

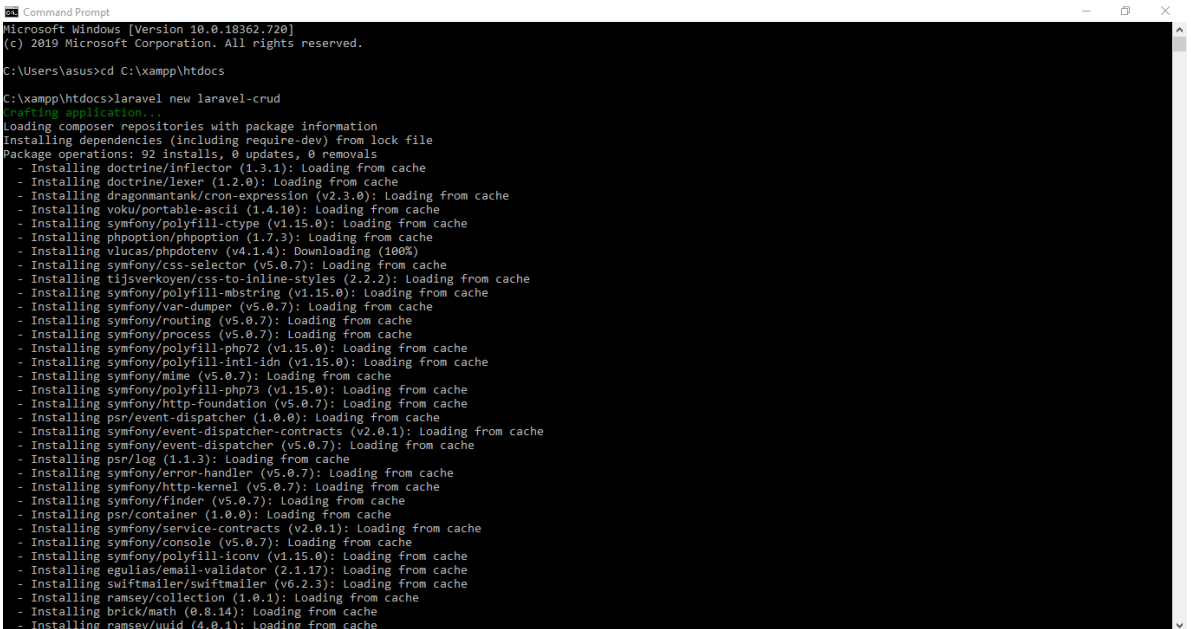


Nama : Septianda Reza Maulana  
Kelas : TI-2A  
NIM : 1841720135

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

Query Builder adalah salah satu fitur untuk menjalankan *query database*. Laravel menyediakan fungsi-fungsi yang dapat digunakan untuk menjalankan *query* sehingga penulisan query menjadi lebih singkat dan efisien.

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

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



```

Command Prompt
sebastian/global-state suggests installing ext-uopz (*)
sebastian/environment suggests installing ext-posix (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.7.2)
phpunit/phpunit suggests installing ext-soap (*)
phpunit/phpunit suggests installing ext-xdebug (*)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0.0)
Generating optimized autoload files
15 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
> @php artisan key:generate --ansi
Application key set successfully.
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
Application ready! Build something amazing.
C:\xampp\htdocs>

```

This PC > Local Disk (C:) > xampp > htdocs

| Name              | Date modified   | Type        | Size |
|-------------------|-----------------|-------------|------|
| belajarjavascript | 10-Nov-19 16:54 | File folder |      |
| Code-Kedelapan    | 07-Apr-20 8:10  | File folder |      |
| dasarWeb          | 09-Dec-19 16:14 | File folder |      |
| dashboard         | 10-Nov-19 16:36 | File folder |      |
| DOM               | 10-Nov-19 16:56 | File folder |      |
| img               | 10-Nov-19 16:36 | File folder |      |
| laravelapp        | 07-Apr-20 10:21 | File folder |      |
| laravel-crud      | 14-Apr-20 8:20  | File folder |      |

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

EXPLORER

OPEN EDITORS

.env laravel-crud

UNTITLED (WORKSPACE)

database

public

resources

routes

storage

tests

vendor

.editorconfig

**.env**

.env.example

.env

laravel-crud > .env

1 APP\_NAME=Laravel

2 APP\_ENV=local

3 APP\_KEY=base64:6/yIppqWFWVI09D2BPH+v2T3JrM2BMfppU3BPY+k1EgI=

4 APP\_DEBUG=true

5 APP\_URL=http://localhost

6

7 LOG\_CHANNEL=stack

8

9 DB\_CONNECTION=mysql

10 DB\_HOST=127.0.0.1

11 DB\_PORT=3306

12 DB\_DATABASE=latihan\_laravel

13 DB\_USERNAME=root

14 DB\_PASSWORD=

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.

### Buat basis data

latihan\_laravel

utf8mb4\_general\_ci

Buat

### Buat tabel

Nama: mahasiswa

Jumlah kolom: 5

Kirim

Server: 127.0.0.1 » Basis data: latihan\_laravel » Tabel: mahasiswa

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Pelacakan

Struktur tabel Tampilan hubungan

| # | Nama    | Jenis        | Penyortiran        | Atribut | Tak Ternilai | Bawaan    | Komentar | Ekstra         | Tindakan           |
|---|---------|--------------|--------------------|---------|--------------|-----------|----------|----------------|--------------------|
| 1 | id      | int(11)      |                    |         | Tidak        | Tidak ada |          | AUTO_INCREMENT | Ubah Hapus Lainnya |
| 2 | nama    | varchar(100) | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 3 | nim     | varchar(15)  | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 4 | email   | varchar(50)  | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 5 | jurusan | varchar(30)  | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |

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

Server: 127.0.0.1 » Basis data: latihan\_laravel » Tabel: mahasiswa

JelajahiStrukturSQLCariTambahkanEksporImporHak AksesOperasi

Menampilkan baris 0 - 1 (total 2, Pencarian dilakukan dalam 0,0006 detik.)

SELECT \* FROM `mahasiswa`

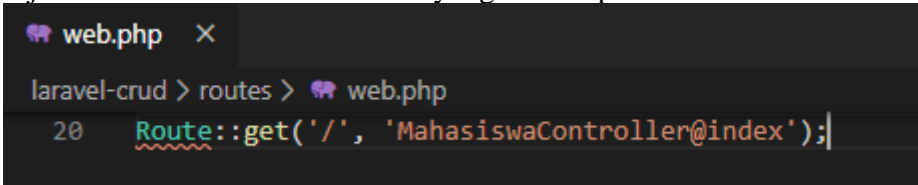
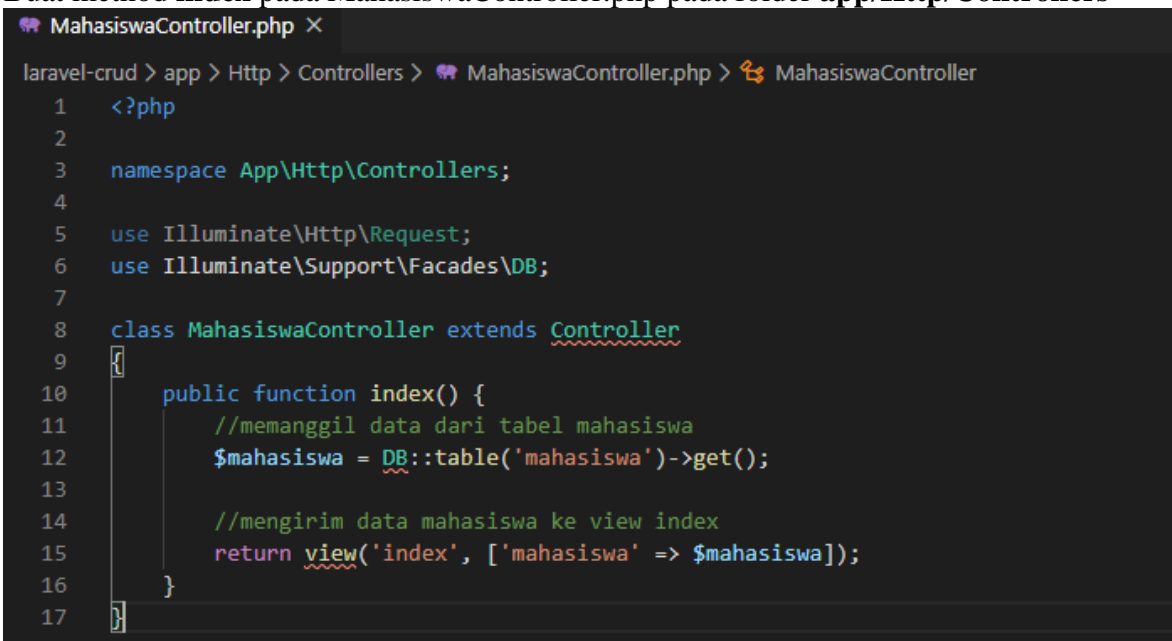
Profil [Edit kotak] [Ubah] [J

