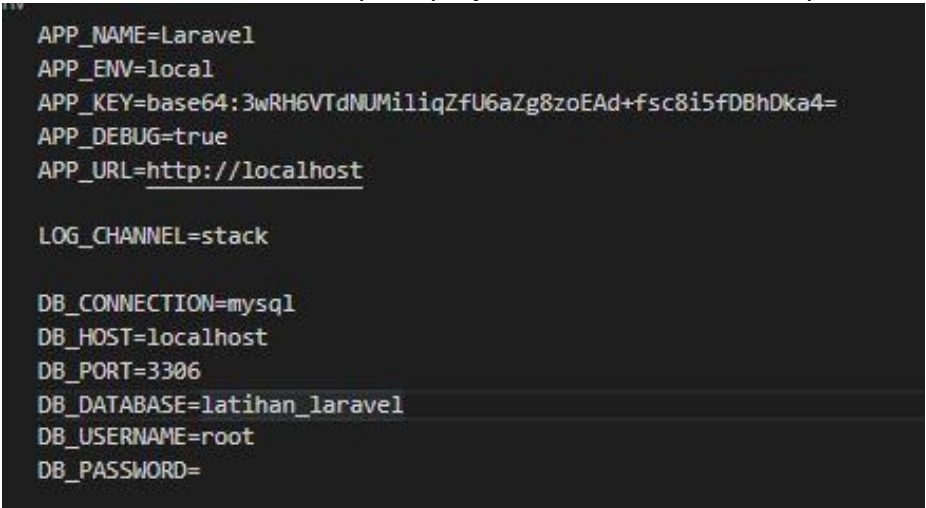


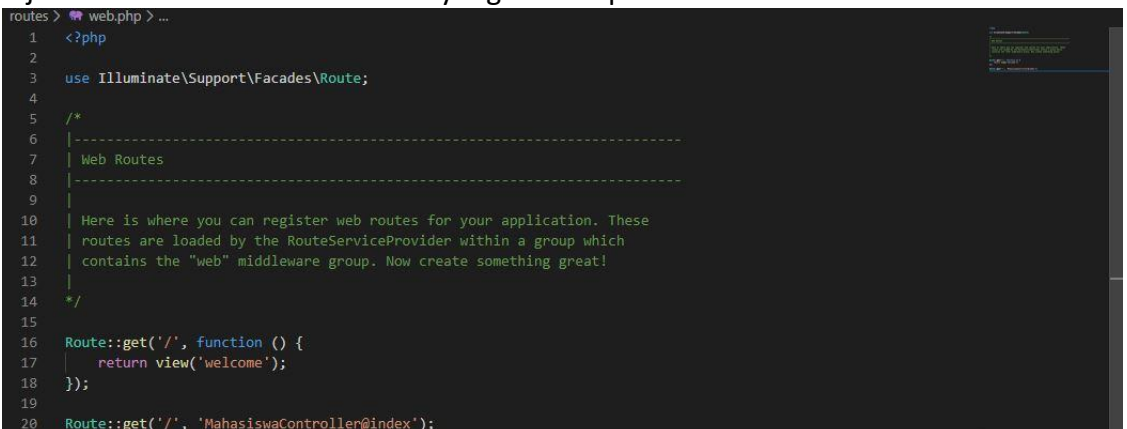
## Praktikum – Bagian 1: Membuat CRUD di Laravel menggunakan Query Builder

### a. Konfigurasi Database dan Pembuatan Tabel di MySQL

Langkah	Keterangan
1	Buatlah project Laravel baru dengan nama <b>laravel-crud</b> . Buka command prompt, tuliskan perintah berikut. <b>cd C:\xampp\htdocs</b> <b>laravel new laravel-crud</b>
2	Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file <b>.env</b> pada project laravel-crud. Ubah seperti di bawah ini.  <pre>APP_NAME=Laravel APP_ENV=local APP_KEY=base64:3wRH6VTdNUMiliqZfU6aZg8zoEAd+fsc8i5fDBhDka4= APP_DEBUG=true APP_URL=http://localhost  LOG_CHANNEL=stack  DB_CONNECTION=mysql DB_HOST=localhost DB_PORT=3306 DB_DATABASE=latihan_laravel DB_USERNAME=root DB_PASSWORD=</pre> <p>Keterangan:</p>

	- Nama database yang akan digunakan adalah latihan_laravel dengan username root.																																																												
3	<p>Jalankan xampp, selanjutnya buat tabel dengan nama <b>mahasiswa</b> di mysql pada database latihan_laravel.</p> <table><tr><th></th><th>#</th><th>Name</th><th>Type</th><th>Collation</th><th>Attributes</th><th>Null</th><th>Default</th><th>Comments</th><th>Extra</th></tr><tr><td><input type="checkbox"/></td><td>1</td><td>id</td><td>int(11)</td><td></td><td></td><td>No</td><td>None</td><td></td><td>AUTO_INCREMENT</td></tr><tr><td><input type="checkbox"/></td><td>2</td><td>nama</td><td>varchar(100)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr><tr><td><input type="checkbox"/></td><td>3</td><td>nim</td><td>varchar(10)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr><tr><td><input type="checkbox"/></td><td>4</td><td>email</td><td>varchar(50)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr><tr><td><input type="checkbox"/></td><td>5</td><td>jurusan</td><td>varchar(20)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr></table>		#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	<input type="checkbox"/>	1	id	int(11)			No	None		AUTO_INCREMENT	<input type="checkbox"/>	2	nama	varchar(100)	utf8mb4_general_ci		No	None			<input type="checkbox"/>	3	nim	varchar(10)	utf8mb4_general_ci		No	None			<input type="checkbox"/>	4	email	varchar(50)	utf8mb4_general_ci		No	None			<input type="checkbox"/>	5	jurusan	varchar(20)	utf8mb4_general_ci		No	None		
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra																																																				
<input type="checkbox"/>	1	id	int(11)			No	None		AUTO_INCREMENT																																																				
<input type="checkbox"/>	2	nama	varchar(100)	utf8mb4_general_ci		No	None																																																						
<input type="checkbox"/>	3	nim	varchar(10)	utf8mb4_general_ci		No	None																																																						
<input type="checkbox"/>	4	email	varchar(50)	utf8mb4_general_ci		No	None																																																						
<input type="checkbox"/>	5	jurusan	varchar(20)	utf8mb4_general_ci		No	None																																																						
4	<p>Isilah beberapa data pada tabel <b>mahasiswa</b> tersebut.</p> <div>+ Options</div> <table><tr><td><input type="checkbox"/></td><td></td><td>Edit</td><td></td><td>Copy</td><td></td><td>Delete</td><td>1</td><td>nanda</td><td>1001001001</td><td>nandan@gmail.com</td><td>teknik informatika</td></tr><tr><td><input type="checkbox"/></td><td></td><td>Edit</td><td></td><td>Copy</td><td></td><td>Delete</td><td>2</td><td>wahyu</td><td>1001001002</td><td>wahyu@gmail.com</td><td>teknik elektro</td></tr></table>	<input type="checkbox"/>		Edit		Copy		Delete	1	nanda	1001001001	nandan@gmail.com	teknik informatika	<input type="checkbox"/>		Edit		Copy		Delete	2	wahyu	1001001002	wahyu@gmail.com	teknik elektro																																				
<input type="checkbox"/>		Edit		Copy		Delete	1	nanda	1001001001	nandan@gmail.com	teknik informatika																																																		
<input type="checkbox"/>		Edit		Copy		Delete	2	wahyu	1001001002	wahyu@gmail.com	teknik elektro																																																		

