LAPORAN PRAKTIKUM

PEMROGRAMAN WEB LANJUT

Jobsheet 10 (Membuat CRUD menggunakan Laravel)

-

Oleh :

Muhammad Royyan Zamzami

NIM. 1841720053

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

APRIL 2020

Praktikum – Bagian 1: Membuat CRUD di Laravel menggunakan Query Builder

a. Konfigurasi Database dan Pembuatan Tabel di MySQL

NO	KETERANGAN

1,	Membuat Project baru laravel
	`C:\xampp\htdocs\webLanjut>composer create-project --prefer-dist laravel/laravel laravel-crud`
	`Creating a "laravel/laravel" project at "./laravel-crud"`
	`Installing laravel/laravel (v7.3.0)`
	`  - Installing laravel/laravel (v7.3.0): Loading from cache`

2.	Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud.
	.env
	`APP_NAME=Laravel`
	`APP_ENV=local`
	`APP_KEY=base64:+OfjfPL/6bkoVWpg8BWfJ5xS3u9/`
	`APP_DEBUG=true`
	`APP_URL=http://localhost`
	
	`LOG_CHANNEL=stack`
	
	`DB_CONNECTION=mysql`
	`DB_HOST=localhost`
	`DB_PORT=3306`
	`DB_DATABASE=latihan_laravel`
	`DB_USERNAME=root`
	`DB_PASSWORD=`

3.	Buat database dengan nama “latihan_laravel” dan membuat table “mahasiswa”

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id	int(11)			Tidak	*Tidak ada*		AUTO_INCREM
2	nama	varchar(100)	latin1_swedish_ci		Tidak	*Tidak ada*		
3	nim	varchar(10)	latin1_swedish_ci		Tidak	*Tidak ada*		
4	email	varchar(50)	latin1_swedish_ci		Tidak	*Tidak ada*		
5	jurusan	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*		

4.	isi data di table mahasiswa

+ Opsi

			id	nama	nim	email	jurusan
Ubah	Salin	Hapus	1	Muhammad royyan zamzami	1841720053	royyanmz87@gmail.com	Teknilogi Informasi
Ubah	Salin	Hapus	2	Rulita Wahyu Mirdasari	1831620128	rulita.mirdasari@gmail.com	Administrasi Niaga

b. Menampilkan Data dari Database

NO	KETERANGAN
1.	Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan. Pertama, buatlah route pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data

2.	Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan
	`C:\xampp\htdocs\webLanjut\laravel-crud>php artisan make:controller MahasiswaController`
	`Controller created successfully.`

3.	Buat method index pada MahasiswaController.php pada folder app/Http/Controllers
	app > Http > Controllers > MahasiswaController.php
	`<?php`
	
	`namespace App\Http\Controllers;`
	
	`use Illuminate\Http\Request;`
	`use Illuminate\Support\Facades\DB;`
	
	`class MahasiswaController extends Controller`
	`{`
	`  public function index()`
	`  {`
	`  $mahasiswa = DB::table('mahasiswa')->get();`
	`  return view('index',['mahasiswa' => $mahasiswa]);`
	`  }`
	`}`

4.	Membuat View dengan nama “master.blade.php”

resources > views > master.blade.php

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" rel="stylesheet">
    <title>Document</title>
</head>
<body>
    <div class="container">
        <div class="row-mt-3">
            <div class="row-md-6">
                <div class="card">
                    <div class="card-header text-center">
                        <h2>@yield('judul_halaman')</h2>
                    </div>
                    <div class="card-body">
                        @yield('konten')
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

5. Sekarang buat file index.blade.php yang menerapkan template master.blade.php dan akan digunakan untuk menampilkan data

resources > views > index.blade.php

```blade
@extends('master')

@section('title', 'Home')

@section('judul_halaman', 'Data Mahasiswa')

@section('konten')
    <a href="/mahasiswa/tambah" class="btn btn-primary">Tambah Data Mahasiswa</a>
    <br>
    <br>
    <table class="table table-bordered table-hover table-striped">
        <thead>
            <tr>
                <th>Nama</th>
                <th>NIM</th>
            </tr>
        </thead>
        <tbody>
            @foreach($mahasiswa as $mhs)
            <tr>
                <td>{{ $mhs->nama }}</td>
                <td>{{ $mhs->nim }}</td>
            </tr>
            @endforeach
        </tbody>
    </table>
@endsection
```

6. Hasil

c. Memasukkan Data (Create) ke Database

NO	KETERANGAN
1.	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController 21 22 `Route::get('/mahasiswa/tambah','MahasiswaController@tambah');`

2.	Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.
	`public function tambah()`
	`{`
	`return view('tambah');`
	`}`

3.	Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. View tambah juga mengaplikasikan master.blade.php.

esources > views > tambah.blade.php

```php
@extends('master')

@section('title', 'Tambah Data')

@section('judul_halaman', 'Tambah Data Mahasiswa')

@section('konten')
    <a href="/mahasiswa" class="btn btn-danger">kembali</a>
    <br>
    <br>
    <form action="/mahasiswa/simpan" methon="post">
        {{ csrf_field() }}
        <div class="form-group">
            <label for="namamhs">Nama</label>
            <input type="text" class="form-control" required="required" name="namamhs"> <br>
        </div>
        <div class="form-group">
            <label for="nimmhs">NIM</label>
            <input type="number" class="form-control" required="required" name="nimmhs"> <br>
        </div>
        <div class="form-group">
            <label for="emailmhs">E-mail</label>
            <input type="email" class="form-control" required="required" name="emailmhs"> <br>
        </div>
        <div class="form-group">
            <label for="jurusanmhs">Jurusan</label>
            <input type="text" class="form-control" required="required" name="jurusanmhs"> <br>
        </div>
        <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button>
    </form>
```

