

LAPORAN PRAKTIKUM  
PEMROGRAMAN WEB LANJUT

“MEMBUAT CRUD MENGGUNAKAN LARAVEL”



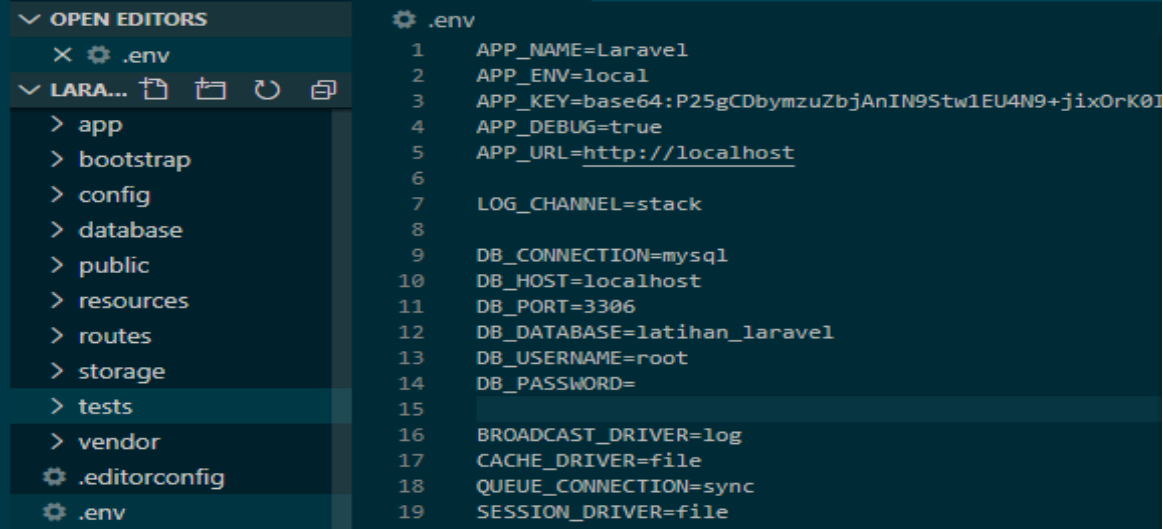
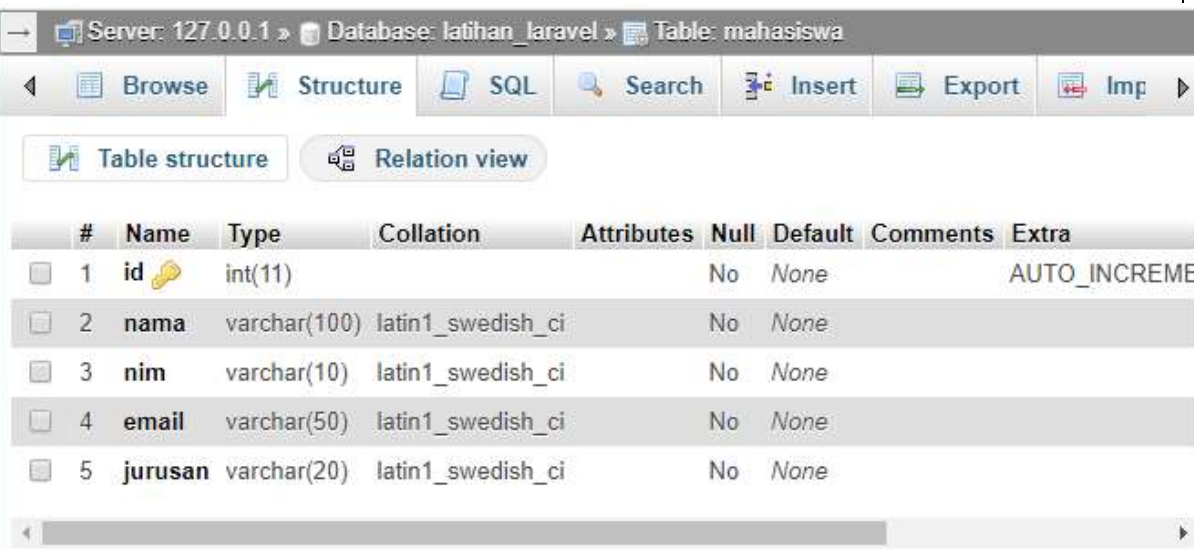
NAMA : SILVI INDAH NOVITA SARI  
NIM : 1841720006  
KELAS : 2B – TI

D-IV TEKNIK INFORMATIKA  
TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG

2020

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

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

1	<p>Buatlah project Laravel baru dengan nama laravel-crud. Buka command prompt, tuliskan perintah berikut.</p> <pre>cd C:\xampp\htdocs laravel new laravel-crud</pre>
2	<p>Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.</p>  <pre>.env 1 APP_NAME=Laravel 2 APP_ENV=local 3 APP_KEY=base64:P25gCDbymzuZbjAnIN9Stw1EU4N9+jixOrK0I 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</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	<p>Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database latihan_laravel</p> 

4 Isilah beberapa data pada tabel mahasiswa tersebut.

☐ Show all | Number of rows: 25 ▼

+ Options

				id	nama	nim	email	jurusan
<input type="checkbox"/>	Edit	Copy	Delete	1	nanda	1001001001	nandan@gmail.com	teknik informatika
<input type="checkbox"/>	Edit	Copy	Delete	2	wahyu	1001001002	wahyu@gmail.com	teknik elektro

↑ ☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 ▼

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 route pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.

```
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
11  | routes are loaded by the RouteServiceProvider with
12  | contains the "web" middleware group. Now create so
13  |
14  */
15
16  // Route::get('/', function () {
17  //     return view('welcome');
18  // });
19
20  Route::get('/', 'MahasiswaController@index');
21
```

Keterangan:

- Ketika route ('/') diakses, akan dijalankan method index pada controller bernama MahasiswaController

2	<p>Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan</p> <p>cd laravel-crud</p> <p>php artisan make:controller MahasiswaController</p> <pre> C:\xampp\htdocs&gt;cd laravel-crud C:\xampp\htdocs\laravel-crud&gt;php artisan make:controller MahasiswaController </pre>
3	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p> <pre> 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         //mengambil data mahasiswa ke view index 16         return view('index', ['mahasiswa' =&gt; \$mahasiswa]); 17     } 18 } 19 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Tambahkan 'use Illuminate\Support\Facades\DB;' (line 6) agar query builder dapat digunakan</li> <li>- Line 13 untuk mengambil data dari tabel mahasiswa menggunakan query builder laravel dan akan disimpan di variabel \$mahasiswa</li> <li>- Line 16 : data akan dikirim ke blade view bernama index</li> </ul>
4	<p>Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat template blade (seperti file header-footer pada pembahasan CI) dengan nama master.blade.php pada folder resources/views.</p>