Tampilkan semua | Jumlah baris: 25 Saring baris: Cari di tabel ini Urut berdasarkan kunci: Tidak ada

+ Opsi

|                          |      |       | id    | nama | nim                    | email      | jurusan                   |                     |
|--------------------------|------|-------|-------|------|------------------------|------------|---------------------------|---------------------|
| <input type="checkbox"/> | Ubah | Salin | Hapus | 1    | Septianda Reza Maulana | 1841720135 | septiandareza07@gmail.com | Teknologi Informasi |
| <input type="checkbox"/> | Ubah | Salin | Hapus | 2    | Ainur Hasan Al Rasyid  | 1842530008 | ainur@gmail.com           | Akuntansi           |

## 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> <p><b>cd laravel-crud</b><br/> <b>php artisan make:controller MahasiswaController</b></p>   |
| 3       | <p>Buat method <b>index</b> pada MahasiswaController.php pada folder <b>app/Http/Controllers</b></p>  <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Tambahkan 'use Illuminate\Support\Facades\DB;' (line 6) agar <i>query builder</i> dapat digunakan</li> <li>- Line 13 untuk mengambil data dari tabel mahasiswa menggunakan <i>query builder</i> laravel dan akan disimpan di variabel \$mahasiswa</li> <li>Line 16 : data akan dikirim ke blade view bernama index</li> </ul>     |

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**.

```
master.blade.php X
laravel-crud > resources > views > master.blade.php > html
1 <html>
2   <head>
3     <meta charset="utf-8">
4     <meta name="viewport" content="width=device-width, initial-scale=1">
5     <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" rel="stylesheet">
6     <title> @yield('title')</title>
7   </head>
8   <body>
9     <div class="container">
10      <div class="row mt-3">
11        <div class="col-md-6">
12          <div class="card">
13            <div class="card-header text-center">
14              <!-- bagian judul halaman -->
15              <h2> @yield('judul_halaman')</h2>
16            </div>
17            <div class="card-body">
18              <!-- bagian konten blog -->
19              @yield('konten')
20            </div>
21          </div>
22        </div>
23      </div>
24    </div>
25  </body>
26 </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`

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

```
index.blade.php X
laravel-crud > 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
```

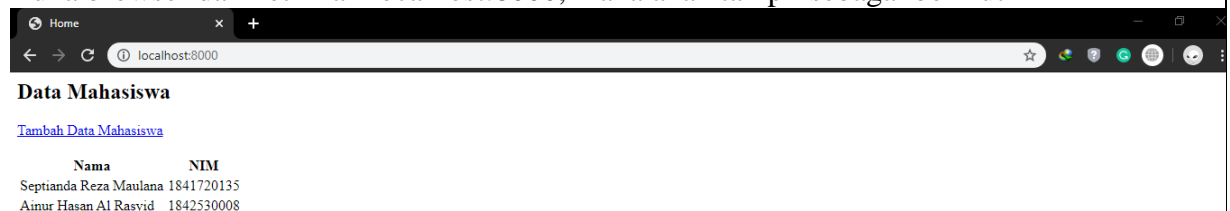
5

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

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

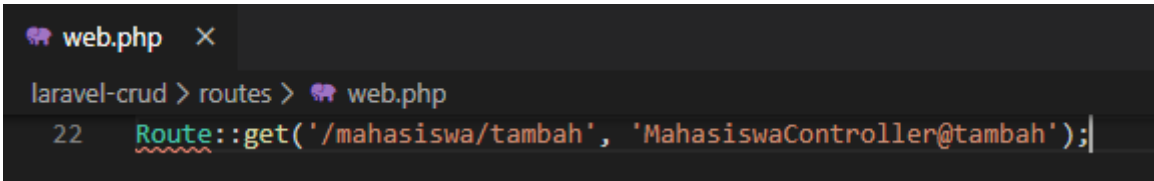
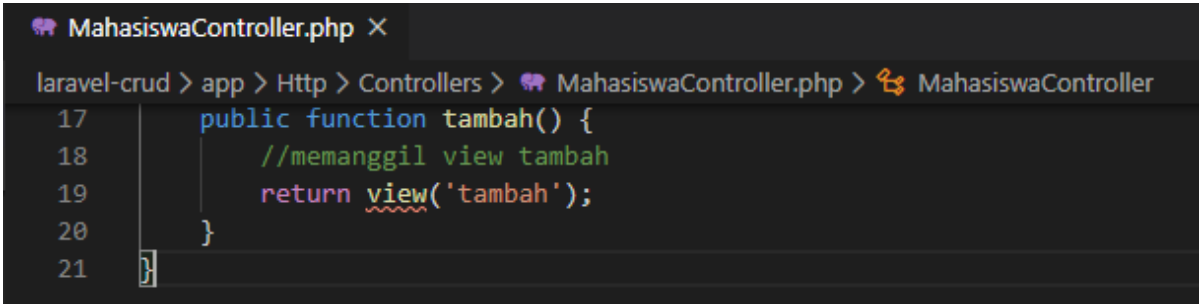
Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut



6



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> web.php × laravel-crud &gt; routes &gt; web.php 22 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> MahasiswaController.php × laravel-crud &gt; app &gt; Http &gt; Controllers &gt; MahasiswaController.php &gt; MahasiswaController 17 public function tambah() { 18     //memanggil view tambah 19     return view('tambah'); 20 } 21 } </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
laravel-crud > 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     <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
12     <br/>
13     <br/>
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"> <br/>
19         </div>
20         <div class="form-group">
21             <label for="nimhs">NIM</label>
22             <input type="number" class="form-control" required="required" name="nimhs"> <br/>
23         </div>
24         <div class="form-group">
25             <label for="emailmhs">Email</label>
26             <input type="email" class="form-control" required="required" name="emailmhs"> <br/>
27         </div>
28         <div class="form-group">
29             <label for="jurusanmhs">Jurusan</label>
30             <input type="text" class="form-control" required="required" name="jurusanmhs"> <br/>
31         </div>
32         <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button>
33     </form>
```

Keterangan:

- Line 14-33 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 14 terdapat action="/mahasiswa/simpan" yang menunjukkan routes /mahasiswa/simpan dimana data pada form tersebut akan dikirimkan ke fungsi simpan pada controller MahasiswaController
- Line 15 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.

```
web.php X
laravel-crud > routes > web.php
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.