## b. Menampilkan Data dari Database

Langkah	Keterangan
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan.</p> <p>Pertama, buatlah <i>route</i> pada <b>routes/web.php</b> sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p>  <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Ketika route ('/') diakses, akan dijalankan method <b>index</b> pada controller bernama <b>MahasiswaController</b>.</li> </ul>
2	<p>Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan</p> <pre>cd laravel-crud php artisan make:controller MahasiswaController</pre>

3

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

```

1 namespace App\Http\Controllers;
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\DB;
5
6 class MahasiswaController extends Controller
7 {
8     public function Index()
9     {
10         $mahasiswa = DB::table('mahasiswa')->get();
11         return view('index', ['mahasiswa' => $mahasiswa]);
12     }
13 }

```

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**.

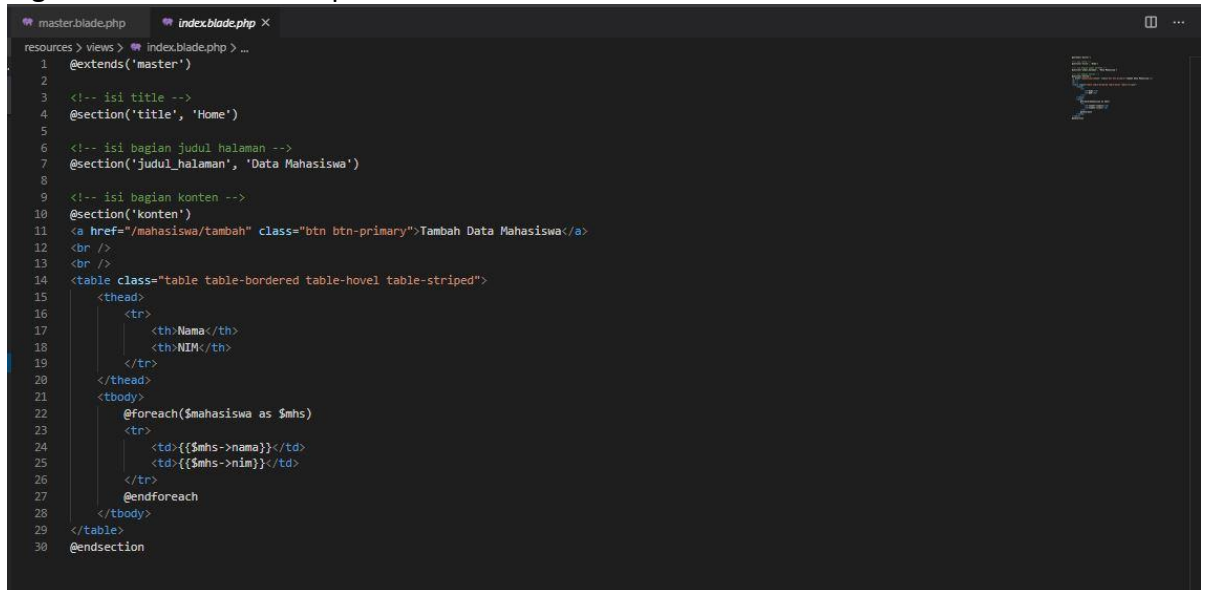
```

1 <!doctype html>
2 <html lang="en">
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet">
8     <title>@yield('title')</title>
9 </head>
10
11 <body>
12     <div class="container">
13         <div class="row mt-3">
14             <div class="col-md-6">
15                 <div class="card">
16                     <div class="card-header text-center">
17                         <!-- bagian judul halaman -->
18                         <h2>@yield('judul_halaman')</h2>
19                     </div>
20                     <div class="card-body">
21                         <!-- bagian konten blog -->
22                         @yield('konten')
23                     </div>
24                     <div class="card">
25                     </div>
26                 </div>
27             </div>
28         </div>
29     </div>
30 </body>
31 </html>

