kita membuat function pada controller dan model maka langkah berikutnya adalah menampilkan isi dari database. Untuk lebih jelasnya ikuti langkah-langkah berikut :</p> <ol style="list-style-type: none"> 1) Buka file <code>views/mahasiswa</code> dan buat sebuah file dengan nama <code>detail.php</code> 2) Tambahkan code berikut pada <code>views/mahasiswa/detail.php</code> <ol style="list-style-type: none"> a. Buatlah div terlebih dahulu <div data-bbox="466 653 1060 921" data-label="Text"> <pre>code-ketiga > application > views > mahasiswa > detail.php 1 <div class="container"> 2 <div class="row mt-3"> 3 <div class="col-md-6"> 4 </div> 5 </div> 6 </div></pre> </div> b. Tambahkan card bootstrap https://getbootstrap.com/docs/4.1/components/card/ cari seperti pada gambar dibawah <div data-bbox="466 1035 1039 1694" data-label="Image"> <p>The screenshot shows the Bootstrap 4.1 documentation for the 'card' component. It features a 'Featured' card example with a title, text, and a button. Below the example is the HTML code used to create it.</p> <pre><div class="card"> <div class="card-header"> Featured </div> <div class="card-body"> <h5 class="card-title">Special title treatment</h5> <p class="card-text">With supporting text below as a natural lead-in to Go somewhere </div> </div></pre> </div> c. Copy paste Code diatas kedalam <code>detail.php</code> seperti pada gambar dibawah ini dan edit beberapa Code

```

autoload.php  mahasiswa.php  mahasiswa_model.php  index.php  detail.php  tambah.php
code-ketiga > application > views > mahasiswa > detail.php > div.container > div.row.mt-3 > div.col-md-6 > div.card >
1 <div class="container">
2 <div class="row mt-3">
3 <div class="col-md-6">
4 <div class="card">
5 <div class="card-header">
6 Detail Data Mahasiswa
7 </div>
8 <div class="card-body">
9 <h5 class="card-title"><?= $mahasiswa['nama']; ?></h5>
10 <p class="card-text">
11 <label for=""><b> Email:</b></label>
12 <?= $mahasiswa['email']; ?></p>
13 <p class="card-text">
14 <label for=""><b> Nim:</b></label>
15 <?= $mahasiswa['nim']; ?></p>
16 <p class="card-text">
17 <label for=""><b> Jurusan:</b></label>
18 <?= $mahasiswa['jurusan']; ?></p>
19 <a href="<?= base_url(); ?>mahasiswa" class="btn btn-primary">kembali</a>
20 </div>
21 </div>
22 </div>
23 </div>
24 </div>

```

d. Hasil



7

Setelah kita berhasil membuat detail mahasiswa maka selanjutnya kita akan membuat proses edit data. Untuk proses edit data ikuti langkahlangkah berikut :

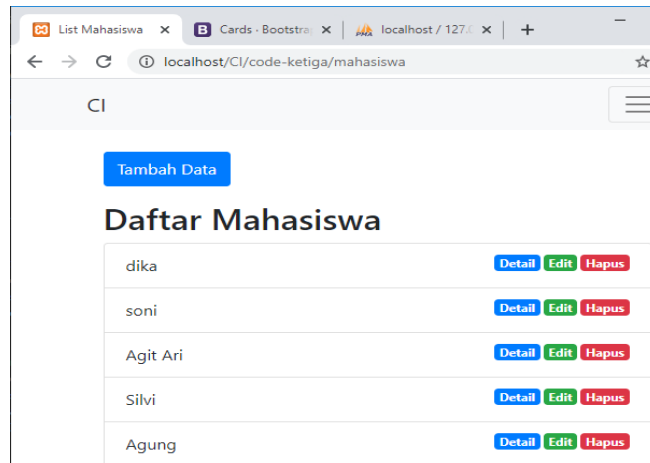
a. Buka file `views/mahasiswa/index.php` dan tambahkan Code berikut (Line 31-32)

```

<li class="list-group-item">
    <?= $mhs['nama']; ?>
    <a href="<?= base_url(); ?>mahasiswa/hapus/<?= $mhs['id']; ?>"
      class="badge badge-danger float-right"
      onclick="return confirm('Yakin Data ini akan dihapus');">Hapus</a>
    <a href="<?= base_url(); ?>mahasiswa/edit/<?= $mhs['id']; ?>"
      class="badge badge-success float-right">Edit</a>
    <a href="<?= base_url(); ?>mahasiswa/detail/<?= $mhs['id']; ?>"
      class="badge badge-primary float-right">Detail</a>
</li>