	<div data-bbox="264 195 1388 1268"> <pre> resources &gt; views &gt; master.blade.php &gt; html 1  &lt;html&gt; 2 3  &lt;head&gt; 4      &lt;meta charset="utf-8"&gt; 5      &lt;meta name="viewport" content="width=device-width, initial-scale=1"&gt; 6      &lt;link href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" rel="stylesheet"&gt; 7      &lt;title&gt; @yield('title') &lt;/title&gt; 8  &lt;/head&gt; 9 10 &lt;body&gt; 11     &lt;div class="container"&gt; 12         &lt;div class="row mt-3"&gt; 13             &lt;div class="col-md-6"&gt; 14                 &lt;div class="card"&gt; 15                     &lt;div class="card-header text-center"&gt; 16                         &lt;!-- bagian judul halaman--&gt; 17                         &lt;h2&gt;@yield('judul_halaman')&lt;/h2&gt; 18                     &lt;/div&gt; 19                     &lt;div class="card-body"&gt; 20 21                         &lt;!-- bagian konten blog--&gt; 22                         @yield('konten') 23                     &lt;/div&gt; 24                     &lt;div class="card"&gt; 25                     &lt;/div&gt; 26                 &lt;/div&gt; 27             &lt;/div&gt; 28         &lt;/div&gt; 29     &lt;/div&gt; 30 &lt;/body&gt; 31 32 &lt;/html&gt; </pre> </div> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- 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</li> </ul> <div data-bbox="207 1465 1421 1560"> <p>5 Sekarang buat file index.blade.php yang menerapkan template master.blade.php dan akan digunakan untuk menampilkan data.</p> </div>
--	--

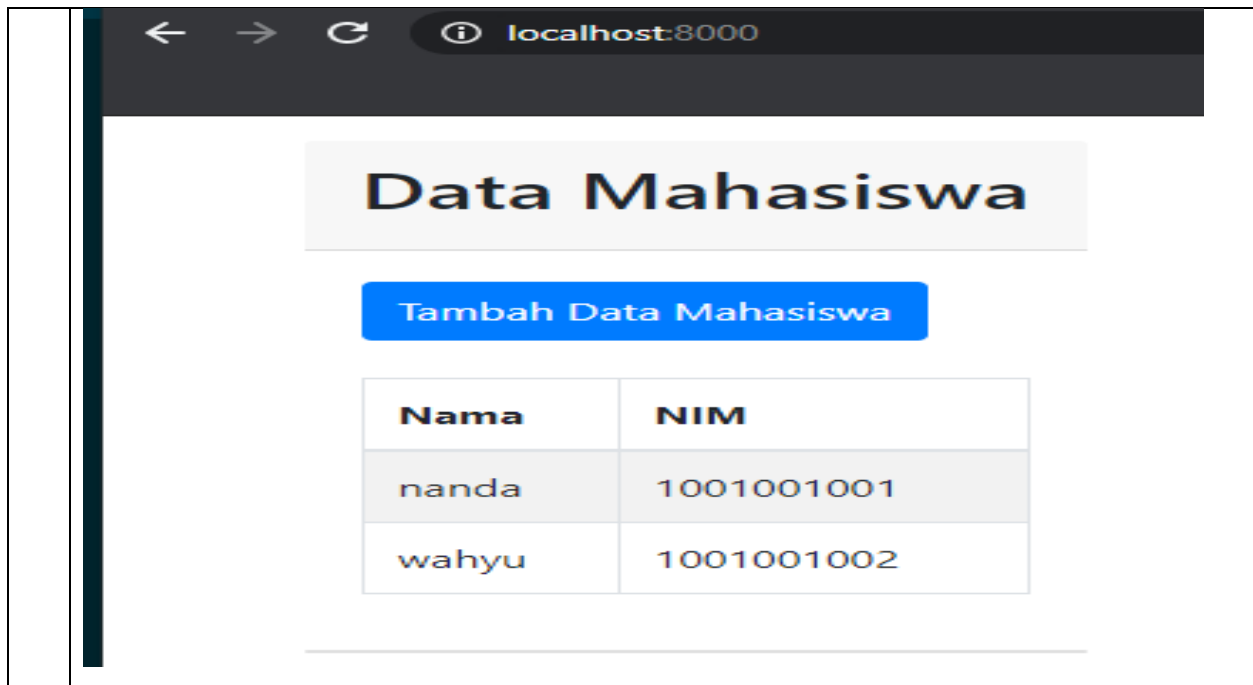
resources > views > index.blade.php > ...

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

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

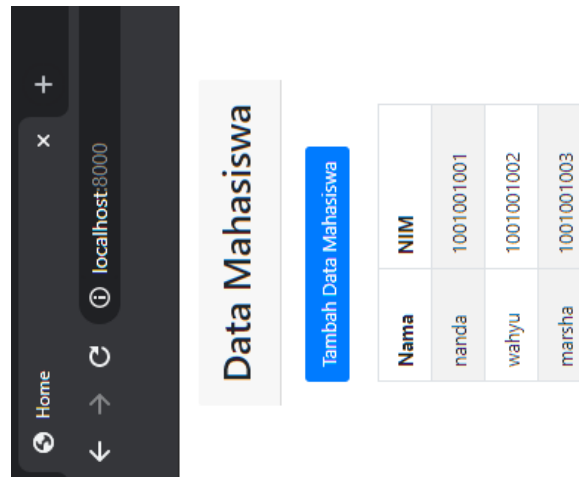
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController</p> <pre> 21 22 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 23 </pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> 18 public function tambah() 19 { 20     //memanggil view tambah 21     return view('tambah'); 22 } 23 24 </pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. View tambah juga mengaplikasikan master.blade.php.</p>

