

# **LAPORAN PRAKTIKUM PEMROGRAMAN WEBSITE LANJUT**

## **TUGAS PRAKTIKUM**

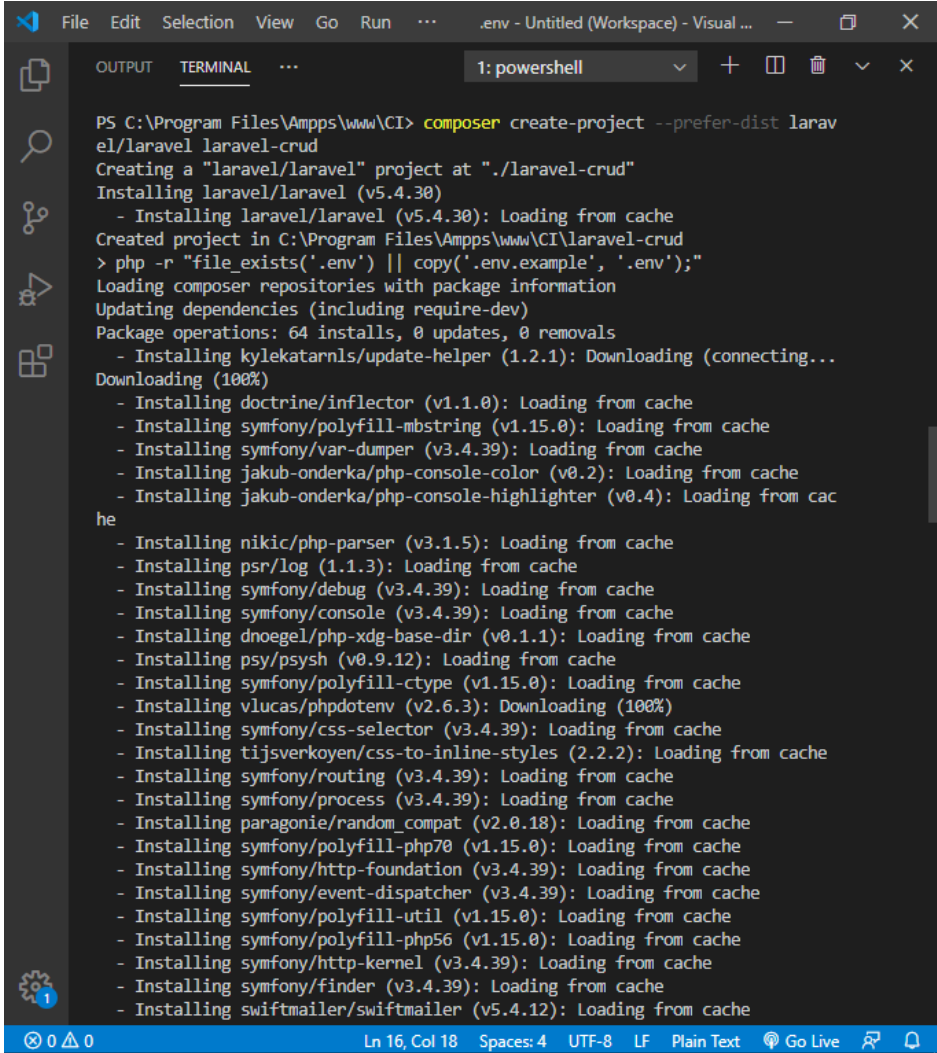
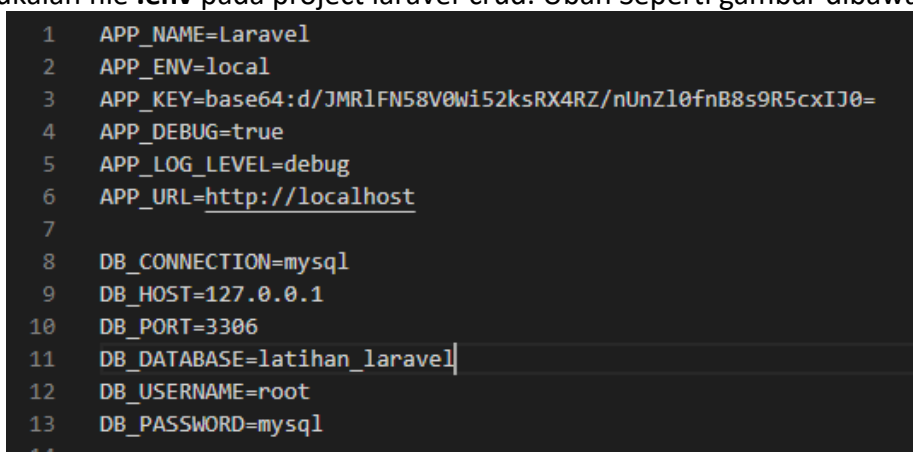


**Oleh :**

**M. WAHYU PRASETYO / NIM : 1841720177  
KELAS TI 2B**

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

a. Konfigurasi Database dan Pembuatan Tabel di MySQL

1	<p>Buatlah project Laravel baru dengan nama <b>laravel-crud</b>. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs composer create-project --prefer-dist laravel/laravel laravel-crud</pre> 
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file <b>.env</b> pada project laravel-crud. Ubah Seperti gambar dibawah ini.</p>  <pre>1 APP_NAME=Laravel 2 APP_ENV=local 3 APP_KEY=base64:d/JMRlFN58V0Wi52ksRX4RZ/nUnZl0fnB8s9R5cxIJ0= 4 APP_DEBUG=true 5 APP_LOG_LEVEL=debug 6 APP_URL=http://localhost 7 8 DB_CONNECTION=mysql 9 DB_HOST=127.0.0.1 10 DB_PORT=3306 11 DB_DATABASE=latihan_laravel 12 DB_USERNAME=root 13 DB_PASSWORD=mysql</pre> <p>Keterangan: Nama database yang akan digunakan adalah latihan_laravel dengan username root.</p>
3	<p>Jalankan xampp, selanjutnya buat tabel dengan nama <b>mahasiswa</b> di mysql pada database latihan_laravel.</p>