```

b. Hasil



- c. Buka file controller/mahasiswa dan buat sebuah function dengan nama edit, seperti pada gambar dibawah ini

```

public function edit($id)
{
    $data['title'] = 'Form Edit Data Mahasiswa';
    $data['mahasiswa'] = $this->mahasiswa_model->getMahasiswaByID($id);

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

    //$this->form_validation->set_rules('fieldname', 'fieldlabel', 'trim|required|min_length[5]|');
    $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) {
        # code...
        $this->load->view('template/header', $data);
        $this->load->view('mahasiswa/edit', $data);
        $this->load->view('template/footer');
    } else {
        # code...
        $this->mahasiswa_model->tambahdatamhs();
        // untukflashdata mmepunyai 2 parameter (nama flashdata/alias, isi dari flashdatanya)
        $this->session->set_flashdata('flash-data', 'diedit');
        redirect('mahasiswa', 'refresh');
    }
}

```

Keterangan :

- Pada line 72 sebuah function dengan nama edit yang mempunyai parameter berupa id dimana id ini nanti digunakan untuk menampilkan isi dari database (berdasarkan id / yang dipilih oleh user/line 75) ke dalam view/mahasiswa/edit (line 83 terdapat parameter id)
 - Rubah tambahdatamhs() menjadi ubahdatamahasiswa() line 87
- d. Setelah kita membuat function edit pada controller mahasiswa maka selanjutnya adalah membuat view dengan nama edit pada views/mahasiswa agar lebih mudah copy paste views/mahasiswa/tambah.php ke edit.php.

```

<div class="card-body">
  <!-- untuk menampilkan pesan error -->
  <?php if (validation_errors()) : ?>
    <div class="alert alert-danger" role="alert">
      <?> validation_errors() ?>
    </div>
  <?php endif ?>
  <form action="" method="post">
    <!-- http://getbootstrap.com/docs/4.1/components/form/ -->
    <div class="form-group">
      <label for="Nama">Nama</label>
      <input type="text"
        class="form-control"
        id="nama"
        name="nama"
        value="<?> $mahasiswa['nama']; ?>"
      <!-- <small class="form-text text-danger"><?> form_error('nama'); ?></small> -->
    </div>
    <div class="form-group">
      <label for="nim">Nim</label>
      <input type="number"
        class="form-control"
        id="nim" name="nim"
        value="<?> $mahasiswa['nim']; ?>"
      </div>
    <div class="form-group">
      <label for="email">Email</label>
      <input type="email"
        class="form-control"
        id="email"
        name="email"
        value="<?> $mahasiswa['email']; ?>"
      </div>
    <div class="form-group">
      <label for="jurusan">Jurusan</label>
      <select class="form-control" id="jurusan" name="jurusan">
        <option selected>----</option>
        <option value="Informatika">Informatika</option>
        <option value="Kimia">Kimia</option>
        <option value="Mesin">Mesin</option>
      </select>
    </div>
    <button type="submit" name="edit" class="btn btn-primary float-right"> Edit </button>
  </form>
</div>

```

Keterangan

- TambahkancanCode pada line 25, 34, 42 agar pada textbox nanti akan keluar isi dari id yang telah dipilih
- value="<?> \$mahasiswa['nama']; ?>" arti dari Code ini adalah bahwa isi dari value nanti diambil dari \$data['mahasiswa']=\$this>mahasiswa_model->getmahasiswaByID(\$id);dan \$mahasiswa adalah objek sedangkan ['nama'] adalah nama field di database (jadi sesuaikan besar kecilnya)
- \$mahasiswa['nama']asal

e. Hasilnya

The screenshot shows a web browser window with the address bar displaying 'localhost/CI/code-ketiga/mahasiswa/edit/2'. The page title is 'CI'. The main content is a form titled 'Form Merubah Data Mahasiswa'. The form has four input fields: 'Nama' with the value 'dika', 'Nim' with the value '1002345101', 'Email' with the value 'dika@gmail.com', and 'Jurusan' with a dropdown menu showing '----'. An 'Edit' button is located at the bottom right of the form.

