

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT**

Jobsheet - 10

Membuat CRUD menggunakan Laravel



Oleh:

Farradila Ayu Damayanti

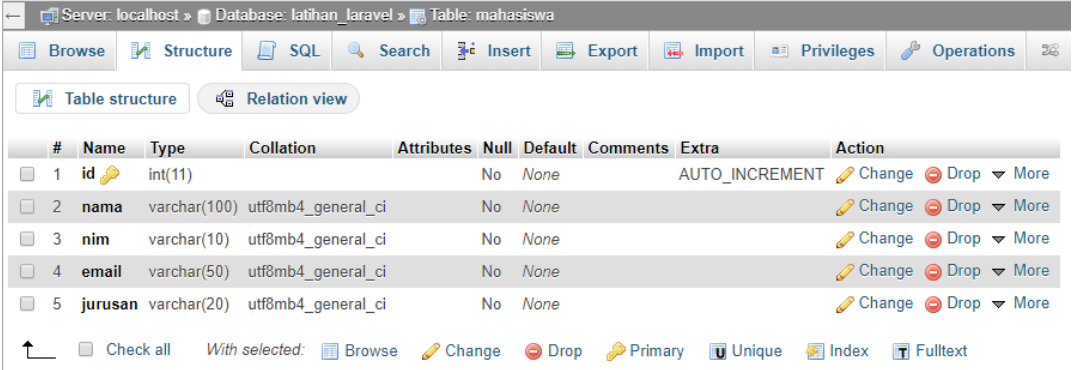
1841720074













TI-2B

**PROGRAM STUDI D-IV 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

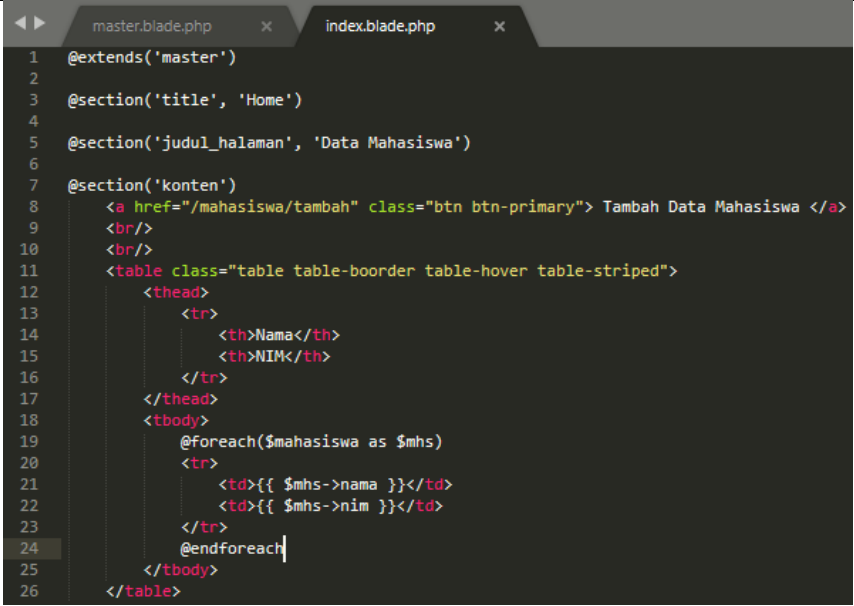
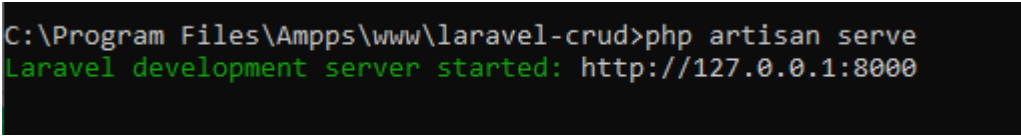
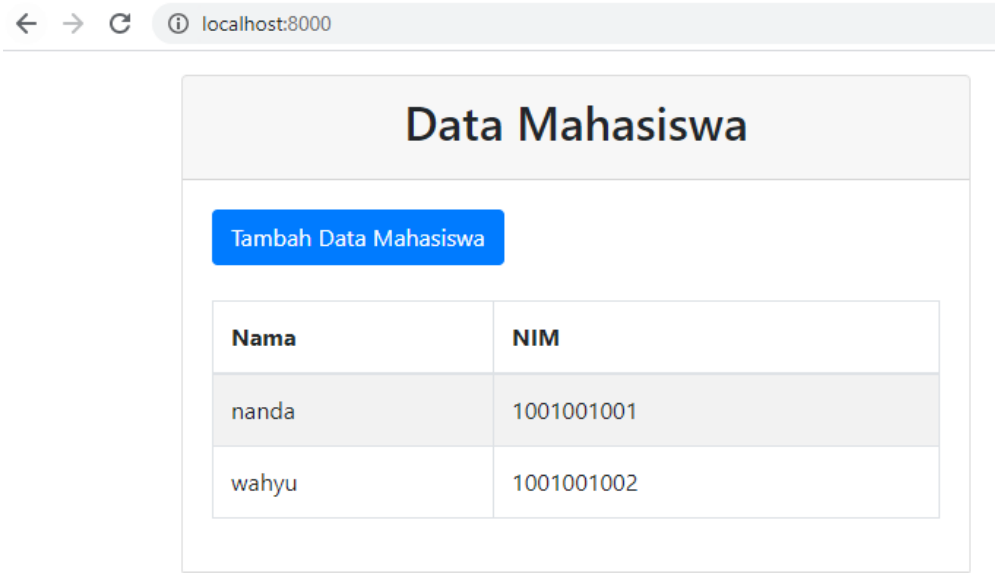
Langkah	Keterangan
1	<p>Buatlah project Laravel baru dengan nama laravel-crud. Buka command prompt, tuliskan perintah berikut.</p> <pre>C:\Program Files\Ampps\www>composer create-project --prefer-dist laravel/laravel laravel-crud Creating a "laravel/laravel" project at "./laravel-crud" Installing laravel/laravel (v7.3.0) - Installing laravel/laravel (v7.3.0): Loading from cache Created project in C:\Program Files\Ampps\www\laravel-crud > @php -r "file_exists('.env') copy('.env.example', '.env');" Loading composer repositories with package information Updating dependencies (including require-dev) Package operations: 92 installs, 0 updates, 0 removals - Installing voku/portable-ascii (1.4.10): Loading from cache - Installing symfony/polyfill-ctype (v1.15.0): Loading from cache</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> <pre>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</pre>
3	<p>Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database latihan_laravel.</p> 
4	<p>Isilah beberapa data pada tabel mahasiswa tersebut.</p>

