



## Topik

Membuat CRUD – Create Read Update Delete Data dengan Framework Laravel

## Tujuan

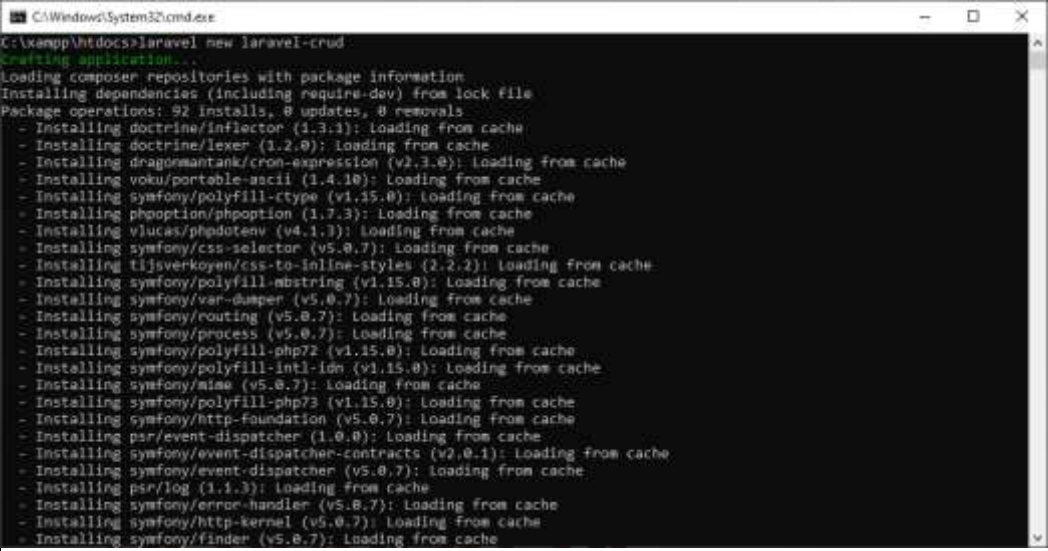
Mahasiswa diharapkan dapat:

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

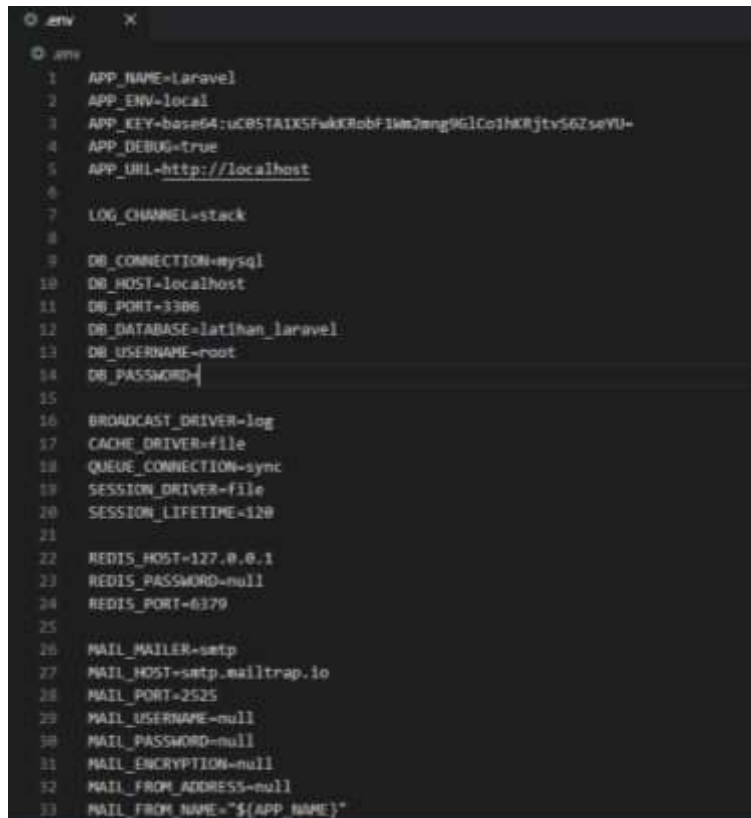
Query Builder adalah salah satu fitur untuk menjalankan *query database*. Laravel menyediakan fungsi- fungsi yang dapat digunakan untuk menjalankan *query* sehingga penulisan *query* menjadi lebih singkat dan efisien.

