

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT**

**Jobsheet - 10:
Membuat CRUD menggunakan Laravel**



Disusun oleh:
Livia Yurike Khuril Maula
D4 TI – 2A / 16
NIM : 1841720025

**POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D4 TEKNIK INFORMATIKA**

APRIL 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

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> 
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> 

3

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

	#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
<input type="checkbox"/>	1	id	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
<input type="checkbox"/>	2	nama	varchar(100)			Tidak	Tidak ada	
<input type="checkbox"/>	3	nim	varchar(10)			Tidak	Tidak ada	
<input type="checkbox"/>	4	email	varchar(50)			Tidak	Tidak ada	
<input type="checkbox"/>	5	jurusan	varchar(20)			Tidak	Tidak ada	

4

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

		id	nama	nim	email	jurusan
<input type="checkbox"/>		1	nanda	1001001001	nandan@gmail.com	Teknik Informatika
<input type="checkbox"/>		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.</p> <p>Pertama, buatlah <i>route</i> pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p> <pre> routes > web.php 1 <?php 2 3 /* 4 ----- 5 Web Routes 6 ----- 7 8 Here is where you can register web routes for your application. These 9 routes are loaded by the RouteServiceProvider within a group which 10 contains the "web" middleware group. Now create something great! 11 12 */ 13 14 Route::get('/', 'MahasiswaController@index'); 15 </pre>
2	<p>Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan</p> <p>cd laravel-crud</p> <p>php artisan make:controller MahasiswaController</p> <pre> c:\xampp\htdocs>cd laravel-crud c:\xampp\htdocs\laravel-crud>php artisan make:controller MahasiswaController Controller created successfully. c:\xampp\htdocs\laravel-crud>_ </pre>

3

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

```
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 }
```

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**.

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

5

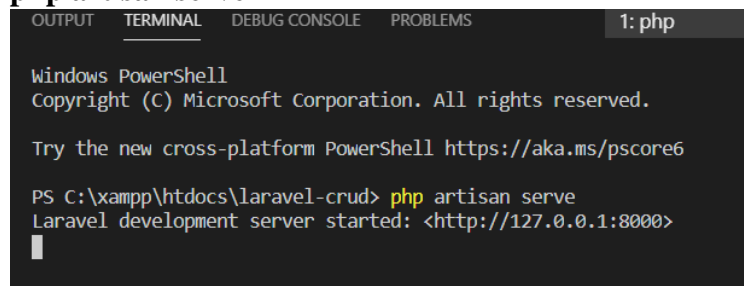
Sekarang buat file **index.blade.php** yang menerapkan template master.blade.php dan akan digunakan untuk menampilkan data.

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

6

Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud

php artisan serve

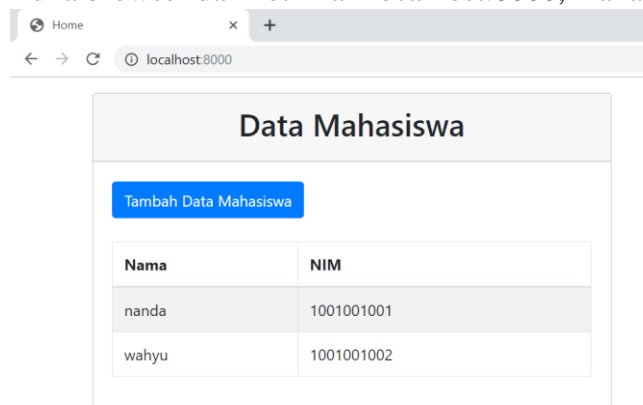


```
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS  1: php
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\laravel-crud> php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

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> 15 16 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 17 </pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> 19 public function tambah() 20 { 21 // memanggil view tambah 22 return view('tambah'); 23 } </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> <pre> resources > views > tambah.blade.php > ... 1 @extends('master') 2 <!-- isi title --> 3 @section('title', 'Tambah Data') 4 5 <!-- isi bagian judul halaman --> 6 @section('judul_halaman', 'Tambah Data Mahasiswa') 7 8 <!-- isi bagian konten --> 9 @section('konten') 10 Kembali 11
 12
 13 <form action="/mahasiswa/simpan" method="post"> 14 {{ csrf_field() }} 15 <div class="form-group"> 16 <label for="namamhs">Nama</label> 17 <input type="text" class="form-control" require="require" name="namamhs">
 18 </div> 19 <div class="form-group"> 20 <label for="nimhs">NIM</label> 21 <input type="number" class="form-control" require="require" name="nimhs">
 22 </div> 23 <div class="form-group"> 24 <label for="emailmhs">E-mail</label> 25 <input type="email" class="form-control" require="require" name="emailmhs">
 26 </div> 27 <div class="form-group"> 28 <label for="jurusanmhs">Jurusan</label> 29 <input type="text" class="form-control" require="require" name="jurusanmhs">
 30 </div> 31 <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button> 32 </form> 33 @endsection </pre>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 17 18 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 19 </pre>

5

Buat method simpan pada **MahasiswaController** untuk menyimpan data ke database.

