LAPORAN TUGAS PEMROGRAMAN WEB LANJUT

“JOBSHEET 10”

Oleh :

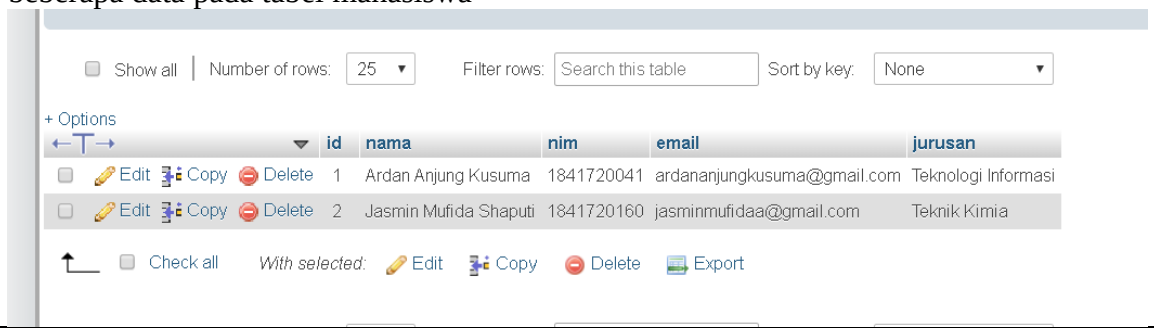
NAMA : ARDAN ANJUNG KUSUMA
NIM : 1841720041

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020

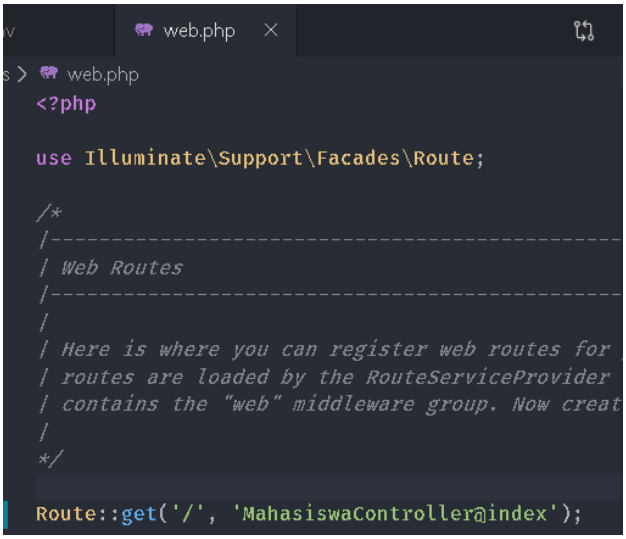
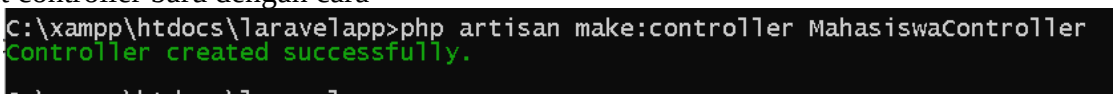
Praktikum – Bagian 1: Membuat CRUD di Laravel menggunakan Query Builder

a. Konfigurasi Database dan Pembuatan Tabel di MySQL

1	Konfigurasi file .env .env APP_NAME=Laravel APP_ENV=local APP_KEY=base64:iYX2snXwAtRSuTNeeZoqjBjixQKlfPJ3 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= Keterangan : - Nama Database latihan_laravel - Username root - Password kosong
2	Buat tabel mahasiswa di mysql pada database dengan isi berikut ardananjungkusuma (Ardan Anju Pemograman-web-lanjut-TI2B | E myadmin/server_databases.php?server=1 Server: 127.0.0.1 » Database: latihan_laravel » Table: mahasiswa Browse Structure SQL Search Insert Table name: mahasiswa Add 1 Name Type Length/Values id INT 11 Pick from Central Columns nama VARCHAR 100 Pick from Central Columns nim VARCHAR 10 Pick from Central Columns email VARCHAR 50 Pick from Central Columns jurusan VARCHAR 20 Pick from Central Columns

3	Isi beberapa data pada tabel mahasiswa 

b. Menampilkan Data dari Database

1	Buatlah route pada **routes/web.php** sehingga saat pertama kali project dijalankan akan terbuka halaman yang menampilkan data  Keterangan - Ketika route / diakses, akan dijalankan method indexd pada controller MahasiswaController
2	Buat controller baru dengan cara 

3	Buat method index pada controller MahasiswaController
	MahasiswaController.php × .env web.php
	app > Http > Controllers > MahasiswaController.php > ...
	1 <?php
	2
	3 namespace App\Http\Controllers;
	4
	5 use Illuminate\Http\Request;
	6 use Illuminate\Support\Facades\DB;
	7
	8 class MahasiswaController extends Controller
	9 {
	10 public function index()
	11 {
	12 //mengambil data dari tabel mahasiswa
	13 $mahasiswa = DB::table('mahasiswa')->get();
	14
	15 //mengirim data mahasiswa ke view index
	16 return view('index', ['mahasiswa' => $mahasiswa]);
	17 }
	18 }
	19
	- Lakukan import library DB agar DB dapat digunakan (line 6)
	- Kita dapat mengambil data dari tabel mahasiswa menggunakan query builder laravel dan disimpan pada variabel mahasiswa pada line 13
	- Pada line 16 kita mengirimkan ke view index data mahasiswa

4	Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat template blade (seperti file header-footer pada pembahasan CI) dengan nama master.blade.php pada folder resources/views.