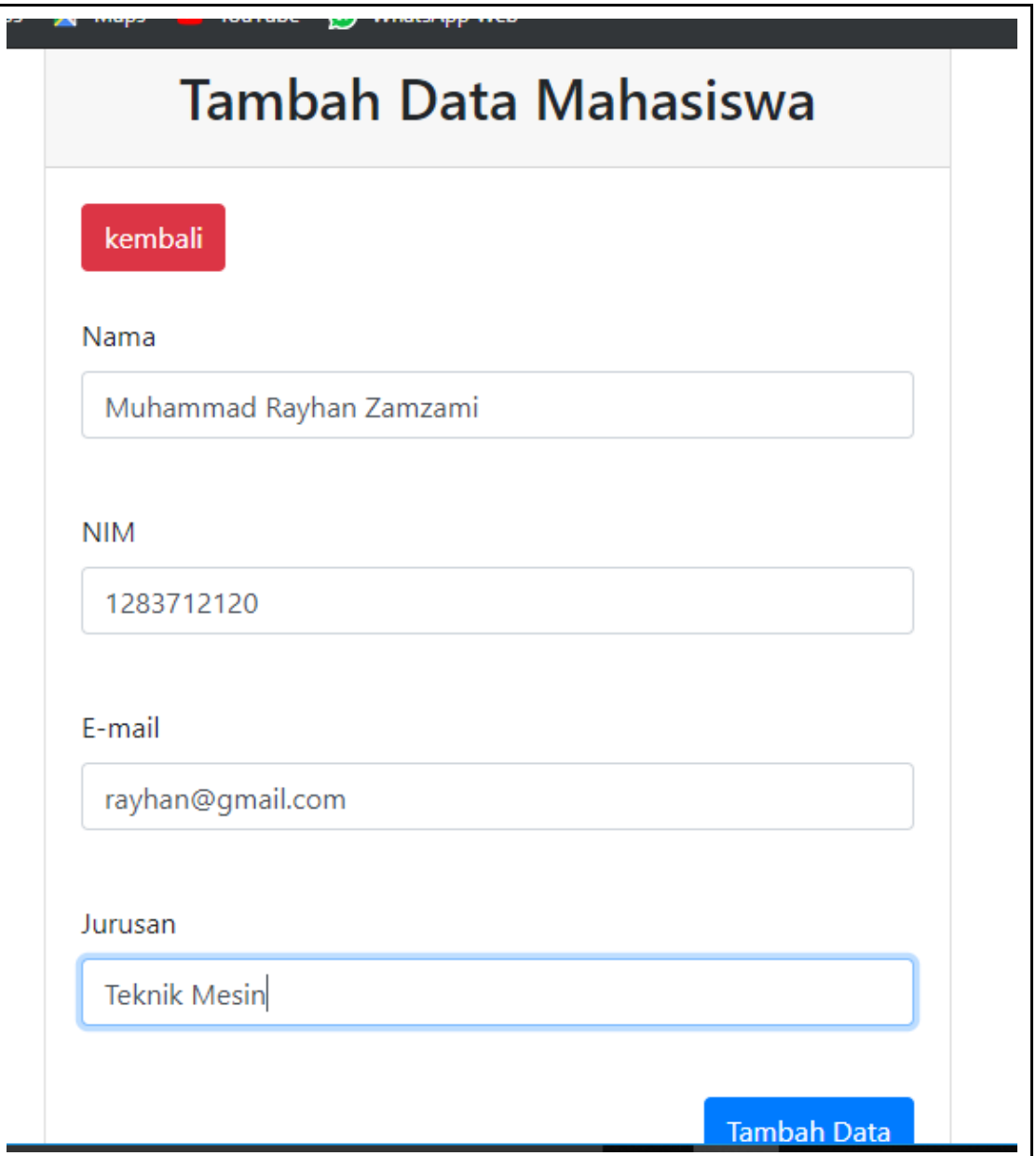
4. Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.

```php

Route::post('/mahasiswa/simpan','MahasiswaController@simpan');
```

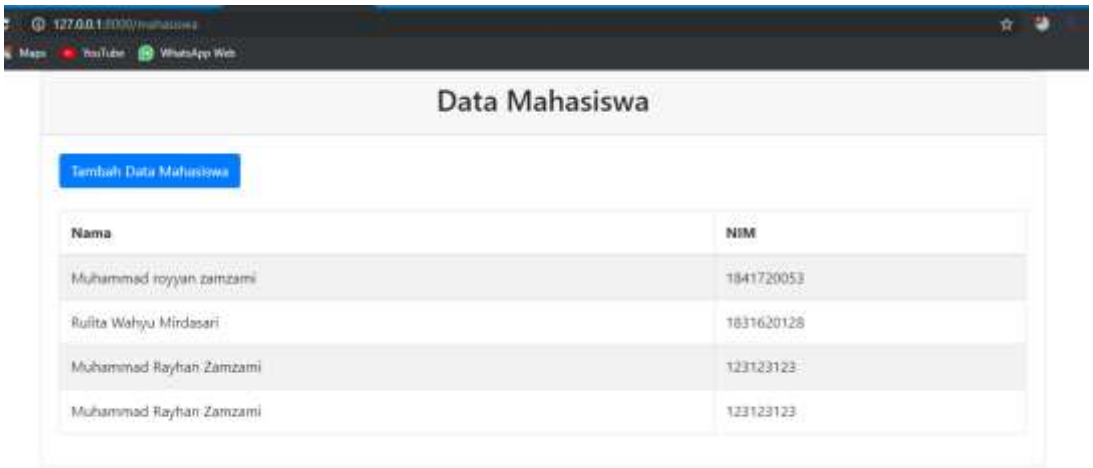
5. Buat Method simpan di mahasiswaController

```php
public function simpan(Request $request)
{
    DB::table('mahasiswa')->insert([
        'nama' => $request->namamhs,
        'nim' => $request->nimmhs,
        'email' => $request->emailmhs,
        'jurusan' => $request->jurusanmhs
    ]);
    return redirect('/mahasiswa');

}
```

6. Kembali coba jalankan localhost:8000 dan pilih tombol ‘Tambah Data Mahasiswa’, maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.



Setelah kita klik tombol tambah data, maka data baru akan ditampilkan pada halaman index.



d. Melihat Detail Data dari Database

NO	KETERANGAN

1.	Buatlah tombol untuk melihat detail data (Line 28) pada file index.blade.php di folder resources/views
	`<td>`
	`  <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a>`
	`</td>`

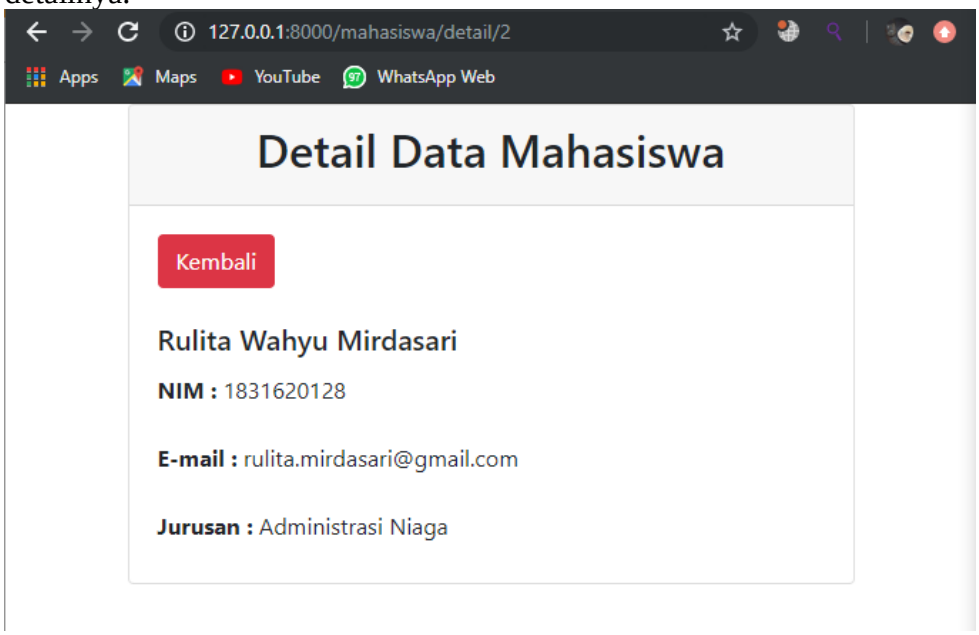
2.	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController `Route::get('/mahasiswa/detail/{id}','MahasiswaController@detail');`