	<pre> resources &gt; views &gt; tambah.blade.php &gt; ... 1  extends('master') 2 3  &lt;!--isi file--&gt; 4  @section ('title','Tambah Data') 5 6  &lt;!--isi bagian judul halaman--&gt; 7  @section('judul_halaman', 'Tambah Data Mahasiswa') 8 9  &lt;!--isi bagian konten--&gt; 10 @section('konten') 11 &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 12 &lt;br /&gt; 13 &lt;br /&gt; 14 &lt;form action="/mahasiswa/simpan" method="post"&gt; 15     {{csrf_field()}} 16     &lt;div class="form-group"&gt; 17         &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 18         &lt;input type="text" class="form-control" required="required" name=" 19     &lt;/div&gt; 20     &lt;div class="form-group"&gt; 21         &lt;label for="namamhs"&gt;NIM&lt;/label&gt; 22         &lt;input type="number" class="form-control" required="required" name=" 23     &lt;/div&gt; 24     &lt;div class="form-group"&gt; 25         &lt;label for="namamhs"&gt;E-Mail&lt;/label&gt; 26         &lt;input type="email" class="form-control" required="required" name=" 27     &lt;/div&gt; 28     &lt;div class="form-group"&gt; 29         &lt;label for="namamhs"&gt;Jurusan&lt;/label&gt; 30         &lt;input type="text" class="form-control" required="required" name=" 31     &lt;/div&gt; 32     &lt;button type="submit" name="tambah" class="btn btn-primary float-right"&gt; 33 &lt;/form&gt; 34 @endsection </pre>
	<p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 13-32 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan</li> <li>- Line 13 terdapat action="/mahasiswa/simpan" yang menunjukkan routes /mahasiswa/simpan dimana data pada form tersebut akan dikirimkan ke fungsi simpan pada controller MahasiswaController</li> <li>- Line 14 terdapat {{ csrf_field() }} yang digunakan untuk menerapkan fitur laravel yaitu csrf protection. csrf protection adalah fitur keamanan untuk mencegah penginputan dari luar aplikasi, csrf akan men-generate kode token otomatis yang dibuat dalam bentuk form hidden</li> </ul>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');</pre>



	<p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php</li> </ul>
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan</p> <pre> 23     public function simpan(Request \$request) 24     { 25         //insert data ke table mahasiswa 26         DB::table('mahasiswa')-&gt;insert([ 27             'nama' =&gt; \$request-&gt;namamhs, 28             'nim' =&gt; \$request-&gt;nimmhs, 29             'email' =&gt; \$request-&gt;emailmhs, 30             'jurusan' =&gt; \$request-&gt;jurusanmhs, 31         ]); 32         return redirect('/mahasiswa'); 33     } 34 } 35 </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li> <li>- Line 28-33 merupakan query builder untuk insert data ke tabel mahasiswa</li> </ul>
6	<p>Kembali coba jalankan localhost:8000 dan pilih tombol ‘Tambah Data Mahasiswa’, maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.</p> 

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



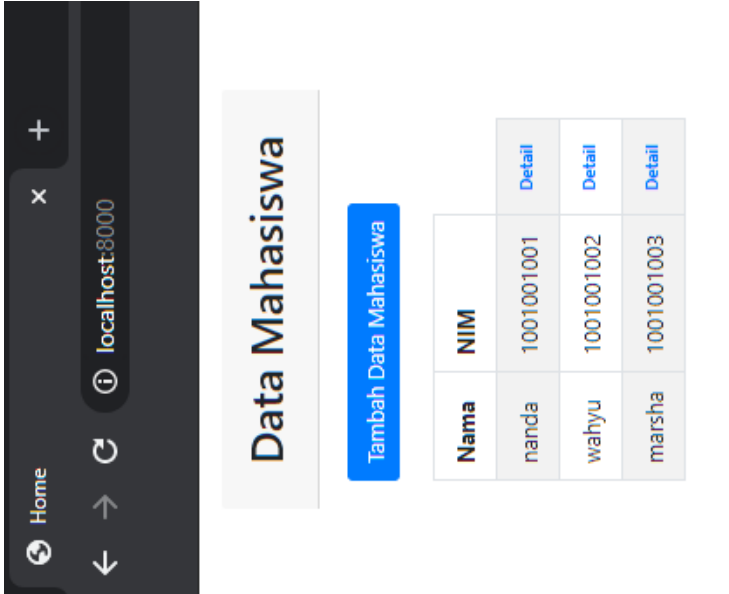
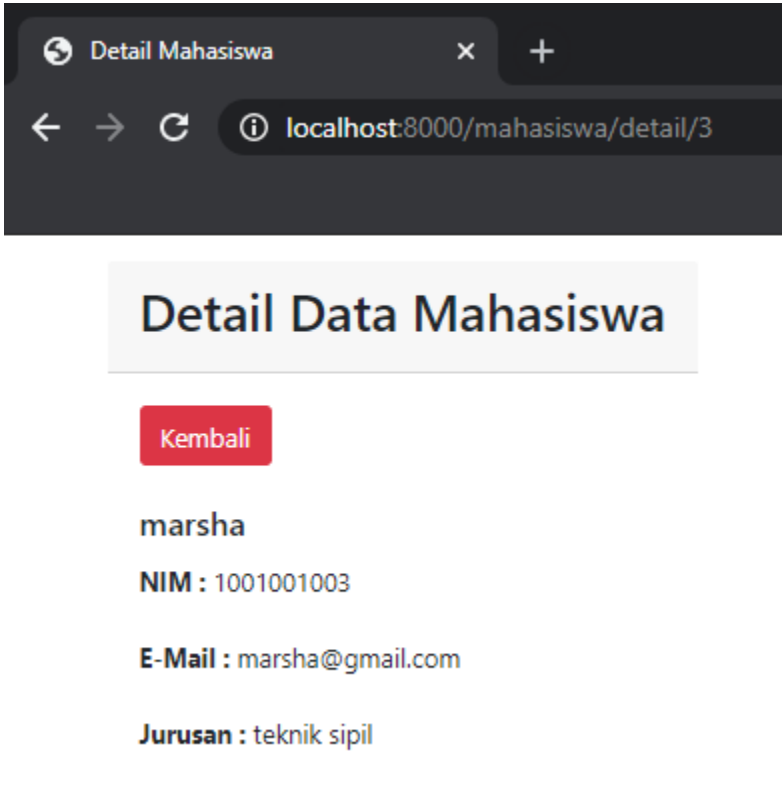
Kita dapat mencoba menambahkan beberapa data lagi agar jumlah data lebih banyak.

