

# **LAPORAN PRAKTIKUM**

## **Jobsheet-10**

**Membuat CRUD Menggunakan Laravel  
Mata Kuliah Pemrograman Web Lanjut**



**Oleh:  
MINGGAR PUTRA DHEA R.      NIM. 1841720111**

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2020**



### Topik

Membuat CRUD – Create Read Update Delete Data dengan Framework Laravel

### Tujuan

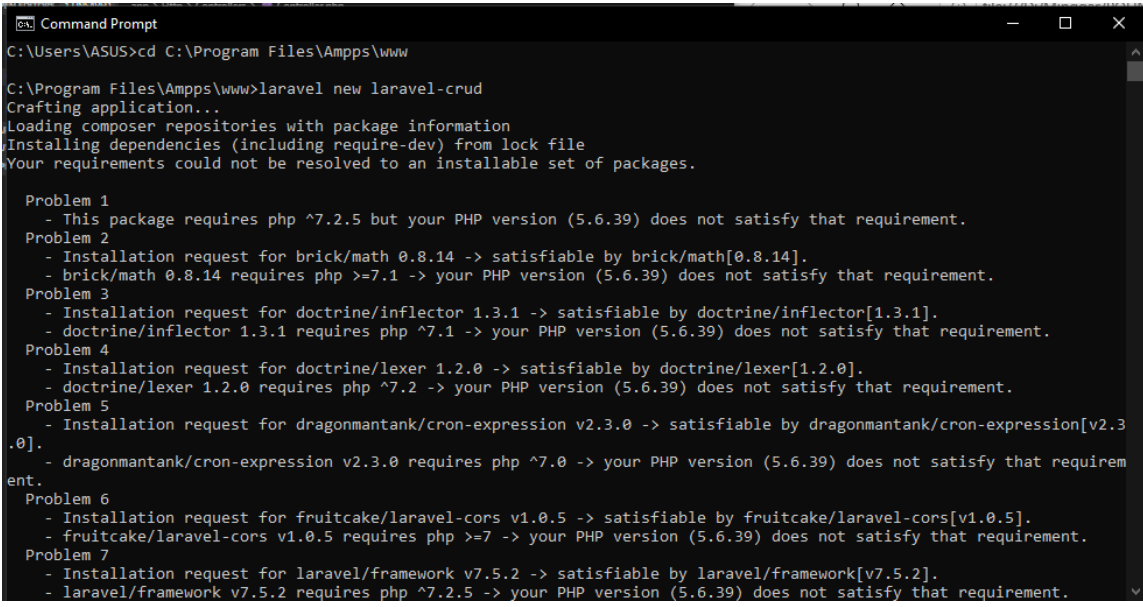
Mahasiswa diharapkan dapat:

1. Memahami bagaimana memasukkan data baru ke basis data menggunakan Laravel
2. Memahami bagaimana menampilkan data dari basis data menggunakan Laravel
3. Memahami bagaimana mengubah data dari basis data menggunakan Laravel
4. Memahami bagaimana menghapus data dari basis data menggunakan Laravel

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

Query Builder adalah salah satu fitur untuk menjalankan *query database*. Laravel menyediakan fungsi-fungsi yang dapat digunakan untuk menjalankan *query* sehingga penulisan query menjadi lebih singkat dan efisien.

#### a. Konfigurasi Database dan Pembuatan Tabel di MySQL

Langkah	Keterangan
1	<p>Buatlah project Laravel baru dengan nama <b>laravel-crud</b>. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud</pre> 

2

Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file **.env** pada project laravel-crud. Ubah seperti di bawah ini.

```

.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:XtQkZ9LSHh/My1umr+d7V5cAEknun4NwEHTFU5SYbFQ=
4 APP_DEBUG=true
5 APP_LOG_LEVEL=debug
6 APP_URL=http://localhost
7
8 DB_CONNECTION=mysql
9 DB_HOST=localhost
10 DB_PORT=3306
11 DB_DATABASE=latihan_laravel
12 DB_USERNAME=root
13 DB_PASSWORD=
14

```

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: localhost » Database: latihan\_laravel » Table: mahasiswa

Table structure

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

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Print Move columns Normalize

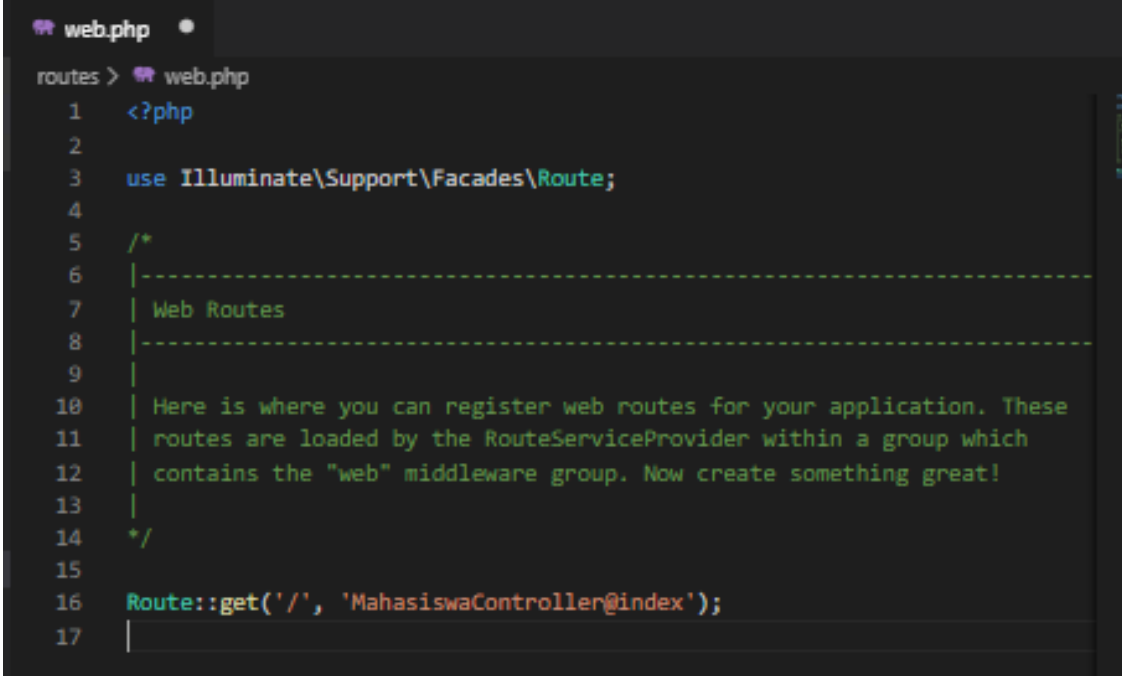
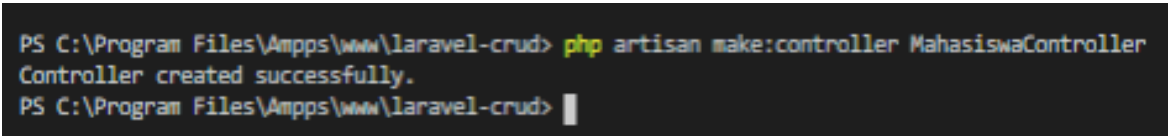
4

Isilah beberapa data pada tabel **mahasiswa** tersebut.

+ 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

## 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 <i>route</i> pada <b>routes/web.php</b> sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p>  <pre>web.php routes &gt; web.php 1  &lt;?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('/', 'MahasiswaController@index'); 17</pre> <p>Keterangan:</p> <ul style="list-style-type: none"><li>- Ketika route ('/') diakses, akan dijalankan method <b>index</b> pada controller bernama <b>MahasiswaController</b>.</li></ul>
2	<p>Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan</p> <p><b>cd laravel-crud</b> <b>php artisan make:controller MahasiswaController</b></p>  <pre>PS C:\Program Files\Amps\www\laravel-crud&gt; php artisan make:controller MahasiswaController Controller created successfully. PS C:\Program Files\Amps\www\laravel-crud&gt;</pre>

3

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

```

MahasiswaController.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

```

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**.

```

master.blade.php
resources > views > master.blade.php
1  <html>
2  <header>
3      <meta charset="utf-8">
4      <meta name="viewport" content="width=device-width, initial-scale=1">
5      <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" rel="stylesheet">
6      <title> @yield('title') </title>
7  </header>
8  <body>
9      <div class="container">
10         <div class="row mt-3">
11             <div class="col-md-6">
12                 <div class="card">
13                     <div class="card-header text-center">
14                         <!-- Bagian Judul Halaman -->
15                         <h2> @yield('judul_halaman') </h2>
16                     </div>
17                     <div class="card-body">
18                         <!-- Bagian Konten Blog -->
19                         @yield('konten')
20                     </div>
21                 </div>
22             </div>
23         </div>
24     </div>
25 </body>
26 </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.

