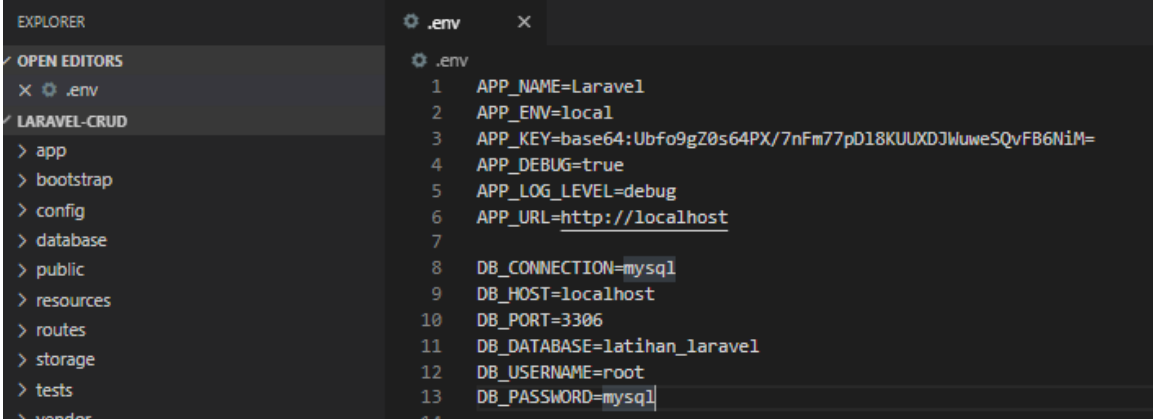

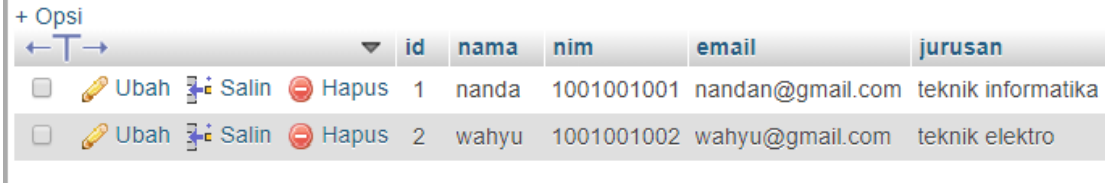


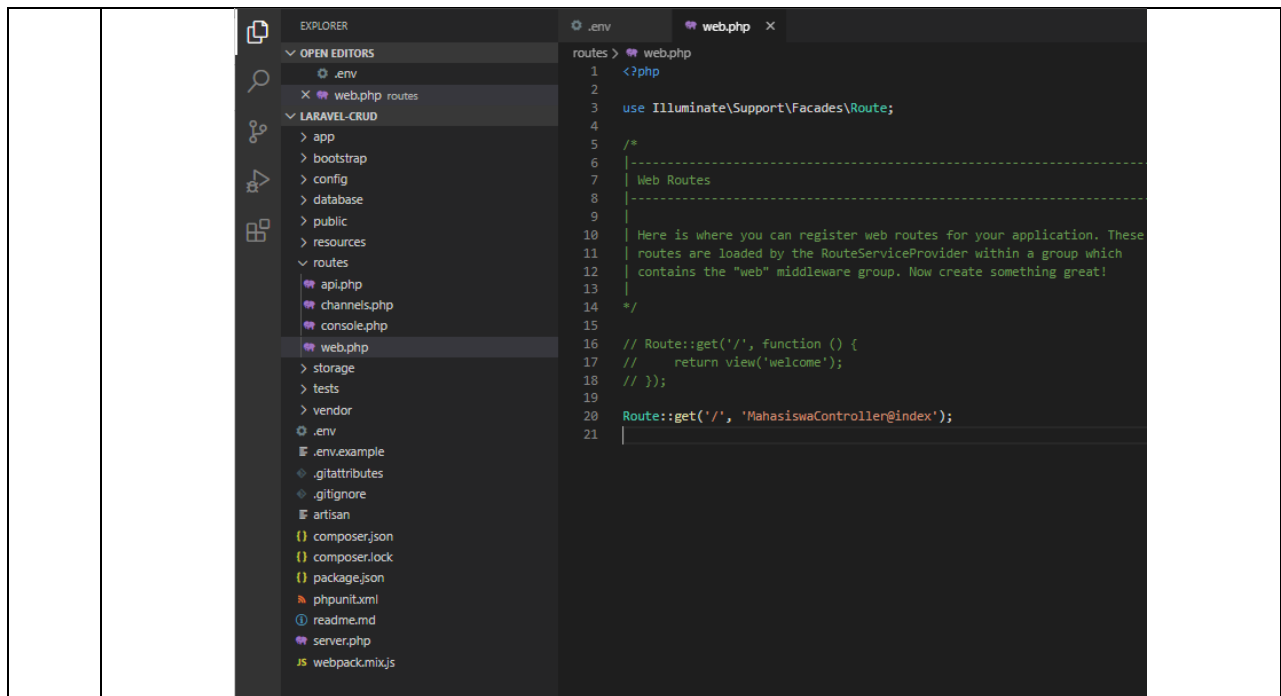
PEMROGRAMAN WEB LANJUT
JOBSHEET 10



Oleh:
SUGENG PRASTIYO
NIM. 1841720113

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020

No	<p style="text-align: center;">Uraian Jawaban Konfigurasi Database dan Pembuatan Tabel di MySQL</p>
1	<p>Buatlah project Laravel baru dengan nama laravel-crud. Buka command prompt, tuliskan perintah berikut. <code>cd C:\xampp\htdocs laravel new laravel-crud</code> Untuk membuat project/folder baru yang berisikan isi laravel</p>
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file <code>.env</code> 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 <code>latihan_laravel</code> dengan username <code>root</code>.
3	<p>Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database <code>latihan_laravel</code>.</p> 
4	<p>Isilah beberapa data pada tabel mahasiswa tersebut.</p> 
b.	<p style="text-align: center;">Menampilkan Data dari Database</p>
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan. Pertama, buatlah route pada <code>routes/web.php</code> 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

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:\Program Files\Ampms\www\laravel-crud>php artisan make:controller MahasiswaController Controller created successfully.</pre>
---	---