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

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

2

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

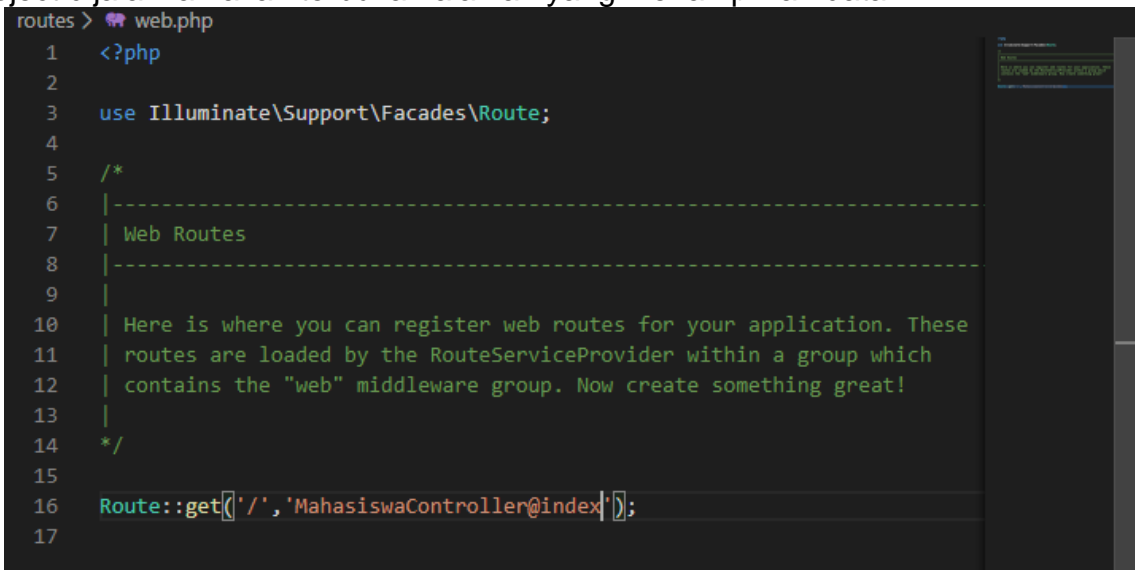


```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:uCR5TAIX5fwkKRobF1m2mg9G1Co1HkRjtv56ZseYU=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=localhost
11 DB_PORT=3306
12 DB_DATABASE=latihan_laravel
13 DB_USERNAME=root
14 DB_PASSWORD=
15
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=127.0.0.1
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
25
26 MAIL_MAILER=smt
27 MAIL_HOST=smt.mailtrap.io
28 MAIL_PORT=2525
29 MAIL_USERNAME=null
30 MAIL_PASSWORD=null
31 MAIL_ENCRYPTION=null
32 MAIL_FROM_ADDRESS=null
33 MAIL_FROM_NAME="${APP_NAME}"
```

Keterangan:

	<ul style="list-style-type: none"> <li>Nama database yang akan digunakan adalah latihan_laravel dengan username root.</li> </ul>
3	<p>Jalankan xampp, selanjutnya buat tabel dengan nama <b>mahasiswa</b> di mysql pada database latihan_laravel.</p> 
4	<p>Isilah beberapa data pada tabel <b>mahasiswa</b> tersebut.</p> 

## b. Menampilkan Data dari Database

Langka h	Keterangan
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan. Pertama, buatlah <b>route</b> pada <b>routes/web.php</b> sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p>  <p>Keterangan:</p> <ul style="list-style-type: none"> <li>Ketika route ('/') diakses, akan dijalankan method <b>index</b> pada controller bernama <b>MahasiswaController</b>.</li> </ul>

2	<p>Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan</p> <p><b>cd laravel-crud</b></p> <p><b>php artisan make:controller MahasiswaController</b></p> <pre>C:\xampp\htdocs&gt;cd laravel-crud C:\xampp\htdocs\laravel-crud&gt;php artisan make:controller MahasiswaController Controller created successfully. C:\xampp\htdocs\laravel-crud&gt;_</pre>
---	---

3

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

```

app > Http > Controllers > MahasiswaController.php > MahasiswaController > index
1  <?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         $mahasiswa = DB::table('mahasiswa')->get();
12
13         return view('index',['mahasiswa' => $mahasiswa]);
14     }
15 }
16

```

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

4

Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat **template blade** (seperti file *header-footer* pada pembahasan CI) dengan nama **master.blade.php** pada folder **resources/views**.

```

resources > views > master.blade.php
1  <html>
2  <head>
3      <meta charset="utf-8">
4      <meta name="viewport" content="width=device-width, initial-scale=1">
5      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css">
6      <title> @yield('title') </title>
7  </head>
8
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 tex-center">
15                         <h2> @yield('judul_halaman') </h2>
16                     </div>
17                     <div class="card-body">
18                         @yield('konten')
19                     </div>
20                 </div>
21             </div>
22         </div>
23     </div>
24 </body>
25 </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  @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                 </tr>
24             @endforeach
25         </tbody>
26     </table>
27 @endsection
```