+ Options	
	id nama nim email jurusan
<input type="checkbox"/>  Edit  Copy  Delete	1 nanda 1001001001 nandan@gmail.com Teknik Informatika
<input type="checkbox"/>  Edit  Copy  Delete	2 wahyu 1001001002 wahyu@gmail.com Teknik Elektro
 <input type="checkbox"/> Check all	With selected:  Edit  Copy  Delete  Export

b. Menampilkan Data dari Database

Langkah	Keterangan
1	<p>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');</pre>
2	<p>Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan</p>  <pre> C:\Program Files\Ampps\www\laravel-crud>php artisan make:controller MahasiswaController Controller created successfully.</pre>
3	Buat method index pada MahasiswaController.php pada folder app/Http/Controllers

		 <pre> 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 </pre>
4	<p>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.</p>	 <pre> 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <meta charset="utf-8"> 5 <meta name="viewport" content="width=device-width, initial-scale=1"> 6 <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet"> 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> </pre>
5	<p>Sekarang buat file index.blade.php yang menerapkan template master.blade.php dan akan digunakan untuk menampilkan data.</p>	

	 <pre> 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 </tr> 24 @endforeach 25 </tbody> 26 </table> </pre>
6	<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	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang

akan menjalankan fungsi tambah pada MahasiswaController

```
web.php x
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('/mahasiswa/tambah', 'MahasiswaController@tambah');
```