3	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p> <pre> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 use Illuminate\Support\Facades\DB; 7 8 class MahasiswaController extends Controller 9 { 10 public function index() 11 { 12 //mengambil data daritabel mahasiswa 13 \$mahasiswa = DB::table('mahasiswa')->get(); 14 15 return view('index', ['mahasiswa' => \$mahasiswa]); 16 } 17 } 18 19 </pre>
---	---

Keterangan:

	<ul style="list-style-type: none"> - 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	<p>Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat template blade (seperti file header-footer pada pembahasan CI) dengan nama master.blade.php pada folder resources/views.</p>  <p>Keterangan:</p> <ul style="list-style-type: none"> - 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	<p>Sekarang buat file index.blade.php yang menerapkan template master.blade.php dan akan digunakan untuk menampilkan data.</p>

```

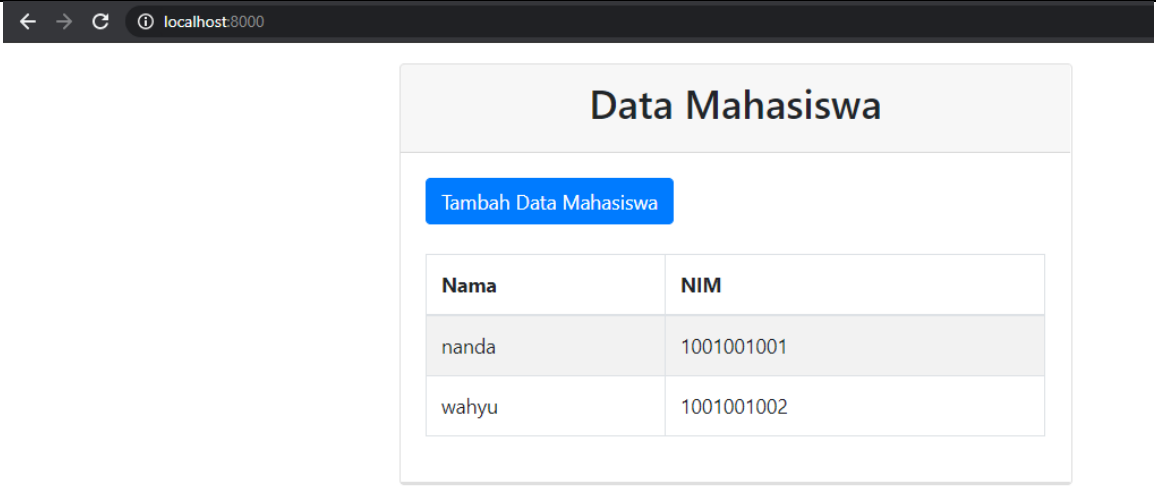
1  @extends('master')
2
3  <!-- isi title -->
4  @section('title', 'Home')
5  <!-- isi bagian judul halaman -->
6  @section('judul_halaman', 'Data Mahasiswa')
7
8  <!-- isi bagian konten -->
9  @section('konten')
10 <a href="/mahasiswa/tambah" class="btn btn-primary">Tambah Data Mahasiswa</a>
11 <br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

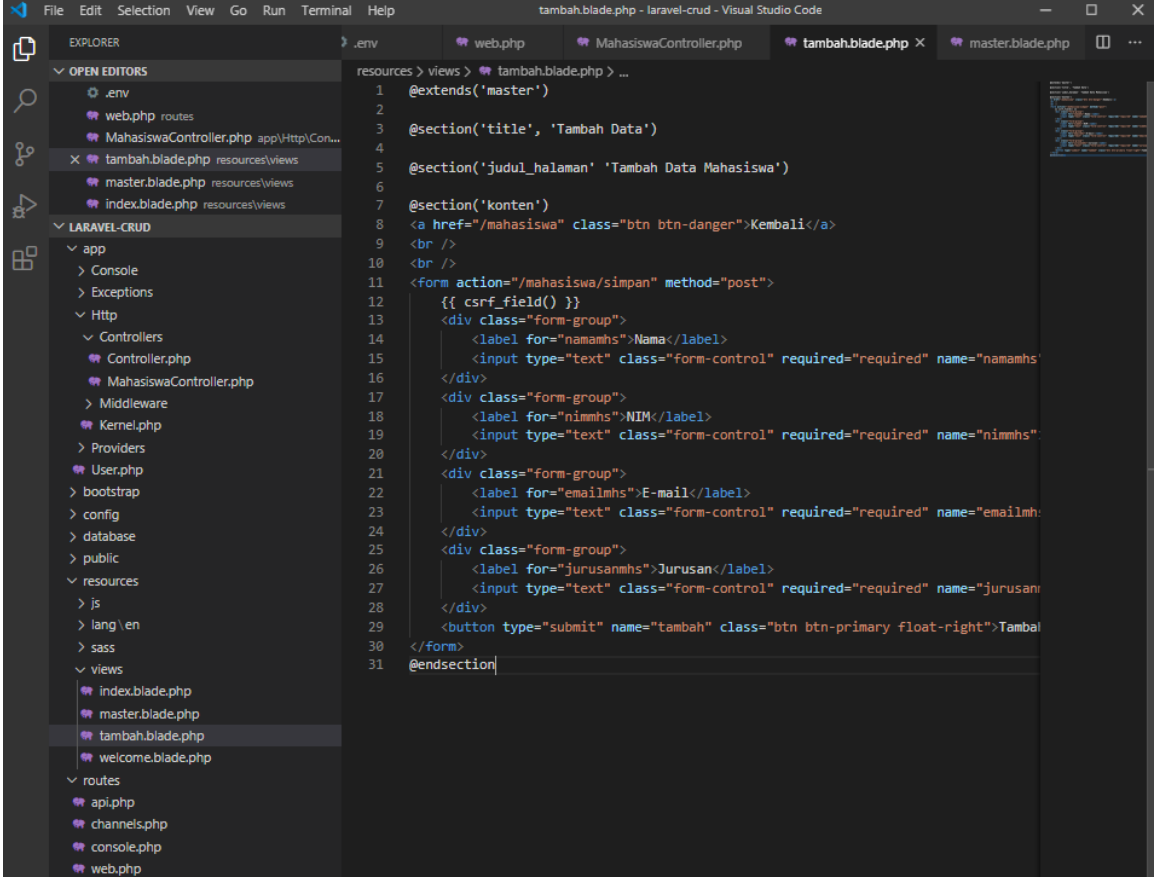
```