JelajahiStrukturSQLCariTambahkanEksporImporHak AksesOperasiTrigger

Struktur tabelTampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	UbahHapusLainnya
2	nama	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			UbahHapusLainnya
3	nim	varchar(10)	utf8mb4_general_ci		Tidak	Tidak ada			UbahHapusLainnya
4	email	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			UbahHapusLainnya
5	jurusan	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			UbahHapusLainnya

Pilih Semua

Dengan pilihan:

JelajahiUbahHapusUtamaUnikIndeksTeks penuh

4Isilah beberapa data pada tabel mahasiswa tersebut.

+ Opsi

idnama\_nim\_email\_jurusan

UbahSalinHapus

1nanda1001001001nanda@gmail.comTeknik Informatika

UbahSalinHapus

2wahyu1001001002wahyu@gmail.comTeknik Elektro

Pilih Semua

Dengan pilihan:

UbahSalinHapusEkspor

b. Menampilkan Data dari Database

1Setelah 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.

laravel-crud > routes > web.php

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 /*
6 |-----
7 | Web Routes
8 |-----
9 |
10 | Here is where you can register web routes for your application.
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great here.
13 |
14 */
15
16 // Route::get('/', function () {
17 //     return view('welcome');
18 // });
19
20 Route::get('/', 'MahasiswaController@index');
```

Keterangan:  
- Ketika route ("/") diakses, akan dijalankan method index pada controller bernama MahasiswaController.

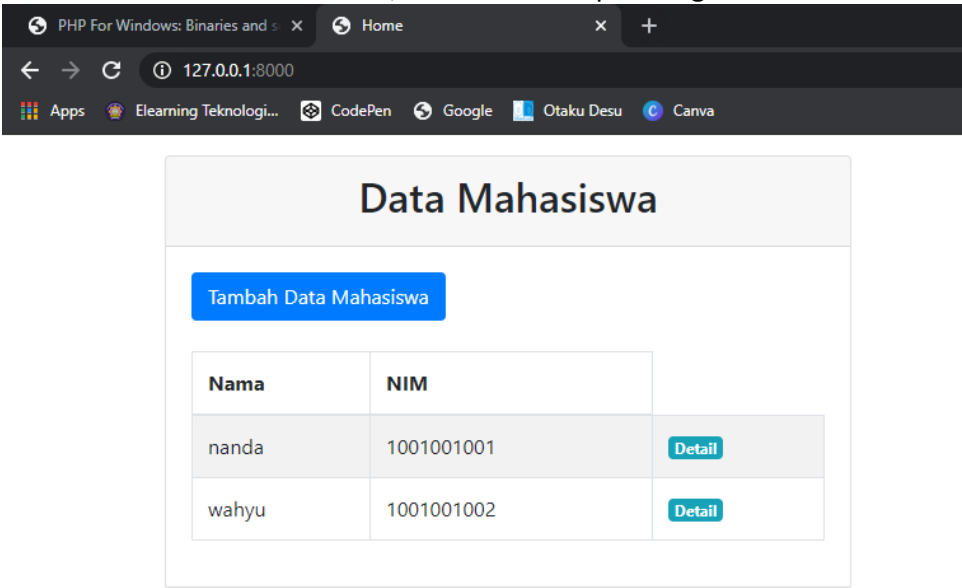
2Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan  
cd laravel-crud  
php artisan make:controller MahasiswaController

	<div><div>OUTPUTTERMINALDEBUG CONSOLEPROBLEMS1: powershell + □ ✕</div><div>Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.  PS C:\Program Files\Amp\www\CI\laravel-crud&gt; php artisan make:controller MahasiswaController Controller created successfully. PS C:\Program Files\Amp\www\CI\laravel-crud&gt;  </div></div>
3	<div>Buat method <b>index</b> pada MahasiswaController.php pada folder <b>app/Http/Controllers</b></div> <div>laravel-crud &gt; app &gt; Http &gt; Controllers &gt; MahasiswaController.php &gt; ... 1 &lt;?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 use Illuminate\Support\Facades\DB; 7 8 class MahasiswaController extends Controller 9 { 10     public function Index() 11     { 12         //mengambil data dari tabel mahasiswa 13         \$mahasiswa = DB::table('mahasiswa')-&gt;get(); 14 15         // mengirim ke view index 16         return view('index',['mahasiswa' =&gt; \$mahasiswa]); 17     } 18 }</div>
4	<div>Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama <b>index.blade.php</b>. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat <b>template blade</b> (seperti file <i>header-footer</i> pada pembahasan CI) dengan nama <b>master.blade.php</b> pada folder <b>resources/views</b>.</div> <div>laravel-crud &gt; resources &gt; views &gt; master.blade.php &gt; html &gt; body &gt; div.container &gt; div.row.mt-3 &gt; div.col-md-6 &gt; div.card &gt; ... 1 &lt;!DOCTYPE html&gt; 2 &lt;html lang="en"&gt; 3 &lt;head&gt; 4     &lt;meta charset="UTF-8"&gt; 5     &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt; 6     &lt;link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"&gt; 7     &lt;title&gt;@yield('title')&lt;/title&gt; 8 &lt;/head&gt; 9 &lt;body&gt; 10     &lt;div class="container"&gt; 11         &lt;div class="row mt-3"&gt; 12             &lt;div class="col-md-6"&gt; 13                 &lt;div class="card"&gt; 14                     &lt;div class="card-header text-center"&gt; 15                         &lt;!-- judul halaman --&gt; 16                         &lt;h2&gt;@yield('judul_halaman')&lt;/h2&gt; 17                     &lt;/div&gt; 18                     &lt;div class="card-body"&gt; 19                         &lt;!-- bagian konten --&gt; 20                         @yield('konten') 21                     &lt;/div&gt; 22                 &lt;/div&gt; 23             &lt;/div&gt; 24         &lt;/div&gt; 25     &lt;/div&gt; 26 &lt;/body&gt; 27 &lt;/html&gt;</div> <div>Keterangan:<ul style="list-style-type: none"><li>- Fungsi <b>@yield</b> pada line 6(title), 15(judul_halaman), dan 19(konten) berfungsi sebagai penanda bagian-bagian pada master blade. Nantinya bagian <b>@yield</b> ini akan diisi sesuai dengan halaman view yang menerapkan master.blade.php</li></ul></div>
5	<div>Sekarang buat file <b>index.blade.php</b> yang menerapkan <b>template</b> master.blade.php dan akan digunakan untuk menampilkan data.</div>

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

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

6 Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud  
**php artisan serve**  
Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut



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

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