2

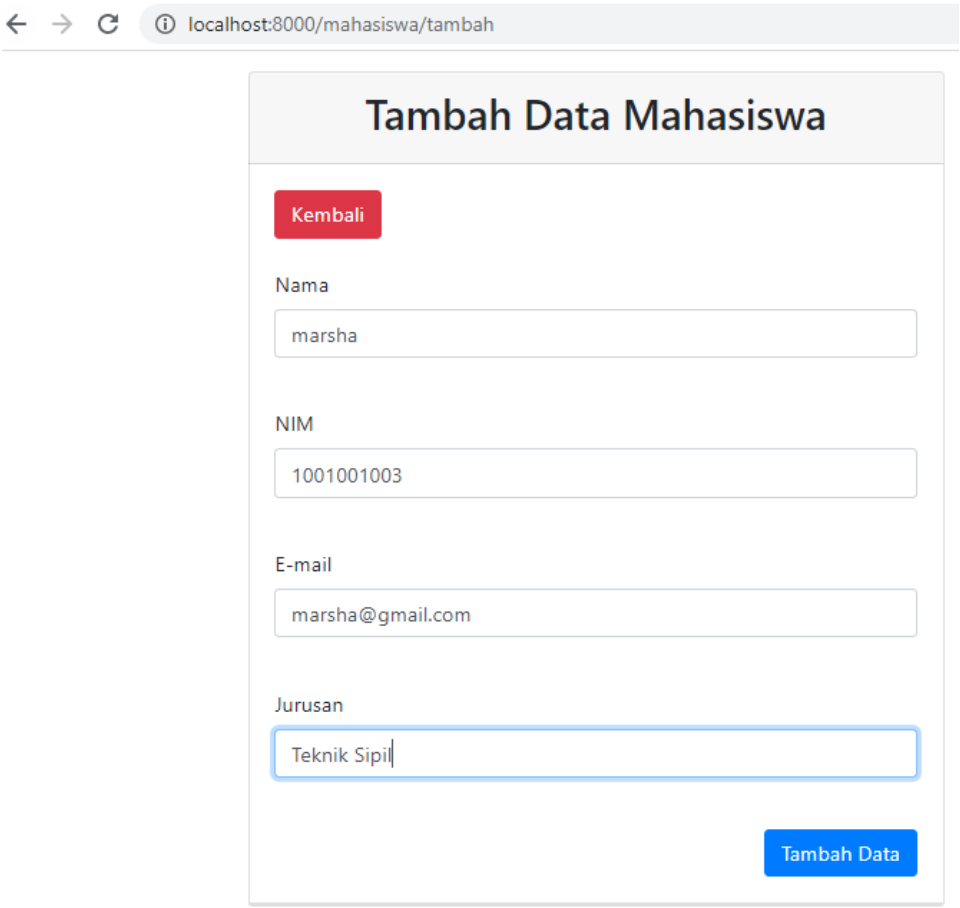
Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah

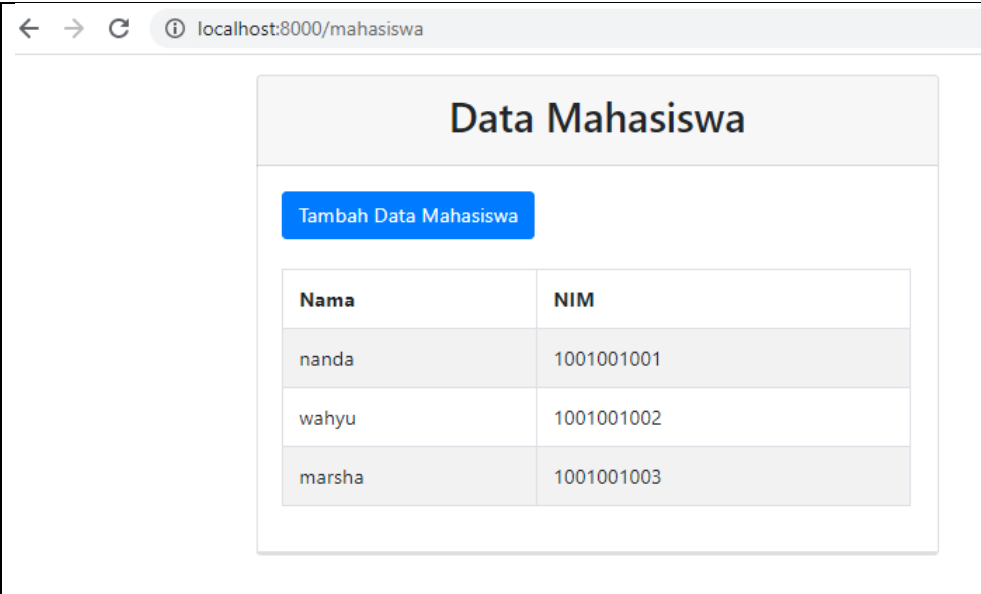
```
18
19 public function tambah(){
20     //memanggil view tambah
21     return view('tambah');
22 }
23
24
```

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.

```
tambah.blade.php x
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="nimhs">NIM</label>
19            <input type="number" class="form-control" required="required" name="nimhs"> <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">Nama</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
30    </form>
31 @endsection
```

4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 21 22 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); </pre>
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre> 24 public function simpan(Request \$request) 25 { 26 //insert data ke table mahasiswa 27 DB::table('mahasiswa')->insert([28 'nama' => \$request->namamhs, 29 'nim' => \$request->nimmhs, 30 'email' => \$request->emailmhs, 31 'jurusan' => \$request->jurusanmhs 32]); 33 return redirect('/mahasiswa'); 34 } 35 36 </pre>
6	<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>  <p>The screenshot shows a web browser window with the address bar displaying 'localhost:8000/mahasiswa/tambah'. The page content is a form titled 'Tambah Data Mahasiswa'. At the top left of the form is a red button labeled 'Kembali'. Below it are four input fields: 'Nama' with the value 'marsha', 'NIM' with the value '1001001003', 'E-mail' with the value 'marsha@gmail.com', and 'Jurusan' with the value 'Teknik Sipil'. At the bottom right of the form is a blue button labeled 'Tambah Data'.</p>