#### d. Melihat Detail Data dari Database

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

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

2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre> 26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 27 </pre>
3	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p> <pre> 33     } 34     public function detail(\$id) 35     { 36         //memanggil data mahasiswa berdasarkan id yang dipilih 37         \$mahasiswa = DB::table('mahasiswa')-&gt;where('id', \$id)-&gt;get(); 38         //kirim data mahasiswa yang diambil ke view edit.blade.php 39         return view('detail', ['mahasiswa' =&gt; \$mahasiswa]); 40     } 41 } 42 </pre>
4	<p>Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php.</p> <pre> resources &gt; views &gt; detail.blade.php &gt; ... 1  @extends('master') 2 3  &lt;!--isi file--&gt; 4  @section('title','Detail Mahasiswa') 5 6  &lt;!--isi bagian judul halaman--&gt; 7  @section('judul_halaman', 'Detail Data Mahasiswa') 8 9  &lt;!--isi bagian konten--&gt; 10 @section('konten') 11 &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 12 &lt;br /&gt; 13 &lt;br /&gt; 14 15 @foreach(\$mahasiswa as \$mhs) 16 &lt;h5 class="card-title"&gt;{{\$mhs-&gt;nama}}&lt;/h5&gt; 17 &lt;p class="card-text"&gt; 18     &lt;label for=""&gt;&lt;b&gt;NIM : &lt;/b&gt;&lt;/label&gt; 19     {{\$mhs-&gt;nim}}&lt;/p&gt; 20 &lt;p class="card-text"&gt; 21     &lt;label for=""&gt;&lt;b&gt;E-Mail : &lt;/b&gt;&lt;/label&gt; 22     {{\$mhs-&gt;email}}&lt;/p&gt; 23 &lt;p class="card-text"&gt; 24     &lt;label for=""&gt;&lt;b&gt;Jurusan : &lt;/b&gt;&lt;/label&gt; 25     {{\$mhs-&gt;jurusan}}&lt;/p&gt; 26 @endforeach 27 @endsection </pre>

	<p>Keterangan:</p> <ul style="list-style-type: none"> <li>- Line 16-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan</li> </ul>
5	<p>Jalankan localhost:8000 dan pilih tombol ‘Detail’ di suatu data yang ingin kita lihat detailnya.</p>  

e. Mengubah Data (Update) dari Database

1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views</p> <pre>&gt; index.blade.php &gt; table.table.table-bordered.table-hover.table-striped &gt; tbody &gt; tr &gt; td &gt; td &gt; 1  @extends('master') 2 3  &lt;!--isi file--&gt; 4  @section ('title','Home') 5 6  &lt;!--isi bagian judul halaman--&gt; 7  @section('judul_halaman', 'Data Mahasiswa') 8 9  &lt;!--isi bagian konten--&gt; 10 @section('konten') 11 &lt;a href="/mahasiswa/tambah" class="btn btn-primary"&gt;Tambah Data Mahasiswa&lt;/a&gt; 12 &lt;br /&gt; 13 &lt;br /&gt; 14 &lt;table class="table table-bordered table-hover table-striped"&gt; 15     &lt;thead&gt; 16         &lt;tr&gt; 17             &lt;th&gt;Nama&lt;/th&gt; 18             &lt;th&gt;NIM&lt;/th&gt; 19         &lt;/tr&gt; 20     &lt;/thead&gt; 21     &lt;tbody&gt; 22         @foreach(\$mahasiswa as \$mhs) 23             &lt;tr&gt; 24                 &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 25                 &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 26                 &lt;td&gt;&lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 27                 &lt;td&gt;&lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Edit&lt;/a&gt; 28             &lt;/td&gt; 29         &lt;/tr&gt; 30     @endforeach 31 &lt;/tbody&gt; 32 &lt;/table&gt; 33 @endsection</pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre>28 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 29</pre>
3	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p>

```

41     public function edit($id)
42     {
43         //mengambil data mahasiswa berdasarkan id yang dipilih
44         $mahasiswa = DB::table('mahasiswa')->where('id', $id)->get();
45         //kirim data mahasiswa yang diambil ke view edit.blade.php
46         return view('edit', ['mahasiswa' => $mahasiswa]);
47     }
48 }
49

```

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.

```

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

```

Keterangan:

- Line 14-36 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 15 terdapat action="/mahasiswa/update" yang menunjukkan routes /mahasiswa/update dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController

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

```
30 Route::post('/mahasiswa/update', 'MahasiswaController@update');
31
```

Keterangan :

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

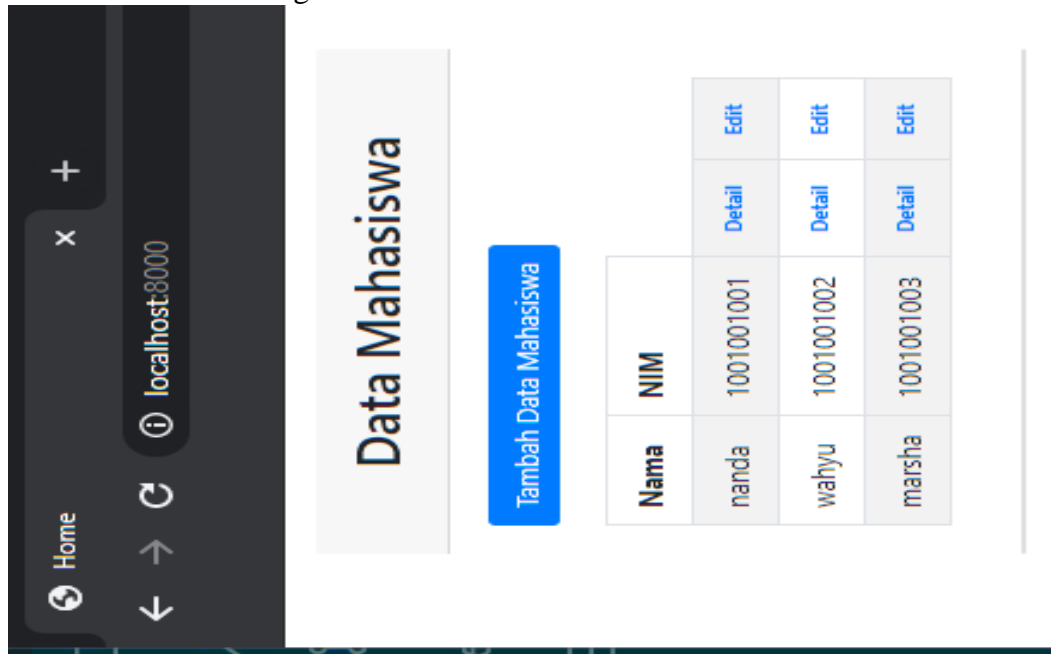
- 6 Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.