```

5

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

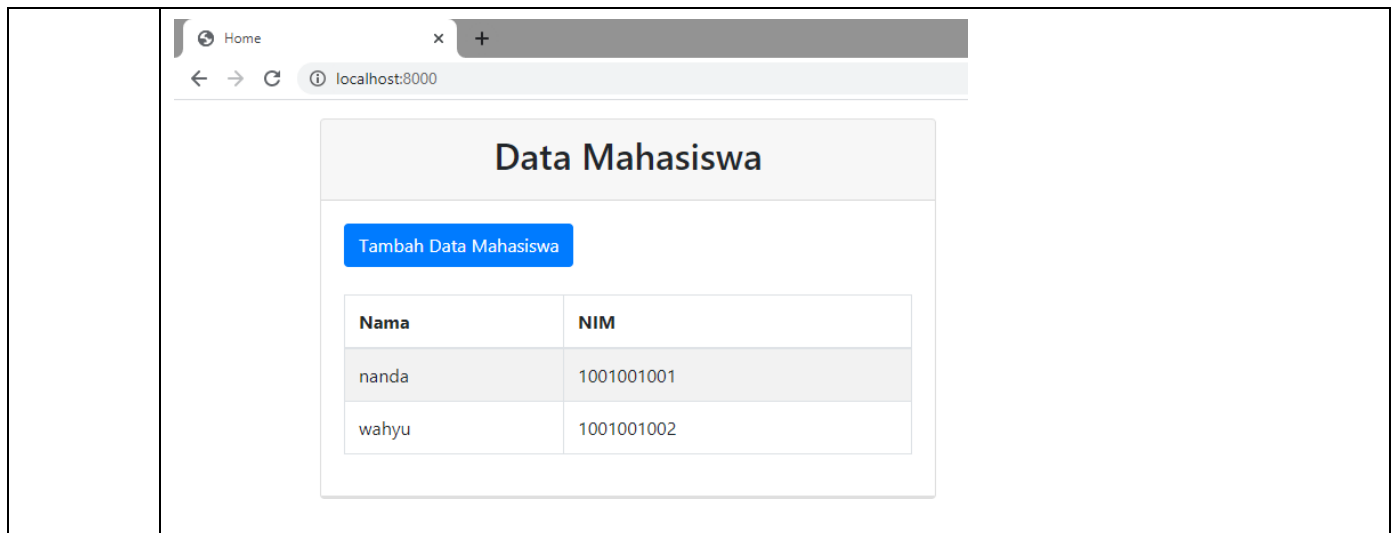


```
resources > views > index.blade.php > ...
1  @extends('master')
2
3  <!-- isi title -->
4  @section('title', 'Home')
5
6  <!-- isi bagian judul halaman -->
7  @section('judul_halaman', 'Data Mahasiswa')
8
9  <!-- isi bagian konten -->
10 @section('konten')
11 <a href="/mahasiswa/tambah" class="btn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

6

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

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 <b>routes/web.php</b> dengan nama <b>/mahasiswa/tambah</b> yang akan menjalankan fungsi tambah pada MahasiswaController</p> <pre>Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');</pre>
2	<p>Buat method <b>tambah</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view tambah.</p> <pre>public function tambah(){     return view('tambah'); }</pre>
3	<p>Kemudian buatlah view <b>tambah.blade.php</b> yang berisi form untuk memasukkan data baru pada folder <b>resources/views</b>. View tambah juga mengaplikasikan master.blade.php.</p>

```

tambah.blade.php X
resources > views > tambah.blade.php
1 @extends('master')
2 @section('title', 'Tambah Data')
3 @section('judul_halaman', 'Tambah Data Mahasiswa')
4 @section('konten')
5     <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
6     <br><br>
7     <form action="/laravel-crud/public/mahasiswa/simpan" method="post">
8         {{ csrf_field() }}
9         <div class="form-group">
10             <label for="namamhs">Nama</label>
11             <input type="text" name="namamhs" class="form-control" required>
12         </div>
13         <div class="form-group">
14             <label for="nimhs">NIM</label>
15             <input type="number" name="nimhs" class="form-control" required>
16         </div>
17         <div class="form-group">
18             <label for="emailmhs">E-mail</label>
19             <input type="text" name="emailmhs" class="form-control" required>
20         </div>
21         <div class="form-group">
22             <label for="jurusanmhs">Jurusan</label>
23             <input type="text" name="jurusanmhs" class="form-control" required>
24         </div>
25         <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button>
26     </form>
27 @endsection
28

```

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.

```

23
24 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.

```

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

```

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:

- Nama:** marsha
- NIM:** 1001001003
- E-mail:** marsha@gmail.com
- Jurusan:** teknik sipil

Buttons: 'Kembali' (red) and 'Tambah Data' (blue).

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

Home

localhost:8000/mahasiswa

## Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM
nanda	1001001001
wahyu	1001001002
marsha	1001001003

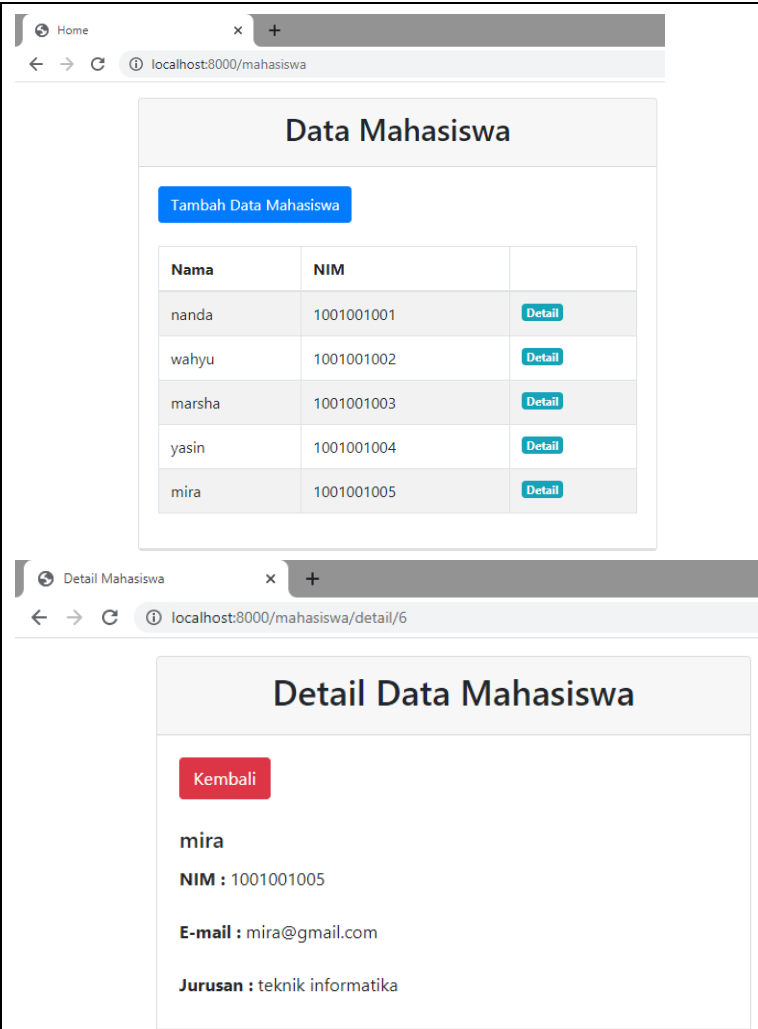
Kita dapat mencoba menambahkan beberapa data lagi agar jumlah data lebih banyak.

