

# **PEMROGRAMAN WEB LANJUT**

## **JOBSHEET 10** **Membuat CRUD dengan Laravel**



**Oleh:**

**MUHAMMAD HIFZHAN SILMI**  
**TI-2A**  
**NIM. 1841720159**


**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**APRIL 2020**

a. Konfigurasi Database dan Pembuatan Tabel di MySQL

Langkah	Keterangan
1	<p>Buatlah project Laravel baru dengan nama laravel-crud. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud</pre> <pre>[[base] Akhdans-MacBook-Air:www akhdanhifzhan\$ composer create-project --prefer-dist laravel/laravel laravel-crud Creating a "laravel/laravel" project at "./laravel-crud" Installing laravel/laravel (v7.3.0) - Installing laravel/laravel (v7.3.0): Loading from cache Created project in /Applications/Amps/www/laravel-crud &gt; @php -r "file_exists('.env')    copy('.env.example', '.env');" Loading composer repositories with package information Updating dependencies (including require-dev) Package operations: 92 installs, 0 updates, 0 removals - Installing voku/portable-ascii (1.4.10): Loading from cache - Installing symfony/polyfill-ctype (v1.15.0): Loading from cache - Installing phpoption/phpoption (1.7.3): Loading from cache - Installing vlucas/phpdotenv (v4.1.4): Downloading (100%) - Installing symfony/css-selector (v5.0.7): Loading from cache - Installing tijsverkoyen/css-to-inline-styles (2.2.2): Loading from cache - Installing symfony/polyfill-mbstring (v1.15.0): Loading from cache - Installing symfony/var-dumper (v5.0.7): Loading from cache - Installing symfony/routing (v5.0.7): Loading from cache</pre>
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.</p> <pre> 1 APP_NAME=Laravel 2 APP_ENV=local 3 APP_KEY=base64:8zaqgvAUtkFgRAAwL2+fZyrLLpPhPyrTvGFvYP8BsYU= 4 APP_DEBUG=true 5 APP_URL=http://localhost 6 7 LOG_CHANNEL=stack 8 9 DB_CONNECTION=mysql 10 DB_HOST=localhost 11 DB_PORT=3306 12 DB_DATABASE=_latihan_laravel 13 DB_USERNAME=root 14 DB_PASSWORD= 15</pre> <p>nama database yang akan digunakan adalah <b>latihan_laravel</b> dengan username <b>root</b>.</p>
3	<p>Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database latihan_laravel.</p>

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	<b>id</b>	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>nama</b>	varchar(100)	utf8_unicode_ci		No	None		
<input type="checkbox"/> 3	<b>nim</b>	varchar(10)	utf8_unicode_ci		No	None		
<input type="checkbox"/> 4	<b>email</b>	varchar(50)	utf8_unicode_ci		No	None		
<input type="checkbox"/> 5	<b>jurusan</b>	varchar(20)	utf8_unicode_ci		No	None		

