

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT

Jobsheet-10
Membuat CRUD menggunakan Laravel



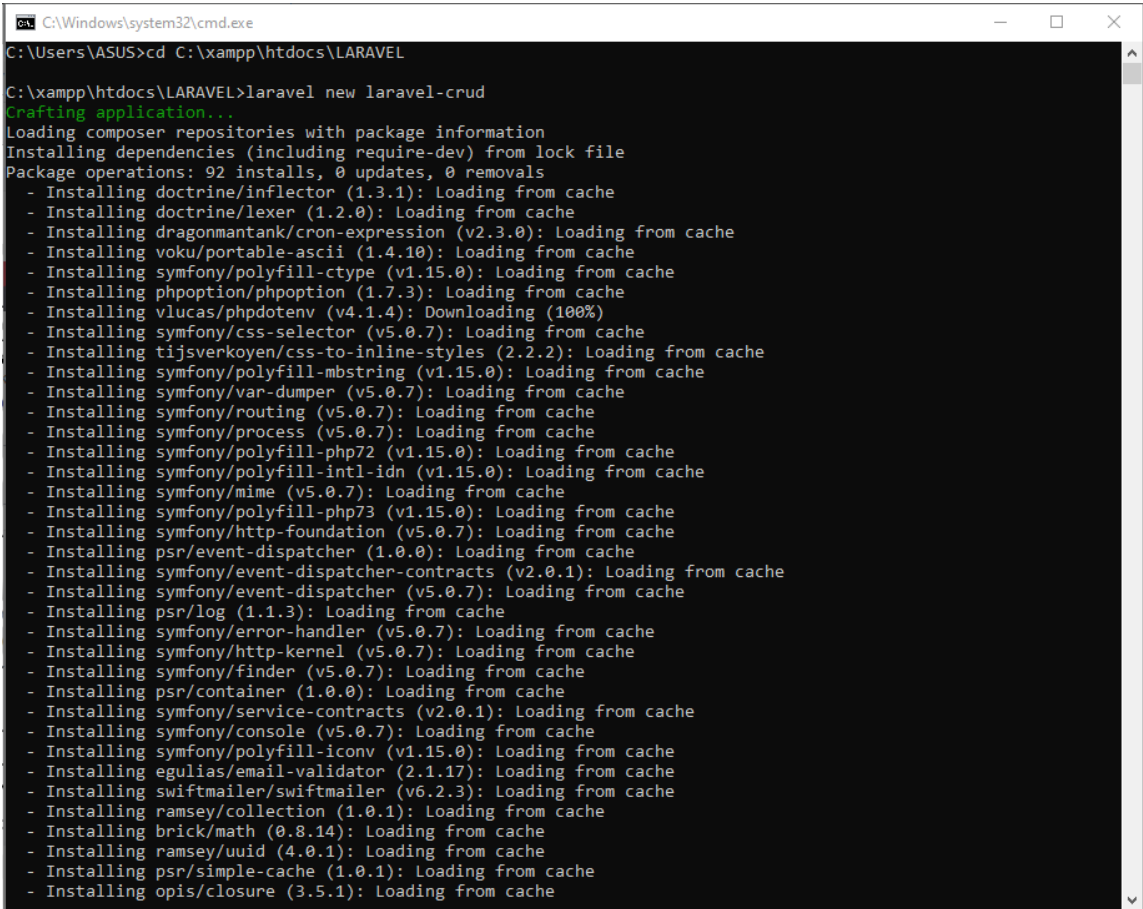
NAMA : DINA RISKY ALIN SAPUTRI
NIM : 1841720016
KELAS : 2B

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020

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

Query Builder adalah salah satu fitur untuk menjalankan *query database*. Laravel menyediakan fungsifungsi 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 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>

```
.env
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:3wRH6VTdNUMiliqZfU6aZg8zoEAd+fsc8i5fDBhDka4=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=latihan_laravel
DB_USERNAME=root
DB_PASSWORD=
```

Keterangan:

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

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

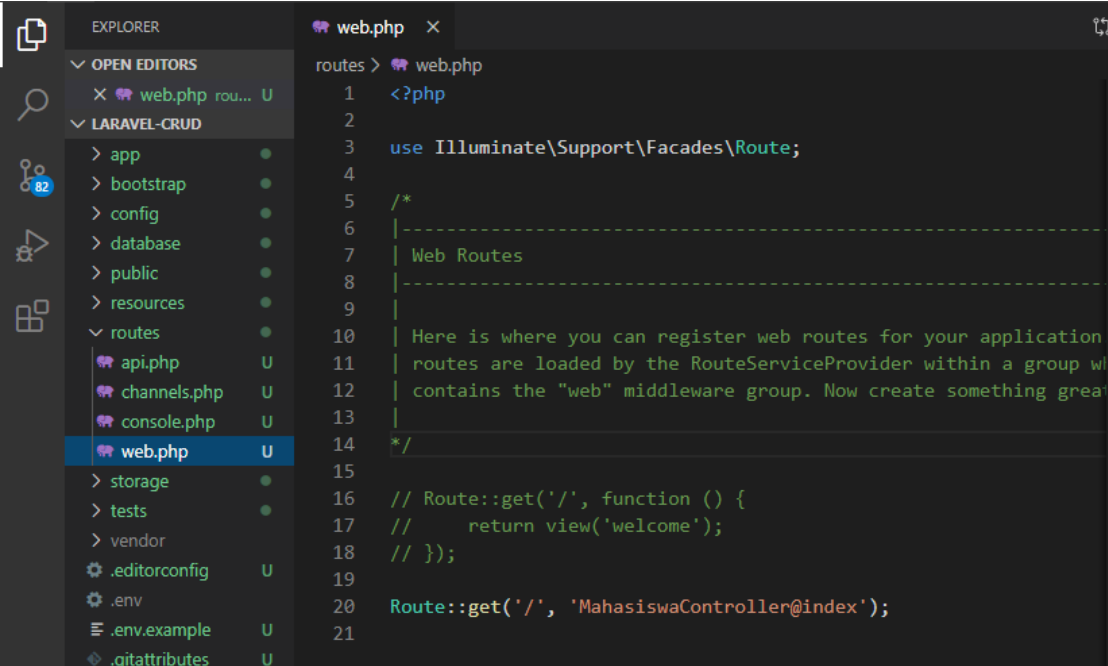
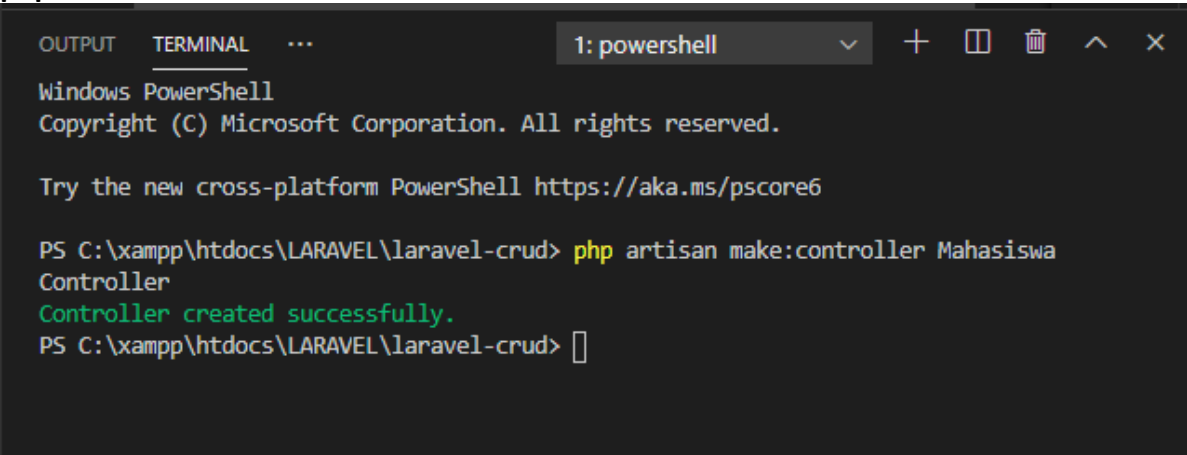
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	nim	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	email	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	jurusan	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

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

	id	nama	nim	email	jurusan
1	nanda	1001001001	nandan@gmail.com	Teknik Informatika	
2	wahyu	1001001002	wahyu@gmail.com	Teknik Elektro	

b. Menampilkan Data dari Database

Langkah	Keterangan
1	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>Keterangan: - Ketika route ('/') diakses, akan dijalankan method index pada controller bernama MahasiswaController.</p>
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> 
3	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p>

```

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