MahasiswaController.php | master.blade.php | .env | web.php

esources > views > master.blade.php > html

```
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boot
    <title> @yield('title') </title>
</head>

<body>
    <div class="container">
        <div class="row mt-3">
            <div class="col-md-6">
                <div class="card-header text-center">
                    <!-- Judul Halaman -->
                    <h2> @yield('judul_halaman') </h2>
                </div>
                <div class="card-body">
                    <!-- Konten Halaman -->
                    @yield('konten')
                </div>
            </div>
        </div>
    </div>
</body>

</html>
```

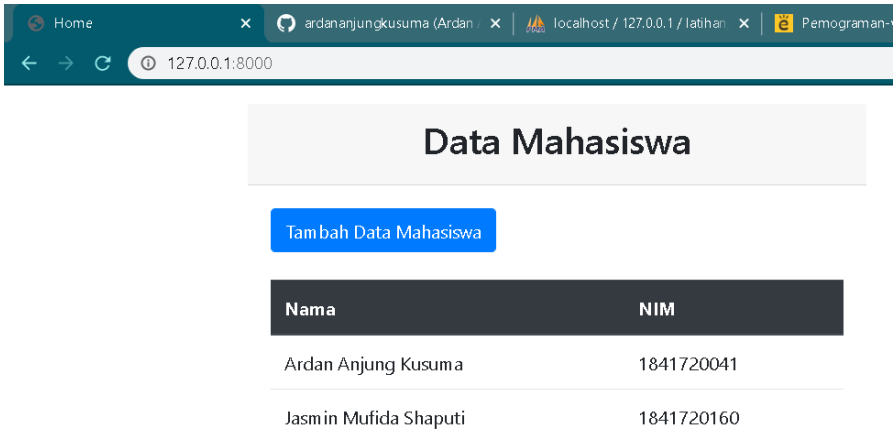
- Fungsi @yield sebagai penanda bagian pada master blade. Nantinya bagian @yield ini akan diisi sesuai dengan halaman view yang menerapkan master.blade.php

5 Buat file index.blade.php yang menerapkan template master.blade.php

er.php | master.blade.php | index.blade.php | .env | web.php

resources > views > index.blade.php > ...

```
@extends('master')

@section('title','Home')

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

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

	- Line extends menunjukan bahwa file index menerapkan blade lain (konsep oop)
6	Jalankan php artisan serve dan buka 127.0.0.1:8000 

c. Memasukkan Data (Create) ke Database

1	Buat route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController `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()`
	`{`
	`  //memanggil view tambah`
	`  return view('tambah');`
	`}`

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

MahasiswaController.php | tambah.blade.php | Extension: Laravel Snippets | master.blade.php | index.bla

esources > views > tambah.blade.php > form

```
1  @extends('master')
2
3  @section('title','Tambah Data')
4
5  @section('judul_halaman','Tambah Data Mahasiswa')
6
7  @section('konten')
8  <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
9  <br>
10 <br>
11 <form action="/mahasiswa/simpan" method="POST">
12     {{ csrf_field() }}
13     <div class="form-group">
14         <label>Nama</label>
15         <input type="text" class="form-control" required name="namamhs"><br>
16     </div>
17     <div class="form-group">
18         <label>NIM</label>
19         <input type="number" class="form-control" required name="nimmhs"><br>
20     </div>
21     <div class="form-group">
22         <label>Email</label>
23         <input type="email" class="form-control" required name="emailmhs"><br>
24     </div>
25     <div class="form-group">
26         <label>Jurusan</label>
27         <input type="text" class="form-control" required name="jurusanmhs"><br>
28     </div>
29     <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</but
30 </form>
```

- Line 13-32 form memasukan nama,nimemail,jurusan
- Line 13 terdapat action mahasiswa/simpan yang menunjukan routes mahasiswa simpan dimana data form tersebut akan dikirimkan ke fungsi simpan pada controller mahasiswaController
- Line 14 terdapat csrf field yang digunakan untuk menerapkan fitur laravel yaitu csrf protection yang mencegah penginputan diluar aplikasi dan akan menggenerate token otomatis yang dibuat dalam bentuk form hidden

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

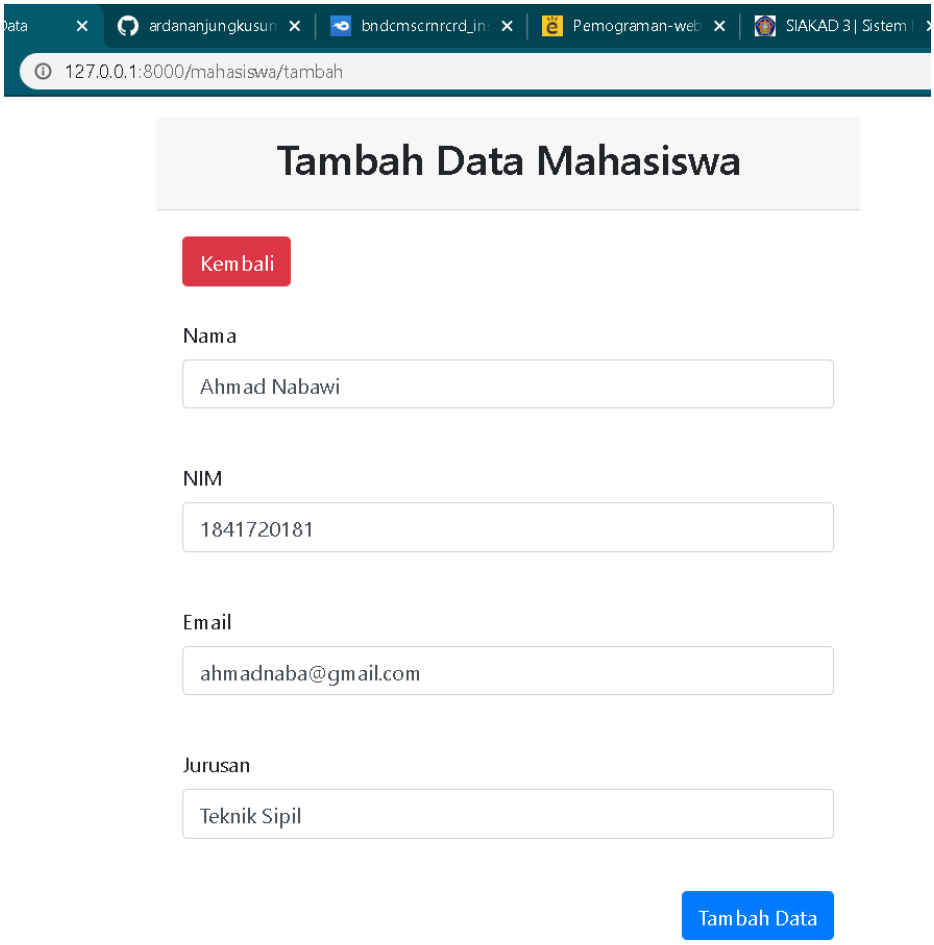
```
Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');
```

5 Buat method simpan pada MahasiswaController untuk menyimpan data ke database

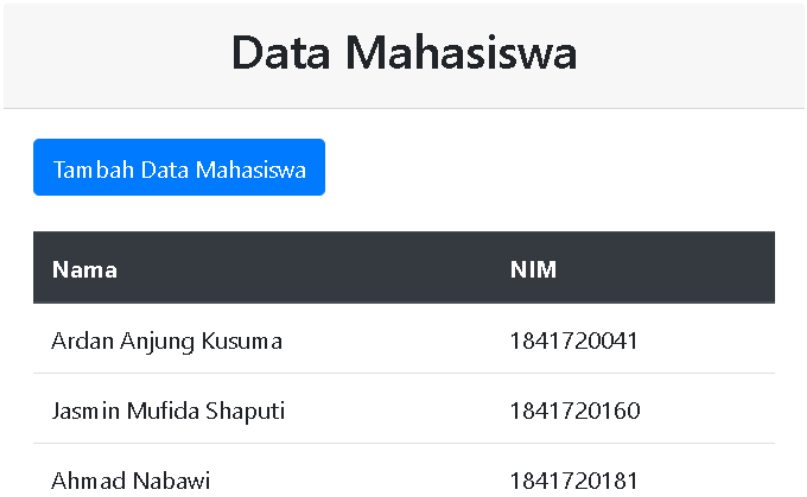
```php
public function simpan(Request $request)
{
    // insert data ke table mahasiswa
    DB::table('mahasiswa')->insert([
        'nama' => $request->namamhs,
        'nim' => $request->nimmhs,
        'email' => $request->emailmhs,
        'jurusan' => $request->jurusanmhs
    ]);
    return redirect('/mahasiswa');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