4	<p>isilah beberapa data pada tabel mahasiswa tersebut.</p> 
---	---

## b. Menampilkan Data dari Database

Langkah	Keterangan
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan. Pertama, buatlah route pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p> <pre> &lt;?php  use Illuminate\Support\Facades\Route;  /*  -----   Web Routes  -----     Here is where you can register web routes for your application. These   routes are loaded by the RouteServiceProvider within a group which   contains the "web" middleware group. Now create something great!   */  // Route::get('/', function () { //     return view('welcome'); // });  Route::get('/', 'MahasiswaController@index');</pre> <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>  <b>php artisan make:controller MahasiswaController</b></p> <pre>[(base) Akhdans-MacBook-Air:laravel-crud akhdanhifzhan\$ php artisan make:controller MahasiswaController Controller created successfully.</pre>
3	<p>Buat method index pada MahasiswaController.php pada folder <b>app/Http/Controllers</b></p> <pre>&lt;?php  namespace App\Http\Controllers;  use Illuminate\Http\Request; use Illuminate\Queue\Connectors\DatabaseConnector; use Illuminate\Support\Facades\DB;  class MahasiswaController extends Controller {     //     public function index()     {         //mengambil data dari tabel mahasiswa         \$mahasiswa = DB::table('mahasiswa')-&gt;get();          //mengirim data mahasiswa ke view index         return view('index',['mahasiswa' =&gt; \$mahasiswa]);     } }</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Tambahkan 'use Illuminate\Support\Facades\DB;' (line 6) agar query builder dapat digunakan</li> <li>- Line 13 untuk mengambil data dari tabel mahasiswa menggunakan query builder laravel dan akan disimpan di variabel \$mahasiswa</li> <li>- Line 16 : data akan dikirim ke blade view bernama index</li> </ul>
4	<p>Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat template blade (seperti file header-footer pada pembahasan CI) dengan nama master.blade.php pada folder resources/views.</p>

```
resources > views > master.blade.php > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css">
7   <title> @yield('title') </title>
8 </head>
9 <body>
10   <div class="container">
11     <div class="row mt-3">
12       <div class="col-md-6">
13         <div class="card">
14           <div class="card-header text-header">
15             <!-- bagian judul halaman -->
16             <h2> @yield('judul_halaman') </h2>
17           </div>
18           <div class="card-body">
19             <!-- bagian konten blog -->
20             <h2> @yield('konten') </h2>
21           </div>
22         </div>
23       </div>
24     </div>
25   </div>
26 </body>
27 </html>
```

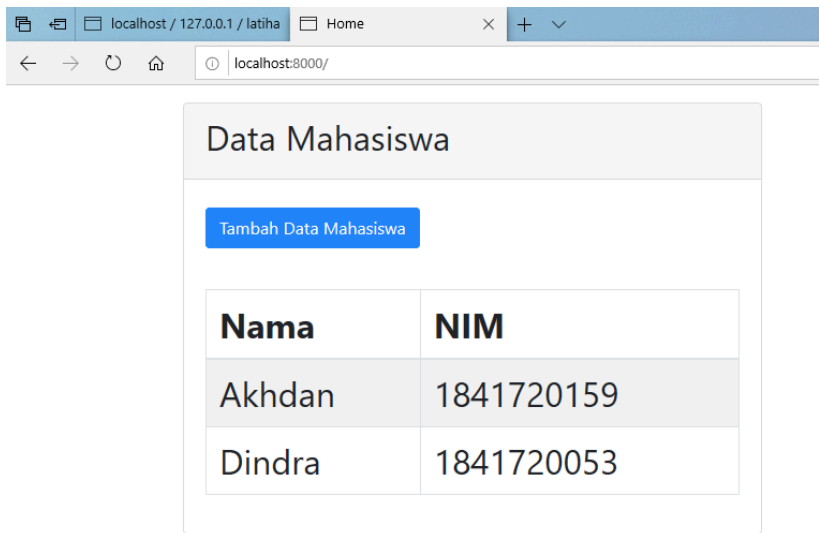
Keterangan:

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

5

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

```
resources > views > index.blade.php > ...
1 @extends('master')
2
3 <!-- isi title -->
4 @section('title', 'Home')
5
6 <!-- isi bagian judul halaman -->
7 @section('judul_halaman', 'Data Mahasiswa')
8
9 <!-- isi bagian konten -->
10 @section('konten')
11   <a href="/mahasiswa/tambah" class="btn btn-primary">Tambah Data Mahasiswa</a>
12   <br>
13   <br>
14   <table class="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
```

