

# **LAPORAN PRAKTIKUM**

## **Pemrograman Web Lanjut (PWL)**



**Oleh :**

**Osa Mahanani Sihono**

**(1841720066) TI**

**2B**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI POLITEKNIK**

**NEGERI MALANG**

**2020**

### **Topik**

Membuat CRUD – Create Read Update Delete Data dengan Framework Laravel

## Tujuan

Mahasiswa diharapkan dapat:

1. Memahami bagaimana memasukkan data baru ke basis data menggunakan Laravel
2. Memahami bagaimana menampilkan data dari basis data menggunakan Laravel
3. Memahami bagaimana mengubah data dari basis data menggunakan Laravel
4. Memahami bagaimana menghapus data dari basis data menggunakan Laravel

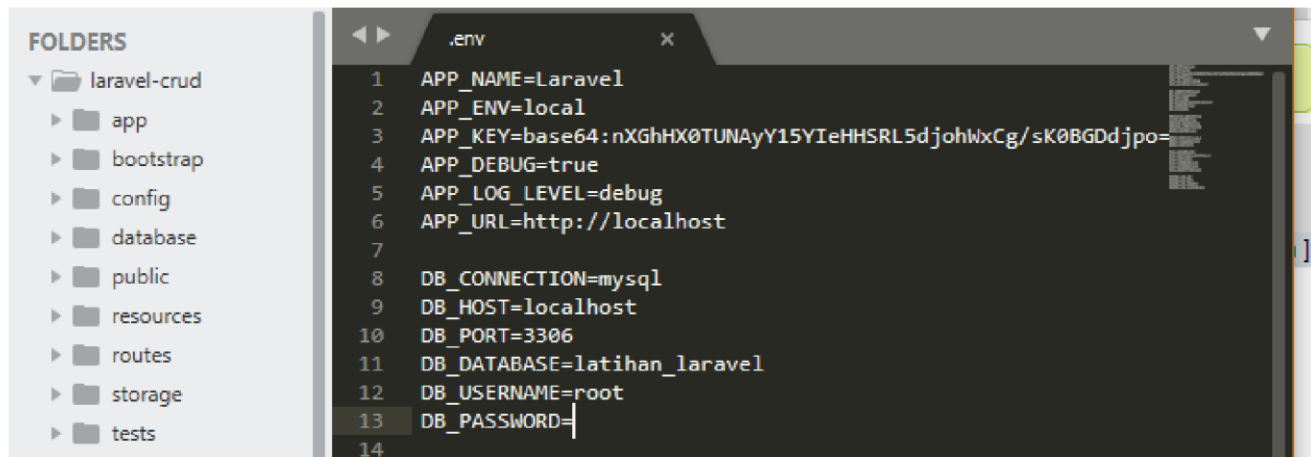
## Praktikum – Bagian 1: Membuat CRUD di Laravel menggunakan Query Builder

Query Builder adalah salah satu fitur untuk menjalankan *query database*. Laravel menyediakan fungsifungsi 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	Buatlah project Laravel baru dengan nama <b>laravel-crud</b> . Buka command prompt, tuliskan perintah berikut. <b>cd C:\xampp\htdocs laravel new laravel-crud</b>

2



```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:nXGhHX0TUNAY15YIeHHSRL5djohWxCg/sK0BGDdjpo=
4 APP_DEBUG=true
5 APP_LOG_LEVEL=debug
6 APP_URL=http://localhost
7
8 DB_CONNECTION=mysql
9 DB_HOST=localhost
10 DB_PORT=3306
11 DB_DATABASE=latihan_laravel
12 DB_USERNAME=root
13 DB_PASSWORD=
14
```

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

Keterangan:

- Nama database yang akan digunakan adalah `latihan_laravel` dengan username `root`.

Server: 127.0.0.1 » Database: latihan\_laravel » Table: mahasiswa

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[More](#)

[Table structure](#)
[Relation view](#)

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

☐ Check all
 With selected:
 [Browse](#)
 Change
 Drop
 Primary
 Unique
 Index

Add to central columns
 Remove from central columns

Jalankan xampp, selanjutnya buat tabel dengan nama **mahasiswa** di mysql pada database latihan\_laravel.

Sort by key: 

None

+ Options

id

nama

nim

email

jurusan

1

nanda

1001001001

nandan@gmail.com

teknik informatika

2

wahyu

1001001002

wahyu@gmail.com

teknik elektro

With selected:

Show all

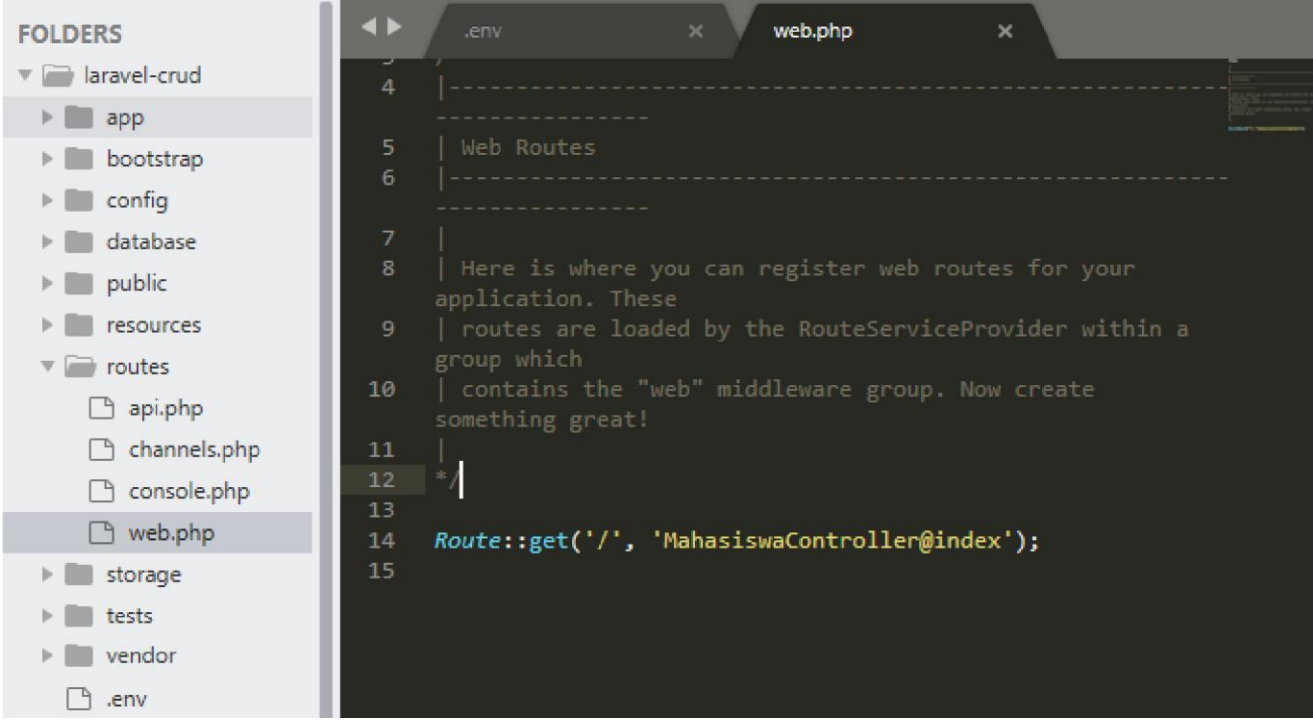
Number of rows: 