- Variabel $request untuk menerima data yang akan ditambahkan ke database

6 Jalankan tambah data mahasiswa lalu isi form tersebut



Setelah kita tambah maka muncul data baru



d. Melihat Detail Data dari Database

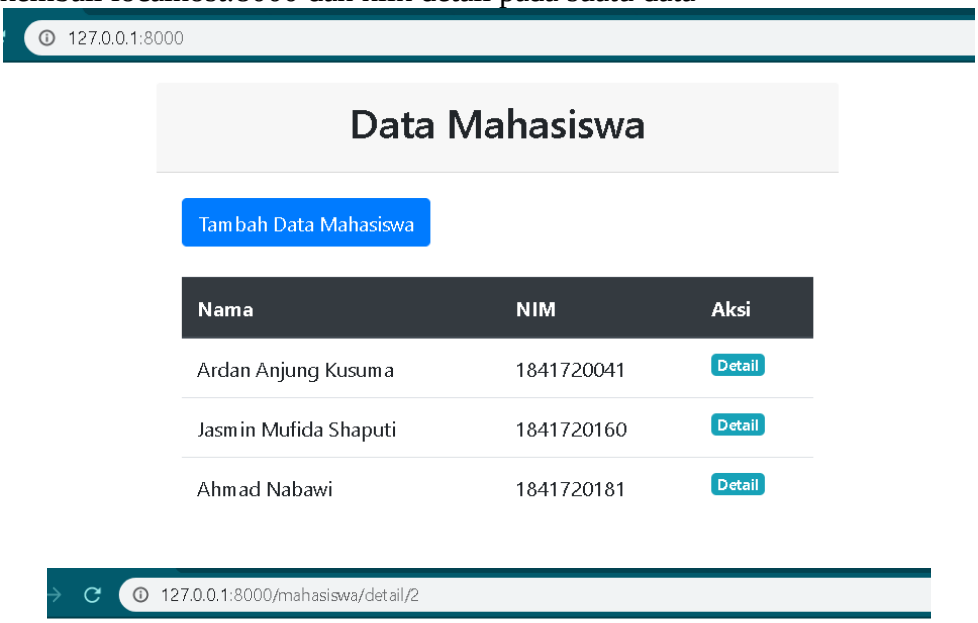
1	Buatlah tombol untuk melihat detail pada index.blade.php `<a href="/mahasiswa/detail{{ $mhs->id }}" class="badge badge-info">Detail</a>`
2	Buat route baru untuk detail untuk menjalankan detail pada mahasiswa controller `Route::get('/mahasiswa/detail{id}', 'MahasiswaController@detail');`

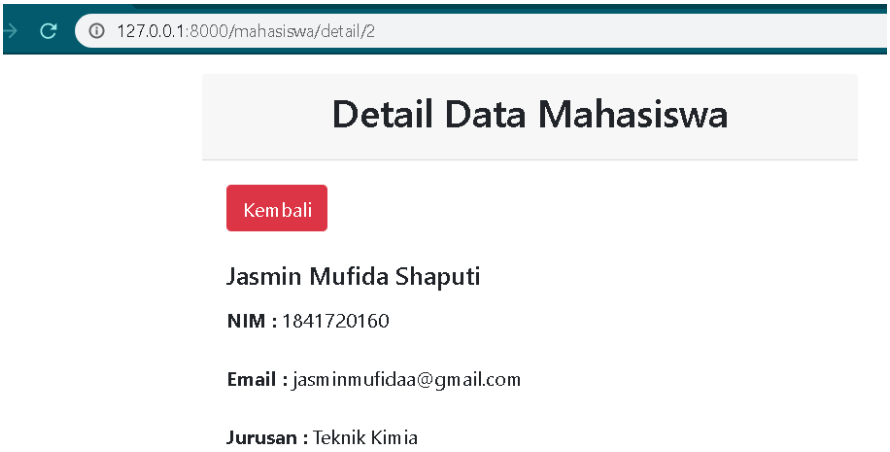
3	Buat method detail pada mahasiswacontroller
	`public function detail($id)`
	`{`
	`$mahasiswa = DB::table('mahasiswa')->where('id', $id)->get();`
	