3.	Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.
	`public function detail()`
	`{`
	`  $mahasiswa = DB::table('mahasiswa')->where('id', $id)->get();`
	`  return view('detail', ['mahasiswa' => $mahasiswa]);`
	`}`

4.	Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa

```
1   @extends('master')
2
3   @section('title', 'Detail Mahasiswa')
4
5   @section('judul_halaman', 'Detail Data Mahasiswa')
6
7   @section('konten')
8       <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
9       <br>
10      <br>
11
12      @foreach($mahasiswa as $mhs)
13      <h5 class="card-title">{{ $mhs->nama }}</h5>
14      <p class="card-text">
15          <label for=""><b>NIM : </b></label>
16          {{ $mhs->nim }}
17      </p>
18      <p class="card-text">
19          <label for=""><b>E-mail : </b></label>
20          {{ $mhs->email }}
21      </p>
22      <p class="card-text">
23          <label for=""><b>Jurusan : </b></label>
24          {{ $mhs->jurusan }}
25      </p>
26      @endforeach
27  @endsection
```

6. Jalankan localhost:8000 dan pilih tombol ‘Detail’ di suatu data yang ingin kita lihat detailnya.



e. Mengubah Data (Update) dari Database

NO	KETERANGAN

1.	Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views
	`<td>`
	`  <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a>`
	`  <a href="/mahasiswa/edit/{{ $mhs->id }}" class="badge badge-warning">Edit</a>`
	`</td>`

2.	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit 27 28 `Route::get('/mahasiswa/edit/{id}','MahasiswaController@edit');`

3.	Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.
	`public function edit($id)`
	`{`
	`  $mahasiswa = DB::table('mahasiswa')->where('id', $id)->get();`
	`  return view('edit', ['mahasiswa' => $mahasiswa]);`
	`}`

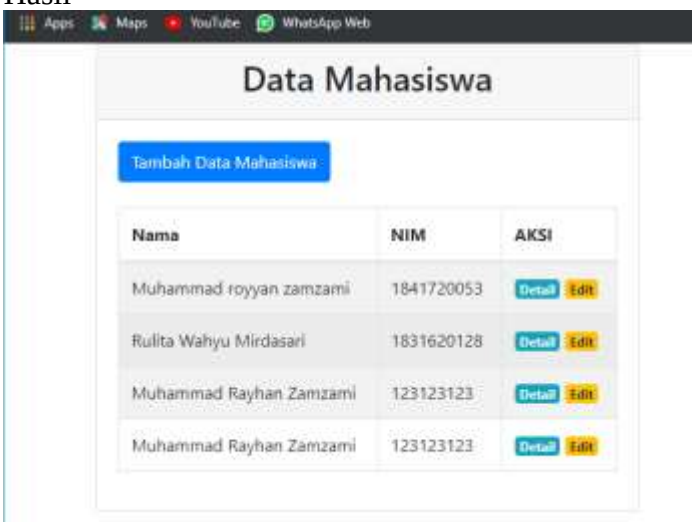
4.	Kemudian buatlah view edit.blade.php yang berisi form untuk mengubah data pada folder resources/views. View edit juga mengaplikasikan master.blade.php. sources > views > edit.blade.php `@section('judul_halaman', 'Edit Data Mahasiswa')` `@section('konten')` `  <a href="/mahasiswa" class="btn btn-danger">Kembali</a>` `   ` `   ` `  @foreach($mahasiswa as $mhs)` `  <form action="/mahasiswa/update" method="post">` `  {{ csrf_field() }}` `  <input type="Hidden" name="id" value="{{ $mhs->id }}">  ` `  <div class="form-group">` `  <label for="namamhs">Nama</label>` `  <input type="text" class="form-control" required="required" name="namamhs" value="{{ $mhs->nama }}">  ` `  </div>` `  <div class="form-group">` `  <label for="nimmhs">NIM</label>` `  <input type="number" class="form-control" required="required" name="nimmhs" value="{{ $mhs->nim }}">  ` `  </div>` `  <div class="form-group">` `  <label for="emailmhs">E-mail</label>` `  <input type="email" class="form-control" required="required" name="emailmhs" value="{{ $mhs->email }}">  ` `  </div>` `  <div class="form-group">` `  <label for="jurusanmhs">Jurusan</label>` `  <input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ $mhs->jurusan }}">  ` `  </div>` `  <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button>` `  </form>` `  @endforeach` `@endsection`
5.	Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update. Oleh karena itu, kita buat terlebih dahulu route tersebut.

```
29
30  Route::post('/mahasiswa/update','MahasiswaController@update');
```

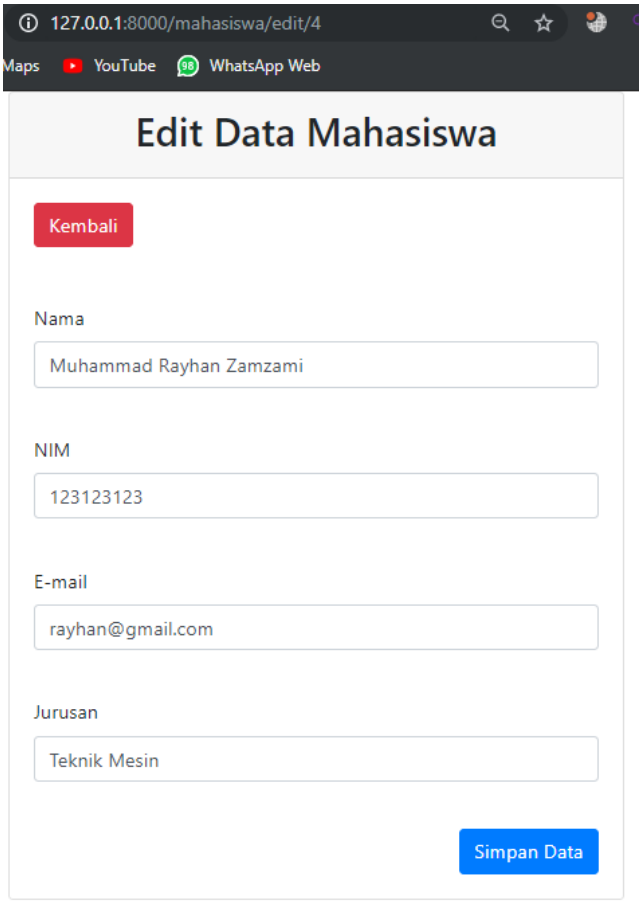
6. Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database

```
public function update(Request $request)
{
    DB::table('mahasiswa')->where('id', $request->id)->update([
        'nama' => $request->namamhs,
        'nim' => $request->nimmhs,
        'email' => $request->emailmhs,
        'jurusan' => $request->jurusanmhs
    ]);
    return redirect('/');
}
```

