

LAPORAN JOOBSHEET 11

(Membuat CRUD)

Disusun sebagai

MATA KULIAH :

Pemrograman Web Lanjut



TI-2B

OLEH

FACKO ELLANDA (1841720154)

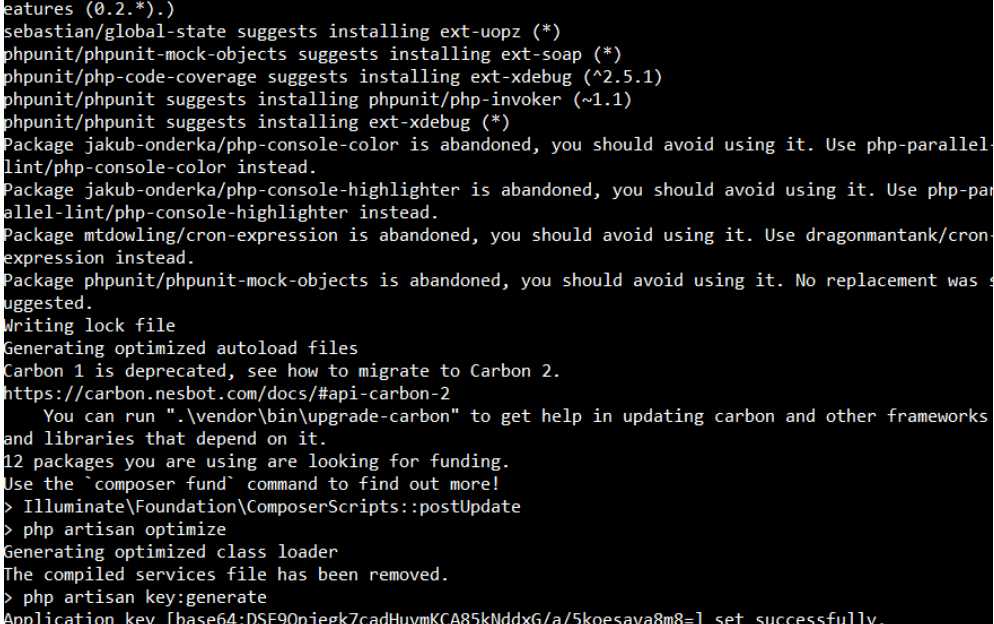
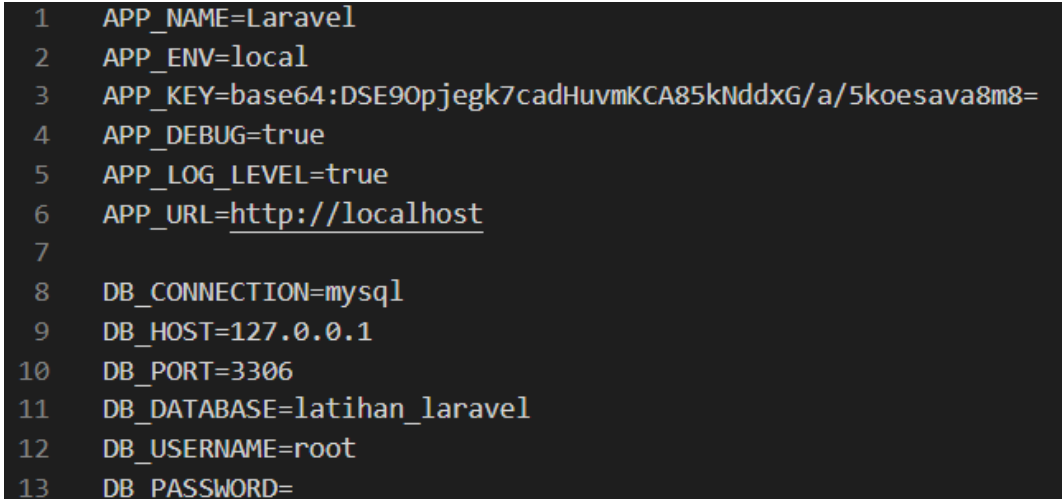
PROGAM STUDI D-IV TEKNIK INFORMATIKA



JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

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 composer create-project --prefer-dist laravel/laravel 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> <p>Keterangan:</p> 

	- Nama database yang akan digunakan adalah latihan_laravel dengan username root.
3	<p>Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database latihan_laravel.</p> 
4	<p>Isilah beberapa data pada tabel mahasiswa tersebut.</p> 

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>  <p>Keterangan:</p> <ul style="list-style-type: none"> - 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>

```
C:\xampp\htdocs\laravel-crud>php artisan make:controller MahasiswaController  
Controller created successfully.
```

--	--

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         $mahasiswa = DB::table('mahasiswa')->get();
12
13         return view('index',['mahasiswa' => $mahasiswa]);
14     }
15 }
16
17

```

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**.