```
laravel-crud > routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6  |-----
7  | Web Routes
8  |-----
9  |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 // Route::get('/', function () {
17 //     return view('welcome');
18 // });
19
20 Route::get('/', 'MahasiswaController@index');
21 Route::get('/', 'MahasiswaController@tambah');
```

2. Buat method **tambah** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menampilkan view tambah.

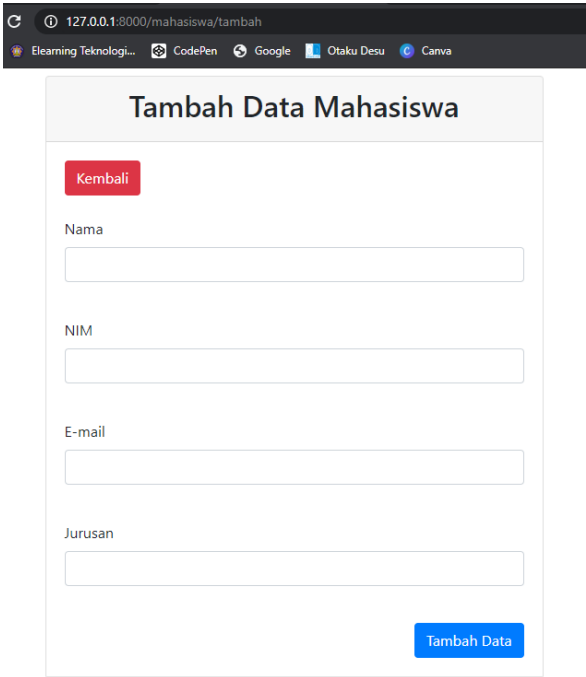
```
public function tambah(){
    //memanggil view tambah
    return view('tambah');
}
```

3. Kemudian buatlah view **tambah.blade.php** yang berisi form untuk memasukkan data baru pada folder **resources/views**. View tambah juga mengaplikasikan master.blade.php.

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

Keterangan:

- Line 13-32 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan

	<p>- Line 13 terdapat <code>action="/mahasiswa/simpan"</code> yang menunjukkan routes <code>/mahasiswa/simpan</code> dimana data pada form tersebut akan dikirimkan ke fungsi <code>simpan</code> pada controller <code>MahasiswaController</code></p> <p>- Line 14 terdapat <code>{{ csrf_field() }}</code> yang digunakan untuk menerapkan fitur laravel yaitu <code>csrf protection</code>. <code>csrf protection</code> adalah fitur keamanan untuk mencegah penginputan dari luar aplikasi, <code>csrf</code> akan men-generate kode token otomatis yang dibuat dalam bentuk <i>form hidden</i>.</p>
4	<p>Ketika tombol <code>simpan</code> ditekan, akan dipanggil routes <code>/mahasiswa/simpan</code>. 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 <code>post</code> karena data mahasiswa dari form akan dikirim ke method <code>simpan</code> di <code>MahasiswaController.php</code></li></ul>
5	<p>Buat method <code>simpan</code> pada <code>MahasiswaController</code> 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 <code>\$request</code> 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 <code>localhost:8000</code> dan pilih tombol <code>'Tambah Data Mahasiswa'</code>, maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.</p> <div></div> <p>Setelah kita klik tombol tambah data, maka data baru akan ditampilkan pada halaman index.</p>

127.0.0.1:8000/mahasiswa

AppsElearning Teknologi...CodePenGoogleOtaku DesuCanva

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda	1001001001	Detail
wahyu	1001001002	Detail
M. Wahyu Prasetyo	1841720177	Detail

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

d. Melihat Detail Data dari Database

1

Buatlah tombol untuk melihat detail data (Line 28) pada file **index.blade.php** di folder **resources/views**

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

2

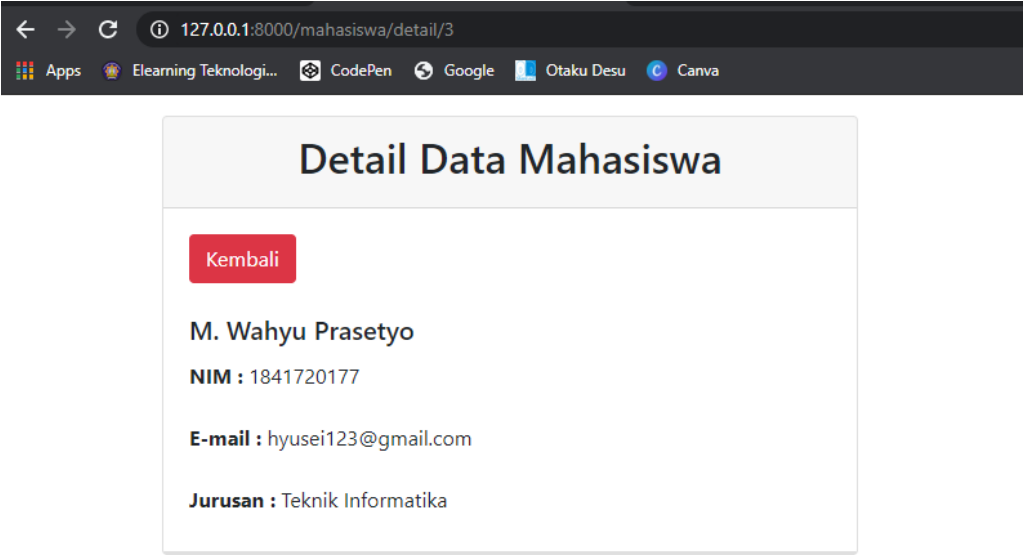
Buatlah **route** baru pada **routes/web.php** dengan nama **/mahasiswa/detail** yang akan menjalankan fungsi detail pada MahasiswaController

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

3

Buat method **detail** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menampilkan data mahasiswa pada view detail.blade.php.



	<pre>public function detail(\$id) {     \$mahasiswa = DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;get();     return view('detail',['mahasiswa' =&gt; \$mahasiswa]); }</pre>	
4	<p>Kemudian buatlah view <b>detail.blade.php</b> yang menampilkan detail data mahasiswa pada folder <b>resources/views</b>. View detail juga mengaplikasikan master.blade.php.</p> <pre>laravel-crud &gt; resources &gt; views &gt; detail.blade.php &gt; ... 1  @extends('master') 2 3  @section('title','Detail Mahasiswa') 4 5  @section('judul_halaman','Detail Data Mahasiswa') 6 7  @section('konten') 8      &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 9      &lt;br/&gt; 10     &lt;br/&gt; 11 12     @foreach(\$mahasiswa as \$mhs) 13         &lt;h5 class="card-title"&gt; {{ \$mhs-&gt;nama }} &lt;/h5&gt; 14         &lt;p class="card-text"&gt; 15             &lt;label for=""&gt;&lt;b&gt; NIM : &lt;/b&gt;&lt;/label&gt; 16             {{ \$mhs-&gt;nim }} &lt;/p&gt; 17         &lt;p class="card-text"&gt; 18             &lt;label for=""&gt;&lt;b&gt; E-mail : &lt;/b&gt;&lt;/label&gt; 19             {{ \$mhs-&gt;email }} &lt;/p&gt; 20         &lt;p class="card-text"&gt; 21             &lt;label for=""&gt;&lt;b&gt; Jurusan : &lt;/b&gt;&lt;/label&gt; 22             {{ \$mhs-&gt;jurusan }} &lt;/p&gt; 23     @endforeach 24 @endsection</pre> <p>Keterangan: - Line 16-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan</p>	