25

Filter rows: 

Search this table

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

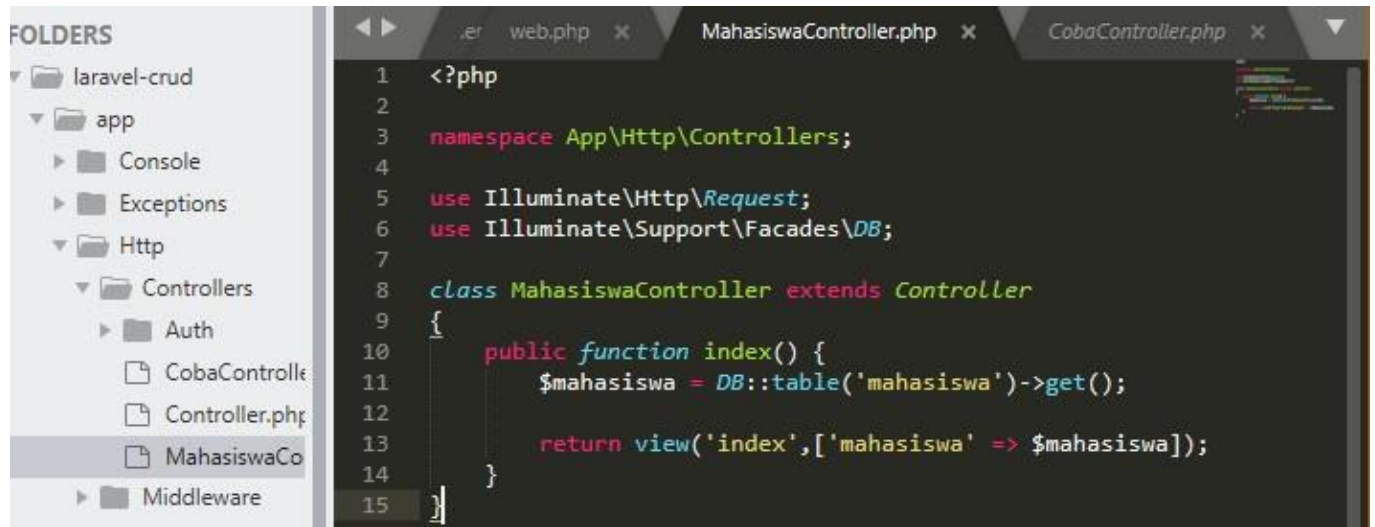
Langkah	Keterangan
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan.</p>  <p>Pertama, buatlah <i>route</i> pada <b>routes/web.php</b> sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan 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>

#### b. Menampilkan Data dari Database

2

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

3



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

Keterangan:

- Tambahkan 'use Illuminate\Support\Facades\DB;' (line 6) agar *query builder* dapat digunakan
- Line 13 untuk mengambil data dari tabel mahasiswa menggunakan *query builder* laravel dan akan disimpan di variabel `$mahasiswa`
- Line 16 : data akan dikirim ke blade view bernama index

```

2 <html>
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width,
7   initial-scale=1, shrink-to-fit=no">
8
9 <!-- Bootstrap CSS -->
10 <link rel="stylesheet" href="
11   https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/
12   bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GX
13   wEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
14   crossorigin="anonymous">
15 <title> @yield('title')</title>
16
17 </head>
18 <body>
19
20 <div class="container">
21   <div class="row mt-3">
22     <div class="col-md-6">
23       <div class="card">
24         <div class="card-header text-center">
25           <h2> @yield('judul_halaman')</h2>
26         </div>
27         <div class="card-body">
28           @yield('konten')
29         </div>
30       </div>
31     </div>
32   </div>
33 </div>
34 </body>
35 </html>

```

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**.



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

```

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><br>
10 <table class="table">
11     <thead class="thead-dark">
12         <th>Nama</th>
13         <th>NIM</th>
14         <th>Aksi</th>
15     </thead>
16     <tbody>
17         @foreach($mahasiswa as $mhs)
18         <tr>
19             <td>{{ $mhs->nama }}</td>
20             <td>{{ $mhs->nim }}</td>
21             <td>
22                 <a href="/mahasiswa/detail/{{ $mhs->id }}"
class="badge badge-info">Detail</a>
23                 <a href="/mahasiswa/edit/{{ $mhs->id }}" class
="badge badge-warning">Edit</a>
24                 <a href="/mahasiswa/hapus/{{ $mhs->id }}"
class="badge badge-danger" onclick="return
confirm('Apakah anda ingin menghapus data
ini?')">Hapus</a>
25             </td>
26         </tr>
27         @endforeach
28     </tbody>
29 </table>
30 @endsection

```

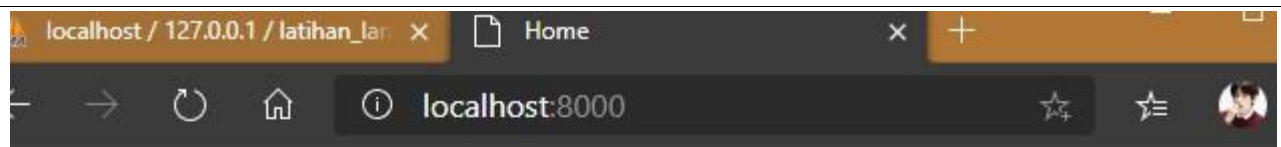
Sekarang buat fil

yang menerapkan *template* master.blade.php dan akan digunakan untuk menampilkan data.