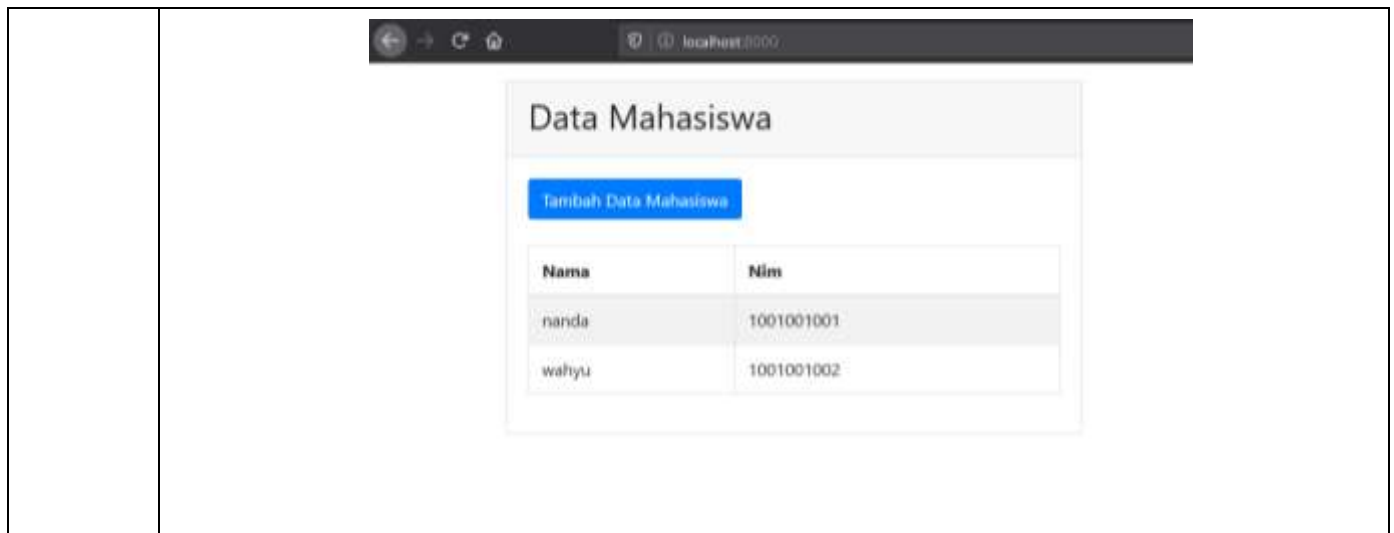
Keterangan:

- Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan bladelain 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

Langka h	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/tambah</b> yang akan menjalankan fungsi tambah pada MahasiswaController</p> <pre> routes &gt; web.php 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('/', 'MahasiswaController@index'); 17   Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 18   21   22   Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');</pre>
2	<p>Buat method <b>tambah</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view tambah.</p> <pre> app &gt; Http &gt; Controllers &gt; MahasiswaController.php &gt; MahasiswaController &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         \$mahasiswa = DB::table('mahasiswa')-&gt;get(); 12 13         return view('index', ['mahasiswa' =&gt; \$mahasiswa]); 14     } 15 16     public function tambah() { 17         return view('tambah'); 18     } 19 } 20</pre>



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.

```
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="nama" >Nama</label>
15             <input type="text" class="form-control" required="required" name="nama"> <br/>
16         </div>
17         <div class="form-group">
18             <label for="nim" >Nim</label>
19             <input type="number" class="form-control" required="required" name="nim"> <br/>
20         </div>
21         <div class="form-group">
22             <label for="email" >E-mail</label>
23             <input type="email" class="form-control" required="required" name="email"> <br/>
24         </div>
25         <div class="form-group">
26             <label for="jurusan" >Jurusan</label>
27             <input type="text" class="form-control" required="required" name="jurusan"> <br/>
28         </div>
29         <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button>
30     </form>
31 @endsection
```

	<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 <i>form hidden</i>.</li> </ul>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/simpan</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> routes &gt; web.php 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('/', 'MahasiswaController@index'); 17   Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 18   Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 19   </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>