5	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p> 	

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

1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre>laravel-crud &gt; resources &gt; views &gt; index.blade.php &gt; table.table-bordered.table-hover.table-striped &gt; tbody &gt; tr &gt; td &gt; a &gt; 1 @extends('master') 2 3 @section('title','Home') 4 5 @section('judul_halaman','Data Mahasiswa') 6 7 @section('konten') 8 &lt;a href="/mahasiswa/tambah" class="btn btn-primary"&gt;Tambah Data Mahasiswa&lt;/a&gt; 9 &lt;br/&gt; 10 &lt;br/&gt; 11 &lt;table class="table table-bordered table-hover table-striped"&gt; 12 &lt;thead&gt; 13 &lt;tr&gt; 14 &lt;th&gt;Nama&lt;/th&gt; 15 &lt;th&gt;NIM&lt;/th&gt; 16 &lt;/tr&gt; 17 &lt;/thead&gt; 18 &lt;tbody&gt; 19 @foreach(\$mahasiswa as \$mhs) 20 &lt;tr&gt; 21 &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 22 &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 23 &lt;td&gt; 24 &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 25 &lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-warning"&gt;Edit&lt;/a&gt; 26 &lt;/td&gt; 27 &lt;/tr&gt; 28 @endforeach 29 &lt;/tbody&gt; 30 &lt;/table&gt; 31 @endsection</pre>
2	<p>Buatlah <b>route</b> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/edit</b> yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre>Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');</pre>
3	<p>Buat method <b>edit</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view edit.</p> <pre>public function edit(\$id) {     \$mahasiswa = DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;get();     return view('edit',['mahasiswa' =&gt; \$mahasiswa]); }</pre> <p>Keterangan :</p> <ul style="list-style-type: none"><li>- Line 49 : query builder untuk mengambil data mahasiswa berdasarkan id yang dipilih</li></ul>
4	<p>Kemudian buatlah view <b>edit.blade.php</b> yang berisi form untuk mengubah data pada folder <b>resources/views</b>. View edit juga mengaplikasikan master.blade.php.</p> <pre>laravel-crud &gt; resources &gt; views &gt; edit.blade.php &gt; ... 5 @section('judul_halaman','Edit Data Mahasiswa') 6 7 @section('konten') 8 &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 9 &lt;br/&gt; 10 &lt;br/&gt; 11 @foreach(\$mahasiswa as \$mhs) 12 &lt;form action="/mahasiswa/update" method="post"&gt; 13 &lt;input type="hidden" name="id" value="{{ \$mhs-&gt;id }}"&gt; &lt;br/&gt; 14 &lt;div class="form-grub"&gt; 15 &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 16 &lt;input type="text" class="form-control" required name="namamhs" value="{{ \$mhs-&gt;nama }}"&gt; &lt;br/&gt; 17 &lt;/div&gt; 18 &lt;div class="form-grub"&gt; 19 &lt;label for="nimhs"&gt;NIM&lt;/label&gt; 20 &lt;input type="text" class="form-control" required name="nimhs" value="{{ \$mhs-&gt;nim }}"&gt; &lt;br/&gt; 21 &lt;/div&gt; 22 &lt;div class="form-grub"&gt; 23 &lt;label for="emailhs"&gt;E-mail&lt;/label&gt; 24 &lt;input type="text" class="form-control" required name="emailhs" value="{{ \$mhs-&gt;email }}"&gt; &lt;br/&gt; 25 &lt;/div&gt; 26 &lt;div class="form-grub"&gt; 27 &lt;label for="jurusanhs"&gt;Jurusan&lt;/label&gt; 28 &lt;input type="text" class="form-control" required name="jurusanhs" value="{{ \$mhs-&gt;jurusan }}"&gt; &lt;br/&gt; 29 &lt;/div&gt; 30 &lt;button type="submit" name="edit" class="btn btn-primary float-right"&gt;Simpan Data&lt;/button&gt; 31 &lt;/form&gt; 32 @endforeach 33 @endsection</pre>

5

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

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

Keterangan:

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

6

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