Keterangan:

- Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan blade lain yaitu master.blade.ph
- Line 4 : @section('title', 'Home') berarti bahwa file index mengisi @yield('title') pada master dengan 'Hom
- Line 7 : @section('judul\_halaman', 'Data Mahasiswa) berarti bahwa file index mengisi @yield('judul\_hala dengan 'Home'
- Line 10-30 : mengisi yield(@konten), karena terdapat banyak baris pada diawali dengan @section('konte 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



Tambah Data Mahasiswa

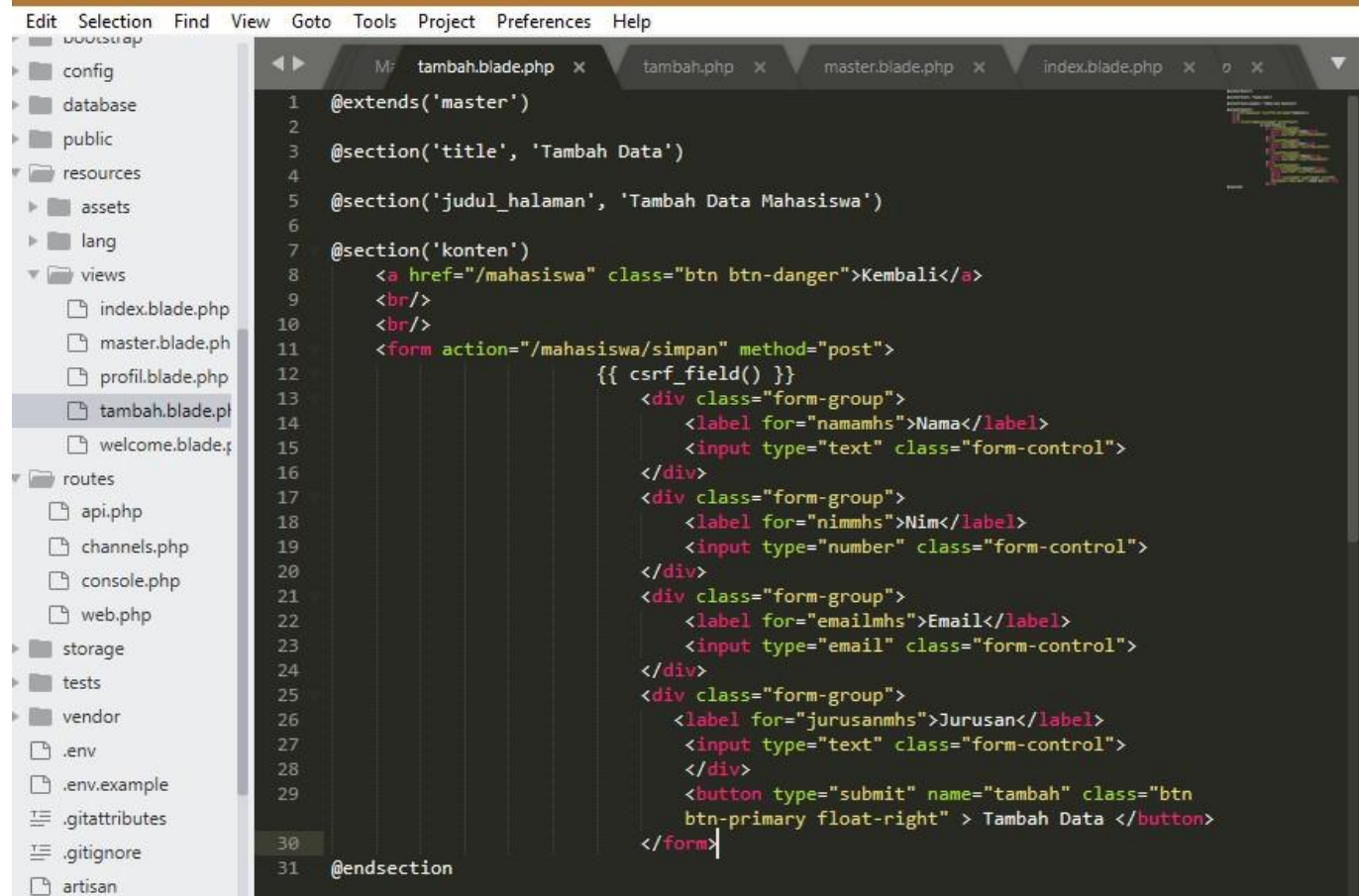
Nama	NIM
nanda	1001001001
wahyu	1001001002

### 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</p> <pre> 15 16 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah 17 '); </pre> <p>menjalankan fungsi tambah pada MahasiswaController</p>
2	<pre> 15 16 public function tambah() { 17     return view('tambah'); 18 } 19 </pre> <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

Kemudian buatlah view **tambah.blade.php** yang berisi form untuk memasukkan data baru pada folder **resources/views**. View tambah juga mengaplikasikan master.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" class="form-control">
16         </div>
17         <div class="form-group">
18             <label for="nimamhs">Nim</label>
19             <input type="number" class="form-control">
20         </div>
21         <div class="form-group">
22             <label for="emailmhs">Email</label>
23             <input type="email" class="form-control">
24         </div>
25         <div class="form-group">
26             <label for="jurusanmhs">Jurusan</label>
27             <input type="text" class="form-control">
28         </div>
29         <button type="submit" name="tambah" class="btn
30             btn-primary float-right" > Tambah Data </button>
31     </form>
32 @endsection
  
```

#### Keterangan:

- Line 13-32 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 13 terdapat `action="/mahasiswa/simpan"` yang menunjukkan routes `/mahasiswa/simpan` dimana data pada form tersebut akan dikirimkan ke fungsi `simpan` pada controller `MahasiswaController`
- Line 14 terdapat `{{ csrf_field() }}` yang digunakan untuk menerapkan fitur laravel yaitu csrf protection. csrf protection adalah fitur keamanan untuk mencegah penginputan dari luar aplikasi, csrf akan men-generate kode token otomatis yang dibuat dalam bentuk *form hidden*.

```

public function simpan(Request $request)
{
    DB::table('mahasiswa')->insert([
        'nama' => $request->namamhs,
        'nim' => $request->nimmhs,
        'email' => $request->emailmh,
        'jurusan' => $request->jurus
    ]);

    return redirect('/mahasiswa');
}

```

Keterangan:

- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 28-33 merupakan query builder untuk insert data ke tabel mahasiswa

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

```

17
18 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');
19

```

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.

## Tambah Data Mahasiswa

[Kembali](#)

Nama

Nim

Email

Jurusan

[Tambah Data](#)

Kembali  
localhost:8000 dan pilih tombol 'Tambah Data Mahasiswa', maka akan ditampilkan halaman tambah yang  
be

memasukkan data baru. Kita coba isikan data pada form tersebut.

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



Data Mahasiswa		
<a href="#">Tambah Data Mahasiswa</a>		
Nama	NIM	
nanda	1001001001	<a href="#">Detail</a>
wahyu	1001001002	<a href="#">Detail</a>
marsha	1001001003	<a href="#">Detail</a>
marsha	1001001003	<a href="#">Detail</a>

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

#### d. Melihat Detail Data dari Database

Langkah	Keterangan
---------	------------

1