```

25 public function simpan(Request $request)
26 {
27     // insert data ke table mahasiswa
28     DB::table('mahasiswa')->insert([
29         'nama' => $request->namamhs,
30         'nim' => $request->nimmhs,
31         'email' => $request->emailmhs,
32         'jurusan' => $request->jurusanmhs
33     ]);
34     return redirect('/mahasiswa');
35 }

```

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 two browser windows. The top window, titled 'Tambah Data', displays a form for adding a new student. The form fields are filled with the following data:

Field	Value
Nama	marsha
NIM	1001001003
E-mail	marsha@gmailcom
Jurusan	teknik sipil

Below the form is a blue button labeled 'Tambah Data'. A red button labeled 'Kembali' is at the top left of the form area.

The bottom window, titled 'Home', displays a table of existing students under the heading 'Data Mahasiswa'.

Nama	NIM
nanda	1001001001
wahyu	1001001002
marsha	1001001003

A blue button labeled 'Tambah Data Mahasiswa' is located above the table.

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> resources > views > index.blade.php > table.table-bordered.table-hover.table-striped > tbody > tr > td 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 Tambah Data Mahasiswa 12
 13
 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 id }}" class="badge badge-info">Detail 29 </td> 30 </tr> 31 @endforeach 32 </tbody> 33 </table> 34 @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/detail/{id}', 'MahasiswaController@detail'); 21 </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> 37 public function detail(\$id) 38 { 39 // mengambil data mahasiswa berdasarkan id yang dipilih 40 \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); 41 // kirim data mahasiswa yang diambil ke view edit.blade.php 42 return view('detail', ['mahasiswa' => \$mahasiswa]); 43 } </pre>

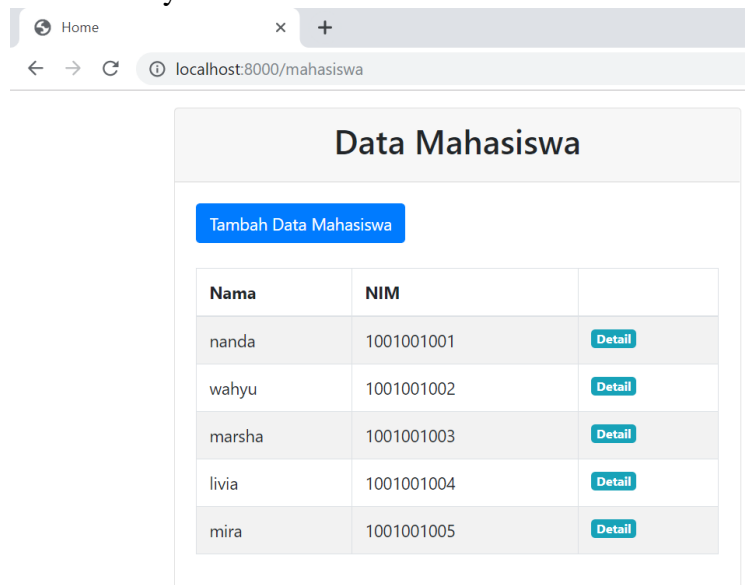
4

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
```

5

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



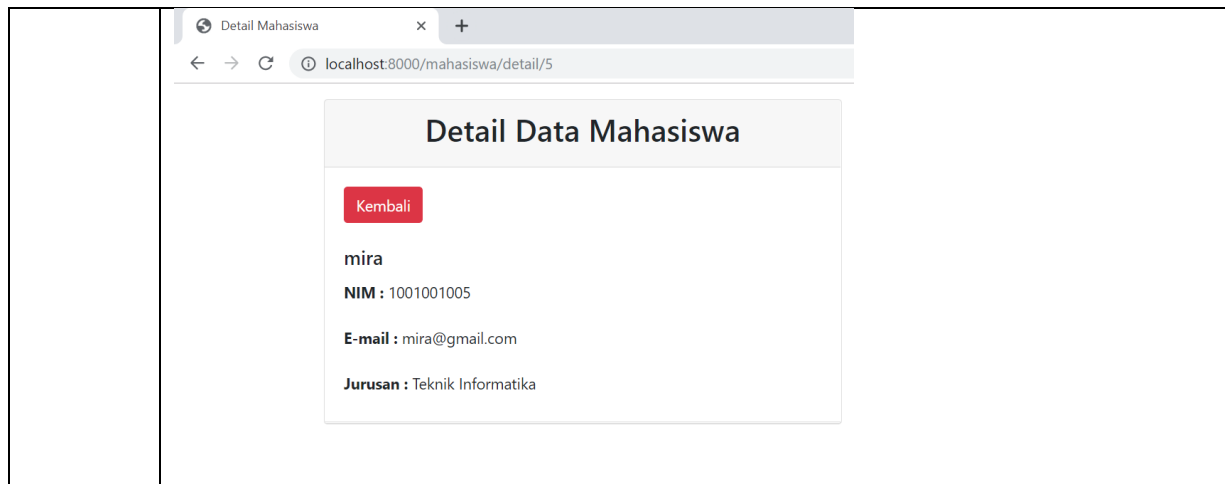
Home x +

localhost:8000/mahasiswa

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda	1001001001	Detail
wahyu	1001001002	Detail
marsha	1001001003	Detail
livia	1001001004	Detail
mira	1001001005	Detail