```

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

```

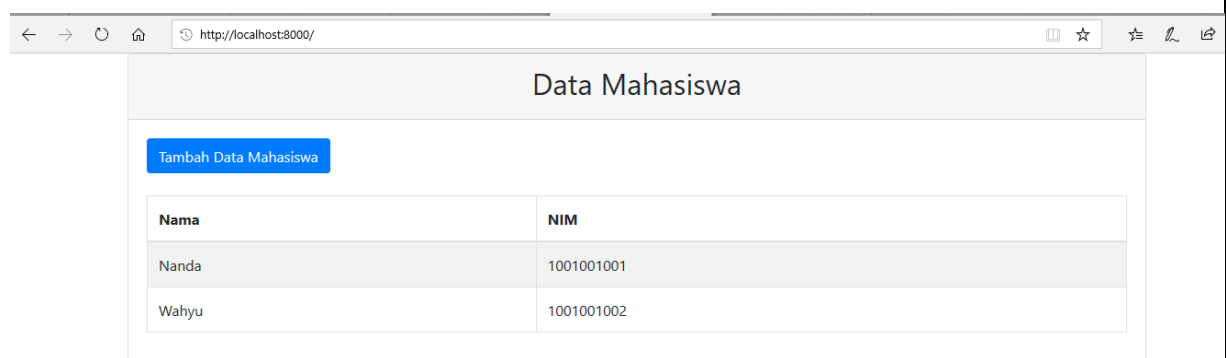
Keterangan:

- 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

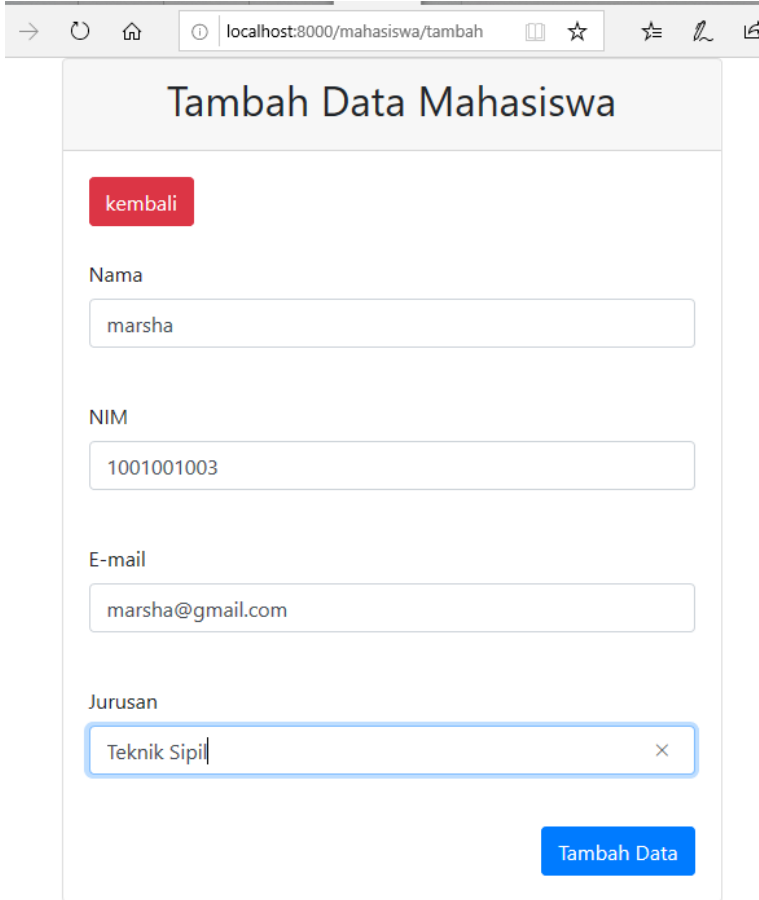
Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud **php artisan serve**

Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut



### c. Memasukkan Data (Create) ke Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/tambah</b> yang akan menjalankan fungsi tambah pada MahasiswaController</p> <pre>18 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');</pre>
2	<p>Buat method <b>tambah</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view tambah.</p> <pre>16 public function tambah() 17 { 18     return view('tambah');</pre>
3	<p>Kemudian buatlah view <b>tambah.blade.php</b> yang berisi form untuk memasukkan data baru pada folder <b>resources/views</b>. View tambah juga mengaplikasikan master.blade.php.</p> <pre>resources &gt; views &gt; 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     &lt;a href="/mahasiswa" class="btn btn-danger"&gt;kembali&lt;/a&gt; 9     &lt;br&gt; 10    &lt;br&gt; 11    &lt;form action="/mahasiswa/simpan" method="post"&gt; 12        {{ csrf_field() }} 13        &lt;div class="form-group"&gt; 14            &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 15            &lt;input type="text" class="form-control" required="required" name="namamhs"&gt; &lt;br&gt; 16        &lt;/div&gt; 17        &lt;div class="form-group"&gt; 18            &lt;label for="nimhs"&gt;NIM&lt;/label&gt; 19            &lt;input type="number" class="form-control" required="required" name="nimhs"&gt; &lt;br&gt; 20        &lt;/div&gt; 21        &lt;div class="form-group"&gt; 22            &lt;label for="emailmhs"&gt;E-mail&lt;/label&gt; 23            &lt;input type="email" class="form-control" required="required" name="emailmhs"&gt; &lt;br&gt; 24        &lt;/div&gt; 25        &lt;div class="form-group"&gt; 26            &lt;label for="jurusanmhs"&gt;Jurusan&lt;/label&gt; 27            &lt;input type="text" class="form-control" required="required" name="jurusanmhs"&gt; &lt;br&gt; 28        &lt;/div&gt; 29        &lt;button type="submit" name="tambah" class="btn btn-primary float-right"&gt;Tambah Data&lt;/button&gt; 30    &lt;/form&gt; 31 @endsection</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 13-32 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan</li> <li>- Line 13 terdapat action="/mahasiswa/simpan" yang menunjukkan routes /mahasiswa/simpan dimana data pada form tersebut akan dikirimkan ke fungsi simpan pada controller MahasiswaController</li> </ul> <p>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 <i>form hidden</i>.</p>

4	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/simpan</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 19 20 Route::post('/mahasiswa/simpan','MahasiswaController@simpan');  21 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php</li> </ul>
5	<p>Buat method <b>simpan</b> pada <b>MahasiswaController</b> untuk menyimpan data ke database.</p> <pre> 21 public function simpan(Request \$request) 22 { 23     DB::table('mahasiswa')-&gt;insert([ 24         'nama' =&gt; \$request-&gt;namamhs, 25         'nim' =&gt; \$request-&gt;nimmhs, 26         'email' =&gt; \$request-&gt;emailmhs, 27         'jurusan' =&gt; \$request-&gt;jurusanmhs 28     ]); 29     return redirect('/');  30 } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li> </ul> <p>Line 28-33 merupakan query builder untuk insert data ke tabel mahasiswa</p>
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>Setelah kita klik tombol tambah data, maka data baru akan ditampilkan pada halaman index.</p>



Home

localhost:8000/mahasiswa

## Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM
nanda	1001001001
wahyu	1001001002
marsha	1001001003

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

d. Melihat Detail Data dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk melihat detail data (Line 28) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre> resources &gt; views &gt; index.blade.php 1  @extends('master') 2 3  &lt;!-- isi title --&gt; 4  @section('title', 'Home') 5 6  &lt;!-- isi bagian judul halaman --&gt; 7  @section('judul_halaman', 'Data Mahasiswa') 8 9  &lt;!-- isi bagian konten --&gt; 10 @section('konten') 11     &lt;a href="/mahasiswa/tambah" class="btn btn-primary"&gt;Tambah Data Mahasiswa&lt;/a&gt; 12     &lt;br/&gt; 13     &lt;br/&gt; 14     &lt;table class="table table-bordered table-hover table-striped"&gt; 15         &lt;thead&gt; 16             &lt;tr&gt; 17                 &lt;th&gt;Nama&lt;/th&gt; 18                 &lt;th&gt;NIM&lt;/th&gt; 19                 &lt;th&gt;&lt;/th&gt; 20             &lt;/tr&gt; 21         &lt;/thead&gt; 22         &lt;tbody&gt; 23             @foreach(\$mahasiswa as \$mhs) 24                 &lt;tr&gt; 25                     &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 26                     &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 27                     &lt;td&gt; 28                         &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 29                     &lt;/td&gt; 30                 &lt;/tr&gt; 31             @endforeach 32         &lt;/tbody&gt; 33     &lt;/table&gt; 34 @endsection </pre>
2	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/detail</b> yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre> 19 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 20 </pre>

3

Buat method **detail** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menampilkan data mahasiswa pada view detail.blade.php.