	
--	--

d. Melihat Detail Data dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk melihat detail data (Line 28) pada file index.blade.php di folder resources/views</p> <pre><td> id }}" class="badge badge-info">Detail </td></pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre>25 26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');</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>36 public function detail(\$id) 37 { 38 \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); 39 return view('detail',['mahasiswa' => \$mahasiswa]); 40 } 41 }</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>

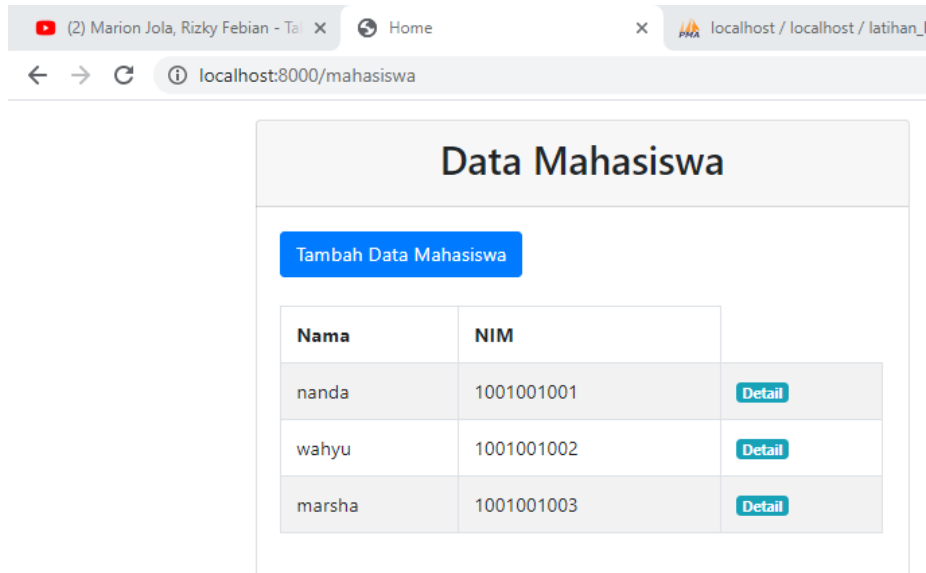

```

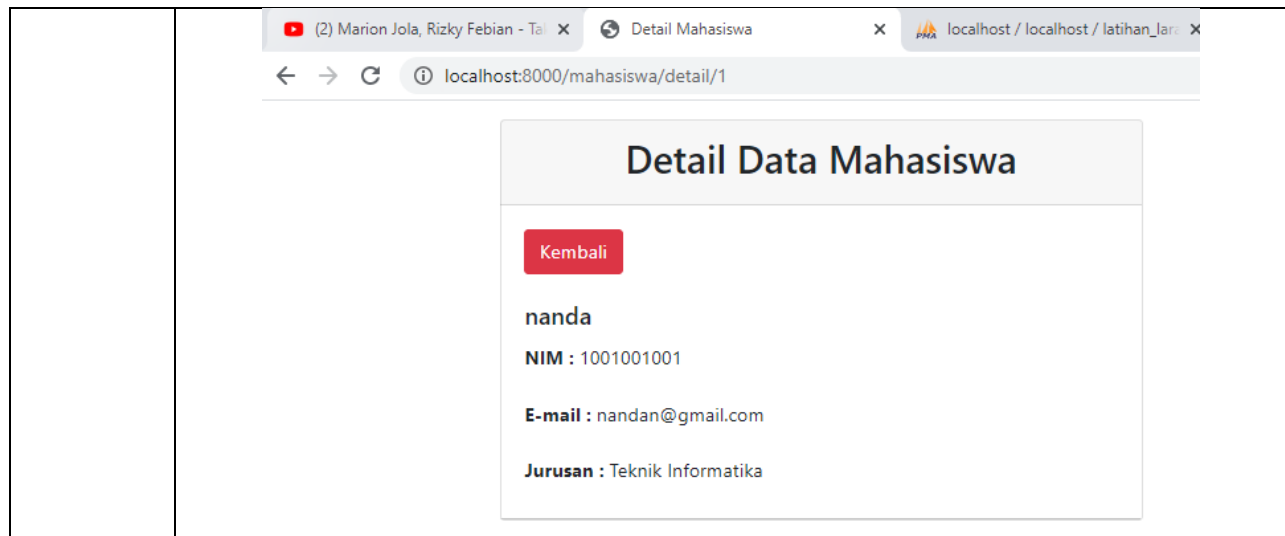
1  @extends('master')
2
3  @section('title','Detail Mahasiswa')
4
5  @section('judul_halaman','Detail Data Mahasiswa')
6
7  @section('konten')
8      <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
9      <br/>
10     <br/>
11
12     @foreach($mahasiswa as $mhs)
13         <h5 class="card-title">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