```

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**.

```

1  <!doctype html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
7      rel="stylesheet">
8      <title>@yield('title')</title>
9  </head>
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 class="card">
24
25
26
27
28
29 </div>
30 </div>
31 </div>
32 </body>
33

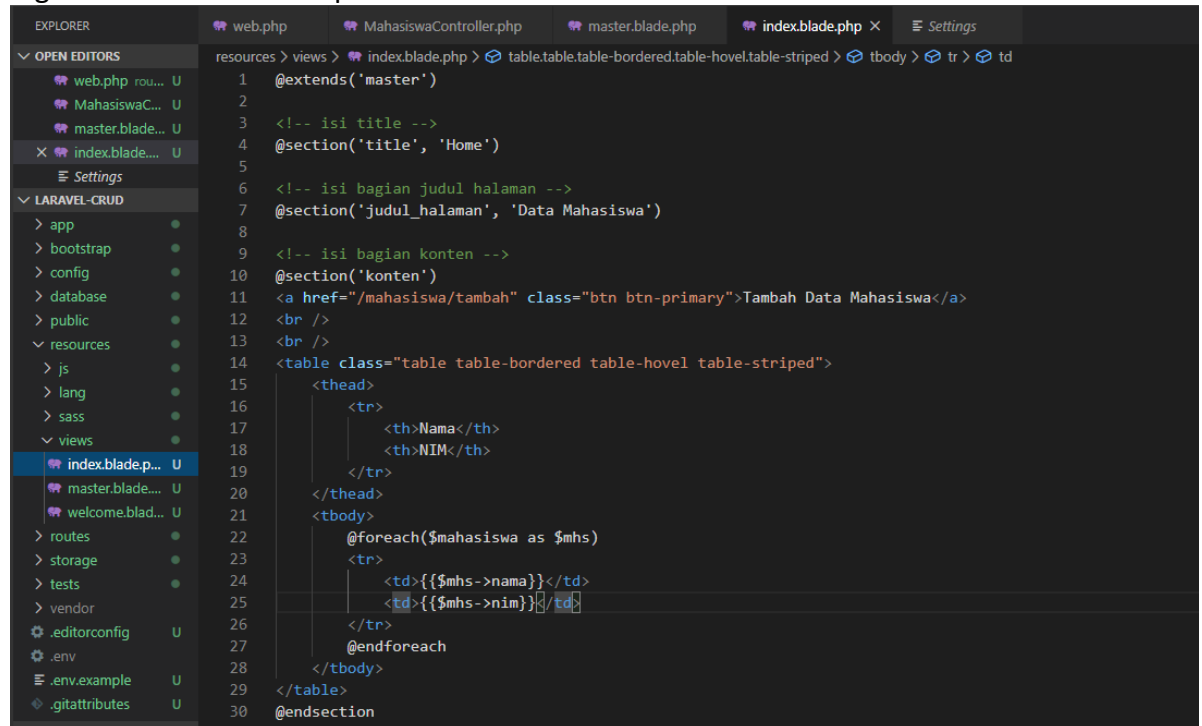
```

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.



```

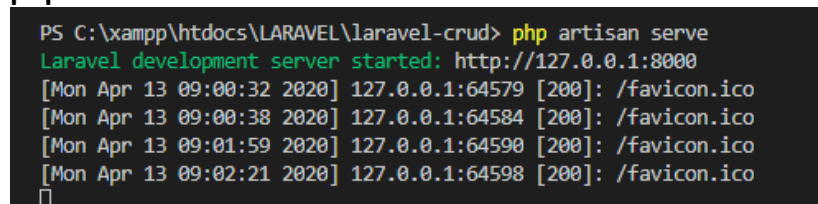
1 @extends('master')
2
3 <!-- isi title -->
4 @section('title', 'Home')
5
6 <!-- isi bagian judul halaman -->
7 @section('judul_halaman', 'Data Mahasiswa')
8
9 <!-- isi bagian konten -->
10 @section('konten')
11 <a href="/mahasiswa/tambah" class="btn btn-primary">Tambah Data Mahasiswa</a>
12 <br />
13 <br />
14 <table class="table table-bordered table-hovel 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

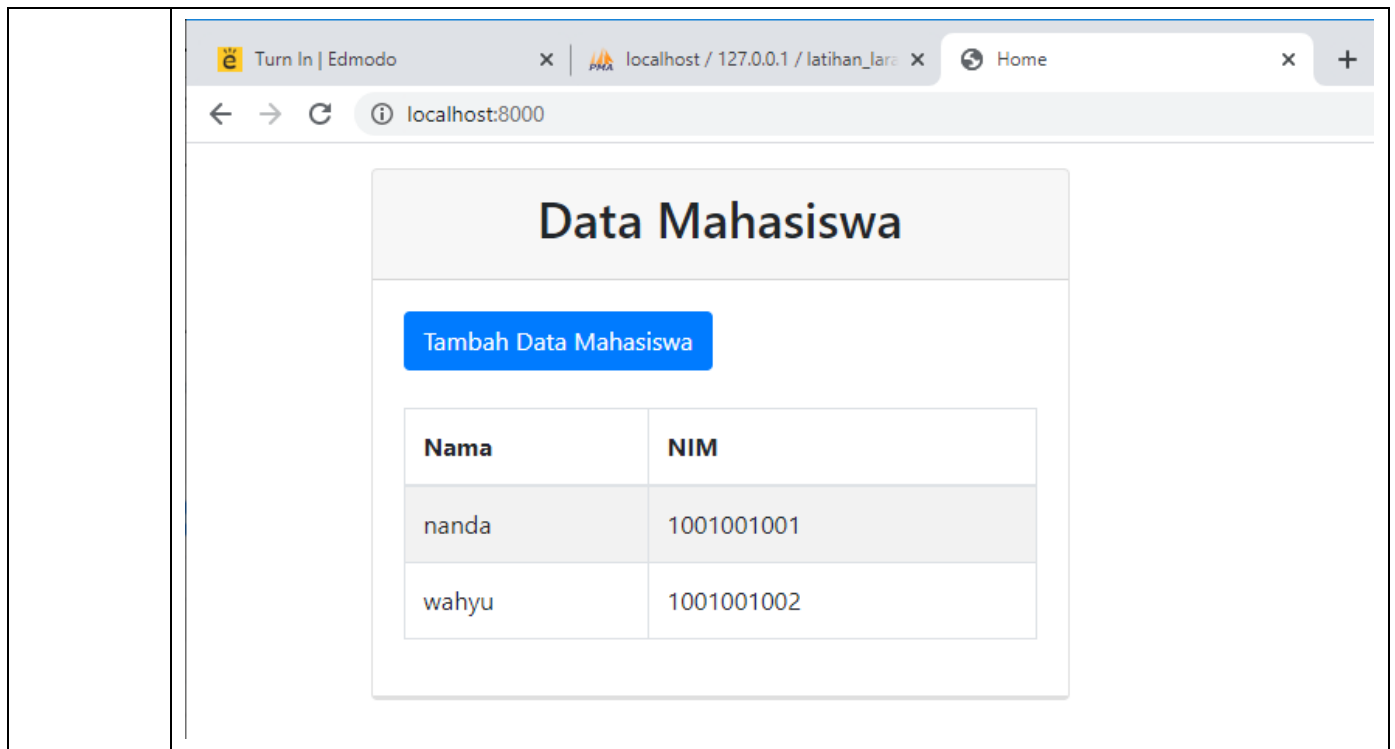
Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud **php artisan serve**



```

PS C:\xampp\htdocs\LARAVEL\laravel-crud> php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Mon Apr 13 09:00:32 2020] 127.0.0.1:64579 [200]: /favicon.ico
[Mon Apr 13 09:00:38 2020] 127.0.0.1:64584 [200]: /favicon.ico
[Mon Apr 13 09:01:59 2020] 127.0.0.1:64590 [200]: /favicon.ico
[Mon Apr 13 09:02:21 2020] 127.0.0.1:64598 [200]: /favicon.ico
  
```

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 routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController</p> <pre> 21 22 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 23 </pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang Akan menampilkan view tambah.</p> <pre> 19 public function tambah() 20 { 21 // memanggil view tambah() 22 return view('tambah'); 23 } 24 25 </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>

```

tambah.blade.php x MahasiswaController.php index.blade.php web.php
resources > views > tambah.blade.php > form
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 <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
11 <br />
12 <br />
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" required="required" name="namamhs"><br />
18     </div>
19     <div class="form-group">
20         <label for="nimhs">NIM</label>
21         <input type="number" class="form-control" required="required" name="nimhs"><br />
22     </div>
23     <div class="form-group">
24         <label for="emailmhs">E-mail</label>
25         <input type="email" class="form-control" required="required" name="emailmhs"><br />
26     </div>
27     <div class="form-group">
28         <label for="jurusanmhs">Jurusan</label>
29         <input type="text" class="form-control" required="required" name="jurusanmhs"><br />
30     </div>
31     <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button>
32 </form>
33 @endsection

```

Keterangan:

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

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

