



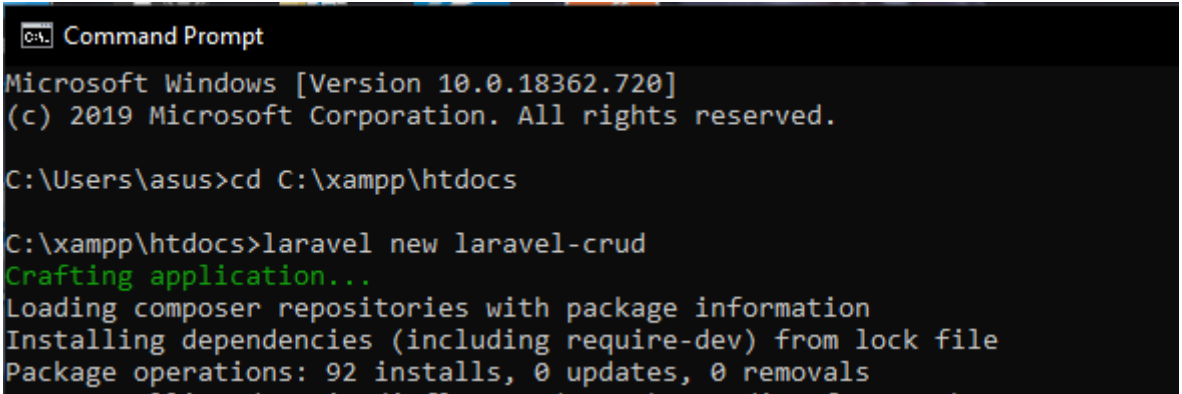
Nama: Darin Zahira Aflah

Kelas: TI2A

NIM: 1841720033

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

a. Konfigurasi Database dan Pembuatan Tabel di MySQL

Langkah	Keterangan
1	<p>Buatlah project Laravel baru dengan nama laravel-crud. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud</pre>  <pre>Microsoft Windows [Version 10.0.18362.720] (c) 2019 Microsoft Corporation. All rights reserved. C:\Users\asus>cd C:\xampp\htdocs C:\xampp\htdocs>laravel new laravel-crud Crafting application... Loading composer repositories with package information Installing dependencies (including require-dev) from lock file Package operations: 92 installs, 0 updates, 0 removals</pre>
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.</p>

```

.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:FAOWJfHcs6umTG2CSJrgwtX3w+Zjc1WY/11p/z1MpUc=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=localhost
11 DB_PORT=3306
12 DB_DATABASE=latihan_laravel
13 DB_USERNAME=root
14 DB_PASSWORD=
15

```

Keterangan:

- Nama database yang akan digunakan adalah latihan_laravel dengan username root.

3

Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database latihan_laravel.

Server: 127.0.0.1 » Database: latihan_laravel » Table: mahasiswa

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	nim	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
4	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
5	jurusan	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More

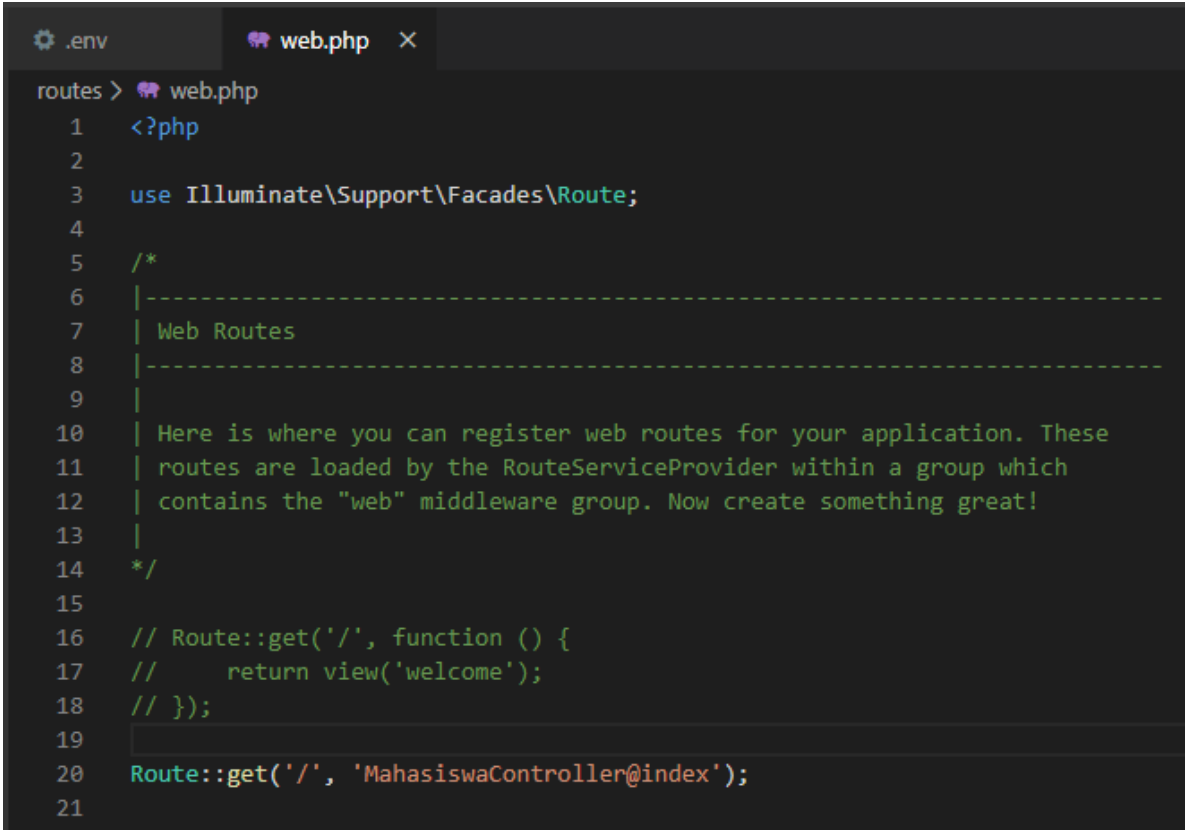
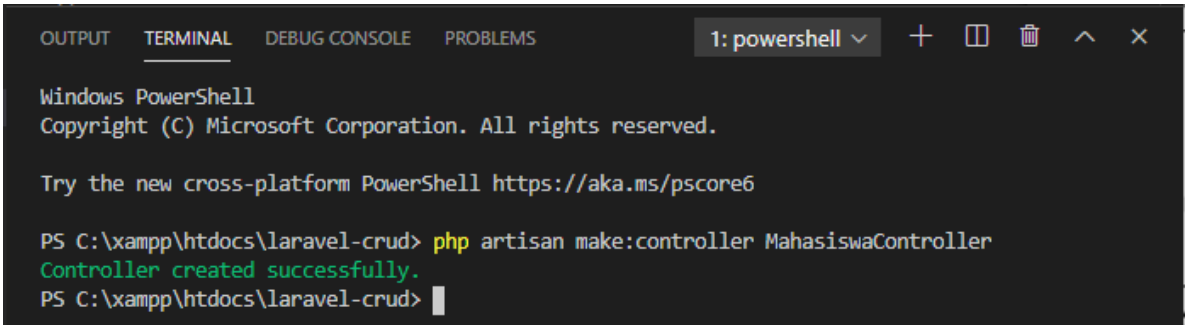
4

Isilah beberapa data pada tabel mahasiswa tersebut.

+ Options

				id	nama	nim	email	jurusan
<input type="checkbox"/>	Edit	Copy	Delete	1	Alsa	2002002001	alsa@gmail.com	teknik informatika
<input type="checkbox"/>	Edit	Copy	Delete	2	Rafi	2001001002	rafi@gmail.com	teknik elektro

b. Menampilkan Data dari Database

Langkah	Keterangan
1	<p>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.</p>  <pre> 1 <?php 2 3 use Illuminate\Support\Facades\Route; 4 5 /* 6 ----- 7 Web Routes 8 ----- 9 10 Here is where you can register web routes for your application. These 11 routes are loaded by the RouteServiceProvider within a group which 12 contains the "web" middleware group. Now create something great! 13 14 */ 15 16 // Route::get('/', function () { 17 // return view('welcome'); 18 // }); 19 20 Route::get('/', 'MahasiswaController@index'); 21 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Ketika route ('/') diakses, akan dijalankan method index pada controller bernama MahasiswaController.
2	<p>Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan cd laravel-crud php artisan make:controller MahasiswaController</p>  <pre> OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 1: powershell Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved. Try the new cross-platform PowerShell https://aka.ms/pscore6 PS C:\xampp\htdocs\laravel-crud> php artisan make:controller MahasiswaController Controller created successfully. PS C:\xampp\htdocs\laravel-crud> </pre>
3	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p>

```

web.php  MahasiswaController.php X
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         $mahasiswa = DB::table('mahasiswa')->get();
13         return view('index', ['mahasiswa' => $mahasiswa]);
14     }
15 }
16

```

Keterangan:

- Tambahkan 'use Illuminate\Support\Facades\DB;' (line 6) agar query builder dapat digunakan
- Line 13 untuk mengambil data dari tabel mahasiswa menggunakan query builder laravel dan akan disimpan di variabel \$mahasiswa
- Line 16 : data akan dikirim ke blade view bernama index

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.

```

web.php  MahasiswaController.php  master.blade.php X
resources > views > master.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
7      <title> @yield('title') </title>
8  </head>
9  <body>
10     <div class="container">
11         <div class="row mt-3">
12             <div class="col-md-6">
13                 <div class="card">
14                     <div class="card-header text-center">
15                         <h2> @yield('judul_halaman') </h2>
16                     </div>
17                     <div class="card-body">
18                         @yield('konten')
19                     </div>
20                 </div>
21             </div>
22         </div>
23     </div>
24 </body>
25 </html>

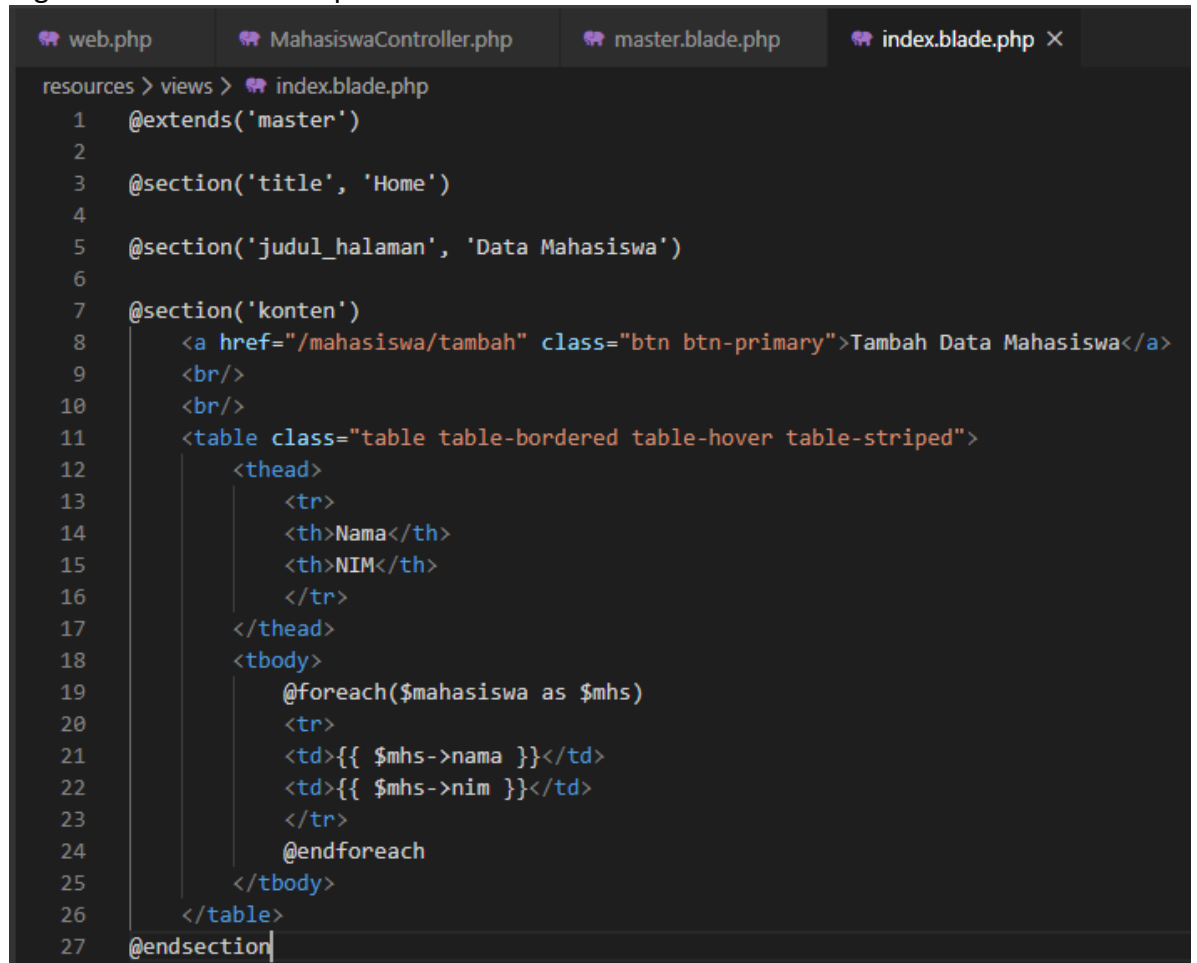
```