```

MahasiswaController.php X
laravel-crud > app > Http > Controllers > MahasiswaController.php > MahasiswaController

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

```

Keterangan:

- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 23-28 merupakan query builder untuk insert data ke tabel mahasiswa

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

NIM

Email

Jurusan

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

Data Mahasiswa

[Tambah Data Mahasiswa](#)

| Nama                   | NIM        |
|------------------------|------------|
| Septianda Reza Maulana | 1841720135 |
| Ainur Hasan Al Rasyid  | 1842530008 |
| Septianda Reza         | 1841720135 |

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 27) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre> index.blade.php × laravel-crud &gt; resources &gt; views &gt; index.blade.php &gt; ... 22         @foreach(\$mahasiswa as \$mhs) 23             &lt;tr&gt; 24                 &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 25                 &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 26                 &lt;td&gt; 27                     &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 28                 &lt;/td&gt; 29             &lt;/tr&gt; 30         @endforeach 31     &lt;/tbody&gt; 32 &lt;/table&gt; 33 @endsection </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> web.php × laravel-crud &gt; routes &gt; web.php 26 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> MahasiswaController.php × laravel-crud &gt; app &gt; Http &gt; Controllers &gt; MahasiswaController.php &gt; MahasiswaController 31     public function detail(\$id) { 32         //mengambil data mahasiswa berdasarkan id yang dipilih 33         \$mahasiswa = DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;get(); 34         //kirim data mahasiswa yang diambil ke view edit.blade.php 35         return view('detail',['mahasiswa' =&gt; \$mahasiswa]); 36     } 37 } </pre>  |

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

4

```

detail.blade.php ✕

laravel-crud > 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="/mahasiswa" class="btn btn-danger">Kembali</a>
12     <br/>
13     <br/>
14
15     @foreach($mahasiswa as $mhs)
16         <h5 class="card-title"> {{ $mhs->nama }}</h5>
17         <p class="card-text">
18             <label for=""><b> NIM : </b></label>
19             {{ $mhs->nim }} </p>
20         <p class="card-text">
21             <label for=""><b> Email : </b></label>
22             {{ $mhs->email }} </p>
23         <p class="card-text">
24             <label for=""><b> Jurusan : </b></label>
25             {{ $mhs->jurusan }} </p>
26     @endforeach
27 @endsection

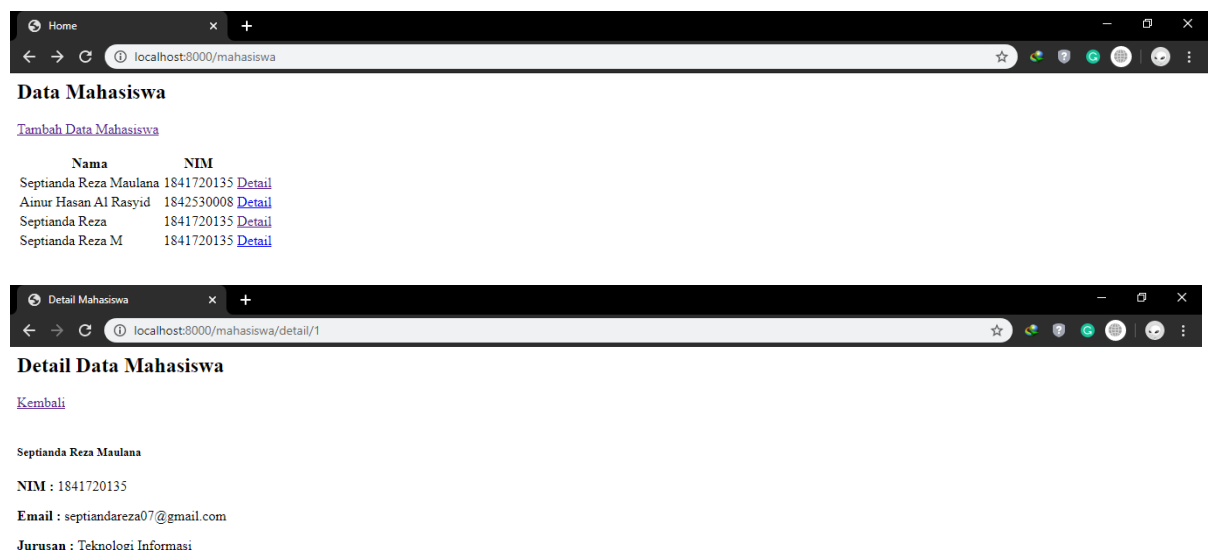
```

Keterangan:

- Line 15-26 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan

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

5



e. Mengubah Data (Update) dari Database

| Langkah | Keterangan  |
|---------|---|
| 1       | <p>Buatlah tombol untuk mengubah data (Line 28) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre> index.blade.php X laravel-crud &gt; resources &gt; views &gt; index.blade.php &gt; ... 22      @foreach(\$mahasiswa as \$mhs) 23      &lt;tr&gt; 24          &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 25          &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 26          &lt;td&gt; 27              &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 28              &lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Edit&lt;/a&gt; 29          &lt;/td&gt; 30      &lt;/tr&gt; 31      @endforeach 32  &lt;/tbody&gt; 33  &lt;/table&gt; 34  @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> web.php X laravel-crud &gt; routes &gt; web.php 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> MahasiswaController.php X laravel-crud &gt; app &gt; Http &gt; Controllers &gt; MahasiswaController.php &gt; MahasiswaController 37  public function edit(\$id) { 38      //mengambil data mahasiswa berdasarkan id yang dipilih 39      \$mahasiswa = DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;get(); 40      //kirim data mahasiswa yang diambil ke view edit.blade.php 41      return view('edit',['mahasiswa' =&gt; \$mahasiswa]); 42  } 43 </pre> <p>Keterangan :</p> <p>-Line 39 : query builder untuk mengambil data mahasiswa berdasarkan id yang dipilih</p>  |

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

```

edit.blade.php X
laravel-crud > resources > views > edit.blade.php > form
1  @extends('master')
2
3  <!-- Isi title -->
4  @section('title', 'Edit Data')
5
6  <!-- Isi bagian judul halaman -->
7  @section('judul_halaman', 'Edit Data Mahasiswa')
8
9  <!-- Isi bagian konten -->
10 @section('konten')
11     <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
12     <br/>
13     <br/>
14     @foreach($mahasiswa as $mhs)
15         <form action="/mahasiswa/update" method="post">
16             {{ csrf_field() }}
17             <input type="hidden" name="id" value="{{ $mhs->id }}"> <br/>
18             <div class="form-group">
19                 <label for="namamhs">Nama</label>
20                 <input type="text" class="form-control" required="required" name="namamhs"> <br/>
21             </div>
22             <div class="form-group">
23                 <label for="nimmhs">NIM</label>
24                 <input type="number" class="form-control" required="required" name="nimmhs"> <br/>
25             </div>
26             <div class="form-group">
27                 <label for="emailmhs">Email</label>
28                 <input type="email" class="form-control" required="required" name="emailmhs"> <br/>
29             </div>
30             <div class="form-group">
31                 <label for="jurusanmhs">Jurusan</label>
32                 <input type="text" class="form-control" required="required" name="jurusanmhs"> <br/>
33             </div>
34             <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button>
35         </form>
36     @endforeach
37 @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

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

```

web.php X
laravel-crud > routes > web.php
30 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

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