```
public function update(Request $request)
{
    DB::table('mahasiswa')->where('id',$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.

127.0.0.1:8000/mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda	1001001001	<div>Detail Edit</div>
wahyu	1001001002	<div>Detail Edit</div>
M. Wahyu Prasetyo	1841720177	<div>Detail Edit</div>

### Edit Data Mahasiswa

Kembali

Nama

nanda

NIM

1001001001

E-mail

nanda@gmail.com

Jurusab

Teknik Informatika

Simpan Data

### Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nandadesu	1001001001	<div>DetailEdit</div>
wahyu	1001001002	<div>DetailEdit</div>
M. Wahyu Prasetyo	1841720177	<div>DetailEdit</div>

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

1	<p>Buatlah tombol untuk menghapus data (Line 30) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre>1 @extends('master') 2 3 @section('title','Home') 4 5 @section('judul_halaman','Data Mahasiswa') 6 7 @section('konten') 8 &lt;a href="/mahasiswa/tambah" class="btn btn-primary"&gt;Tambah Data Mahasiswa&lt;/a&gt; 9 &lt;br/&gt; 10 &lt;br/&gt; 11 &lt;table class="table table-bordered table-hover table-striped"&gt; 12     &lt;thead&gt; 13         &lt;tr&gt; 14             &lt;th&gt;Nama&lt;/th&gt; 15             &lt;th&gt;NIM&lt;/th&gt; 16         &lt;/tr&gt; 17     &lt;/thead&gt; 18     &lt;tbody&gt; 19         @foreach(\$mahasiswa as \$mhs) 20             &lt;tr&gt; 21                 &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 22                 &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 23                 &lt;td&gt; 24                     &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 25                     &lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-warning"&gt;Edit&lt;/a&gt; 26                     &lt;a href="/mahasiswa/hapus/{{ \$mhs-&gt;id }}" class="badge badge-danger"&gt;Hapus&lt;/a&gt; 27                 &lt;/td&gt; 28             &lt;/tr&gt; 29         @endforeach 30     &lt;/tbody&gt; 31 &lt;/table&gt; 32 @endsection</pre>
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> <pre>Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');</pre>
3	<p>Buat method <b>hapus</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menjalankan fungsi hapus.</p> <pre>public function hapus(\$id) {     DB::table('mahasiswa')-&gt;where('id',\$id)-&gt;delete();     return redirect('/mahasiswa'); }</pre>
4	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir.</p>

## Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nandadesu	1001001001	<div>DetailEditHapus</div>
wahyu	1001001002	<div>DetailEditHapus</div>
M. Wahyu Prasetyo	1841720177	<div>DetailEditHapus</div>

## Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nandadesu	1001001001	<div>DetailEditHapus</div>
wahyu	1001001002	<div>DetailEditHapus</div>

Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

a. Konfigurasi Database

1

Buatlah project Laravel baru dengan nama **laravel-crud-kedua**. Buka command prompt, tuliskan perintah berikut.  
**cd C:\xampp\htdocs**  
**laravel new laravel-crud-kedua**