```

30 public function detail($id){
31     //mengambil data mahasiswa berdasarkan id yang dipilih
32     $mahasiswa = DB::table('mahasiswa')->where('id',$id)->get();
33     //kirim data mahasiswa yang diambil ke view edit.blade.php
34     return view('detail',['mahasiswa' => $mahasiswa]);
35 }
36

```

4

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

```

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

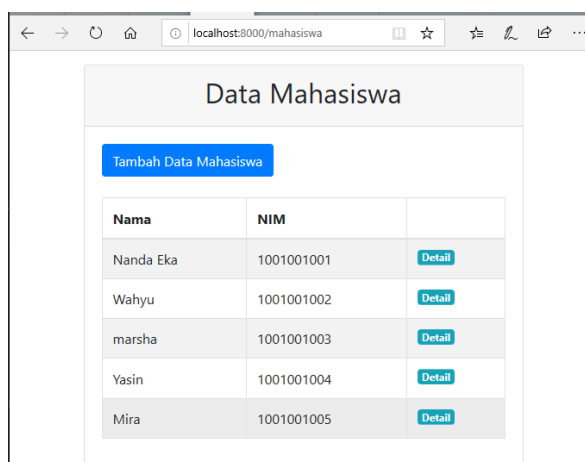
```

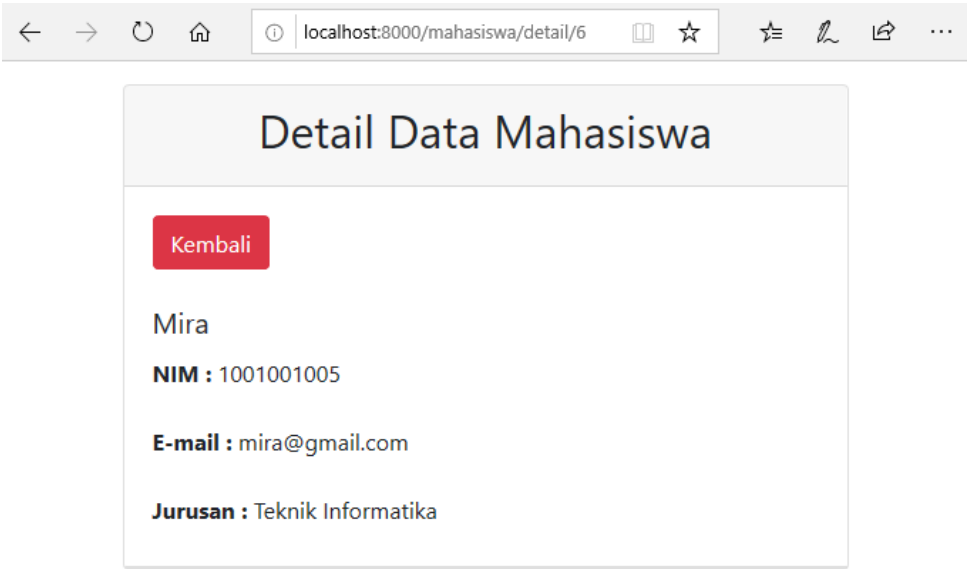
Keterangan:

- Line 16-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan

5

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

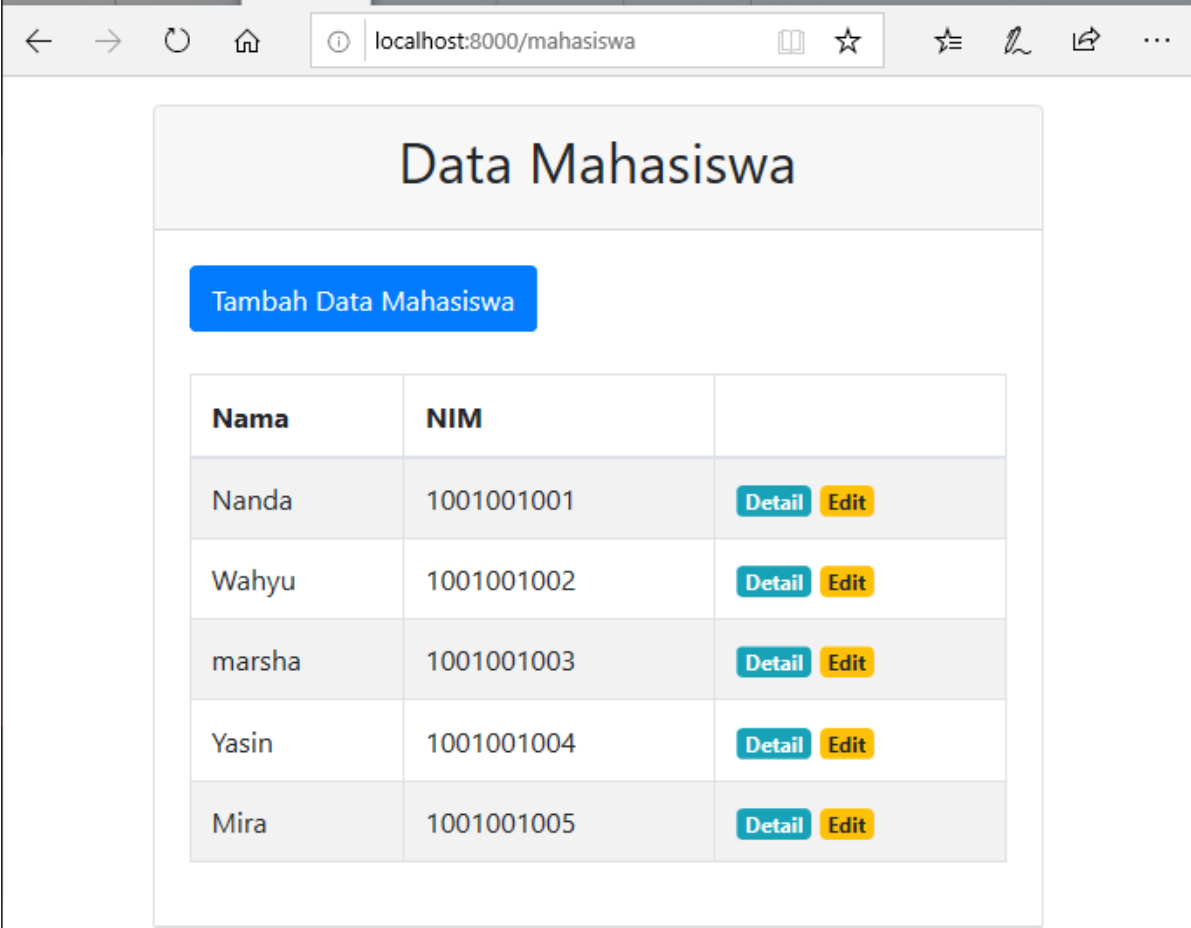


	
--	--

e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre> resources &gt; views &gt; index.blade.php 1  @extends('master') 2 3  &lt;!-- isi title --&gt; 4  @section('title', 'Home') 5 6  &lt;!-- isi bagian judul halaman --&gt; 7  @section('judul_halaman', 'Data Mahasiswa') 8 9  &lt;!-- isi bagian konten --&gt; 10 @section('konten') 11     &lt;a href="/mahasiswa/tambah" class="btn btn-primary"&gt;Tambah Data Mahasiswa&lt;/a&gt; 12     &lt;br/&gt; 13     &lt;br/&gt; 14     &lt;table class="table table-bordered table-hover table-striped"&gt; 15         &lt;thead&gt; 16             &lt;tr&gt; 17                 &lt;th&gt;Nama&lt;/th&gt; 18                 &lt;th&gt;NIM&lt;/th&gt; 19                 &lt;th&gt;&lt;/th&gt; 20             &lt;/tr&gt; 21         &lt;/thead&gt; 22         &lt;tbody&gt; 23             @foreach(\$mahasiswa as \$mhs) 24                 &lt;tr&gt; 25                     &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 26                     &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 27                     &lt;td&gt; 28                         &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 29                         &lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-warning"&gt;Edit&lt;/a&gt; 30                     &lt;/td&gt; 31                 &lt;/tr&gt; 32             @endforeach 33         &lt;/tbody&gt; 34     &lt;/table&gt; 35 @endsection </pre>
2	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/edit</b> yang akan menjalankan fungsi edit pada MahasiswaController</p>

	<pre> 20 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 21 </pre>
3	<p>Buat method <b>edit</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view edit.</p> <pre> 37 public function edit(\$id){ 38     //mengambil data mahasiswa berdasarkan id yang dipilih 39     \$mahasiswa = DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;get(); 40     //kirim data mahasiswa yang diambil ke view edit.blade.php 41     return view('edit',['mahasiswa' =&gt; \$mahasiswa]); 42 } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 49 : query builder untuk mengambil data mahasiswa berdasarkan id yang dipilih</li> </ul>
4	<p>Kemudian buatlah view <b>edit.blade.php</b> yang berisi form untuk mengubah data pada folder <b>resources/views</b>. View edit juga mengaplikasikan master.blade.php.</p> <pre> resources &gt; views &gt; edit.blade.php 1 @extends('master') 2 &lt;!-- isi title --&gt; 3 @section('title', 'Edit Data') 4 5 &lt;!-- isi bagian judul halaman --&gt; 6 @section('judul_halaman', 'Edit Data Mahasiswa') 7 8 &lt;!-- isi bagian konten --&gt; 9 @section('konten') 10 &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 11 &lt;br/&gt; 12 &lt;br/&gt; 13 @foreach(\$mahasiswa as \$mhs) 14 &lt;form action="/mahasiswa/update" method="post"&gt; 15     {{ csrf_field() }} 16     &lt;input type="hidden" name="id" value="{{ \$mhs-&gt;id }}"&gt; &lt;br/&gt; 17     &lt;div class="form-group"&gt; 18         &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 19         &lt;input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mhs-&gt;nama }}"&gt; &lt;br/&gt; 20     &lt;/div&gt; 21     &lt;div class="form-group"&gt; 22         &lt;label for="nimhs"&gt;NIM&lt;/label&gt; 23         &lt;input type="number" class="form-control" required="required" name="nimhs" value="{{ \$mhs-&gt;nim }}"&gt; &lt;br/&gt; 24     &lt;/div&gt; 25     &lt;div class="form-group"&gt; 26         &lt;label for="emailmhs"&gt;E-mail&lt;/label&gt; 27         &lt;input type="email" class="form-control" required="required" name="emailmhs" value="{{ \$mhs-&gt;email }}"&gt; &lt;br/&gt; 28     &lt;/div&gt; 29     &lt;div class="form-group"&gt; 30         &lt;label for="jurusanmhs"&gt;Jurusan&lt;/label&gt; 31         &lt;input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mhs-&gt;jurusan }}"&gt; &lt;br/&gt; 32     &lt;/div&gt; 33     &lt;button type="submit" name="edit" class="btn btn-primary float-right"&gt;Simpan Data&lt;/button&gt; 34 &lt;/form&gt; 35 @endforeach 36 @endsection </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 14-36 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan</li> <li>- Line 15 terdapat action="/mahasiswa/update" yang menunjukkan routes /mahasiswa/update dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController</li> </ul>