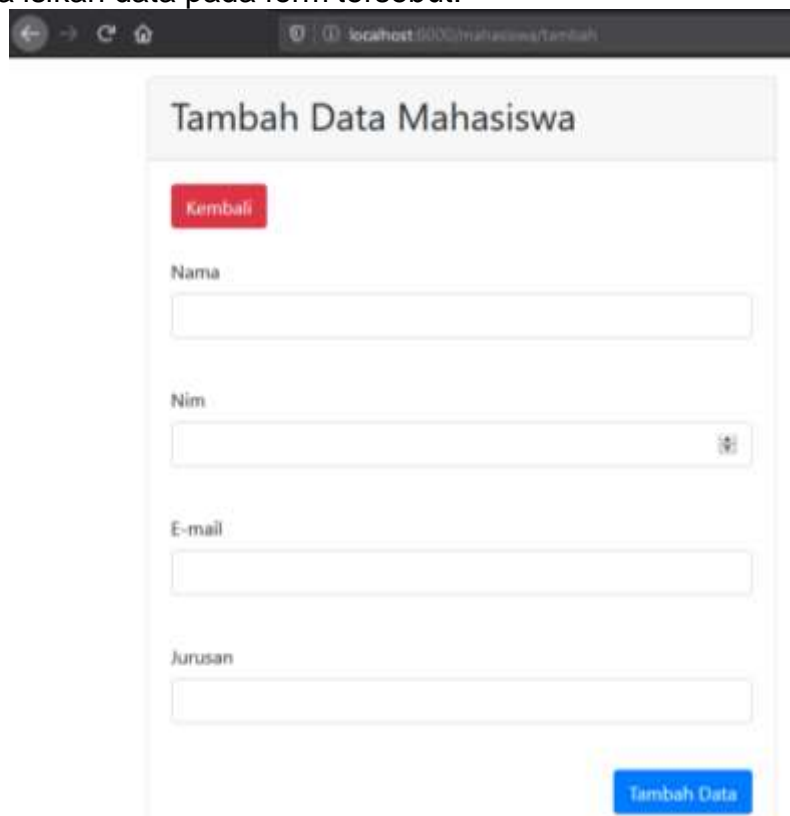
```
19
20     public function simpan(Request $request) {
21         DB::table('mahasiswa')->insert([
22             'nama' => $request->namamhs,
23             'nim' => $request->nimmhs,
24             'email' => $request->emailmhs,
25             'jurusan' => $request->jurusanmhs
26         ]);
27         return redirect('/mahasiswa');
28     }
29 }
30
```

Keterangan:

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

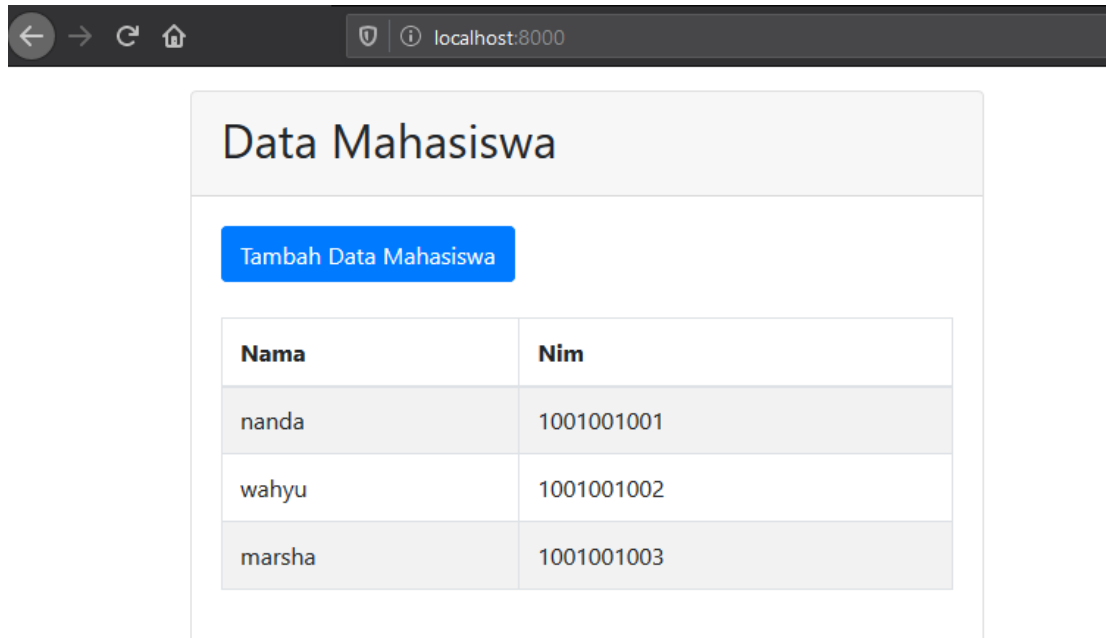
6

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



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/mahasiswa/tambah'. The main content area features a form titled 'Tambah Data Mahasiswa'. At the top left of the form is a red button labeled 'Kembali'. Below this are four input fields: 'Nama', 'Nim', 'E-mail', and 'Jurusan'. At the bottom right of the form is a blue button labeled 'Tambah Data'.

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



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

#### d. Melihat Detail Data dari Database

Langka h	Keterangan
1	<p>Buatlah tombol untuk melihat detail data (Line 28) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre> resources &gt; views &gt; index.blade.php 1  @extends('master') 2 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;/td&gt; 26                 &lt;/tr&gt; 27             @endforeach 28         &lt;/tbody&gt; 29     &lt;/table&gt; 30 @endsection                     </pre>

2	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/detail</b> yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre data-bbox="284 197 1449 470">15 16 Route::get('/mahasiswa','MahasiswaController@index'); 17 Route::get('/mahasiswa/tambah','MahasiswaController@tambah'); 18 Route::post('/mahasiswa/simpan','MahasiswaController@simpan'); 19 Route::get('/mahasiswa/detail/{id}','MahasiswaController@detail'); 20</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 data-bbox="284 645 1476 902">public function detail(\$id) {     \$mahasiswa = DB::table('mahasiswa')-&gt;where('id', \$id)-&gt;get();      return view('detail',['mahasiswa' =&gt; \$mahasiswa]); }</pre>