```

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
9      btn-primary">Tambah Data Mahasiswa</a>
10     <br/>
11     <br/>
12     <table class="table table-bordered
13     table-hover table-striped">
14         <thead>
15             <tr>
16                 <th>Nama</th>
17                 <th>NIM</th>
18             </tr>
19         </thead>
20         <tbody>
21             @foreach($mahasiswa as $mhs)
22                 <tr>
23                     <td>{{ $mhs->nama }}</td>
24                     <td>{{ $mhs->nim }}</td>
25                     <td>
26                         <a href="/mahasiswa/
27                         detail/{{ $mhs->id }}" class
28                         ="badge badge-info">Detail</a>
29                     </td>
30                 </tr>
31             @endforeach
32         </tbody>
33     </table>
34 @endsection

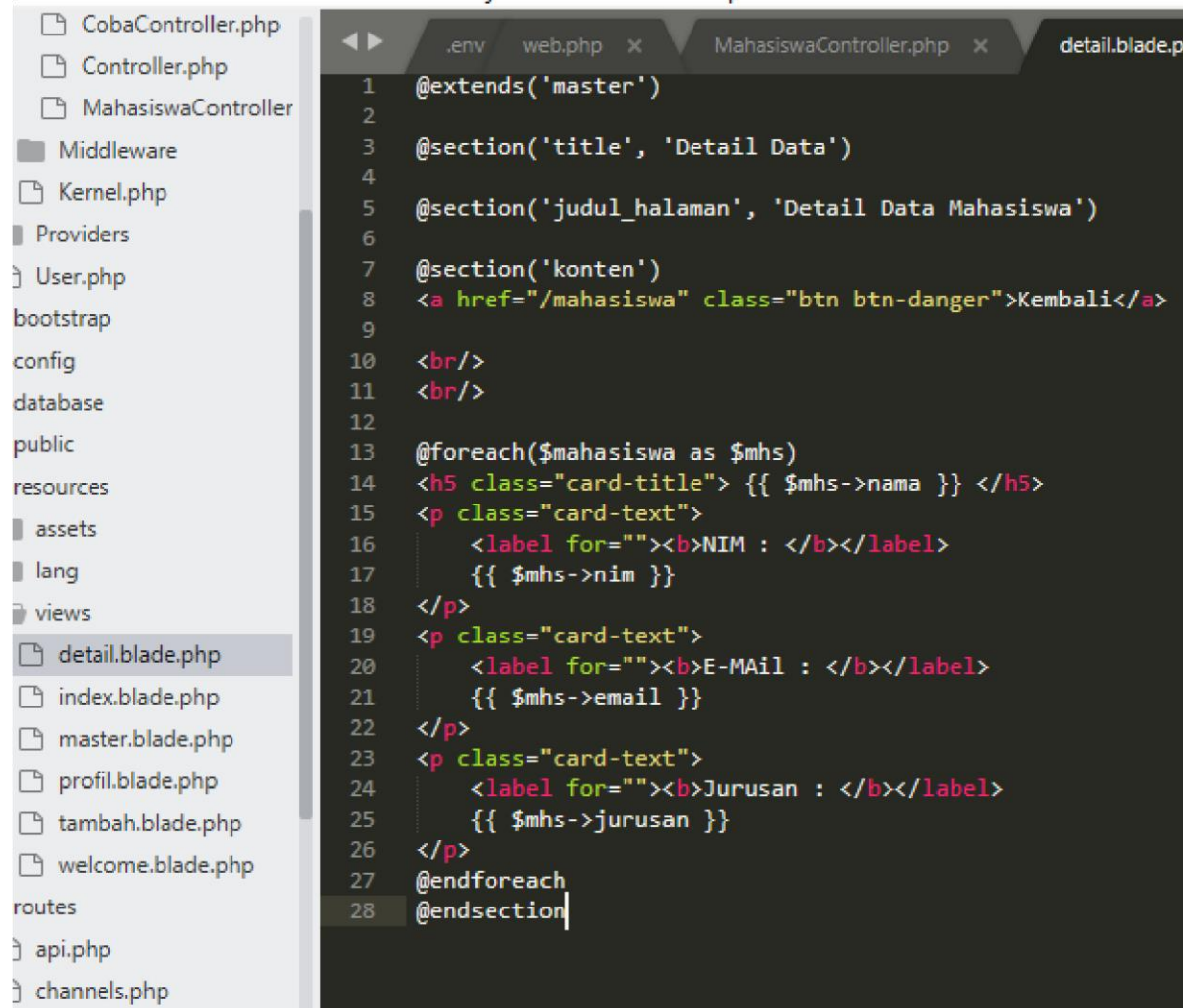
```

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

2	<div data-bbox="268 141 1366 230" data-label="Text"> <pre> 19 20 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 21 </pre> </div> <div data-bbox="268 241 1535 315" data-label="Text"> <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> </div>
3	<div data-bbox="268 454 1535 528" data-label="Text"> <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> </div>

```
public function detail($id) {  
    $mahasiswa = DB::table('mahasiswa')->where('id',$id)  
    ;  
    return view('detail',['mahasiswa' => $mahasiswa]);  
}
```

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



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the following structure:

- CobaController.php
- Controller.php
- MahasiswaController
- Middleware
- Kernel.php
- Providers
- User.php
- bootstrap
- config
- database
- public
- resources
  - assets
  - lang
  - views
    - detail.blade.php (selected)
    - index.blade.php
    - master.blade.php
    - profil.blade.php
    - tambah.blade.php
    - welcome.blade.php
- routes
  - api.php
  - channels.php

The code editor shows the content of **detail.blade.php**:

```
1 @extends('master')
2
3 @section('title', 'Detail Data')
4
5 @section('judul_halaman', 'Detail Data Mahasiswa')
6
7 @section('konten')
8 <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
9
10 <br/>
11 <br/>
12
13 @foreach($mahasiswa as $mhs)
14 <h5 class="card-title"> {{ $mhs->nama }} </h5>
15 <p class="card-text">
16     <label for=""><b>NIM : </b></label>
17     {{ $mhs->nim }}
18 </p>
19 <p class="card-text">
20     <label for=""><b>E-MAil : </b></label>
21     {{ $mhs->email }}
22 </p>
23 <p class="card-text">
24     <label for=""><b>Jurusan : </b></label>
25     {{ $mhs->jurusan }}
26 </p>
27 @endforeach
28 @endsection
```

Keterangan:

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

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



## Detail Data Mahasiswa

Kembali

**nanda**

**NIM :** 1001001001

**E-MAil :** nandan@gmail.com

**Jurusan :** teknik informatika

Langkah	Keterangan
---------	------------

e. Mengubah Data (Update) dari Database