	<p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan blade lain yaitu master.blade.php</li> <li>- Line 4 : @section('title', 'Home') berarti bahwa file index mengisi @yield('title') pada master dengan 'Home'</li> <li>- Line 7 : @section('judul_halaman', 'Data Mahasiswa) berarti bahwa file index mengisi @yield('judul_halaman) pada master dengan 'Home'</li> <li>- Line 10-30 : mengisi yield(@konten), karena terdapat banyak baris pada diawali dengan @section('konten') dan diakhiri dengan @endsection</li> <li>- Line 24-25 menampilkan data mahasiswa dengan kolom nama dan nim</li> </ul>
6	<p>Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud</p> <p><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 route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController</p> <pre>Route::get('/mahasiswa/tambah','MahasiswaController@tambah');</pre>

2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre data-bbox="411 309 847 506"> public function tambah() {     //memanggil view tambah     return view('tambah'); } </pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. View tambah juga mengaplikasikan master.blade.php.</p> <pre data-bbox="411 696 1477 1509"> resources &gt; views &gt; tambah.blade.php 1  @extends('master') 2 3  &lt;!-- isi title --&gt; 4  @section('title', 'Tambah Data') 5 6  &lt;!-- isi bagian judul halaman --&gt; 7  @section('judul_halaman', 'Tambah 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     &lt;form action="/mahasiswa/simpan" method="post"&gt; 15         &lt;div class="form-group"&gt; 16             &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 17             &lt;input type="text" class="form-control" required="required" name="namamhs"&gt; &lt;br&gt; 18         &lt;/div&gt; 19         &lt;div class="form-group"&gt; 20             &lt;label for="nimhs"&gt;NIM&lt;/label&gt; 21             &lt;input type="text" class="form-control" required="required" name="nimhs"&gt; &lt;br&gt; 22         &lt;/div&gt; 23         &lt;div class="form-group"&gt; 24             &lt;label for="emailmhs"&gt;E-mail&lt;/label&gt; 25             &lt;input type="text" class="form-control" required="required" name="emailmhs"&gt; &lt;br&gt; 26         &lt;/div&gt; 27         &lt;div class="form-group"&gt; 28             &lt;label for="jurusanmhs"&gt;Jurusan&lt;/label&gt; 29             &lt;input type="text" class="form-control" required="required" name="jurusanmhs"&gt; &lt;br&gt; 30         &lt;/div&gt; 31         &lt;button type="submit" name="tambah" class="btn btn-primary float-right"&gt;Tambah Data&lt;/button&gt; 32     &lt;/form&gt; 33 @endsection </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 13-32 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan</li> <li>- Line 13 terdapat action="/mahasiswa/simpan" yang menunjukkan routes /mahasiswa/simpan dimana data pada form tersebut akan dikirimkan ke fungsi simpan pada controller MahasiswaController</li> <li>- 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.</li> </ul>

4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>Route::post('/mahasiswa/simpan','MahasiswaController@simpan');</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <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 simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre>public function simpan(Request \$request) {     //insert data ke table mahasiswa     DB::table('mahasiswa')-&gt;insert([         'nama' =&gt; \$request-&gt;namamhs,         'nim' =&gt; \$request-&gt;nimmhs,         'email' =&gt; \$request-&gt;emailmhs,         'jurusan' =&gt; \$request-&gt;jurusanmhs     ]);     return redirect('/mahasiswa'); }</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li> <li>- Line 28-33 merupakan query builder untuk insert data ke tabel mahasiswa</li> </ul>
6	<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> <div data-bbox="427 1357 887 2047"> <div>Tambah Data Mahasiswa</div> <div>Kembali</div> <div>Nama</div> <div>Shindid</div> <div>NIM</div> <div>1841720172</div> <div>E-mail</div> <div>shindid@gmail.com</div> <div>Jurusan</div> <div>Bisnis Manajemen</div> <div>Tambah Data</div> </div>



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

Data Mahasiswa	
<div>Tambah Data Mahasiswa</div>	
Nama	NIM
Akhdan	1841720159
Dindra	1841720053
Shindid	1841720172