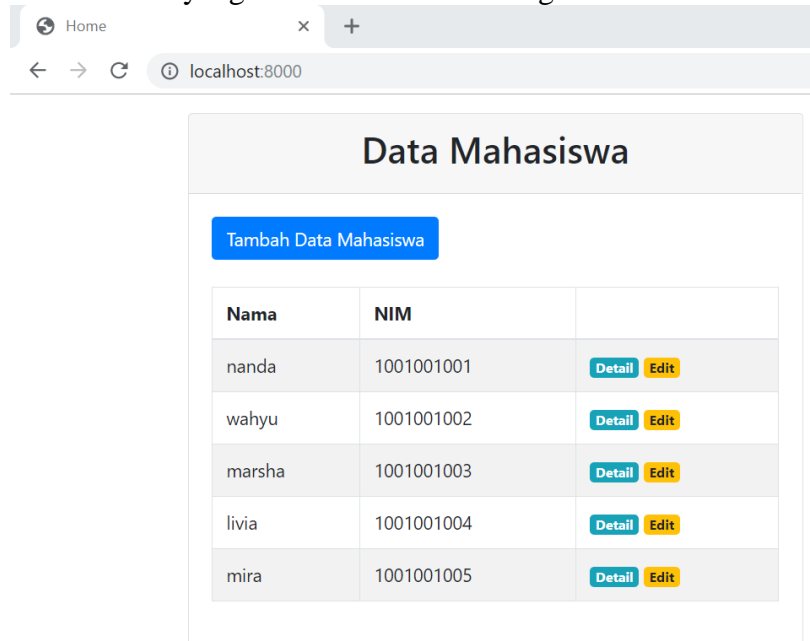
e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views</p> <pre> 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 Tambah Data Mahasiswa 12
 13
 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 id }}" class="badge badge-info">Detail 29 id }}" class="badge badge-warning">Edit 30 </td> 31 </tr> 32 @endforeach 33 </tbody> 34 </table> 35 @endsection </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre> 23 24 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 25 </pre>

3	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre> 45 public function edit(\$id) 46 { 47 // mengambil data mahasiswa berdasarkan id yang dipilih 48 \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); 49 //kirim data mahasiswa yang diambil ke view edit.blade.php 50 return view('edit', ["mahasiswa" => \$mahasiswa]); 51 } </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 > form > div.form-group 1 @extends('master') 2 3 <!-- isi title --> 4 @section('title', 'Edit Data') 5 6 <!-- isi bagian judul halaman --> 7 @section('judul_halaman', 'Edit Data Mahasiswa') 8 9 <!-- isi bagian konten --> 10 @section('konten') 11 Kembali 12
 13
 14 @foreach(\$mahasiswa as \$mhs) 15 <form action="/mahasiswa/update" method="post"> 16 {{ csrf_field() }} 17 <input type="hidden" name="id" value="{{ \$mhs->id }}">
 18 <div class="form-group"> 19 <label for="namamhs">Nama</label> 20 <input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mhs->nama }}">
 21 </div> 22 <div class="form-group"> 23 <label for="nimhs">NIM</label> 24 <input type="number" class="form-control" required="required" name="nimhs" value="{{ \$mhs->nim }}">
 25 </div> 26 <div class="form-group"> 27 <label for="emailhs">E-mail</label> 28 <input type="email" class="form-control" required="required" name="emailhs" value="{{ \$mhs->email }}">
 29 </div> 30 <div class="form-group"> 31 <label for="jurusanhs">Jurusan</label> 32 <input type="text" class="form-control" required="required" name="jurusanhs" value="{{ \$mhs->jurusan }}">
 33 </div> 34 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button> 35 </form> 36 @endforeach 37 @endsection </pre>
5	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 25 26 Route::post('/mahasiswa/update', 'MahasiswaController@update'); 27 </pre>
6	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p> <pre> 53 public function update(Request \$request) 54 { 55 // update data mahasiswa 56 DB::table('mahasiswa')->where('id', \$request->id)->update([57 'nama' => \$request->namamhs, 58 'nim' => \$request->nimmhs, 59 'email' => \$request->emailmhs, 60 'jurusan' => \$request->jurusanmhs 61]); 62 return redirect('/mahasiswa'); 63 } </pre>

