



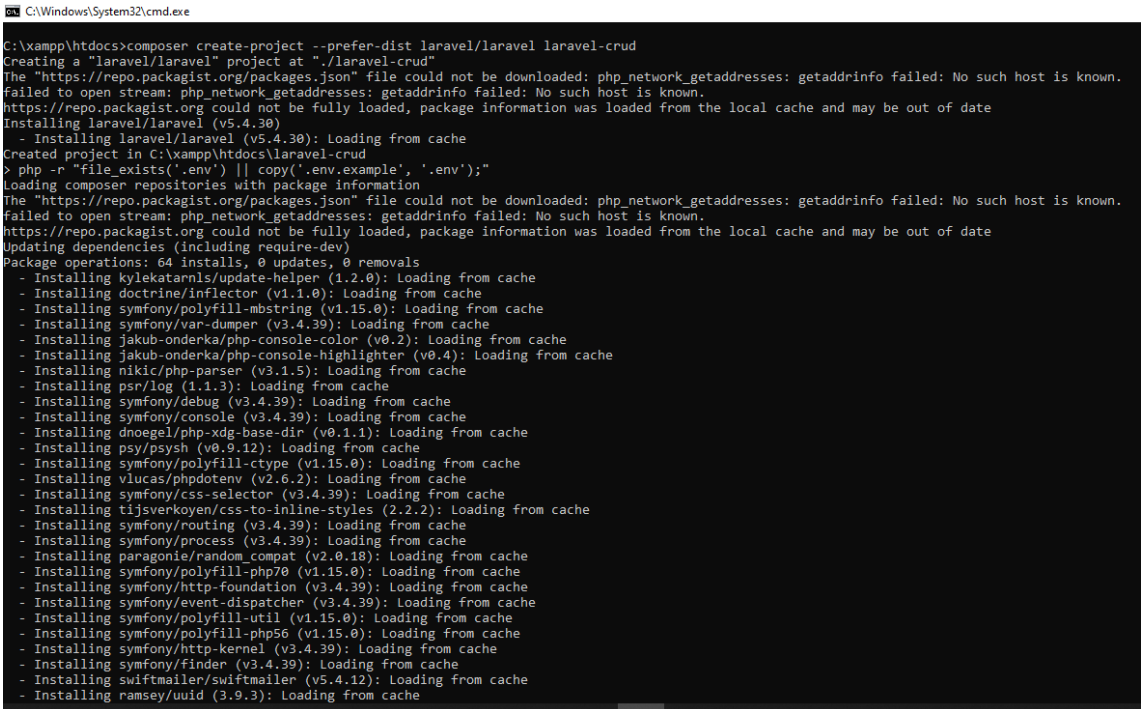
Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: Membuat CRUD menggunakan Laravel
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2020

NAMA : HUSNUL HOTIMAH

NIM : 1841720014

KELAS : TI-2A

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>C:\xampp\htdocs>composer create-project --prefer-dist laravel/laravel laravel-crud</pre> 

2

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

```

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:dPWcm+eRfeqgtkQYr/kS2gQC3muJAughzPIGJ9ko3Jw=
4 APP_DEBUG=true
5 APP_LOG_LEVEL=debug
6 APP_URL=http://localhost
7
8 LOG_CHANNEL=stack
9
10 DB_CONNECTION=mysql
11 DB_HOST=localhost
12 DB_PORT=3306
13 DB_DATABASE=latihan_laravel
14 DB_USERNAME=root
15 DB_PASSWORD=
16
17 BROADCAST_DRIVER=log
18 CACHE_DRIVER=file
19 SESSION_DRIVER=file
20 QUEUE_DRIVER=sync
21
22 REDIS_HOST=127.0.0.1
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
25
26 MAIL_DRIVER=smtp
27 MAIL_HOST=smtp.mailtrap.io
28 MAIL_PORT=2525
29 MAIL_USERNAME=null
30 MAIL_PASSWORD=null
31 MAIL_ENCRYPTION=null

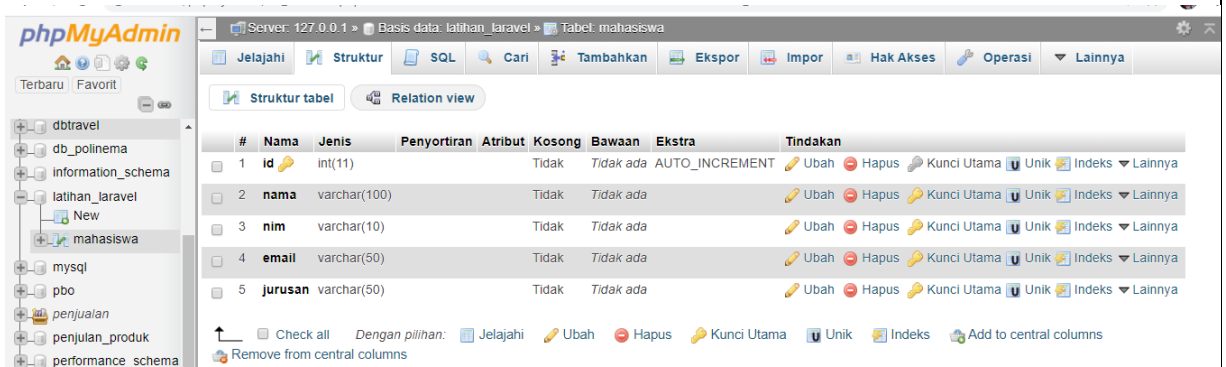
```

Keterangan:

Nama database yang akan digunakan adalah latihan_laravel dengan username root.