```

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 }}"
25                         class="badge badge-info">Detail</a>
26                     <td>
27                         <a href="/mahasiswa/edit/{{ $mhs->id }}" class=
28                         "badge badge-info">Edit</a>
29                     </td>
30                 </tr>
31             @endforeach
32         </tbody>
33     </table>
34 @endsection

```

Buatlah tombol untuk mengubah data (Line 29) pada file **index.blade.php** di folder **resources/views**

2	Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/edit</b> yang akan menjalankan fungsi <i>edit</i> pada MahasiswaController
---	---

```
1  
2 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');  
3
```

3

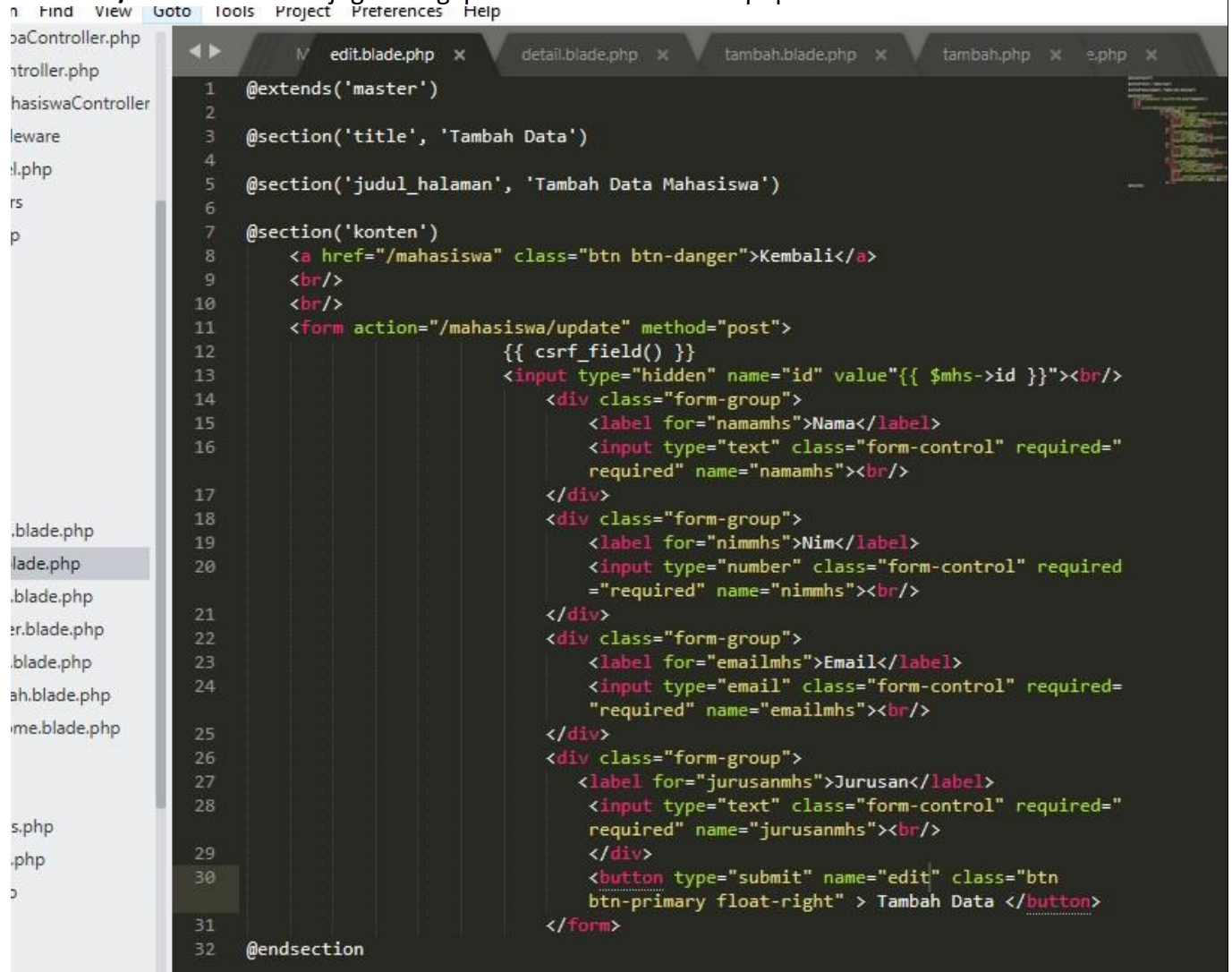
```
public function edit($id) {  
    $mahasiswa = DB::table('mahasiswa')->where('id',$id)->get()  
    ;  
  
    return view('edit',['mahasiswa' => $mahasiswa]);  
}
```

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

Keterangan :

- Line 49 : query builder untuk mengambil data mahasiswa berdasarkan id yang dipilih

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

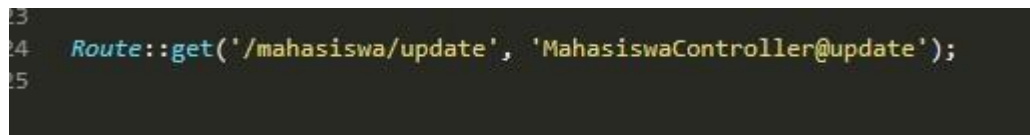


```
1 @extends('master')
2
3 @section('title', 'Tambah Data')
4
5 @section('judul_halaman', 'Tambah Data Mahasiswa')
6
7 @section('konten')
8     <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
9     <br/>
10    <br/>
11    <form action="/mahasiswa/update" method="post">
12        {{ csrf_field() }}
13        <input type="hidden" name="id" value="{{ $mhs->id }}"><br/>
14        <div class="form-group">
15            <label for="namamhs">Nama</label>
16            <input type="text" class="form-control" required="
17            required" name="namamhs"><br/>
18        </div>
19        <div class="form-group">
20            <label for="nimhs">Nim</label>
21            <input type="number" class="form-control" required=
22            ="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=
27            ="required" name="emailmhs"><br/>
28        </div>
29        <div class="form-group">
30            <label for="jurusanmhs">Jurusan</label>
31            <input type="text" class="form-control" required="
32            required" name="jurusanmhs"><br/>
33        </div>
34        <button type="submit" name="edit" class="btn
35        btn-primary float-right" > Tambah Data </button>
36    </form>
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

5



```
13
14 Route::get('/mahasiswa/update', 'MahasiswaController@update');
15
```

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

Keterangan:

	<p>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method upda</p>
6	<p>Buat method <b>update</b> pada <b>MahasiswaController</b> untuk menyimpan data yang diubah ke database.</p> <pre> public function update(Request \$request) {     // update data mahasiswa     DB::table('mahasiswa')-&gt;where('id', \$request-&gt;id)         'nama' =&gt; \$request-&gt;namamhs,         'nim' =&gt; \$request-&gt;nimmhs,         'email' =&gt; \$request-&gt;emailmhs,         'jurusan' =&gt; \$request-&gt;jurusanmhs     ]);     return redirect('/'); } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li> <li>- Line 58-63 merupakan query builder untuk update data ke tabel mahasiswa</li> </ul>
7	<p>Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk</p>

## Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM		
nanda	1001001001	<a href="#">Detail</a>	<a href="#">Edit</a>
wahyu	1001001002	<a href="#">Detail</a>	<a href="#">Edit</a>
marsha	1001001003	<a href="#">Detail</a>	<a href="#">Edit</a>
marsha	1001001003	<a href="#">Detail</a>	<a href="#">Edit</a>

Klik edit di data pertama, kita akan mengubah namanya.