	`return view('detail', ['mahasiswa' => $mahasiswa]);`
	`}`

4	Buat view detail.blade.php pada resources/views `@extends('master')` `@section('title','Detail Mahasiswa')` `@section('judul_halaman','Detail Data Mahasiswa')` `@section('konten')` `<a href="/" class="btn btn-danger">Kembali</a>` ` ` ` `

```
@foreach($mahasiswa as $mhs)
<h5 class="card-title">{{ $mhs->nama }}</h5>
<p class="card-text">
    <label for=""><b>NIM : </b></label>
    {{ $mhs->nim }}
</p>
<p class="card-text">
    <label for=""><b>Email : </b></label>
    {{ $mhs->email }}
</p>
<p class="card-text">
    <label for=""><b>Jurusan : </b></label>
    {{ $mhs->jurusan }}
</p>
@endforeach
@endsection
```

5 Jalankan kembali localhost:8000 dan klik detail pada suatu data





e. Mengubah Data (Update) dari Database

1	Tambahkan tombol untuk edit pada index.blade.php `<a href="/mahasiswa/edit/{{ $mhs->id }}" class="badge badge-warning">Edit</a>`
2	Buat routes baru pada web.php mahasiswa edit yang nantinya akan menjalankan fungsi controller mahasiswaController `Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');`

3	Buat edit function pada controller mahasiswa
	`public function edit($id)`
	`{`
	`$mahasiswa = DB::table('mahasiswa')->where('id', $id)->get();`
	
	`return view('edit', ['mahasiswa' => $mahasiswa]);`
	`}`

4	Buat view Edit blade php pada folder views

```
@extends('master')

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

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

@section('konten')
<a href="/" class="btn btn-danger">Kembali</a>
<br>
<br>
<form action="/mahasiswa/update" method="POST">
    {{ csrf_field() }}
@foreach($mahasiswa as $mhs)
    <input type="hidden" name="id" value="{{ $mhs->id }}">
    <div class="form-group">
        <label>Nama</label>
        <input type="text" class="form-control" value="{{ $mhs-
>nama }}" required="required" name="namamhs"><br>
    </div>
    <div class="form-group">
        <label>NIM</label>
        <input type="number" class="form-control" value="{{ $mhs-
>nim }}" required="required" name="nimmhs"><br>
    </div>
    <div class="form-group">
        <label>Email</label>
        <input type="email" class="form-control" value="{{ $mhs-
>email }}" required="required" name="emailmhs"><br>
    </div>
    <div class="form-group">
        <label>Jurusan</label>
```

```
        <input type="text" class="form-control" value="{{ $mhs-
>jurusan }}" required="required" name="jurusanmhs"><br>
    </div>
    <input type="submit" class="btn btn-primary float-
right" value="Update Data">
@endforeach
</form>
@endsection
```

- Form action mahasiswa update yang menunjukan routes mahasiswa update dimana data pada form tersebut akan dikirimkan pada fungsi update pada mahasiswa controller

5 Membuat routes update pada web.php

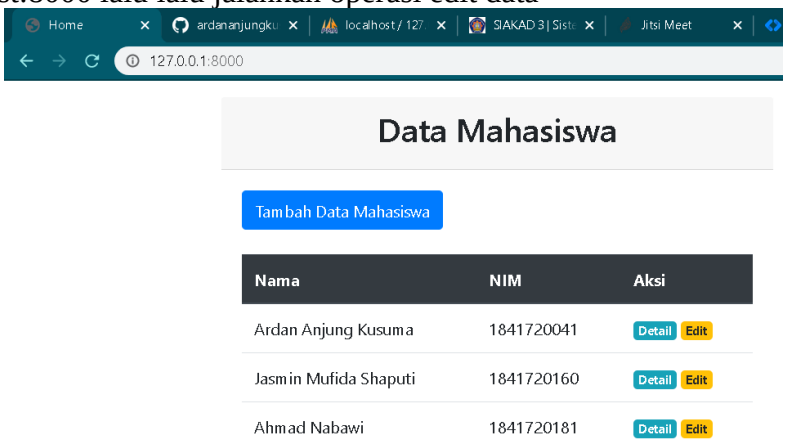
```
Route::post('/mahasiswa/update', 'MahasiswaController@update');
```