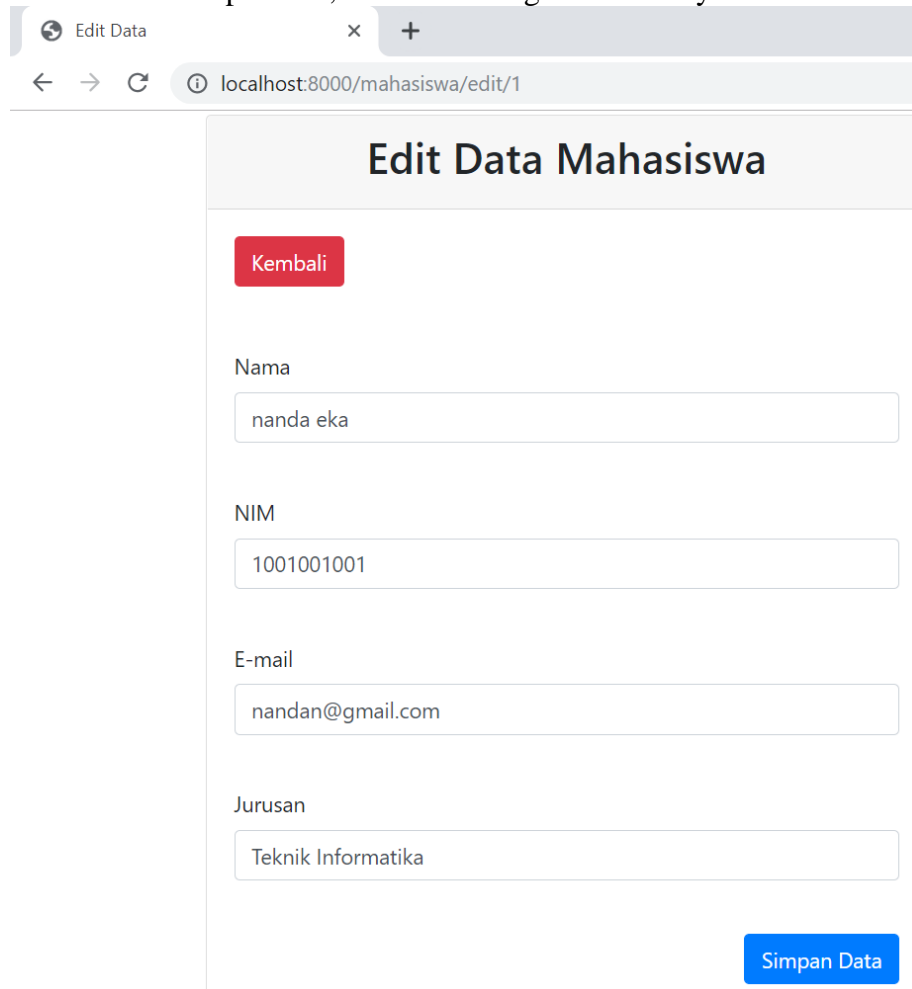
7

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



Nama	NIM	
nanda	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
marsha	1001001003	Detail Edit
livia	1001001004	Detail Edit
mira	1001001005	Detail Edit

Klik edit di data pertama, kita akan mengubah namanya.



[Kembali](#)

Nama

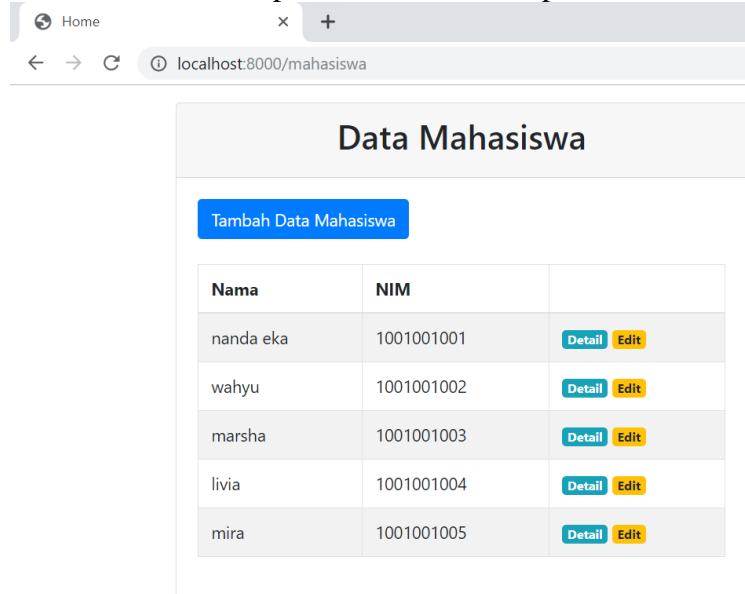
NIM

E-mail

Jurusan

[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> resources > views > index.blade.php > table.table-bordered.table-hover.table-striped > tbody > tr > td > a.badge.badge-info 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 Tambah Data Mahasiswa 12
 13
 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 id }}" class="badge badge-info">Detail 29 id }}" class="badge badge-warning">Edit 30 id }}" class="badge badge-danger">Hapus 31 </td> 32 </tr> 33 @endforeach 34 </tbody> 35 </table> 36 @endsection </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> 27 28 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 29 </pre>

3

Buat method hapus pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menjalankan fungsi hapus.

```

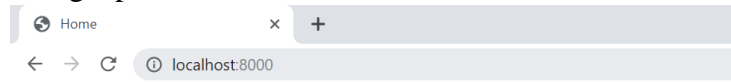
65     public function hapus($id)
66     {
67         // menghapus data mahasiswa berdasarkan id yang dipilih
68         DB::table('mahasiswa')->where('id', $id)->delete();
69
70         return redirect('/mahasiswa');
71     }

```

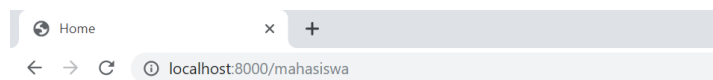
4

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

Menghapus data terakhir



Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
livia	1001001004	Detail Edit Hapus
mira	1001001005	Detail Edit Hapus