← → ↻ 🏠 ⓘ localhost:8000/mahasiswa/edit/1 ☆ ☆ 👤 ⋮

## Edit Data Mahasiswa

[Kembali](#)

Nama

NIM

Email

Jurusan

[Update Data](#)

laravelapp-00-igbshe\_zin

nanda eka	1001001001	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
wahyu	1001001002	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
marsha	1001001003	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
marsha	1001001003	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>

berubah. Setelah kita klik simpan

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



oller.php

hp

Controller

hp

p

hp

php

hp

e.php

le.php

detail

tambah.blade.php

tambah.php

master.blade.php

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><br>
10 <table class="table">
11     <thead class="thead-dark">
12         <th>Nama</th>
13         <th>NIM</th>
14         <th>Aksi</th>
15     </thead>
16     <tbody>
17         @foreach($mahasiswa as $mhs)
18         <tr>
19             <td>{{ $mhs->nama }}</td>
20             <td>{{ $mhs->nim }}</td>
21             <td>
22                 <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info"
23                 >Detail</a>
24                 <a href="/mahasiswa/edit/{{ $mhs->id }}" class="badge
25                 badge-warning">Edit</a>
26                 <a href="/mahasiswa/hapus/{{ $mhs->id }}" class="badge
27                 badge-danger" onclick="return confirm('Apakah anda ingin
28                 menghapus data ini?')">Hapus</a>
29             </td>
30         </tr>
31         @endforeach
32     </tbody>
33 </table>
34 @endsection

```

2

```

25
26 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');
27

```

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

3

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

```

62
63     public function hapus($id)
64     {
65         // hapus data mahasiswa by id
66         DB::table('mahasiswa')->where('id', $id)->delete();
67         return redirect('/');
68     }
69
70