- Data pada form akan dikirimkan pada method update di controller mahasiswa

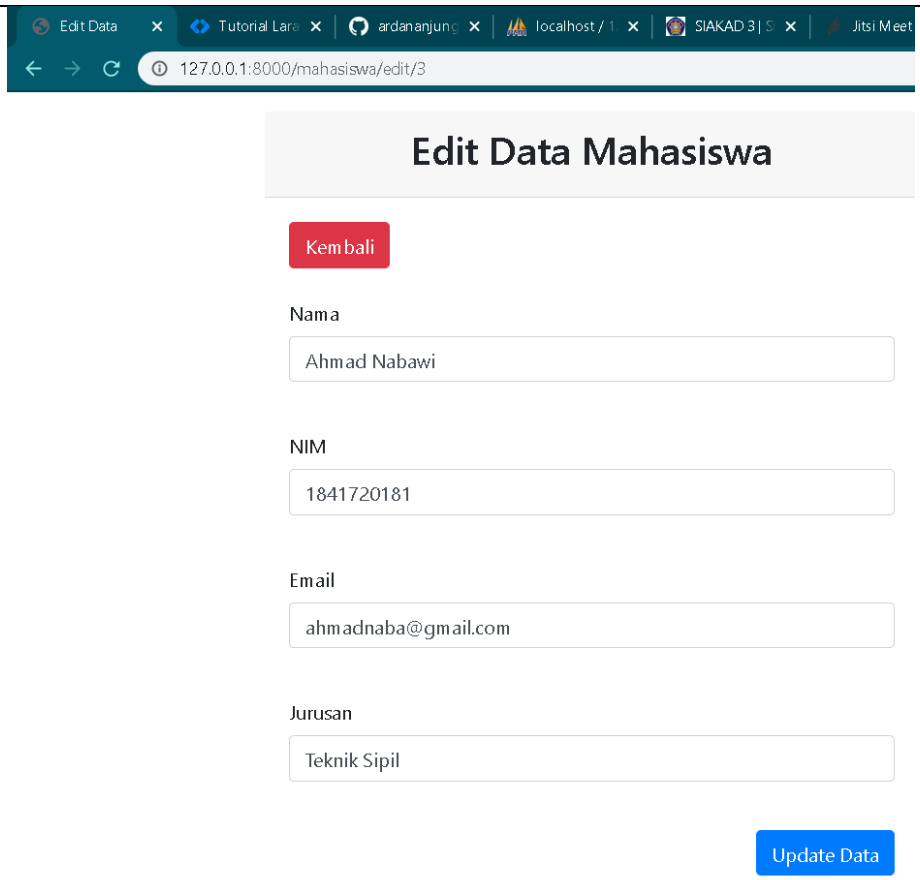
6 Membuat method update pada controller mahasiswa yang digunakan untuk **update** dan disimpan ke database

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

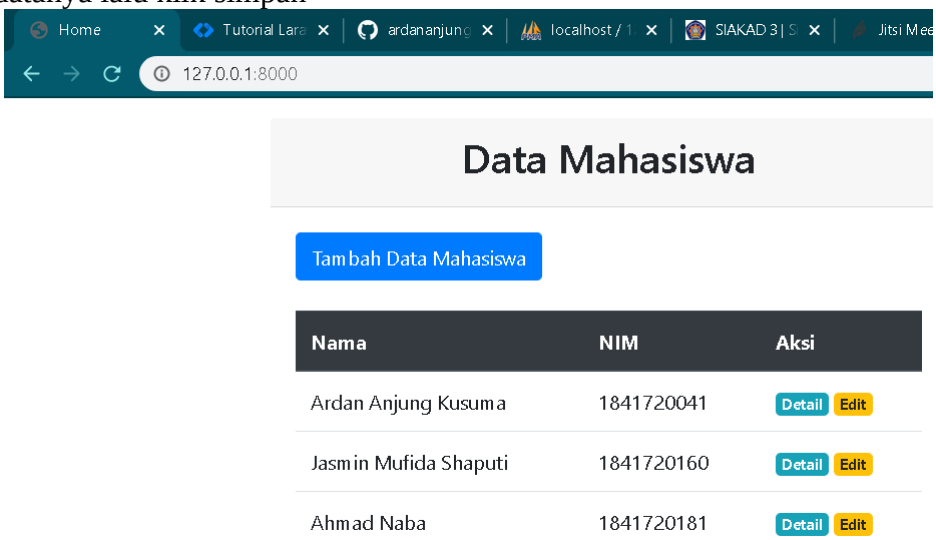
7 Jalankan localhost:8000 lalu lalu jalankan operasi edit data



Klik edit data ketiga, kita akan mengubah namanya



Kita ubah datanya lalu klik simpan



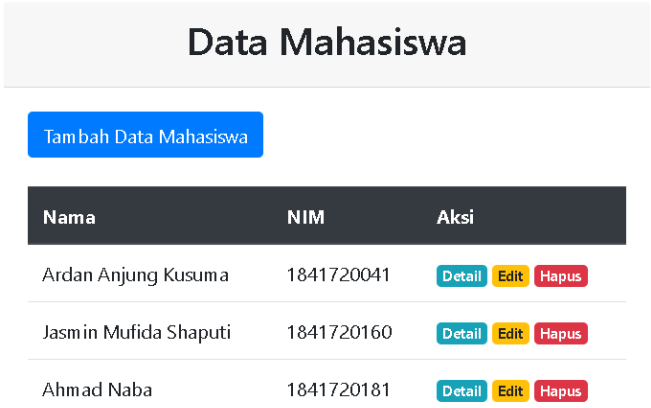
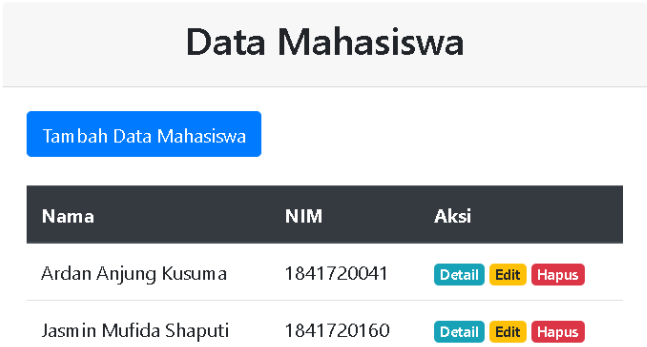
Data sukses terupdate menjadi Ahmad Naba yang awalnya Ahmad Nabawi

f. Menghapus Data (Delete) dari Database

1	Tambahkan button hapus pada index

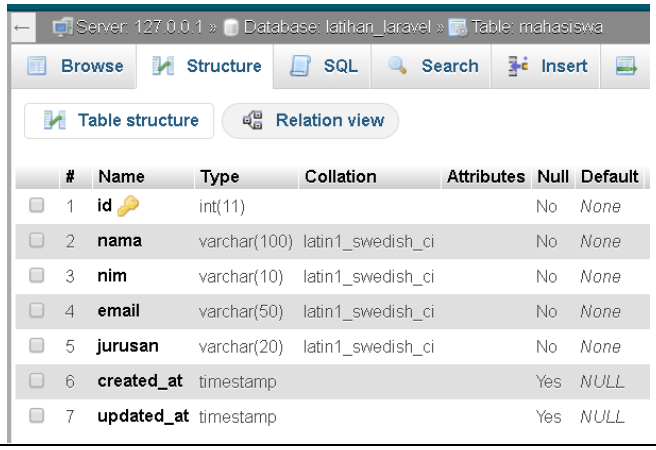
	`<a href="/mahasiswa/hapus/{{ $mhs->id }}" class="badge badge-danger" onclick="return confirm('Apakah anda ingin menghapus data ini?')">Hapus</a>`
2	Tambahkan routes web.php dengan function hapus `Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');`