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

Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

a. Konfigurasi Database

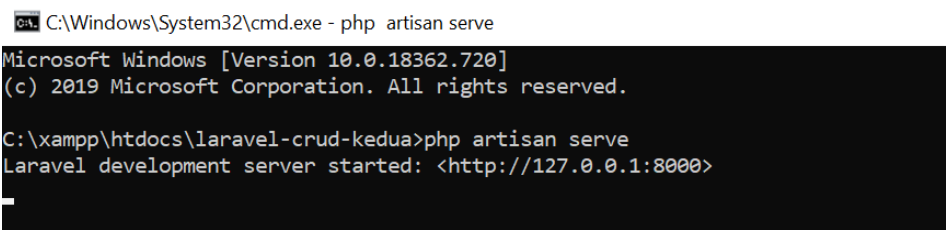
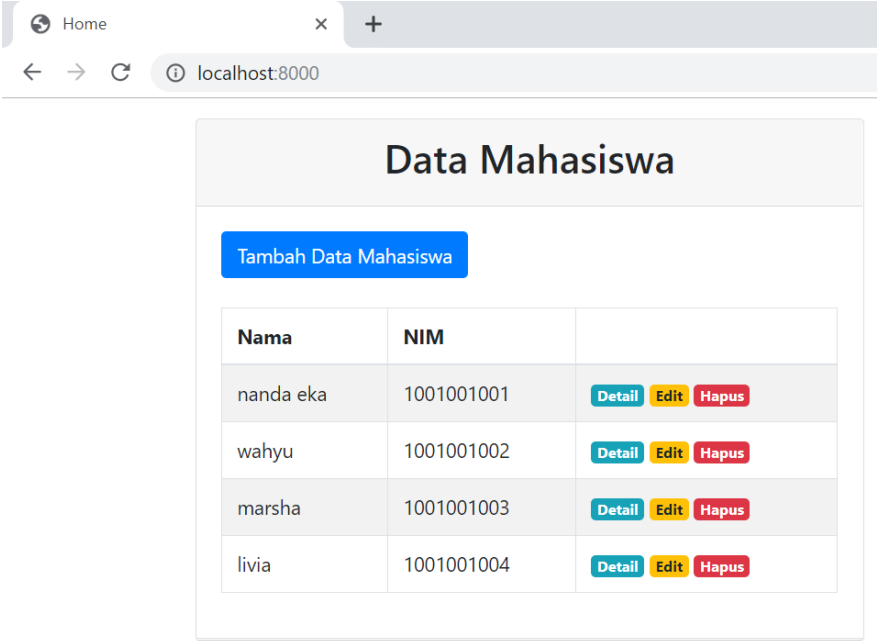
Langkah	Keterangan
1	<p>Buatlah project Laravel baru dengan nama laravel-crud-kedua. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud-kedua</pre> 
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.</p> 

3	<p>Pada project ini kita gunakan database dan tabel yang sebelumnya digunakan pada Praktikum Bagian 1. Tambahkan kolom <code>created_at</code> dan <code>updated_at</code> yang bertipe <code>TIMESTAMP</code> dan default nilainya <code>NULL</code></p> <table border="1"> <thead> <tr> <th></th><th>#</th><th>Nama</th><th>Jenis</th><th>Penyortiran</th><th>Atribut</th><th>Kosong</th><th>Bawaan</th><th>Ekstra</th></tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td><td>1</td><td>id </td><td>int(11)</td><td></td><td></td><td>Tidak</td><td>Tidak ada</td><td>AUTO_INCREMENT</td></tr> <tr> <td><input type="checkbox"/></td><td>2</td><td>nama</td><td>varchar(100)</td><td></td><td></td><td>Tidak</td><td>Tidak ada</td><td></td></tr> <tr> <td><input type="checkbox"/></td><td>3</td><td>nim</td><td>varchar(10)</td><td></td><td></td><td>Tidak</td><td>Tidak ada</td><td></td></tr> <tr> <td><input type="checkbox"/></td><td>4</td><td>email</td><td>varchar(50)</td><td></td><td></td><td>Tidak</td><td>Tidak ada</td><td></td></tr> <tr> <td><input type="checkbox"/></td><td>5</td><td>jurusan</td><td>varchar(20)</td><td></td><td></td><td>Tidak</td><td>Tidak ada</td><td></td></tr> <tr> <td><input type="checkbox"/></td><td>6</td><td>created_at</td><td>timestamp</td><td></td><td></td><td>Ya</td><td>NULL</td><td></td></tr> <tr> <td><input type="checkbox"/></td><td>7</td><td>updated_at</td><td>timestamp</td><td></td><td></td><td>Ya</td><td>NULL</td><td></td></tr> </tbody> </table>									#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	<input type="checkbox"/>	1	id	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	<input type="checkbox"/>	2	nama	varchar(100)			Tidak	Tidak ada		<input type="checkbox"/>	3	nim	varchar(10)			Tidak	Tidak ada		<input type="checkbox"/>	4	email	varchar(50)			Tidak	Tidak ada		<input type="checkbox"/>	5	jurusan	varchar(20)			Tidak	Tidak ada		<input type="checkbox"/>	6	created_at	timestamp			Ya	NULL		<input type="checkbox"/>	7	updated_at	timestamp			Ya	NULL	
	#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra																																																																								
<input type="checkbox"/>	1	id	int(11)			Tidak	Tidak ada	AUTO_INCREMENT																																																																								
<input type="checkbox"/>	2	nama	varchar(100)			Tidak	Tidak ada																																																																									
<input type="checkbox"/>	3	nim	varchar(10)			Tidak	Tidak ada																																																																									
<input type="checkbox"/>	4	email	varchar(50)			Tidak	Tidak ada																																																																									
<input type="checkbox"/>	5	jurusan	varchar(20)			Tidak	Tidak ada																																																																									
<input type="checkbox"/>	6	created_at	timestamp			Ya	NULL																																																																									
<input type="checkbox"/>	7	updated_at	timestamp			Ya	NULL																																																																									

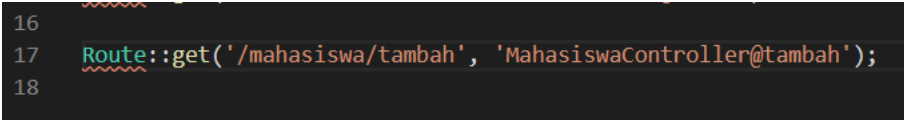
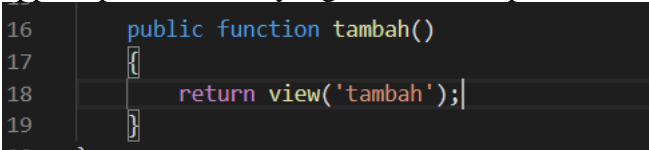
b. Menampilkan Data dari Database