```
MahasiswaController.php X
laravel-crud > app > Http > Controllers > MahasiswaController.php > MahasiswaController

43 public function update(Request $request) {
44     //update data mahasiswa
45     DB::table('mahasiswa')->where('id',$request->id)->update([
46         'nama'      => $request->namamhs,
47         'nim'       => $request->nimmhs,
48         'email'     => $request->emailmhs,
49         'jurusan'   => $request->jurusanmhs
50     ]);
51     return redirect('/mahasiswa');
52 }
53
```

Keterangan:

- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 45-50 merupakan query builder untuk update data ke tabel mahasiswa

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

Home x +  
localhost:8000/mahasiswa  
Data Mahasiswa  
[Tambah Data Mahasiswa](#)  


| Nama                   | NIM        |   |
|------------------------|------------|---|
| Septianda Reza Maulana | 1841720135 | <a href="#">Detail</a> <a href="#">Edit</a> |
| Ainur Hasan Al Rasyid  | 1842530008 | <a href="#">Detail</a> <a href="#">Edit</a> |
| Septianda Reza         | 1841720135 | <a href="#">Detail</a> <a href="#">Edit</a> |
| Septianda Reza M       | 1841720135 | <a href="#">Detail</a> <a href="#">Edit</a> |

Klik edit di data keempat, kita akan mengubah namanya.

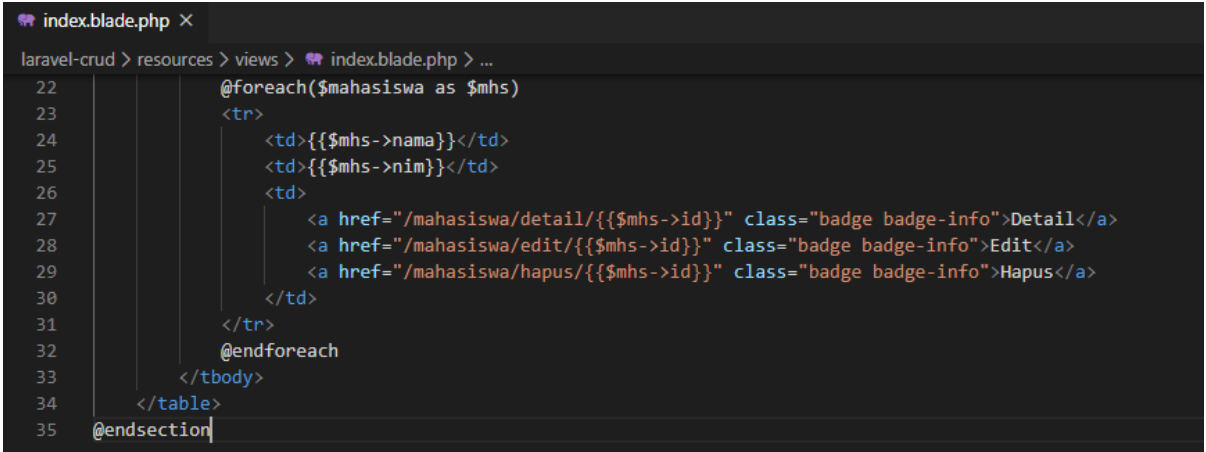
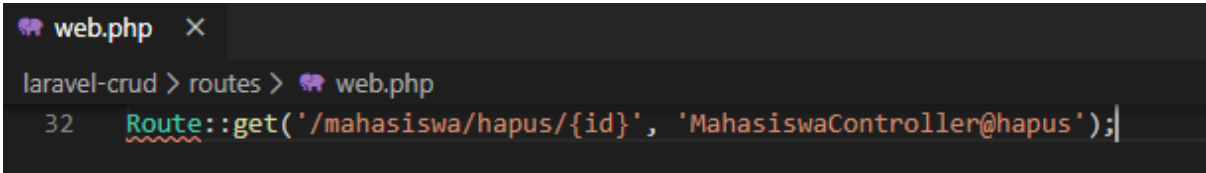
Edit Data x +  
localhost:8000/mahasiswa/edit/6  
Edit Data Mahasiswa  
[Kembali](#)  

|  |  |
|--|--|
| Nama                                       | <input type="text" value="Septianda"/>                 |
| NIM  | <input type="text" value="1841720135"/>                |
| Email                                      | <input type="text" value="septiandareza07@gmail.com"/> |
| Jurusan                                    | <input type="text" value="Teknologi Informasi"/>       |
| <input type="button" value="Simpan Data"/> |  |



|  |  |
|--|--|
|  | <p>Setelah kita klik simpan Data, maka data pertama akan berubah.</p>  |
|--|--|

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

| Langkah | Keterangan   |
|---------|--|
| 1       | <p>Buatlah tombol untuk menghapus data (Line 29) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p>                                        |
| 2       | <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>  |

3

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

```
MahasiswaController.php X
C: > xampp > htdocs > laravel-crud > app > Http > Controllers > MahasiswaController.php > MahasiswaController