```

24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');
25 |

```

Keterangan:

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

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


```

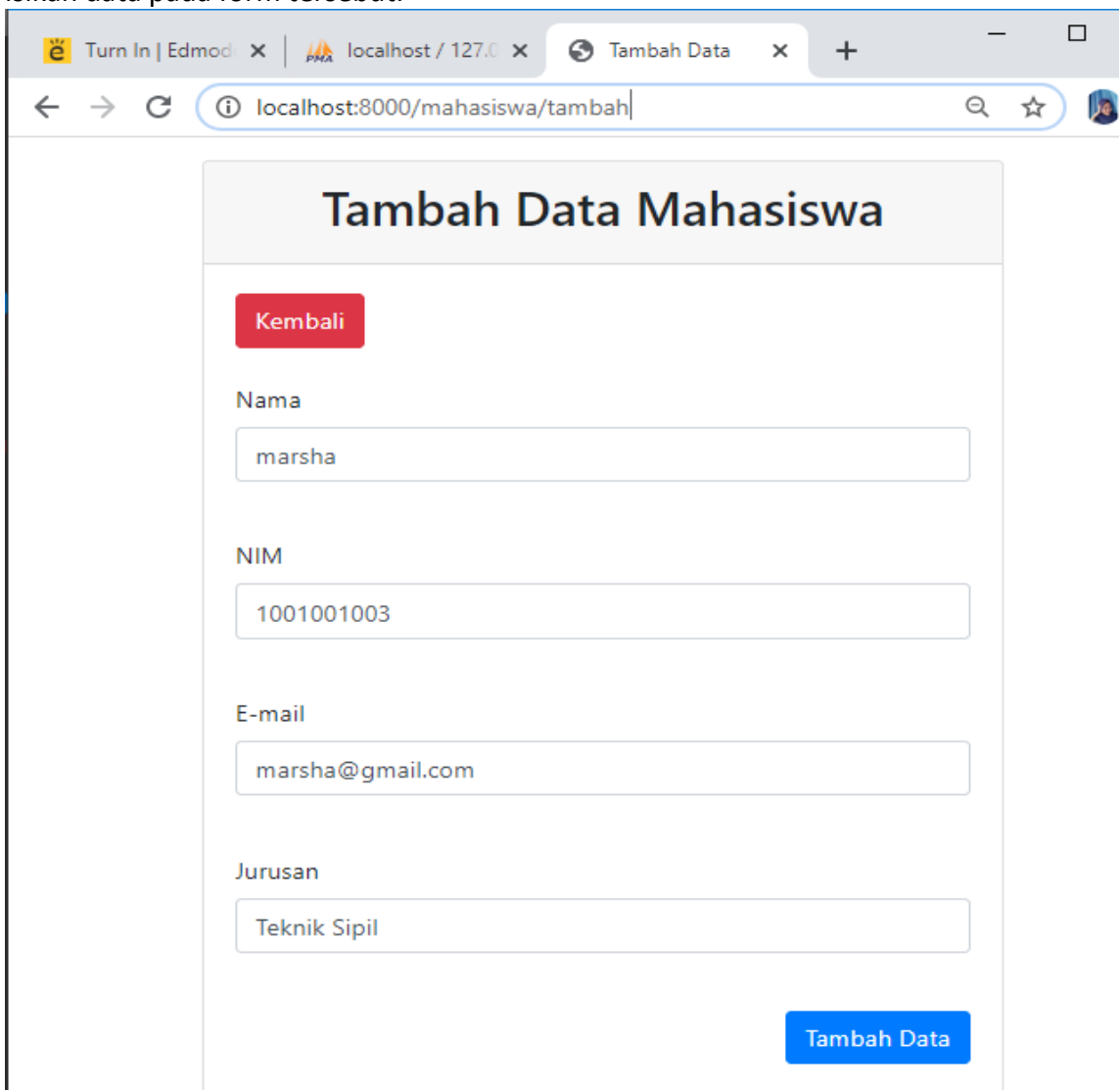
25     public function simpan(Request $request)
26     {
27         // insert data ke table mahasiswa
28         DB::table('mahasiswa')->insert([
29             'nama' => $request->namamhs,
30             'nim' => $request->nimmhs,
31             'email' => $request->emailmhs,
32             'jurusan' => $request->jurusanmhs
33         ]);
34         return redirect('/');
35     }
36 }
37

```

Keterangan:

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

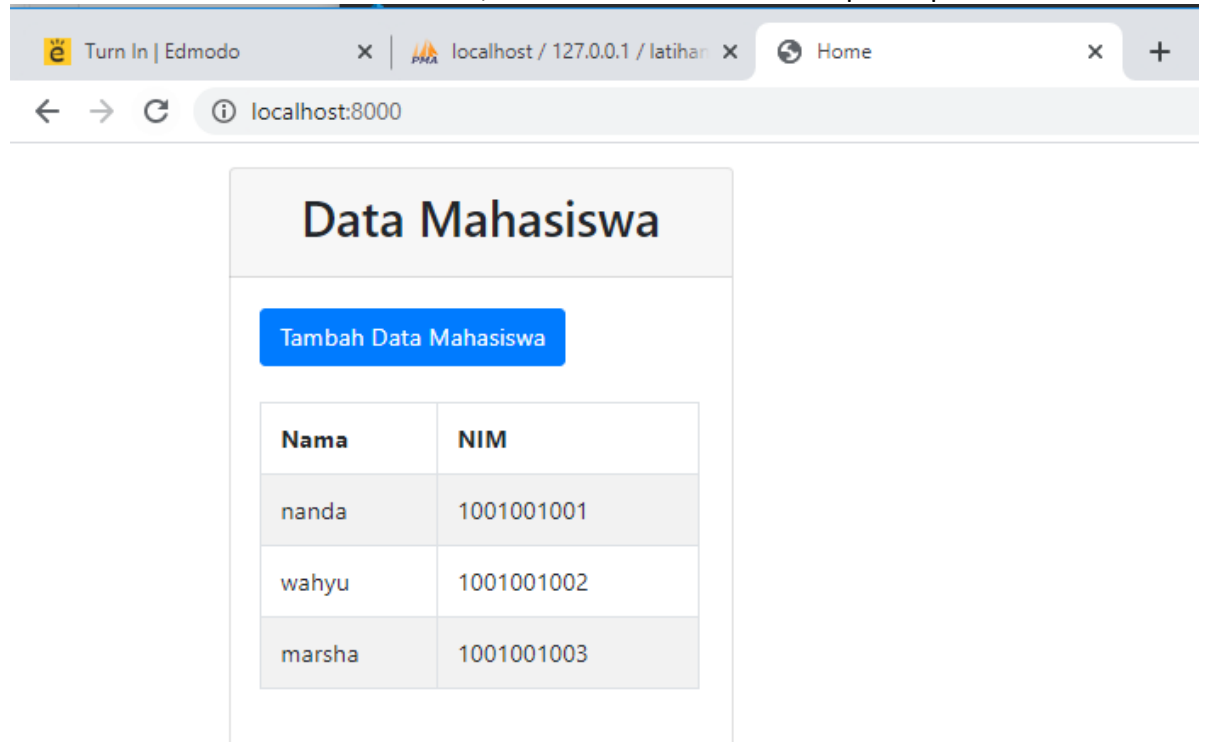
- 6 Kembali coba jalankan localhost:8000 dan pilih tombol 'Tambah Data Mahasiswa', maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.



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

- Kembali** (button)
- Nama**: marsha
- NIM**: 1001001003
- E-mail**: marsha@gmail.com
- Jurusan**: Teknik Sipil
- Tambah Data** (button)

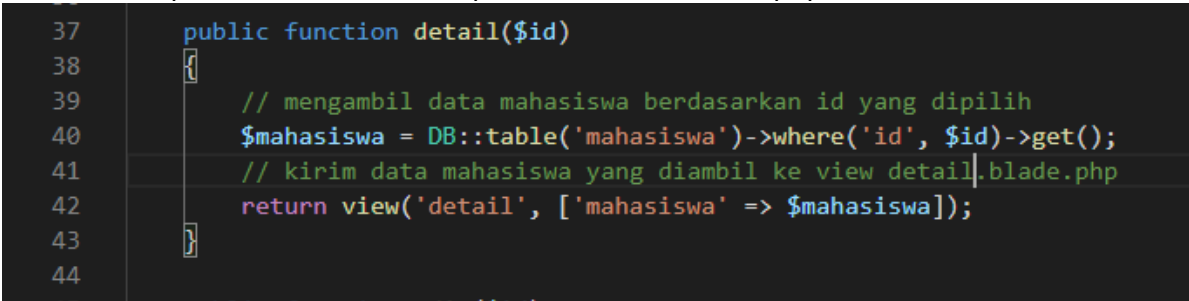
Setelah kita klik tombol tambah data, maka data baru akan ditampilkan pada halaman index.



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

d. Melihat Detail Data dari Database

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

	 <pre> 9 <!-- isi bagian konten --> 10 @section('konten') 11 Tambah Data Mahasiswa 12
 13
 14 <table class="table table-bordered table-hovel 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> 25 26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 27 </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 detail.blade.php 42 return view('detail', ['mahasiswa' => \$mahasiswa]); 43 } 44 </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>

EXPLORER

index.blade.php

detail.blade.php

web.php

OPEN EDITORS

index.blade.php

detail.blade.php

web.php

LARAVEL-CRUD

resources > views > detail.blade.php > ...

index.blade.php

detail.blade.php

master.blade.php

tambah.blade.php

welcome.blade.php

Providers

User.php

bootstrap

config

database

public

resources

js

lang

sass

views

detail.blade.php

index.blade.php

master.blade.php

tambah.blade.php

welcome.blade.php

routes

api.php

channels.php

console.php

web.php

