

LAPORAN JOOBSHEET 10

(Membuat CRUD Menggunakan Laravel)

**Disusun sebagai
MATA KULIAH :
Pemrograman Web Lanjut**



TI-2B

OLEH

WILDAN DAWAM BASH (1841720166)

PROGAM STUDI D-IV TEKNIK INFORMATIKA

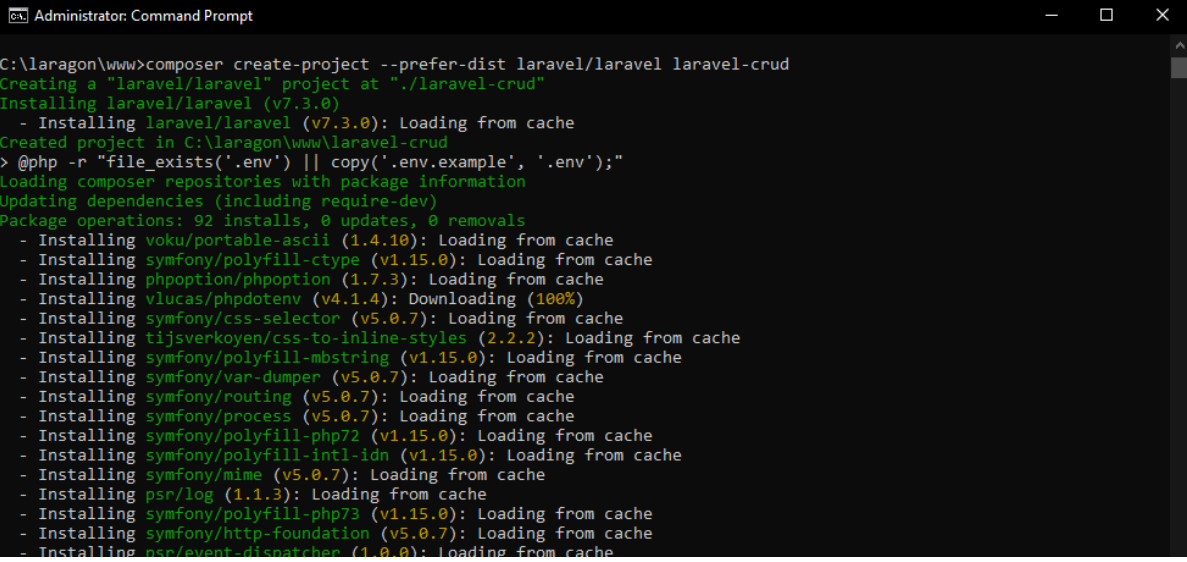
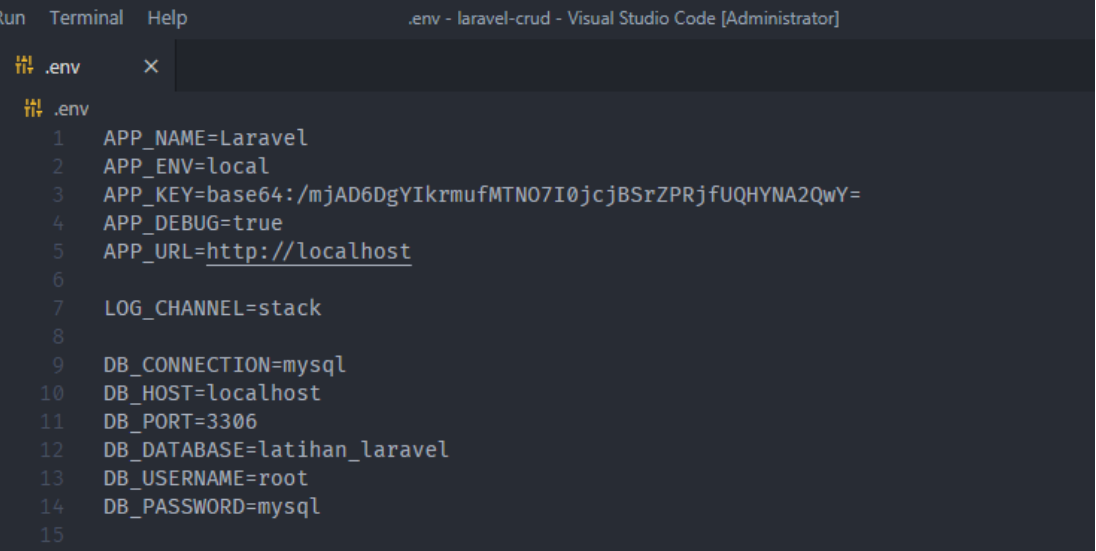
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

Bagian 1 – Membuat CRUD di Laravel menggunakan Query Builder

a. Konfigurasi Database dan Pembuatan Tabel di MySQL

No	Keterangan
1	<p>Buatlah project Laravel baru dengan nama laravel-crud.</p> 
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> 

- 3 Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database latihan_laravel.