4

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

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

## Data Mahasiswa

Tambah Data Mahasiswa

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

## Detail Data Mahasiswa

Kembali

nanda

**NIM :** 1001001001

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

**Jurusan :** teknik informatika

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

Langka h	Keterangan
-------------	------------

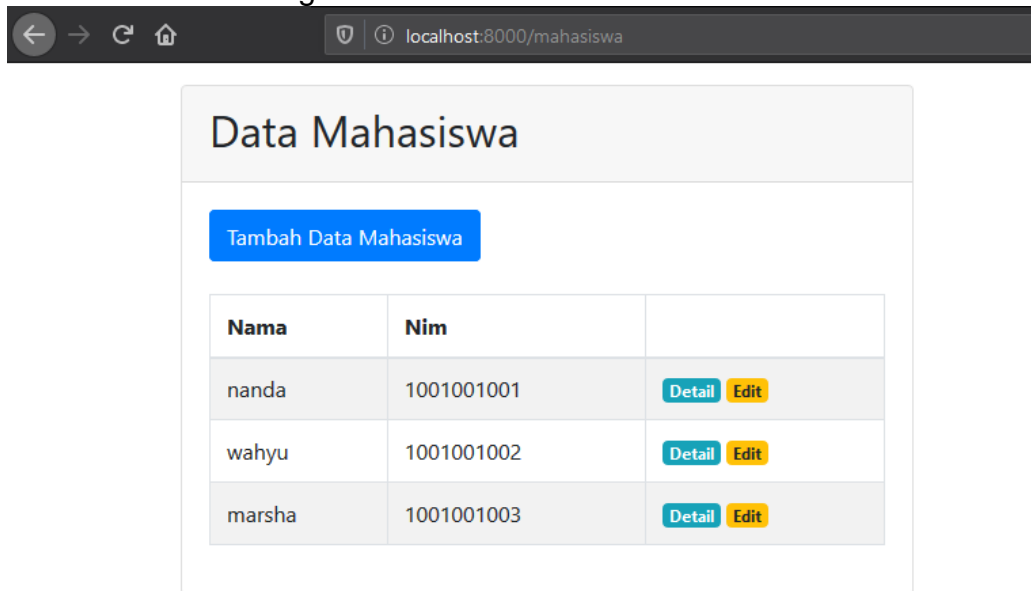


1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file <b>index.blade.php</b> di folder <b>resources/views</b></p> <pre> &lt;/thead&gt; &lt;tbody&gt;     @foreach(\$mahasiswa as \$mhs)         &lt;tr&gt;             &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt;             &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt;             &lt;td&gt;                 &lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt;                 &lt;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; &lt;/table&gt; @endsection </pre>
2	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/edit</b> yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre> Route::get('/mahasiswa','MahasiswaController@index'); Route::get('/mahasiswa/tambah','MahasiswaController@tambah'); Route::post('/mahasiswa/simpan','MahasiswaController@simpan'); Route::get('/mahasiswa/detail/{id}','MahasiswaController@detail'); 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> resources &gt; views &gt; edit.blade.php 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     {{ csrf_field() }} 14     &lt;input type="hidden" name="id" value="{{ \$mhs-&gt;id }}"&gt; &lt;br/&gt; 15     &lt;div class="form-group"&gt; 16         &lt;label for="nama{{ \$mhs }}"&gt;Nama&lt;/label&gt; 17         &lt;input type="text" class="form-control" required="required" name="nama{{ \$mhs }}" value="{{ \$mhs-&gt;nama }}"&gt; 18     &lt;/div&gt; 19     &lt;div class="form-group"&gt; 20         &lt;label for="nim{{ \$mhs }}"&gt;Nim&lt;/label&gt; 21         &lt;input type="number" class="form-control" required="required" name="nim{{ \$mhs }}" value="{{ \$mhs-&gt;nim }}"&gt; 22     &lt;/div&gt; 23     &lt;div class="form-group"&gt; 24         &lt;label for="email{{ \$mhs }}"&gt;E-mail&lt;/label&gt; 25         &lt;input type="email" class="form-control" required="required" name="email{{ \$mhs }}" value="{{ \$mhs-&gt;email }}"&gt; 26     &lt;/div&gt; 27     &lt;div class="form-group"&gt; 28         &lt;label for="jurusan{{ \$mhs }}"&gt;Jurusan&lt;/label&gt; 29         &lt;input type="text" class="form-control" required="required" name="jurusan{{ \$mhs }}" value="{{ \$mhs-&gt;jurusan }}"&gt; 30     &lt;/div&gt; 31     &lt;button type="submit" name="edit" class="btn btn-primary float-right"&gt;Simpan Data&lt;/button&gt; 32 &lt;/form&gt; 33 @endforeach 34 @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 <b>/mahasiswa/update</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p>

```
15
16 Route::get('/mahasiswa','MahasiswaController@index');
17 Route::get('/mahasiswa/tambah','MahasiswaController@tambah');
18 Route::post('/mahasiswa/simpan','MahasiswaController@simpan');
19 Route::get('/mahasiswa/detail/{id}','MahasiswaController@detail');
20 Route::get('/mahasiswa/edit/{id}','MahasiswaController@edit');
21 Route::post('/mahasiswa/update','MahasiswaController@update');
```