5	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/update</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 21 Route::post('/mahasiswa/update', 'MahasiswaController@update'); 22 </pre>

	<p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php</li> </ul>
6	<p>Buat method <b>update</b> pada <b>MahasiswaController</b> untuk menyimpan data yang diubah ke database.</p> <pre> 44 public function update(Request \$request){ 45     //update data mahasiswa 46     DB::table('mahasiswa')-&gt;where('id',\$request-&gt;id)-&gt;update([ 47         'nama' =&gt; \$request-&gt;namamhs, 48         'nim' =&gt; \$request-&gt;nimmhs, 49         'email' =&gt; \$request-&gt;emailmhs, 50         'jurusan' =&gt; \$request-&gt;jurusanmhs 51     ]); 52     return redirect('/mahasiswa'); 53 } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li> <li>- Line 58-63 merupakan query builder untuk update data ke tabel mahasiswa</li> </ul>
7	<p>Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru.</p>  <p>Klik edit di data pertama, kita akan mengubah namanya.</p>

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

Nama	NIM	
Nanda Eka	1001001001	<a href="#">Detail</a> <a href="#">Edit</a>
Wahyu	1001001002	<a href="#">Detail</a> <a href="#">Edit</a>
marsha	1001001003	<a href="#">Detail</a> <a href="#">Edit</a>
Yasin	1001001004	<a href="#">Detail</a> <a href="#">Edit</a>
Mira	1001001005	<a href="#">Detail</a> <a href="#">Edit</a>

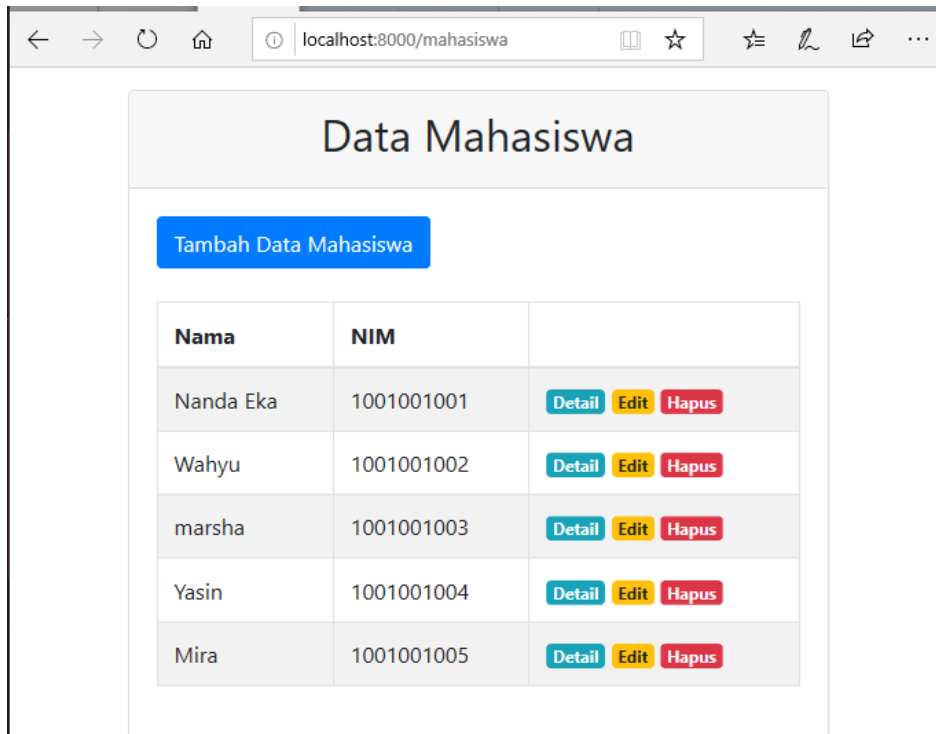
#### f. Menghapus Data (Delete) dari Database

Langkah	Keterangan
1	Buatlah tombol untuk menghapus data (Line 30) pada file <b>index.blade.php</b> di folder <b>resources/views</b>

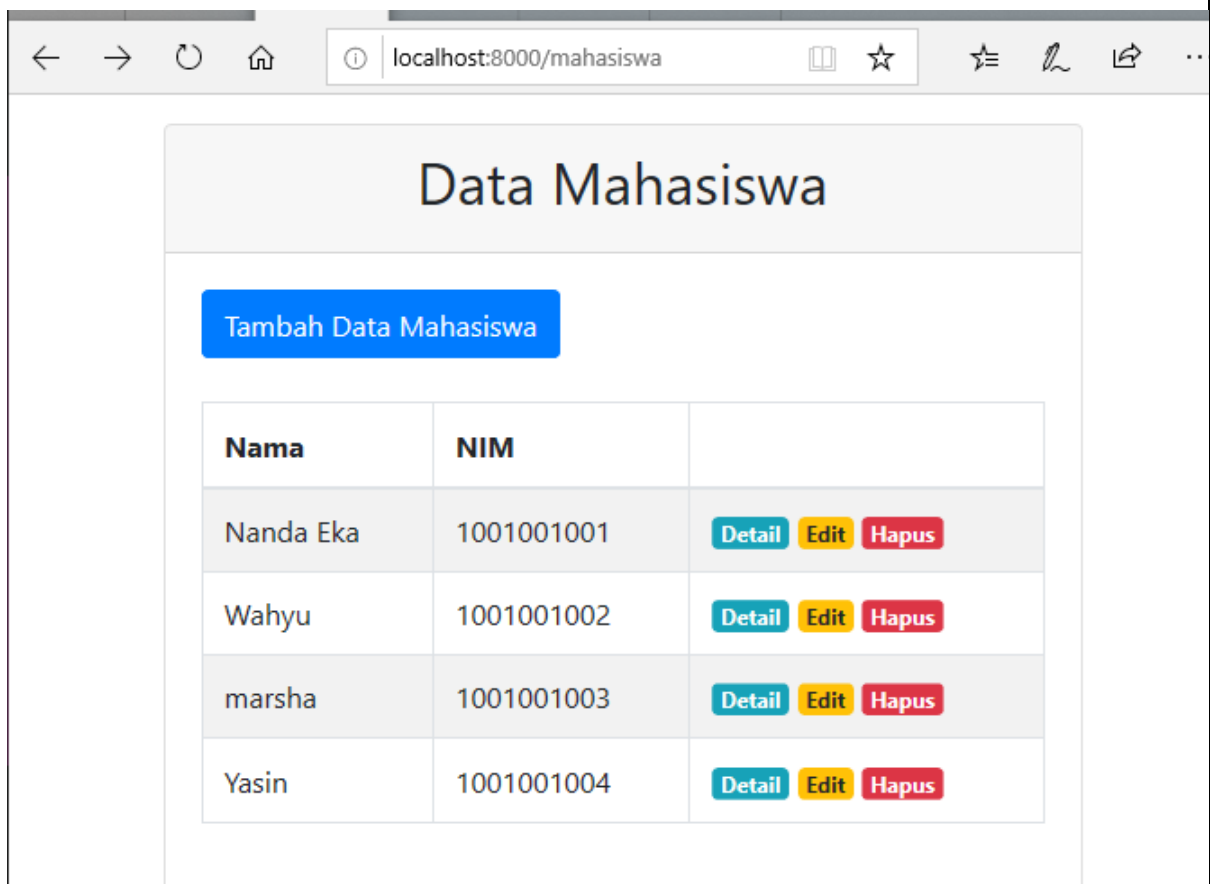
	<pre> resources &gt; views &gt; index.blade.php 1  @extends('master') 2 3  &lt;!-- isi title --&gt; 4  @section('title', 'Home') 5 6  &lt;!-- isi bagian judul halaman --&gt; 7  @section('judul_halaman', 'Data Mahasiswa') 8 9  &lt;!-- isi bagian konten --&gt; 10 @section('konten') 11     &lt;a href="/mahasiswa/tambah" class="btn btn-primary"&gt;Tambah Data Mahasiswa&lt;/a&gt; 12     &lt;br/&gt; 13     &lt;br/&gt; 14     &lt;table class="table table-bordered table-hover table-striped"&gt; 15         &lt;thead&gt; 16             &lt;tr&gt; 17                 &lt;th&gt;Nama&lt;/th&gt; 18                 &lt;th&gt;NIM&lt;/th&gt; 19                 &lt;th&gt;&lt;/th&gt; 20             &lt;/tr&gt; 21         &lt;/thead&gt; 22         &lt;tbody&gt; 23             @foreach(\$mahasiswa as \$mhs) 24                 &lt;tr&gt; 25                     &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 26                     &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 27                     &lt;td&gt; 28                         &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 29                         &lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-warning"&gt;Edit&lt;/a&gt; 30                         &lt;a href="/mahasiswa/hapus/{{ \$mhs-&gt;id }}" class="badge badge-danger"&gt;Hapus&lt;/a&gt; 31                     &lt;/td&gt; 32                 &lt;/tr&gt; 33             @endforeach 34         &lt;/tbody&gt; 35     &lt;/table&gt; 36 @endsection </pre>
2	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/hapus</b> yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> 22 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 23 </pre>
3	<p>Buat method <b>hapus</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menjalankan fungsi hapus.</p> <pre> 55 public function hapus(\$id){ 56     //menghapus data mahasiswa berdasarkan id yang dipilih 57     DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;delete(); 58 59     return redirect('/mahasiswa'); 60 61 </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 67 : query builder untuk menghapus data mahasiswa berdasarkan id yang dipilih</li> </ul>

4

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



Setelah dihapus :

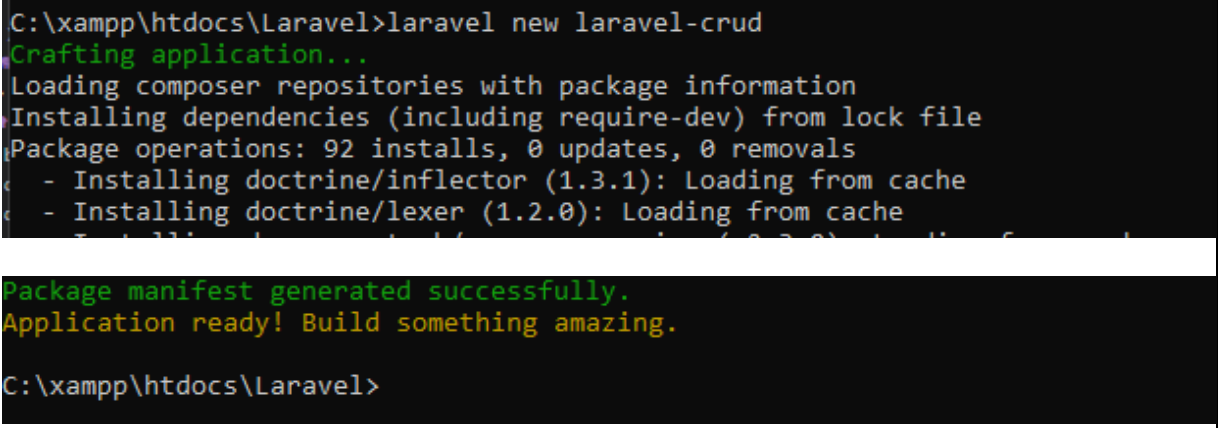
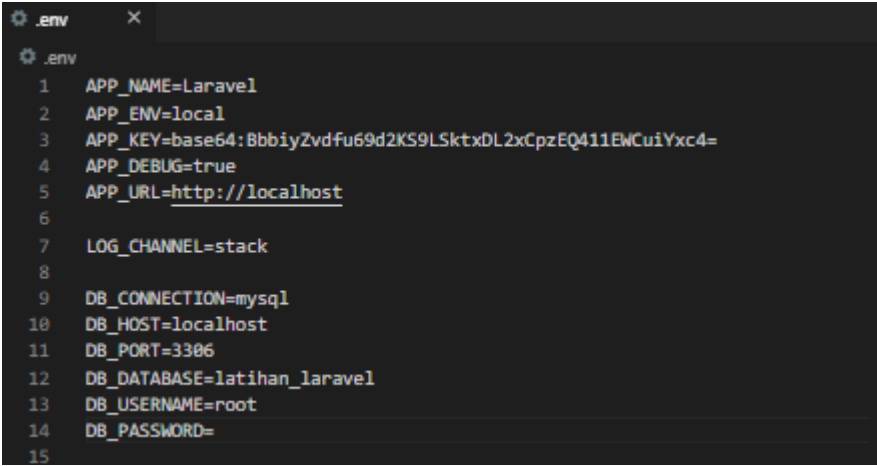




## Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

Eloquent adalah sebuah fitur untuk mengelola data yang ada pada database. Eloquent ORM menyediakan fungsi-fungsi active record, atau fungsi-fungsi query sql untuk mengelola data pada database. Dengan menggunakan Eloquent pada Laravel, kita bisa mengelola data pada database dari hanya satu buah model.