```

5

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





e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views</p> <pre> <td> id }}" class="badge badge-info">Detail id }}" class="badge badge-warning">Edit </td> </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre> 27 28 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); </pre>
3	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit</p> <pre> 41 42 public function edit(\$id) 43 { 44 \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); 45 return view('edit', ['mahasiswa' => \$mahasiswa]); 46 } 47 </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>

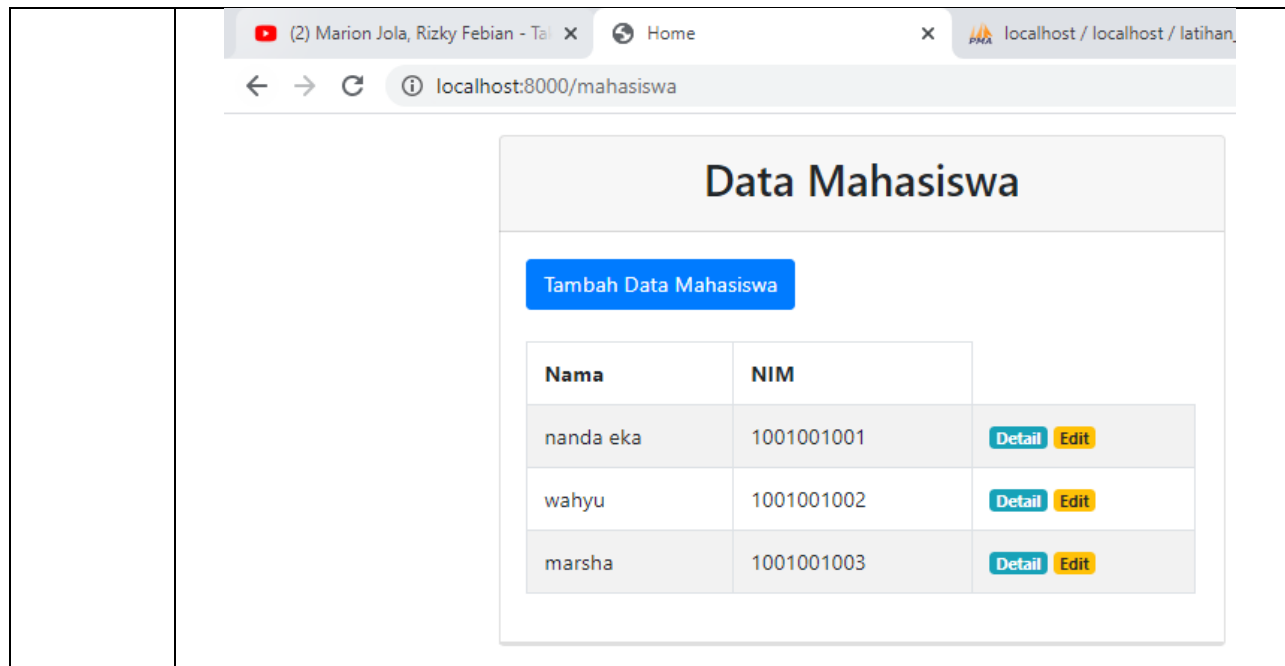
	<pre> 1 @extends('master') 2 3 @section('title','Edit Data') 4 5 @section('judul_halaman','Edit Data Mahasiswa') 6 7 @section('konten') 8 Kembali 9
 10
 11 @foreach(\$mahasiswa as \$mhs) 12 <form action="/mahasiswa/update" method="post"> 13 {{ csrf_field() }} 14 <input type="hidden" name="id" value="{{ \$mhs->id }}">
 15 <div class="form-group"> 16 <label for="namamhs">Nama</label> 17 <input type="text" name="namamhs" class="form-control" required="required" value="{{ \$mhs->nama }}">
 18 </div> 19 <div class="form-group"> 20 <label for="nimmhs">NIM</label> 21 <input type="number" name="nimmhs" class="form-control" required="required" value="{{ \$mhs->nim }}">
 22 </div> 23 <div class="form-group"> 24 <label for="emailmhs">E-mail</label> 25 <input type="email" name="emailmhs" class="form-control" required="required" value="{{ \$mhs->email }}">
 26 </div> 27 <div class="form-group"> 28 <label for="jurusanmhs">Jurusan</label> 29 <input type="text" name="jurusanmhs" class="form-control" required="required" 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>	
5	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 29 30 Route::post('/mahasiswa/update','MahasiswaController@update'); </pre>	
6	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database</p> <pre> 48 public function update(Request \$request) 49 { 50 //insert data ke table mahasiswa 51 DB::table('mahasiswa')->where('id',\$request->id)->update([52 'nama' => \$request->namamhs, 53 'nim' => \$request->nimmhs, 54 'email' => \$request->emailmhs, 55 'jurusan' => \$request->jurusanmhs 56]); 57 return redirect('/mahasiswa'); 58 } 59 </pre>	
7	<p>Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru.</p>	