3

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

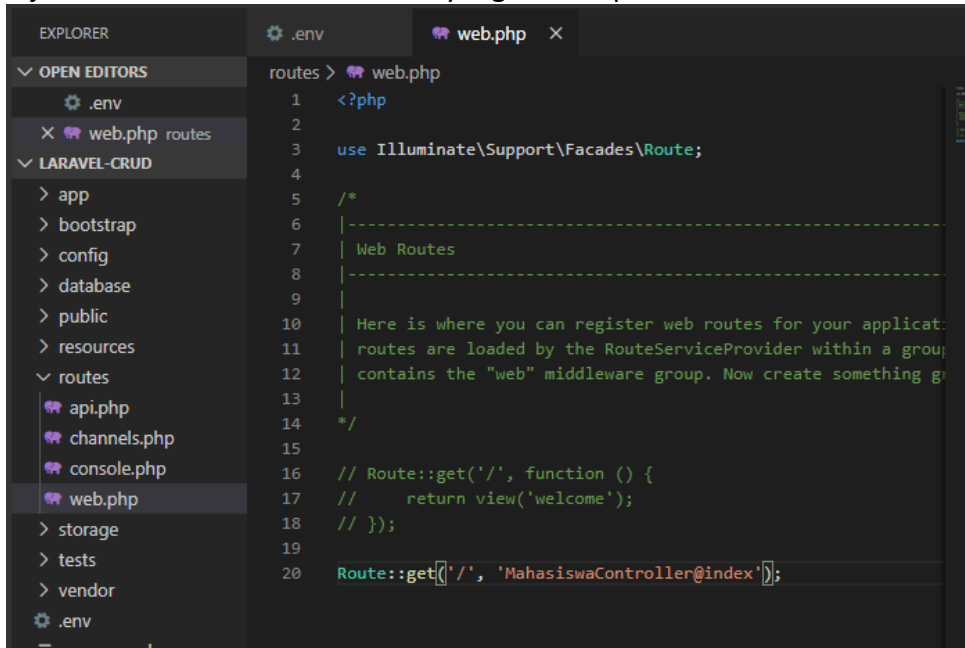
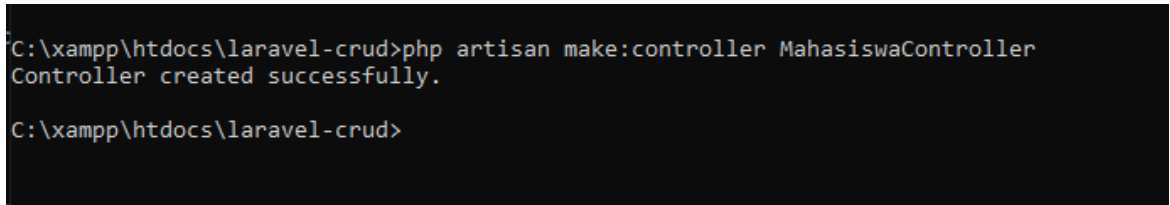


4

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



b. Menampilkan Data dari Database

Langkah	Keterangan
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan. Pertama, buatlah <i>route</i> pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p>  <pre> 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 applicat 11 routes are loaded by the RouteServiceProvider within a group 12 contains the "web" middleware group. Now create something g 13 14 */ 15 16 // Route::get('/', function () { 17 // return view('welcome'); 18 // }); 19 20 Route::get('/', 'MahasiswaController@index');</pre> <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> <pre>cd laravel-crud php artisan make:controller MahasiswaController</pre>  <pre> C:\xampp\htdocs\laravel-crud>php artisan make:controller MahasiswaController Controller created successfully. C:\xampp\htdocs\laravel-crud></pre>

3

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

```

1  use Illuminate\Http\Request;
2  use Illuminate\Support\Facades\DB;
3
4  class MahasiswaController extends Controller
5  {
6
7      public function index()
8      {
9          // mengambil data dari tabel mahasiswa
10         $mahasiswa = DB::table('mahasiswa')->get();
11
12         // mengambil data mahasiswa ke view index
13         return view('index', ['mahasiswa' => $mahasiswa]);
14     }
15 }

```

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=1.0">
6      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css">
7      <title>Document</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                         @yield('konten')
19                     </div>
20                 </div>
21             </div>
22         </div>
23     </div>
24 </body>
25 </html>

```

Keterangan:

- Fungsi @yield pada line 6(title), 15(judul_halaman), dan 19(konten) berfungsi sebagai penanda bagian-bagian pada master blade. Nantinya bagian @yield ini akan diisi sesuai dengan halaman view yang menerapkan master.blade.php

5

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

```

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

```

Keterangan:

- Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan bladelain 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

← → ↻ ⓘ localhost:8000

Data Mahasiswa

Tambah Data Mahasiswa

nama	NIM
Husnul Hotimah	1841720014
Indah Sekar	1841720055

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 23 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> 18 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>  <pre> resources > views > tambah.blade.php > ... 1 @extends('master') 2 3 <!-- isi title --> 4 @section('title', 'Tambah Data') 5 6 <!-- isi bagian judul halaman --> 7 @section('judul_halaman', 'Tambah Data Mahasiswa') 8 9 <!-- isi bagian konten --> 10 @section('konten') 11 Kembali 12
 13
 14 <form action="/mahasiswa/simpan" method="post"> 15 {{ csrf_field() }} 16 <div class="form-group"> 17 <label for="namamhs">Nama</label> 18 <input type="text" class="form-control" required="required" name="namamhs"> 19 </div> 20 <div class="form-group"> 21 <label for="nimhs">Nim</label> 22 <input type="text" class="form-control" required="required" name="nimhs"> 23 </div> 24 <div class="form-group"> 25 <label for="emailhs">Email</label> 26 <input type="text" class="form-control" required="required" name="emailhs"> 27 </div> 28 <div class="form-group"> 29 <label for="jurusanhs">Jurusan</label> 30 <input type="text" class="form-control" required="required" name="jurusanhs"> 31 </div> 32 <button type="submit" name="tambah" class="btn btn-primary float-right"> Tambah Data </button> 33 </form> 34 @endsection</pre>

	<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 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 <i>form hidden</i>.
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. 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 simpan di MahasiswaController.php
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre>public function simpan(Request \$request) { // insert data ke tabel mahasiswa DB::table('mahasiswa')->insert(['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 <p>Line 28-33 merupakan query builder untuk insert data ke tabel mahasiswa</p>

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.

Tambah Data Mahasiswa

[Kembali](#)

Nama
Zaskia Sungkar

NIM
1841720012

E-mail
zaskia@gmail.com

Jurusan
Teknik Elektro

[Tambah Data](#)

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

Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM
Husnul Hotimah	1841720014
Indah Sekar	1841720055
Zaskia Sungkar	1841720012

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> 1 @extends('master') 2 3 <!-- isi title --> 4 @section('title', 'Home') 5 6 <!-- isi bagian judul halaman --> 7 @section('judul_halaman', 'Data Mahasiswa') 8 9 <!-- isi bagian konten --> 10 @section('konten') 11 Tambah Data Mahasiswa 12
 13
 14 <table class="tabel table-bordered table-hover table-striped"> 15 <thead> 16 <tr> 17 <th>Nama</th> 18 <th>NIM</th> 19 </tr> 20 </thead> 21 <tbody> 22 @foreach(\$mahasiswa as \$mhs) 23 <tr> 24 <td>{{ \$mhs->nama }}</td> 25 <td>{{ \$mhs->nim }}</td> 26 <td> 27 id }}" class="btn btn-info">Detail 28 </td> 29 </tr> 30 @endforeach 31 </tbody> 32 </table> 33 @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> 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> public function detail(\$id) { // mengambil data mahasiswa berdasarkan id yang dipilih \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); /// kirim data mahasiswa yang diambil ke view detail.blade.php return view('detail', ['mahasiswa' => \$mahasiswa]); } </pre>

4

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