```

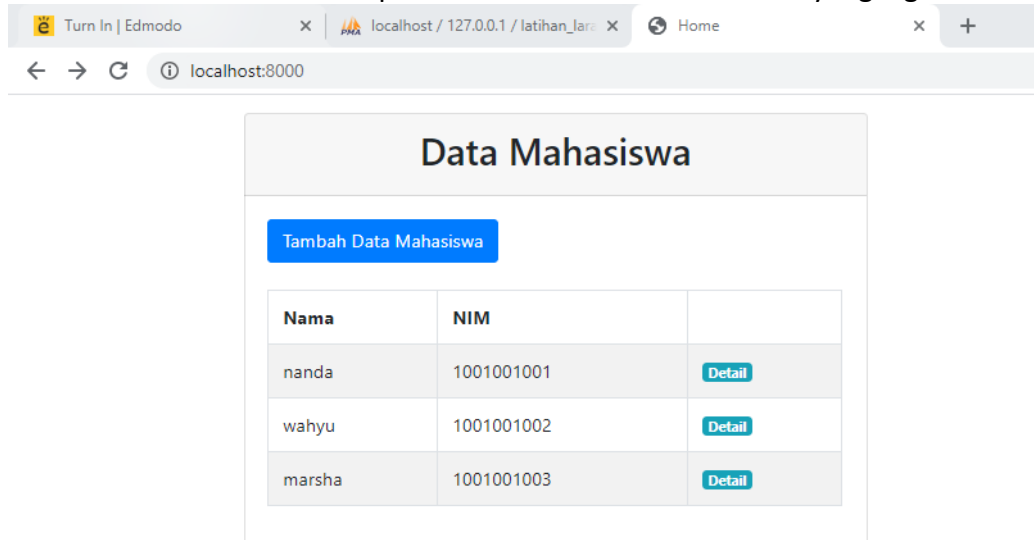
1  @extends('master')
2
3  <!-- isi title -->
4  @section('title', 'Detail Mahasiswa')
5
6  <!-- isi bagian judul halaman -->
7  @section('judul_halaman', 'Detail Data Mahasiswa')
8
9  <!-- isi bagian konten -->
10 @section('konten')
11 <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
12
13 <br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