d. Melihat Detail Data dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk melihat detail data (Line 28) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre> resources &gt; views &gt; index.blade.php 1  @extends('master') 2  @section('title', 'Home') 3  @section('judul_halaman', 'Data Mahasiswa') 4 5  @section('konten') 6      &lt;a href="/mahasiswa/tambah" class="btn btn-primary"&gt;Tambah Data Mahasiswa&lt;/a&gt; 7      &lt;br&gt;&lt;br&gt; 8      &lt;table class="table table-striped table-bordered table-hover"&gt; 9          &lt;thead class="thead-inverse"&gt; 10             &lt;tr&gt; 11                 &lt;th&gt;Nama&lt;/th&gt; 12                 &lt;th&gt;NIM&lt;/th&gt; 13             &lt;/tr&gt; 14         &lt;/thead&gt; 15         &lt;tbody&gt; 16             @foreach(\$mahasiswa as \$mhs) 17                 &lt;tr&gt; 18                     &lt;td scope="row"&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 19                     &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 20                     &lt;td&gt; 21                         &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 22                         &lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-warning"&gt;Edit&lt;/a&gt; 23                         &lt;a href="/mahasiswa/hapus/{{ \$mhs-&gt;id }}" class="badge badge-danger"&gt;Hapus&lt;/a&gt; 24                     &lt;/td&gt; 25                 &lt;/tr&gt; 26             @endforeach 27         &lt;/tbody&gt; 28     &lt;/table&gt; 29 @endsection 30 31 </pre>
2	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/detail</b> yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre> Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); </pre>
3	<p>Buat method <b>detail</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan data mahasiswa pada view detail.blade.php.</p>



	<pre> public function detail(\$id){     \$mahasiswa = DB::table('mahasiswa')-&gt;where('id', \$id)-&gt;get();     return view('detail', ['mahasiswa' =&gt; \$mahasiswa]); } </pre>
4	<p>Kemudian buatlah view <b>detail.blade.php</b> yang menampilkan detail data mahasiswa pada folder <b>resources/views</b>. View detail juga mengaplikasikan master.blade.php.</p> <pre> tambah.blade.php  detail.blade.php × resources &gt; views &gt; detail.blade.php 1  @extends('master') 2  @section('title', 'Detail Mahasiswa') 3  @section('judul_halaman', 'Detail Data Mahasiswa') 4 5  @section('konten') 6      &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 7      &lt;br&gt;&lt;br&gt; 8      @foreach(\$mahasiswa as \$mhs) 9          &lt;h5 class="card-title"&gt;{{ \$mhs-&gt;nama }}&lt;/h5&gt; 10         &lt;p class="card-text"&gt; 11             &lt;label for=""&gt;&lt;b&gt;NIM:&lt;/b&gt;&lt;/label&gt; 12             {{ \$mhs-&gt;nim }} 13         &lt;/p&gt; 14         &lt;p class="card-text"&gt; 15             &lt;label for=""&gt;&lt;b&gt;EMAIL&lt;/b&gt;&lt;/label&gt; 16             {{ \$mhs-&gt;email }} 17         &lt;/p&gt; 18         &lt;p class="card-text"&gt; 19             &lt;label for=""&gt;&lt;b&gt;JURUSAN&lt;/b&gt;&lt;/label&gt; 20             {{ \$mhs-&gt;jurusan }} 21         &lt;/p&gt; 22     @endforeach 23 @endsection 24 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 16-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan</li> </ul>
5	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p>