Keterangan:

```
48 public function update(Request $request)
49 {
50     //update data mahasiswa
51     DB::table('mahasiswa')->where('id', $request->id)->update([
52         'nama' => $request->namamhs,
53         'nim' => $request->nimmhs,
54         'email' => $request->emailmhs,
55         'jurusan' => $request->jurusanmhs
56     ]);
57     return redirect('/mahasiswa');
58 }
```

- 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.



Klik edit di data pertama, kita akan mengubah namanya.

Detail Data Mahasiswa

Kembali

Nama  
nanda eka

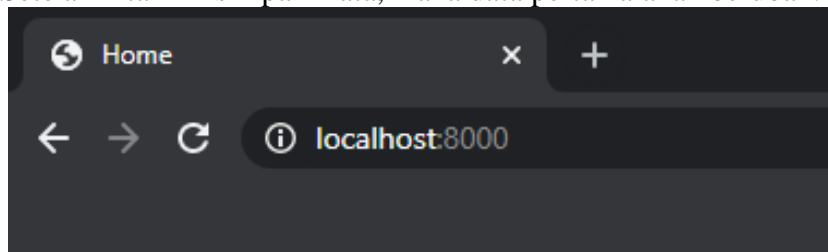
NIM  
1001001001

E-Mail  
nandan@gmail.com

Jurusan  
teknik informatika

Simpan Data

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



## Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM		
nanda eka	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>



f. Menghapus Data (Delete) dari Database

1	<p>Buatlah tombol untuk menghapus data (Line 30) pada file index.blade.php di folder resources/views</p> <pre>resources &gt; views &gt; index.blade.php &gt; table.table-bordered.table-hover.table-striped &gt; tbody &gt; 1  @extends('master') 2 3  &lt;!--isi file--&gt; 4  @section ('title','Home') 5 6  &lt;!--isi bagian judul halaman--&gt; 7  @section('judul_halaman', 'Data Mahasiswa') 8 9  &lt;!--isi bagian konten--&gt; 10 @section('konten') 11 &lt;a href="/mahasiswa/tambah" class="btn btn-primary"&gt;Tambah Data Mahasiswa&lt;/a&gt; 12 &lt;br /&gt; 13 &lt;br /&gt; 14 &lt;table class="table table-bordered table-hover table-striped"&gt; 15     &lt;thead&gt; 16         &lt;tr&gt; 17             &lt;th&gt;Nama&lt;/th&gt; 18             &lt;th&gt;NIM&lt;/th&gt; 19         &lt;/tr&gt; 20     &lt;/thead&gt; 21     &lt;tbody&gt; 22         @foreach(\$mahasiswa as \$mhs) 23             &lt;tr&gt; 24                 &lt;td&gt;{{ \$mhs-&gt;nama }}&lt;/td&gt; 25                 &lt;td&gt;{{ \$mhs-&gt;nim }}&lt;/td&gt; 26                 &lt;td&gt;&lt;a href="/mahasiswa/detail/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Detail&lt;/a&gt; 27                 &lt;td&gt;&lt;a href="/mahasiswa/edit/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Edit&lt;/a&gt; 28                 &lt;td&gt;&lt;a href="/mahasiswa/hapus/{{ \$mhs-&gt;id }}" class="badge badge-info"&gt;Hapus&lt;/a&gt; 29             &lt;/td&gt; 30         &lt;/tr&gt; 31     @endforeach 32 &lt;/tbody&gt; 33 &lt;/table&gt; 34 @endsection</pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre>32 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 33</pre>
3	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p>

```

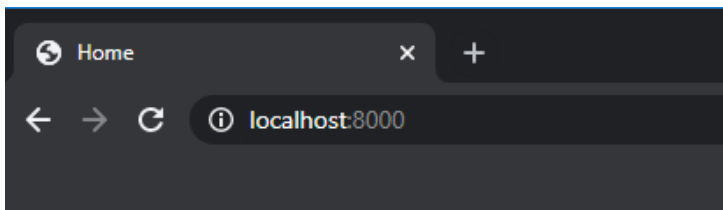
58     }
59     public function hapus($id)
60     {
61         //hapus data mahasiswa berdasarkan id
62         DB::table('mahasiswa')->where('id', $id)->delete();
63
64         return redirect('/mahasiswa');
65     }
66 }
67

```

Keterangan :

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

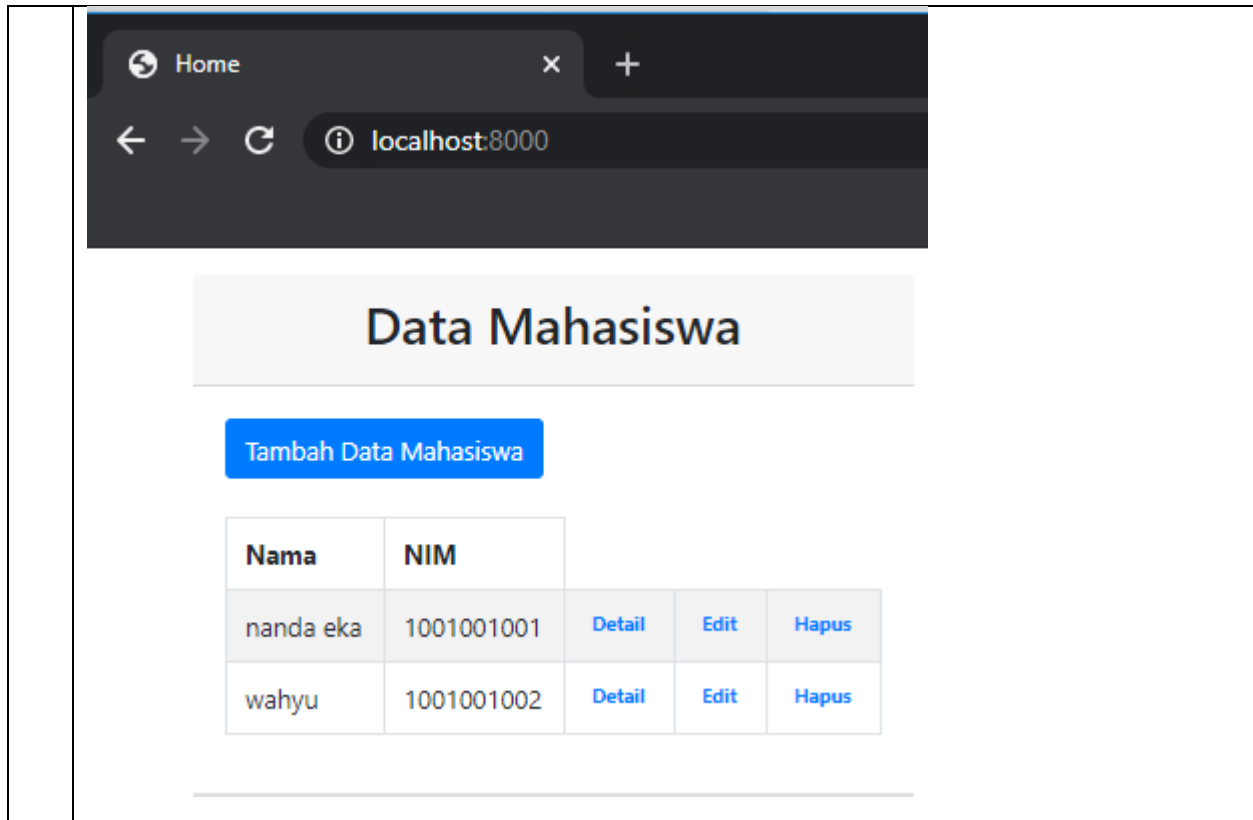
- 4 Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir.