Keterangan:

	<ul style="list-style-type: none"> <li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php</li> </ul>
6	<p>Buat method <b>update</b> pada <b>MahasiswaController</b> untuk menyimpan data yang diubah ke database.</p> <pre> 41 42     public function update(Request \$request) { 43         DB::table('mahasiswa')-&gt;where('id',\$request-&gt;id)-&gt;update([ 44             'nama' =&gt; \$request-&gt;namamhs, 45             'nim' =&gt; \$request-&gt;nimmhs, 46             'email' =&gt; \$request-&gt;emailmhs, 47             'jurusan' =&gt; \$request-&gt;jurusanmhs 48         ]); 49         return redirect('/mahasiswa'); 50     } 51 } 52 </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 mengubah data baru.</p>  <p>Klik edit di data pertama, kita akan mengubah namanya.</p>

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

Nama	Nim	
nanda eka	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
marsha	1001001003	Detail Edit

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

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

```

2         <td>{{ $mhs->nama }}</td>
3         <td>{{ $mhs->nim }}</td>
4         <td>
5             <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a>
6             <a href="/mahasiswa/edit/{{ $mhs->id }}" class="badge badge-warning">Edit</a>
7             <a href="/mahasiswa/hapus/{{ $mhs->id }}" class="badge badge-danger">Hapus</a>
8         </td>
9     </tr>
10 @endforeach
11 </tbody>
12 </table>
13 @endsection

```

2

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

```

Route::get('/mahasiswa', 'MahasiswaController@index');
Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');
Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');
Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');
Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');
Route::post('/mahasiswa/update', 'MahasiswaController@update');
Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');

```

3

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

```

public function hapus($id) {
    DB::table('mahasiswa')->where('id', $id)->delete();

    return redirect('/mahasiswa');
}

```

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.



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>

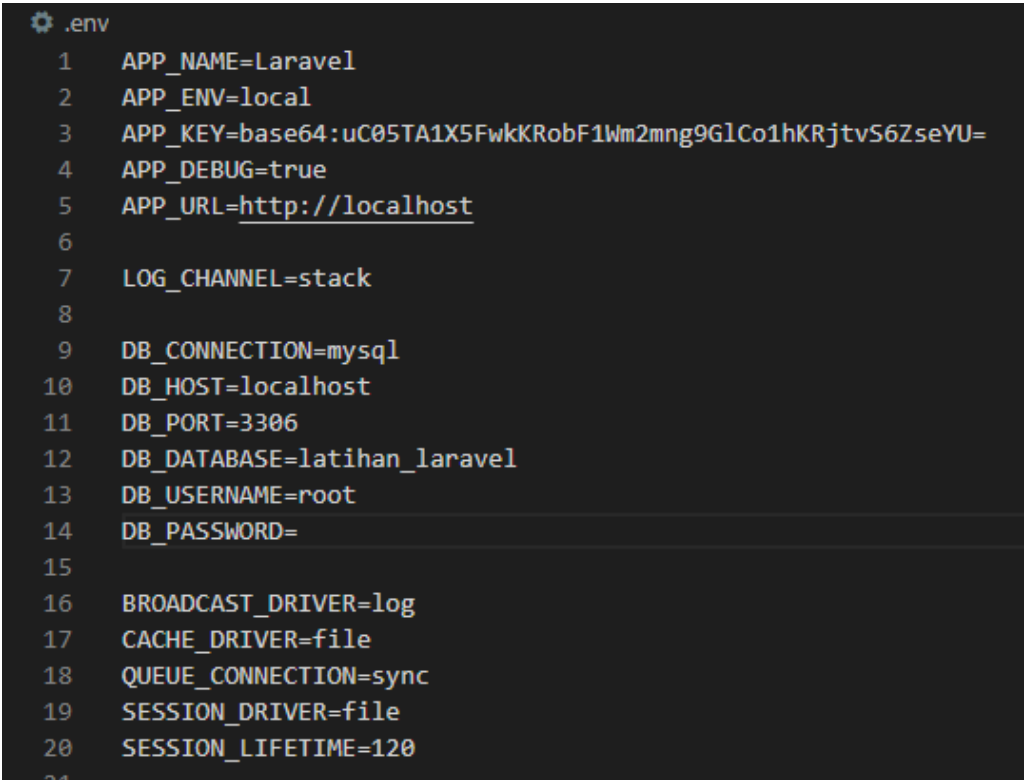


Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	Nim	
nanda eka	1001001001	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
wahyu	1001001002	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