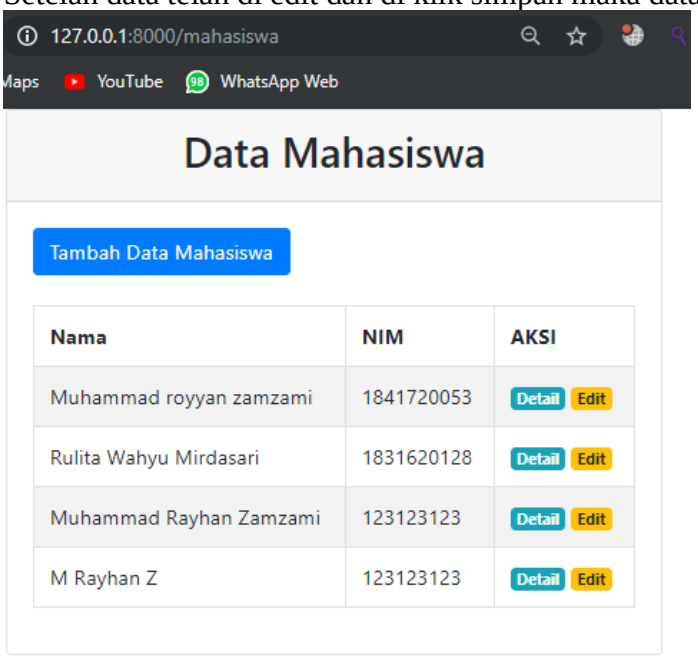
7. Hasil



Setelah di klik edit akan muncul halaman seperti berikut



Setelah data telah di edit dan di klik simpan maka data akan terupdate



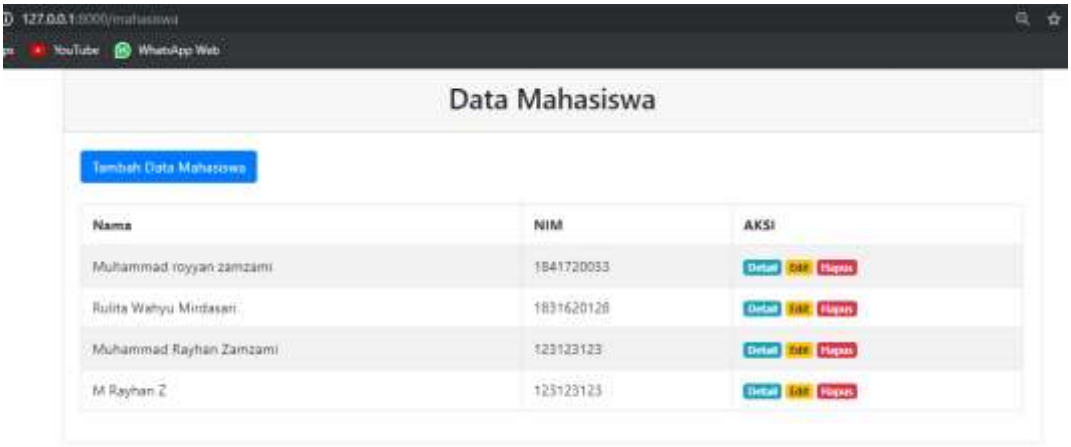
f. Menghapus Data (Delete) dari Database

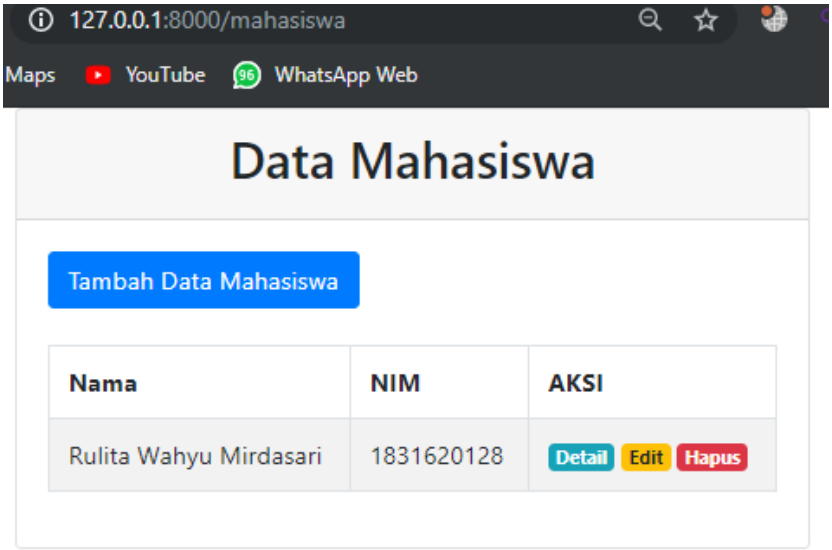
NO	KETERANGAN

1.	Buatlah tombol untuk menghapus data (Line 30) pada file index.blade.php di folder resources/views
	`<a href="/mahasiswa/hapus/{{ $mhs->id }}" class="badge badge-danger">Hapus</a>`
	`</td>`
	`tr>`

2.	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController
	`31`
	`32 Route::get('/mahasiswa/hapus/{id}','MahasiswaController@hapus');`

3.	Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.
	`public function hapus($id)`
	`{`
	`DB::table('mahasiswa')->where('id', $id)->delete();`
	`return redirect('/mahasiswa');`
	`}`

4.	Hasil  Selelah di klik hapus



Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

a. Konfigurasi Database

NO	KETERANGAN

1,	Membuat Project baru laravel
	`C:\xampp\htdocs\webLanjut>composer create-project --prefer-dist laravel/laravel laravel-crud`
	`Creating a "laravel/laravel" project at "./laravel-crud"`
	`Installing laravel/laravel (v7.3.0)`
	`- Installing laravel/laravel (v7.3.0): Loading from cache`