#### d. Melihat detail data dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk melihat detail data (Line 28) pada file index.blade.php di folder resources/views</p> <pre> &lt;table class="table table-bordered table-hover table-striped"&gt;   &lt;thead&gt;     &lt;tr&gt;       &lt;th&gt;Nama&lt;/th&gt;       &lt;th&gt;NIM&lt;/th&gt;     &lt;/tr&gt;   &lt;/thead&gt;   &lt;tbody&gt;     @foreach(\$mahasiswa as \$mhs)       &lt;tr&gt;         &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt;         &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt;         &lt;td&gt;           &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt;         &lt;/td&gt;       &lt;/tr&gt;     @endforeach   &lt;/tbody&gt; &lt;/table&gt; @endsection </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre> Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); </pre>

3	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p> <pre> public function detail(\$id) {     //mengambil data mahasiswa berdasarkan id yang dipilih     \$mahasiswa = DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;get();     //kirim data mahasiswa yang diambil ke view edit.blade.php     return view(['detail',['mahasiswa' =&gt; \$mahasiswa]]; } </pre>
4	<p>Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php.</p> <pre> resources &gt; views &gt; detail.blade.php 1  @extends('master') 2 3  &lt;!-- isi title --&gt; 4  @section('title', 'Tambah Data') 5 6  &lt;!-- isi bagian judul halaman --&gt; 7  @section('judul_halaman', 'Tambah 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     @foreach(\$mahasiswa as \$mhs) 16     &lt;h5 class="card-title"&gt;{{ \$mhs-&gt;nama }}&lt;/h5&gt; 17     &lt;p class="card-text"&gt; 18         &lt;label for=""&gt;&lt;b&gt;NIM :&lt;/b&gt;&lt;/label&gt; 19         {{ \$mhs-&gt;nim }} 20     &lt;/p&gt; 21     &lt;p class="card-text"&gt; 22         &lt;label for=""&gt;&lt;b&gt;E-mail :&lt;/b&gt;&lt;/label&gt; 23         {{ \$mhs-&gt;email }} 24     &lt;/p&gt; 25     &lt;p class="card-text"&gt; 26         &lt;label for=""&gt;&lt;b&gt;Jurusan :&lt;/b&gt;&lt;/label&gt; 27         {{ \$mhs-&gt;jurusan }} 28     &lt;/p&gt; 29     @endforeach 30 @endsection </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>

### Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	Action
Akhdan	1841720159	<a href="#">Detail</a>
Dindra	1841720053	<a href="#">Detail</a>
Shindid	1841720172	<a href="#">Detail</a>

### Tambah Data Mahasiswa

Kembali

Akhdan

**NIM :** 1841720159

**E-mail :** akhdanx@gmail.com

**Jurusan :** Teknik Informatika

e. Mengubah data (update) dari database

Langkah	Keterangan
1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views</p> <pre> &lt;tbody&gt;   @foreach(\$mahasiswa as \$mhs)     &lt;tr&gt;       &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt;       &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt;       &lt;td&gt;         &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt;         &lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-warning"&gt;Edit&lt;/a&gt;       &lt;/td&gt;     &lt;/tr&gt;   @endforeach &lt;/tbody&gt; </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre> Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); </pre>