```

swaController.php  detail.blade.php X  tambah.blade.php  mast
resources > views > detail.blade.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="/" class="btn btn-danger">Kembali</a>
12     <br/>
13     <br/>
14     @foreach($mahasiswa as $mahasiswa)
15         <h5 class="card-title"> {{ $mhs->nama }} </h5>
16         <p class="card-text">
17             <label for=""><b>NIM : </b></label>
18             {{ $mhs->nim }}
19         </p>
20         <p class="card-text">
21             <label for=""><b>E-mail : </b></label>
22             {{ $mhs->email }}
23         </p>
24         <p class="card-text">
25             <label for=""><b>Jurusan : </b></label>
26             {{ $mhs->jurusan }}
27         </p>
28     @endforeach
29 @endsection

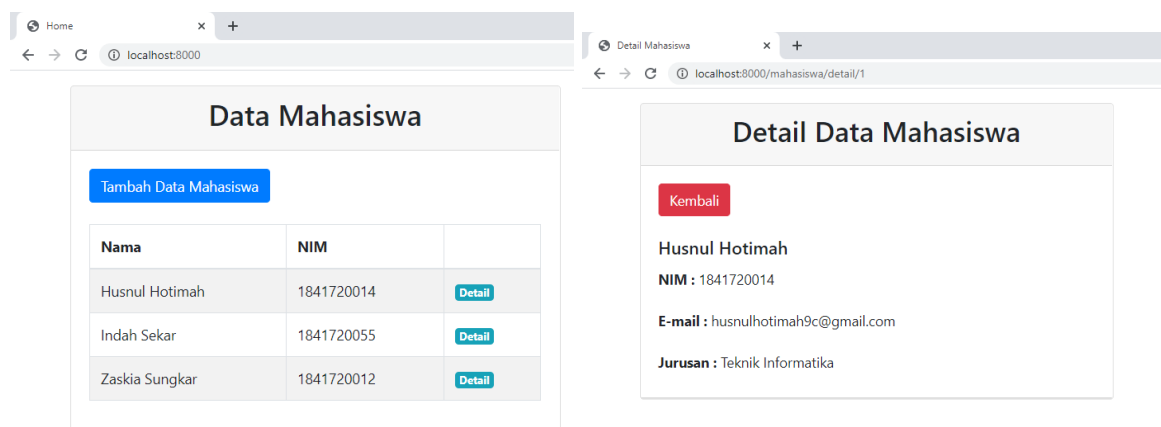
```

Keterangan:

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

5

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



e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views</p>  <pre> 2 3 <!-- isi title --> 4 @section('title', 'Home') 5 6 <!-- isi bagian judul halaman --> 7 @section('judul_halaman', 'Data Mahasiswa') 8 9 <!-- isi bagian konten --> 10 @section('konten') 11 Tambah Data Mahasiswa 12
 13
 14 <table class="tabel table-bordered table-hover table-striped"> 15 <thead> 16 <tr> 17 <th>Nama</th> 18 <th>NIM</th> 19 </tr> 20 </thead> 21 <tbody> 22 @foreach(\$mahasiswa as \$mhs) 23 <tr> 24 <td>{{ \$mhs->nama }}</td> 25 <td>{{ \$mhs->nim }}</td> 26 <td> 27 id }}" class="btn btn-info">Detail 28 id }}" class="btn btn-warning">Edit 29 </td> 30 </tr> 31 @endforeach 32 </tbody> 33 </table> 34 @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) { // mengambil data mahasiswa berdasarkan id yang dipilih \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); /// kirim data mahasiswa yang diambil ke view edit.blade.php 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

Kemudian buatlah view **edit.blade.php** yang berisi form untuk mengubah data pada folder **resources/views**. View edit juga mengaplikasikan master.blade.php.

```
resources > views > edit.blade.php > ...
1  @extends('master')
2  <!-- isi title -->
3  @section('title', 'Edit Data')
4  <!-- isi bagian judul halaman -->
5  @section('judul_halaman', 'Edit Data Mahasiswa')
6  <!-- isi bagian konten -->
7  @section('konten')
8      <a href="/" class="btn btn-danger">Kembali</a>
9      <br/>
10     <br/>
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 }}"> <br/>
18             </div>
19             <div class="form-group">
20                 <label for="nimmhs">NIM</label>
21                 <input type="text" class="form-control" required="required" name="nimmhs" value="{{ $mhs->nim }}"> <br/>
22             </div>
23             <div class="form-group">
24                 <label for="emailmhs">E-mail</label>
25                 <input type="text" class="form-control" required="required" name="emailmhs" value="{{ $mhs->email }}"> <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" value="{{ $mhs->jurusan }}"> <br/>
30             </div>
31             <button type="submit" name="edit" class="btn btn-primary float-right"> Simpan Data </button>
32         </form>
33     @endforeach
34 @endsection
```

Keterangan:

- 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

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

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

Keterangan:

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

6

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

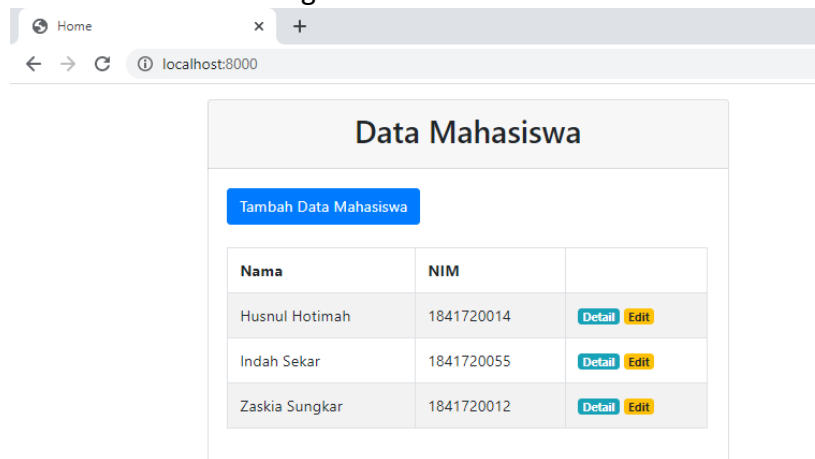
```
public function update(Request $request)
{
    // Update data mahasiswa
    DB::table('mahasiswa')->where('id', $request->id)->update([
        'nama' => $request->namamhs,
        'nim' => $request->nimmhs,
        'email' => $request->emailmhs,
        'jurusan' => $request->jurusanmhs
    ]);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