Keterangan:

- Fungsi @yield pada line 6(title), 15(judul_halaman), dan 19(konten) berfungsi sebagai penanda bagian-bagian pada master blade. Nantinya bagian @yield ini akan diisi sesuai dengan halaman view yang menerapkan master.blade.php

5

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



```

resources > views > index.blade.php
1  @extends('master')
2
3  @section('title', 'Home')
4
5  @section('judul_halaman', 'Data Mahasiswa')
6
7  @section('konten')
8      <a href="/mahasiswa/tambah" class="btn btn-primary">Tambah Data Mahasiswa</a>
9      <br/>
10     <br/>
11     <table class="table table-bordered table-hover table-striped">
12         <thead>
13             <tr>
14                 <th>Nama</th>
15                 <th>NIM</th>
16             </tr>
17         </thead>
18         <tbody>
19             @foreach($mahasiswa as $mhs)
20                 <tr>
21                     <td>{{ $mhs->nama }}</td>
22                     <td>{{ $mhs->nim }}</td>
23                 </tr>
24             @endforeach
25         </tbody>
26     </table>
27 @endsection

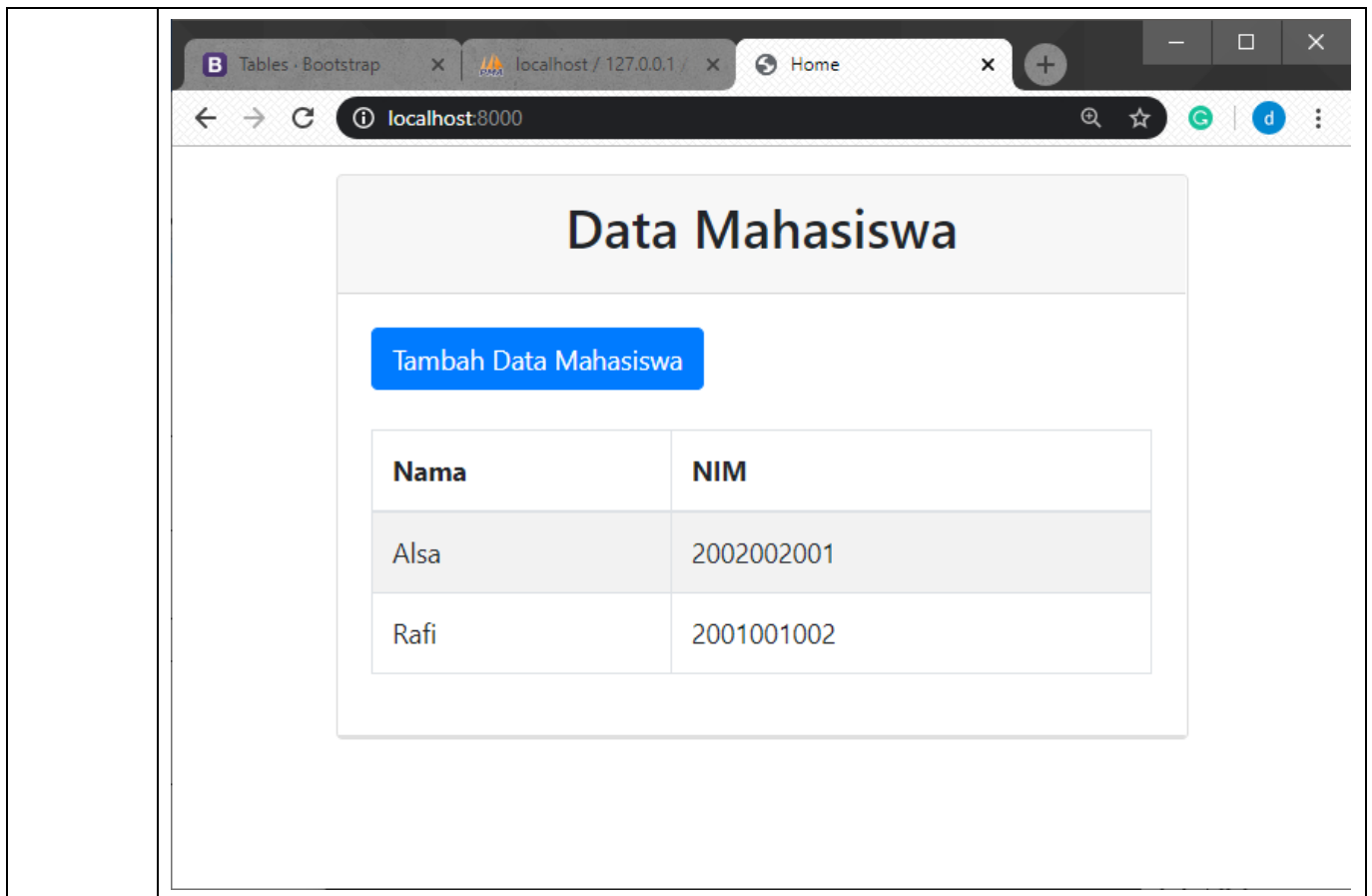
```

Keterangan:

- Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan blade lain yaitu master.blade.php
- Line 4 : @section('title', 'Home') berarti bahwa file index mengisi @yield('title') pada master dengan 'Home'
- Line 7 : @section('judul_halaman', 'Data Mahasiswa') berarti bahwa file index mengisi @yield('judul_halaman') pada master dengan 'Home'
- Line 10-30 : mengisi yield(@konten), karena terdapat banyak baris pada diawali dengan @section('konten') dan diakhiri dengan @endsection
- Line 24-25 menampilkan data mahasiswa dengan kolom nama dan nim

6

Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud php artisan serve Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut



c. Memasukkan Data (Create) ke Database

Langkah	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController</p> <pre> 19 20 Route::get('/', 'MahasiswaController@index'); 21 22 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 23 </pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> 15 16 public function tambah() 17 { 18 return view('tambah'); 19 } 20 </pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. View tambah juga mengaplikasikan master.blade.php</p>

```

web.php  MahasiswaController.php  tambah.blade.php X
resources > views > tambah.blade.php
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 for="namamhs">Nama</label>
15             <input type="text" class="form-control" required="required" name="namamhs"><br/>
16         </div>
17         <div class="form-group">
18             <label for="nimamhs">NIM</label>
19             <input type="number" class="form-control" required="required" name="nimamhs"><br/>
20         </div>
21         <div class="form-group">
22             <label for="emailmhs">E-mail</label>
23             <input type="email" class="form-control" required="required" name="emailmhs"><br/>
24         </div>
25         <div class="form-group">
26             <label for="jurusanmhs">Jurusan</label>
27             <input type="text" class="form-control" required="required" name="jurusanmhs"><br/>
28         </div>
29         <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button>
30     </form>
31 @endsection

```

Keterangan:

- Line 13-32 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 13 terdapat action="/mahasiswa/simpan" yang menunjukkan routes /mahasiswa/simpan dimana data pada form tersebut akan dikirimkan ke fungsi simpan pada controller MahasiswaController
- Line 14 terdapat {{ csrf_field() }} yang digunakan untuk menerapkan fitur laravel yaitu csrf protection. csrf protection adalah fitur keamanan untuk mencegah penginputan dari luar aplikasi, csrf akan men-generate kode token otomatis yang dibuat dalam bentuk form hidden.

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

```

20 Route::get('/', 'MahasiswaController@index');
21
22 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');
23
24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');
25

```

Keterangan:

- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php

- 5 Buat method simpan pada MahasiswaController untuk menyimpan data ke database.

```

21     public function simpan(Request $request)
22     {
23         DB::table('mahasiswa')->insert([
24             'nama' => $request->namamhs,
25             'nim' => $request->nimmhs,
26             'email' => $request->emailmhs,
27             'jurusan' => $request->jurusanmhs,
28         ]);
29         return redirect('/mahasiswa');
30     }
31

```

Keterangan:

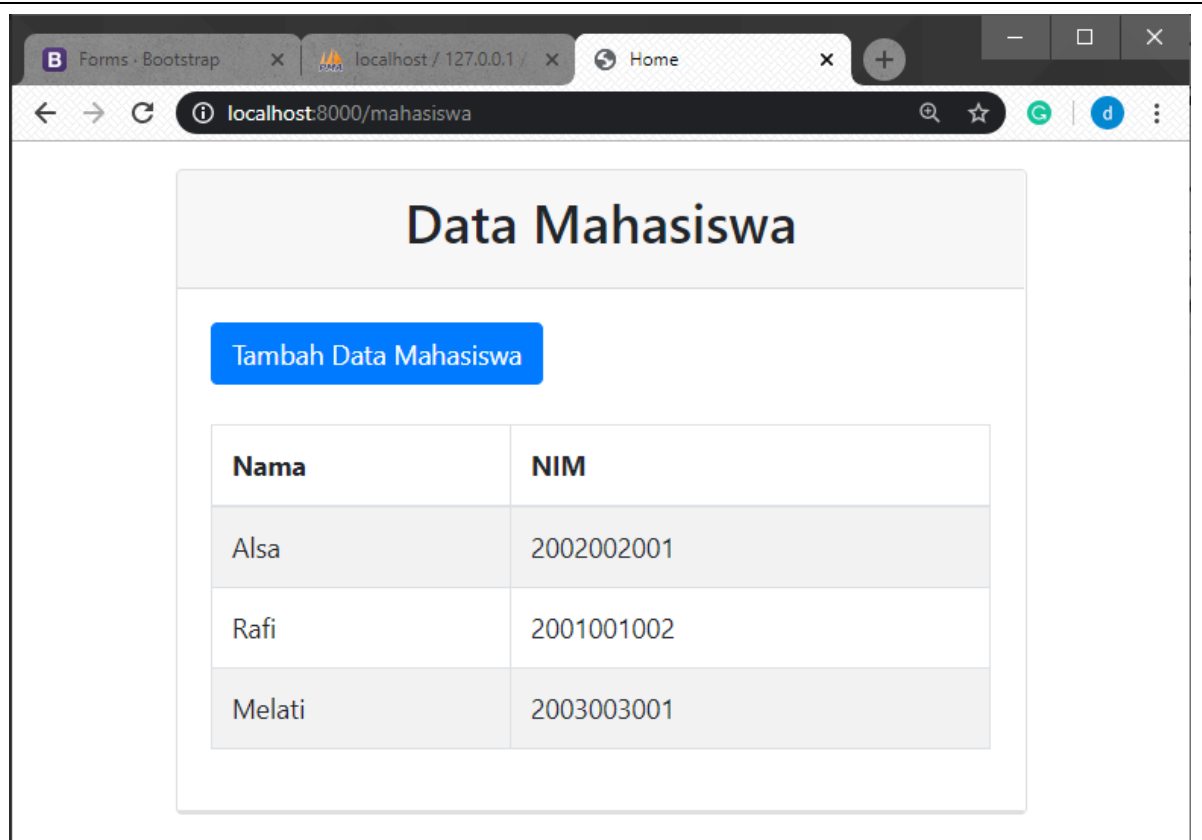
- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 28-33 merupakan query builder untuk insert data ke tabel mahasiswa

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.