3	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre> public function edit(\$id) {     //mengambil data mahasiswa berdasarkan id yang dipilih     \$mahasiswa = DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;get();     //kirim data mahasiswa yang diambil ke view edit.blade.php     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 edit.blade.php yang berisi form untuk mengubah data pada folder resources/views. View edit juga mengaplikasikan master.blade.php.</p> <pre> resources &gt; views &gt; edit.blade.php 1  @extends('master') 2 3  &lt;!-- isi title --&gt; 4  @section('title', 'Edit Data') 5 6  &lt;!-- isi bagian judul halaman --&gt; 7  @section('judul_halaman', 'Edit 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     &lt;form action="/mahasiswa/update" method="post"&gt; 15         {{ csrf_field() }} 16         &lt;input type="hidden" name="id" value="{{ \$mhs-&gt;id }}"&gt;&lt;br&gt; 17         &lt;div class="form-group"&gt; 18             &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 19             &lt;input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mhs-&gt;nama }}"&gt; &lt;br&gt; 20         &lt;/div&gt; 21         &lt;div class="form-group"&gt; 22             &lt;label for="nimhs"&gt;NIM&lt;/label&gt; 23             &lt;input type="text" class="form-control" required="required" name="nimhs" value="{{ \$mhs-&gt;nim }}"&gt; &lt;br&gt; 24         &lt;/div&gt; 25         &lt;div class="form-group"&gt; 26             &lt;label for="emailmhs"&gt;E-mail&lt;/label&gt; 27             &lt;input type="text" class="form-control" required="required" name="emailmhs" value="{{ \$mhs-&gt;email }}"&gt; &lt;br&gt; 28         &lt;/div&gt; 29         &lt;div class="form-group"&gt; 30             &lt;label for="jurusanmhs"&gt;Jurusan&lt;/label&gt; 31             &lt;input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mhs-&gt;jurusan }}"&gt; &lt;br&gt; 32         &lt;/div&gt; 33         &lt;button type="submit" name="edit" class="btn btn-primary float-right"&gt;Simpan Data&lt;/button&gt; 34     &lt;/form&gt; 35 @endsection </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 /mahasiswa/update. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p>

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

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

6

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

```
public function update(Request $request)
{
    //insert data ke table mahasiswa
    DB::table('mahasiswa')->where('id',$request->id)->update([
        '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 58-63 merupakan query builder untuk update data ke tabel mahasiswa

7

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

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	Action
Akhdan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a>
Dindra	1841720053	<a href="#">Detail</a> <a href="#">Edit</a>
Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a>

Klik edit di data pertama, kita akan mengubah namanya.

## Edit Data Mahasiswa

[Kembali](#)

Nama

Akhdan Hifzhan

NIM

1841720159

E-mail

akhdanx@gmail.com

Jurusan

Teknik Informatika

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

## Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM	Action
Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a>
Dindra	1841720053	<a href="#">Detail</a> <a href="#">Edit</a>
Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a>

## f. Menghapus Data (delete) dari Database

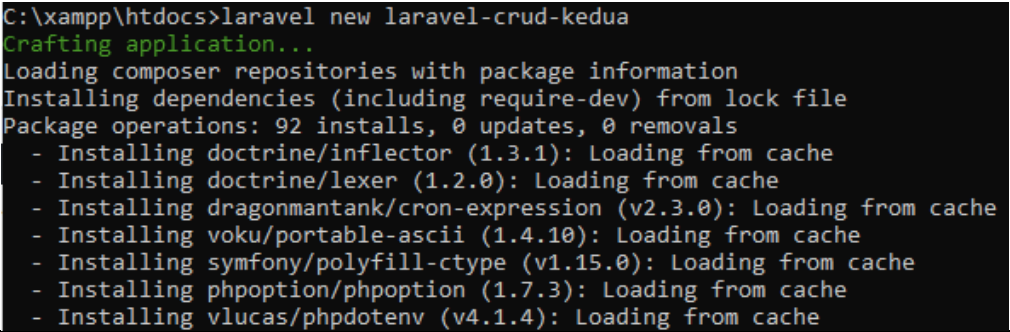
Langkah	Keterangan
1	<p>Buatlah tombol untuk menghapus data (Line 30) pada file index.blade.php di folder resources/views</p> <pre> &lt;tbody&gt;   @foreach(\$mahasiswa as \$mhs)     &lt;tr&gt;       &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt;       &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt;       &lt;td&gt;         &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt;         &lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-warning"&gt;Edit&lt;/a&gt;         &lt;a href="/mahasiswa/hapus/{{ \$mhs-&gt;id }}" class="badge badge-warning"&gt;Hapus&lt;/a&gt;       &lt;/td&gt;     &lt;/tr&gt;   @endforeach &lt;/tbody&gt; </pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); </pre>
3	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p> <pre> public function hapus(\$id) {     //menghapus data mahasiswa berdasarkan id yang dipilih     DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;delete();      return redirect('/mahasiswa'); } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 67 : query builder untuk menghapus data mahasiswa berdasarkan id yang dipilih</li> </ul>
4	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir.</p>

	<div> <div>Data Mahasiswa</div> <div>Tambah Data Mahasiswa</div> <table> <tr> <th>Nama</th><th>NIM</th><th>Action</th></tr> <tr> <td>Akhdan Hifzhan</td><td>1841720159</td><td> <a href="#">Detail</a>  <a href="#">Edit</a>  <a href="#">Hapus</a> </td></tr> <tr> <td>Dindra</td><td>1841720053</td><td> <a href="#">Detail</a>  <a href="#">Edit</a>  <a href="#">Hapus</a> </td></tr> <tr> <td>Shindid</td><td>1841720172</td><td> <a href="#">Detail</a>  <a href="#">Edit</a>  <a href="#">Hapus</a> </td></tr> </table> </div>	Nama	NIM	Action	Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	Dindra	1841720053	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	<div> <div>Data Mahasiswa</div> <div>Tambah Data Mahasiswa</div> <table> <tr> <th>Nama</th><th>NIM</th><th>Action</th></tr> <tr> <td>Akhdan Hifzhan</td><td>1841720159</td><td> <a href="#">Detail</a>  <a href="#">Edit</a>  <a href="#">Hapus</a> </td></tr> <tr> <td>Shindid</td><td>1841720172</td><td> <a href="#">Detail</a>  <a href="#">Edit</a>  <a href="#">Hapus</a> </td></tr> </table> </div>	Nama	NIM	Action	Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
Nama	NIM	Action																					
Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																					
Dindra	1841720053	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																					
Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																					
Nama	NIM	Action																					
Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																					
Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																					

## Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

Eloquent adalah sebuah fitur untuk mengelola data yang ada pada database. Eloquent ORM menyediakan fungsi-fungsi active record, atau fungsi-fungsi query sql untuk mengelola data pada database. Dengan menggunakan Eloquent pada Laravel, kita bisa mengelola data pada database dari hanya satu buah model.

### a. Konfigurasi Database

Langkah	Keterangan
1	<p>Buatlah project Laravel baru dengan nama laravel-crud-kedua. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud-kedua</pre> 
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.</p>