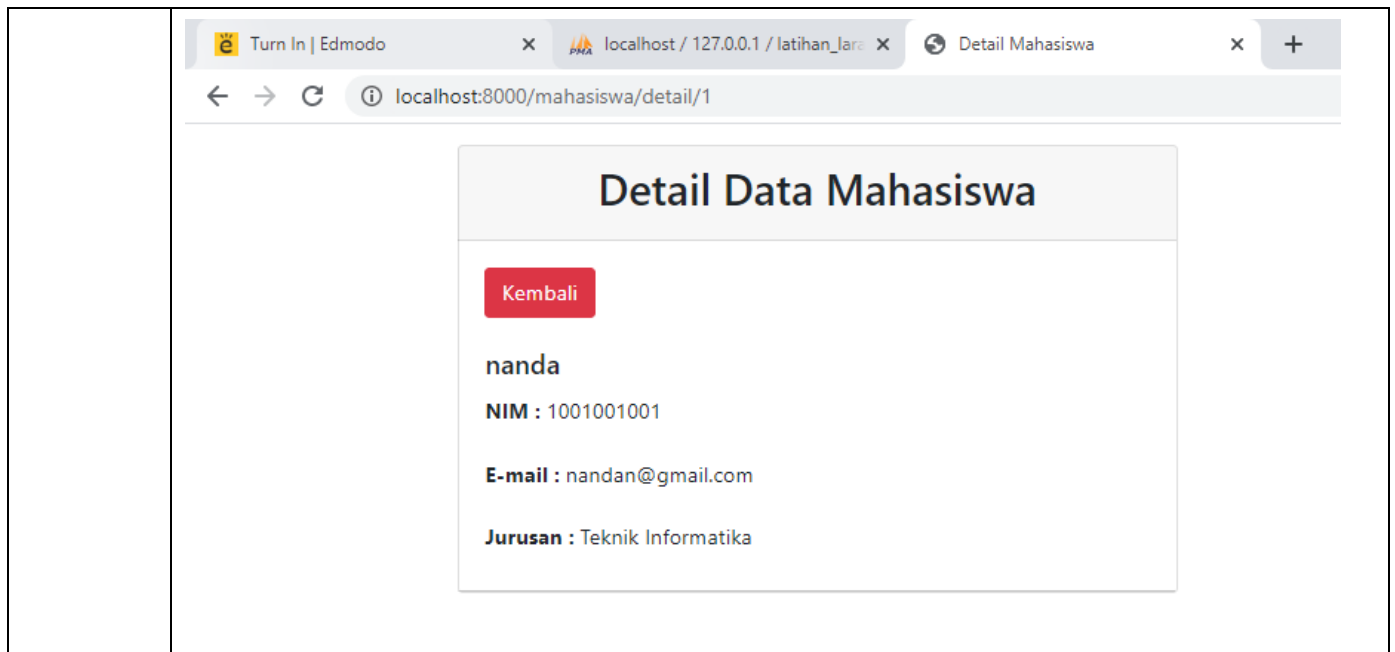
```

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.

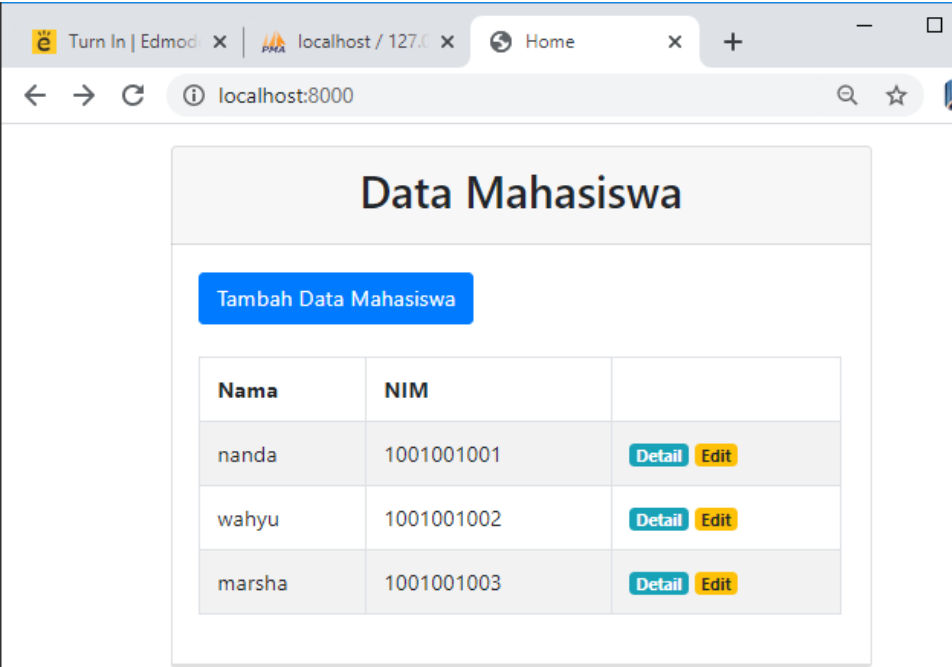




e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk melihat detail data (Line 29) pada file index.blade.php di folder resources/views</p> <pre> index.blade.php x MahasiswaController.php detail.blade.php web.php resources > views > index.blade.php > table.table-bordered.table-hovel.table-striped > thead 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-hovel 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> 27 28 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 29 </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 } 52 53 </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Line 49 : query builder untuk mengambil data mahasiswa berdasarkan id yang dipilih
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 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 <form action="/mahasiswa/update" method="POST"> 15 @csrf 16 @foreach(\$mahasiswa as \$mhs) 17 <input type="hidden" name="id" value="{{ \$mhs->id }}"> 18 <div class="form-group"> 19 <label>Nama</label> 20 <input type="text" class="form-control" value="{{ \$mhs->nama }}" required="required" name="namamhs">
 21 </div> 22 <div class="form-group"> 23 <label>NIM</label> 24 <input type="number" class="form-control" value="{{ \$mhs->nim }}" required="required" name="nimamhs">
 25 </div> 26 <div class="form-group"> 27 <label>Email</label> 28 <input type="email" class="form-control" value="{{ \$mhs->email }}" required="required" name="emailmhs">
 29 </div> 30 <div class="form-group"> 31 <label>Jurusan</label> 32 <input type="text" class="form-control" value="{{ \$mhs->jurusan }}" required="required" name="jurusanmhs">
 33 </div> 34 <input type="submit" class="btn btn-primary float-right" value="Update Data"> 35 @endforeach 36 </form> 37 @endsection </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Line 14-36 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan - Line 15 terdapat action="/mahasiswa/update" yang menunjukkan routes /mahasiswa/update dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController

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'); 31 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php
6	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p> <pre> public function update(Request \$request) { // update data mahasiswa DB::table('mahasiswa')->where('id', \$request->id)->update(['nama' => \$request->namamhs, 'nim' => \$request->nimmhs, 'email' => \$request->emailmhs, 'jurusan' => \$request->jurusanmhs]); return redirect('/'); } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Variabel \$request untuk menerima data yang akan ditambahkan ke database - Line 58-63 merupakan query builder untuk update data ke tabel mahasiswa
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>

Edit Data Mahasiswa

[Kembali](#)

Nama
nanda eka

NIM
1001001001

Email
nandan@gmail.com

Jurusan
Teknik Informatika

[Update Data](#)

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

Data Mahasiswa

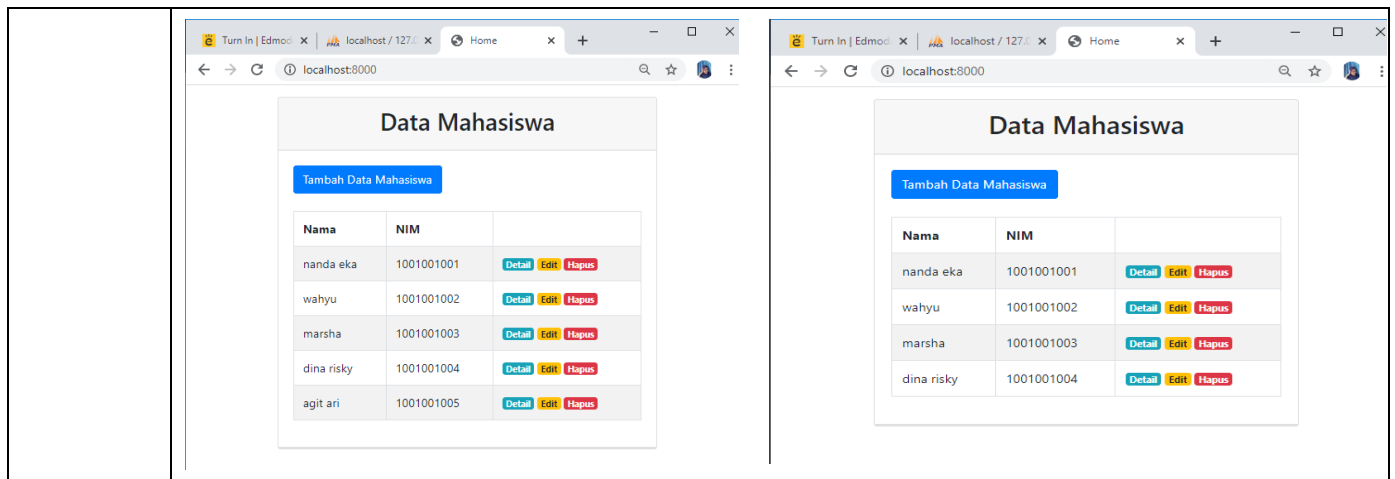
[Tambah Data Mahasiswa](#)

Nama	NIM	
nanda eka	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
marsha	1001001003	Detail Edit

f. Menghapus Data (Delete) dari Database

Langkah	Keterangan
---------	------------

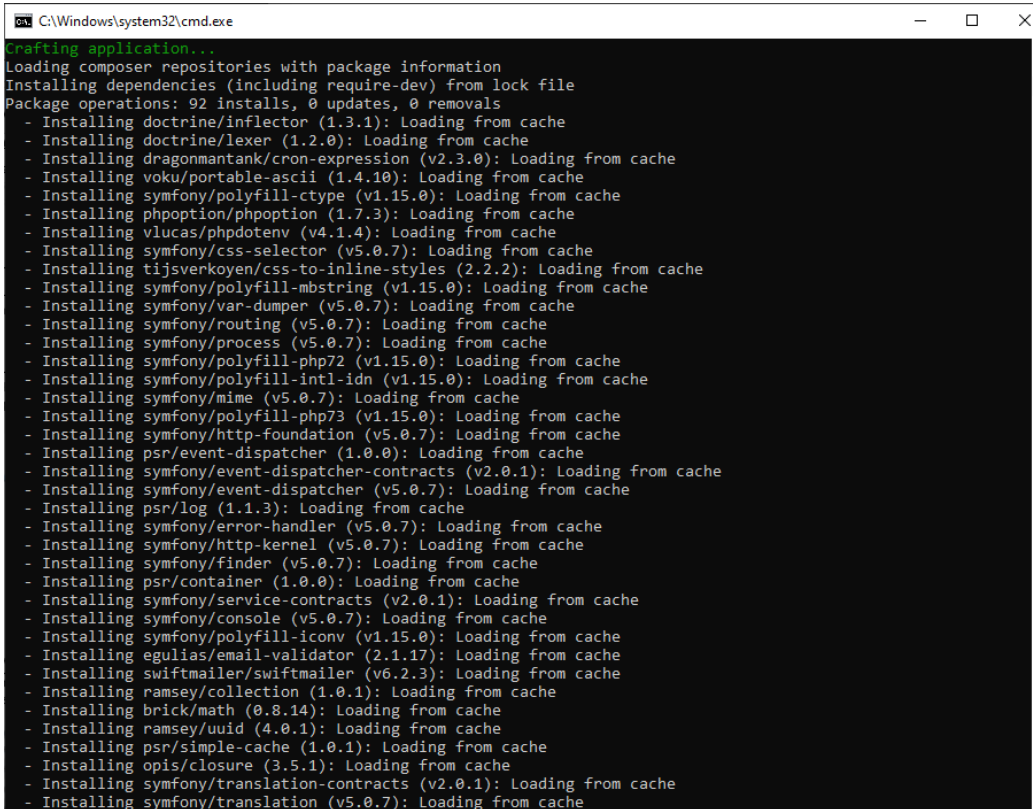
1	<p>Buatlah tombol untuk melihat detail data (Line 30) pada file index.blade.php di folder resources/views</p>  <pre> 9 <!-- isi bagian konten --> 10 @section('konten') 11 Tambah Data Mahasiswa 12
 13
 14 <table class="table table-bordered table-hovel table-striped"> 15 <thead> 16 <tr> 17 <th>Nama</th> 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/edit yang akan menjalankan fungsi edit pada MahasiswaController</p>  <pre> 31 32 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 33 </pre>
3	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p>  <pre> public function hapus(\$id) { // menghapus data mahasiswa berdasarkan id yang dipilih DB::table('mahasiswa')->where('id', \$id)->delete(); return redirect('/'); } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Line 67 : query builder untuk menghapus data mahasiswa berdasarkan id yang dipilih
4	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir.</p>

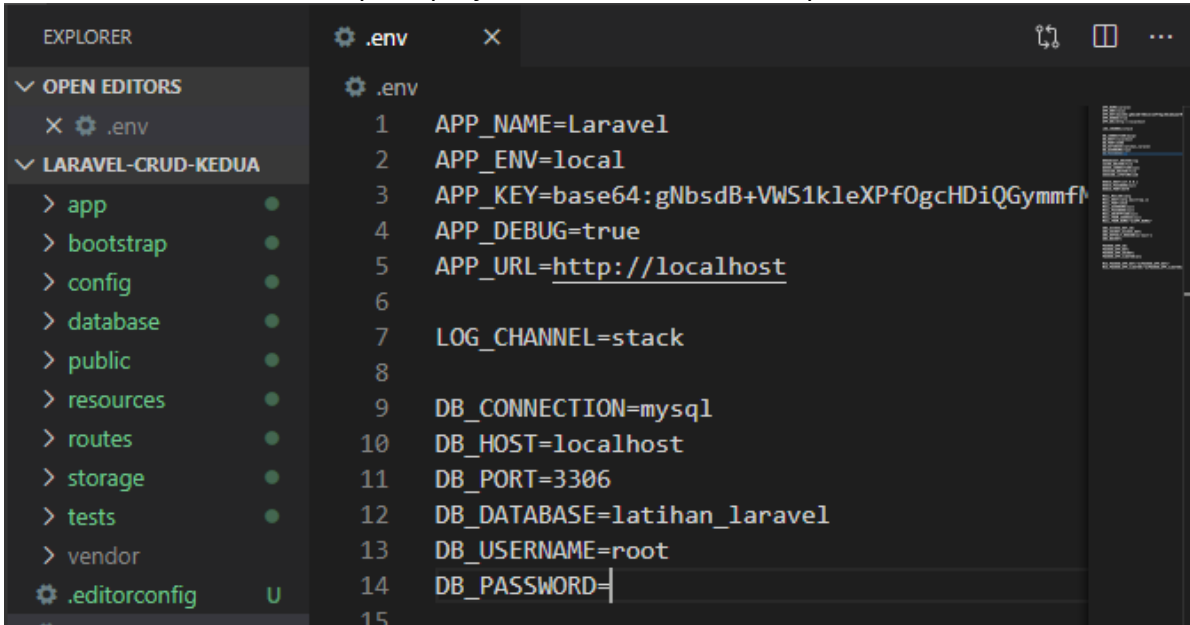


Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

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

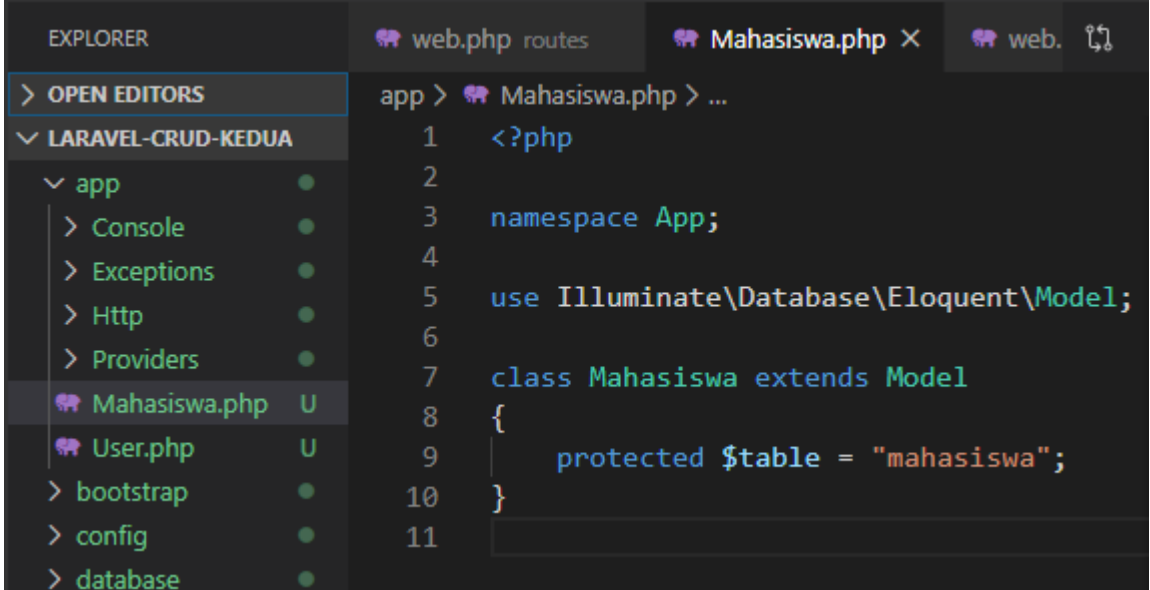
Langkah	Keterangan
1	<p>Buatlah project Laravel baru dengan nama laravel-crud-kedua. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud-kedua</pre> 

2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.</p>  <p>Keterangan:</p> <ul style="list-style-type: none"> - Nama database yang akan digunakan adalah latihan_laravel dengan username root.
---	---

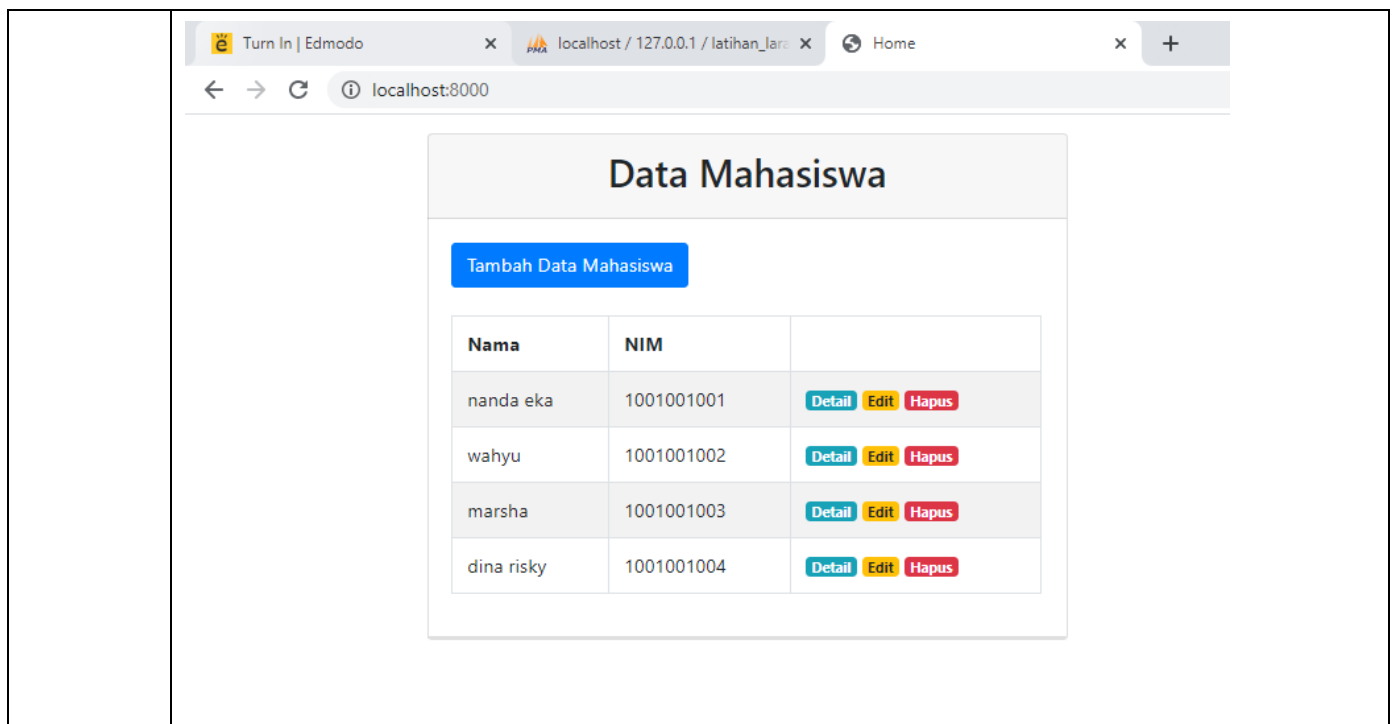
3	<p>Pada project ini kita gunakan database dan tabel yang sebelumnya digunakan pada Praktikum Bagian 1. Tambahkan kolom created_at dan updated_at yang bertipe TIMESTAMP dan default nilainya NULL</p> <table> <tr> <th>#</th><th>Nama</th><th>Jenis</th><th>Penyortiran</th><th>Atribut</th><th>Tak Ternilai</th><th>Bawaan</th><th>Komentar</th><th>Ekstra</th><th>Tindakan</th></tr> <tr> <td><input type="checkbox"/> 1</td><td>id</td><td>int(11)</td><td></td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td>AUTO_INCREMENT</td><td> Ubah Hapus Lainnya</td></tr> <tr> <td><input type="checkbox"/> 2</td><td>nama</td><td>varchar(100)</td><td>latin1_swedish_ci</td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td></td><td> Ubah Hapus Lainnya</td></tr> <tr> <td><input type="checkbox"/> 3</td><td>nim</td><td>varchar(10)</td><td>latin1_swedish_ci</td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td></td><td> Ubah Hapus Lainnya</td></tr> <tr> <td><input type="checkbox"/> 4</td><td>email</td><td>varchar(50)</td><td>latin1_swedish_ci</td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td></td><td> Ubah Hapus Lainnya</td></tr> <tr> <td><input type="checkbox"/> 5</td><td>jurusan</td><td>varchar(20)</td><td>latin1_swedish_ci</td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td></td><td> Ubah Hapus Lainnya</td></tr> <tr> <td><input type="checkbox"/> 