Keterangan:

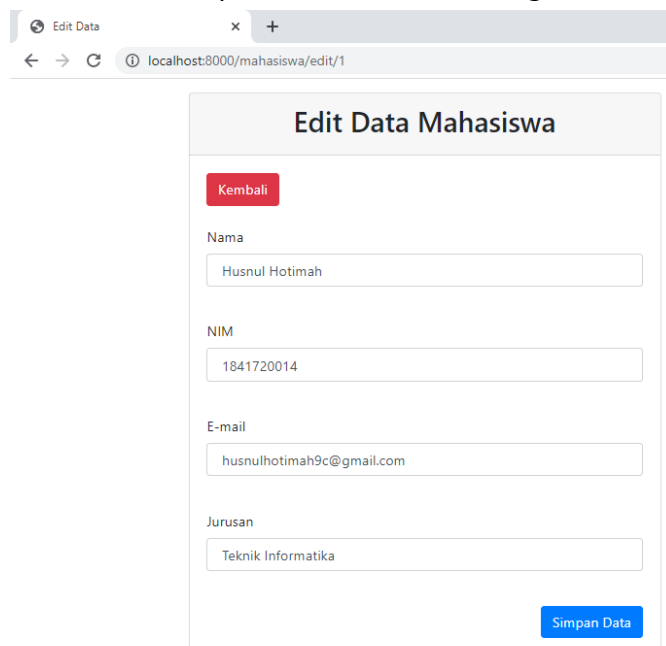
- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 58-63 merupakan query builder untuk update data ke tabel mahasiswa

7

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



Klik edit di data pertama, kita akan mengubah namanya.



Kemudian kita edit namanya menjadi Husnul

Browser: Edit Data x +
localhost:8000/mahasiswa/edit/1

Edit Data Mahasiswa

[Kembali](#)

Nama

NIM

E-mail

Jurusan

[Simpan Data](#)

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

Browser: Home x +
localhost:8000

Data Mahasiswa

[Tambah Data Mahasiswa](#)

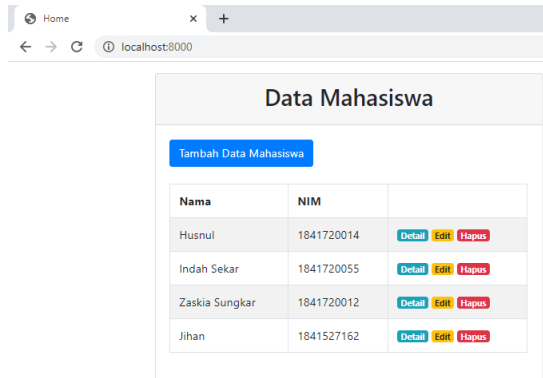
Nama	NIM	
Husnul	1841720014	Detail Edit
Indah Sekar	1841720055	Detail Edit
Zaskia Sungkar	1841720012	Detail Edit

f. Menghapus Data (Delete) dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk menghapus data (Line 30) pada file index.blade.php di folder resources/views</p> <pre> 9 <!-- isi bagian konten --> 10 @section('konten') 11 Tambah Data Mahasiswa 12
 13
 14 <table class="table table-bordered table-hover table-striped"> 15 <thead> 16 <tr> 17 <th>Nama</th> 18 <th>NIM</th> 19 <th></th> 20 </tr> 21 </thead> 22 <tbody> 23 @foreach(\$mahasiswa as \$mhs) 24 <tr> 25 <td>{{ \$mhs->nama }}</td> 26 <td>{{ \$mhs->nim }}</td> 27 <td> 28 id }}" class="badge badge-info">Detail 29 id }}" class="badge badge-warning">Edit 30 id }}" class="badge badge-warning">Hapus 31 </td> 32 </tr> 33 @endforeach 34 </tbody> 35 </table> 36 @endsection </pre>
2	<p>Buatlah <i>route</i> 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>
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 \$mahasiswa = 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

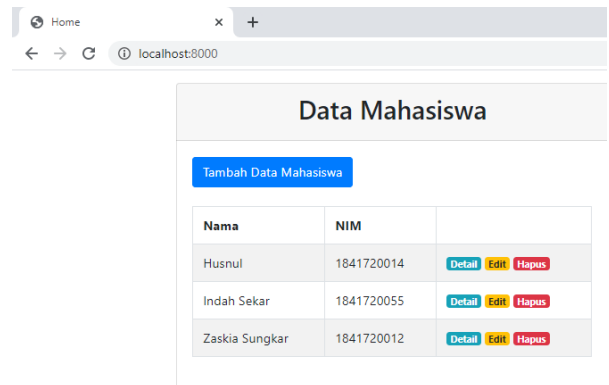
4

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



Home x +
localhost:8000

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
Husnul	1841720014	Detail Edit Hapus
Indah Sekar	1841720055	Detail Edit Hapus
Zaskia Sungkar	1841720012	Detail Edit Hapus
Jihan	1841527162	Detail Edit Hapus