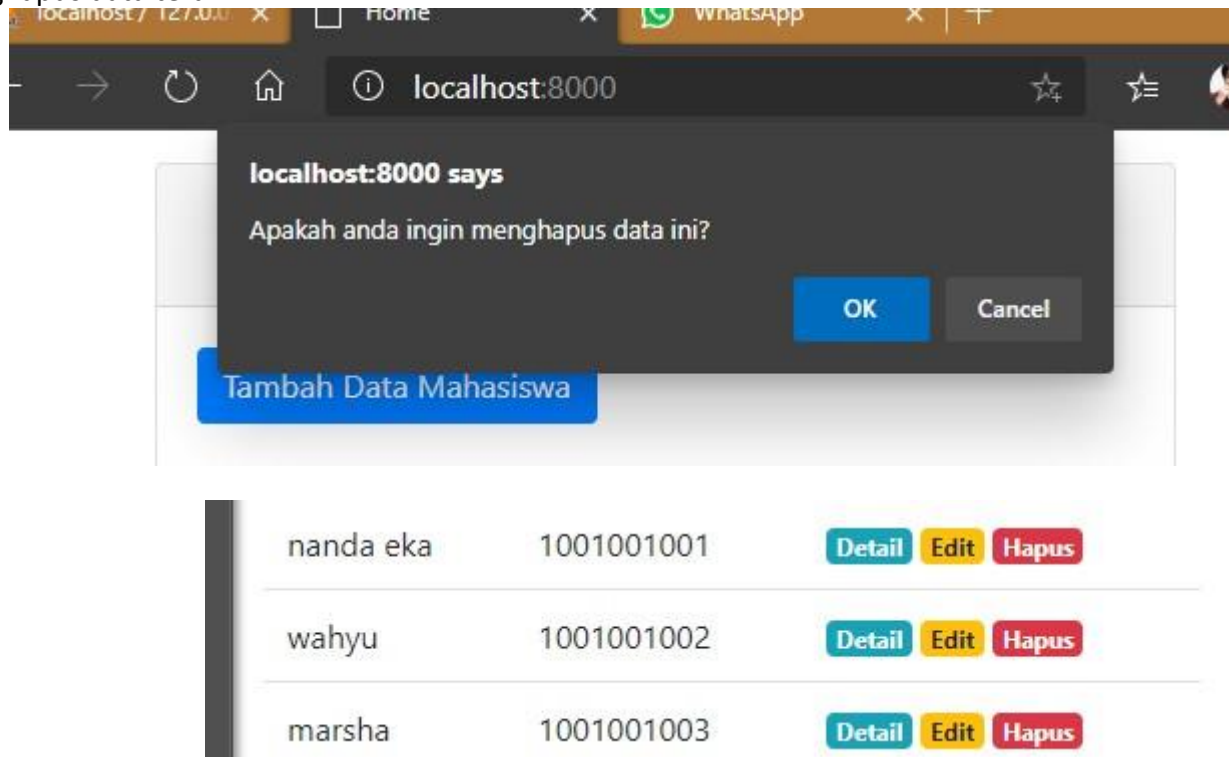
```

Keterangan :

- Line 67 : query builder untuk menghapus data mahasiswa berdasarkan id yang dipilih

4

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

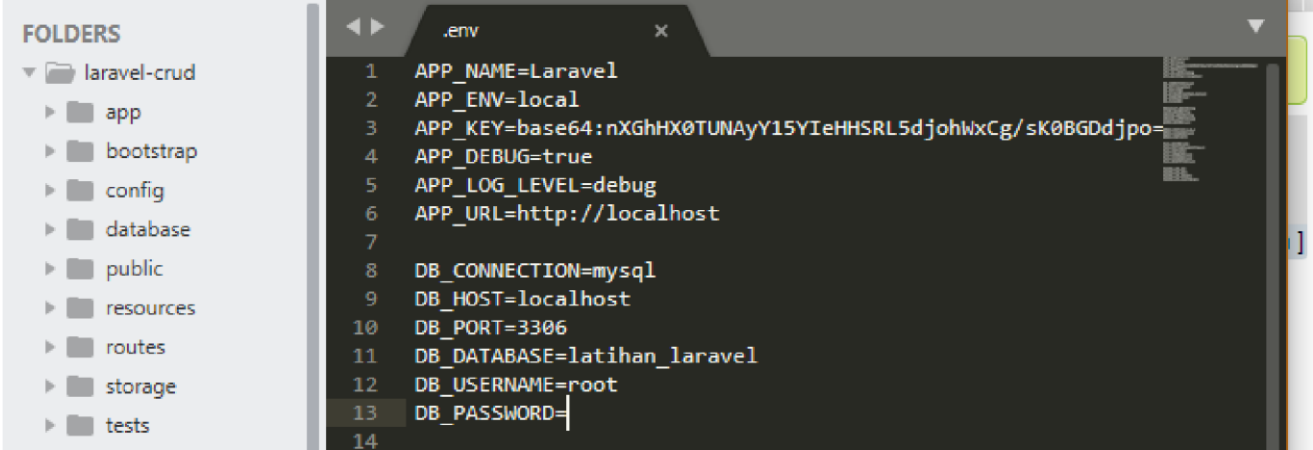


#### f. Menghapus Data (Delete) dari Database

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

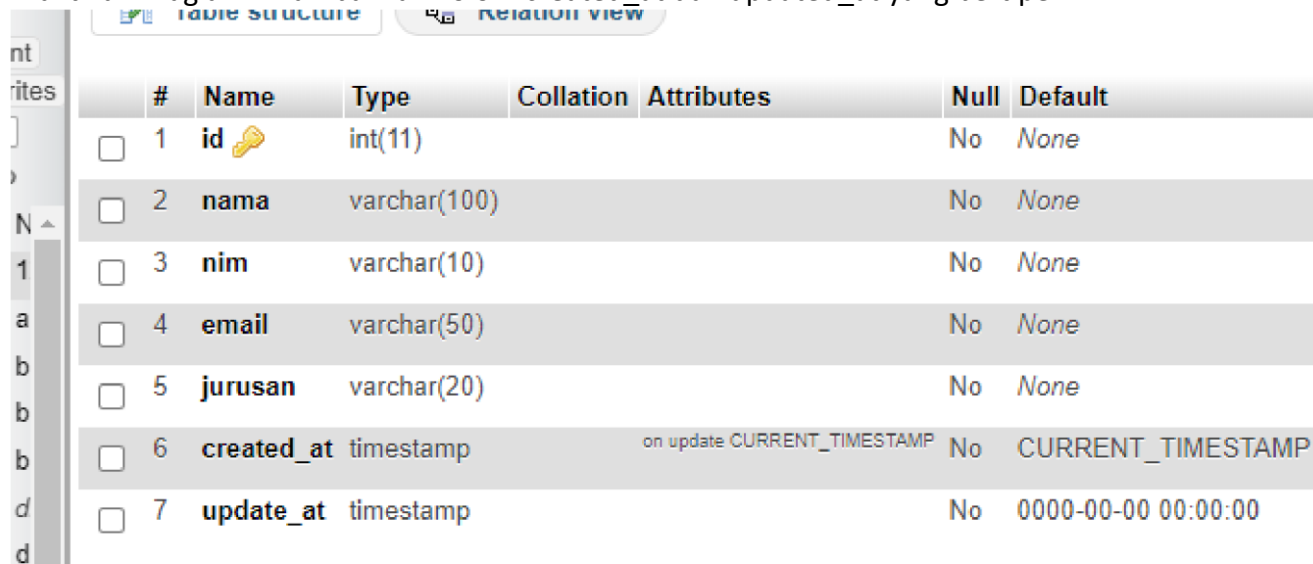
Eloquent adalah sebuah fitur untuk mengelola data yang ada pada database. Eloquent ORM menyediakan fungsi-fungsi active record, atau fungsi-fungsi query sql untuk mengelola data pada database. Dengan menggunakan Eloquent pada Laravel, kita bisa mengelola data pada database dari hanya satu buah model.

### 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>C:\xampp\htdocs laravel new laravel-crud-kedua</pre>
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, buka file <b>.env</b> pada project laravel-crud. Ubah seperti di bawah ini.</p>  <p>Keterangan:</p> <ul style="list-style-type: none"><li>- Nama database yang akan digunakan adalah <code>latihan_laravel</code> dengan username <code>root</code>.</li></ul>

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



The screenshot shows a database table structure with the following columns:

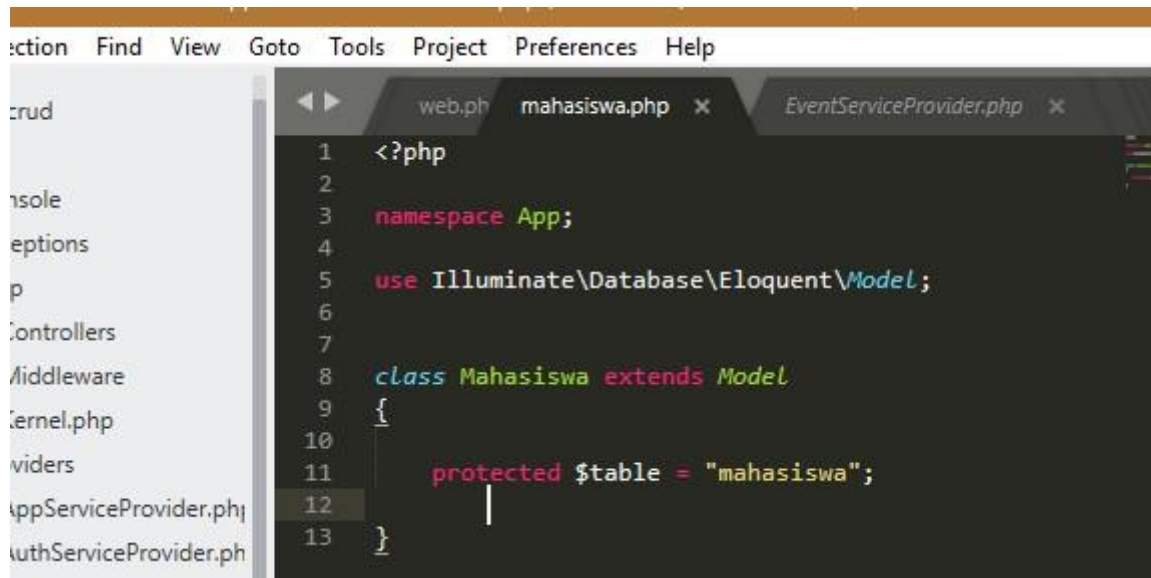
#	Name	Type	Collation	Attributes	Null	Default
1	id	int(11)			No	None
2	nama	varchar(100)			No	None
3	nim	varchar(10)			No	None
4	email	varchar(50)			No	None
5	jurusan	varchar(20)			No	None
6	created_at	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP
7	update_at	timestamp			No	0000-00-00 00:00:00

TIMESTAMP dan default nilainya 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> <pre> 13 14 Route::get('/', 'MahasiswaController@index'); 15 Route::get('/mahasiswa', 'MahasiswaController@index'); 16 </pre> <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>
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>

3