## Data Mahasiswa

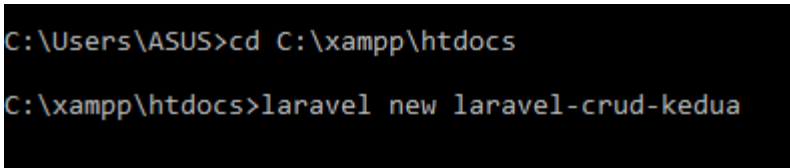
Tambah Data Mahasiswa

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>



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

### a. Konfigurasi Database

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

```

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:xHEwHpxnD4LmCE6XiUgZQjkMExjr5+YcV
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

```

Keterangan:

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

- 3 Pada project ini kita gunakan database dan tabel yang sebelumnya digunakan pada Praktikum Bagian 1. Tambahkan kolom created\_at dan updated\_at yang bertipe TIMESTAMP dan default nilainya NULL

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	nama	varchar(100)	latin1_swedish_ci		No	None		
3	nim	varchar(10)	latin1_swedish_ci		No	None		
4	email	varchar(50)	latin1_swedish_ci		No	None		
5	jurusan	varchar(20)	latin1_swedish_ci		No	None		
6	create_at	timestamp			Yes	NULL		
7	updated_at	timestamp			Yes	NULL		

## b. Menampilkan Data dari Database

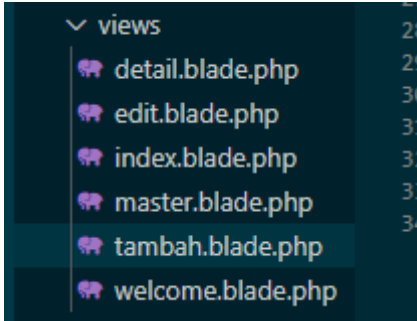
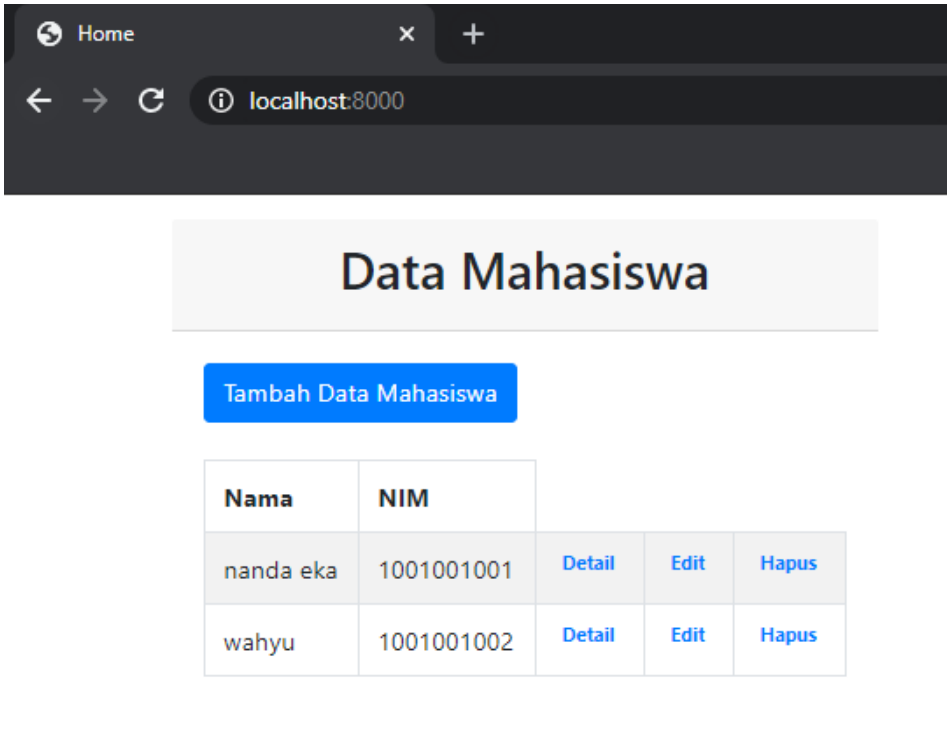
- 1 Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan. Pertama, buatlah route pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.

```

21 Route::get('/', 'MahasiswaController@index');
22 Route::get('/mahasiswa', 'MahasiswaController@index');
23

```

2	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan cd laravel-crud-kedua php artisan make:model Mahasiswa</p> <pre> C:\xampp\htdocs&gt;cd laravel-crud-kedua C:\xampp\htdocs\laravel-crud-kedua&gt;php artisan make:model Mahasiswa </pre>
3	<p>Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.</p> <pre> app &gt; Mahasiswa.php &gt; PHP IntelliSense &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> <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 php artisan make:controller MahasiswaController</p> <pre> C:\xampp\htdocs\laravel-crud-kedua&gt;php artisan make:controller MahasiswaController Controller created successfully. </pre>
5	<p>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</p> <pre> app &gt; Http &gt; Controllers &gt; MahasiswaController.php &gt; PHP Intelephense &gt; 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>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)</p> 
7	<p>Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud php artisan serve Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut</p> 