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> 14 Route::get('/', 'MahasiswaController@index'); 15 Route::get('/mahasiswa', 'MahasiswaController@index'); 16 </pre>
2	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan</p> <p>cd laravel-crud-kedua</p> <p>php artisan make:model Mahasiswa</p> <pre> c:\xampp\htdocs>cd laravel-crud-kedua c:\xampp\htdocs\laravel-crud-kedua>php artisan make:model Mahasiswa Model created successfully. c:\xampp\htdocs\laravel-crud-kedua>_ </pre>
3	<p>Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.</p> <pre> app > Mahasiswa.php > Mahasiswa 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>

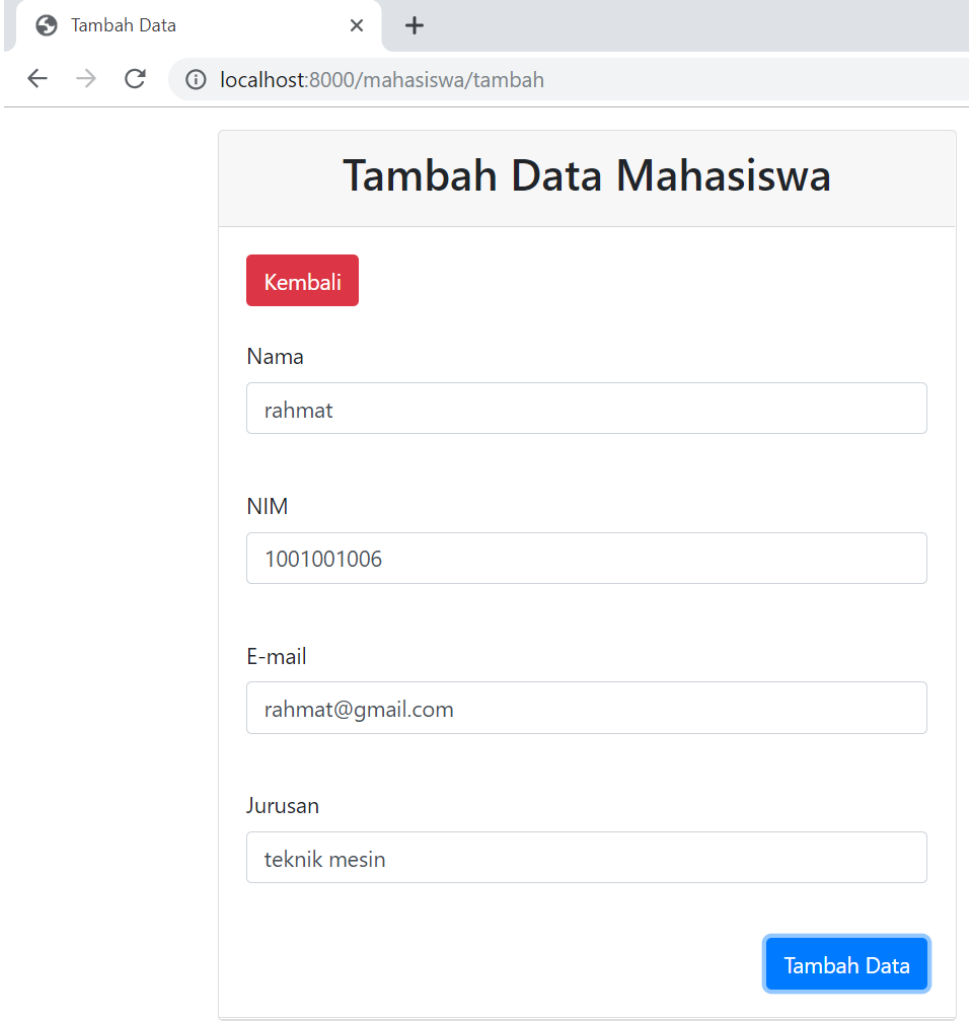
4	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan php artisan make:controller MahasiswaController</p> <pre>c:\xampp\htdocs\laravel-crud-kedua>php artisan make:controller MahasiswaController Controller created successfully. c:\xampp\htdocs\laravel-crud-kedua></pre>
5	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p> <pre>app > Http > Controllers > MahasiswaController.php > index 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 11 public function index() 12 { 13 \$mahasiswa = Mahasiswa::all(); 14 return view('index', ['mahasiswa' => \$mahasiswa]); 15 } 16 }</pre>
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 template blade (seperti pada Bagian 1). Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)</p> <pre>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 Tambah Data Mahasiswa 12
 13
 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 <td> 27 id }}" class="badge badge-info">Detail 28 id }}" class="badge badge-warning">Edit 29 id }}" class="badge badge-danger">Hapus 30 </td> 31 </tr> 32 @endforeach 33 </tbody> 34 </table> 35 @endsection</pre>