2.	Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud.
	.env
	`APP_NAME=Laravel`
	`APP_ENV=local`
	`APP_KEY=base64:+OfjfPL/6bkoVWpg8BWfJ5xS3u9/`
	`APP_DEBUG=true`
	`APP_URL=http://localhost`
	
	`LOG_CHANNEL=stack`
	
	`DB_CONNECTION=mysql`
	`DB_HOST=localhost`
	`DB_PORT=3306`
	`DB_DATABASE=latihan_laravel`
	`DB_USERNAME=root`
	`DB_PASSWORD=`

3.	Pada project ini kita gunakan database dan tabel yang sebelumnya digunakan pada

	Praktikum Bagian 1. Tambahkan kolom created_at dan updated_at yang bertipe TIMESTAMP dan default nilainya NULL

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	nim	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	email	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	jurusan	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
6	created_at	timestamp			Ya	NULL			Ubah Hapus Lainnya
7	updated_at	timestamp			Ya	NULL			Ubah Hapus Lainnya

b. Menampilkan Data dari Database

NO	KETERANGAN

1.	Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan. Pertama, buatlah route pada routes/web.php sehingga keti
	`Route::get('/','MahasiswaController@index');`
	`Route::get('/mahasiswa','MahasiswaController@index');`

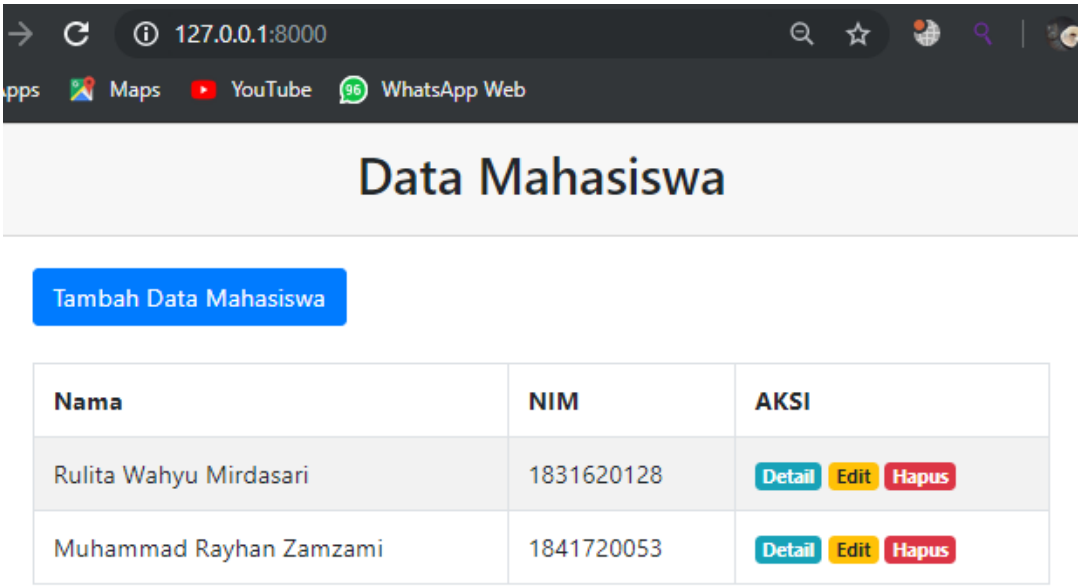
2.	Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan
	`C:\xampp\htdocs\webLanjut\laravel-crud-kedua>php artisan make:model Mahasiswa`
	`Model created successfully.`

3.	Ubah model Mahasiswa.php pada folder App menjadi seperti berikut
	`<?php`
	
	`namespace App;`
	
	`use Illuminate\Database\Eloquent\Model;`
	
	`class Mahasiswa extends Model`
	`{`
	`  protected $table = "mahasiswa";`
	`}`

4.	Buat controller baru yaitu MahasiswaController menggunakan php artisan
	`C:\xampp\htdocs\webLanjut\laravel-crud-kedua>php artisan make:controller MahasiswaController`
	`Controller created successfully.`

5.	Buat method index pada MahasiswaController.php pada folder app/Http/Controllers

	`public function index()`
	`{`
	`  $mahasiswa = Mahasiswa::all();`
	`  return view('index',['mahasiswa' => $mahasiswa]);`
	`}`

6.	untuk file .blade saya copy dari praktikum 1
7.	Hasil 

c. Memasukkan Data (Create) ke Database

NO	KETERANGAN
1.	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan. `Route::get('/mahasiswa/tambah','MahasiswaController@tambah');`

2.	Buat method tambah pada MahasiswaController.php
	`public function tambah()`
	`{`
	`  return view('tambah');`
	`}`

3.	untuk file .blade saya copy dari praktikum 1
4.	Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh

karena itu, kita buat terlebih dahulu route tersebut.

```
24
25    Route::post('/mahasiswa/simpan','MahasiswaController@simpan');
26
```

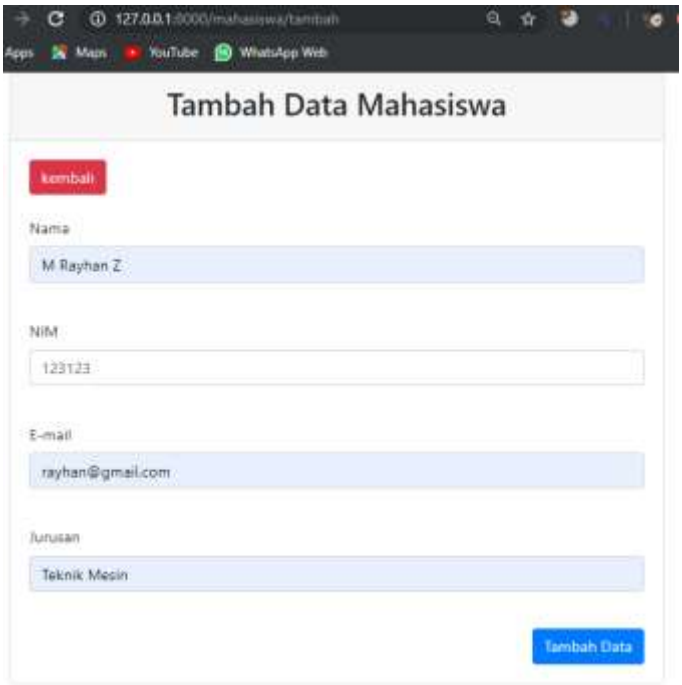
5. Buat method simpan pada MahasiswaController untuk menyimpan data