- f. Untuk model dropdown pada jurusan tidak dapat kita menggunakan metode seperti pada textbox. Untuk itu kita akan merubah sedikit code pada controller/mahasiswa/edit dan views/mahasiswa/edit, untuk code seperti pada gambar dibawah ini

- o Pada Controller/mahasiswa/edit tambahkan code berikut (line 76)

```
public function edit($id)
{
    $data['title'] = 'Form Edit Data Mahasiswa';
    $data['mahasiswa'] = $this->mahasiswa_model->getMahasiswaByID($id);
    $data['jurusan'] = ['Teknik Informatika', 'Teknik Kimia', 'Teknik Industri', 'Teknik Mesin'];

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

    //$this->form_validation->set_rules('fielname', 'fieldlabel', 'trim|required|min_length[5]|max_
    $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) {
        # code...
        $this->load->view('template/header', $data);
        $this->load->view('mahasiswa/edit', $data);
        $this->load->view('template/footer');
    } else {
        # code...
        $this->mahasiswa_model->ubahdatamhs();
        // untuk flashdata mmepunyai 2 parameter (nama flashdata/alias, isi dari flashdatanya)
        $this->session->set_flashdata('flash-data', 'diedit');
        redirect('mahasiswa', 'refresh');
    }
}
/* End of file Controllername.php */
```

Keterangan :

- Pada line 76 kita membuat array dengan nama jurusan yang dimana isi arraynya adalah nama-nama jurusan yang nantinya akan ditampilkan pada views/mahasiswa/edit

- o Pada views/mahasiswa/edit tambahkan code berikut

```
</div>
<div class="form-group">
    <label for="jurusan">Jurusan</label>
    <select class="form-control" id="jurusan" name="jurusan">
        <?php foreach ($jurusan as $key) : ?>
            <?php if ($key == $mahasiswa['jurusan']) : ?>
                <option value="<?= $key ?>" selected><?= $key ?></option>
            <?php else : ?>
                <option value="<?= $key ?>"><?= $key ?></option>
            <?php endif; ?>
        <?php endforeach; ?>
    </select>
```

Keterangan :

- Pada line 47 kita tambahkan perulangan dengan menggunakan foreach dimana parameternya adalah \$jurusan alias \$key dan jangan lupa untuk ditutup menggunakan endforeach pada line 53
- Setelah kita membuat perulangan, maka kita tambahkan pengkondisian pada line 48 dengan parameter jika isi dari \$key sama dengan

\$mahasiswa['jurusan'] maka sistem akan mengselect \$key dan jika tidak sama maka sistem tidak akan melakukan apa-apa

- Kita rubah juga pada controller/mahasiswa/tambah dan views/mahasiswa/tambah

- Controller/mahasiswa/tambah.php

```
public function tambah()
{
    $data['title'] = 'Form Menambahkan Data Mahasiswa';
    $data['jurusan'] = ['Teknik Informatika', 'Teknik Kimia', 'Teknik Industri', 'Teknik Mesin'];
    $this->load->library('form_validation');

    //$this->form_validation->set_rules('fielname', 'fieldlabel', 'trim|required|min_length[5]|ma
    $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) {
        # code...
        $this->load->view('template/header', $data);
        $this->load->view('mahasiswa/tambah', $data);
        $this->load->view('template/footer');
    } else {
        # code...
        $this->mahasiswa_model->tambahdatamhs();
        // untuk flashdata mmepunyai 2 parameter (nama flashdata/alias, isi dari flashdatanya)
        $this->session->set_flashdata('flash-data', 'ditambahkan');
        redirect('mahasiswa', 'refresh');
    }
}
```