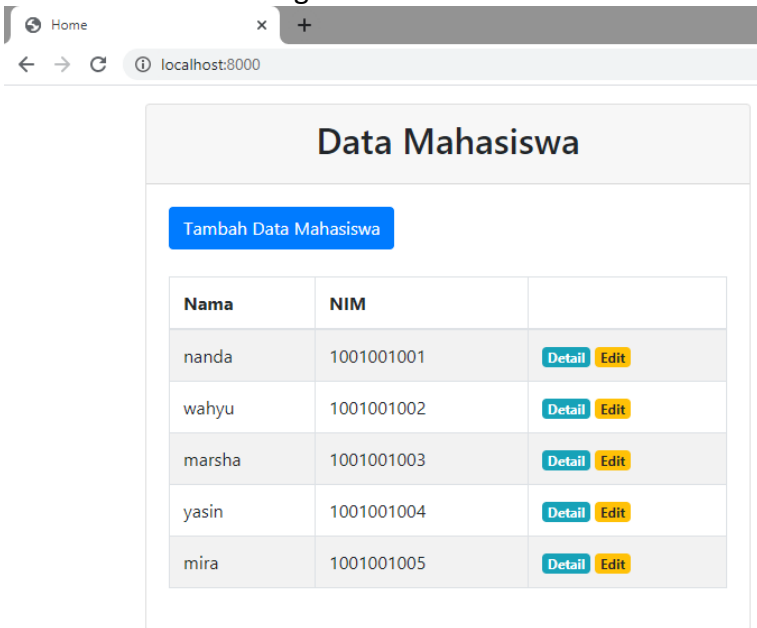


#### e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre> 17         @foreach(\$mahasiswa as \$mhs) 18             &lt;tr&gt; 19                 &lt;td scope="row"&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 20                 &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 21                 &lt;td&gt; 22                     &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="btn btn-info"&gt;Detail&lt;/a&gt; 23                     &lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="btn btn-warning"&gt;Edit&lt;/a&gt; 24                     &lt;a href="/mahasiswa/hapus/{{ \$mhs-&gt;id }}" class="btn btn-danger"&gt;Hapus&lt;/a&gt; 25                 &lt;/td&gt; 26             &lt;/tr&gt; 27         @endforeach 28     &lt;/tbody&gt; 29 &lt;/table&gt; 30 @endsection </pre>
2	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/edit</b> yang akan menjalankan fungsi edit pada MahasiswaController</p>

	<pre> 27 28 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); </pre>
3	<p>Buat method <b>edit</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view edit.</p> <pre> public function edit(\$id){     \$mahasiswa = DB::table('mahasiswa')-&gt;where('id', \$id)-&gt;get();     return view('edit', ['mahasiswa' =&gt; \$mahasiswa]); } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 49 : query builder untuk mengambil data mahasiswa berdasarkan id yang dipilih</li> </ul>
4	<p>Kemudian buatlah view <b>edit.blade.php</b> yang berisi form untuk mengubah data pada folder <b>resources/views</b>. View edit juga mengaplikasikan master.blade.php.</p> <pre> tambah.blade.php  edit.blade.php X resources &gt; views &gt; edit.blade.php 1  @extends('master') 2  @section('title', 'Edit Data') 3  @section('judul_halaman', 'Edit Data Mahasiswa') 4 5  @section('konten') 6      &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 7      &lt;br&gt;&lt;br&gt; 8      @foreach(\$mahasiswa as \$mhs) 9          &lt;form action="/laravel-crud/public/mahasiswa/update" method="post"&gt; 10             {{ csrf_field() }} 11             &lt;input type="hidden" name="id" value="{{ \$mhs-&gt;id }}"&gt;&lt;br&gt; 12             &lt;div class="form-group"&gt; 13                 &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 14                 &lt;input type="text" name="namamhs" class="form-control" value="{{ \$mhs-&gt;nama }}" rec 15             &lt;/div&gt; 16             &lt;div class="form-group"&gt; 17                 &lt;label for="nimmmhs"&gt;NIM&lt;/label&gt; 18                 &lt;input type="number" name="nimmmhs" class="form-control" value="{{ \$mhs-&gt;nim }}" rec 19             &lt;/div&gt; 20             &lt;div class="form-group"&gt; 21                 &lt;label for="emailmhs"&gt;E-mail&lt;/label&gt; 22                 &lt;input type="text" name="emailmhs" class="form-control" value="{{ \$mhs-&gt;email }}" r 23             &lt;/div&gt; 24             &lt;div class="form-group"&gt; 25                 &lt;label for="jurusanmhs"&gt;Jurusan&lt;/label&gt; 26                 &lt;input type="text" name="jurusanmhs" class="form-control" value="{{ \$mhs-&gt;jurusan }}" 27             &lt;/div&gt; 28             &lt;button type="submit" name="tambah" class="btn btn-primary float-right"&gt;Simpan Data&lt;/ 29         &lt;/form&gt; 30     @endforeach 31 @endsection 32 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 14-36 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan</li> <li>- Line 15 terdapat action="/mahasiswa/update" yang menunjukkan routes /mahasiswa/update dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController</li> </ul>

5	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/update</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>Route::post('/mahasiswa/update', 'MahasiswaController@update');</pre> <p>Keterangan:</p>
---	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<ul style="list-style-type: none"> <li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php</li> </ul>
6	<p>Buat method <b>update</b> pada <b>MahasiswaController</b> untuk menyimpan data yang diubah ke database.</p> <pre> public function update(Request \$request){     DB::table('mahasiswa')-&gt;where('id', \$request-&gt;id)-&gt;update([         'nama' =&gt; \$request-&gt;namamhs,         'nim' =&gt; \$request-&gt;nimmhs,         'email' =&gt; \$request-&gt;emailmhs,         'jurusan' =&gt; \$request-&gt;jurusanmhs     ]);     return redirect('/'); } </pre>
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	<a href="#">Detail</a> <a href="#">Edit</a>
wahyu	1001001002	<a href="#">Detail</a> <a href="#">Edit</a>
marsha	1001001003	<a href="#">Detail</a> <a href="#">Edit</a>
yasin	1001001004	<a href="#">Detail</a> <a href="#">Edit</a>
mira	1001001005	<a href="#">Detail</a> <a href="#">Edit</a>

#### f. Menghapus Data (Delete) dari Database

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

```

<td scope="row">{{ $mhs->nama }}</td>
<td>{{ $mhs->nim }}</td>
<td>
    <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a>
    <a href="/mahasiswa/edit/{{ $mhs->id }}" class="badge badge-warning">Edit</a>
    <a href="/mahasiswa/hapus/{{ $mhs->id }}" class="badge badge-danger">Hapus</a>
</td>
</tr>
@endforeach
</tbody>