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
nanda	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
marsha	1001001003	Detail Edit

Klik edit di data pertama, kita akan mengubah namanya

Edit Data Mahasiswa	
Kembali	
Nama	<input type="text" value="nanda eka"/>
NIM	<input type="text" value="1001001001"/>
E-mail	<input type="text" value="nandan@gmail.com"/>
Jurusan	<input type="text" value="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> <td> id }}" class="badge badge-info">Detail id }}" class="badge badge-warning">Edit id }}" class="badge badge-danger">Hapus </td> </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> 31 32 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); </pre>
3	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus</p> <pre> 60 public function hapus(\$id) 61 { 62 \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->delete(); 63 return redirect('/mahasiswa'); 64 } </pre>
4	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir. Data Awal :</p>

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
farradila	1841720074	Detail Edit Hapus
wafa	1001001004	Detail Edit Hapus

Menghapus data Marsha, hasilnya :

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
farradila	1841720074	Detail Edit Hapus
wafa	1001001004	Detail Edit Hapus

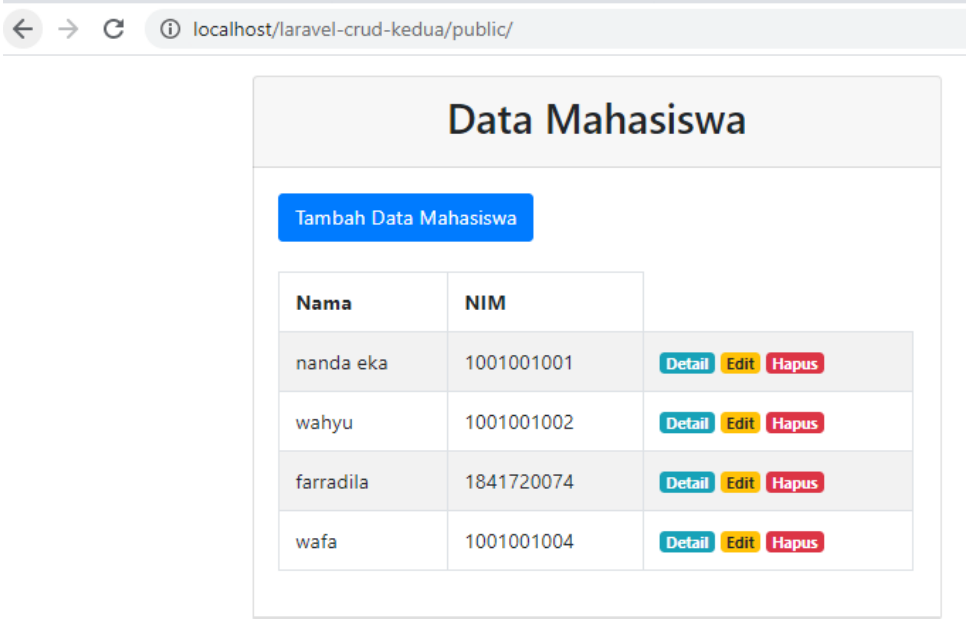
Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