Home x +
localhost:8000

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
Husnul	1841720014	Detail Edit Hapus
Indah Sekar	1841720055	Detail Edit Hapus
Zaskia Sungkar	1841720012	Detail Edit Hapus

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

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`



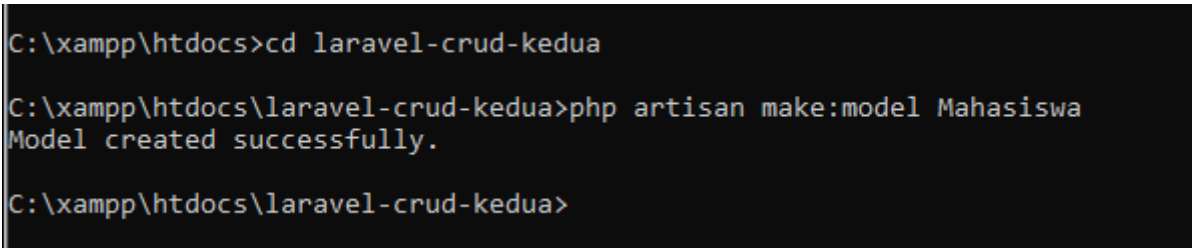
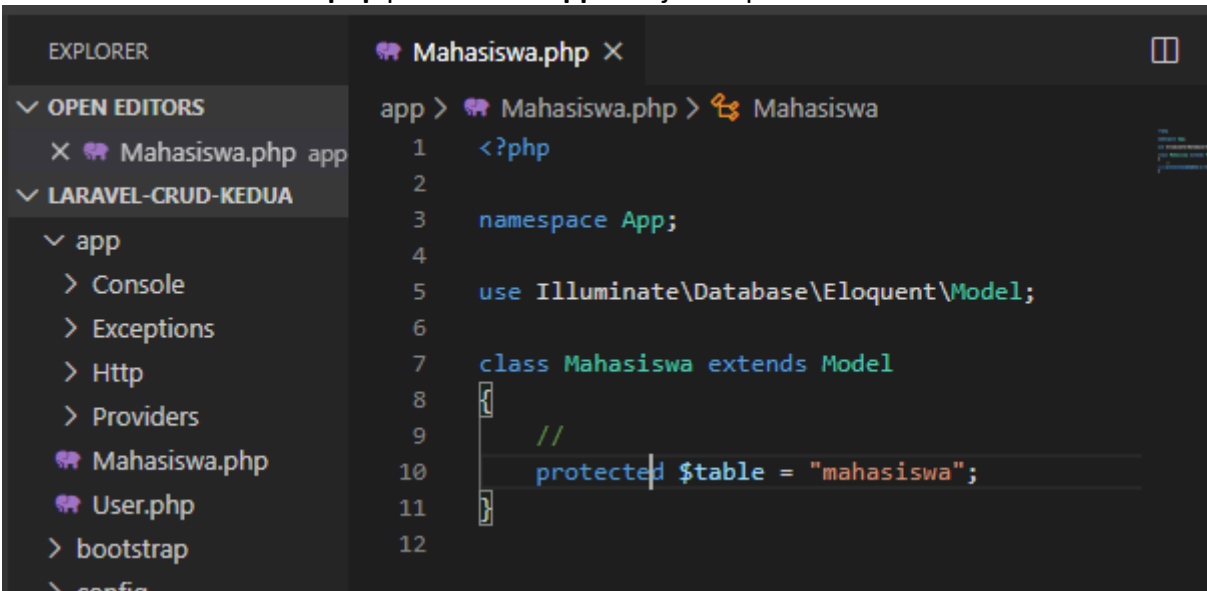
Server: 127.0.0.1 » Basis data: latihan_laravel » Tabel: mahasiswa

Struktur tabel

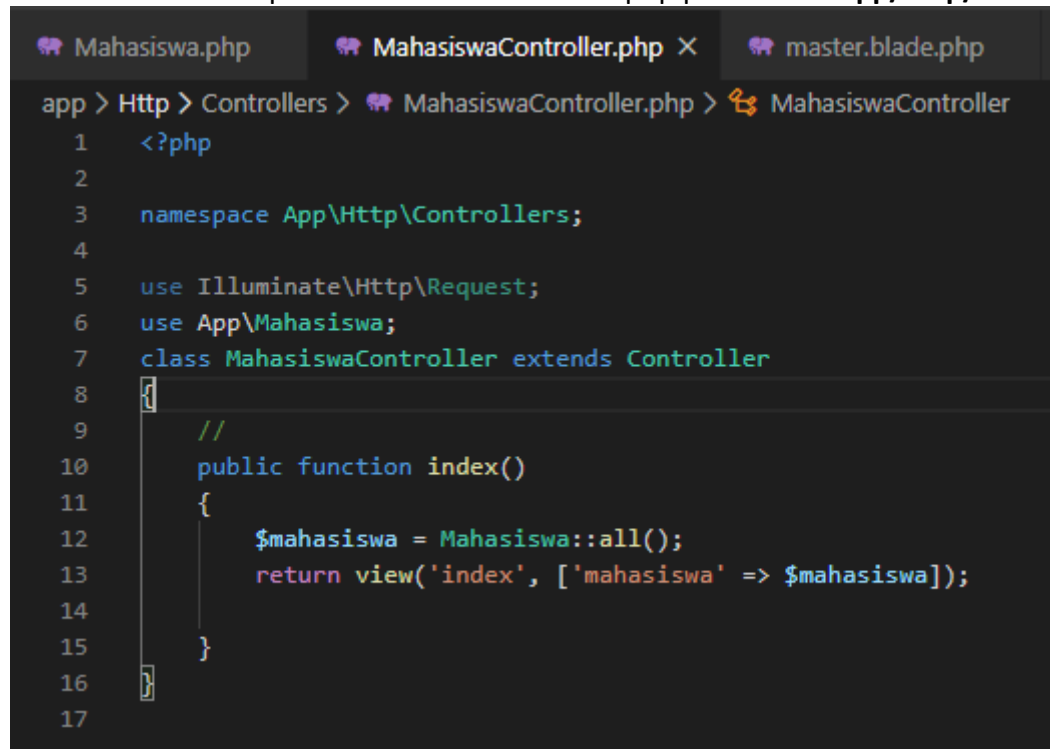
#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	id	int(11)		Tidak	Tidak ada	AUTO_INCREMENT		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
2	nama	varchar(100)		Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
3	nim	varchar(10)		Tidak	Tidak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