### a. Konfigurasi Database

Langkah	Keterangan
1	<p>Buatlah project Laravel baru dengan nama <b>laravel-crud-kedua</b>. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud-kedua</pre>  <p>The screenshot shows the terminal output of the 'laravel new laravel-crud' command. It starts with 'Crafting application...', followed by 'Loading composer repositories with package information' and 'Installing dependencies (including require-dev) from lock file'. It then shows 'Package operations: 92 installs, 0 updates, 0 removals' and lists two packages being installed from cache: 'doctrine/inflector (1.3.1)' and 'doctrine/lexer (1.2.0)'. The output ends with 'Package manifest generated successfully.' and 'Application ready! Build something amazing.'</p>
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file <b>.env</b> pada project laravel-crud. Ubah seperti di bawah ini.</p>  <p>The screenshot shows the contents of the .env file. It lists various configuration variables for the application, including APP_NAME, APP_ENV, APP_KEY, APP_DEBUG, APP_URL, LOG_CHANNEL, DB_CONNECTION, DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, and DB_PASSWORD. The DB_DATABASE is set to 'latihan_laravel' and DB_USERNAME is set to 'root'.</p> <p>Keterangan:</p> <ul style="list-style-type: none"><li>- Nama database yang akan digunakan adalah latihan_laravel dengan username root.</li></ul>

3

Pada project ini kita gunakan database dan tabel yang sebelumnya digunakan pada Praktikum Bagian 1. Tambahkan kolom `created_at` dan `updated_at` yang bertipe `TIMESTAMP` dan default nilainya `NULL`

#	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
6	created_at	timestamp			Yes	NULL			Change Drop More
7	updated_at	timestamp			Yes	NULL			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.</p> <p>Pertama, buatlah <i>route</i> pada <b>routes/web.php</b> sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p> <pre> 16 Route::get('/', 'MahasiswaController@index'); 17 Route::get('/mahasiswa', 'MahasiswaController@index'); 18 </pre>
2	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan</p> <p><b>cd laravel-crud-kedua</b>  <b>php artisan make:model Mahasiswa</b></p>
3	<p>Ubah model <b>Mahasiswa.php</b> pada folder <b>App</b> menjadi seperti berikut.</p> <pre> app &gt; Mahasiswa.php 1 &lt;?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:</p> <ul style="list-style-type: none"> <li>- Model Mahasiswa akan menangani tabel mahasiswa</li> </ul>

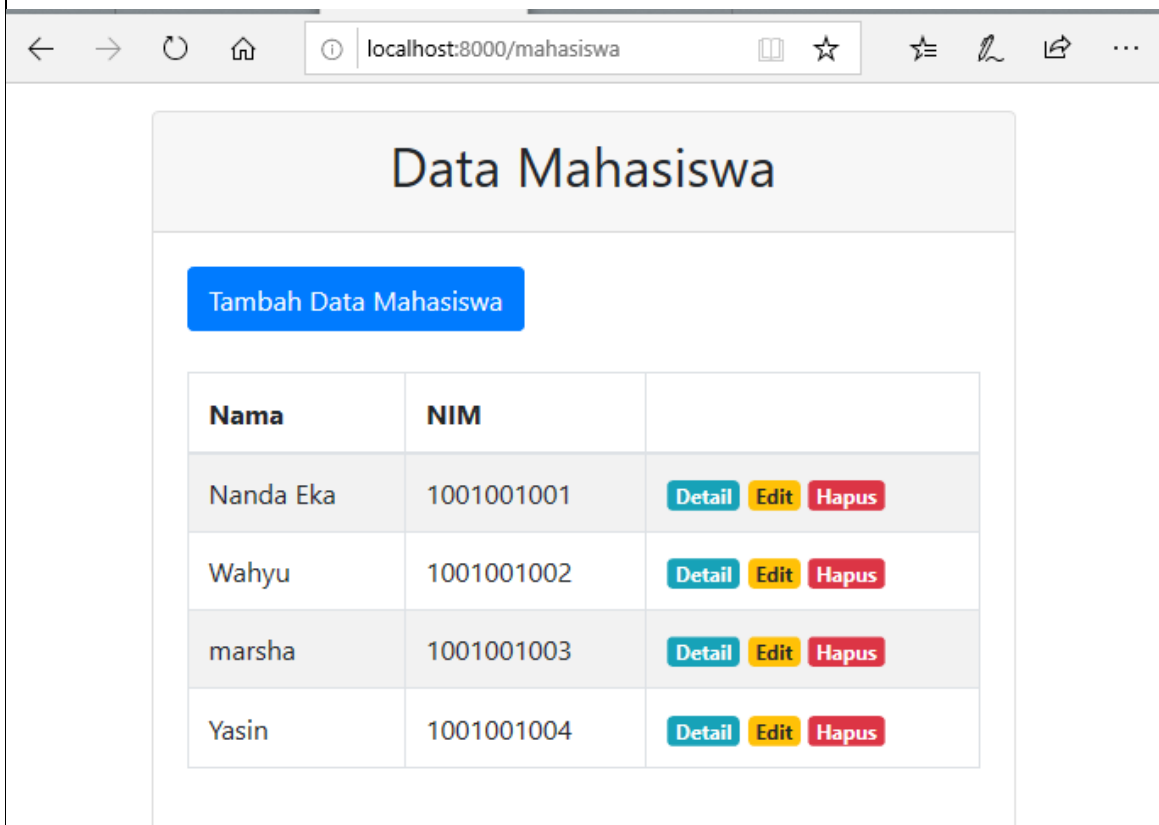
4	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan <b>php artisan make:controller MahasiswaController</b></p>
5	<p>Buat method <b>index</b> pada MahasiswaController.php pada folder <b>app/Http/Controllers</b></p> <pre data-bbox="494 241 1287 696"> app &gt; Http &gt; Controllers &gt; MahasiswaController.php 1  &lt;?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         \$mahasiswa = Mahasiswa::all(); 12         return view('index', ['mahasiswa' =&gt; \$mahasiswa]); 13     } 14 } 15 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Tambahkan 'use App\Mahasiswa' (line 6) untuk menggunakan model Mahasiswa</li> <li>- Line 12 untuk mengambil semua data dari model/tabel Mahasiswa dan akan disimpan di variabel \$mahasiswa</li> <li>- Line 16 : data akan dikirim ke blade view bernama index</li> </ul>
6	<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 <b>template blade</b> (seperti pada Bagian 1).</p> <p>Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)</p> <pre data-bbox="279 1133 1442 1771"> resources &gt; views &gt; master.blade.php 1  &lt;html lang="en"&gt; 2  &lt;head&gt; 3      &lt;meta charset="utf-8"&gt; 4      &lt;meta name="viewport" content="width=device-width, initial-scale=1"&gt; 5      &lt;link href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" rel="stylesheet"&gt; 6      &lt;title&gt; @yield('title') &lt;/title&gt; 7  &lt;/head&gt; 8  &lt;body&gt; 9      &lt;div class="container"&gt; 10         &lt;div class="row mt-3"&gt; 11             &lt;div class="col-md-6"&gt; 12                 &lt;div class="card"&gt; 13                     &lt;div class="card-header text-center"&gt; 14                         &lt;!-- bagian judul halaman --&gt; 15                         &lt;h2&gt; @yield('judul_halaman') &lt;/h2&gt; 16                     &lt;/div&gt; 17                     &lt;div class="card-body"&gt; 18                         &lt;!-- bagian konten blog --&gt; 19                         @yield('konten') 20                     &lt;/div&gt; 21                 &lt;/div&gt; 22             &lt;/div&gt; 23         &lt;/div&gt; 24     &lt;/div&gt; 25 &lt;/body&gt; 26 &lt;/html&gt; </pre>