3	Buat method hapus pada controller mahasiswa
	`public function hapus($id)`
	`{`
	`// hapus data mahasiswa by id`
	`DB::table('mahasiswa')->where('id', $id)->delete();`
	
	`return redirect('/');`
	`}`

4	Jalankan localhost:8000 dan pilih data yang ingin dihapus  

<u>Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent</u>

a. Konfigurasi Database

1	Konfigurasi .env DB_USERNAME=root DB_PASSWORD=
2	Lalu buka phpmyadmin 

b. Menampilkan Data dari Database

1	Lalu kita disini menambahkan routes web.php
	`Route::get('/', 'MahasiswaController@index');`
	`Route::get('/mahasiswa', 'MahasiswaController@index');`

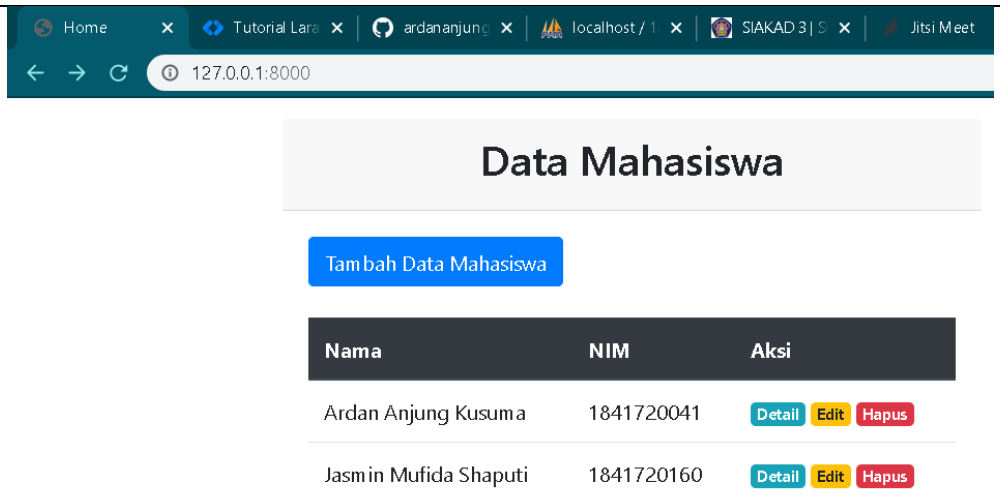
2	Membuat model
	`C:\xampp\htdocs\laravelapp>php artisan make:model Mahasiswa`
	`Model created successfully.`

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

4	Buat controller baru Mahasiswa
	`C:\xampp\htdocs\laravelapp>php artisan make:controller MahasiswaController`
	`Controller created successfully.`

5	Buat method index pada mahasiswa controller
	`<?php`
	
	`namespace App\Http\Controllers;`
	
	`use Illuminate\Http\Request;`
	`use App\Mahasiswa;`
	
	`class MahasiswaController extends Controller`
	`{`
	`  public function index()`
	`  {`
	`  $mahasiswa = Mahasiswa::all();`
	`  return view('index', ['mahasiswa' => $mahasiswa]);`
	`  }`
	`}`
	- Tambahkan use App\Mahasiswa untuk menggunakan model mahasiswa
	- Untuk mengambil semua data yang disimpan lihat variabel $mahasiswa
	- Lalu kita mereturn data

6	Kita membuat view sama seperti yang awal tadi template blade
7	Jalankan php artisan serve



c. Memasukkan Data (Create) ke Database

1. Buat route baru untuk tambah data

```
Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');
```

2. Buat method tambah pada controller mahassiwa

```
public function tambah()
    {
        return view('tambah');
    }
```