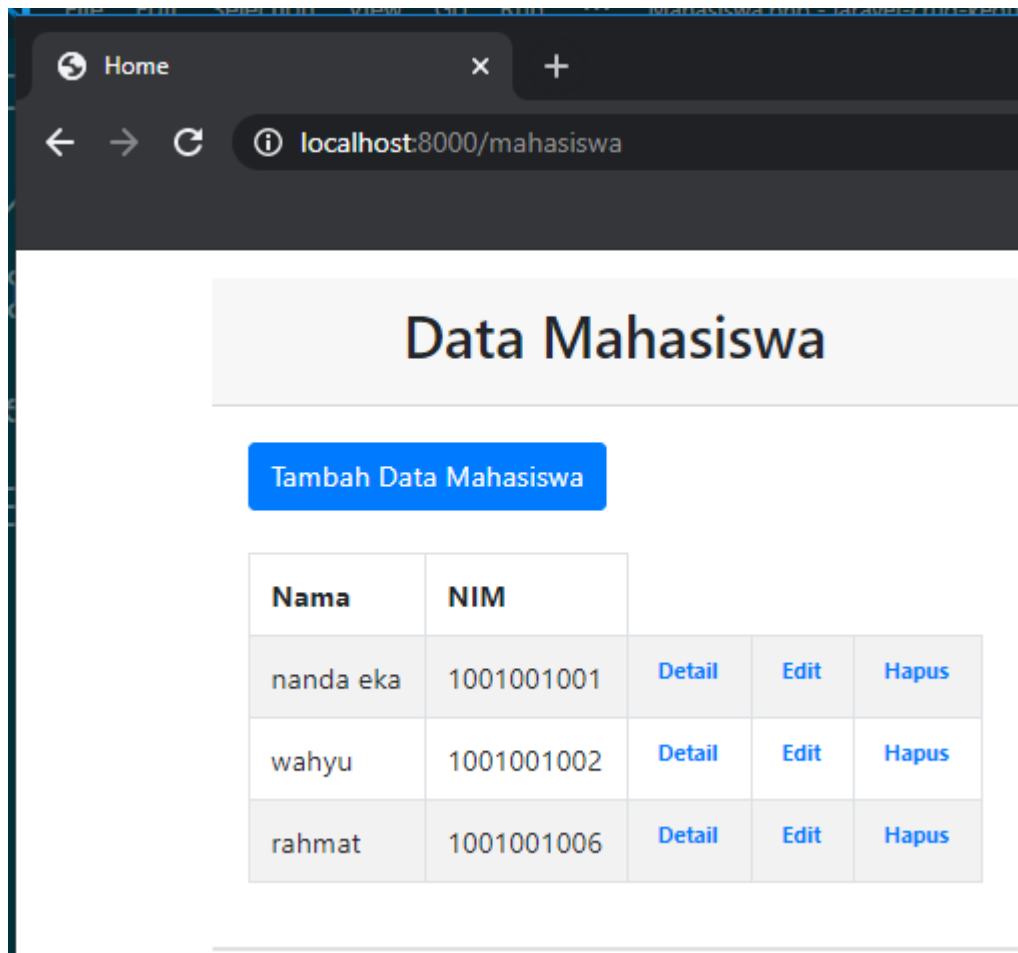
c. Memasukkan Data (Create) ke Database

1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre>24 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 25</pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre>15 public function tambah() 16 { 17     return view('tambah'); 18 } 19 } 20</pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. Kita lakukan copy paste dari tambah.blade.php di Bagian 1</p>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');</pre> <p>Keterangan:</p> <ul style="list-style-type: none"><li>- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php</li></ul>
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre>19 public function simpan(Request \$request) 20 { 21     Mahasiswa::create([ 22         'nama' =&gt; \$request-&gt;namamhs, 23         'nim' =&gt; \$request-&gt;nimmhs, 24         'email' =&gt; \$request-&gt;emailmhs, 25         'jurusan' =&gt; \$request-&gt;jurusanmhs 26     ]); 27     return redirect('/mahasiswa'); 28 } 29 } 30</pre> <p>Keterangan:</p> <ul style="list-style-type: none"><li>- Variabel \$request untuk menerima data yang akan ditambahkan ke database</li></ul>

	<p>- Line 23-28 merupakan fitur eloquent menggunakan fungsi create() untuk insert data ke tabel mahasiswa</p>										
6	<p>Karena kita menggunakan fungsi create pada Controller, maka butuh ditambahkan code pada Line 10 yang disebut Mass Assignment pada model Mahasiswa.php. Mass Assignment digunakan untuk memfilter kolom mana yang boleh dan tidak boleh diinput.</p> <pre> app &gt; Mahasiswa.php &gt; PHP Intelephense &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 } 12 </pre>										
7	<p>Kembali coba jalankan localhost:8000 dan pilih tombol ‘Tambah Data Mahasiswa’, maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.</p>  <p>The screenshot shows a web browser window with the address bar displaying 'localhost:8000/mahasiswa/tambah'. The page title is 'Tambah Data Mahasiswa'. The form contains the following data:</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Nama</td> <td>rahmat</td> </tr> <tr> <td>NIM</td> <td>1001001006</td> </tr> <tr> <td>E-Mail</td> <td>rahmat@gmail.com</td> </tr> <tr> <td>Jurusan</td> <td>teknik mesin</td> </tr> </tbody> </table> <p>Buttons: 'Kembali' (red) and 'Tambah Data' (blue).</p>	Field	Value	Nama	rahmat	NIM	1001001006	E-Mail	rahmat@gmail.com	Jurusan	teknik mesin
Field	Value										
Nama	rahmat										
NIM	1001001006										
E-Mail	rahmat@gmail.com										
Jurusan	teknik mesin										



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



d. Melihat Detail Data dari Database

1. 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');
```

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