Check all Dengan pilihan: Jelajahi Ubah Hapus Kunci Utama Unik Indeks Add to central columns Remove from central columns

b. Menampilkan Data dari Database

Langkah	Keterangan
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan.</p> <p>Pertama, buatlah <i>route</i> pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p> <pre> 21 Route::get('/', 'MahasiswaController@index'); 22 Route::get('/mahasiswa', 'MahasiswaController@index');</pre>
2	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan</p> <pre> cd laravel-crud-kedua php artisan make:model Mahasiswa</pre>  <pre> C:\xampp\htdocs>cd laravel-crud-kedua C:\xampp\htdocs\laravel-crud-kedua>php artisan make:model Mahasiswa Model created successfully. C:\xampp\htdocs\laravel-crud-kedua></pre>
3	<p>Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.</p>  <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</p> <pre> php artisan make:controller MahasiswaController</pre>

5

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

```
app > Http > Controllers > MahasiswaController.php > MahasiswaController
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Mahasiswa;
7  class MahasiswaController extends Controller
8  {
9      //
10     public function index()
11     {
12         $mahasiswa = Mahasiswa::all();
13         return view('index', ['mahasiswa' => $mahasiswa]);
14     }
15 }
16
17
```

Keterangan:

- 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

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).

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scal
6
7   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com
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             <h2> @yield('judul_halaman')</h2>
17           </div>
18           <div class="card-body">
19             @yield('konten')
20           </div>
21           <div class="card">
22             </div>
23           </div>
24         </div>
25       </div>
26     </div>
27 </body>
28 </html>