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
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController</p> <pre> 22 Route::get('/', 'MahasiswaController@tambah'); 23 </pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> 18 public function tambah() 19 { 20 return view('tambah'); 21 } 22 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>

	 <p>Keterangan:</p> <ul style="list-style-type: none"> - Line 13-32 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan - Line 13 terdapat <code>action="/mahasiswa/simpan"</code> yang menunjukkan routes <code>/mahasiswa/simpan</code> dimana data pada form tersebut akan dikirimkan ke fungsi <code>simpan</code> pada controller <code>MahasiswaController</code> - Line 14 terdapat <code>{{ csrf_field() }}</code> 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	<p>Ketika tombol simpan ditekan, akan dipanggil routes <code>/mahasiswa/simpan</code>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');</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 <code>simpan</code> di <code>MahasiswaController.php</code>
5	<p>Buat method <code>simpan</code> pada <code>MahasiswaController</code> untuk menyimpan data ke database.</p>

```

22 public function simpan(Request $request)
23 {
24     DB::table('mahasiswa')->insert([
25         'nama' => $request->namamhs,
26         'nim' => $request->nimmhs,
27         'email' => $request->emailmhs,
28         'jurusan' => $request->jurusanmhs
29     ]);
30     return redirect('/');
31 }
32 }
33

```

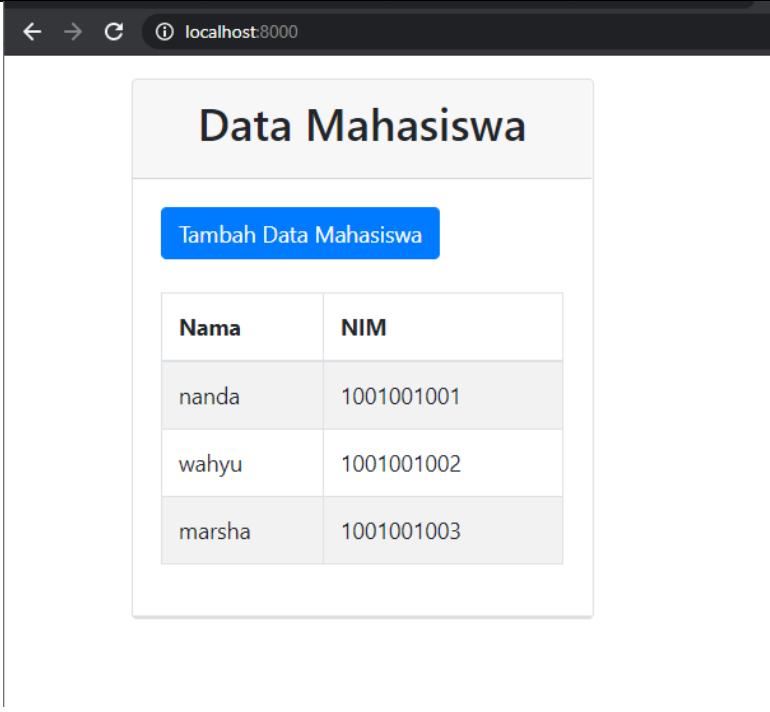
Keterangan:

- 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 content is a form titled 'Tambah Data Mahasiswa'. At the top left of the form is a red button labeled 'Kembali'. Below it are four input fields: 'Nama' with the value 'marsha', 'NIM' with the value '1001001003', 'E-mail' with the value 'marsha@gmail.com', and 'Jurusan' with the value 'teknik sipil'. At the bottom right of the form is a blue button labeled 'Tambah Data'.

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

	 <p>Kita dapat mencoba menambahkan beberapa data lagi agar jumlah data lebih banyak.</p>
D	Melihat Detail Data dari Database
1	<p>Buatlah tombol untuk melihat detail data (Line 28) pada file index.blade.php di folder resources/views</p> <pre> 23 <td> 24 id }}" class="badge badge-info">Detail 25 </td> 26 </tr> </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre> 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> 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 // 13 \$mahasiswa = DB::table('mahasiswa')->get(); 14 15 // mengirim data mahasiswa ke view index 16 return view('index', ['mahasiswa' => \$mahasiswa]); 17 } 18 public function tambah() 19 { 20 return view('tambah'); 21 } 22 public function simpan(Request \$request) 23 { 24 DB::table('mahasiswa')->insert([25 'nama' => \$request->namamhs, 26 'nim' => \$request->nimmhs, 27 'email' => \$request->emailmhs, 28 'jurusan' => \$request->jurusanmhs 29]); 30 return redirect('/'); 31 } 32 public function detail(\$id) 33 { 34 \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); 35 return view('detail', ['mahasiswa' => \$mahasiswa]); 36 } 37 } </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>

	<pre> resources > views > detail.blade.php > ... 1 @extends('master') 2 @section('title', 'Detail Mahasiswa') 3 @section('judul_halaman', 'Detail Data Mahasiswa') 4 5 @section('konten') 6 Kembali 7

 8 @foreach(\$mahasiswa as \$mhs) 9 <h5 class="card-title">{{ \$mhs->nama }}</h5> 10 <p class="card-text"> 11 <label for="">NIM:</label> 12 {{ \$mhs->nim }} 13 </p> 14 <p class="card-text"> 15 <label for="">EMAIL</label> 16 {{ \$mhs->email }} 17 </p> 18 <p class="card-text"> 19 <label for="">JURUSAN</label> 20 {{ \$mhs->jurusan }} 21 </p> 22 @endforeach 23 @endsection </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - 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.

	<div><div><div><div>← → ↻ ⓘ localhost:8000</div><div><div><div><div>Data Mahasiswa</div><div>Tambah Data Mahasiswa</div><table><tr><th>Nama</th><th>NIM</th><th></th></tr><tr><td>nanda</td><td>1001001001</td><td>Detail</td></tr><tr><td>wahyu</td><td>1001001002</td><td>Detail</td></tr><tr><td>marsha</td><td>1001001003</td><td>Detail</td></tr></table></div></div></div></div><div><div>← → ↻ ⓘ localhost:8000/mahasiswa/detail/1</div><div><div><div><div>Detail Data Mahasiswa</div><div><div>Kembali</div><div>nanda</div><div>NIM: 1001001001</div><div>EMAIL nandan@gmail.com</div><div>JURUSAN teknik informatika</div></div></div></div></div><div>Ini adalah hasildari program diatas yang menampilkan detail</div></div></div></div>	Nama	NIM		nanda	1001001001	Detail	wahyu	1001001002	Detail	marsha	1001001003	Detail
Nama	NIM												
nanda	1001001001	Detail											
wahyu	1001001002	Detail											
marsha	1001001003	Detail											
E	Mengubah Data (Update) dari Database												
1	Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views												

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

2

Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController

```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6  |-----
7  | Web Routes
8  |-----
9  |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 // Route::get('/', function () {
17 //     return view('welcome');
18 // });
19
20 Route::get('/', 'MahasiswaController@index');
21
22 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');
23
24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');
25
26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');
27
28
29 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');
30
```

3	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre> 37 public function edit(\$id) 38 { 39 \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); 40 return view('edit', ['mahasiswa' => \$mahasiswa]); 41 } 42 </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Line 39 : 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>  <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 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 30 31 Route::post('/mahasiswa/update', 'MahasiswaController@update'); 32 </pre>
6	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p>

```

public function update(Request $request)
{
    DB::table('mahasiswa')->where('id', $request->id)->update([
        'nama' => $request->namamhs,
        'nim' => $request->nimmhs,
        'email' => $request->emailmhs,
        'jurusan' => $request->jurusanmhs
    ]);
    return redirect('/');
}