The screenshot shows a web browser window with the address bar displaying 'localhost:8000/mahasiswa/tambah'. The page title is 'Tambah Data Mahasiswa'. The form contains a red 'Kembali' button at the top left. Below it are four input fields labeled 'Nama', 'NIM', 'E-mail', and 'Jurusan'. At the bottom right of the form is a blue 'Tambah Data' button.

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



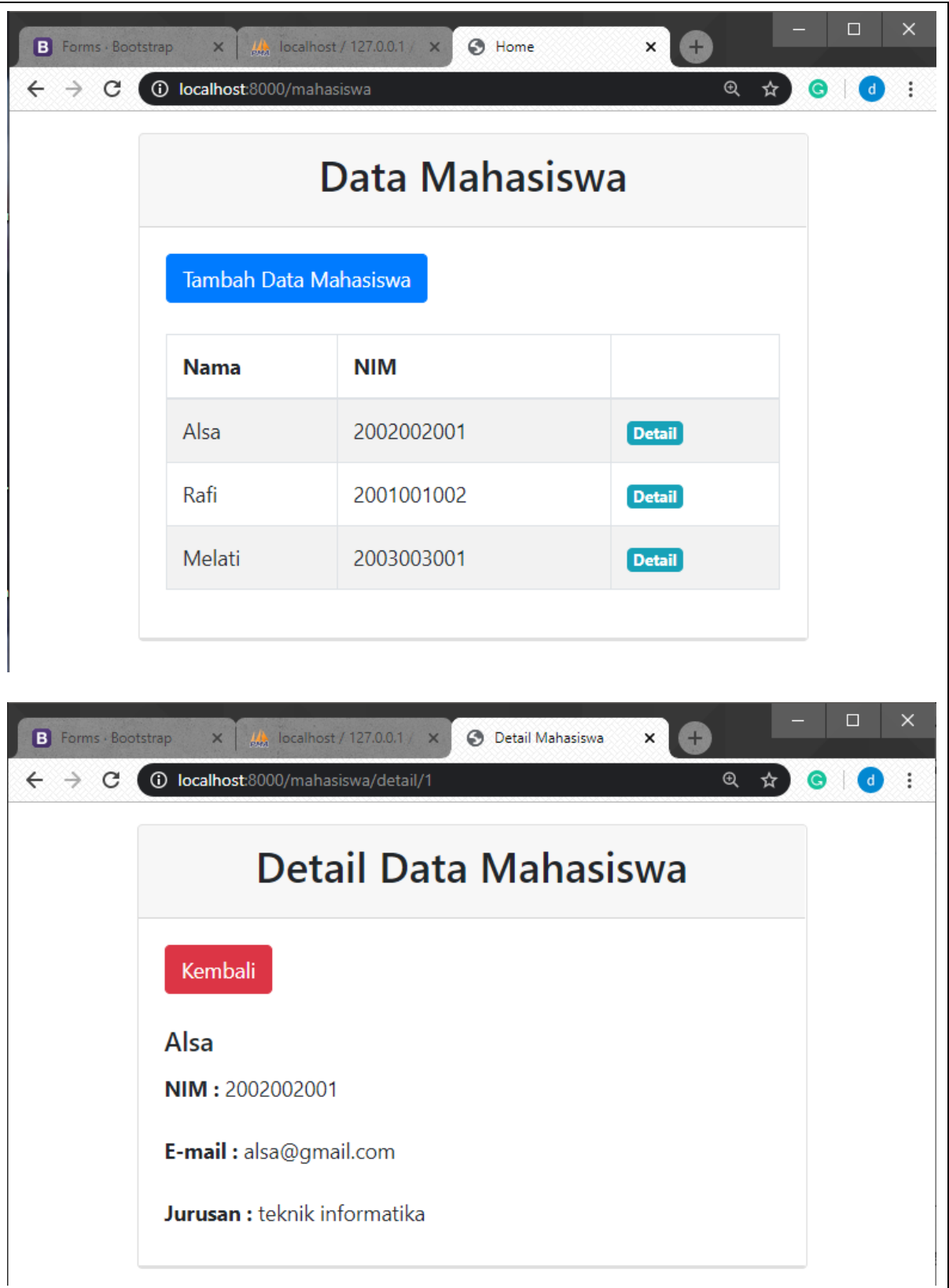
Kita dapat mencoba menambahkan beberapa data lagi agar jumlah data lebih banyak.

d. Melihat Detail Data dari Database

Langkah	Keterangan
1	Buatlah tombol untuk melihat detail data (Line 28) pada file index.blade.php di folder resources/views

	<div> web.php MahasiswaController.php index.blade.php X </div> <pre> resources > views > index.blade.php 1 @extends('master') 2 3 @section('title', 'Home') 4 5 @section('judul_halaman', 'Data Mahasiswa') 6 7 @section('konten') 8 Tambah Data Mahasiswa 9
 10
 11 <table class="table table-bordered table-hover table-striped"> 12 <thead> 13 <tr> 14 <th>Nama</th> 15 <th>NIM</th> 16 </tr> 17 </thead> 18 <tbody> 19 @foreach(\$mahasiswa as \$mhs) 20 <tr> 21 <td>{{ \$mhs->nama }}</td> 22 <td>{{ \$mhs->nim }}</td> 23 <td> 24 id }}" class="badge badge-info">Detail 25 </td> 26 </tr> 27 @endforeach 28 </tbody> 29 </table> 30 @endsection </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre> 19 20 Route::get('/mahasiswa', 'MahasiswaController@index'); 21 22 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 23 24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 25 26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 27 28 </pre>
3	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p> <pre> 33 public function detail(\$id) 34 { 35 \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); 36 return view('detail', ['mahasiswa' => \$mahasiswa]); 37 } 38 </pre>
4	<p>Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php.</p>

	<div data-bbox="300 107 1484 1070"> <div> web.php MahasiswaController.php detail.blade.php X </div> <div>resources > views > detail.blade.php</div> <pre> 1 @extends('master') 2 3 @section('title', 'Detail Mahasiswa') 4 5 @section('judul_halaman', 'Detail Data Mahasiswa') 6 7 @section('konten') 8 Kembali 9 10
 11
 12 13 @foreach(\$mahasiswa as \$mhs) 14 <h5 class="card-title"> {{ \$mhs->nama }} </h5> 15 <p class="card-text"> 16 <label for=""> NIM : </label> {{ \$mhs->nim }} 17 </p> 18 <p class="card-text"> 19 <label for=""> E-mail : </label> {{ \$mhs->email }} 20 </p> 21 <p class="card-text"> 22 <label for=""> Jurusan : </label> {{ \$mhs->jurusan }} 23 </p> 24 @endforeach 25 @endsection </pre> </div> <p>Keterangan:</p> <ul style="list-style-type: none"> - Line 16-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan
5	<p>Jalankan localhost:8000 dan pilih tombol ‘Detail’ di suatu data yang ingin kita lihat detailnya.</p>



e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views

	<pre> 24 <td> 25 id }}" class="badge badge-info">Detail 26 id }}" class="badge badge-warning">Edit 27 </td> </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre> 19 20 Route::get('/mahasiswa', 'MahasiswaController@index'); 21 22 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 23 24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 25 26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 27 28 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 29 30 </pre>
3	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre> 38 39 public function edit(\$id) 40 { 41 \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); 42 return view('edit', ['mahasiswa' => \$mahasiswa]); 43 } 44 </pre>
4	<p>Kemudian buatlah view edit.blade.php yang berisi form untuk mengubah data pada folder resources/views. View edit juga mengaplikasikan master.blade.php.</p> <pre> 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 Kembali 9
 10
 11 @foreach(\$mahasiswa as \$mhs) 12 <form action="/mahasiswa/update" method="post"> 13 {{ csrf_field() }} 14 <input type="hidden" name="id" value="{{ \$mhs->id }}">
 15 <div class="form-group"> 16 <label for="namamhs">Nama</label> 17 <input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mhs->nama }}">
 18 </div> 19 <div class="form-group"> 20 <label for="nimmhs">NIM</label> 21 <input type="number" class="form-control" required="required" name="nimmhs" value="{{ \$mhs->nim }}">
 22 </div> 23 <div class="form-group"> 24 <label for="emailmhs">E-mail</label> 25 <input type="email" class="form-control" required="required" name="emailmhs" value="{{ \$mhs->email }}">
 26 </div> 27 <div class="form-group"> 28 <label for="jurusanmhs">Jurusan</label> 29 <input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mhs->jurusan }}">
 30 </div> 31 <button type="submit" name="tambah" class="btn btn-primary float-right">Simpan Data</button> 32 </form> 33 @endforeach 34 @endsection </pre> <p>Keterangan:</p>

	<p>- Line 14-36 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan</p> <p>- Line 15 terdapat action="/mahasiswa/update" yang menunjukkan routes /mahasiswa/update dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController</p>
5	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 19 20 Route::get('/mahasiswa', 'MahasiswaController@index'); 21 22 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 23 24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 25 26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 27 28 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 29 30 Route::post('/mahasiswa/update', 'MahasiswaController@update'); 31 </pre> <p>Keterangan:</p> <p>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php</p>
6	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p> <pre> 45 public function update(Request \$request) 46 { 47 DB::table('mahasiswa')->where('id',\$request->id)->update([48 'nama' => \$request->namamhs, 49 'nim' => \$request->nimmhs, 50 'email' => \$request->emailmhs, 51 'jurusan' => \$request->jurusanmhs, 52]); 53 return redirect('/mahasiswa'); 54 } 55 56 </pre> <p>Keterangan:</p> <p>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</p> <p>- Line 58-63 merupakan query builder untuk update data ke tabel mahasiswa</p>
7	<p>Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru.</p>

localhost / 127.0.0.1 / latihan_lat... x Home x +

localhost:8000/mahasiswa

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
Alsa	2002002001	Detail Edit
Rafi	2001001002	Detail Edit
Melati	2003003001	Detail Edit

Klik edit di data pertama, kita akan mengubah namanya.

localhost / 127.0.0.1 / latihan_jar... x Edit Data x +

localhost:8000/mahasiswa/edit/1

Edit Data Mahasiswa

[Kembali](#)