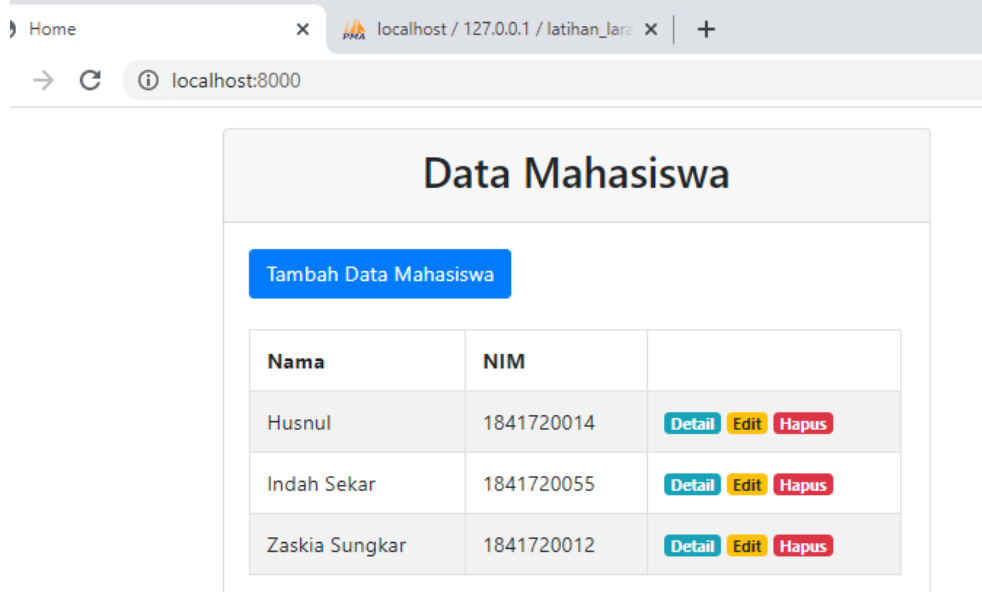
```

Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)

7

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

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



c. Memasukkan Data (Create) ke Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</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>public function tambah() { return view('tambah'); }</pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views.</p> <p>Kita lakukan copy paste dari tambah.blade.php di Bagian 1</p>  <pre> 1 @extends('master') 2 3 <!-- isi title --> 4 @section('title', 'Tambah Data') 5 6 <!-- isi bagian judul halaman --> 7 @section('judul_halaman', 'Tambah Data Mahasiswa') 8 9 <!-- isi bagian konten --> 10 @section('konten') 11 Kembali 12
 13
 14 <form action="/mahasiswa/simpan" method="post"> 15 {{ csrf_field() }} 16 <div class="form-group"> 17 <label for="namamhs">Nama</label> 18 <input type="text" class="form-control" required="required" name="namamhs"> 19 </div> 20 <div class="form-group"> 21 <label for="nimhs">NIM</label> 22 <input type="text" class="form-control" required="required" name="nimhs"> 23 </div> 24 <div class="form-group"> 25 <label for="emailmhs">E-mail</label> 26 <input type="text" class="form-control" required="required" name="emailmhs"> 27 </div> 28 <div class="form-group"> 29 <label for="jurusanmhs">Jurusan</label> 30 <input type="text" class="form-control" required="required" name="jurusanmhs"> 31 </div> 32 <button type="submit" name="tambah" class="btn btn-primary float-right"> Tambah Data </button> 33 </form> 34 @endsection </pre>

4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>Route::get('/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 simpan di MahasiswaController.php
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre>public function simpan(Request \$request) { // insert data ke tabel mahasiswa Mahasiswa::create(['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 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> Mahasiswa.php 1 <?php 2 3 namespace App; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class Mahasiswa extends Model 8 { 9 // 10 protected \$table = "mahasiswa"; 11 protected \$fillable = ['nama', 'nim', 'email', 'jurusan']; 12 } 13 </pre>

7

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.

Tambah Data Mahasiswa

[Kembali](#)

Nama
Muhammad Amar

NIM
1841720013

E-mail
amar@gmail.com

Jurusan
Teknik Mesin

[Tambah Data](#)

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

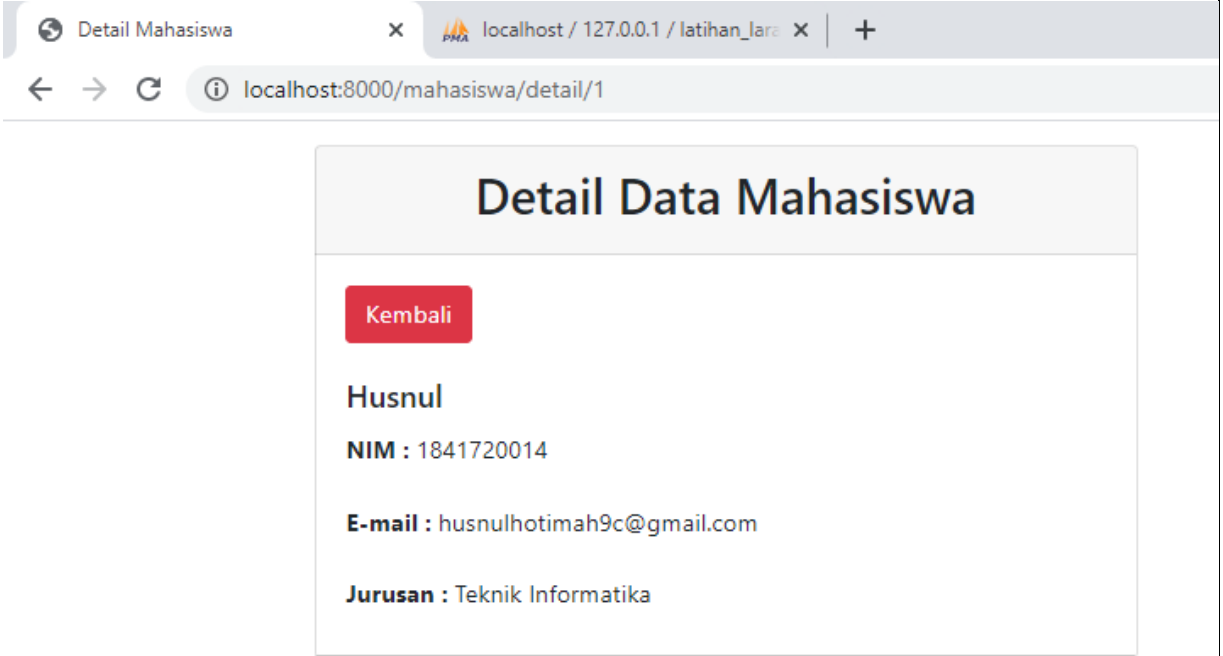
Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM	
Husnul	1841720014	Detail Edit Hapus
Indah Sekar	1841720055	Detail Edit Hapus
Zaskia Sungkar	1841720012	Detail Edit Hapus
Muhammad Amar	1841720013	Detail Edit Hapus

d. Melihat Detail Data dari Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre>Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');</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>public function detail(\$id) { // insert data ke tabel mahasiswa \$mahasiswa = Mahasiswa::find(\$id); return view('detail', ['mahasiswa' => \$mahasiswa]); }</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> 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 Kembali 12
 13
 14 <h5 class="card-title"> {{ \$mahasiswa->nama }} </h5> 15 <p class="card-text"> 16 <label for="">NIM : </label> 17 {{ \$mahasiswa->nim }} 18 </p> 19 <p class="card-text"> 20 <label for="">E-mail : </label> 21 {{ \$mahasiswa->email }} 22 </p> 23 <p class="card-text"> 24 <label for="">Jurusan : </label> 25 {{ \$mahasiswa->jurusan }} 26 </p> 27 @endsection </pre>

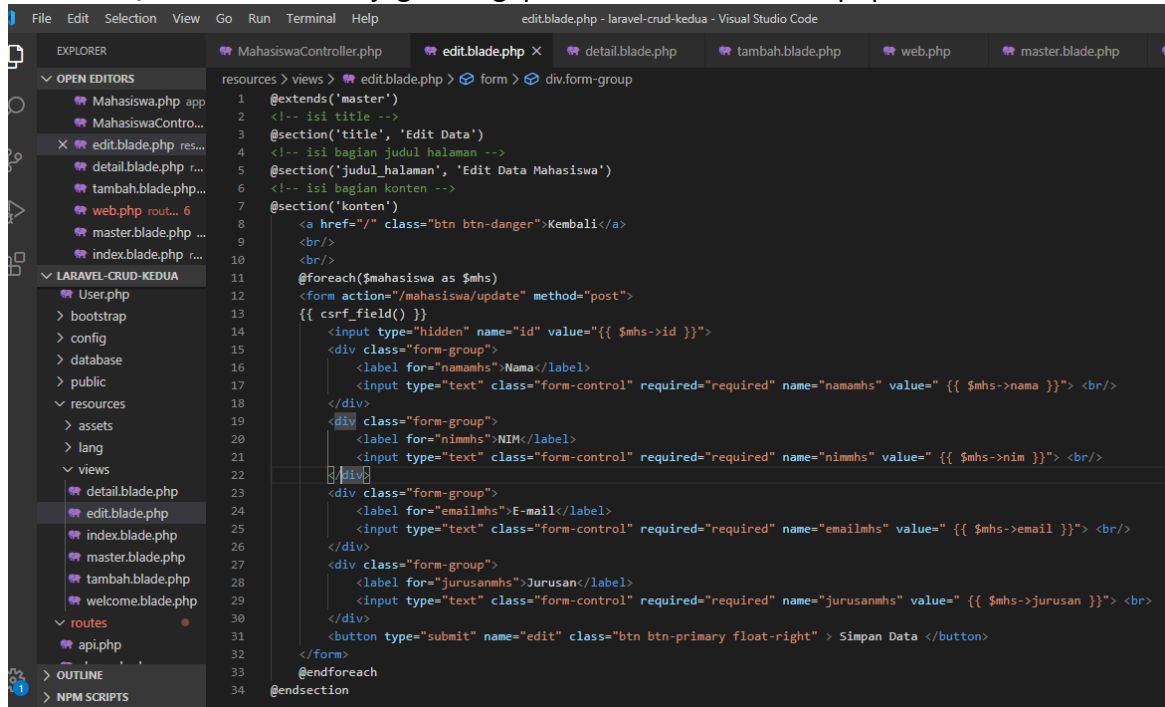
4	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p> 
---	--

e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada 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) { // insert data ke tabel mahasiswa \$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

Kemudian buatlah view **edit.blade.php** yang berisi form untuk mengubah data pada folder **resources/views**. View edit juga mengaplikasikan master.blade.php.