```

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('/');
}

```

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.

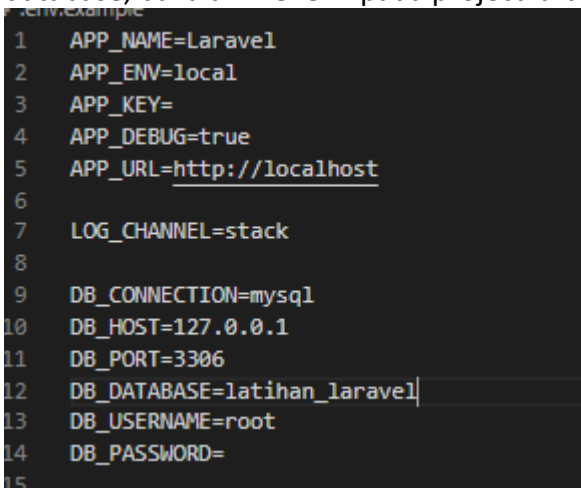



Tambah Data Mahasiswa		
Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
yasin	1001001004	Detail Edit Hapus
mira	1001001005	Detail Edit Hapus

Tambah Data Mahasiswa		
Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
yasin	1001001004	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	Buatlah project Laravel baru dengan nama <b>laravel-crud-kedua</b> . Buka command prompt, tuliskan perintah berikut. <b>cd C:\xampp\htdocs</b> <b>laravel new laravel-crud-kedua</b>																																																																								
2	Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file <b>.env</b> 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 <table><tr><th>#</th><th>Name</th><th>Type</th><th>Collation</th><th>Attributes</th><th>Null</th><th>Default</th><th>Comments</th><th>Extra</th></tr><tr><td><input type="checkbox"/> 1</td><td>id </td><td>int(11)</td><td></td><td></td><td>No</td><td>None</td><td></td><td>AUTO_INCREMENT</td></tr><tr><td><input type="checkbox"/> 2</td><td>nama</td><td>varchar(100)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr><tr><td><input type="checkbox"/> 3</td><td>nim</td><td>varchar(10)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr><tr><td><input type="checkbox"/> 4</td><td>email</td><td>varchar(50)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr><tr><td><input type="checkbox"/> 5</td><td>jurusan</td><td>varchar(20)</td><td>utf8mb4_general_ci</td><td></td><td>No</td><td>None</td><td></td><td></td></tr><tr><td><input type="checkbox"/> 6</td><td>created_at</td><td>timestamp</td><td></td><td></td><td>Yes</td><td>NULL</td><td></td><td></td></tr><tr><td><input type="checkbox"/> 7</td><td>updated_at</td><td>timestamp</td><td></td><td></td><td>Yes</td><td>NULL</td><td></td><td></td></tr></table>	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	<input type="checkbox"/> 1	id 	int(11)			No	None		AUTO_INCREMENT	<input type="checkbox"/> 2	nama	varchar(100)	utf8mb4_general_ci		No	None			<input type="checkbox"/> 3	nim	varchar(10)	utf8mb4_general_ci		No	None			<input type="checkbox"/> 4	email	varchar(50)	utf8mb4_general_ci		No	None			<input type="checkbox"/> 5	jurusan	varchar(20)	utf8mb4_general_ci		No	None			<input type="checkbox"/> 6	created_at	timestamp			Yes	NULL			<input type="checkbox"/> 7	updated_at	timestamp			Yes	NULL		
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra																																																																	
<input type="checkbox"/> 1	id 	int(11)			No	None		AUTO_INCREMENT																																																																	
<input type="checkbox"/> 2	nama	varchar(100)	utf8mb4_general_ci		No	None																																																																			
<input type="checkbox"/> 3	nim	varchar(10)	utf8mb4_general_ci		No	None																																																																			
<input type="checkbox"/> 4	email	varchar(50)	utf8mb4_general_ci		No	None																																																																			
<input type="checkbox"/> 5	jurusan	varchar(20)	utf8mb4_general_ci		No	None																																																																			
<input type="checkbox"/> 6	created_at	timestamp			Yes	NULL																																																																			
<input type="checkbox"/> 7	updated_at	timestamp			Yes	NULL																																																																			



## b. Menampilkan Data dari Database

Langkah	Keterangan
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan.</p> <p>Pertama, buatlah <i>route</i> pada <b>routes/web.php</b> sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p> <pre>Route::get('/', 'MahasiswaController@index');  Route::get('/mahasiswa', 'MahasiswaController@index');</pre>
2	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan</p> <p><b>cd laravel-crud-kedua</b> <b>php artisan make:model Mahasiswa</b></p>
3	<p>Ubah model <b>Mahasiswa.php</b> pada folder <b>App</b> menjadi seperti berikut.</p> <pre>env.example web.php Mahasiswa.php X app &gt; Mahasiswa.php 1  &lt;?php 2 3  namespace App; 4 5  use Illuminate\Database\Eloquent\Model; 6 7  class Mahasiswa extends Model 8  { 9      protected \$table = "mahasiswa"; 10     protected \$fillable = ['nama', 'nim', 'email', 'jurusan']; 11 } 12</pre> <p>Keterangan:</p> <ul style="list-style-type: none"><li>- Model Mahasiswa akan menangani tabel mahasiswa</li></ul>
4	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan</p> <p><b>php artisan make:controller MahasiswaController</b></p>

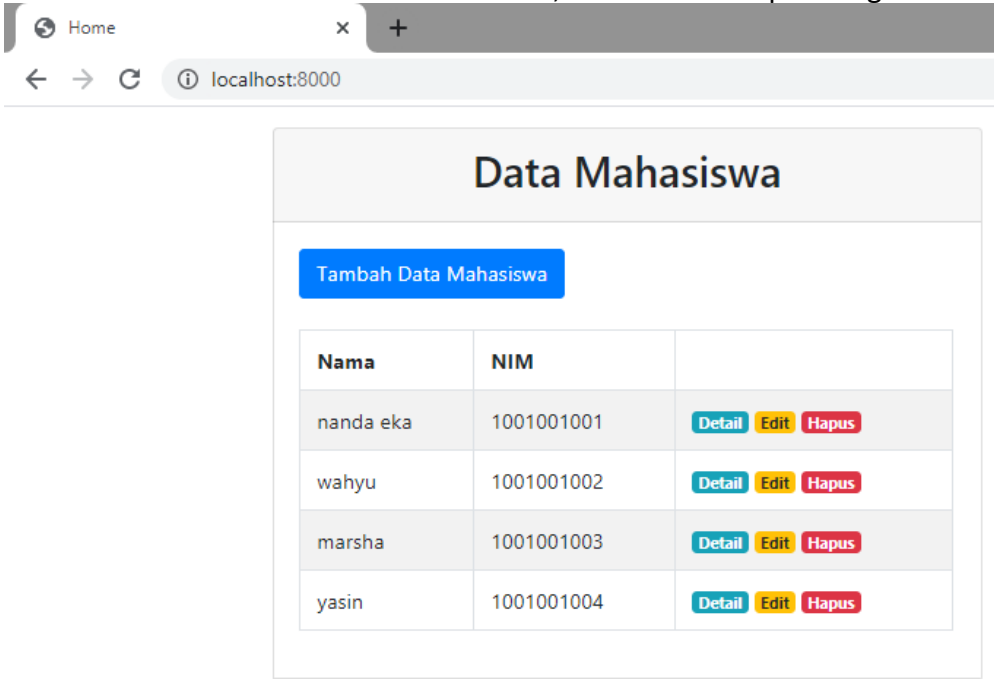
5

Buat method **index** pada MahasiswaController.php pada folder **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]);
    }
}
```