- views/mahasiswa/tambah.php

```
<div class="form-group">
    <label for="jurusan">Jurusan</label>
    <select class="form-control" id="jurusan" name="jurusan">
        <?php foreach ($jurusan as $key) : ?>
            <option
                value="<?= $key ?>"
                selected<?= $key ?>
            </option>
        <?php endforeach; ?>
    </select>
</div>
```

- Jika berhasil maka tampilan akan seperti pada gambar dibawah ini (Hasilnya akan sama dengan detail)

The screenshot shows a web browser window with the address bar displaying 'localhost/CI/code-ketiga/mahasiswa/edit/2'. The page title is 'CI'. The main content is a form titled 'Form Merubah Data Mahasiswa'. The form has four input fields: 'Nama' with the value 'dika', 'Nim' with the value '1002345101', 'Email' with the value 'dika@gmail.com', and 'Jurusan' which is a dropdown menu currently showing 'Teknik Informatika'. At the bottom right of the form is a blue button labeled 'Edit'.

- Tambahkan code pada line 18 pada views/mahasiswa/edit.php ini nanti digunakan untuk mengirim data (merubah data berdasarkan id)

```

16 <form action="" method="post">
17 <input type="hidden" name="id" value="<?= $mahasiswa['id']; ?>">
18 <!-- http://getbootstrap.com/docs/4.1/components/form/ -->
19 <div class="form-group">
20 <label for="Nama">Nama</label>
21 <input type="text"
22 class="form-control"
23 id="nama"
24 name="nama"
25 value="<?= $mahasiswa['nama']; ?>"
26 <!-- <small class="form-text text-danger"><?= form_error('nama'); ?></small>
27 </div>

```

8

Setelah kita berhasil membuat view untuk proses edit maka untuk selanjutnya kita akan membuat model (\$this->mahasiswa_model>ubahdatamhs()); (lihat pada controller/mahasiswa/edit). Untuk lebih ikuti langkah-langkah berikut:

- Buat sebuah function pada models/mahasiswa_model/ dengan nama ubahdatamhs()

```

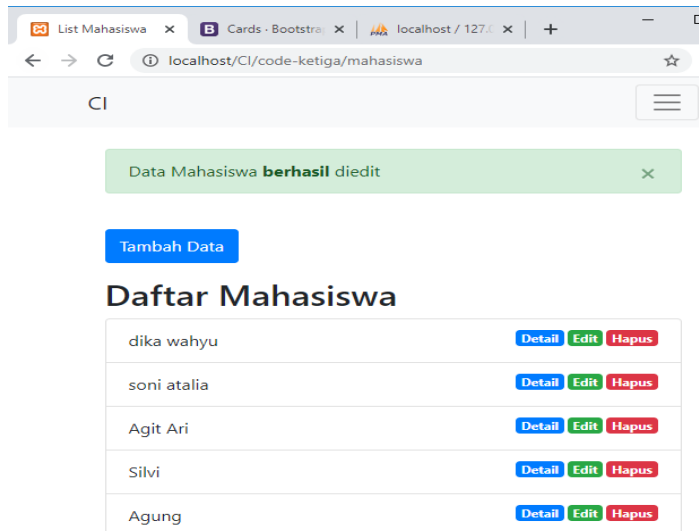
34 public function ubahdatamhs()
35 {
36     $data = [
37         "nama" => $this->input->post('nama', true),
38         "nim" => $this->input->post('nim', true),
39         "email" => $this->input->post('email', true),
40         "jurusan" => $this->input->post('jurusan', true)
41     ];
42     $this->db->where('id', $this->input->post('id'));
43     $this->db->update('mahasiswa', $data);
44 }
45