```

Keterangan:

- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 58-63 merupakan query builder untuk update data ke tabel mahasiswa

7

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

Data Mahasiswa

[Tambah Data Mahasiswa](#)

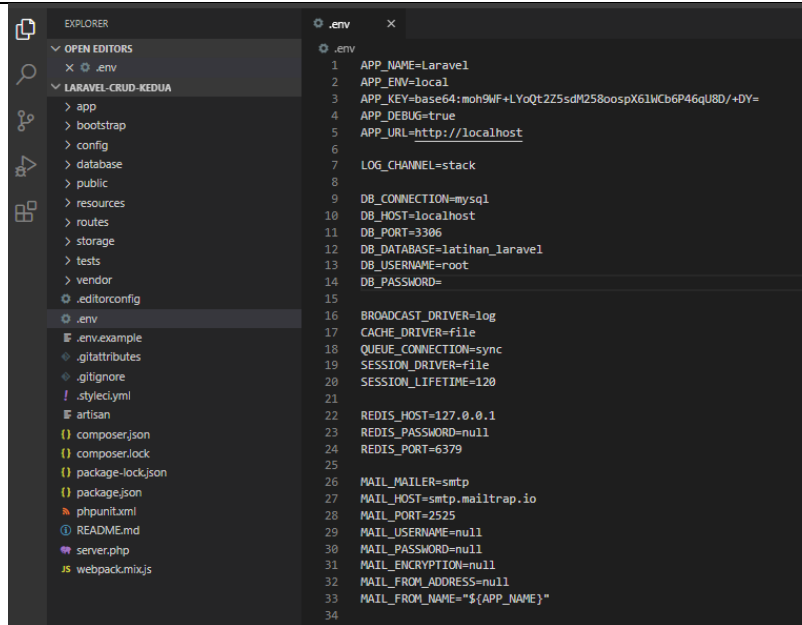
Nama	NIM	
nanda	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
marsha	1001001003	Detail Edit

Klik edit di data pertama, kita akan mengubah namanya.

	<div><div><h3>Edit Data Mahasiswa</h3><div>Kembali</div><div><div>Nama</div><div><input type="text" value="nanda eka"/></div></div><div><div>NIM</div><div><input type="text" value="1001001001"/></div></div><div><div>E-mail</div><div><input type="text" value="nandan@gmail.com"/></div></div><div><div>Jurusan</div><div><input type="text" value="teknik informatika"/></div></div><div>Simpan Data</div></div></div> <div><div><h3>Data Mahasiswa</h3><div>Tambah Data Mahasiswa</div><table><tr><th>Nama</th><th>NIM</th><th></th></tr><tr><td>nanda eka</td><td>1001001001</td><td>Detail Edit</td></tr><tr><td>wahyu</td><td>1001001002</td><td>Detail Edit</td></tr><tr><td>marsha</td><td>1001001003</td><td>Detail Edit</td></tr></table></div></div>	Nama	NIM		nanda eka	1001001001	Detail Edit	wahyu	1001001002	Detail Edit	marsha	1001001003	Detail Edit
Nama	NIM												
nanda eka	1001001001	Detail Edit											
wahyu	1001001002	Detail Edit											
marsha	1001001003	Detail Edit											
f.	Menghapus Data(Delete) dari Database												
1	Buatlah tombol untuk menghapus data (Line 30) pada file index.blade.php di folder resources/views												

	<pre> resources > views > index.blade.php > ... 1 @extends('master') 2 3 @section('title', 'Home') 4 5 @section('judul_halaman', 'Data Mahasiswa') 6 7 @section('konten') 8 Tambah Data Mahasiswa 9
 10
 11 <table class="table table-bordered table-hover table-striped"> 12 <thead> 13 <tr> 14 <th>Nama</th> 15 <th>NIM</th> 16 <th></th> 17 </tr> 18 </thead> 19 <tbody> 20 @foreach(\$mahasiswa as \$mhs) 21 <tr> 22 <td>{{ \$mhs->nama }}</td> 23 <td>{{ \$mhs->nim }}</td> 24 <td> 25 id }}" class="badge badge-info">Detail 26 id }}" class="badge badge-warning">Edit 27 id }}" class="badge badge-danger">Hapus 28 </td> 29 </tr> 30 @endforeach 31 </tbody> 32 </table> 33 @endsection </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> 33 34 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 35 </pre>
3	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p> <pre> 54 public function hapus(\$id) 55 { 56 DB::table('mahasiswa')->where('id', \$id)->delete(); 57 return redirect('/mahasiswa'); 58 } 59 } 60 </pre>
4	<p>Jalankan localhost:8000 dan pilih tombol ‘Hapus’, maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir.</p>