```

resources > views > master.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" rel="stylesheet">
7      <title >@yield('title')</title>
8  </head>
9  <body>
10     <div class="container">
11         <div class="row mt-3">
12             <div class="col-md-6">
13                 <div class="card">
14                     <div class="card-header text-center">
15                         <h2> @yield('judul_halaman')</h2>
16                     </div>
17                     <div class="card-body">
18
19                         @yield('konten')
20
21                     </div>
22                 </div>
23             </div>
24         </div>
25     </div>
26 </body>
27 </html>
28
29

```

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.

```
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>Name</th>
15                 <th>NIM</th>
16             </tr>
17         </thead>
18         <tbody>
19             @foreach($mahasiswa as $mhs)
20                 <tr>
21                     <td>{{ $mhs->nama }}</td>
22                     <td>{{ $mhs->nim }}</td>
23                 </tr>
24             @endforeach
25         </tbody>
26     </table>
27 @endsection
```

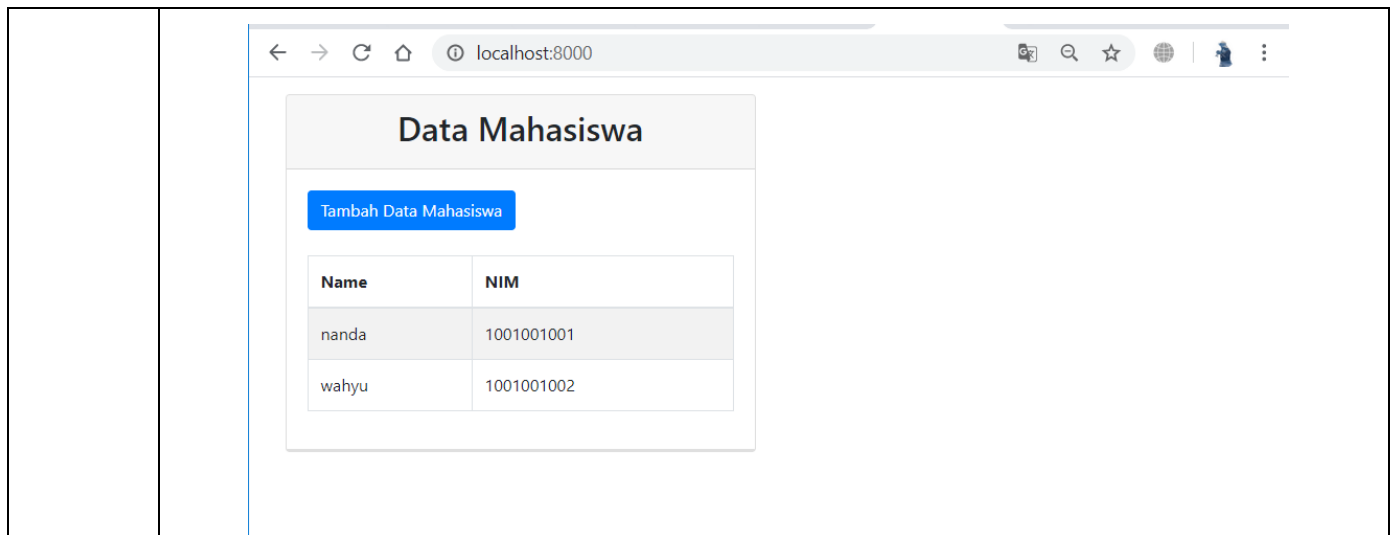
Keterangan:

- Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan blade lain yaitu master.blade.php
- Line 4 : @section('title', 'Home') berarti bahwa file index mengisi @yield('title') pada master dengan 'Home'
- Line 7 : @section('judul_halaman', 'Data Mahasiswa') berarti bahwa file index mengisi @yield('judul_halaman') pada master dengan 'Home'
- Line 10-30 : mengisi yield(@konten), karena terdapat banyak baris pada diawali dengan @section('konten') dan diakhiri dengan @endsection
- Line 24-25 menampilkan data mahasiswa dengan kolom nama dan nim

6

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

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



c. Memasukkan Data (Create) ke Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController</p> <pre>Route::get('/mahasiswa/tambah','MahasiswaController@tambah');</pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre>16 public function tambah(){ 17 return view('tambah'); 18 } 19</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>