```

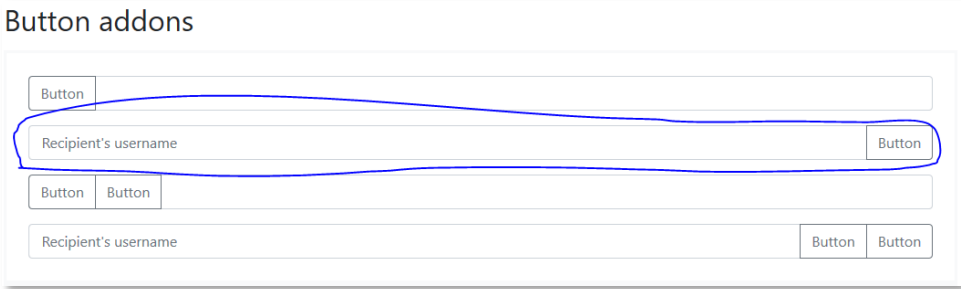
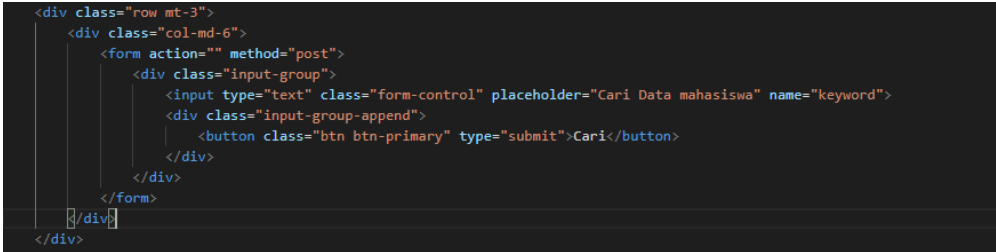
Keterangan :

- Pada line 51 terdapat 2 parameter yaitu
 - id = sesuaikan dengan field database
 - \$this->input->post('id') , id pada parameter post berasal dari name pada code `<input type="hidden" name="id" value="<?= $mahasiswa['id']; ?>">` yang ada pada views/mahasiswa/edit.php

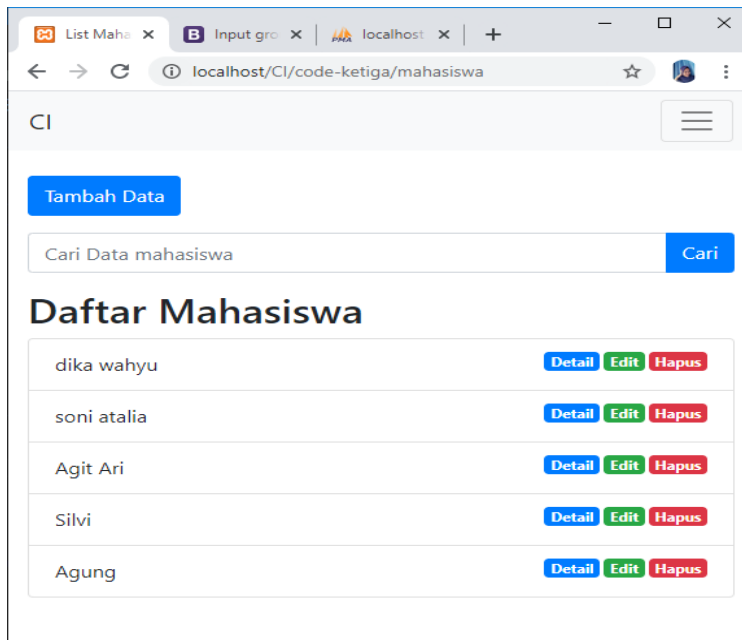
- Jika berhasil maka data akan berhasil dirubah atau diedit