## 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

Langka h	Keterangan
1	Buatlah project Laravel baru dengan nama <b>laravel-crud-kedua</b> . Buka command prompt, tuliskan perintah berikut. <b>cd C:\xampp\htdocs</b> <b>laravel new laravel-crud-kedua</b>
2	Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file <b>.env</b> pada project laravel-crud. Ubah seperti di bawah ini.  <pre>.env 1 APP_NAME=Laravel 2 APP_ENV=local 3 APP_KEY=base64:uC05TA1X5FwkKRobF1Wm2mng9GlCo1hKRjtvS6ZseYU= 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 16 BROADCAST_DRIVER=log 17 CACHE_DRIVER=file 18 QUEUE_CONNECTION=sync 19 SESSION_DRIVER=file 20 SESSION_LIFETIME=120 21</pre> <p>Keterangan:</p> <ul style="list-style-type: none"><li>- Nama database yang akan digunakan adalah latihan_laravel dengan username root.</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 `TIMESTAMP` dan default nilainya `NULL`

#	Name	Type	Collation	Attributes	Null	Default	Comments	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)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	<b>nim</b>	varchar(10)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 4	<b>email</b>	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 5	<b>jurusan</b>	varchar(20)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 6	<b>created_at</b>	timestamp			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 7	<b>updated_at</b>	timestamp			Yes	NULL			Change  Drop  More

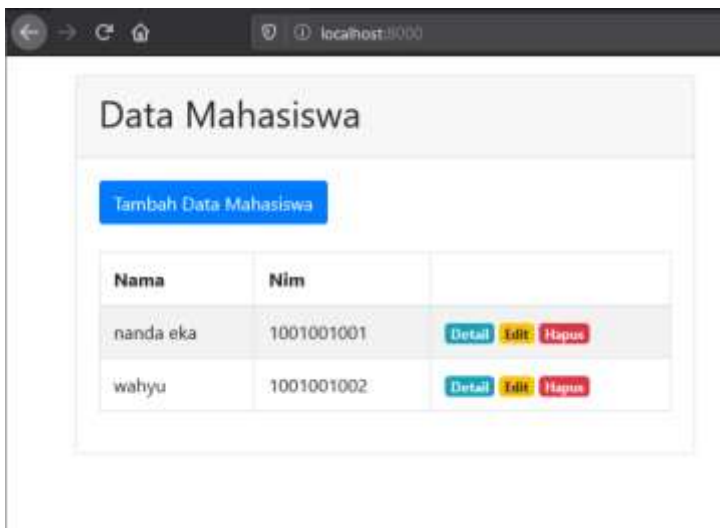
☐ Check all    With selected: Browse Change Drop Primary Unique Index Fulltext

#	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)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 3	<b>nim</b>	varchar(10)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 4	<b>email</b>	varchar(50)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 5	<b>jurusan</b>	varchar(20)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 6	<b>created_at</b>	timestamp			Yes	NULL		
<input type="checkbox"/> 7	<b>updated_at</b>	timestamp			Yes	NULL		

## b. Menampilkan Data dari Database

Langka h	Keterangan
1	<p>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.</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 <b>cd laravel-crud-kedua</b> <b>php artisan make:model Mahasiswa</b></p> <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>
3	<p>Ubah model <b>Mahasiswa.php</b> pada folder <b>App</b> menjadi seperti berikut.</p> <pre>app &gt; 🐘 Mahasiswa.php &gt; 🗑 Mahasiswa 1 &lt;?php 2 3 namespace App; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class Mahasiswa extends Model 8 { 9     protected \$table = "mahasiswa"; 10 } 11</pre>

	<div data-bbox="279 100 1372 593">  <p>The screenshot shows the VS Code interface. The Explorer on the left shows the project structure with 'Mahasiswa.php' selected under 'app'. The Editor on the right shows the code for 'Mahasiswa.php' with the following content:</p> <pre> 1  &lt;?php 2 3  namespace App; 4 5  use Illuminate\Database\Eloquent\Model; 6 7  class Mahasiswa extends Model 8  { 9      protected \$table = "mahasiswa"; 10 }</pre> </div> <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 <b>php artisan make:controller MahasiswaController</b></p>
5	<p>Buat method <b>index</b> pada MahasiswaController.php pada folder <b>app/Http/Controllers</b></p> <div data-bbox="279 884 1460 1601">  <p>The screenshot shows the VS Code Editor with the file 'MahasiswaController.php' open. The code is as follows:</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         \$mahasiswa = Mahasiswa::all(); 12 13         return view('index',['mahasiswa' =&gt; \$mahasiswa]); 14     } 15 }</pre> </div> <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 <b>template blade</b> (seperti pada Bagian 1).</p> <p>Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)</p>
7	<p>Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud <b>php artisan serve</b></p> <p>Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut</p> 