```

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.

```
21 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');
```

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.


```

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

```

Keterangan:

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

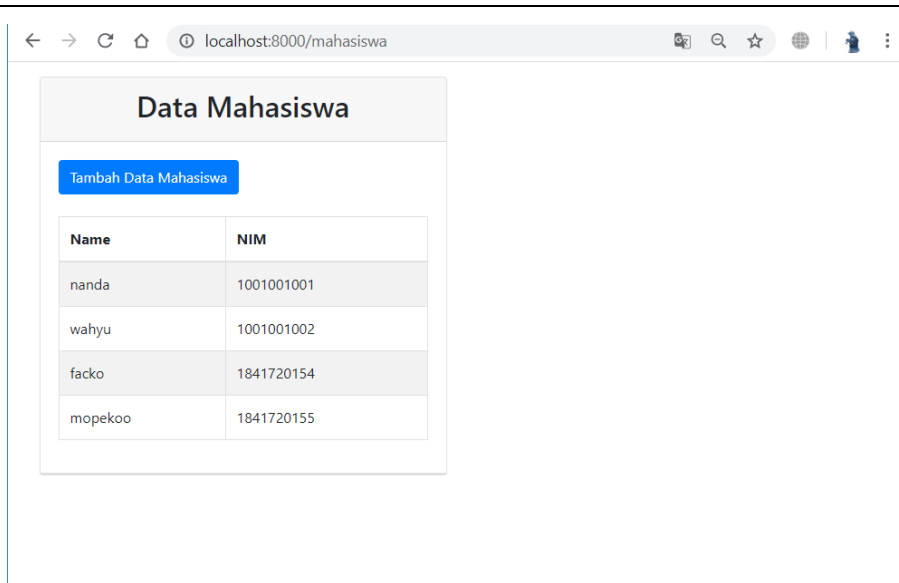
6

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**: facko
- NIM**: 1841720153
- E-mail**: facko@gmail.com
- Jurusan**: teknik sipil
- Tambah Data** (button)

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

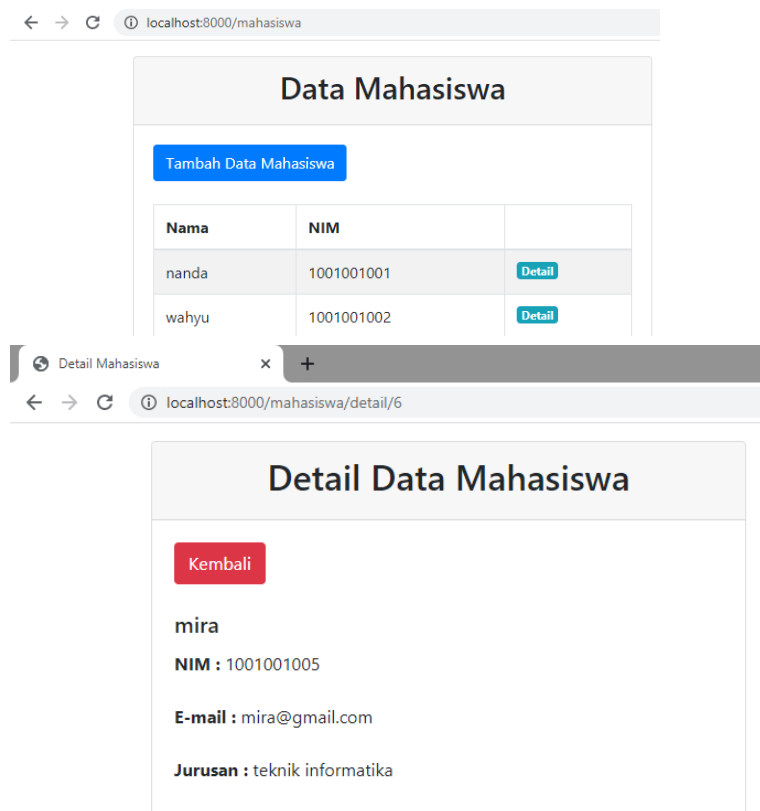


Kita dapat mencoba menambahkan beberapa data lagi agar jumlah data lebih banyak.

d. Melihat Detail Data dari Database

Langkah	Keterangan
1	<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 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>Name</th> 15 <th>NIM</th> 16 </tr> 17 </thead> 18 <tbody> 19 @foreach(\$mahasiswa as \$mhs) 20 <tr> 21 <td>{{ \$mhs->nama }}</td> 22 <td>{{ \$mhs->nim }}</td> 23 id }}" class="badge badge-info">Detail 24 </tr> 25 @endforeach 26 </tbody> 27 </table> 28 @endsection </pre>
2	<p>Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre> 22 23 Route::get('/mahasiswa/detail(id)', 'MahasiswaController@detail'); </pre>
3	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p>