```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:ISukQLehe+CEZzitI1Q/OTygPmvUzyAmYMhmp02Q0Qg=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=localhost
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

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

#	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)	utf8_unicode_ci		No	None		
<input type="checkbox"/> 3	nim	varchar(10)	Unicode, case-insensitive		No	None		
<input type="checkbox"/> 4	email	varchar(50)	utf8_unicode_ci		No	None		
<input type="checkbox"/> 5	jurusan	varchar(20)	utf8_unicode_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 route pada routes/web.php 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> <pre> cd laravel-crud-kedua php artisan make:model Mahasiswa </pre>

	<pre>C:\xampp\htdocs\laravel-crud-kedua&gt;php artisan make:model Mahasiswa Model created successfully.</pre>
3	<p>Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.</p> <pre>&lt;?php  namespace App;  use Illuminate\Database\Eloquent\Model;  class Mahasiswa extends Model {     //     protected \$table = "mahasiswa"; }</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>C:\xampp\htdocs\laravel-crud-kedua&gt;php artisan make:controller MahasiswaController Controller created successfully.</pre>
5	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p> <pre>&lt;?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class MahasiswaController extends Controller {     public function index()     {         \$mahasiswa = Mahasiswa::all();         return view('index', ['mahasiswa' =&gt; \$mahasiswa]);     } }</pre> <p>Keterangan :</p> <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 template blade (seperti pada Bagian 1).</p> <p>Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)</p>

7

Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud

**php artisan serve**

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

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	Action
Akhdan Hifzhan	1841720159	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
Shindid	1841720172	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