Praktikum – Bagian 2: Membuat Searching Data

Langka h	Keterangan
1	Buka halaman index.php pada views/mahasiswa/index.php
2	<p>Tambahkan code berikut pada index.php</p> <ol style="list-style-type: none"> https://getbootstrap.com/docs/4.1/components/inputgroup/#button-addons Cari yang modelnya seperti pada gambar 9. <p>Button addons</p>  <ol style="list-style-type: none"> Copy paste code getbootstrap ke index.php <pre> <div class="input-group mb-3"> <div class="input-group-prepend"> <button class="btn btn-outline-secondary" type="button" id="button-addon1">Button</button> </div> <input type="text" class="form-control" placeholder="" aria-label="Example text with button addon" aria-describedby="button-addon1"> </div> <div class="input-group mb-3"> <input type="text" class="form-control" placeholder="Recipient's username" aria-label="Recipient's username" aria-describedby="button-addon2"> <div class="input-group-append"> <button class="btn btn-outline-secondary" type="button" id="button-addon2">Button</button> </div> </div> <div class="input-group mb-3"> <div class="input-group-prepend" id="button-addon3"> <button class="btn btn-outline-secondary" type="button">Button</button> <button class="btn btn-outline-secondary" type="button">Button</button> </div> <input type="text" class="form-control" placeholder="" aria-label="Example text with two button addons" aria-describedby="button-addon3"> </div> <div class="input-group"> <input type="text" class="form-control" placeholder="Recipient's username" aria-label="Recipient's username with two button addons"> <div class="input-group-append" id="button-addon4"> <button class="btn btn-outline-secondary" type="button">Button</button> <button class="btn btn-outline-secondary" type="button">Button</button> </div> </div> </pre> <ol style="list-style-type: none"> Sehingga seperti pada gambar dibawah ini (line 21-32) 

e. Hasil



3

Tambahkan atribut name dengan isinya keyword pada input. Seperti pada gambar dibawah ini

```
<div class="input-group">
  <input type="text" class="form-control" placeholder="Cari Data mahasiswa" name="keyword">
  <div class="input-group-append">
```

4

Tambahkan beberapa code pada controller mahasiswa/index.php seperti pada gambar dibawah ini (line 24-26)

```
public function index()
{
    //modul untuk load database
    //$this->load->database();
    $data['title'] = 'List Mahasiswa';
    $data['mahasiswa'] = $this->mahasiswa_model->getAllMahasiswa();
    if ($this->input->post('keyword')) {
        #code...
        $data['mahasiswa'] = $this->mahasiswa_model->cariDataMahasiswa();
    }
    $this->load->view('template/header', $data);
    $this->load->view('mahasiswa/index', $data);
    $this->load->view('template/footer');
}
```

5

Buatlah model dengan nama cariDataMahasiswa(); pada models/mahasiswa_model dan tambahkan code berikut:

```
public function cariDataMahasiswa()
{
    $keyword = $this->input->post('keyword');
    $this->db->like('id', $keyword);
    $this->db->update('mahasiswa')->result_array();
}
```

6	<p>Hasil</p>  <p>The screenshot shows a web browser at localhost/CI/code-ketiga/mahasiswa. It features a search bar with 'soni' entered and a 'Cari' button. Below the search bar, the title 'Daftar Mahasiswa' is displayed, followed by a table containing one entry: 'soni atalia'. To the right of this entry are three buttons: 'Detail' (blue), 'Edit' (green), and 'Hapus' (red).</p>
7	<p>Kita juga dapat mencari berdasarkan jurusan dengan cara</p> <ol style="list-style-type: none"> Tambahkan code berikut pada models/mahasiswa_model line 60  <pre> public function cariDataMahasiswa() { \$keyword = \$this->input->post('keyword'); \$this->db->like('nama', \$keyword); \$this->db->or_like('jurusan', \$keyword); return \$this->db->get('mahasiswa')->result_array(); } </pre> <ol style="list-style-type: none"> Hasil  <p>The screenshot shows the same web application with 'informatika' entered in the search bar. The search results table now displays 'Ani' as the student name.</p>
8	<p>Kita tambahkan alert jika mahasiswa tidak kita temukan. Ikuti langkah-langkah berikut</p> <ol style="list-style-type: none"> Tambahkan code berikut pada views/mahasiswa/index.php line 41-45  <pre> <!--mt-3 artinya margin top 3 --> <div class="row mt-3"> <div class="col-md-6"> <h2>Daftar Mahasiswa</h2> <!--alert--> <?php if (empty(\$mahasiswa)) : ?> <div class="alert alert-danger" role="alert"> Data Mahasiswa tidak ditemukan </div> <?php endif; ?> </div> </div> </pre>

b. Hasil

CI

Tambah Data

qualista Cari

Daftar Mahasiswa

Data Mahasiswa tidak ditemukan