6</td><td>created_at</td><td>timestamp</td><td></td><td></td><td>Ya</td><td>NULL</td><td></td><td></td><td> Ubah Hapus Lainnya</td></tr> <tr> <td><input type="checkbox"/> 7</td><td>updated_at</td><td>timestamp</td><td></td><td></td><td>Ya</td><td>NULL</td><td></td><td></td><td> Ubah Hapus Lainnya</td></tr> </table>	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan	<input type="checkbox"/> 1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya	<input type="checkbox"/> 2	nama	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya	<input type="checkbox"/> 3	nim	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya	<input type="checkbox"/> 4	email	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya	<input type="checkbox"/> 5	jurusan	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya	<input type="checkbox"/> 6	created_at	timestamp			Ya	NULL			Ubah Hapus Lainnya	<input type="checkbox"/> 7	updated_at	timestamp			Ya	NULL			Ubah Hapus Lainnya
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan																																																																								
<input type="checkbox"/> 1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya																																																																								
<input type="checkbox"/> 2	nama	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya																																																																								
<input type="checkbox"/> 3	nim	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya																																																																								
<input type="checkbox"/> 4	email	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya																																																																								
<input type="checkbox"/> 5	jurusan	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya																																																																								
<input type="checkbox"/> 6	created_at	timestamp			Ya	NULL			Ubah Hapus Lainnya																																																																								
<input type="checkbox"/> 7	updated_at	timestamp			Ya	NULL			Ubah Hapus Lainnya																																																																								

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 data-bbox="288 1644 1417 1794"> 21 Route::get('/', 'MahasiswaController@index'); 22 Route::get('/mahasiswa', 'MahasiswaController@index'); 23 </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>PS C:\xampp\htdocs\LARAVEL\laravel-crud-kedua> php artisan make:model Mahasiswa Model created successfully. PS C:\xampp\htdocs\LARAVEL\laravel-crud-kedua> </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> <p>Keterangan:</p> <ul style="list-style-type: none"> - Model Mahasiswa akan menangani tabel mahasiswa
4	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan php artisan make:controller MahasiswaController</p> <pre>PS C:\xampp\htdocs\LARAVEL\laravel-crud-kedua> php artisan make:controller MahasiswaController Controller created successfully. PS C:\xampp\htdocs\LARAVEL\laravel-crud-kedua> </pre>
5	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p>

	<div data-bbox="288 107 1460 931">  <pre> app > Http > Controllers > MahasiswaController.php > ... 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 use App\Mahasiswa; 7 8 class MahasiswaController extends Controller 9 { 10 public function index() 11 { 12 \$mahasiswa = Mahasiswa::all(); 13 return view('index', ['mahasiswa' => \$mahasiswa]); 14 } 15 } 16 </pre> </div> <p>Keterangan:</p> <ul style="list-style-type: none"> - Tambahkan 'use App\Mahasiswa' (line 6) untuk menggunakan model Mahasiswa - Line 12 untuk mengambil semua data dari model/tabel Mahasiswa dan akan disimpan di variabel \$mahasiswa - Line 16 : data akan dikirim ke blade view bernama index
6	<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>
7	<p>Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud</p> <p>php artisan serve</p> <div data-bbox="288 1408 1267 1608">  <pre> PS C:\xampp\htdocs\LARAVEL\laravel-crud-kedua> php artisan serve Laravel development server started: http://127.0.0.1:8000 </pre> </div> <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> <pre> 23 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 24 </pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> 16 public function tambah() 17 { 18 return view('tambah'); 19 } 20 } </pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. Kita lakukan copy paste dari tambah.blade.php di Bagian 1</p>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 25 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php

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->jurusanmhs 28]); 29 return redirect('/'); 30 } 31 } 32 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Variabel \$request untuk menerima data yang akan ditambahkan ke database - Line 23-28 merupakan fitur eloquent menggunakan fungsi create() untuk insert data ke tabel mahasiswa
6	<p>Karena kita menggunakan fungsi create pada Controller, maka butuh ditambahkan code pada Line 10 yang disebut Mass Assignment pada model Mahasiswa.php. Mass Assignment digunakan untuk memfilter kolom mana yang boleh dan tidak boleh diinput.</p>  <pre> app > Mahasiswa.php > ... 1 <?php 2 3 namespace App; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class Mahasiswa extends Model 8 { 9 protected \$table = "mahasiswa"; 10 protected \$fillable = ['nama', 'nim', 'email', 'jurusan']; 11 } 12 </pre>
7	<p>Kembali coba jalankan localhost:8000 dan pilih tombol ‘Tambah Data Mahasiswa’, maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.</p>

Turn In | Edmodo x localhost / 127.0.0.1 / latihan_jar... x Tambah Data x +

localhost:8000/mahasiswa/tambah

Tambah Data Mahasiswa

[Kembali](#)

Nama
rahmat

NIM
1001001006

E-mail
rahmat@gmail.com

Jurusan
Teknik Mesin

[Tambah Data](#)

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

Turn In | Edmodo x localhost / 127.0.0.1 / latihan_jar... x Home x +

localhost:8000

Data Mahasiswa

[Tambah Data Mahasiswa](#)

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

d. Melihat Detail Data dari Database

Langkah	Keterangan
1	Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController <pre>Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');</pre>
2	Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.