Nama

Alsa Putri

NIM

2002002001

E-mail

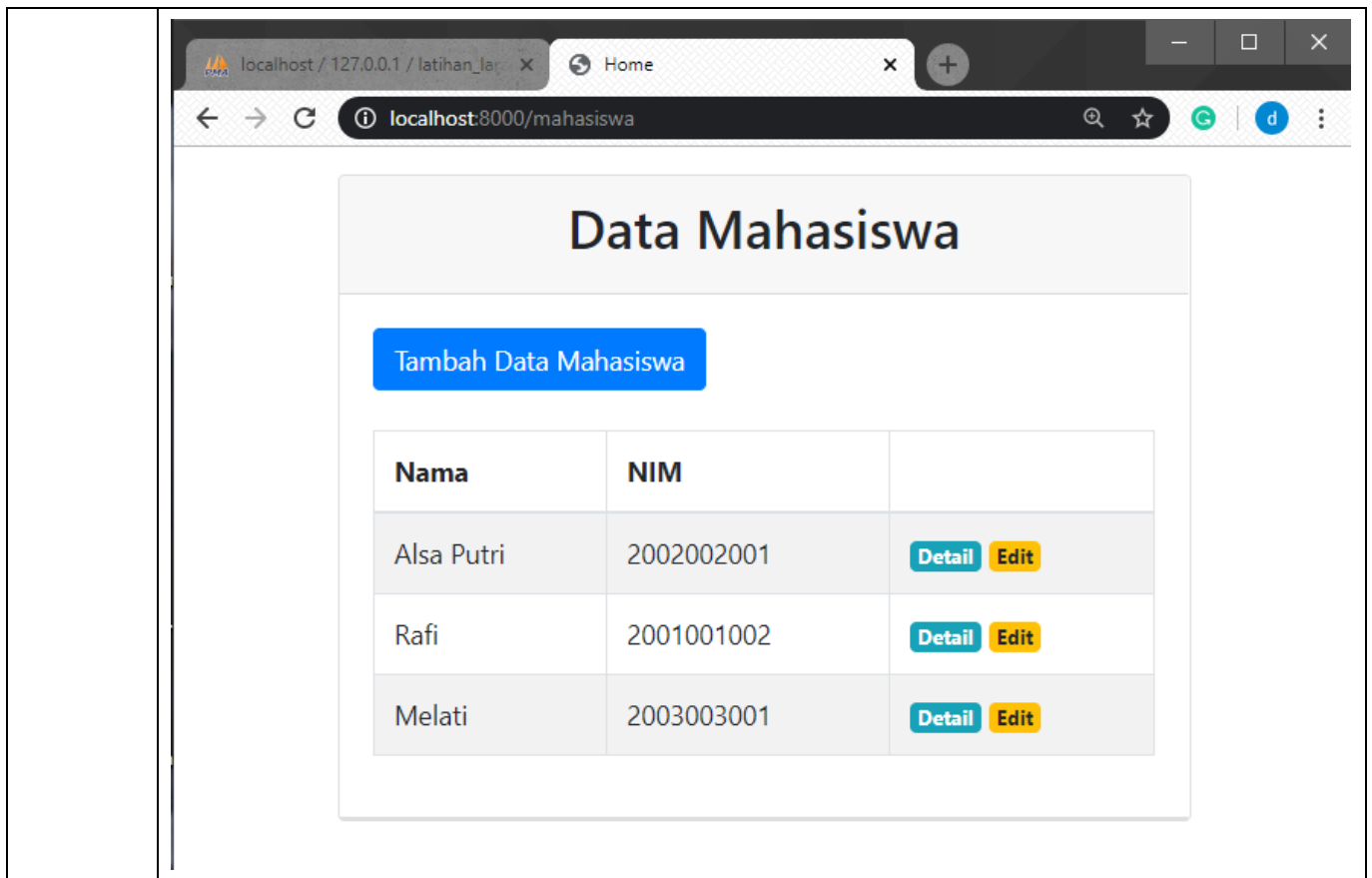
alsa@gmail.com

Jurusan

teknik informatika

[Simpan Data](#)

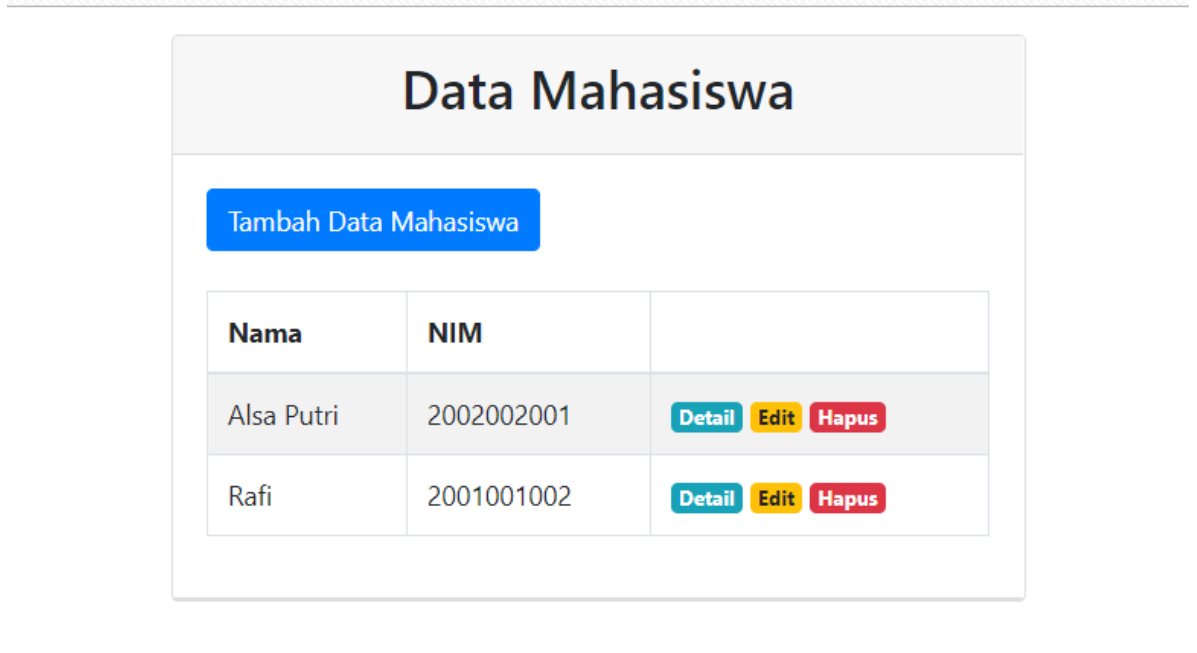
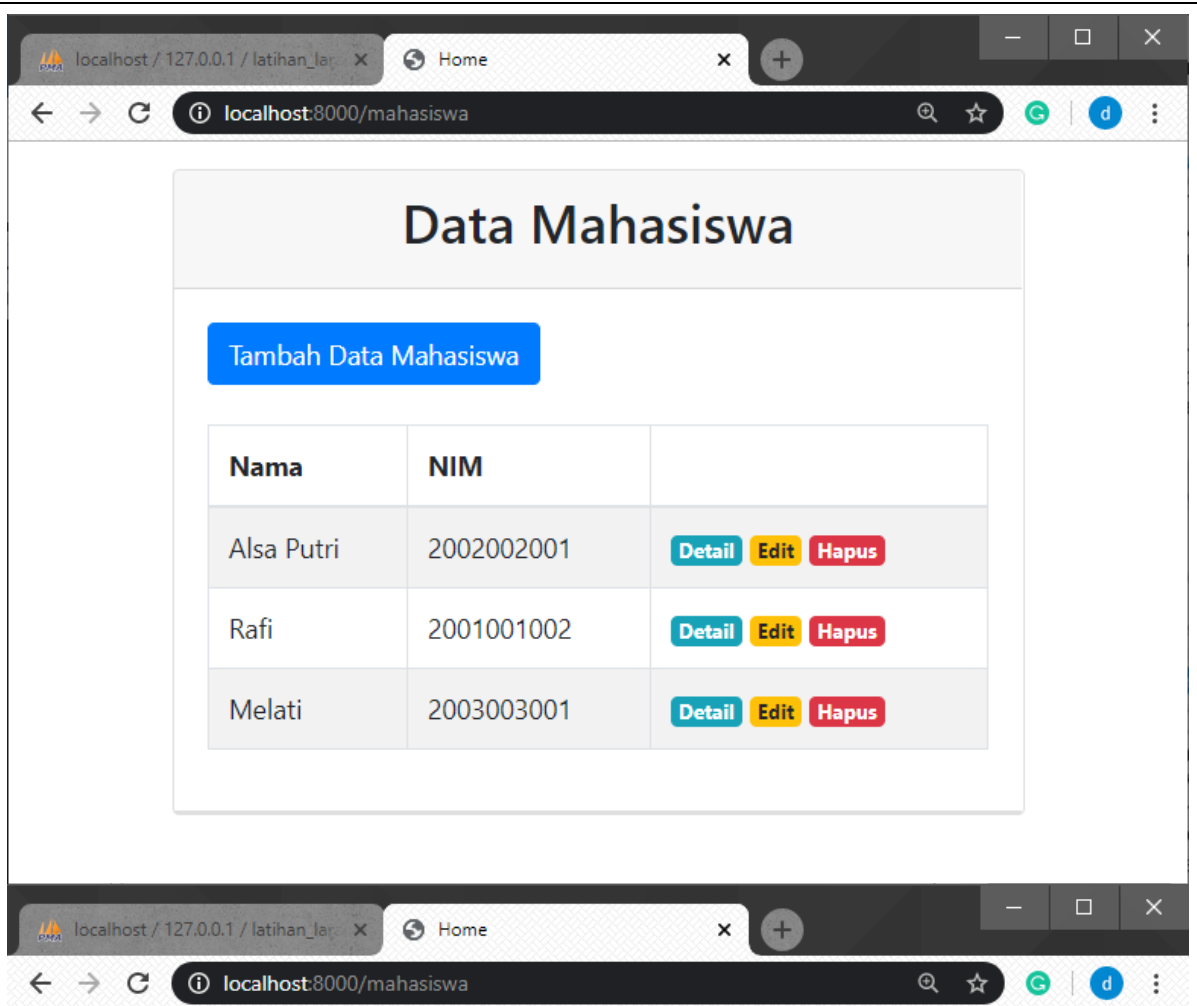
Setelah kita klik simpan Data, maka data pertama akan berubah.