2

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

.env

1

APP\_NAME=Laravel

2

APP\_ENV=local

3

APP\_KEY=base64:9S+rhXvW0gNGAPFMSaHh1nJn0YnxKt0GqeV8uTPBjho=

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=

15

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

Struktur tabel

Tampilan hubungan

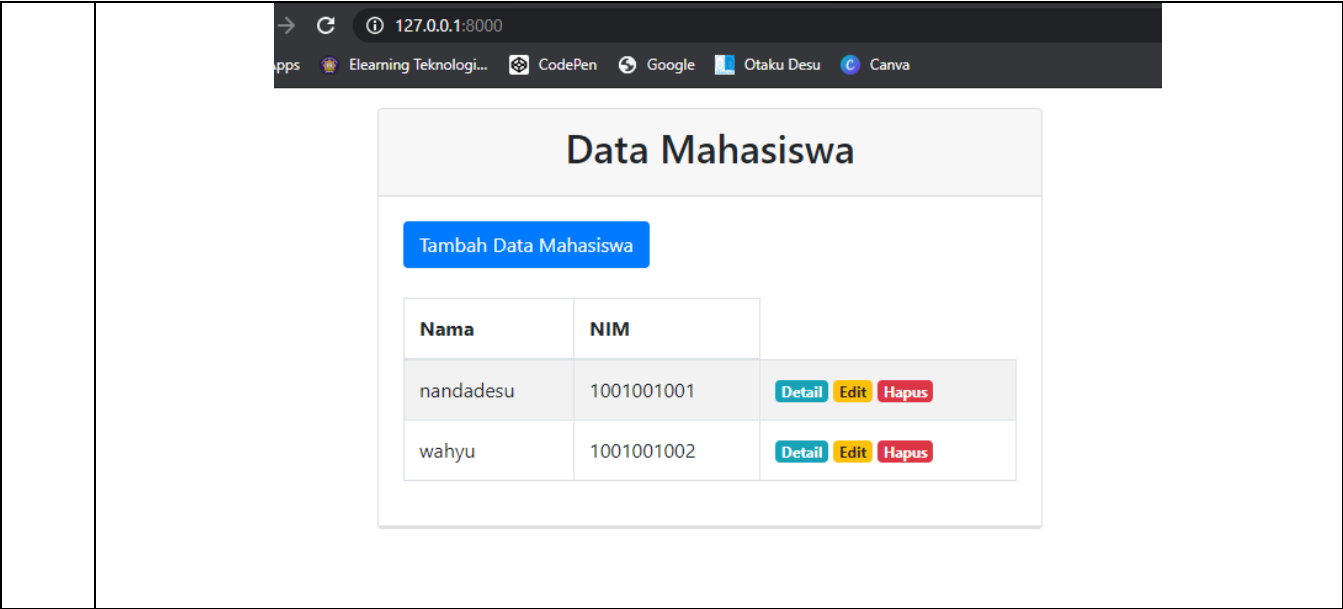
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	id	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah  Hapus  Lainnya
<input type="checkbox"/>	2	nama	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah  Hapus  Lainnya
<input type="checkbox"/>	3	nim	varchar(10)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah  Hapus  Lainnya
<input type="checkbox"/>	4	email	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah  Hapus  Lainnya
<input type="checkbox"/>	5	jurusan	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah  Hapus  Lainnya
<input type="checkbox"/>	6	created-at	timestamp		Ya	NULL			Ubah  Hapus  Lainnya
<input type="checkbox"/>	7	updated_at	timestamp		Ya	NULL			Ubah  Hapus  Lainnya

b. Menampilkan Data dari Database

1	Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan. Pertama, buatlah <i>route</i> pada <b>routes/web.php</b> sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data. <div><pre>20 Route::get('/', 'MahasiswaController@index'); 21 Route::get('/mahasiswa', 'MahasiswaController@index');</pre></div>
2	Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan <b>cd laravel-crud-kedua</b> <b>php artisan make:model Mahasiswa</b> <div><pre>PS C:\xampp\htdocs\laravel-crud-kedua&gt; php artisan make:model Mahasiswa Model created successfully. PS C:\xampp\htdocs\laravel-crud-kedua&gt;</pre></div>

3	<p>Ubah model <b>Mahasiswa.php</b> pada folder <b>App</b> menjadi seperti berikut.</p> <pre>app &gt; Mahasiswa.php &gt; ... 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 } 11</pre>
4	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan <b>php artisan make:controller MahasiswaController</b></p> <pre>PS C:\xampp\htdocs\laravel-crud-kedua&gt; php artisan make:controller MahasiswaC artisan make:controller MahasiswaController Controller created successfully. PS C:\xampp\htdocs\laravel-crud-kedua&gt; </pre>
5	<p>Buat method <b>index</b> pada MahasiswaController.php pada folder <b>app/Http/Controllers</b></p> <pre>1  &lt;?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     { 12         \$mahasiswa = Mahasiswa::all(); 13         return view('index', ['mahasiswa' =&gt; \$mahasiswa]); 14     } 15 } 16</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>njutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat <b>template blade</b> (seperti pada Bagian 1). Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)</p>
7	<p>Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud <b>php artisan serve</b> Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut</p>





c. Memasukkan Data (Create) ke Database

1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/tambah</b> yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre>Route::get('/mahasiswa/tambah','MahasiswaController@tambah');</pre>
2	<p>Buat method <b>tambah</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view tambah.</p> <pre>public function tambah() {     return view('tambah'); }</pre>
3	<p>Kemudian buatlah view <b>tambah.blade.php</b> yang berisi form untuk memasukkan data baru pada folder <b>resources/views</b>. 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>Route::post('/mahasiswa/simpan','MahasiswaController@simpan');</pre> <p>Keterangan:</p> <ul style="list-style-type: none"><li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php</li></ul>
5	<p>Buat method <b>simpan</b> pada <b>MahasiswaController</b> untuk menyimpan data ke database.</p> <pre>public function simpan(Request \$request) {     Mahasiswa::create([         'nama' =&gt; \$request-&gt;namamhs,         'nim' =&gt; \$request-&gt;nimmhs,         'email' =&gt; \$request-&gt;emailmhs,         'jurusan' =&gt; \$request-&gt;jurusanmhs     ]);     return redirect('/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

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.

```
class Mahasiswa extends Model
{
    protected $table = "mahasiswa";
    protected $fillable = ['nama', 'nim', 'email', 'jurusan'];
}
```

7

Kembali coba jalankan localhost:8000 dan pilih tombol ‘Tambah Data Mahasiswa’, maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.