```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7
8 class Mahasiswa extends Model
9 {
10
11     protected $table = "mahasiswa";
12
13 }
```

Ubah model **Mahasiswa.php** pada folder **App** menjadi seperti berikut.

Keterangan:

- Model Mahasiswa akan menangani tabel mahasiswa

4

Buat controller baru yaitu MahasiswaController menggunakan php artisan **php artisan make:controller MahasiswaController**

5



```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App/Mahasiswa;
7
8  class MahasiswaController2 extends Controller
9  {
10     public function index()
11     {
12         //mengambil data dari tabel mahasiswa
13         $mahasiswa = Mahasiswa::all();
14
15         //mengirim data mahasiswa ke view index
16         return view('index', ['mahasiswa' => $mahasiswa]);
17     }

```

Buat

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

Keterangan:

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

6

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

7

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

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

nanda eka	1001001001	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
wahyu	1001001002	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
marsha	1001001003	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>

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

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/tambah</b> yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre>16 17 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 18</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>18 19 public function tambah() 20 { 21     //memanggil view tambah 22     return view('tambah'); 23 } 24</pre>
3	<p>Kemudian buatlah view <b>tambah.blade.php</b> yang berisi form untuk memasukkan data baru pada folder <b>resources/views</b>.</p>
	<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>18 19 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 20</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

Buat method **simpan** pada **MahasiswaController** untuk menyimpan data ke database.

```
24
25 public function simpan(Request $request)
26 {
27     // insert data ke table mahasiswa
28     Mahasiswa::create([
29         'nama' => $request->namamhs,
30         'nim' => $request->nimmhs,
31         'email' => $request->emailmhs,
32         'jurusan' => $request->jurusanmhs
33     ]);
34     return redirect('/');
35 }
```

Keterangan:

- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 23-28 merupakan fitur eloquent menggunakan fungsi create() untuk insert data ke tabel mahasiswa

6

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.

```
14  
15 class Mahasiswa extends Model {  
16  
17     protected $table = "mahasiswa";  
18     protected $fillable = ['nama', 'nim', 'email', 'jurusan'];  
19  
20 }
```

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

### Tambah Data Mahasiswa

[Kembali](#)

Nama

rahmat

NIM

1001001006

E-mail

rahmat@gmail.com



Jurusan

teknik mesin

[Tambah Data](#)

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

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

d. Melihat Detail Data dari Database

Langkah	Keterangan
---------	------------

1	Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/detail</b> yang akan menjalankan fungsi detail pada MahasiswaController
	<pre> 20 21 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');  22 </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> 48 49 public function detail(\$id) 50 { 51     \$mahasiswa = Mahasiswa::find(\$id); 52 53     return view('detail', ['mahasiswa' =&gt; \$mahasiswa]); 54 } 55 </pre>



```

1  @extends('master')
2
3  @section('title', 'Detail Data')
4
5  @section('judul_halaman', 'Detail Data Mahasiswa')
6
7  @section('konten')
8  <a href="/mahasiswa" class="btn btn-danger">Kembali
9  </a>
10
11  <br/>
12
13  <h5 class="card-title"> {{ $mhs->nama }} </h5>
14  <p class="card-text">
15      <label for=""><b>NIM : </b></label>
16      {{ $mhs->nim }}
17  </p>
18  <p class="card-text">
19      <label for=""><b>E-Mail : </b></label>
20      {{ $mhs->email }}
21  </p>
22  <p class="card-text">
23      <label for=""><b>Jurusan : </b></label>
24      {{ $mhs->jurusan }}
25  </p>
26
27  @endsection

```

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

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

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

Langkah	Keterangan
1	<p>Buatlah <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> 22 23  Route::get('/mahasiswa/edit/{id}', '     MahasiswaController@edit'); 24 </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>



```

56     public function edit($id)
57     {
58         $mahasiswa = Mahasiswa::find($id);
59         return view('edit', ['mahasiswa' => $mahasiswa]);
60     }

```

Keterangan :

- Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih

3 Kemudian buatlah view **edit.blade.php** yang berisi form untuk mengubah data pada folder

```

1  @extends('master')
2
3  @section('title', 'Edit Data')
4
5  @section('judul_halaman', 'Edit Data Mahasiswa')
6
7  @section('konten')
8      <a href="/" class="btn btn-danger">Kembali</a>
9      <br>
10     <br>
11     <form action="/mahasiswa/update" method="POST">
12         {{ csrf_field() }}
13         <input type="hidden" name="id" value="{{ $mhs->id }}">
14         <div class="form-group">
15             <label>Nama</label>
16             <input type="text" class="form-control" value="{{ $mhs->nama }}" required="required" name="namamhs"><br>
17         </div>
18         <div class="form-group">
19             <label>NIM</label>
20             <input type="number" class="form-control" value="{{ $mhs->nim }}" required="required" name="nimmhs"><br>
21         </div>
22         <div class="form-group">
23             <label>Email</label>
24             <input type="email" class="form-control" value="{{ $mhs->email }}" required="required" name="emailmhs"><br>
25         </div>
26         <div class="form-group">
27             <label>Jurusan</label>
28             <input type="text" class="form-control" value="{{ $mhs->jurusan }}" required="required" name="jurusanmhs"><br>
29         </div>
30         <input type="submit" class="btn btn-primary float-right">
31     </form>
32 @endsection

```

**resources/views.** View edit juga mengaplikasikan master.blade.php.

	<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> 24 25 Route::post('/mahasiswa/update', 'MahasiswaController@update'); 26 </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>

```

5
6
7 public function update($id, Request $request)
8 {
9     // update data mahasiswa
10    $mahasiswa = Mahasiswa::find($id);
11    $mahasiswa->nama => $request->namamhs;
12    $mahasiswa->nim => $request->nimmhs;
13    $mahasiswa->email => $request->emailmhs;
14    $mahasiswa->jurusan => $request->jurusanmhs;
15    $mahasiswa->save();
16    });
17    return redirect('/');
18 }

```

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.

nanda eka	1001001001	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
wahyu	1001001002	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
marsha	1001001003	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
marsha	1001001003	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>

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

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/hapus</b> yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> 26 27 Route::get('/mahasiswa/hapus/{id}', '     MahasiswaController@hapus'); 28 </pre>

2

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

```
public function hapus($id)
{
    $mahasiswa = Mahasiswa::find($id);
    $mahasiswa->delete();

    return redirect('/');
}
```

Keterangan :

- Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih

3

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

nanda eka	1001001001	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
wahyu	1001001002	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
marsha	1001001003	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>