a. Konfigurasi Database

Langkah	Keterangan																																																																																
1	<p>Buatlah project Laravel baru dengan nama laravel-crud-kedua. Buka command prompt, tuliskan perintah berikut.</p> <pre>C:\Program Files\Ampss\www>composer create-project --prefer-dist laravel/laravel laravel-crud-kedua Creating a "laravel/laravel" project at "./laravel-crud-kedua" Installing laravel/laravel (v7.3.0) - Installing laravel/laravel (v7.3.0): Loading from cache Created project in C:\Program Files\Ampss\www\laravel-crud-kedua > @php -r "file_exists('.env') copy('.env.example', '.env');" Loading composer repositories with package information Updating dependencies (including require-dev) Package operations: 92 installs, 0 updates, 0 removals - Installing voku/portable-ascii (1.4.10): Loading from cache - Installing symfony/polyfill-ctype (v1.15.0): Loading from cache</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> <pre>.env 1 APP_NAME=Laravel 2 APP_ENV=local 3 APP_KEY=base64:kBD1nbk8a2zLmMNuYPYUIqY8q4xz1A0qDd1GfEoA7EA= 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=mysql 15 16 BROADCAST_DRIVER=log 17 CACHE_DRIVER=file 18 QUEUE_CONNECTION=sync 19 SESSION_DRIVER=file 20 SESSION_LIFETIME=120</pre>																																																																																
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> <table><thead><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>Action</th></tr></thead><tbody><tr><td><input type="checkbox"/> 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 Drop More</td></tr><tr><td><input type="checkbox"/> 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 Drop More</td></tr><tr><td><input type="checkbox"/> 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 Drop More</td></tr><tr><td><input type="checkbox"/> 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 Drop More</td></tr><tr><td><input type="checkbox"/> 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 Drop More</td></tr><tr><td><input type="checkbox"/> 6</td><td>created_at</td><td>timestamp</td><td></td><td></td><td>No</td><td>None</td><td></td><td></td><td> Change Drop More</td></tr><tr><td><input type="checkbox"/> 7</td><td>update_at</td><td>timestamp</td><td></td><td></td><td>No</td><td>None</td><td></td><td></td><td> Change Drop More</td></tr></tbody></table>	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More	<input type="checkbox"/> 2	nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More	<input type="checkbox"/> 3	nim	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More	<input type="checkbox"/> 4	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More	<input type="checkbox"/> 5	jurusan	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More	<input type="checkbox"/> 6	created_at	timestamp			No	None			Change Drop More	<input type="checkbox"/> 7	update_at	timestamp			No	None			Change Drop More
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action																																																																								
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More																																																																								
<input type="checkbox"/> 2	nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More																																																																								
<input type="checkbox"/> 3	nim	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More																																																																								
<input type="checkbox"/> 4	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More																																																																								
<input type="checkbox"/> 5	jurusan	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More																																																																								
<input type="checkbox"/> 6	created_at	timestamp			No	None			Change Drop More																																																																								
<input type="checkbox"/> 7	update_at	timestamp			No	None			Change Drop More																																																																								

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>19 20 Route::get('/', 'MahasiswaController@index'); 21 Route::get('/mahasiswa', 'MahasiswaController@index');</pre>
2	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan</p> <pre>C:\Program Files\Amps\www\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>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>
4	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan</p> <pre>C:\Program Files\Amps\www\laravel-crud-kedua>php artisan make:controller MahasiswaController Controller created successfully.</pre>
5	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p> <pre>index.blade.php x MahasiswaController.php x 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</pre>

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)
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> <pre>C:\Program Files\Ampss\www\laravel-crud-kedua>php artisan serve Laravel development server started: http://127.0.0.1:8000</pre> 

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>22 23 Route::get('/mahasiswa/tambah','MahasiswaController@tambah');</pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre>public function tambah() { return view('tambah'); }</pre>

3	Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views.
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 25 Route::post('/mahasiswa/simpan','MahasiswaController@simpan'); 26 </pre>
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <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 } </pre>
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> 7 class Mahasiswa extends Model 8 { 9 protected \$table = "mahasiswa"; 10 protected \$fillable = ['nama', 'nim', 'email', 'jurusan']; 11 } 12 </pre>
7	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.

Tambah Data Mahasiswa

Kembali

Nama

rahmat

NIM

1001001006

E-mail

rahmat@gmail.com

Jurusan

Teknik Mesin

Tambah Data

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
farradila	1841720074	Detail Edit Hapus
wafa	1001001004	Detail Edit Hapus
rahmat	1001001006	Detail Edit Hapus

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> 26 27 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 28 </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 public function detail(\$id) 33 { 34 \$mahasiswa = Mahasiswa::find(\$id); 35 return view('detail', ['mahasiswa' => \$mahasiswa]); 36 } 37 </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> 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 <h5 class="card-title">{{ \$mahasiswa->nama }}</h5> 13 <p class="card-text"> 14 <label for="">NIM : </label> 15 {{ \$mahasiswa->nim}} 16 </p> 17 <p class="card-text"> 18 <label for="">E-Mail : </label> 19 {{ \$mahasiswa->email}} 20 </p> 21 <p class="card-text"> 22 <label for="">Jurusan : </label> 23 {{ \$mahasiswa->jurusan}} 24 </p> 25 26 @endsection </pre>
4	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p>