127.0.0.1:8000/mahasiswa/tambah

Kembali

Nama

NIM

E-mail

Jurusan

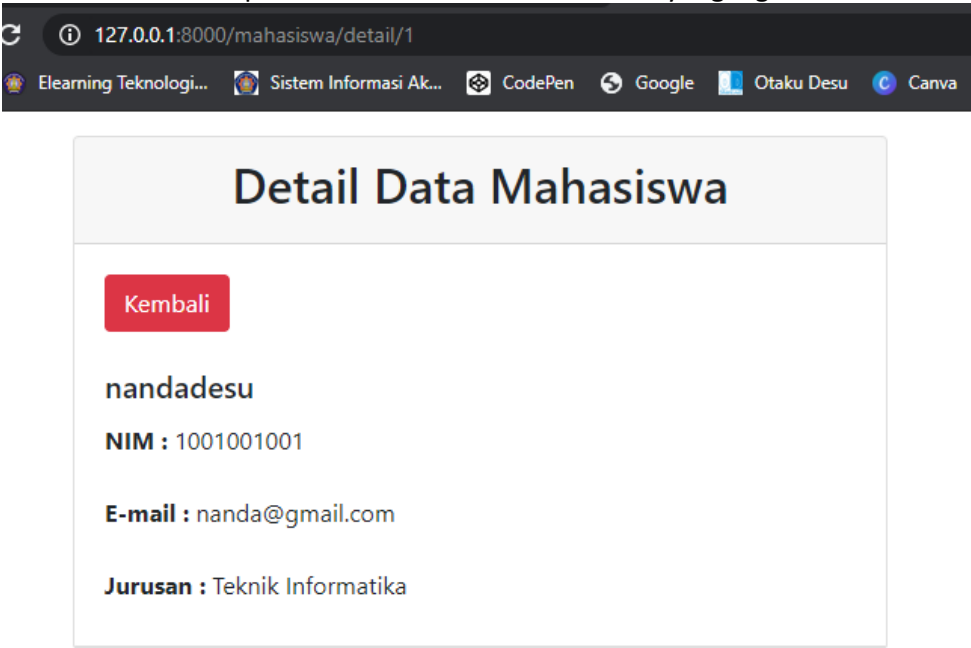
Tambah Data

127.0.0.1:8000/mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nandadesu	1001001001	<div>Detail Edit Hapus</div>
wahyu	1001001002	<div>Detail Edit Hapus</div>
M. Wahyu Prasetyo	1841720177	<div>Detail Edit Hapus</div>