53     public function hapus($id) {
54         //menghapus data mahasiswa berdasarkan id yang dipilih
55         DB::table('mahasiswa')->where('id',$id)->delete();
56
57         return redirect('mahasiswa');
58     }
59 }
```

Keterangan :

- Line 53 : 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.



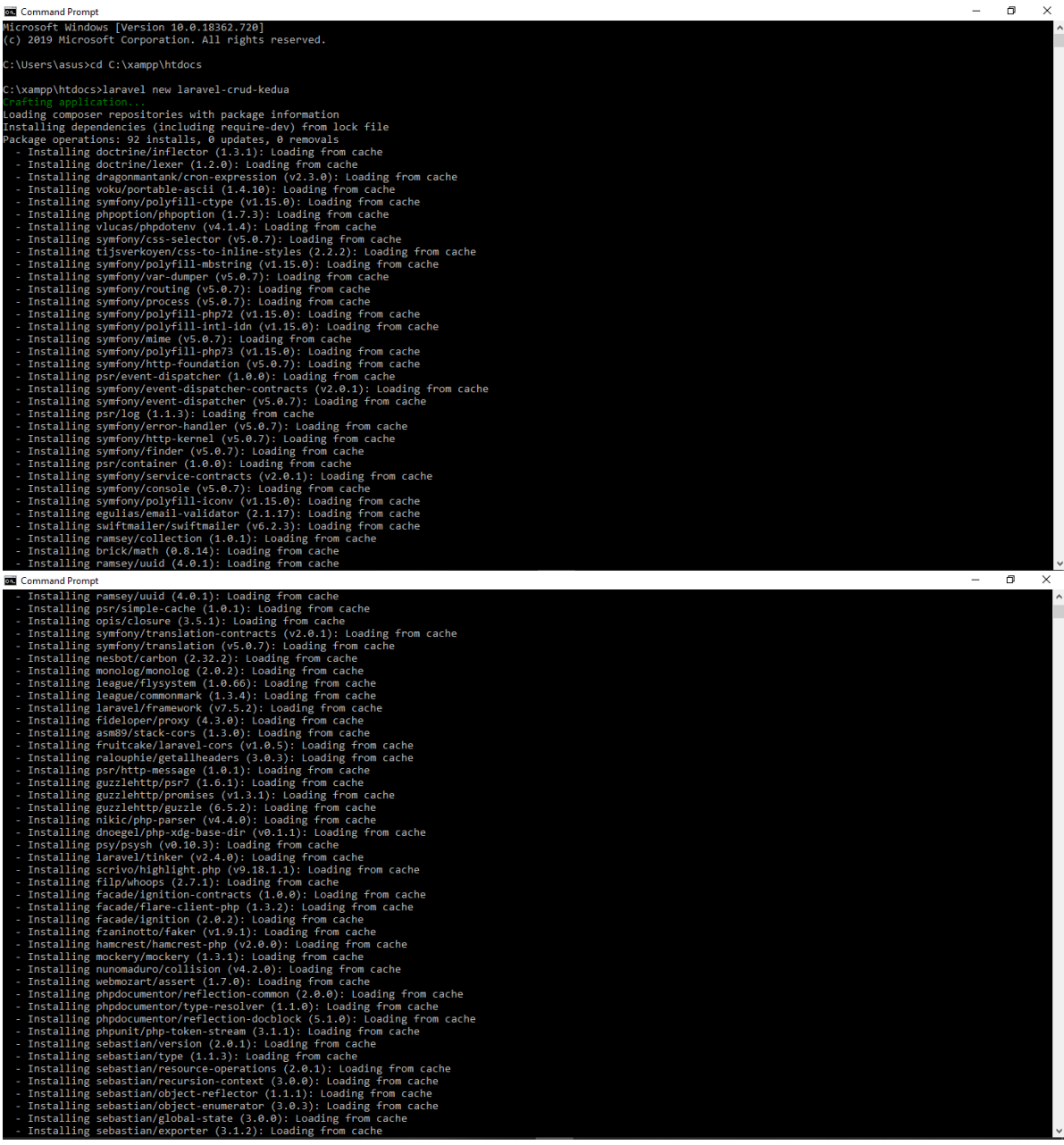
Hasil :



## 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 <b>laravel-crud-kedua</b>. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud-kedua</pre>  |



This PC > Local Disk (C:) > xampp > htdocs

| Name               | Date modified   | Type        | Size |
|--------------------|-----------------|-------------|------|
| belajarjavascript  | 10-Nov-19 16:54 | File folder |      |
| Code-Kedelapan     | 07-Apr-20 8:10  | File folder |      |
| dasarWeb           | 09-Dec-19 16:14 | File folder |      |
| dashboard          | 10-Nov-19 16:36 | File folder |      |
| DOM                | 10-Nov-19 16:56 | File folder |      |
| img                | 10-Nov-19 16:36 | File folder |      |
| laravelapp         | 07-Apr-20 10:21 | File folder |      |
| laravel-crud       | 14-Apr-20 8:20  | File folder |      |
| laravel-crud-kedua | 14-Apr-20 10:59 | File folder |      |

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:zK0/2yqBGFnURbh08fiYyTLmlFtgnLtb3b+sBj1H3E0=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=latihan_laravel
13 DB_USERNAME=root
14 DB_PASSWORD=
```

Keterangan:

- Nama database yang akan digunakan adalah latihan\_laravel dengan username root.

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

| # | Nama    | Jenis        | Penyortiran        | Atribut | Tak Ternilai | Bawaan    | Komentar | Ekstra         | Tindakan           |
|---|---------|--------------|--------------------|---------|--------------|-----------|----------|----------------|--------------------|
| 1 | id      | int(11)      |                    |         | Tidak        | Tidak ada |          | AUTO_INCREMENT | Ubah Hapus Lainnya |
| 2 | nama    | varchar(100) | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 3 | nim     | varchar(15)  | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 4 | email   | varchar(50)  | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 5 | jurusan | varchar(30)  | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |

Tambahkan 2 kolom setelah jurusan **Kirim**

Server: 127.0.0.1 » Basis data: latihan\_laravel » Tabel: mahasiswa

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Pelacakan Trigger

| Nama       | Jenis     | Panjang/Nilai | Bawaan    | Penyortiran | Atribut | Tak Ternilai                        | Indeks |
|------------|-----------|---------------|-----------|-------------|---------|-------------------------------------|--------|
| created_at | TIMESTAMP |               | Tidak ada |             |         | <input checked="" type="checkbox"/> | ---    |
| updated_at | TIMESTAMP |               | Tidak ada |             |         | <input checked="" type="checkbox"/> | ---    |

Pratinjau SQL Simpan

---

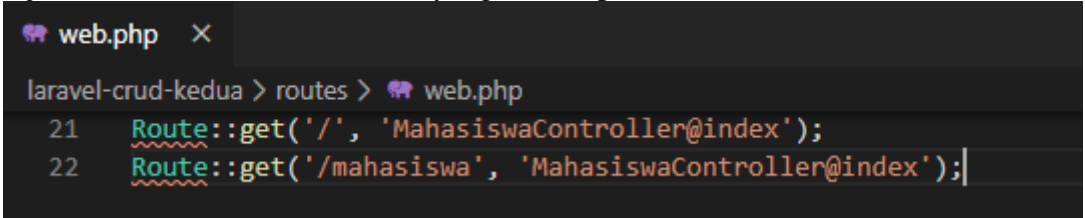

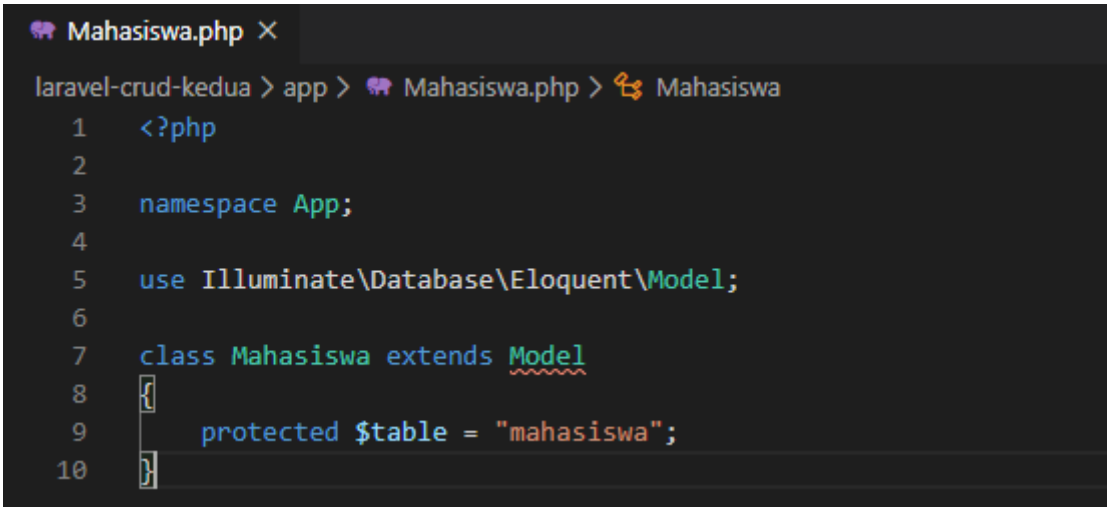

Server: 127.0.0.1 » Basis data: latihan\_laravel » Tabel: mahasiswa

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Pelacakan

Struktur tabel Tampilan hubungan

| # | Nama       | Jenis        | Penyortiran        | Atribut | Tak Ternilai | Bawaan    | Komentar | Ekstra         | Tindakan           |
|---|------------|--------------|--------------------|---------|--------------|-----------|----------|----------------|--------------------|
| 1 | id         | int(11)      |                    |         | Tidak        | Tidak ada |          | AUTO_INCREMENT | Ubah Hapus Lainnya |
| 2 | nama       | varchar(100) | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 3 | nim        | varchar(15)  | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 4 | email      | varchar(50)  | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 5 | jurusan    | varchar(30)  | utf8mb4_general_ci |         | Tidak        | Tidak ada |          |                | Ubah Hapus Lainnya |
| 6 | created_at | timestamp    |                    |         | Ya           | NULL      |          |                | Ubah Hapus Lainnya |
| 7 | updated_at | timestamp    |                    |         | Ya           | NULL      |          |                | Ubah Hapus Lainnya |