```

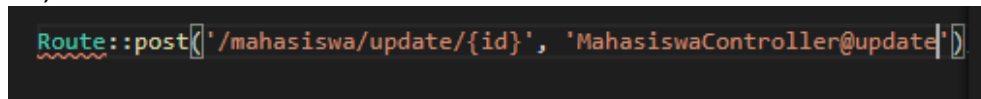
1 @extends('master')
2 <!-- isi title -->
3 @section('title', 'Edit Data')
4 <!-- isi bagian judul halaman -->
5 @section('judul_halaman', 'Edit Data Mahasiswa')
6 <!-- isi bagian konten -->
7 @section('konten')
8 <a href="/" class="btn btn-danger">Kembali</a>
9 <br/>
10 <br/>
11 @foreach($mahasiswa 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 }}"> <br/>
18 </div>
19 <div class="form-group">
20 <label for="nimhs">NIM</label>
21 <input type="text" class="form-control" required="required" name="nimhs" value="{{ $mhs->nim }}"> <br/>
22 </div>
23 <div class="form-group">
24 <label for="emailhs">E-mail</label>
25 <input type="text" class="form-control" required="required" name="emailhs" value="{{ $mhs->email }}"> <br/>
26 </div>
27 <div class="form-group">
28 <label for="jurusanhs">Jurusan</label>
29 <input type="text" class="form-control" required="required" name="jurusanhs" value="{{ $mhs->jurusan }}"> <br/>
30 </div>
31 <button type="submit" name="edit" class="btn btn-primary float-right"> Simpan Data </button>
32 </form>
33 @endforeach
34 @endsection
  
```

Keterangan:

- 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

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



```

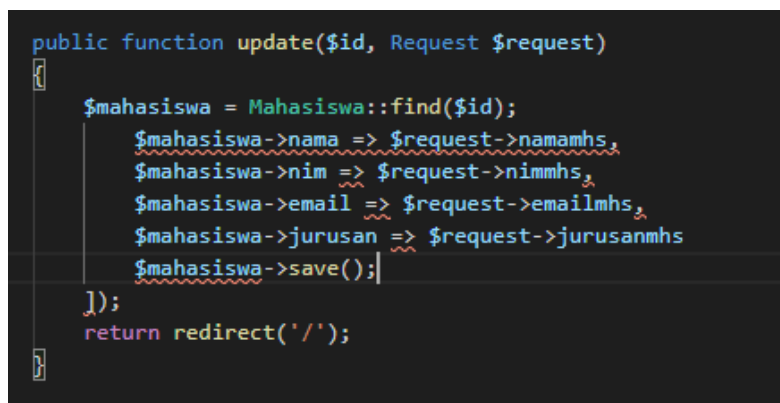
Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update');
  
```

Keterangan:

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

5

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



```

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('/');
  
```

Keterangan:

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

6

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

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

Top Screenshot: Edit Data Mahasiswa

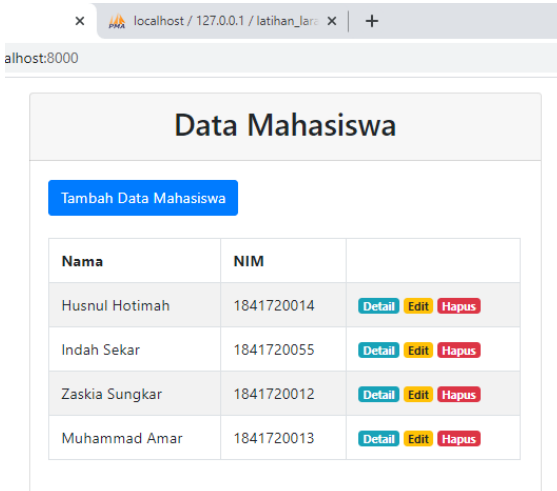
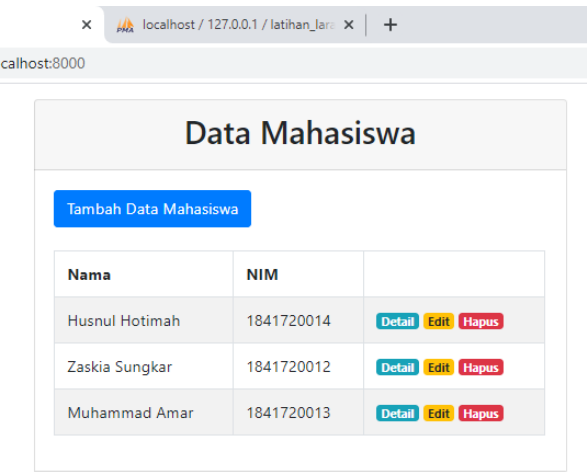
The browser address bar shows `localhost:8000/mahasiswa/edit/1`. The page title is "Edit Data Mahasiswa". It features a red "Kembali" button at the top left. Below it, there are four input fields with labels: "Nama" (containing "Husnul Hotimah"), "NIM" (containing "1841720014"), "E-mail" (containing "husnulhotimah9c@gmail.com"), and "Jurusan" (containing "Teknik Informatika"). A blue "Simpan Data" button is at the bottom right.

Bottom Screenshot: Data Mahasiswa

The browser address bar shows `localhost:8000`. The page title is "Data Mahasiswa". It features a blue "Tambah Data Mahasiswa" button at the top left. Below it is a table with the following data:

Nama	NIM	
Husnul Hotimah	1841720014	Detail Edit Hapus
Indah Sekar	1841720055	Detail Edit Hapus
Zaskia Sungkar	1841720012	Detail Edit Hapus
Muhammad Amar	1841720013	Detail Edit Hapus

f. Menghapus Data (Delete) dari Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre>Route::post('/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) { // insert data ke tabel mahasiswa \$mahasiswa = Mahasiswa::find(\$id); \$mahasiswa->delete(); return redirect('/'); }</pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih
3	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.</p> <div>   </div>