f. Menghapus Data (Delete) dari Database

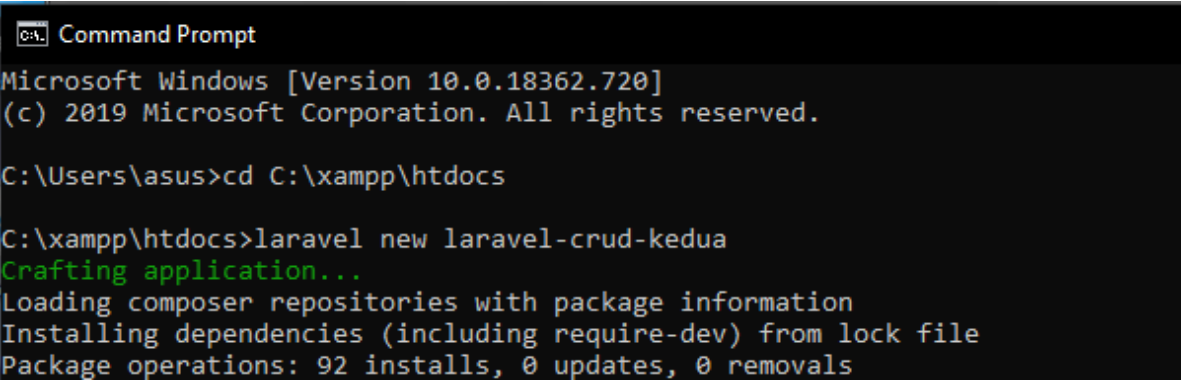
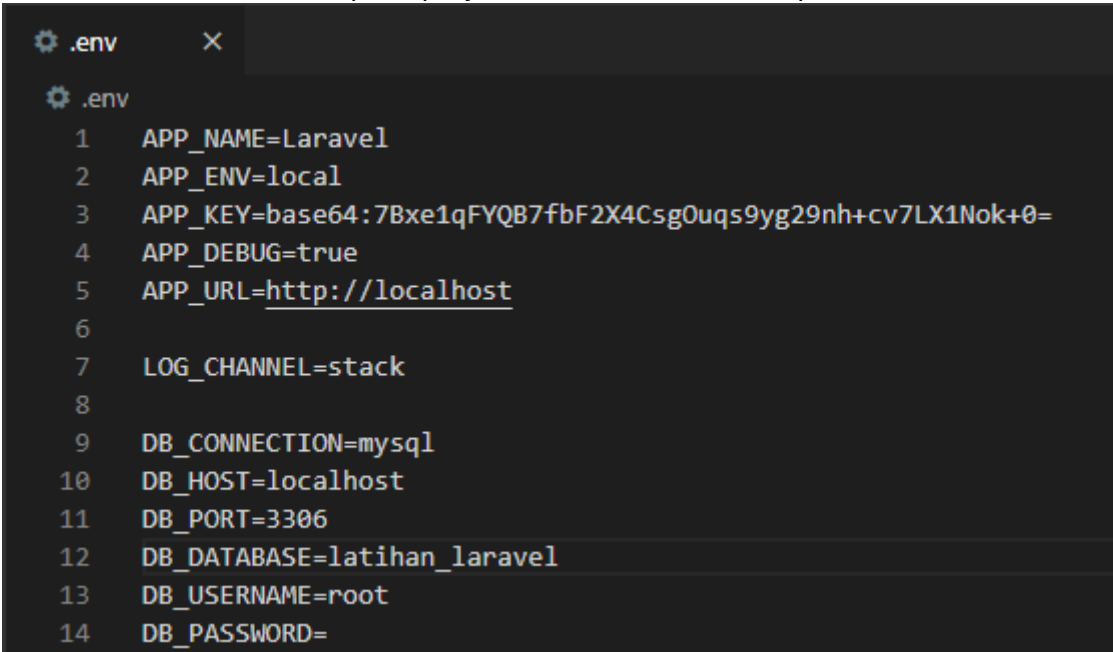
Langkah	Keterangan
1	<p>Buatlah tombol untuk menghapus data (Line 30) pada file index.blade.php di folder resources/views</p> <pre> 24 <td> 25 id }}" class="badge badge-info">Detail 26 id }}" class="badge badge-warning">Edit 27 id }}" class="badge badge-danger">Hapus 28 </td> 29 </tr> </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p>

	<pre> 20 Route::get('/mahasiswa', 'MahasiswaController@index'); 21 22 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 23 24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 25 26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 27 28 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 29 30 Route::post('/mahasiswa/update', 'MahasiswaController@update'); 31 32 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 33 </pre>
3	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p> <pre> 56 57 public function hapus(\$id) 58 { 59 DB::table('mahasiswa')->where('id',\$id)->delete(); 60 return redirect('/mahasiswa'); 61 } 62 63 </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Line 67 : query builder untuk menghapus data mahasiswa berdasarkan id yang dipilih
4	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir.</p>



Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

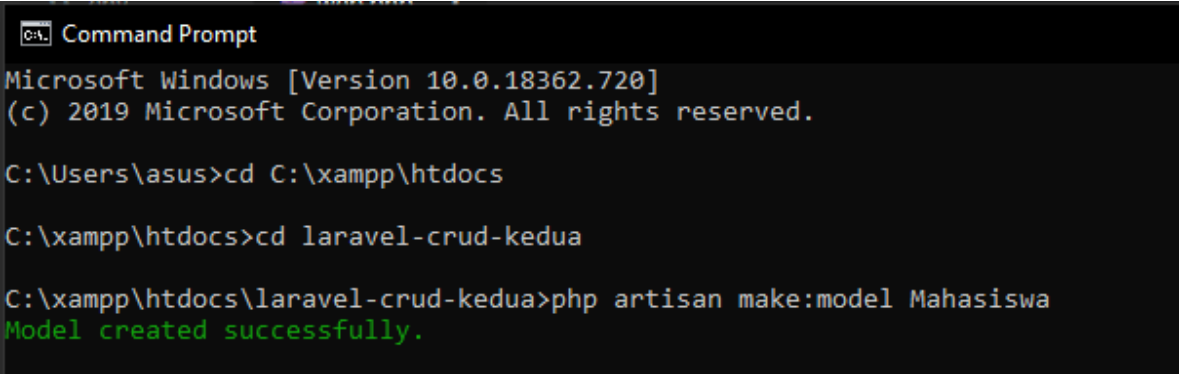
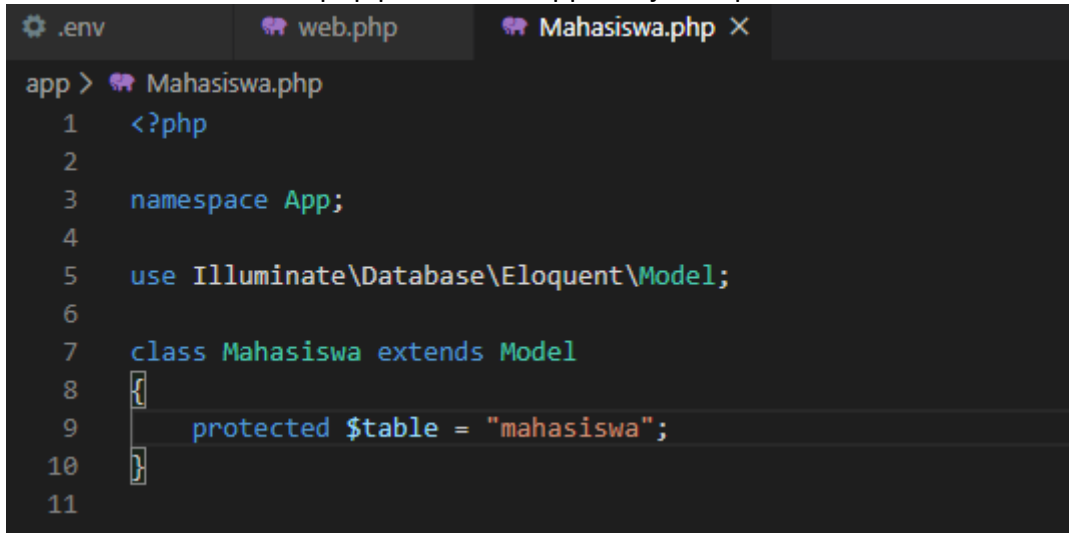
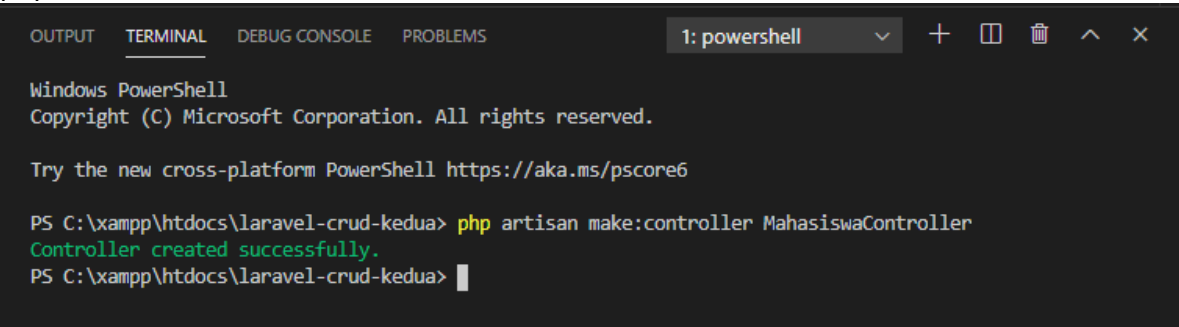
a. Konfigurasi Database

Langkah	Keterangan
1	<p>Buatlah project Laravel baru dengan nama laravel-crud-kedua. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud-kedua</pre> 
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.</p>  <p>Keterangan:</p> <ul style="list-style-type: none">- Nama database yang akan digunakan adalah latihan_laravel dengan username root.
3	<p>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</p>

<div> <div>Server: 127.0.0.1 » Database: latihan_laravel » Table: mahasiswa</div> <div> Browse Structure SQL Search Insert Export Import Privileges Operations Tran </div> </div> <div> <div>Table structure</div> <div>Relation view</div> </div> <table> <tr> <th>#</th><th>Name</th><th>Type</th><th>Collation</th><th>Attributes</th><th>Null</th><th>Default</th><th>Comments</th><th>Extra</th><th colspan="2">Action</th></tr> <tr> <td>1</td><td>id</td><td>int(11)</td><td></td><td></td><td>No</td><td>None</td><td></td><td>AUTO_INCREMENT</td><td>Change</td><td>Drop</td></tr> <tr> <td>2</td><td>nama</td><td>varchar(100)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td><td>Change</td><td>Drop</td></tr> <tr> <td>3</td><td>nim</td><td>varchar(10)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td><td>Change</td><td>Drop</td></tr> <tr> <td>4</td><td>email</td><td>varchar(50)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td><td>Change</td><td>Drop</td></tr> <tr> <td>5</td><td>jurusan</td><td>varchar(20)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td><td>Change</td><td>Drop</td></tr> <tr> <td>6</td><td>created_at</td><td>timestamp</td><td></td><td></td><td>Yes</td><td>NULL</td><td></td><td></td><td>Change</td><td>Drop</td></tr> <tr> <td>7</td><td>updated_at</td><td>timestamp</td><td></td><td></td><td>Yes</td><td>NULL</td><td></td><td></td><td>Change</td><td>Drop</td></tr> </table>											#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		1	id	int(11)			No	None		AUTO_INCREMENT	Change	Drop	2	nama	varchar(100)	utf8mb4_general_ci		No	None			Change	Drop	3	nim	varchar(10)	utf8mb4_general_ci		No	None			Change	Drop	4	email	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	5	jurusan	varchar(20)	utf8mb4_general_ci		No	None			Change	Drop	6	created_at	timestamp			Yes	NULL			Change	Drop	7	updated_at	timestamp			Yes	NULL			Change	Drop
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action																																																																																									
1	id	int(11)			No	None		AUTO_INCREMENT	Change	Drop																																																																																								
2	nama	varchar(100)	utf8mb4_general_ci		No	None			Change	Drop																																																																																								
3	nim	varchar(10)	utf8mb4_general_ci		No	None			Change	Drop																																																																																								
4	email	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop																																																																																								
5	jurusan	varchar(20)	utf8mb4_general_ci		No	None			Change	Drop																																																																																								
6	created_at	timestamp			Yes	NULL			Change	Drop																																																																																								
7	updated_at	timestamp			Yes	NULL			Change	Drop																																																																																								

b. Menampilkan Data dari Database

Langkah	Keterangan
1	<p>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.</p> <pre> .env web.php x routes > web.php 1 <?php 2 3 use Illuminate\Support\Facades\Route; 4 5 /* 6 ----- 7 Web Routes 8 ----- 9 10 Here is where you can register web routes for your application. These 11 routes are loaded by the RouteServiceProvider within a group which 12 contains the "web" middleware group. Now create something great! 13 14 */ 15 16 // Route::get('/', function () { 17 // return view('welcome'); 18 // }); 19 20 Route::get('/', 'MahasiswaController@index'); 21 Route::get('/mahasiswa', 'MahasiswaController@index'); 22 </pre>
2	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan</p> <pre>cd laravel-crud-kedua php artisan make:model Mahasiswa</pre>

	 <pre> C:\Users\asus>cd C:\xampp\htdocs C:\xampp\htdocs>cd laravel-crud-kedua C:\xampp\htdocs\laravel-crud-kedua>php artisan make:model Mahasiswa Model created successfully. </pre>
3	<p>Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.</p>  <pre> app > Mahasiswa.php 1 <?php 2 3 namespace App; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class Mahasiswa extends Model 8 { 9 protected \$table = "mahasiswa"; 10 } 11 </pre> <p>Keterangan: - Model Mahasiswa akan menangani tabel mahasiswa</p>
4	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan php artisan make:controller MahasiswaController</p>  <pre> Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved. Try the new cross-platform PowerShell https://aka.ms/pscore6 PS C:\xampp\htdocs\laravel-crud-kedua> php artisan make:controller MahasiswaController Controller created successfully. PS C:\xampp\htdocs\laravel-crud-kedua> </pre>
5	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p>

```

Mahasiswa.php MahasiswaController.php X
app > Http > Controllers > MahasiswaController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Mahasiswa;
7
8  class MahasiswaController extends Controller
9  {
10     public function index()
11     {
12         $mahasiswa = Mahasiswa::all();
13         return view('index', ['mahasiswa' => $mahasiswa]);
14     }
15 }
16

```

Keterangan:

- Tambahkan 'use App\Mahasiswa' (line 6) untuk menggunakan model Mahasiswa
- Line 12 untuk mengambil semua data dari model/tabel Mahasiswa dan akan disimpan di variabel \$mahasiswa
- Line 16 : data akan dikirim ke blade view bernama index

6

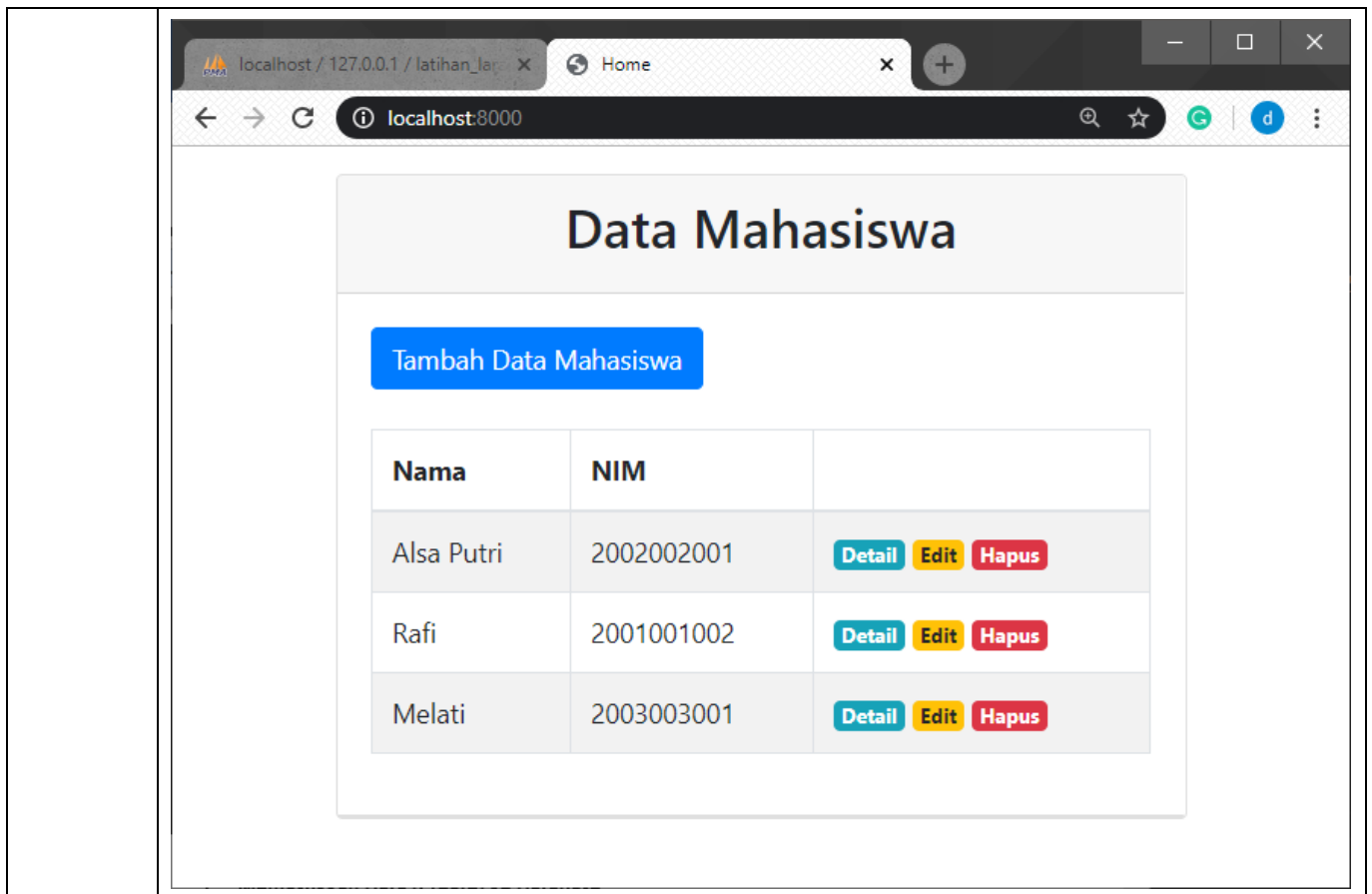
Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat template blade (seperti pada Bagian 1). Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)

```

Mahasiswa.php MahasiswaController.php master.blade.php X
resources > views > master.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
7      <title> @yield('title') </title>
8  </head>
9  <body>
10     <div class="container">
11         <div class="row mt-3">
12             <div class="col-md-6">
13                 <div class="card">
14                     <div class="card-header text-center">
15                         <h2> @yield('judul_halaman') </h2>
16                     </div>
17                     <div class="card-body">
18                         @yield('konten')
19                     </div>
20                 </div>
21             </div>
22         </div>
23     </div>
24 </body>
25 </html>

```

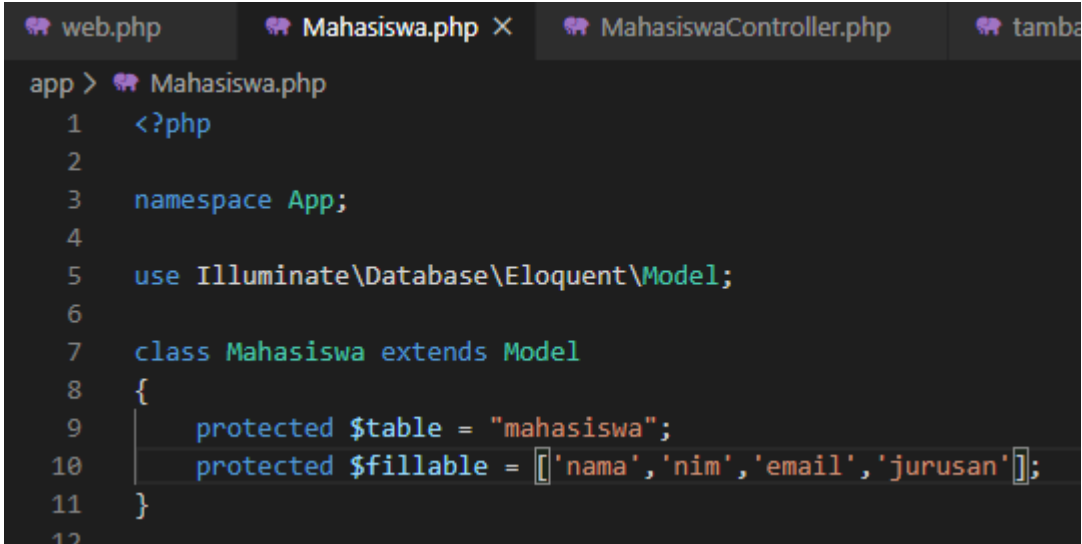
	<div> Mahasiswa.php MahasiswaController.php index.blade.php X </div> <pre> resources > views > index.blade.php 1 @extends('master') 2 3 @section('title', 'Home') 4 5 @section('judul_halaman', 'Data Mahasiswa') 6 7 @section('konten') 8 Tambah Data Mahasiswa 9
 10
 11 <table class="table table-bordered table-hover table-striped"> 12 <thead> 13 <tr> 14 <th>Nama</th> 15 <th>NIM</th> 16 <th> </th> 17 </tr> 18 </thead> 19 <tbody> 20 @foreach(\$mahasiswa as \$mhs) 21 <tr> 22 <td>{{ \$mhs->nama }}</td> 23 <td>{{ \$mhs->nim }}</td> 24 <td> 25 id }}" class="badge badge-info">Detail 26 id }}" class="badge badge-warning">Edit 27 id }}" class="badge badge-danger">Hapus 28 </td> 29 </tr> 30 @endforeach 31 </tbody> 32 </table> 33 @endsection </pre>
7	<p>Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud php artisan serve Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut</p>



c. Memasukkan Data (Create) ke Database

Langkah	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre> 19 20 Route::get('/', 'MahasiswaController@index'); 21 Route::get('/mahasiswa', 'MahasiswaController@index'); 22 23 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 24 </pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> 15 16 public function tambah() 17 { 18 return view('tambah'); 19 } </pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. Kita lakukan copy paste dari tambah.blade.php di Bagian 1</p>

	<div data-bbox="300 118 1181 152"> web.php Mahasiswa.php MahasiswaController.php tambah.blade.php × </div> <div data-bbox="316 163 660 188">resources > views > tambah.blade.php</div> <pre data-bbox="331 192 1469 1010"> 1 @extends('master') 2 3 @section('title', 'Tambah Data') 4 5 @section('judul_halaman', 'Tambah Data Mahasiswa') 6 7 @section('konten') 8 Kembali 9
 10
 11 <form action="/mahasiswa/simpan" method="post"> 12 {{ csrf_field() }} 13 <div class="form-group"> 14 <label for="namamhs">Nama</label> 15 <input type="text" class="form-control" required="required" name="namamhs">
 16 </div> 17 <div class="form-group"> 18 <label for="nimmhs">NIM</label> 19 <input type="number" class="form-control" required="required" name="nimmhs">
 20 </div> 21 <div class="form-group"> 22 <label for="emailmhs">E-mail</label> 23 <input type="email" class="form-control" required="required" name="emailmhs">
 24 </div> 25 <div class="form-group"> 26 <label for="jurusanmhs">Jurusan</label> 27 <input type="text" class="form-control" required="required" name="jurusanmhs">
 28 </div> 29 <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button> 30 </form> 31 @endsection </pre>
--	--

	<pre> 20 21 public function simpan(Request \$request) 22 { 23 Mahasiswa::create([24 'nama' => \$request->namamhs, 25 'nim' => \$request->nimmhs, 26 'email' => \$request->emailmhs, 27 'jurusan' => \$request->jurusanmhs, 28]); 29 return redirect('/mahasiswa'); 30 31 } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Variabel \$request untuk menerima data yang akan ditambahkan ke database - Line 23-28 merupakan fitur eloquent menggunakan fungsi create() untuk insert data ke tabel mahasiswa
6	<p>Karena kita menggunakan fungsi create pada Controller, maka butuh ditambahkan code pada Line 10 yang disebut Mass Assignment pada model Mahasiswa.php. Mass Assignment digunakan untuk memfilter kolom mana yang boleh dan tidak boleh diinput.</p>  <pre> app > Mahasiswa.php 1 <?php 2 3 namespace App; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class Mahasiswa extends Model 8 { 9 protected \$table = "mahasiswa"; 10 protected \$fillable = ['nama', 'nim', 'email', 'jurusan']; 11 } 12 </pre>
7	<p>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.</p>

localhost / 127.0.0.1 / latihan_lat... x Tambah Data x +

localhost:8000/mahasiswa/tambah

Tambah Data Mahasiswa

[Kembali](#)