```
resources > views > index.blade.php
2
3 <!-- isi title -->
4 @section('title', 'Home')
5
6 <!-- isi bagian judul halaman -->
7 @section('judul_halaman', 'Data Mahasiswa')
8
9 <!-- isi bagian konten -->
10 @section('konten')
11 <a href="/mahasiswa/tambah" class="btn btn-primary">Tambah Data Mahasiswa</a>
12 <br/>
13 <br/>
14 <table class="table table-bordered table-hover table-striped">
15 <thead>
16 <tr>
17 <th>Nama</th>
18 <th>NIM</th>
19 <th></th>
20 </tr>
21 </thead>
22 <tbody>
23 @foreach($mahasiswa as $mhs)
24 <tr>
25 <td>{{ $mhs->nama }}</td>
26 <td>{{ $mhs->nim }}</td>
27 <td>
28 <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a>
29 <a href="/mahasiswa/edit/{{ $mhs->id }}" class="badge badge-warning">Edit</a>
30 <a href="/mahasiswa/hapus/{{ $mhs->id }}" class="badge badge-danger">Hapus</a>
31 </td>
32 </tr>
33 @endforeach
34 </tbody>
35 </table>
36 @endsection
```

7

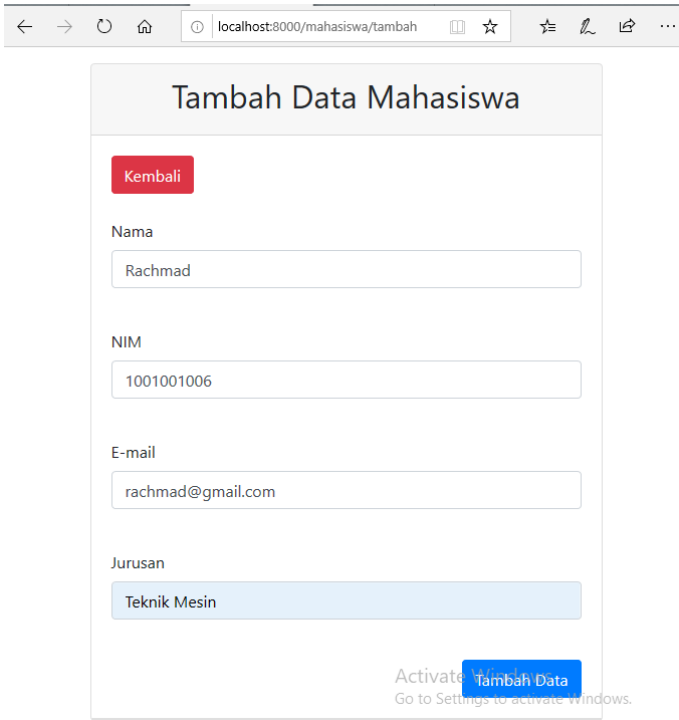
Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud  
**php artisan serve**

Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut

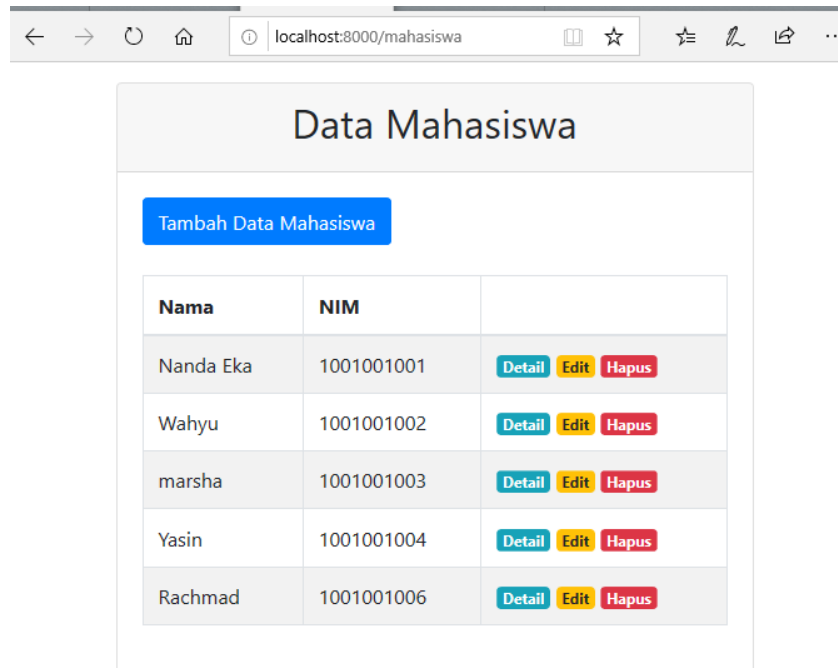


### c. Memasukkan Data (Create) ke Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/tambah</b> yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre> 18 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 19 </pre>
2	<p>Buat method <b>tambah</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view tambah.</p> <pre> 15 public function tambah(){ 16     return view('tambah'); 17 } </pre>
3	<p>Kemudian buatlah view <b>tambah.blade.php</b> yang berisi form untuk memasukkan data baru pada folder <b>resources/views</b>.</p> <p>Kita lakukan copy paste dari tambah.blade.php di Bagian 1</p> <pre> resources &gt; views &gt; tambah.blade.php 1 @extends('master') 2 &lt;!-- isi title --&gt; 3 @section('title', 'Tambah Data') 4 5 &lt;!-- isi bagian judul halaman --&gt; 6 @section('judul_halaman', 'Tambah Data Mahasiswa') 7 8 &lt;!-- isi bagian konten --&gt; 9 @section('konten') 10 &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 11 &lt;br/&gt; 12 &lt;br/&gt; 13 &lt;form action="/mahasiswa/simpan" method="post"&gt; 14     {{ csrf_field() }} 15     &lt;div class="form-group"&gt; 16         &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 17         &lt;input type="text" class="form-control" required="required" name="namamhs"&gt; &lt;br/&gt; 18     &lt;/div&gt; 19     &lt;div class="form-group"&gt; 20         &lt;label for="nimhs"&gt;NIM&lt;/label&gt; 21         &lt;input type="number" class="form-control" required="required" name="nimhs"&gt; &lt;br/&gt; 22     &lt;/div&gt; 23     &lt;div class="form-group"&gt; 24         &lt;label for="emailmhs"&gt;E-mail&lt;/label&gt; 25         &lt;input type="email" class="form-control" required="required" name="emailmhs"&gt; &lt;br/&gt; 26     &lt;/div&gt; 27     &lt;div class="form-group"&gt; 28         &lt;label for="jurusanmhs"&gt;Jurusan&lt;/label&gt; 29         &lt;input type="text" class="form-control" required="required" name="jurusanmhs"&gt; &lt;br/&gt; 30     &lt;/div&gt; 31     &lt;button type="submit" name="tambah" class="btn btn-primary float-right"&gt;Tambah Data&lt;/button&gt; 32 &lt;/form&gt; 33 @endsection </pre>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/simpan</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 19 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 20 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php</li> </ul>

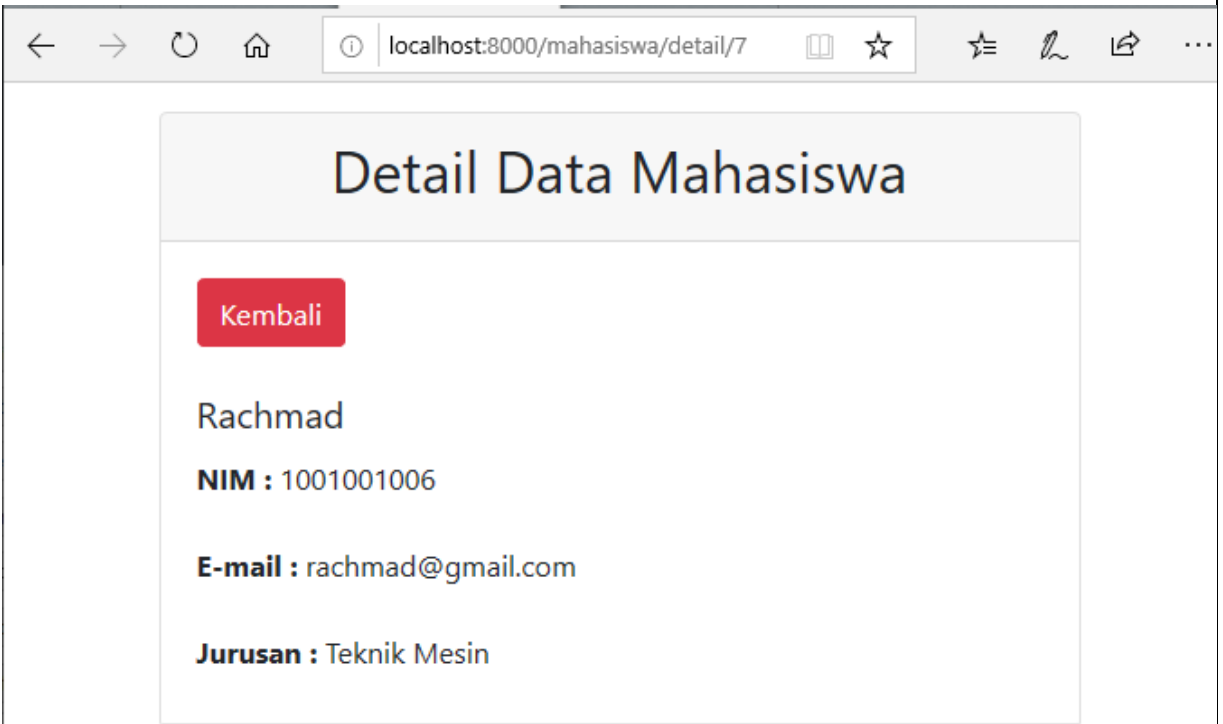
5	<p>Buat method <b>simpan</b> pada <b>MahasiswaController</b> untuk menyimpan data ke database.</p> <pre> 19     public function simpan(Request \$request){ 20         Mahasiswa::create([ 21             'nama' =&gt; \$request-&gt;namamhs, 22             'nim' =&gt; \$request-&gt;nimmhs, 23             'email' =&gt; \$request-&gt;emailmhs, 24             'jurusan' =&gt; \$request-&gt;jurusanmhs 25         ]); 26         return redirect('/mahasiswa'); 27     } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li> <li>- Line 23-28 merupakan fitur eloquent menggunakan fungsi create() untuk insert data ke tabel mahasiswa</li> </ul>
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 &gt; Mahasiswa.php 1  &lt;?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> 

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