```
29     public function detail($id)
30     {
31         $mahasiswa = Mahasiswa::find($id);
32         return view('detail', ['mahasiswa' => $mahasiswa]);
33     }
34 }
35
```

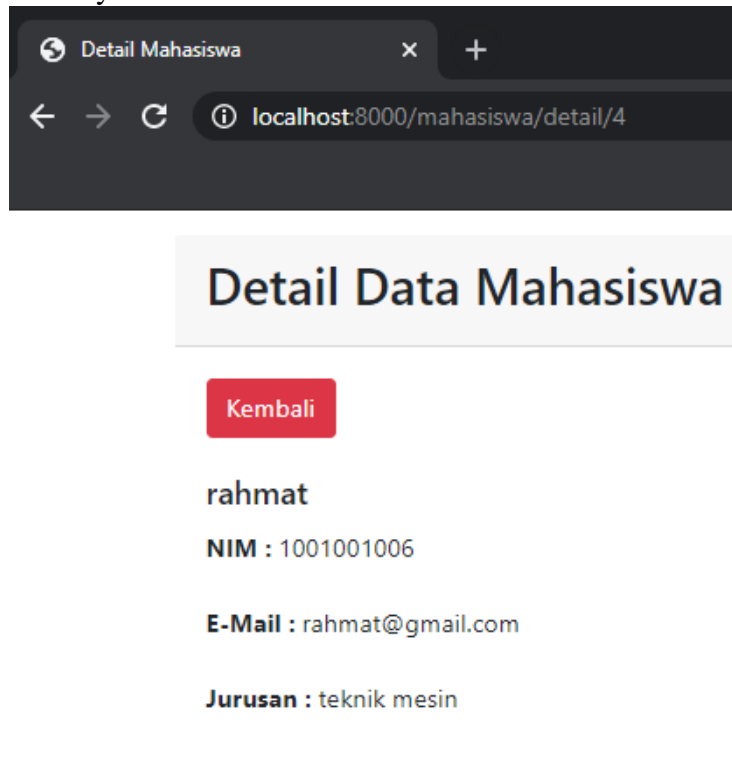
3

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

5

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



e. Mengubah Data (Update) dari Database

1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre>Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');</pre>
2	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre>public function edit(\$id) {     \$mahasiswa = Mahasiswa::find(\$id);     return view('edit', ['mahasiswa' =&gt; \$mahasiswa]); }</pre> <p>Keterangan :</p> <ul style="list-style-type: none"><li>- Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih</li></ul>
3	<p>Kemudian buatlah view edit.blade.php yang berisi form untuk mengubah data pada folder resources/views. View edit juga mengaplikasikan master.blade.php.</p> <pre>resources &gt; views &gt; edit.blade.php &gt; form 1  @extends('master') 2 3  &lt;!-- isi file --&gt; 4  @section('title','Edit Data') 5 6  &lt;!-- isi bagian judul halaman --&gt; 7  @section('judul_halaman', 'Edit Data Mahasiswa') 8 9  &lt;!-- isi bagian konten --&gt; 10 @section('konten') 11 &lt;a href="/mahasiswa" class="btn btn-danger"&gt;Kembali&lt;/a&gt; 12 &lt;br /&gt; 13 &lt;br /&gt; 14 15 &lt;form action="/mahasiswa/update/{{ \$mahasiswa-&gt;id }}" method="post"&gt; 16     {{ csrf_field() }} 17     &lt;input type="hidden" name="id" value="{{ \$mahasiswa-&gt;id }}"&gt;&lt;br /&gt; 18     &lt;div class="form-group"&gt; 19         &lt;label for="namamhs"&gt;Nama&lt;/label&gt; 20         &lt;input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mahasiswa-&gt;nama }}"&gt;&lt;br /&gt; 21     &lt;/div&gt; 22     &lt;div class="form-group"&gt; 23         &lt;label for="namamhs"&gt;NIM&lt;/label&gt; 24         &lt;input type="number" class="form-control" required="required" name="nimmhs" value="{{ \$mahasiswa-&gt;nim }}"&gt;&lt;br /&gt; 25     &lt;/div&gt; 26     &lt;div class="form-group"&gt; 27         &lt;label for="namamhs"&gt;E-Mail&lt;/label&gt; 28         &lt;input type="email" class="form-control" required="required" name="emailmhs" value="{{ \$mahasiswa-&gt;email }}"&gt;&lt;br /&gt; 29     &lt;/div&gt; 30     &lt;div class="form-group"&gt; 31         &lt;label for="namamhs"&gt;Jurusan&lt;/label&gt; 32         &lt;input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mahasiswa-&gt;jurusan }}"&gt;&lt;br /&gt; 33     &lt;/div&gt; 34     &lt;button type="submit" name="edit" class="btn btn-primary float-right"&gt;Simpan Data&lt;/button&gt; 35 &lt;/form&gt; 36 @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 <code>action="/mahasiswa/update/{{{\$mahasiswa-&gt;id}}}"</code> yang menunjukkan routes <code>/mahasiswa/update/{id}</code> dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController</li> </ul>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes <code>/mahasiswa/update/{id}</code>. 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 update pada MahasiswaController untuk menyimpan data yang diubah ke database</p> <pre>public function update(\$id, 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 <code>\$request</code> 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. Cobalah untuk mengedit data.</p>

← → ↻ ⓘ localhost:8000/mahasiswa

## Data Mahasiswa

Tambah Data Mahasiswa

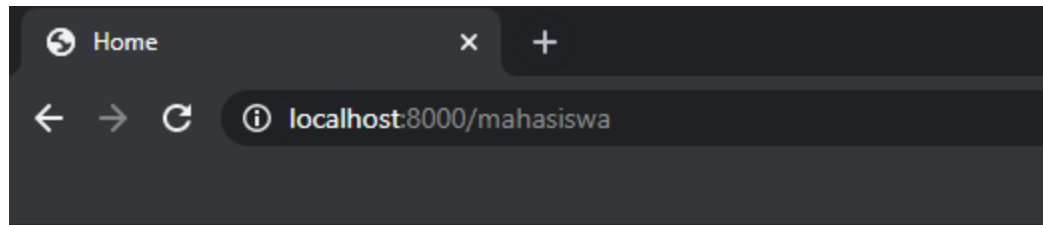
Nama	NIM			
nanda eka	1001001001	Detail	Edit	Hapus
wahyu	1001001002	Detail	Edit	Hapus
rahmat jaya	1001001006	Detail	Edit	Hapus

f. Menghapus Data (Delete) dari Database

1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre>Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');</pre>
2	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p> <pre>     public function hapus(\$id)     {         \$mahasiswa = Mahasiswa::find(\$id);         \$mahasiswa-&gt;delete();         return redirect('/mahasiswa');     } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih</li> </ul>

3

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



## 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>