Detail Data Mahasiswa

[Kembali](#)

farradila

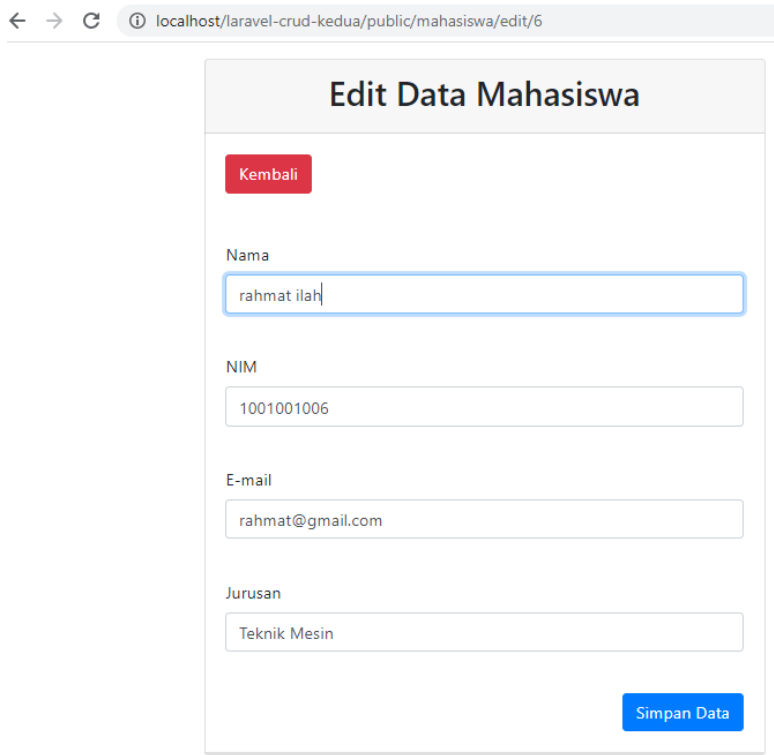
NIM : 1841720074

E-Mail : farradila429@gmail.com

Jurusan : Teknik Listrik

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> 24 25 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 26 </pre>
2	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre> 38 public function edit(\$id) 39 { 40 \$mahasiswa = Mahasiswa::find(\$id); 41 return view('edit', ['mahasiswa' => \$mahasiswa]); 42 } </pre>
3	<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> 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 12 <form action="/laravel-crud-kedua/public/mahasiswa/update/{ \$mahasiswa->id}" 13 method="post"> 14 {{ csrf_field() }} 15 <input type="hidden" name="id" value="{{ \$mahasiswa->id }}">
 16 <div class="form-group"> 17 <label for="namamhs">Nama</label> 18 <input type="text" name="namamhs" class="form-control" required="required" 19 value="{{ \$mahasiswa->nama }}">
 20 </div> 21 <div class="form-group"> 22 <label for="nimhs">NIM</label> 23 <input type="number" name="nimhs" class="form-control" required=" 24 required" value="{{ \$mahasiswa->nim }}">
 25 </div> 26 <div class="form-group"> 27 <label for="emailhs">E-mail</label> 28 <input type="email" name="emailhs" class="form-control" required=" 29 required" value="{{ \$mahasiswa->email }}">
 30 </div> 31 <div class="form-group"> 32 <label for="jurusanhs">Jurusan</label> 33 <input type="text" name="jurusanhs" class="form-control" required=" 34 required" value="{{ \$mahasiswa->jurusan }}">
 35 </div> 36 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan 37 Data</button> 38 </form> 39 @endsection </pre>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update/{id}. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 26 27 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update'); 28 </pre>
5	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p>

		<pre> public function update(\$id, 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'); } </pre>	
6	<p>Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru. Cobalah untuk mengedit data.</p> <p>Data awal :</p>  <p>Data berhasil dirubah :</p>		

<

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 style="background-color: #2d3748; color: #e2e3e5; padding: 10px; border-radius: 5px;"> 28 29 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');</pre>
2	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p> <pre style="background-color: #2d3748; color: #e2e3e5; padding: 10px; border-radius: 5px;"> 56 public function hapus(\$id) 57 { 58 \$mahasiswa = Mahasiswa::find(\$id); 59 \$mahasiswa->delete(); 60 return redirect('/mahasiswa'); 61 } 62</pre>
3	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.</p> <p>Data Awal :</p>

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
farradila	1841720074	Detail Edit Hapus
wafa	1001001004	Detail Edit Hapus
rahmat ilah	1001001006	Detail Edit Hapus

Data berhasil dihapus :

[←](#) [→](#) [↻](#) [localhost/laravel-crud-kedua/public/mahasiswa](#)

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
farradila	1841720074	Detail Edit Hapus
wafa	1001001004	Detail Edit Hapus