#### d. Melihat Detail Data dari Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/detail</b> yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre>20 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 21</pre>
3	<p>Buat method <b>detail</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan data mahasiswa pada view detail.blade.php.</p> <pre>29 public function detail(\$id){ 30     \$mahasiswa = Mahasiswa::find(\$id); 31     return view('detail', ['mahasiswa' =&gt; \$mahasiswa]); 32 } 33</pre>
4	<p>Kemudian buatlah view <b>detail.blade.php</b> yang menampilkan detail data mahasiswa pada folder <b>resources/views</b>. View detail juga mengaplikasikan master.blade.php.</p>

	<pre>resources &gt; views &gt; detail.blade.php 1 @extends('master') 2 &lt;!-- isi title --&gt; 3 @section('title', 'Detail Mahasiswa') 4 5 &lt;!-- isi bagian judul halaman --&gt; 6 @section('judul_halaman', 'Detail Data Mahasiswa') 7 8 &lt;!-- isi bagian konten --&gt; 9 @section('konten') 10 &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 11 12 &lt;br/&gt; 13 &lt;br/&gt; 14 15 &lt;h5 class="card-title"&gt;{{ \$mahasiswa-&gt;nama }} &lt;/h5&gt; 16 &lt;p class="card-text"&gt; 17 &lt;label for=""&gt;&lt;b&gt; NIM : &lt;/b&gt;&lt;/label&gt; 18 {{ \$mahasiswa-&gt;nim }} &lt;/p&gt; 19 &lt;p class="card-text"&gt; 20 &lt;label for=""&gt;&lt;b&gt; E-mail : &lt;/b&gt;&lt;/label&gt; 21 {{ \$mahasiswa-&gt;email }} &lt;/p&gt; 22 &lt;p class="card-text"&gt; 23 &lt;label for=""&gt;&lt;b&gt; Jurusan : &lt;/b&gt;&lt;/label&gt; 24 {{ \$mahasiswa-&gt;jurusan }} &lt;/p&gt; 25 @endsection</pre>
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	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/edit</b> yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre>21 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');</pre>
2	<p>Buat method <b>edit</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view edit.</p>



	<pre> 34     public function edit(\$id){ 35         \$mahasiswa = Mahasiswa::find(\$id); 36         return view('edit',['mahasiswa' =&gt; \$mahasiswa]); 37     } 38 </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih</li> </ul>
3	<p>Kemudian buatlah view <b>edit.blade.php</b> yang berisi form untuk mengubah data pada folder <b>resources/views</b>. View edit juga mengaplikasikan master.blade.php.</p> <pre> resources &gt; views &gt; edit.blade.php 1  @extends('master') 2  &lt;!-- isi title --&gt; 3  @section('title', 'Edit Data') 4 5  &lt;!-- isi bagian judul halaman --&gt; 6  @section('judul_halaman', 'Edit Data Mahasiswa') 7 8  &lt;!-- isi bagian konten --&gt; 9  @section('konten') 10     &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 11     &lt;br/&gt; 12     &lt;br/&gt; 13     &lt;form action="/mahasiswa/update/{{ \$mahasiswa-&gt;id }}" method="post"&gt; 14         {{ csrf_field() }} 15         &lt;input type="hidden" name="id" value="{{ \$mahasiswa-&gt;id }}"&gt; &lt;br/&gt; 16         &lt;div class="form-group"&gt; 17             &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 18             &lt;input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mahasiswa-&gt;nama }}"&gt; &lt;br/&gt; 19         &lt;/div&gt; 20         &lt;div class="form-group"&gt; 21             &lt;label for="nimhs"&gt;NIM&lt;/label&gt; 22             &lt;input type="number" class="form-control" required="required" name="nimhs" value="{{ \$mahasiswa-&gt;nim }}"&gt; &lt;br/&gt; 23         &lt;/div&gt; 24         &lt;div class="form-group"&gt; 25             &lt;label for="emailmhs"&gt;E-mail&lt;/label&gt; 26             &lt;input type="email" class="form-control" required="required" name="emailmhs" value="{{ \$mahasiswa-&gt;email }}"&gt; &lt;br/&gt; 27         &lt;/div&gt; 28         &lt;div class="form-group"&gt; 29             &lt;label for="jurusanmhs"&gt;Jurusan&lt;/label&gt; 30             &lt;input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mahasiswa-&gt;jurusan }}"&gt; &lt;br/&gt; 31         &lt;/div&gt; 32         &lt;button type="submit" name="edit" class="btn btn-primary float-right"&gt;Simpan Data&lt;/button&gt; 33     &lt;/form&gt; 34 @endsection </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 11-31 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan</li> <li>- Line 11 terdapat action="/mahasiswa/update/{{ \$mahasiswa-&gt;id }}" yang menunjukkan routes /mahasiswa/update/{id} dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController</li> </ul>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/update/{id}</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 22 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update'); 23 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php</li> </ul>
5	<p>Buat method <b>update</b> pada <b>MahasiswaController</b> untuk menyimpan data yang diubah ke database.</p>