3. Buatlah tambah blade yang isinya form sama seperti tadi dicopas saja

4. Buat route simpan pada mahasiswa

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

5. Buat function simpan pada controllermahasiswa untuk menyimpan ke database

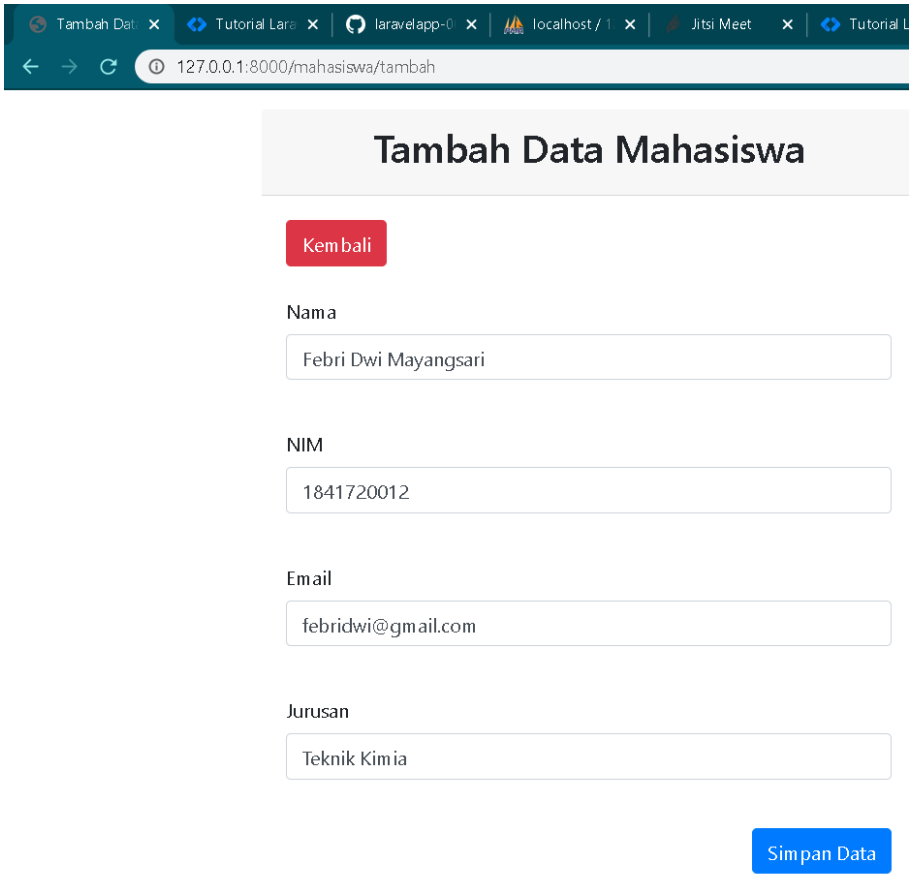
```
public function simpan(Request $request)
    {
        // insert data ke table mahasiswa
        Mahasiswa::create([
            'nama' => $request->namamhs,
            'nim' => $request->nimmhs,
            'email' => $request->emailmhs,
            'jurusan' => $request->jurusanmhs
        ]);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

6. Lalu kita ke model dan tambahkan kode dibawah ini

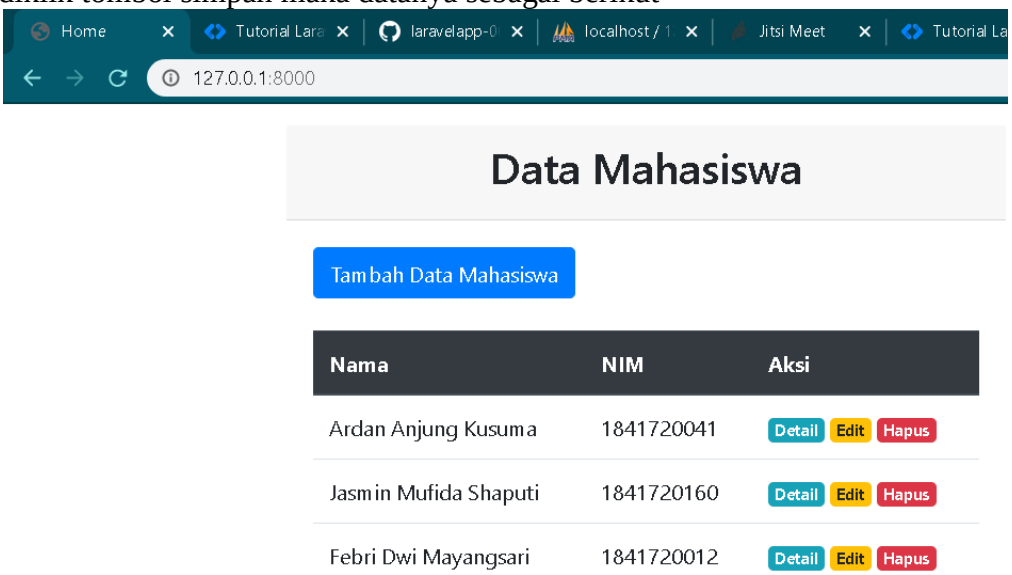
```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Mahasiswa extends Model
```

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

7 Lalu kita coba tambahkan data mahasiswa



Setelah diklik tombol simpan maka datanya sebagai berikut

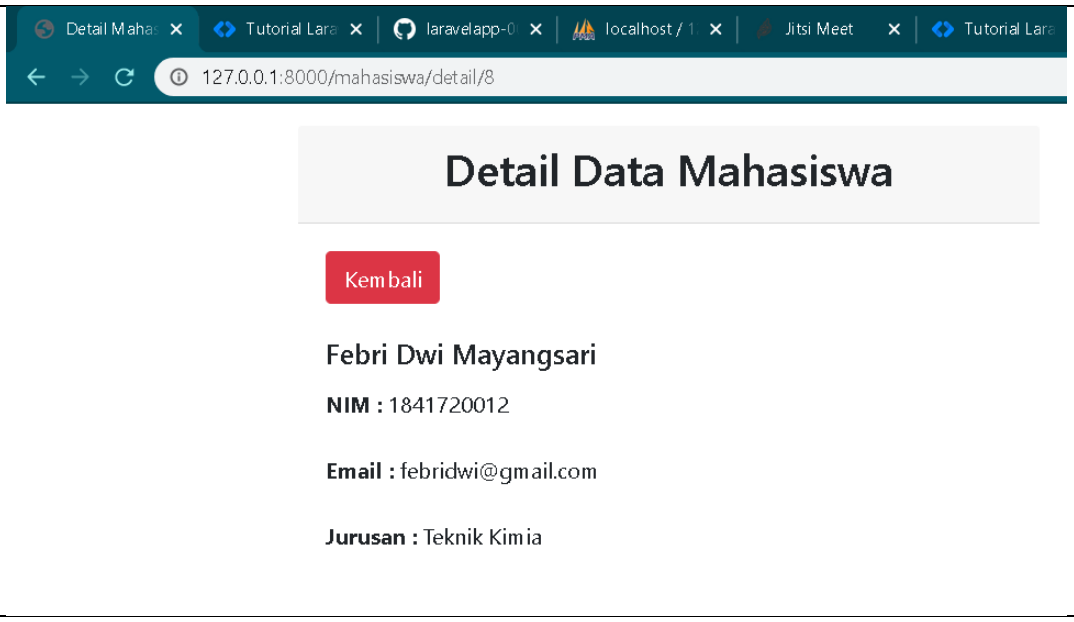


d. Melihat Detail Data dari Database

1	Buat route baru pada routes detail `Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');`

2	Buat detail pada mahasiswa controller yang akan menampilkan view detail
	`public function detail($id)`
	`{`
	`$mahasiswa = Mahasiswa::find($id);`
	`return view('detail', ['mahasiswa' => $mahasiswa]);`
	`}`

4	Kemudian buat view detail `@extends('master')` `@section('title','Detail Mahasiswa')` `@section('judul_halaman','Detail Data Mahasiswa')` `@section('konten')` `<a href="/" class="btn btn-danger">Kembali</a>` ` ` `  `@foreach($mahasiswa as $mhs)` `<h5 class="card-title">{{ $mhs->nama }}</h5>` ` ` `<label for="">NIM : </label>` `{{ $mhs->nim }}` ` ` ` ` `<label for="">Email : </label>` `{{ $mhs->email }}` ` ` ` ` `<label for="">Jurusan : </label>` `{{ $mhs->jurusan }}` ` ` `@endforeach` `@endsection`
5	Coba fitur detail



e. Mengubah Data (Update) dari Database

1	Membaut routes baru edit yang akan menjalankan fungsi edit pada mahasiswacontroller `Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');`

2	Buat method edit pada mahasiswa controller yang akan menampilkan view edit
	`public function edit($id)`
	`{`
	`$mahasiswa = Mahasiswa::find($id);`
	
	`return view('edit', ['mahasiswa' => $mahasiswa]);`
	`}`

3	Buat view edit `@extends('master')` `@section('title','Edit Data')` `@section('judul_halaman','Edit Data Mahasiswa')` `@section('konten')` `<a href="/" class="btn btn-danger">Kembali</a>` ` ` ` `<form action="/mahasiswa/update" method="POST">` `{{ csrf_field() }}` `<input type="hidden" name="id" value="{{ $mahasiswa->id }}">` ` ` `<label>Nama</label>` `<input type="text" class="form-control" value="{{ $mahasiswa->nama }}" required="required" name="namamhs"> `