Nama

Budi

NIM

2004004001

E-mail

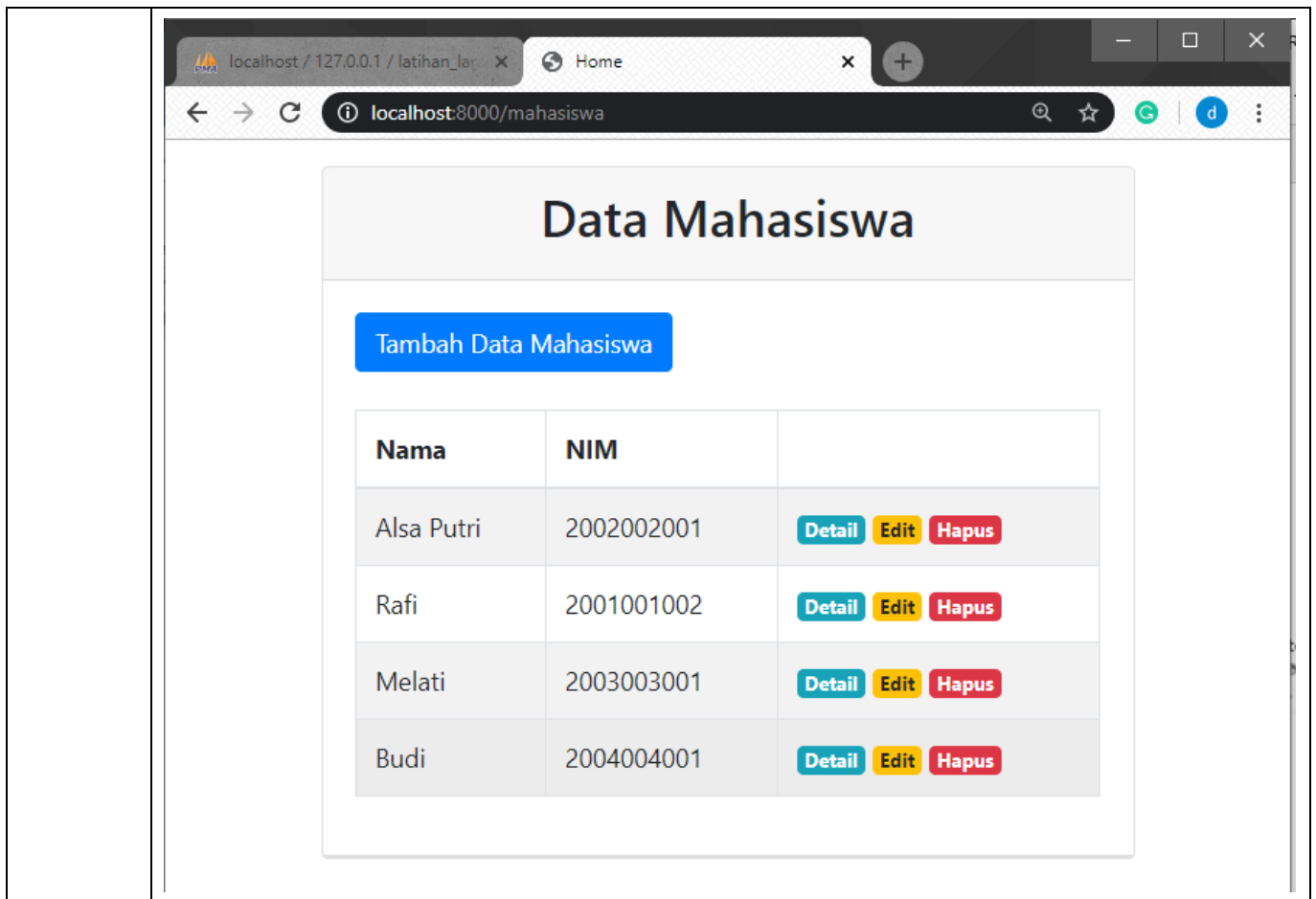
budi@gmail.com

Jurusan

teknik mesin

[Tambah Data](#)

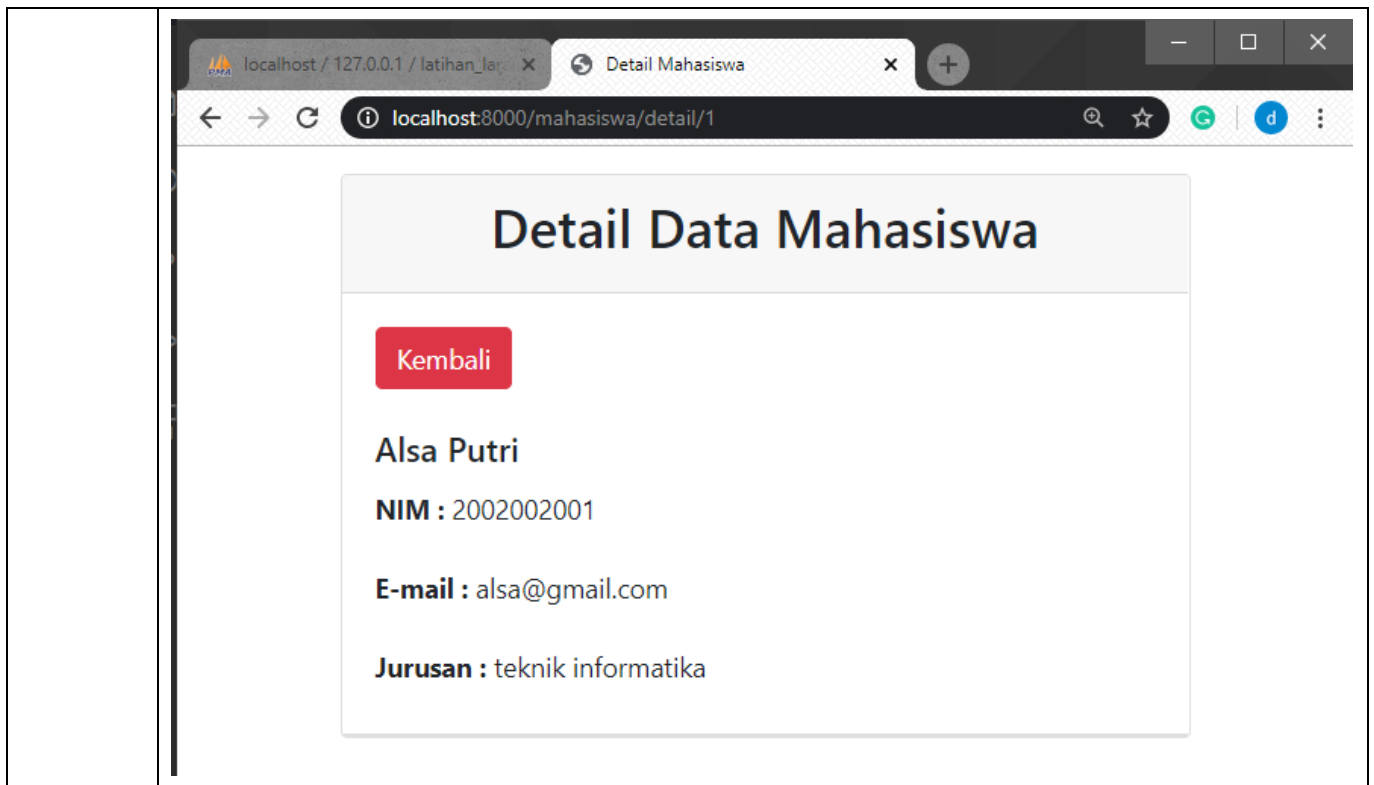
Setelah kita klik tombol tambah data, maka hasilnya sebagai berikut.



d. Melihat Detail Data dari Database

Langkah	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre> 19 20 Route::get('/', 'MahasiswaController@index'); 21 Route::get('/mahasiswa', 'MahasiswaController@index'); 22 23 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 24 25 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 26 27 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 28 29 </pre>
2	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p>

	<pre> 32 33 public function detail(\$id) 34 { 35 \$mahasiswa = Mahasiswa::find(\$id); 36 return view('detail', ['mahasiswa' => \$mahasiswa]); 37 } 38 } </pre>
3	<p>Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php.</p> <pre> resources > views > detail.blade.php 1 @extends('master') 2 3 @section('title', 'Detail Mahasiswa') 4 5 @section('judul_halaman', 'Detail Data Mahasiswa') 6 7 @section('konten') 8 Kembali 9 10
 11
 12 13 <h5 class="card-title"> {{ \$mahasiswa->nama }} </h5> 14 <p class="card-text"> 15 <label for=""> NIM : </label> {{ \$mahasiswa->nim }} 16 </p> 17 <p class="card-text"> 18 <label for=""> E-mail : </label> {{ \$mahasiswa->email }} 19 </p> 20 <p class="card-text"> 21 <label for=""> Jurusan : </label> {{ \$mahasiswa->jurusan }} 22 </p> 23 @endsection </pre>
4	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p>



e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre> 19 20 Route::get('/', 'MahasiswaController@index'); 21 Route::get('/mahasiswa', 'MahasiswaController@index'); 22 23 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 24 25 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 26 27 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 28 29 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 30 </pre>
2	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre> 38 39 public function edit(\$id) 40 { 41 \$mahasiswa = Mahasiswa::find(\$id); 42 return view('edit', ['mahasiswa' => \$mahasiswa]); 43 } 44 </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih

3

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

```

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/{{ $mahasiswa->id }}" method="post">
12         {{ csrf_field() }}
13         <input type="hidden" name="id" value="{{ $mahasiswa->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="nimamhs">NIM</label>
20             <input type="number" class="form-control" required="required" name="nimamhs" 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 }}"><br/>
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="tambah" class="btn btn-primary float-right">Simpan Data</button>
31     </form>
32 @endsection

```

Keterangan:

- Line 11-31 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 11 terdapat action="/mahasiswa/update/{{ \$mahasiswa->id }}" yang menunjukkan routes /mahasiswa/update/{id} dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController

4

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

```

19
20 Route::get('/', 'MahasiswaController@index');
21 Route::get('/mahasiswa', 'MahasiswaController@index');
22
23 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');
24
25 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');
26
27 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');
28
29 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');
30
31 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update');
32

```

Keterangan: - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php

5

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

```

44
45     public function update($id, Request $request)
46     {
47         $mahasiswa = Mahasiswa::find($id);
48         $mahasiswa->nama = $request->namamhs;
49         $mahasiswa->nim = $request->nimmhs;
50         $mahasiswa->email = $request->emailmhs;
51         $mahasiswa->jurusan = $request->jurusanmhs;
52         $mahasiswa->save();
53         return redirect('/mahasiswa');
54     }
55