## b. Menampilkan Data dari Database

| 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> web.php laravel-crud-kedua &gt; routes &gt; web.php 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> <p><b>cd laravel-crud-kedua</b><br/> <b>php artisan make:model Mahasiswa</b></p>  <pre> Command Prompt Microsoft Windows [Version 10.0.18362.720] (c) 2019 Microsoft Corporation. All rights reserved.  C:\Users\asus&gt;cd C:\xampp\htdocs C:\xampp\htdocs&gt;cd laravel-crud-kedua C:\xampp\htdocs\laravel-crud-kedua&gt;php artisan make:model Mahasiswa Model created successfully. C:\xampp\htdocs\laravel-crud-kedua&gt; </pre>  |
| 3       | <p>Ubah model <b>Mahasiswa.php</b> pada folder <b>App</b> menjadi seperti berikut.</p>  <pre> Mahasiswa.php laravel-crud-kedua &gt; app &gt; Mahasiswa.php &gt; Mahasiswa 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 } </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>  <pre> Command Prompt Microsoft Windows [Version 10.0.18362.720] (c) 2019 Microsoft Corporation. All rights reserved.  C:\Users\asus&gt;cd C:\xampp\htdocs C:\xampp\htdocs&gt;cd laravel-crud-kedua C:\xampp\htdocs\laravel-crud-kedua&gt;php artisan make:model Mahasiswa Model created successfully. C:\xampp\htdocs\laravel-crud-kedua&gt;php artisan make:controller MahasiswaController Controller created successfully. C:\xampp\htdocs\laravel-crud-kedua&gt; </pre> |



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

```

MahasiswaController.php X
laravel-crud-kedua > 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
8  class MahasiswaController extends Controller
9  {
10     public function index() {
11         $mahasiswa = Mahasiswa::all();
12         return view('index', ['mahasiswa' => $mahasiswa]);
13     }
14 }

```

Keterangan:


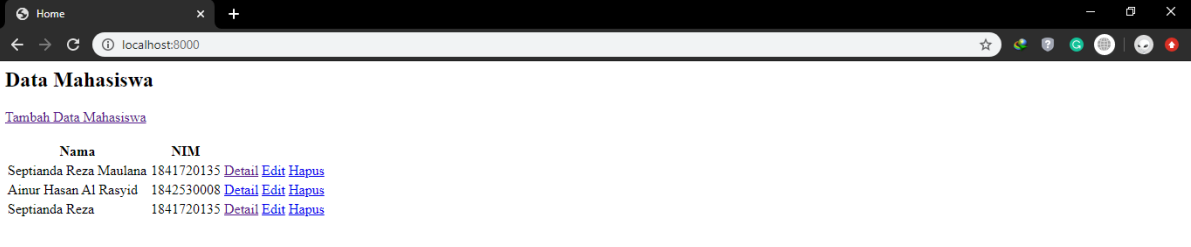
- Tambahkan 'use App\Mahasiswa' (line 6) untuk menggunakan model Mahasiswa
- Line 12 untuk mengambil semua data dari model/tabel Mahasiswa dan akan disimpan di variabel \$mahasiswa
- Line 16 : data akan dikirim ke blade view bernama index

Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat **template blade** (seperti pada Bagian 1).

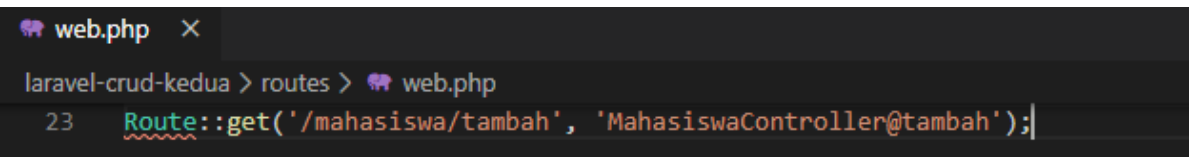
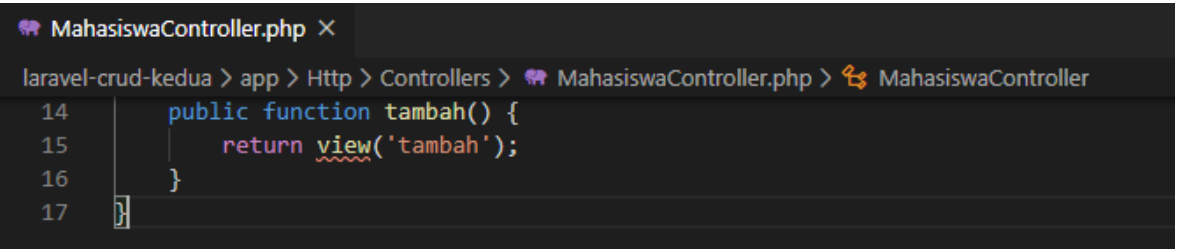
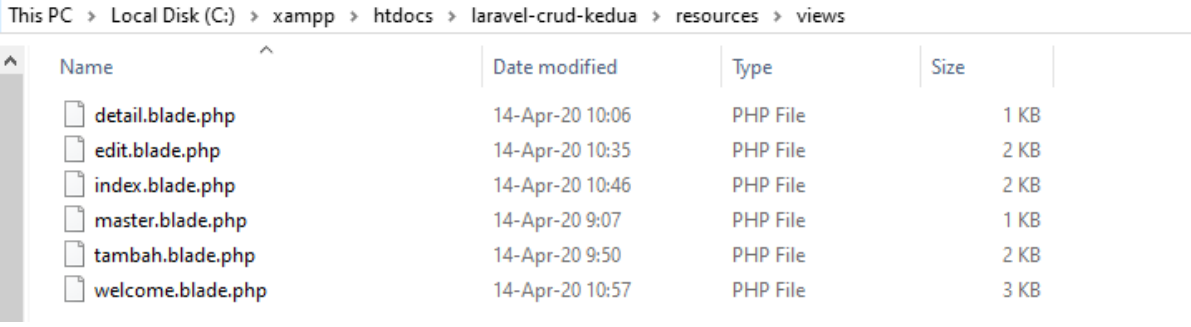
Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)