```

32     public function detail($id)
33     {
34         $mahasiswa = Mahasiswa::find($id);
35         return view('detail', ['mahasiswa' => $mahasiswa]);
36     }
37 }
38

```

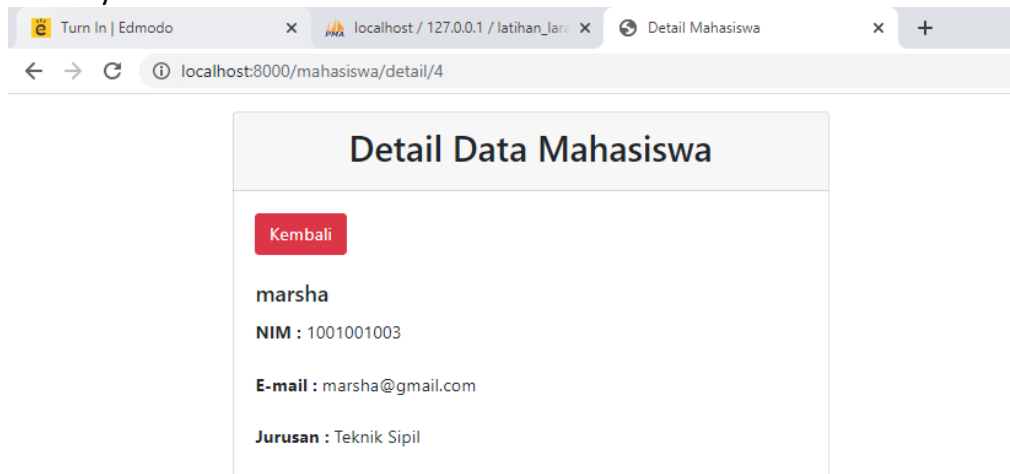
- 3 Kemudian buatlah view **detail.blade.php** yang menampilkan detail data mahasiswa pada folder **resources/views**. View detail juga mengaplikasikan master.blade.php.