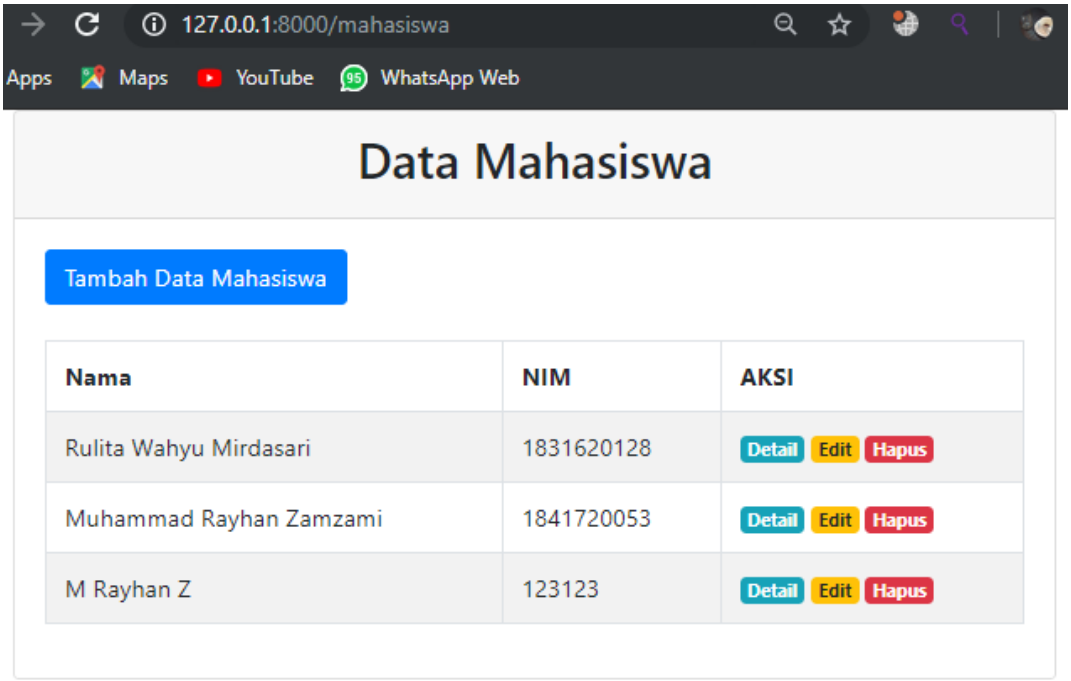
```
public function simpan(Request $request)
{
    Mahasiswa::create([
        'nama' => $request->namamhs,
        'nim' => $request->nimmhs,
        'email' => $request->emailmhs,
        'jurusan' => $request->jurusanmhs
    ]);
    return redirect('/mahasiswa');
}
```

6. Karena menggunakan fungsi create pada Controller,maka butuh ditambahkan code pada model

```
protected $table = "mahasiswa";
protected $fillable = ['nama', 'nim', 'email', 'jurusan'];
```

7. Hasil





d. Melihat Detail Data dari Database

NO	KETERANGAN
1.	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController `Route::get('/mahasiswa/detail/{id}','MahasiswaController@detail');`

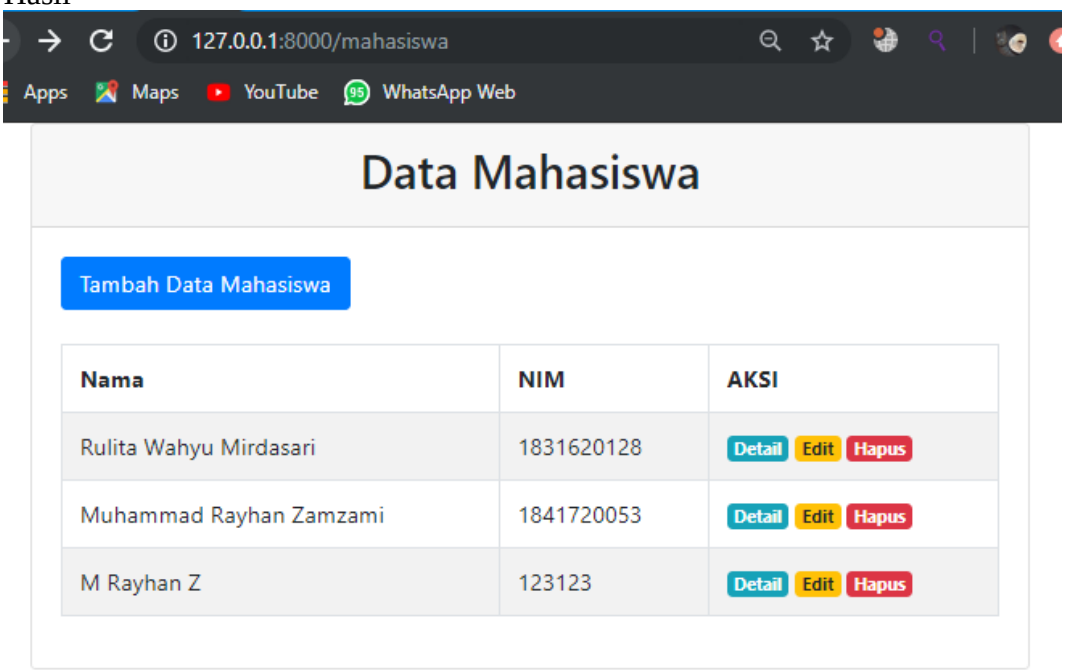
2.	Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.
	`public function detail($id)`
	`{`
	`$mahasiswa = Mahasiswa::find($id);`
	`return view('detail', ['mahasiswa' => $mahasiswa]);`
	`}`

3.	Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php

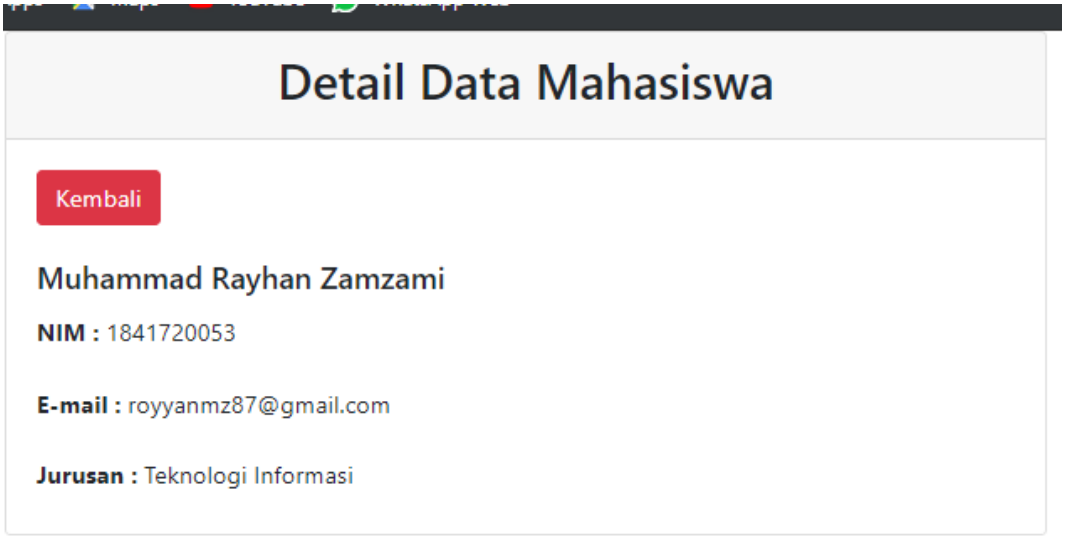
resources > views > detail.blade.php

```php
@extends('master')

@section('title', 'Detail Mahasiswa')

@section('judul_halaman', 'Detail Data Mahasiswa')

@section('konten')
    <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
    <br>
    <br>

    <h5 class="card-title">{{ $mahasiswa->nama }}</h5>
    <p class="card-text">
        <label for=""><b>NIM : </b></label>
        {{ $mahasiswa->nim }}
    </p>
    <p class="card-text">
        <label for=""><b>E-mail : </b></label>
        {{ $mahasiswa->email }}
    </p>
    <p class="card-text">
        <label for=""><b>Jurusan : </b></label>
        {{ $mahasiswa->jurusan }}
    </p>
@endsection
```