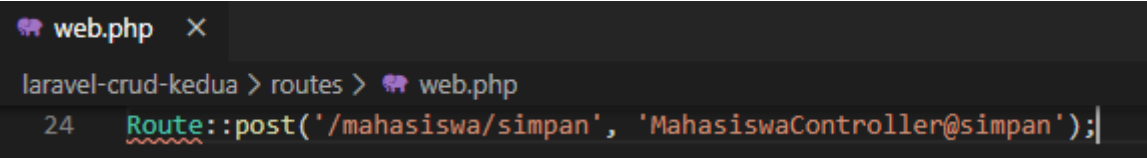
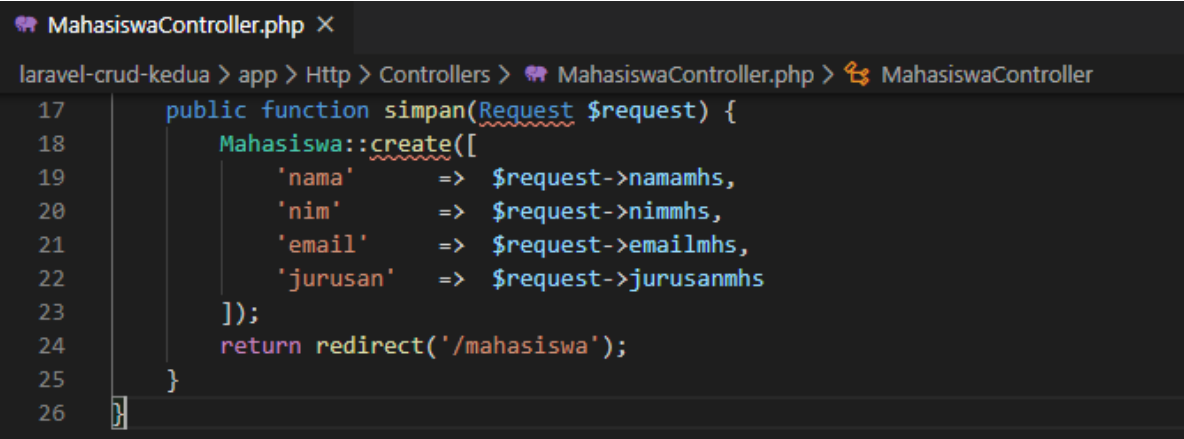
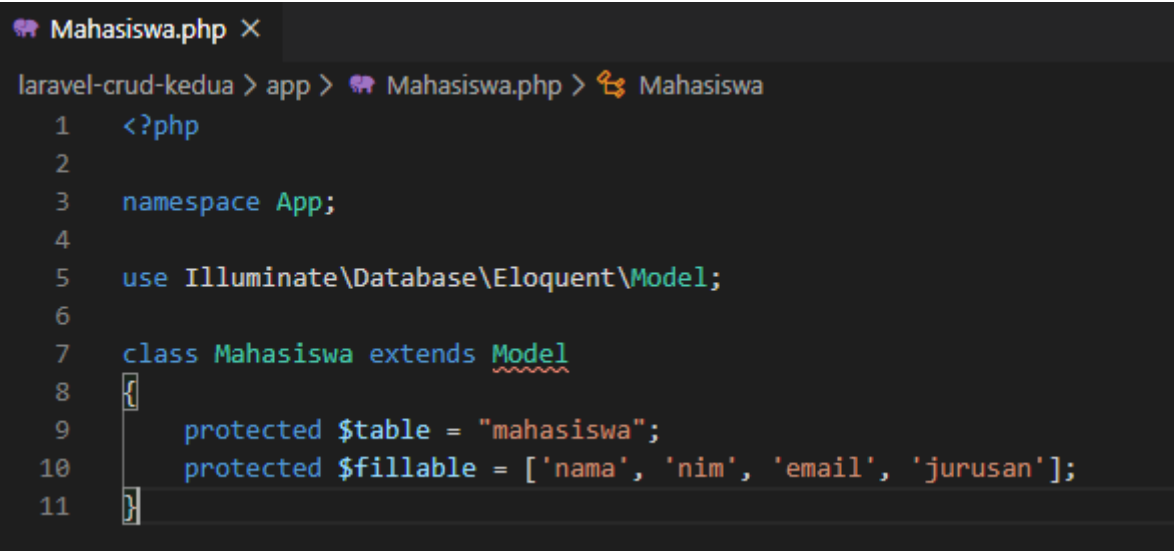
This PC > Local Disk (C:) > xampp > htdocs > laravel-crud-kedua > resources > views

| Name              | Date modified   | Type     | Size |
|-------------------|-----------------|----------|------|
| detail.blade.php  | 14-Apr-20 10:06 | PHP File | 1 KB |
| edit.blade.php    | 14-Apr-20 10:35 | PHP File | 2 KB |
| index.blade.php   | 14-Apr-20 10:46 | PHP File | 2 KB |
| master.blade.php  | 14-Apr-20 9:07  | PHP File | 1 KB |
| tambah.blade.php  | 14-Apr-20 9:50  | PHP File | 2 KB |
| welcome.blade.php | 14-Apr-20 10:57 | PHP File | 3 KB |

|   |   |
|---|---|
| 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>  |
| 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>    |
| 3       | <p>Kemudian buatlah view <b>tambah.blade.php</b> yang berisi form untuk memasukkan data baru pada folder <b>resources/views</b>.</p> <p>Kita lakukan copy paste dari tambah.blade.php di Bagian 1</p>      |

|   |   |
|---|---|
| 4 | <p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/simpan</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p>  <pre>laravel-crud-kedua &gt; routes &gt; web.php 24 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>laravel-crud-kedua &gt; app &gt; Http &gt; Controllers &gt; MahasiswaController.php &gt; MahasiswaController 17 public function simpan(Request \$request) { 18     Mahasiswa::create([ 19         'nama' =&gt; \$request-&gt;namamhs, 20         'nim' =&gt; \$request-&gt;nimmhs, 21         'email' =&gt; \$request-&gt;emailmhs, 22         'jurusan' =&gt; \$request-&gt;jurusanmhs 23     ]); 24     return redirect('/mahasiswa'); 25 } 26</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li> <li>- Line 18-23 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>  <pre>laravel-crud-kedua &gt; app &gt; Mahasiswa.php &gt; Mahasiswa 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 }</pre>  |

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'. Below the title is a link labeled 'Kembali'. The form contains the following fields: 'Nama' with the value 'Reza Maulana', 'NIM' with the value '1841720135', 'Email' with the value 'septiandareza07@gmail.com', and 'Jurusan' with the value 'Teknik Informatika'. At the bottom of the form is a button labeled 'Tambah Data'.

7

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




The screenshot shows a web browser window with the address bar displaying 'localhost:8000/mahasiswa'. The page title is 'Data Mahasiswa'. Below the title is a link labeled 'Tambah Data Mahasiswa'. The table displays the following data:

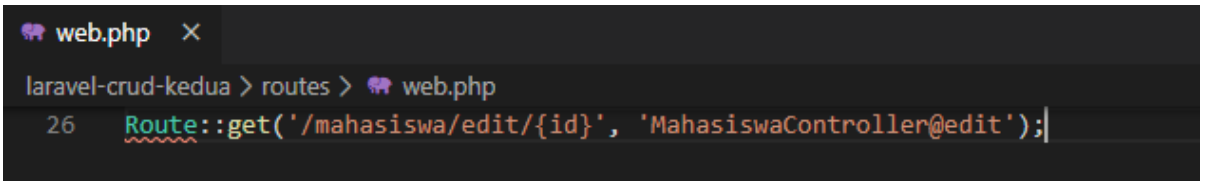
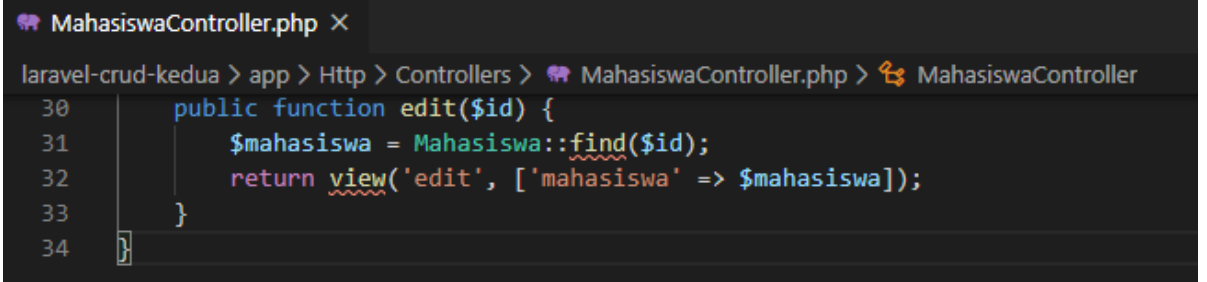
| Nama                   | NIM        |   |
|------------------------|------------|---|
| Septianda Reza Maulana | 1841720135 | <a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a> |
| Ainur Hasan Al Rasyid  | 1842530008 | <a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a> |
| Septianda Reza         | 1841720135 | <a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a> |
| Reza Maulana           | 1841720135 | <a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a> |

#### d. Melihat Detail Data dari Database

| Langkah | Keterangan  |
|---------|---|
| 1       | <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> web.php × laravel-crud-kedua &gt; routes &gt; web.php 25 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');  </pre>  |
| 2       | <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> MahasiswaController.php × laravel-crud-kedua &gt; app &gt; Http &gt; Controllers &gt; MahasiswaController.php &gt; MahasiswaController 26 public function detail(\$id) { 27     \$mahasiswa = Mahasiswa::find(\$id); 28     return view('detail', ['mahasiswa' =&gt; \$mahasiswa]); 29 } 30 </pre>  |
| 3       | <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> detail.blade.php × laravel-crud-kedua &gt; resources &gt; views &gt; detail.blade.php &gt; ... 1 @extends('master') 2 3 &lt;!-- Isi title --&gt; 4 @section('title', 'Detail Mahasiswa') 5 6 &lt;!-- Isi bagian judul halaman --&gt; 7 @section('judul_halaman', 'Detail Data Mahasiswa') 8 9 &lt;!-- Isi bagian konten --&gt; 10 @section('konten') 11     &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 12     &lt;br/&gt; 13     &lt;br/&gt; 14 15     &lt;h5 class="card-title"&gt; {{ \$mahasiswa-&gt;nama }}&lt;/h5&gt; 16     &lt;p class="card-text"&gt; 17         &lt;label for=""&gt;&lt;b&gt; NIM : &lt;/b&gt;&lt;/label&gt; 18         {{ \$mahasiswa-&gt;nim }} &lt;/p&gt; 19     &lt;p class="card-text"&gt; 20         &lt;label for=""&gt;&lt;b&gt; Email : &lt;/b&gt;&lt;/label&gt; 21         {{ \$mahasiswa-&gt;email }} &lt;/p&gt; 22     &lt;p class="card-text"&gt; 23         &lt;label for=""&gt;&lt;b&gt; Jurusan : &lt;/b&gt;&lt;/label&gt; 24         {{ \$mahasiswa-&gt;jurusan }} &lt;/p&gt; 25 @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 <b>routes/web.php</b> dengan nama <b>/mahasiswa/edit</b> yang akan menjalankan fungsi edit pada MahasiswaController</p>   |
| 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>  <p>Keterangan :</p> <p>-Line 31 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih</p> |

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

```

edit.blade.php X
laravel-crud-kedua > resources > views > edit.blade.php > ...
1  @extends('master')
2
3  <!-- Isi title -->
4  @section('title', 'Edit Data')
5
6  <!-- Isi bagian judul halaman -->
7  @section('judul_halaman', 'Edit Data Mahasiswa')
8
9  <!-- Isi bagian konten -->
10 @section('konten')
11     <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
12     <br/>
13     <br/>
14
15     <form action="/mahasiswa/update/{{ $mahasiswa->id }}" method="post">
16         {{ csrf_field() }}
17         <input type="hidden" name="id" value="{{ $mahasiswa->id }}" <br/>
18         <div class="form-group">
19             <label for="namamhs">Nama</label>
20             <input type="text" class="form-control" required="required" name="namamhs" value="{{ $mahasiswa->nama }}" <br/>
21         </div>
22         <div class="form-group">
23             <label for="nimamhs">NIM</label>
24             <input type="number" class="form-control" required="required" name="nimamhs" value="{{ $mahasiswa->nim }}" <br/>
25         </div>
26         <div class="form-group">
27             <label for="emailmhs">Email</label>
28             <input type="email" class="form-control" required="required" name="emailmhs" value="{{ $mahasiswa->email }}" <br/>
29         </div>
30         <div class="form-group">
31             <label for="jurusanmhs">Jurusan</label>
32             <input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ $mahasiswa->jurusan }}" <br/>
33         </div>
34         <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button>
35     </form>
36 @endsection

```

Keterangan:

- Line 15-35 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 15 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

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

```

web.php X
laravel-crud-kedua > routes > web.php
27 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.

```

MahasiswaController.php X
laravel-crud-kedua > app > Http > Controllers > MahasiswaController.php > MahasiswaController
34 public function update($id, Request $request) {
35     $mahasiswa = Mahasiswa::find($id);
36     $mahasiswa->nama = $request->namamhs;
37     $mahasiswa->nim = $request->nimmhs;
38     $mahasiswa->email = $request->emailmhs;
39     $mahasiswa->jurusan = $request->jurusanmhs;
40     $mahasiswa->save();
41     return redirect('/mahasiswa');
42 }
43

```

Keterangan:

- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 36-40 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.

**Sebelum diedit :**



Detail Data Mahasiswa

[Kembali](#)

Reza Maulana

NIM : 1841720135

Email : septiandareza07@gmail.com

Jurusan : Teknik Informatika

**Sesudah diedit :**



Detail Data Mahasiswa

[Kembali](#)

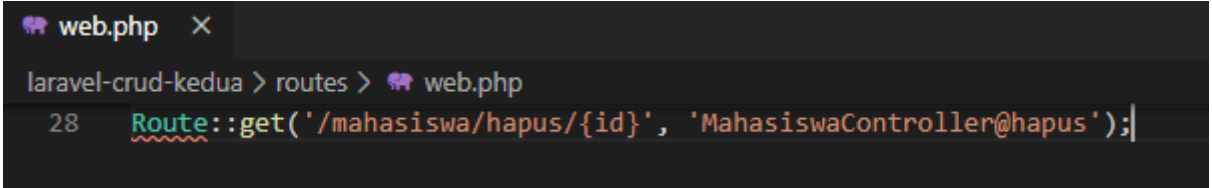
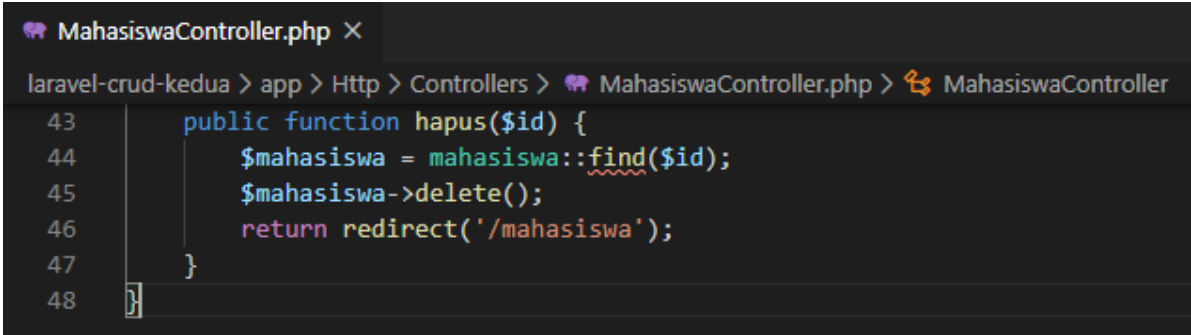


Reza Maulana

NIM : 1841720135

Email : septiandareza15@gmail.com

Jurusan : Teknik Informatika

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>   |
| 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>  <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 45 :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> <p><b>Sebelum dihapus :</b></p>  <p><b>Sesudah dihapus :</b></p>   |