```
    </div>
    <div class="form-group">
        <label>NIM</label>
        <input type="number" class="form-control" value="{{ $mahasiswa-
>nim }}" required="required" name="nimmhs"><br>
    </div>
    <div class="form-group">
        <label>Email</label>
        <input type="email" class="form-control" value="{{ $mahasiswa-
>email }}" required="required" name="emailmhs"><br>
    </div>
    <div class="form-group">
        <label>Jurusan</label>
        <input type="text" class="form-control" value="{{ $mahasiswa-
>jurusan }}" required="required" name="jurusanmhs"><br>
    </div>
    <input type="submit" class="btn btn-primary float-
right" value="Update Data">
</form>
@endsection
```

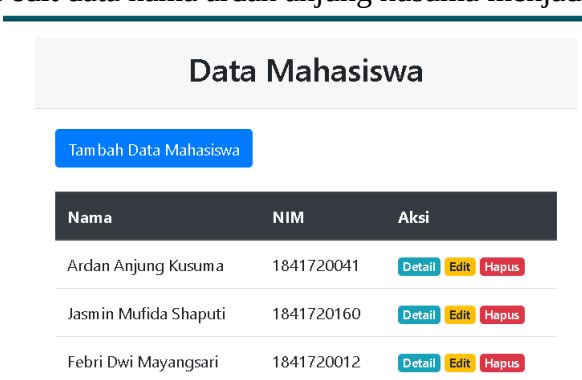
4 Membuat routes baru untuk update data

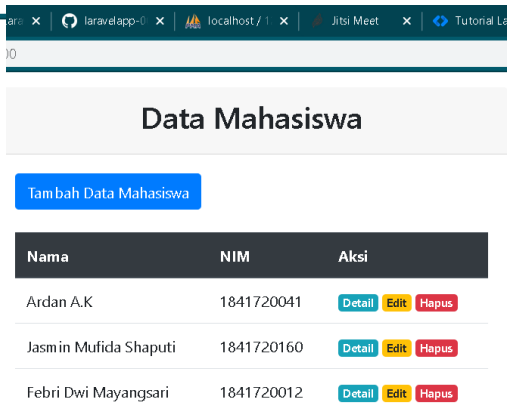
```
Route::post('/mahasiswa/update', 'MahasiswaController@update');
```

5 Buat method update pada mahasiwa controller untuk menyimpan data yang nantinya akan diubah ke database

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

6 Coba edit data nama ardan anjung kusuma menjadi ardan a.k



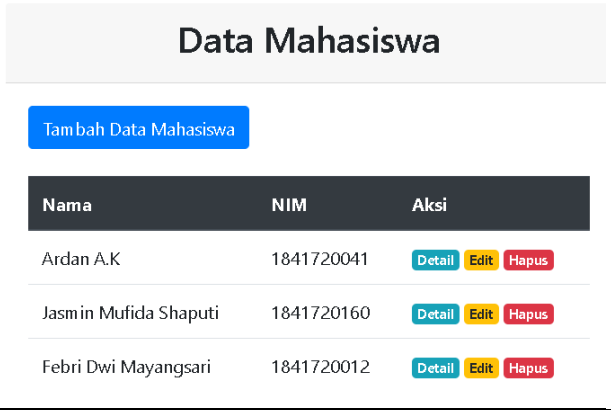
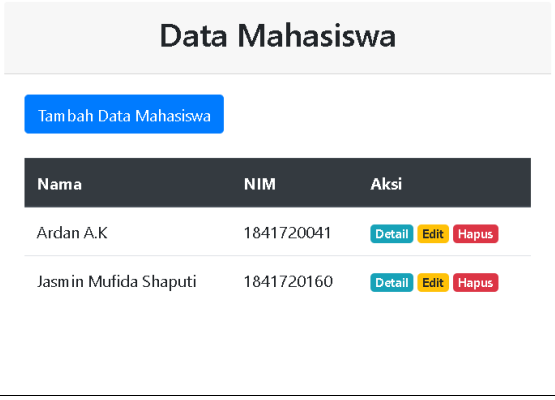


f. Menghapus Data (Delete) dari Database

1 Buatlah routes baru untuk hapus

```
Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');
```

2 Buat method hapus pada controller mahasiswa

```
public function hapus($id)
    {
        // hapus data mahasiswa by id
        $mahasiswa = Mahasiswa::find($id);
        $mahasiswa->delete();
        return redirect('/');
    }
```

3 Testing Hapus data Febri Dwi Mayangsari