```

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

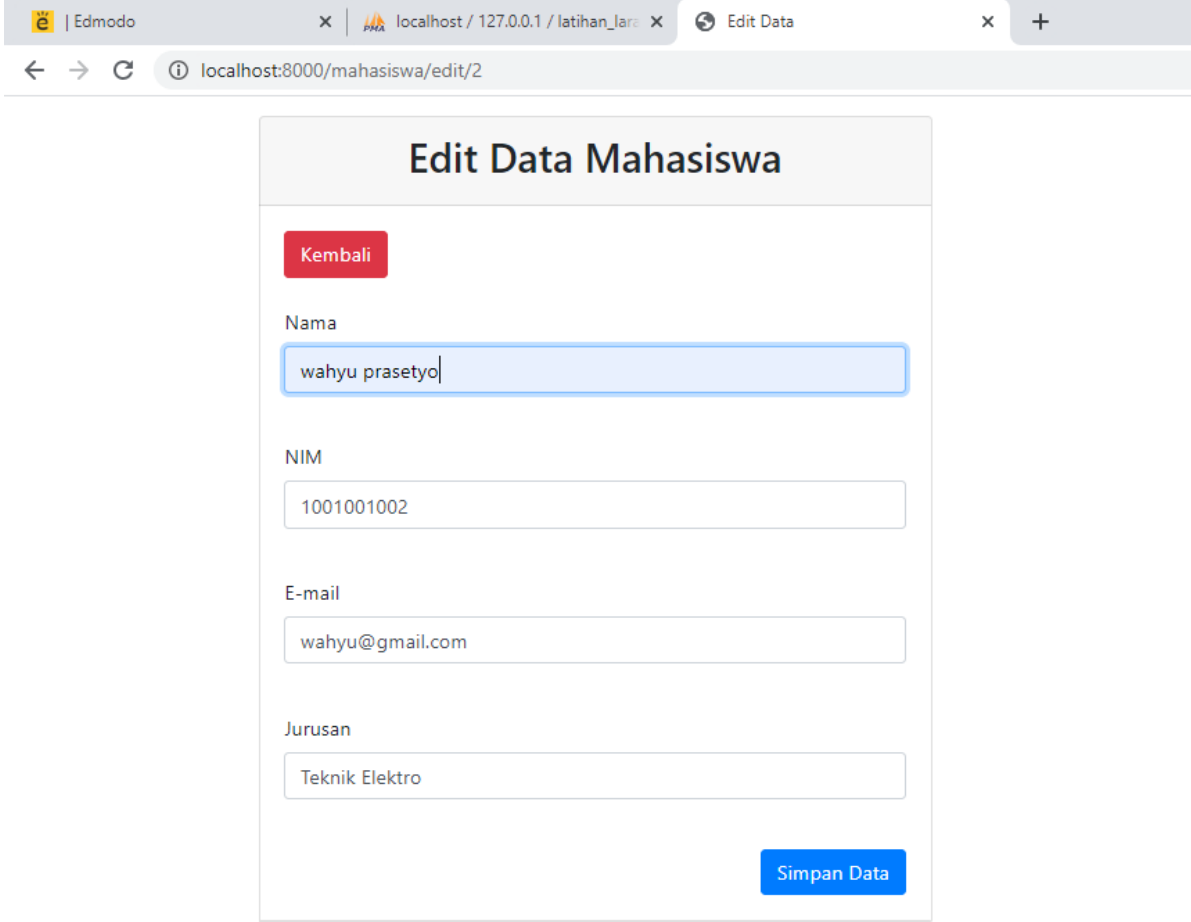
```

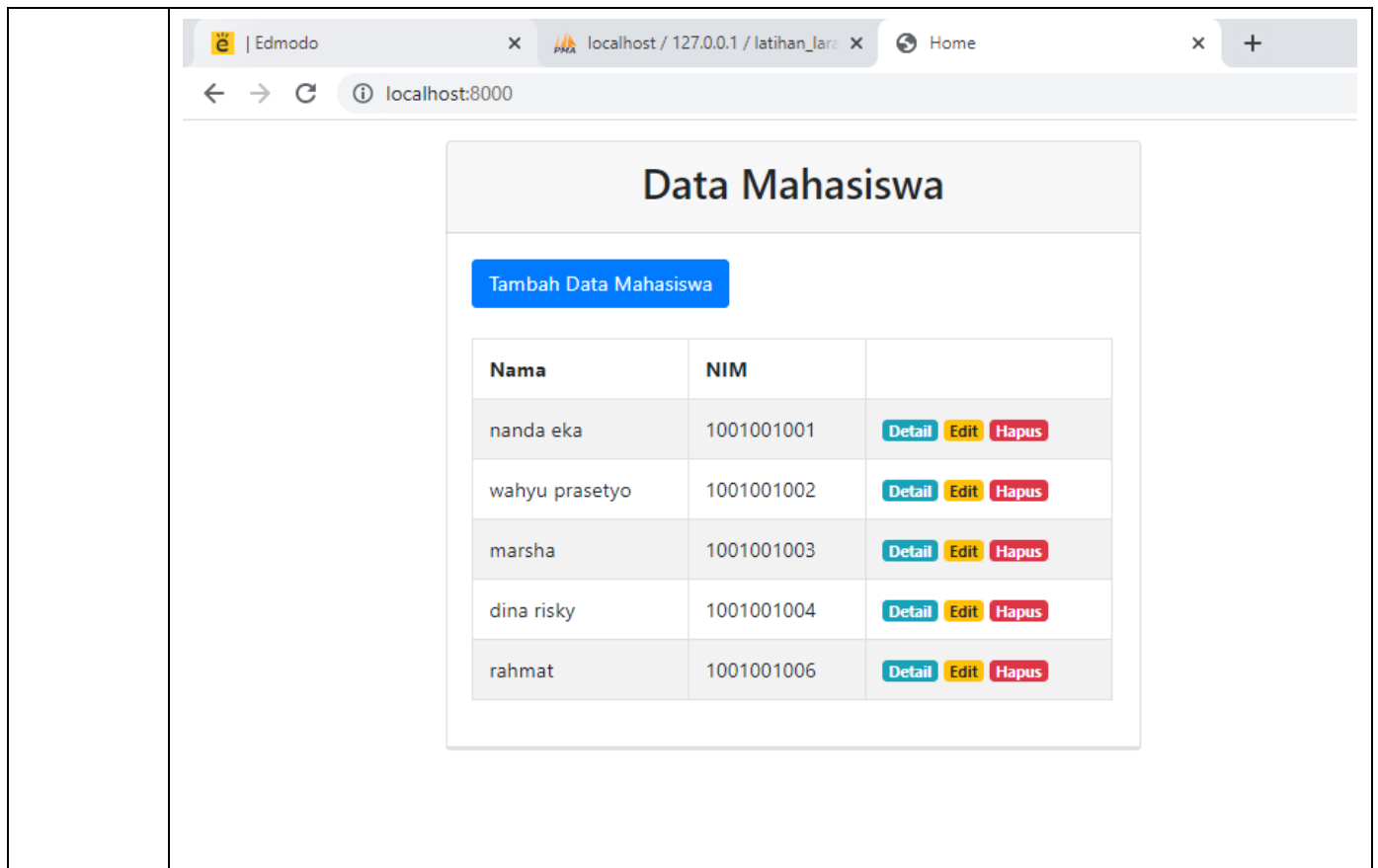
- 4 Jalankan localhost:8000 dan pilih tombol ‘Detail’ di suatu data yang ingin kita lihat detailnya.



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>Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');</pre>
2	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre>public function edit(\$id) { \$mahasiswa = Mahasiswa::find(\$id); return view('edit', ['mahasiswa' => \$mahasiswa]); }</pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih
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 > form > div.form-group > label 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 <form action="/mahasiswa/update/{{ \$mahasiswa->id }}" method="POST"> 15 {{ csrf_field() }} 16 <input type="hidden" name="id" value="{{ \$mahasiswa->id }}"> 17 <div class="form-group"> 18 <label for="namamhs">Nama</label> 19 <input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mahasiswa->nama }}">
 20 </div> 21 <div class="form-group"> 22 <label for="nimhs">NIM</label> 23 <input type="number" class="form-control" required="required" name="nimhs" value="{{ \$mahasiswa->nim }}">
 24 </div> 25 <div class="form-group"> 26 <label for="emailhs">E-mail</label> 27 <input type="email" class="form-control" required="required" name="emailhs" value="{{ \$mahasiswa->email }}">
 28 </div> 29 <div class="form-group"> 30 <label for="jurusanhs">Jurusan</label> 31 <input type="text" class="form-control" required="required" name="jurusanhs" value="{{ \$mahasiswa->jurusan }}">
 32 </div> 33 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button> 34 </form> 35 @endsection</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Line 11-31 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan - Line 11 terdapat action="/mahasiswa/update/{{ \$mahasiswa->id }}" yang menunjukkan routes /mahasiswa/update/{id} dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController

4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update/{id}. Oleh karena itu, kita buat terlebih dahulu route tersebut</p> <pre>Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update');</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php
5	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p> <pre>public function update(Request \$request) { \$mahasiswa = Mahasiswa::find(\$request->id); \$mahasiswa->nama = \$request->namamhs; \$mahasiswa->nim = \$request->nimmhs; \$mahasiswa->email = \$request->emailmhs; \$mahasiswa->jurusan = \$request->jurusanmhs; \$mahasiswa->save(); return redirect('/'); }</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Variabel \$request untuk menerima data yang akan ditambahkan ke database - Line 48-52 merupakan fungsi eloquent untuk update data ke tabel mahasiswa
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> 



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>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>public function hapus(\$id) { \$mahasiswa = Mahasiswa::find(\$id); \$mahasiswa->delete(); return redirect('/'); }</pre> <p>Keterangan : Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih</p>
3	Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.

Edmodolocalhost / 127.0.0.1 / latihan_jar...Home

localhost:8000

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu prasetyo	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
dina risky	1001001004	Detail Edit Hapus
rahmat	1001001006	Detail Edit Hapus

Edmodolocalhost / 127.0.0.1 / latihan_jar...Home

localhost:8000

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu prasetyo	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
dina risky	1001001004	Detail Edit Hapus