	<pre> 80 public function detail(\$id){ 81 \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); 82 83 return view('detail',['mahasiswa' =>\$mahasiswa]); 84 } 85 </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 3 @section('title', 'Detail Mahasiswa') 4 5 @section('judul_halaman', 'Detail Data Mahasiswa') 6 7 @section('konten') 8 Kembali 9 10
 11
 12 13 @foreach (\$mahasiswa as \$mhs) 14 <h5 class="card-title">{{ \$mhs->nama }}</h5> 15 <p class="card-text"> 16 <label for=""> NIM : </label> 17 {{ \$mhs->nim }} 18 </p> 19 <p class="card-text"> 20 <label for=""> Email : </label> 21 {{ \$mhs->email }} 22 </p> 23 <p class="card-text"> 24 <label for=""> Jurusan : </label> 25 {{ \$mhs->jurusan }} 26 </p> 27 @endforeach 28 @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	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p>

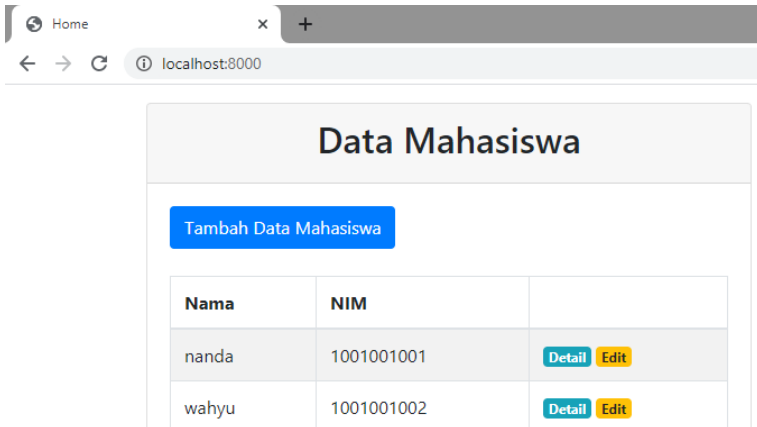


e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views</p> <pre> 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 </td> 28 </tr> 29 @endforeach 30 </tbody> 31 </table> 32 @endsection </pre>
2	<p>Buatlah <i>route</i> 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>
3	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre>public function edit(\$id){ \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); return view('edit', ['mahasiswa' => \$mahasiswa]); }</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 1 @extends('master') 2 3 @section('title', 'Edit Data') 4 5 @section('judul_halaman', 'Edit Data Mahasiswa') 6 7 @section('konten') 8 Kembali 9
 10
 11 @foreach (\$mahasiswa as \$mhs) 12 <form action="/mahasiswa/update" method="POST"> 13 {{ csrf_field() }} 14 <input type="hidden" name="id" value="{{ \$mhs->id }}"> 15 <div class="form-group"> 16 <label for="namamhs">Nama</label> 17 <input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mhs->nama }}">
 18 </div> 19 <div class="form-group"> 20 <label for="nimmhs">NIM</label> 21 <input type="number" class="form-control" required="required" name="nimmhs" value="{{ \$mhs->nim }}">
 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" value="{{ \$mhs->email }}">
 26 </div> 27 <div class="form-group"> 28 <label for="jurusanmhs">Jurusan</label> 29 <input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mhs->jurusan }}">
 30 </div> 31 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button> 32 </form> 33 @endforeach 34 @endsection 35</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 data-bbox="284 203 1485 286">Route::post('/mahasiswa/update', '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
6	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p> <pre> 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('/mahasiswa'); } </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>

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

Nama	NIM	
nanda eka	1001001001	Detail Edit
wahyu	1001001002	Detail Edit

f. Menghapus Data (Delete) dari Database

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


```

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                     <a href="/mahasiswa/hapus/{{ $mhs->id }}" class="badge badge-danger">Hapus</a>
28                 </td>
29             </tr>
30         @endforeach
31     </tbody>
32 </table>
33 @endsection

```

2

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

```
Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');
```

3

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

```

public function hapus($id){
    DB::table('mahasiswa')->where('id', $id)->delete();

    return redirect('/mahasiswa');
}

```

Keterangan :

- Line 67 : query builder untuk menghapus data mahasiswa berdasarkan id yang dipilih

4

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

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus