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

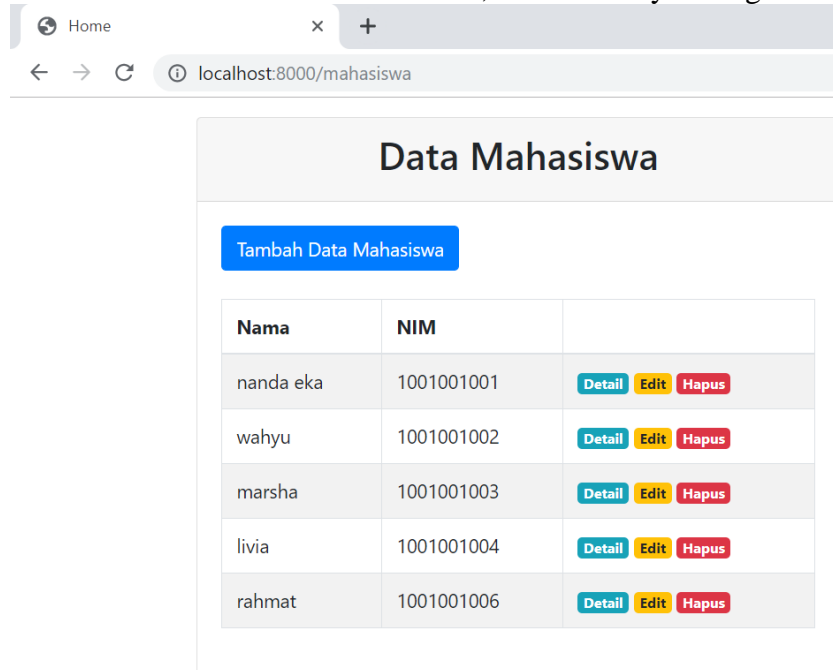
c. Memasukkan Data (Create) ke Database

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

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> <pre> resources > views > tambah.blade.php > ... 1 @extends('master') 2 <!-- isi title --> 3 @section('title', 'Tambah Data') 4 5 <!-- isi bagian judul halaman --> 6 @section('judul_halaman', 'Tambah Data Mahasiswa') 7 8 <!-- isi bagian konten --> 9 @section('konten') 10 Kembali 11
 12
 13 <form action="/mahasiswa/simpan" method="post"> 14 {{ csrf_field() }} 15 <div class="form-group"> 16 <label for="namamhs">Nama</label> 17 <input type="text" class="form-control" require="require" name="namamhs">
 18 </div> 19 <div class="form-group"> 20 <label for="nimmhs">NIM</label> 21 <input type="number" class="form-control" require="require" name="nimmhs">
 22 </div> 23 <div class="form-group"> 24 <label for="emailmhs">E-mail</label> 25 <input type="email" class="form-control" require="require" name="emailmhs">
 26 </div> 27 <div class="form-group"> 28 <label for="jurusanmhs">Jurusan</label> 29 <input type="text" class="form-control" require="require" name="jurusanmhs">
 30 </div> 31 <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button> 32 </form> 33 @endsection </pre>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 18 19 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 20 </pre>
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre> 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->jurusan 28]); 29 30 return redirect('/mahasiswa'); 31 } </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> app > Mahasiswa.php > Mahasiswa 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> 

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>20 21 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 22</pre>
2	<p>Buat method detail pada MahasiswaController.php di folder <code>app/Http/Controllers</code> yang akan menampilkan data mahasiswa pada view <code>detail.blade.php</code>.</p> <pre>32 33 public function detail(\$id) 34 { 35 \$mahasiswa = Mahasiswa::find(\$id); 36 return view('detail', ['mahasiswa' => \$mahasiswa]); 37 }</pre>

3

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

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

4

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

The image shows two screenshots of a web application running on localhost:8000.

The first screenshot shows the 'Data Mahasiswa' page. It has a 'Tambah Data Mahasiswa' button and a table with the following data:

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
livia	1001001004	Detail Edit Hapus
rahmat	1001001006	Detail Edit Hapus

The second screenshot shows the 'Detail Data Mahasiswa' page for the student 'rahmat'. It features a 'Kembali' button and the following details:

rahmat
NIM : 1001001006
E-mail : rahmat@gmail.com
Jurusan : teknik mesin

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> 22 23 Route::get('/mahasiswa/edit{id}', 'MahasiswaController@edit'); 24 </pre>
2	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre> 39 public function edit(\$id) 40 { 41 \$mahasiswa = Mahasiswa::find(\$id); 42 return view('edit', ['mahasiswa' => \$mahasiswa]); 43 } </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> resources > views > edit.blade.php > br 1 @extends('master') 2 <!-- isi title --> 3 @section('title', 'Edit Data') 4 <!-- isi bagian judul halaman --> 5 @section('judul_halaman', 'Edit Data Mahasiswa') 6 <!-- isi bagian konten --> 7 @section('konten') 8 Kembali 9
 10
 11 <form action="/mahasiswa/update/{{ \$mahasiswa->id }}" method="post"> 12 {{ csrf_field() }} 13 <input type="hidden" name="id" value="{{ \$mahasiswa->id }}">
 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 }}">
 17 </div> 18 <div class="form-group"> 19 <label for="nimhs">NIM</label> 20 <input type="number" class="form-control" required="required" name="nimhs" value="{{ \$mahasiswa->nim }}">
 21 </div> 22 <div class="form-group"> 23 <label for="emailhs">E-mail</label> 24 <input type="email" class="form-control" required="required" name="emailhs" value="{{ \$mahasiswa->email }}">
 25 </div> 26 <div class="form-group"> 27 <label for="jurusanhs">Jurusan</label> 28 <input type="text" class="form-control" required="required" name="jurusanhs" value="{{ \$mahasiswa->jurusan }}">
 29 </div> 30 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button> 31 </form> 32 @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> 24 25 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update'); 26 </pre>

5

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

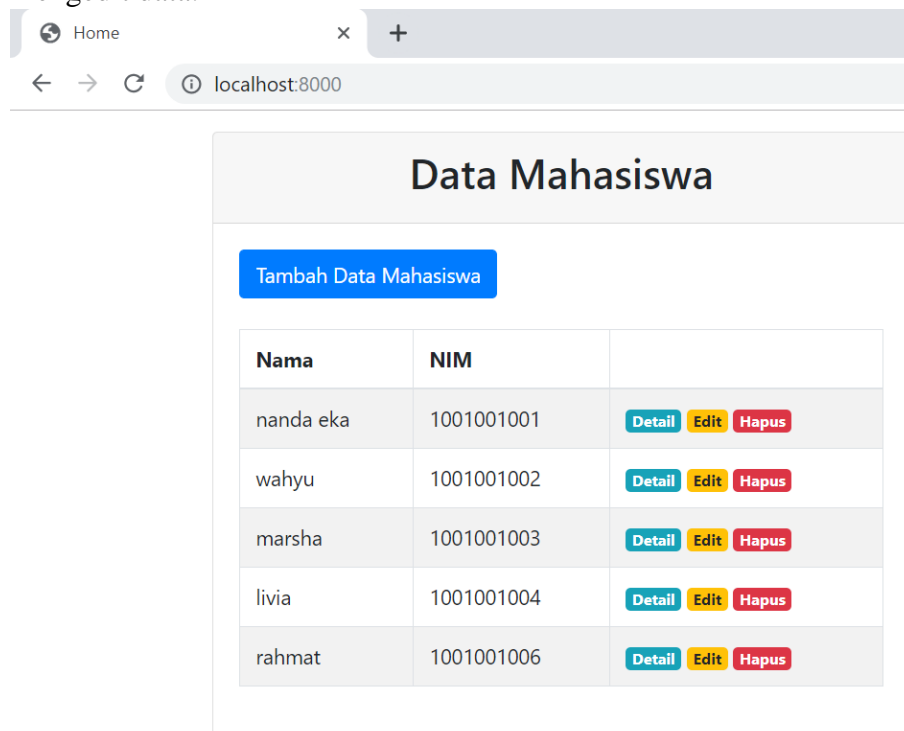
```

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 }

```

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 set to localhost:8000. The page title is 'Data Mahasiswa'. Below the title is a blue button labeled 'Tambah Data Mahasiswa'. Below the button is a table with the following data:

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
livia	1001001004	Detail Edit Hapus
rahmat	1001001006	Detail Edit Hapus

Edit Data

localhost:8000/mahasiswa/edit/4

Edit Data Mahasiswa

Kembali

Nama

livia yurike

NIM

1001001004

E-mail

livia@gmail.com

Jurusan

Teknik Informatika

Simpan Data

Home

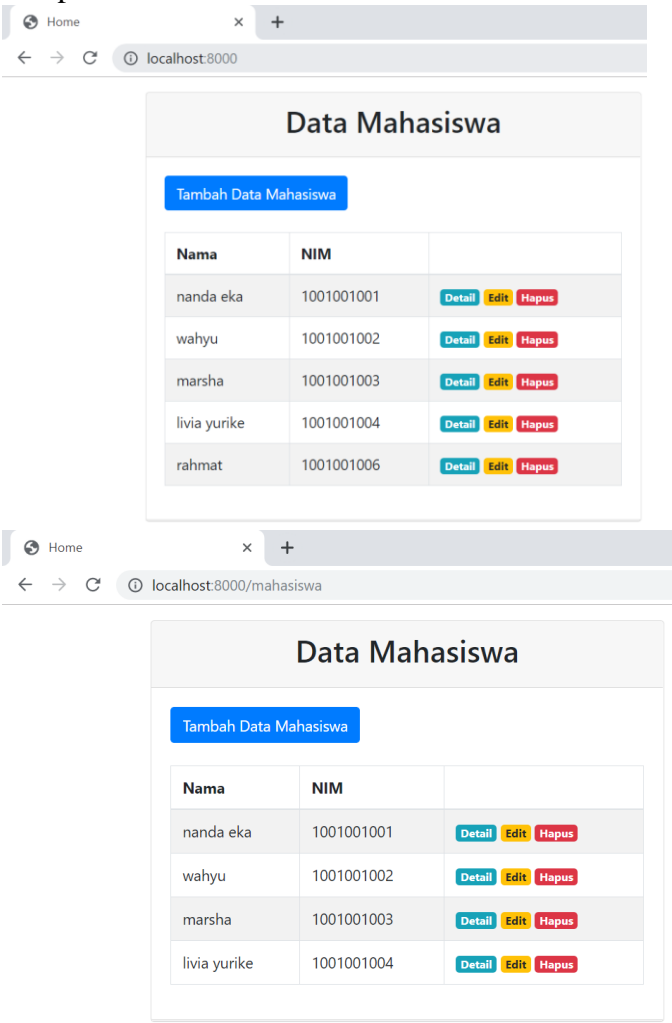
localhost:8000/mahasiswa

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	<div>DetailEditHapus</div>
wahyu	1001001002	<div>DetailEditHapus</div>
marsha	1001001003	<div>DetailEditHapus</div>
livia yurike	1001001004	<div>DetailEditHapus</div>
rahmat	1001001006	<div>DetailEditHapus</div>

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> 26 27 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 28 </pre>
2	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p> <pre> 56 public function hapus(\$id) 57 { 58 \$mahasiswa = Mahasiswa::find(\$id); 59 \$mahasiswa->delete(); 60 return redirect('/mahasiswa'); 61 } </pre>
3	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.</p>  <p>The first screenshot shows the 'Data Mahasiswa' page with a table containing 6 rows of student data. The 'rahmat' row is highlighted, and the 'Hapus' button is clicked. The second screenshot shows the same page but with only 4 rows of data, indicating that 'rahmat' has been successfully deleted.</p>