```

39 public function update($id, Request $request){
40     $mahasiswa = Mahasiswa::find($id);
41     $mahasiswa->nama = $request->namamhs;
42     $mahasiswa->nim = $request->nimmhs;
43     $mahasiswa->email = $request->emailmhs;
44     $mahasiswa->jurusan = $request->jurusanmhs;
45     $mahasiswa->save();
46     return redirect('/mahasiswa');
47 }
48

```

Keterangan:

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

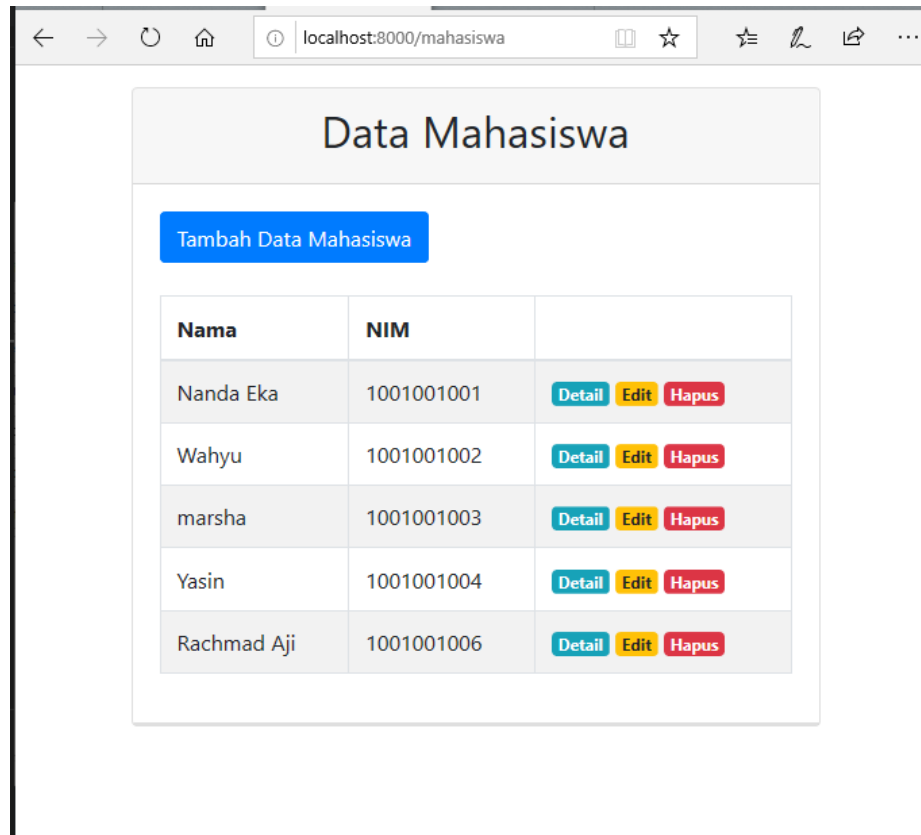
6

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/7'. The main content area is titled 'Edit Data Mahasiswa'. It features a form with the following elements:

- A red button labeled 'Kembali' (Back) at the top left.
- A text input field for 'Nama' (Name) containing 'Rachmad Aji'.
- A text input field for 'NIM' containing '1001001006'.
- A text input field for 'E-mail' containing 'rachmad@gmail.com'.
- A text input field for 'Jurusan' (Department) containing 'Teknik Mesin'.
- A blue button labeled 'Simpan Data' (Save Data) at the bottom right.

An 'Activate Windows' watermark is visible in the bottom right corner of the browser window.



#### f. Menghapus Data (Delete) dari Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/hapus</b> yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> 23 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 24 25 </pre>
2	<p>Buat method <b>hapus</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menjalankan fungsi hapus.</p> <pre> 49 public function hapus(\$id){ 50     \$mahasiswa = Mahasiswa::find(\$id); 51     \$mahasiswa-&gt;delete(); 52     return redirect('/mahasiswa'); 53 } 54 </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih</li> </ul>
3	Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.

← → ↻ 🏠 ⓘ localhost:8000/mahasiswa 📖 ☆ ⚙️ 🖋️ 📄 ⋮

### Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM	
Nanda Eka	1001001001	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
Wahyu	1001001002	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
marsha	1001001003	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
Yasin	1001001004	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
Rachmad Aji	1001001006	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Setelah data dihapus :

← → ↻ 🏠 ⓘ localhost:8000/mahasiswa 📖 ☆ ⚙️ 🖋️ 📄 ⋮

### Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM	
Nanda Eka	1001001001	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
Wahyu	1001001002	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
marsha	1001001003	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
Yasin	1001001004	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

-- Selamat Mengerjakan --