4. Hasil



Setelah di klik detail



e. Mengubah Data (Update) dari Database

NO	KETERANGAN
1.	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController `Route::get('/mahasiswa/edit/{id}','MahasiswaController@edit');`

2.	Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.
	`public function edit($id)`
	`{`
	`  $mahasiswa = Mahasiswa::find($id);`
	`  return view('edit', ['mahasiswa' => $mahasiswa]);`
	`}`

3.	Kemudian buatlah view edit.blade.php yang berisi form untuk mengubah data pada folder resources/views. View edit juga mengaplikasikan master.blade.php.

<table>
<tr><td></td><td>

resources / views / edit.blade.php

```
1  @extends('master')
2
3  @section('title', 'Edit Data')
4
5  @section('judul_halaman', 'Edit Data Mahasiswa')
6
7  @section('konten')
8      <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
9      <br>
10     <br>
11     <form action="/mahasiswa/update" method="post">
12         {{ csrf_field() }}
13         <input type="Hidden" name="id" value="{{ $mhs->id }}"> <br>
14         <div class="form-group">
15             <label for="namamhs">Nama</label>
16             <input type="text" class="form-control" required="required" name="namamhs" value="{{ $mahasiswa->nama }}"> <br>
17         </div>
18         <div class="form-group">
19             <label for="nimmhs">NIM</label>
20             <input type="number" class="form-control" required="required" name="nimmhs" value="{{ $mahasiswa->nim }}"> <br>
21         </div>
22         <div class="form-group">
23             <label for="emailmhs">E-mail</label>
24             <input type="email" class="form-control" required="required" name="emailmhs" value="{{ $mahasiswa->email }}"> <
25         </div>
26         <div class="form-group">
27             <label for="jurusanmhs">Jurusan</label>
28             <input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ $mahasiswa->jurusan }}"
29         </div>
30         <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button>
31     </form>
32 @endsection
```

</td></tr>
<tr><td>4.</td><td>

Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update/{id}. Oleh karena itu, kita buat terlebih dahulu route tersebut.

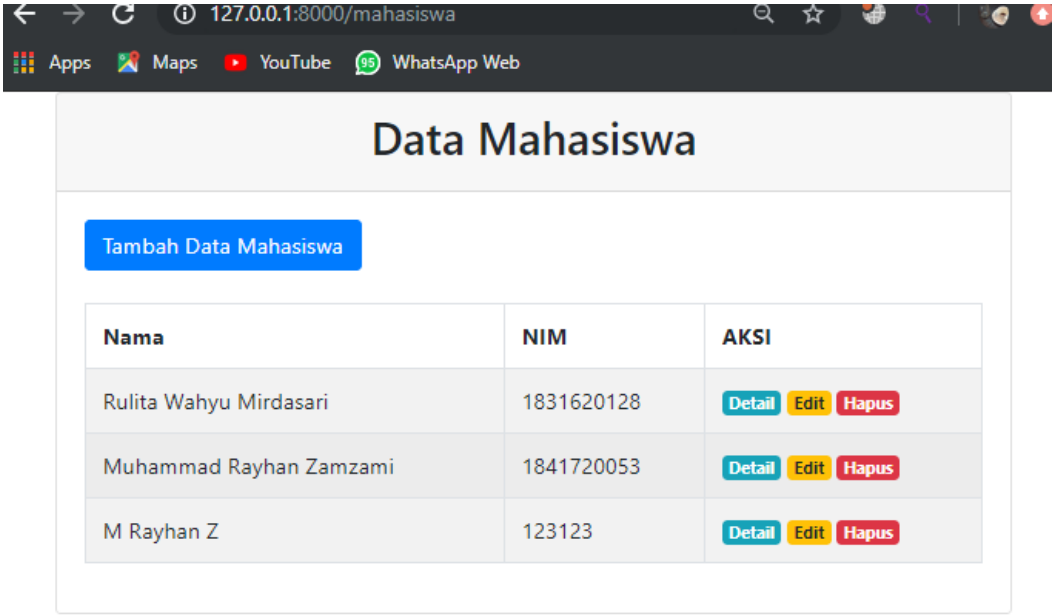
```
30
31    Route::post('/mahasiswa/update','MahasiswaController@update');
```