**c. Memasukkan data(create) ke database**

Langkah	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre style="background-color: #2d2d2d; color: #c0c0c0; padding: 5px;">Route::get('/mahasiswa/tambah','MahasiswaController@tambah');</pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre style="background-color: #2d2d2d; color: #c0c0c0; padding: 5px;">public function tambah() {     return view('tambah'); }</pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views.</p> <p>Kita lakukan copy paste dari tambah.blade.php di Bagian 1</p>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p>

	<pre>Route::post('/mahasiswa/simpan','MahasiswaController@simpan');</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <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 simpan pada MahasiswaController 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('/mahasiswa'); }</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> <pre>class Mahasiswa extends Model {     //     protected \$table = "mahasiswa";     protected \$fillable = ['nama','nim','email','jurusan']; }</pre>
7	<p>Kembali coba jalankan localhost:8000 dan pilih tombol 'Tambah Data Mahasiswa', maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.</p>

## Tambah Data Mahasiswa

[Kembali](#)

Nama

Geraldny

NIM

1841725812

E-mail

geraldny@gmail.com

Jurusan

Teknik Elektro

[Tambah Data](#)

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

## Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM	Action
Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
Geraldny	1841725812	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

#### d. Melihat detail data dari database

Langkah	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre>Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');</pre>
2	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p> <pre>public function detail(\$id) {     \$mahasiswa = Mahasiswa::find(\$id);     return view('detail', ['mahasiswa' =&gt; \$mahasiswa]); }</pre>
3	<p>Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php.</p> <pre>resources &gt; views &gt; detail.blade.php 1  @extends('master') 2 3  &lt;!-- isi title --&gt; 4  @section('title', 'Detail Data') 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     &lt;h5 class="card-title"&gt;{{ \$mahasiswa-&gt;nama }}&lt;/h5&gt; 15     &lt;p class="card-text"&gt; 16         &lt;label for=""&gt;&lt;b&gt;NIM :&lt;/b&gt;&lt;/label&gt; 17         {{ \$mahasiswa-&gt;nim }} 18     &lt;/p&gt; 19     &lt;p class="card-text"&gt; 20         &lt;label for=""&gt;&lt;b&gt;E-mail :&lt;/b&gt;&lt;/label&gt; 21         {{ \$mahasiswa-&gt;email }} 22     &lt;/p&gt; 23     &lt;p class="card-text"&gt; 24         &lt;label for=""&gt;&lt;b&gt;Jurusan :&lt;/b&gt;&lt;/label&gt; 25         {{ \$mahasiswa-&gt;jurusan }} 26     &lt;/p&gt; 27 @endsection</pre>
4	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p> <div><p>Detail Data Mahasiswa</p><p>Kembali</p><p>Geraldny</p><p>NIM : 1841725812</p><p>E-mail : geraldny@gmail.com</p><p>Jurusan : Teknik Elektro</p></div>