Keterangan:

	<ul style="list-style-type: none"> <li>- Tambahkan 'use App\Mahasiswa' (line 6) untuk menggunakan model Mahasiswa</li> <li>- Line 12 untuk mengambil semua data dari model/tabel Mahasiswa dan akan disimpan di variabel \$mahasiswa</li> <li>- Line 16 : data akan dikirim ke blade view bernama index</li> </ul>
6	<p>Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat <b>template blade</b> (seperti pada Bagian 1). Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)</p>
7	<p>Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud <b>php artisan serve</b></p> <p>Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut</p> 

### c. Memasukkan Data (Create) ke Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/tambah</b> 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 <b>tambah</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view tambah.</p> <pre>public function tambah(){     return view('tambah'); }</pre>
3	<p>Kemudian buatlah view <b>tambah.blade.php</b> yang berisi form untuk memasukkan data baru pada folder <b>resources/views</b>.</p>

	Kita lakukan copy paste dari tambah.blade.php di Bagian 1
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/simpan</b>. 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"> <li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php</li> </ul>
5	<p>Buat method <b>simpan</b> pada <b>MahasiswaController</b> untuk menyimpan data ke database.</p> <pre>public function simpan(Request \$request){     Mahasiswa::create([         'nama' =&gt; \$request-&gt;namamhs,         'nim' =&gt; \$request-&gt;nimmhs,         'email' =&gt; \$request-&gt;emailmhs,         'jurusan' =&gt; \$request-&gt;jurusanmhs     ]);     return redirect('/'); }</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li> <li>- Line 23-28 merupakan fitur eloquent menggunakan fungsi create() untuk insert data ke tabel mahasiswa</li> </ul>
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>
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

localhost:8000/mahasiswa/tambah

### Tambah Data Mahasiswa

[Kembali](#)

Nama  
rahmat

NIM  
1001001006

E-mail  
rahmat@gmail.com

Jurusan  
teknik mesin

[Tambah Data](#)

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

Home

localhost:8000/mahasiswa

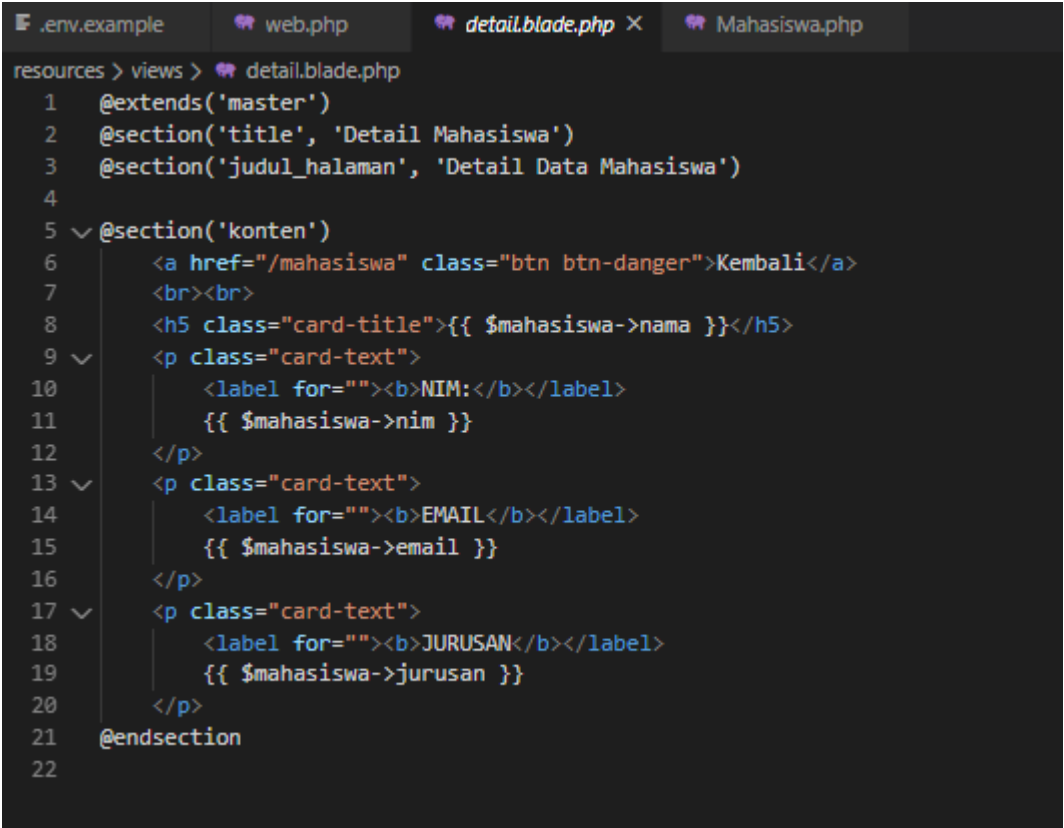
### Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM	
nanda eka	1001001001	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
wahyu	1001001002	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
marsha	1001001003	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
yasin	1001001004	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
rahmat	1001001006	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