	<div><div><h3>Data Mahasiswa</h3><div>Tambah Data Mahasiswa</div><table><tr><th>Nama</th><th>NIM</th><th></th></tr><tr><td>nanda eka</td><td>1001001001</td><td>Detail Edit Hapus</td></tr><tr><td>wahyu</td><td>1001001002</td><td>Detail Edit Hapus</td></tr><tr><td>marsha</td><td>1001001003</td><td>Detail Edit Hapus</td></tr></table></div><div><h3>Data Mahasiswa</h3><div>Tambah Data Mahasiswa</div><table><tr><th>Nama</th><th>NIM</th><th></th></tr><tr><td>wahyu</td><td>1001001002</td><td>Detail Edit Hapus</td></tr><tr><td>marsha</td><td>1001001003</td><td>Detail Edit Hapus</td></tr></table></div></div>	Nama	NIM		nanda eka	1001001001	Detail Edit Hapus	wahyu	1001001002	Detail Edit Hapus	marsha	1001001003	Detail Edit Hapus	Nama	NIM		wahyu	1001001002	Detail Edit Hapus	marsha	1001001003	Detail Edit Hapus
Nama	NIM																					
nanda eka	1001001001	Detail Edit Hapus																				
wahyu	1001001002	Detail Edit Hapus																				
marsha	1001001003	Detail Edit Hapus																				
Nama	NIM																					
wahyu	1001001002	Detail Edit Hapus																				
marsha	1001001003	Detail Edit Hapus																				
	Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent																					
A	Konfigurasi Database																					
1	Buatlah project Laravel baru dengan nama laravel-crud-kedua. Buka command prompt, tuliskan perintah berikut. cd C:\xampp\htdocs laravel new laravel-crud-kedua																					
2	Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.																					



Keterangan:

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

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

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama Lainnya
2	nama	varchar(100)			Tidak	Tidak ada		Ubah Hapus Lainnya
3	nim	varchar(10)			Tidak	Tidak ada		Ubah Hapus Lainnya
4	email	varchar(50)			Tidak	Tidak ada		Ubah Hapus Lainnya
5	jurusan	varchar(20)			Tidak	Tidak ada		Ubah Hapus Lainnya
6	created_at	timestamp		Ya	NULL			Ubah Hapus Lainnya
7	updated_at	timestamp		Ya	NULL			Ubah Hapus Lainnya

B Menampilkan Data dari Database

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.

```

16 Route::get('/', 'MahasiswaController@index');
17
18 Route::get('/mahasiswa', 'MahasiswaController@index');
19

```

2 Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan cd laravel-crud-kedua php artisan make:model Mahasiswa

3 Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.

	<pre> app > Mahasiswa.php > PHP Intelephense > 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> <p>Keterangan:</p> <ul style="list-style-type: none"> - Model Mahasiswa akan menangani tabel mahasisw
4	Buat controller baru yaitu MahasiswaController menggunakan php artisan php artisan make:controller MahasiswaController
5	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p> <pre> <?php namespace App\Http\Controllers; use Illuminate\Http\Request; use App\Mahasiswa; class MahasiswaController extends Controller { public function index() { \$mahasiswa = Mahasiswa::all(); return view('index', ['mahasiswa' => \$mahasiswa]); } } </pre>
6	Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat template blade (seperti pada Bagian 1). Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)

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

Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM	
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus

C

Memasukkan Data (Create) Ke database

1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre> 19 20 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 21 </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 } </pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views.</p>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 23 24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 25 </pre>
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre> 20 21 public function simpan(Request \$request) 22 { 23 Mahasiswa::create([24 'nama' => \$request->namamhs, 25 'nim' => \$request->nimmhs, 26 'email' => \$request->emailmhs, 27 'jurusan' => \$request->jurusanmhs 28]); 29 return redirect('/'); 30 } </pre>
6	<p>Karena kita menggunakan fungsi create pada Controller, maka butuh ditambahkan code pada Line 10 yang disebut Mass Assignment pada model Mahasiswa.php. Mass Assignment digunakan untuk memfilter kolom mana yang boleh dan tidak boleh diinput.</p> <pre> 6 7 class Mahasiswa extends Model 8 { 9 protected \$table = "mahasiswa"; 10 protected \$fillable = ['nama', 'nim', 'email', 'jurusan']; 11 // 12 } 13 </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>