</td></tr>
<tr><td>5.</td><td>

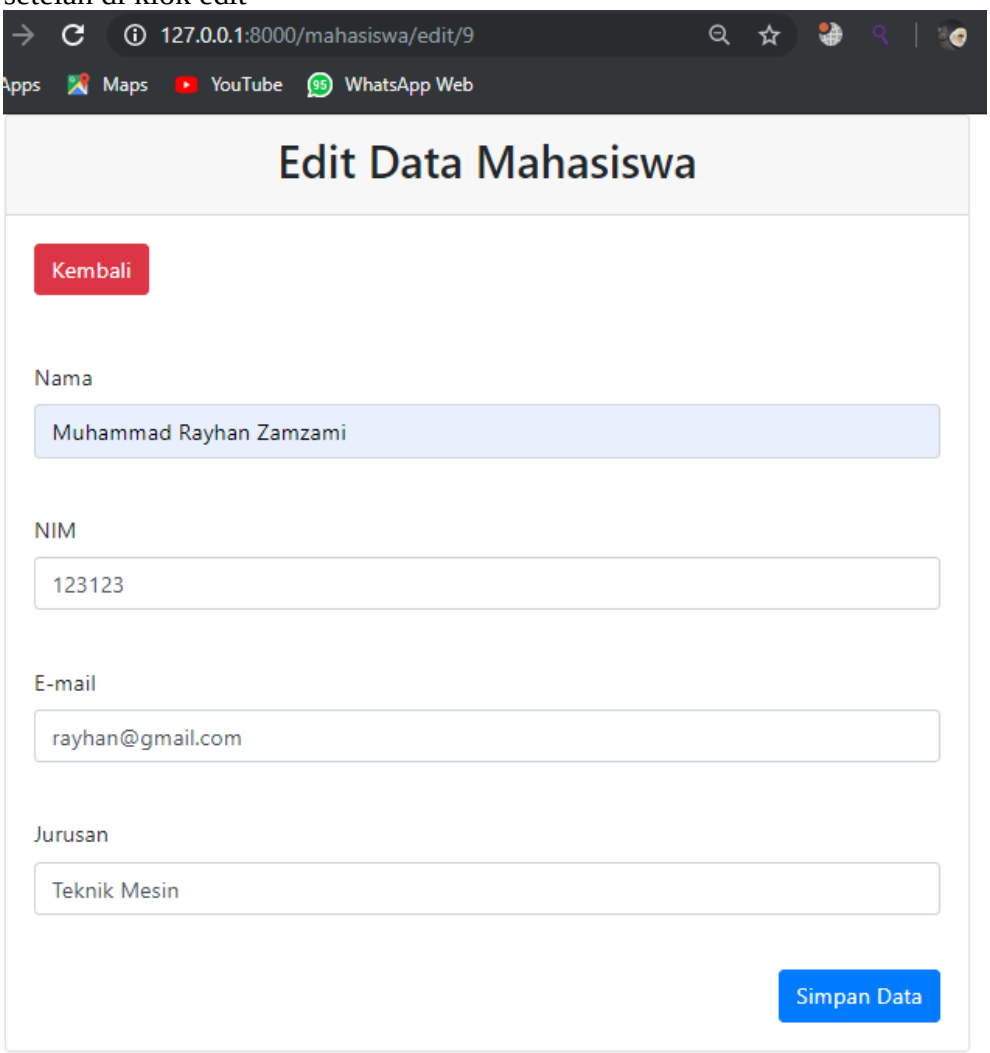
Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database

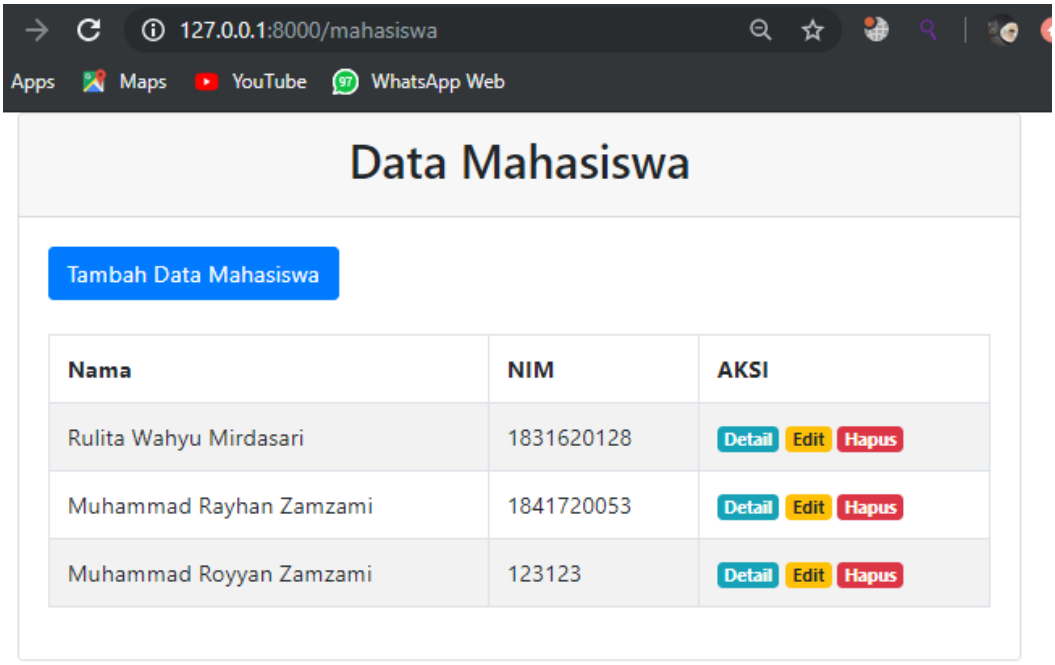
```
public function update(Request $request)
{
    $mahasiswa = Mahasiswa::find($id);
    $mahasiswa->nama = $request->namamhs;
    $mahasiswa->nim = $request->nimmhs;
    $mahasiswa->email = $request->emailmhs;
    $mahasiswa->jurusan = $request->jurusanmhs;
    $mahasiswa->save();

    return redirect('/mahasiswa');
}
```

</td></tr>
<tr><td>6.</td><td>Hasil</td></tr>
</table>



setelah di klok edit



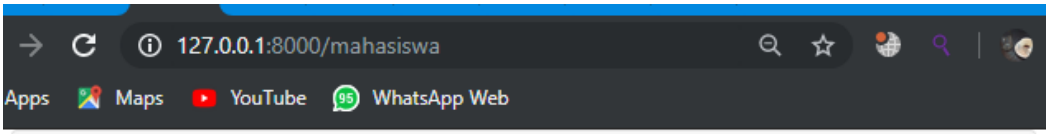


f. Menghapus Data (Delete) dari Database

NO	KETERANGAN
1.	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController `Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');`

2.	Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.
	`public function hapus($id)`
	`{`
	`$mahasiswa = Mahasiswa::find($id);`
	`$mahasiswa->delete();`
	`return redirect('/mahasiswa');`
	`}`

3.	Hasil

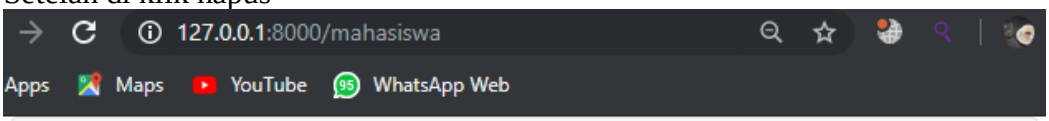


Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	AKSI
Rulita Wahyu Mirdasari	1831620128	Detail Edit Hapus
Muhammad Rayhan Zamzami	1841720053	Detail Edit Hapus
Muhammad Royyan Zamzami	123123	Detail Edit Hapus

Setelah di klik hapus



Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	AKSI
Rulita Wahyu Mirdasari	1831620128	Detail Edit Hapus
Muhammad Royyan Zamzami	123123	Detail Edit Hapus