**e. Mengubah data (update) dari database**

Langkah	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre>Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');</pre>
2	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre>public function edit(\$id) {     \$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 edit.blade.php yang berisi form untuk mengubah data pada folder resources/views. View edit juga mengaplikasikan master.blade.php.</p> <pre>resources &gt; views &gt; edit.blade.php 1 @extends('master') 2 3 &lt;!-- isi title --&gt; 4 @section('title', 'Edit Data') 5 6 &lt;!-- isi bagian judul halaman --&gt; 7 @section('judul_halaman', 'Edit 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 @foreach(\$mahasiswa as \$mhs) 15 &lt;form action="/mahasiswa/update" method="post"&gt; 16     {{ csrf_field() }} 17     &lt;input type="hidden" name="id" value="{{ \$mhs-&gt;id }}"&gt;&lt;br&gt; 18     &lt;div class="form-group"&gt; 19         &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 20         &lt;input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mhs-&gt;nama }}"&gt; &lt;br&gt; 21     &lt;/div&gt; 22     &lt;div class="form-group"&gt; 23         &lt;label for="nimhs"&gt;NIM&lt;/label&gt; 24         &lt;input type="text" class="form-control" required="required" name="nimhs" value="{{ \$mhs-&gt;nim }}"&gt; &lt;br&gt; 25     &lt;/div&gt; 26     &lt;div class="form-group"&gt; 27         &lt;label for="emailhs"&gt;E-mail&lt;/label&gt; 28         &lt;input type="text" class="form-control" required="required" name="emailhs" value="{{ \$mhs-&gt;email }}"&gt; &lt;br&gt; 29     &lt;/div&gt; 30     &lt;div class="form-group"&gt; 31         &lt;label for="jurusanhs"&gt;Jurusan&lt;/label&gt; 32         &lt;input type="text" class="form-control" required="required" name="jurusanhs" value="{{ \$mhs-&gt;jurusan }}"&gt; &lt;br&gt; 33     &lt;/div&gt; 34     &lt;button type="submit" name="edit" class="btn btn-primary float-right"&gt;Simpan Data&lt;/button&gt; 35 &lt;/form&gt; 36 @endforeach 37 @endsection</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> </ul>

- Line 11 terdapat action="/mahasiswa/update/{{\$mahasiswa->id}}}" yang menunjukkan routes /mahasiswa/update/{id} dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController

4

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

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

Keterangan:

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

5

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

```
public function update($id, Request $request)
{
    $mahasiswa = Mahasiswa::find($id);
    $mahasiswa->nama = $request->namamhs;
    $mahasiswa->nim = $request->nimmhs;
    $mahasiswa->email = $request->emailmhs;
    $mahasiswa->jurusan = $request->jurusanmhs;
    $mahasiswa->save();
    return redirect('/mahasiswa');
}
```

Keterangan:

- Variabel \$request untuk menerima data yang akan ditambahkan ke database

- Line 48-52 merupakan fungsi eloquent untuk update data ke tabel mahasiswa

6

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

Nama	NIM	Action
Akhdan Hifzhan	1841720159	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
Shindid	1841720172	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
Geraldny	1841725812	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
Aqil	123456789	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

setelah update->

Nama	NIM	Action
Akhdan Hifzhan	1841720159	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
Shindid	1841720172	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
Geraldny	1841725812	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
Aqil Hafizh	184172018	<div>Detail</div> <div>Edit</div> <div>Hapus</div>



**f. Menghapus data (delete) dari database**

Langkah	Keterangan																											
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre>Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');</pre>																											
2	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p> <pre>public function hapus(\$id) {     \$mahasiswa = Mahasiswa::find(\$id);     \$mahasiswa-&gt;delete();     return redirect('/mahasiswa'); }</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> <div><table><tr><th>Nama</th><th>NIM</th><th>Action</th></tr><tr><td>Akhdan Hifzhan</td><td>1841720159</td><td><a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a></td></tr><tr><td>Shindid</td><td>1841720172</td><td><a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a></td></tr><tr><td>Geraldly</td><td>1841725812</td><td><a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a></td></tr><tr><td>Aqil Hafizh</td><td>184172018</td><td><a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a></td></tr></table></div> <div><table><tr><th>Nama</th><th>NIM</th><th>Action</th></tr><tr><td>Akhdan Hifzhan</td><td>1841720159</td><td><a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a></td></tr><tr><td>Shindid</td><td>1841720172</td><td><a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a></td></tr><tr><td>Geraldly</td><td>1841725812</td><td><a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a></td></tr></table></div>	Nama	NIM	Action	Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	Geraldly	1841725812	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	Aqil Hafizh	184172018	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	Nama	NIM	Action	Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>	Geraldly	1841725812	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
Nama	NIM	Action																										
Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																										
Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																										
Geraldly	1841725812	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																										
Aqil Hafizh	184172018	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																										
Nama	NIM	Action																										
Akhdan Hifzhan	1841720159	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																										
Shindid	1841720172	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																										
Geraldly	1841725812	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>																										