#### d. Melihat Detail Data dari Database

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

1	<p>Buatlah <b>route</b> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/edit</b> yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre>Route::get('/mahasiswa/edit{id}','MahasiswaController@edit');</pre>
2	<p>Buat method <b>edit</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view edit.</p> <pre>public function edit(\$id) {     \$mahasiswa = Mahasiswa::find(\$id);     return view('edit',['mahasiswa' =&gt; \$mahasiswa]); }</pre> <p>Keterangan :</p> <ul style="list-style-type: none"><li>- Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih</li></ul>
3	<p>Kemudian buatlah view <b>edit.blade.php</b> yang berisi form untuk mengubah data pada folder <b>resources/views</b>. View edit juga mengaplikasikan master.blade.php.</p> <pre>@extends('master')  @section('title','Edit Data')  @section('judul_halaman', 'Edit Data Mahasiswa')  @section('konten') &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; &lt;br/&gt; &lt;br/&gt;  &lt;form action="/mahasiswa/update" method="post"&gt;     {{ csrf_field() }}     &lt;input type="hidden" name="id" value="{{ \$mahasiswa-&gt;id }}"&gt; &lt;br/&gt;     &lt;div class="form-grub"&gt;         &lt;label for="namamhs"&gt;Nama&lt;/label&gt;         &lt;input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mahasiswa-&gt;nama }}"&gt; &lt;br/&gt;     &lt;/div&gt;     &lt;div class="form-grub"&gt;         &lt;label for="nimhs"&gt;NIM&lt;/label&gt;         &lt;input type="number" class="form-control" required="required" name="nimhs" value="{{ \$mahasiswa-&gt;nim }}"&gt; &lt;br/&gt;     &lt;/div&gt;     &lt;div class="form-grub"&gt;         &lt;label for="emailhs"&gt;E-mail&lt;/label&gt;         &lt;input type="email" class="form-control" required="required" name="emailhs" value="{{ \$mahasiswa-&gt;email }}"&gt; &lt;br/&gt;     &lt;/div&gt;     &lt;div class="form-grub"&gt;         &lt;label for="jurusanhs"&gt;Jurusan&lt;/label&gt;         &lt;input type="text" class="form-control" required="required" name="jurusanhs" value="{{ \$mahasiswa-&gt;jurusan }}"&gt; &lt;br/&gt;     &lt;/div&gt;     &lt;button type="submit" name="edit" class="btn btn-primary float-right"&gt;Simpan Data&lt;/button&gt; &lt;/form&gt; @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><li>- Line 11 terdapat action="/mahasiswa/update/{{ \$mahasiswa-&gt;id }}" yang menunjukkan routes /mahasiswa/update/{id} dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController</li></ul>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/update/{id}</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>Route::post('mahasiswa/update/{id}','MahasiswaController@update');</pre> <p>Keterangan:</p> <ul style="list-style-type: none"><li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php</li></ul>
5	<p>Buat method <b>update</b> pada <b>MahasiswaController</b> untuk menyimpan data yang diubah ke database.</p>

```
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.

127.0.0.1:8000/mahasiswa/edit/1

Elearning Teknologi... Sistem Informasi Ak... CodePen Google Otaku Desu C

## Edit Data Mahasiswa

Kembali

Nama

nanda

NIM

1001001001

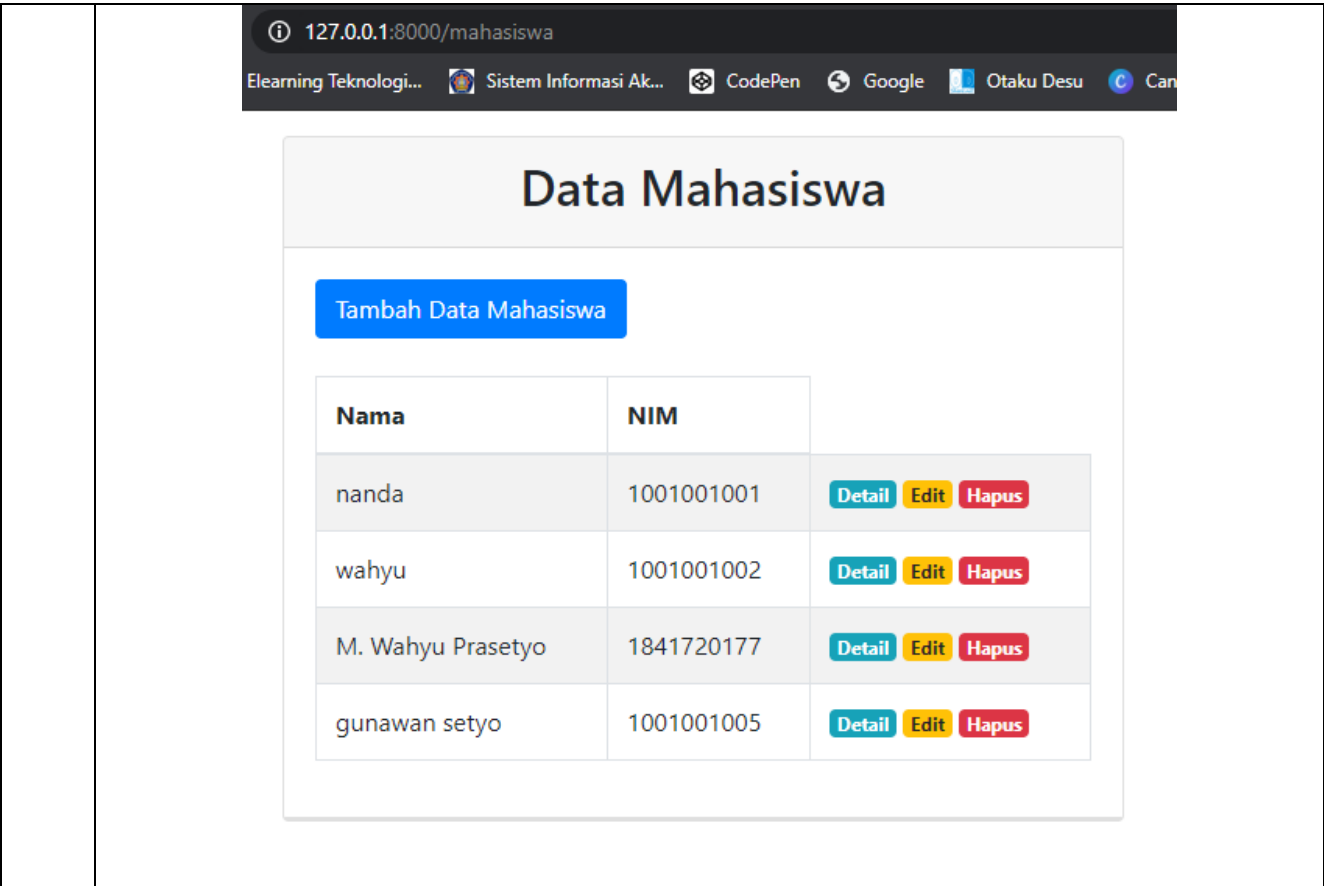
E-mail

nanda@gmail.com

Jurusab

Teknik Informatika

Simpan Data



f. Menghapus Data (Delete) dari Database

1	Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/hapus</b> yang akan menjalankan fungsi hapus pada MahasiswaController
2	Buat method <b>hapus</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menjalankan fungsi hapus.
3	Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.

## Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda	1001001001	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
wahyu	1001001002	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
M. Wahyu Prasetyo	1841720177	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
gunawan setyo	1001001005	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

## Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda	1001001001	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
wahyu	1001001002	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
M. Wahyu Prasetyo	1841720177	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>