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

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/tambah</b> yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre> 18 19  Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 20 </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	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
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes <b>/mahasiswa/simpan</b>. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 20 21 Route::post('/mahasiswa/simpan','MahasiswaController@simpan'); 22 </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	<p>Karena kita menggunakan fungsi create pada Controller, maka butuh ditambahkan code pada Line 10 yang disebut Mass Assignment pada model Mahasiswa.php.</p> <p>Mass Assignment digunakan untuk memfilter kolom mana yang boleh dan tidak boleh diinput.</p> <pre> app &gt; Mahasiswa.php &gt; Mahasiswa 1 &lt;?php 2 3 namespace App; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class Mahasiswa extends Model 8 { 9     protected \$table = "mahasiswa"; 10    protected \$fillable = ['nama','nim','email','jurusan']; 11 } </pre>

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.
---	--

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

Nama	Nim	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
rahmat	1001001006	Detail Edit Hapus

#### d. Melihat Detail Data dari Database

Langka h	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



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

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 <b>detail.blade.php</b>.</p> <pre> public function detail(\$id) {     \$mahasiswa = Mahasiswa::find(\$id);      return view('detail', ['mahasiswa' =&gt; \$mahasiswa]); } </pre>
4	<p>Kemudian buatlah view <b>detail.blade.php</b> yang menampilkan detail data mahasiswa pada folder <b>resources/views</b>. View detail juga mengaplikasikan <b>master.blade.php</b>.</p> <pre> resources &gt; views &gt; detail.blade.php 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>
5	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p>

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

Langka h	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> 24 25  Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 26 </pre>
2	<p>Buat method <b>edit</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menampilkan view edit.</p>

	<pre> public function edit(\$id) {     \$mahasiswa = Mahasiswa::find(\$id);      return view('edit', ['mahasiswa' =&gt; \$mahasiswa]); } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih</li> </ul>
3	<p>Kemudian buatlah view <b>edit.blade.php</b> yang berisi form untuk mengubah data pada folder <b>resources/views</b>. View edit juga mengaplikasikan master.blade.php.</p> <pre> resources &gt; views &gt; edit.blade.php 1  extends('master') 2 3  section('title', 'Edit Data') 4 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 12     &lt;form action="/mahasiswa/update" method="post"&gt; 13         {{ csrf_field() }} 14         &lt;input type="hidden" name="id" value="{{ \$mahasiswa-&gt;id }}"&gt; &lt;br/&gt; 15         &lt;div class="form-group"&gt; 16             &lt;label for="nama"&gt;Nama&lt;/label&gt; 17             &lt;input type="text" class="form-control" required="required" name="nama" value="{{ \$mahasiswa-&gt;nama }}"&gt; 18         &lt;/div&gt; 19         &lt;div class="form-group"&gt; 20             &lt;label for="nim"&gt;Nim&lt;/label&gt; 21             &lt;input type="number" class="form-control" required="required" name="nim" value="{{ \$mahasiswa-&gt;nim }}"&gt; 22         &lt;/div&gt; 23         &lt;div class="form-group"&gt; 24             &lt;label for="email"&gt;E-mail&lt;/label&gt; 25             &lt;input type="email" class="form-control" required="required" name="email" value="{{ \$mahasiswa-&gt;email }}"&gt; 26         &lt;/div&gt; 27         &lt;div class="form-group"&gt; 28             &lt;label for="jurusan"&gt;Jurusan&lt;/label&gt; 29             &lt;input type="text" class="form-control" required="required" name="jurusan" value="{{ \$mahasiswa-&gt;jurusan }}"&gt; 30         &lt;/div&gt; 31         &lt;button type="submit" name="edit" class="btn btn-primary float-right"&gt;Simpan Data&lt;/button&gt; 32     &lt;/form&gt; 33 </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>

	<pre> public function update(Request \$request) {     \$mahasiswa = Mahasiswa::find(\$id);     \$mahasiswa-&gt;nama = \$request-&gt;namamhs;     \$mahasiswa-&gt;nim = \$request-&gt;nimmhs;     \$mahasiswa-&gt;email = \$request-&gt;emailmhs;     \$mahasiswa-&gt;jurusan = \$request-&gt;jurusanmhs;     \$mahasiswa-&gt;save();     return redirect('/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 48-52 merupakan fungsi eloquent untuk update data ke tabel mahasiswa</li> </ul>
6	<p>Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru.</p> <p>Cobalah untuk mengedit data.</p>

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

Langka h	Keterangan
1	<p>Buatlah <i>route</i> baru pada <b>routes/web.php</b> dengan nama <b>/mahasiswa/hapus</b> yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); </pre>
2	<p>Buat method <b>hapus</b> pada <b>MahasiswaController.php</b> di folder <b>app/Http/Controllers</b> yang akan menjalankan fungsi hapus.</p> <pre> public function hapus(\$id) {     \$mahasiswa = Mahasiswa::find(\$id);     \$mahasiswa-&gt;delete();     return redirect('/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	Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.
---	---

**-- Selamat  
Mengerjakan --**