```

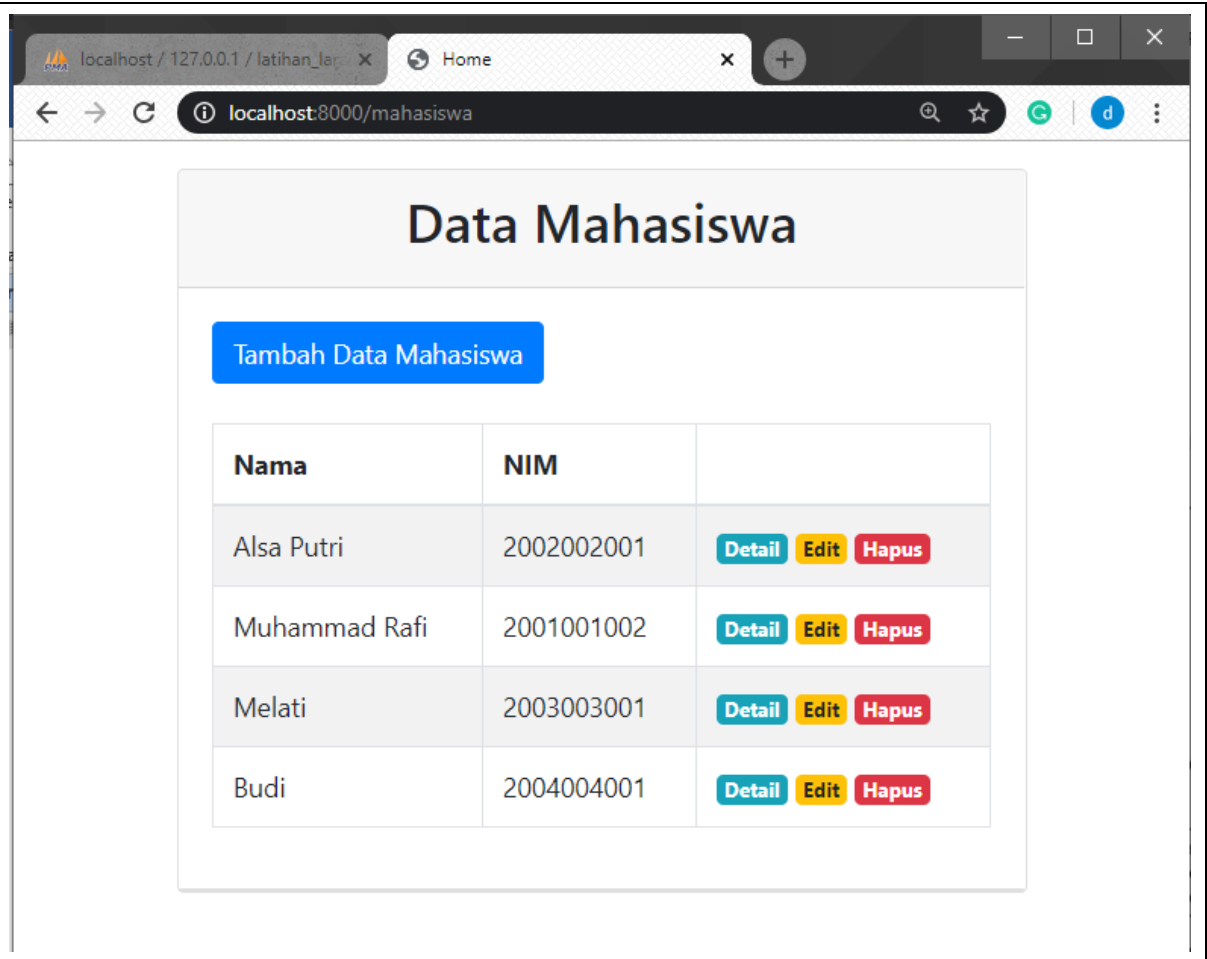
Keterangan: - Variabel \$request untuk menerima data yang akan ditambahkan ke database - Line 48-52 merupakan fungsi eloquent untuk update data ke tabel mahasiswa

6

Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru. Cobalah untuk mengedit data.

The screenshot shows a web browser window with the address bar displaying 'localhost:8000/mahasiswa/edit/2'. The page title is 'Edit Data Mahasiswa'. The form contains the following fields and buttons:

- Kembali** (Red button)
- Nama**: Input field containing 'Muhammad Rafi'.
- NIM**: Input field containing '2001001002'.
- E-mail**: Input field containing 'rafi@gmail.com'.
- Jurusan**: Input field containing 'teknik elektro'.
- Simpan Data** (Blue button)



f. Menghapus Data (Delete) dari Database

Langkah	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> 19 20 Route::get('/', 'MahasiswaController@index'); 21 Route::get('/mahasiswa', 'MahasiswaController@index'); 22 23 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 24 25 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 26 27 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 28 29 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 30 31 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update'); 32 33 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 34 </pre>
2	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p>

```

56
57     public function hapus($id)
58     {
59         $mahasiswa = Mahasiswa::find($id);
60         $mahasiswa->delete();
61         return redirect('/mahasiswa');
62     }
63
64

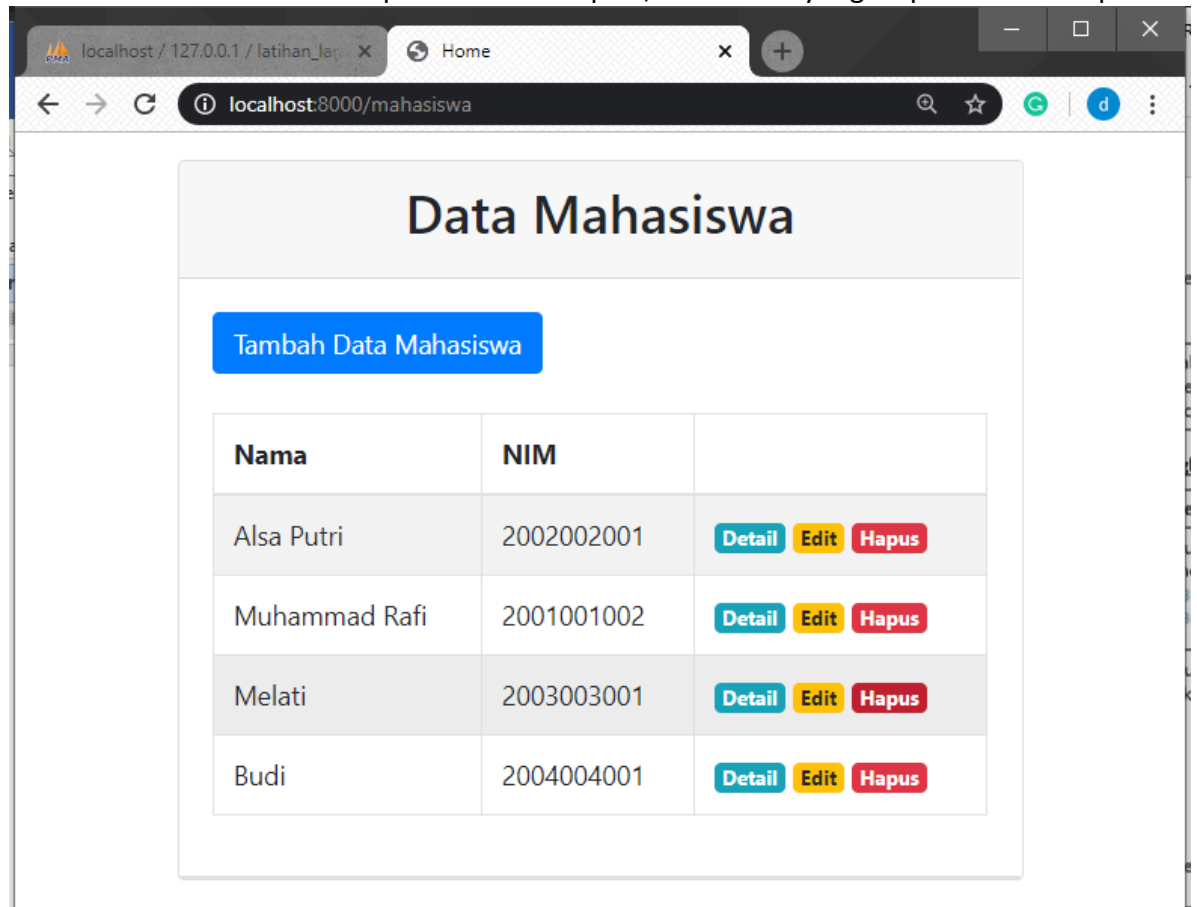
```

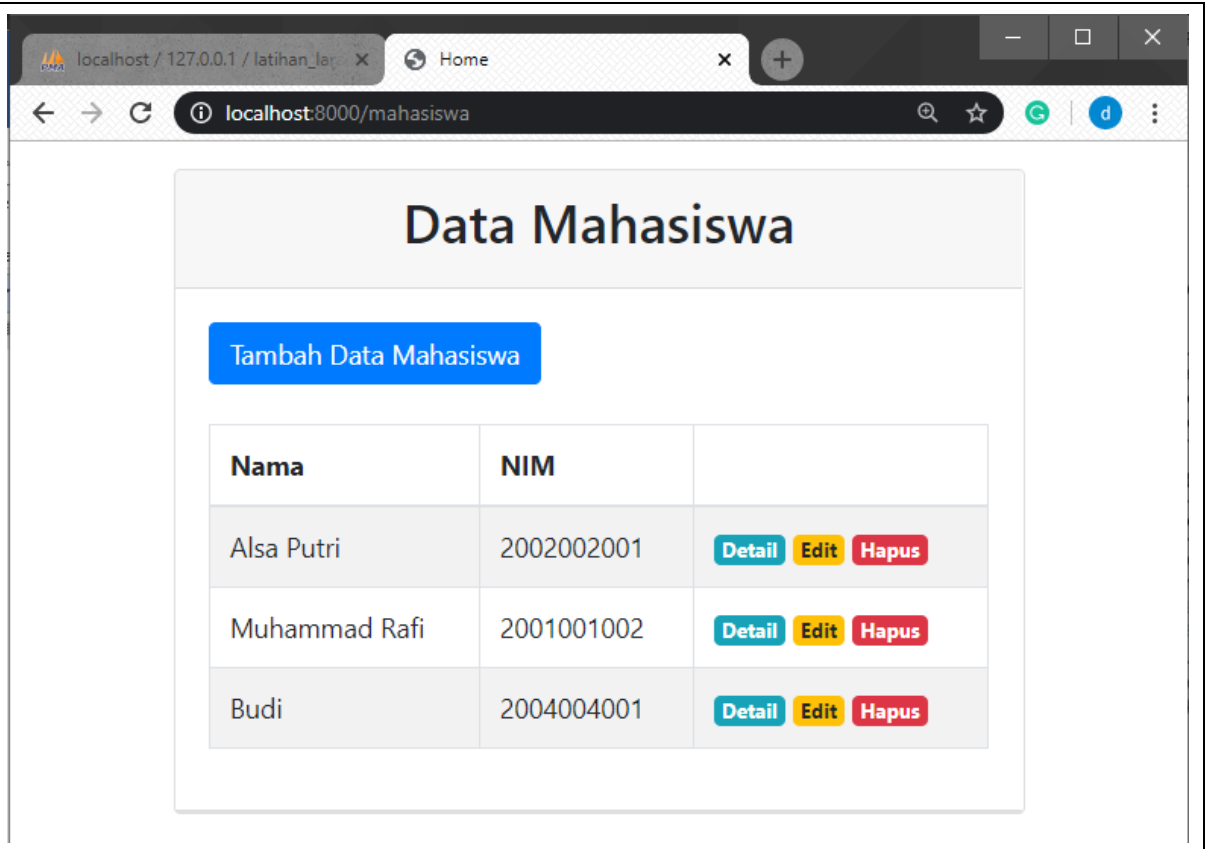
Keterangan :

- Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih

3

Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.





-- Selamat Mengerjakan --