Tambah Data Mahasiswa

Kembali

Nama

rahmat

NIM

1001001006

E-mail

rahmat@gmail.com

Jurusan

teknik mesin

Tambah Data

Data Mahasiswa

Tambah Data Mahasiswa

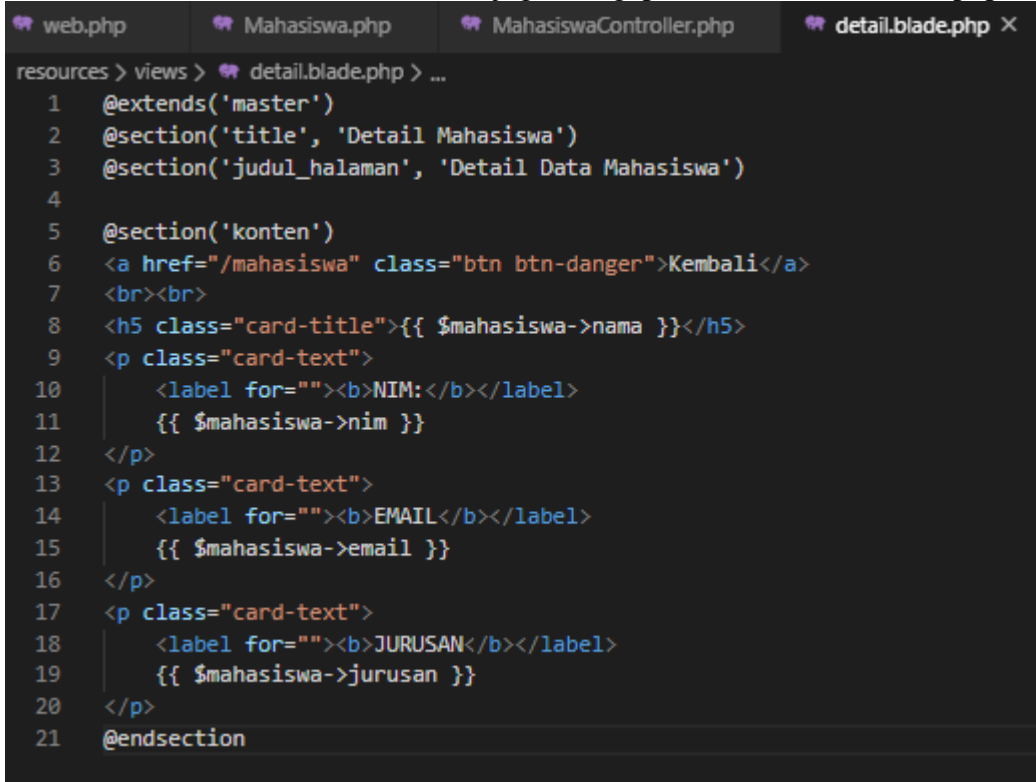
Nama	NIM	
wahyu	1001001002	<div>DetailEditHapus</div>
marsha	1001001003	<div>DetailEditHapus</div>
rahmat	1001001006	<div>DetailEditHapus</div>

D

Melihat Detail Data dari Database

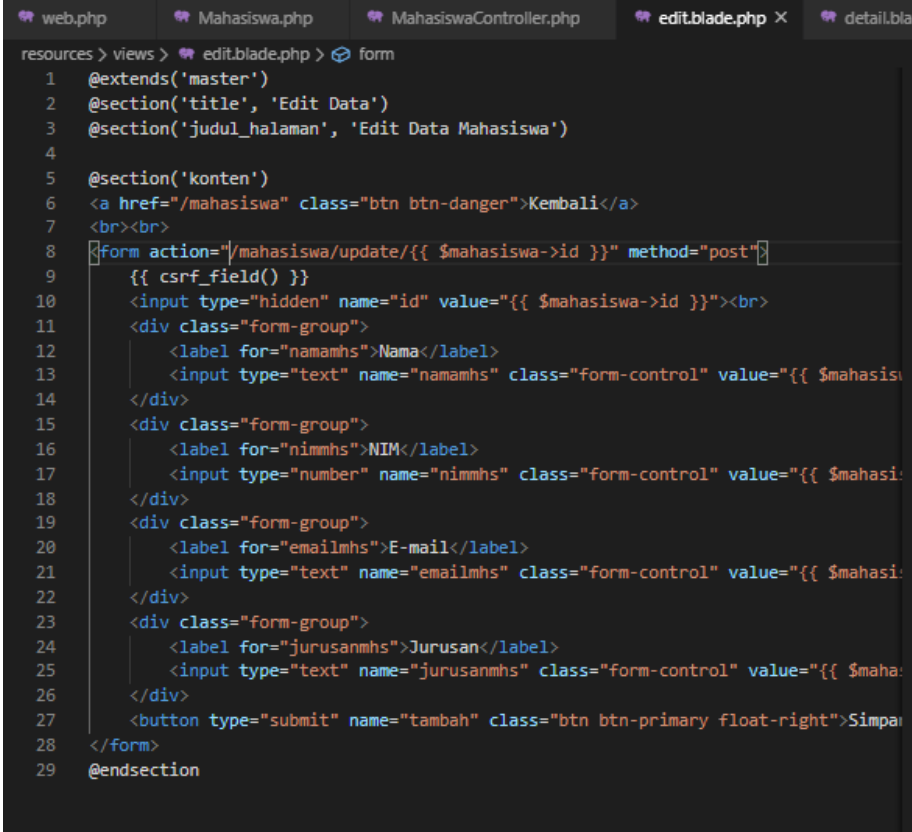
1

Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController

		<pre> 23 24 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 25 </pre>	
2	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p>	<pre> 31 32 public function detail(\$id) 33 { 34 \$mahasiswa = Mahasiswa::find(\$id); 35 return view('detail', ['mahasiswa' => \$mahasiswa]); 36 } 37 </pre>	
3	<p>Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php.</p>	 <pre> resources > views > detail.blade.php > ... 1 @extends('master') 2 @section('title', 'Detail Mahasiswa') 3 @section('judul_halaman', 'Detail Data Mahasiswa') 4 5 @section('konten') 6 Kembali 7