#### d. Melihat Detail Data dari Database

Langkah	Keterangan
1	Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/detail</b> yang akan menjalankan fungsi detail pada MahasiswaController

	<pre>Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');</pre>
3	<p>Buat method <b>detail</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan data mahasiswa pada view detail.blade.php.</p> <pre>public function detail(\$id){     \$mahasiswa = Mahasiswa::find(\$id);     return view('detail', ['mahasiswa' =&gt; \$mahasiswa]); }</pre>
4	<p>Kemudian buatlah view <b>detail.blade.php</b> yang menampilkan detail data mahasiswa pada folder <b>resources/views</b>. View detail juga mengaplikasikan master.blade.php.</p>  <pre>resources &gt; views &gt; detail.blade.php 1 @extends('master') 2 @section('title', 'Detail Mahasiswa') 3 @section('judul_halaman', 'Detail Data Mahasiswa') 4 5 @section('konten') 6     &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 7     &lt;br&gt;&lt;br&gt; 8     &lt;h5 class="card-title"&gt;{{ \$mahasiswa-&gt;nama }}&lt;/h5&gt; 9     &lt;p class="card-text"&gt; 10         &lt;label for=""&gt;&lt;b&gt;NIM:&lt;/b&gt;&lt;/label&gt; 11         {{ \$mahasiswa-&gt;nim }} 12     &lt;/p&gt; 13     &lt;p class="card-text"&gt; 14         &lt;label for=""&gt;&lt;b&gt;EMAIL&lt;/b&gt;&lt;/label&gt; 15         {{ \$mahasiswa-&gt;email }} 16     &lt;/p&gt; 17     &lt;p class="card-text"&gt; 18         &lt;label for=""&gt;&lt;b&gt;JURUSAN&lt;/b&gt;&lt;/label&gt; 19         {{ \$mahasiswa-&gt;jurusan }} 20     &lt;/p&gt; 21 @endsection 22</pre>
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 <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/edit</b> yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre>Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');</pre>
2	<p>Buat method <b>edit</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view edit.</p>

	<pre> public function edit(\$id){     \$mahasiswa = Mahasiswa::find(\$id);     return view('edit', ['mahasiswa' =&gt; \$mahasiswa]); } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih</li> </ul>
3	<p>Kemudian buatlah view <b>edit.blade.php</b> yang berisi form untuk mengubah data pada folder <b>resources/views</b>. View edit juga mengaplikasikan master.blade.php.</p> <pre> resources &gt; views &gt; edit.blade.php 1  @extends('master') 2  @section('title', 'Edit Data') 3  @section('judul_halaman', 'Edit Data Mahasiswa') 4 5  @section('konten') 6      &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 7      &lt;br&gt;&lt;br&gt; 8      &lt;form action="/laravel-crud-kedua/public/mahasiswa/update/{{ \$mahasiswa-&gt;id }}" method="p 9          {{ csrf_field() }} 10         &lt;input type="hidden" name="id" value="{{ \$mahasiswa-&gt;id }}"&gt;&lt;br&gt; 11         &lt;div class="form-group"&gt; 12             &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 13             &lt;input type="text" name="namamhs" class="form-control" value="{{ \$mahasiswa-&gt;nama }} 14         &lt;/div&gt; 15         &lt;div class="form-group"&gt; 16             &lt;label for="nimhs"&gt;NIM&lt;/label&gt; 17             &lt;input type="number" name="nimhs" class="form-control" value="{{ \$mahasiswa-&gt;nim }} 18         &lt;/div&gt; 19         &lt;div class="form-group"&gt; 20             &lt;label for="emailmhs"&gt;E-mail&lt;/label&gt; 21             &lt;input type="text" name="emailmhs" class="form-control" value="{{ \$mahasiswa-&gt;email 22         &lt;/div&gt; 23         &lt;div class="form-group"&gt; 24             &lt;label for="jurusanmhs"&gt;Jurusan&lt;/label&gt; 25             &lt;input type="text" name="jurusanmhs" class="form-control" value="{{ \$mahasiswa-&gt;jur 26         &lt;/div&gt; 27         &lt;button type="submit" name="tambah" class="btn btn-primary float-right"&gt;Simpan Data&lt;/ 28     &lt;/form&gt; 29 @endsection 30 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 11-31 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan</li> <li>- Line 11 terdapat action="/mahasiswa/update/{{ \$mahasiswa-&gt;id }}" yang menunjukkan routes /mahasiswa/update/{id} dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController</li> </ul>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/update/{id}</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update'); </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php</li> </ul>

5	Buat method <b>update</b> pada <b>MahasiswaController</b> untuk menyimpan data yang diubah ke database.
---	---------------------------------------------------------------------------------------------------------



	<pre> public function update(\$id, Request \$request){     \$mahasiswa = Mahasiswa::find(\$id);     \$mahasiswa-&gt;nama = \$request-&gt;namamhs;     \$mahasiswa-&gt;nim = \$request-&gt;nimmhs;     \$mahasiswa-&gt;email = \$request-&gt;emailmhs;     \$mahasiswa-&gt;jurusan = \$request-&gt;jurusanmhs;     \$mahasiswa-&gt;save();     return redirect('/'); } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li> <li>- Line 48-52 merupakan fungsi eloquent untuk update data ke tabel mahasiswa</li> </ul>
6	<p>Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru.</p> <p>Cobalah untuk mengedit data.</p>

#### f. Menghapus Data (Delete) dari Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/hapus</b> yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); </pre>
2	<p>Buat method <b>hapus</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menjalankan fungsi hapus.</p> <pre> public function hapus(\$id){     \$mahasiswa = Mahasiswa::find(\$id);     \$mahasiswa-&gt;delete();     return redirect('/'); } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih</li> </ul>
3	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.</p>

-- Selamat Mengerjakan --