 8 <h5 class="card-title">{{ \$mahasiswa->nama }}</h5> 9 <p class="card-text"> 10 <label for="">NIM:</label> 11 {{ \$mahasiswa->nim }} 12 </p> 13 <p class="card-text"> 14 <label for="">EMAIL</label> 15 {{ \$mahasiswa->email }} 16 </p> 17 <p class="card-text"> 18 <label for="">JURUSAN</label> 19 {{ \$mahasiswa->jurusan }} 20 </p> 21 @endsection </pre>	
4	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p>		

	<div> <div>Detail Data Mahasiswa</div> <div> <div>Kembali</div> <div>wahyu</div> <div>NIM: 1001001002</div> <div>EMAIL wahyu@gmail.com</div> <div>JURUSAN teknik elektro</div> </div> </div>
E	Mengubah Data (Update)dari Database
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre> 25 26 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 27 </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>
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 1 @extends('master') 2 @section('title', 'Edit Data') 3 @section('judul_halaman', 'Edit Data Mahasiswa') 4 5 @section('konten') 6 Kembali 7

 8 <form action="/mahasiswa/update/{{ \$mahasiswa->id }}" method="post"> 9 {{ csrf_field() }} 10 <input type="hidden" name="id" value="{{ \$mahasiswa->id }}">
 11 <div class="form-group"> 12 <label for="namamhs">Nama</label> 13 <input type="text" name="namamhs" class="form-control" value="{{ \$mahasiswa->nama }}"> 14 </div> 15 <div class="form-group"> 16 <label for="nimmhs">NIM</label> 17 <input type="number" name="nimmhs" class="form-control" value="{{ \$mahasiswa->nim }}"> 18 </div> 19 <div class="form-group"> 20 <label for="emailmhs">E-mail</label> 21 <input type="text" name="emailmhs" class="form-control" value="{{ \$mahasiswa->email }}"> 22 </div> 23 <div class="form-group"> 24 <label for="jurusanmhs">Jurusan</label> 25 <input type="text" name="jurusanmhs" class="form-control" value="{{ \$mahasiswa->jurusan }}"> 26 </div> 27 <button type="submit" name="tambah" class="btn btn-primary float-right">Simpan 28 </form> 29 @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> 28 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update'); 29 </pre>
5	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p> <pre> public function update(\$id, Request \$request) { \$mahasiswa = Mahasiswa::find(\$id); \$mahasiswa->nama = \$request->namamhs; \$mahasiswa->nim = \$request->nimmhs; \$mahasiswa->email = \$request->emailmhs; \$mahasiswa->jurusan = \$request->jurusanmhs; \$mahasiswa->save(); return redirect('/'); } </pre>
6	<p>Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru. Cobalah untuk mengedit data.</p>

	<div><div><h3>Data Mahasiswa</h3><div><div>Tambah Data Mahasiswa</div><table><thead><tr><th>Nama</th><th>NIM</th><th></th></tr></thead><tbody><tr><td>wahyu</td><td>1001001002</td><td><div>DetailEditHapus</div></td></tr><tr><td>marsha</td><td>1001001003</td><td><div>DetailEditHapus</div></td></tr><tr><td>rahmat</td><td>1001001006</td><td><div>DetailEditHapus</div></td></tr></tbody></table></div></div></div> <div><div><h3>Data Mahasiswa</h3><div><div>Tambah Data Mahasiswa</div><table><thead><tr><th>Nama</th><th>NIM</th><th></th></tr></thead><tbody><tr><td>wahyu</td><td>1001001002</td><td><div>DetailEditHapus</div></td></tr><tr><td>marsha</td><td>1001001003</td><td><div>DetailEditHapus</div></td></tr><tr><td>rahmat suryadi</td><td>1001001006</td><td><div>DetailEditHapus</div></td></tr></tbody></table></div></div></div>	Nama	NIM		wahyu	1001001002	<div>DetailEditHapus</div>	marsha	1001001003	<div>DetailEditHapus</div>	rahmat	1001001006	<div>DetailEditHapus</div>	Nama	NIM		wahyu	1001001002	<div>DetailEditHapus</div>	marsha	1001001003	<div>DetailEditHapus</div>	rahmat suryadi	1001001006	<div>DetailEditHapus</div>
Nama	NIM																								
wahyu	1001001002	<div>DetailEditHapus</div>																							
marsha	1001001003	<div>DetailEditHapus</div>																							
rahmat	1001001006	<div>DetailEditHapus</div>																							
Nama	NIM																								
wahyu	1001001002	<div>DetailEditHapus</div>																							
marsha	1001001003	<div>DetailEditHapus</div>																							
rahmat suryadi	1001001006	<div>DetailEditHapus</div>																							
F	Menghapus Data (Delete) dari Database																								
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre>29 30 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 31</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>																								
3	Jalankan localhost:8000 dan pilih tombol ‘Hapus’, maka data yang terpilih akan dihapus.																								

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
rahmat suryadi	1001001006	Detail Edit Hapus

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus

Sudah terhapus