Server: localhost:3306 » Basis data: latihan_laravel » Tabel: mahasiswa

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Pelacak

Struktur tabel Tampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	nim	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	email	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	jurusan	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

- 4 Isilah beberapa data pada tabel mahasiswa tersebut.

+ Opsi

				id	nama	nim	email	jurusan
<input type="checkbox"/>	Ubah	Salin	Hapus	1	nanda	1001001001	nandan@gmail.com	Teknik Informatika
<input type="checkbox"/>	Ubah	Salin	Hapus	2	wahyu	1001001002	wahyu@gmail.com	Teknik Elektro

Pilih Semua Dengan pilihan: Ubah Salin Hapus Ekspor

b. Menampilkan Data dari Database

No	Keterangan
1	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 application.
11 | routes are loaded by the RouteServiceProvider within a group
12 | contains the "web" middleware group. Now create something great.
13 |
14 */
15
16 // Route::get('/', function () {
17 //     return view('welcome');
18 // });
19
20 Route::get('/', 'MahasiswaController@index');
```

- 2 Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan

cd laravel-crud

php artisan make:controller MahasiswaController

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\laragon\www\laravel-crud> php artisan make:controller MahasiswaController
Controller created successfully.
PS C:\laragon\www\laravel-crud> 
```

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

```
app > Http > Controllers > MahasiswaController.php > ...
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
12         $mahasiswa = DB::table('mahasiswa')->get();
13
14         return view('index', ['mahasiswa' => $mahasiswa]);
15     }
16 }
17
```

- 4 Membuat **template blade** dengan nama **master.blade.php** pada folder **resources/views**

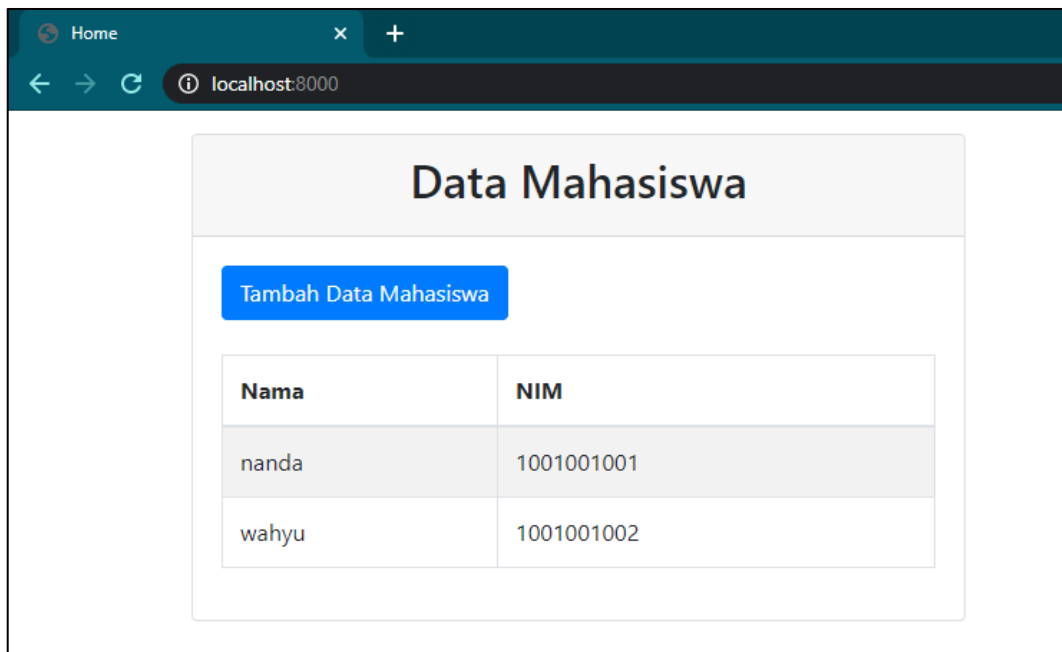
```
resources > views > master.blade.php > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title> @yield('title')</title>
8  </head>
9  <body>
10     <div class="container">
11         <div class="row mt-3">
12             <div class="col-md-6">
13                 <div class="card">
14                     <div class="card-header text-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>
```

- 5 buatlah 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
28
```

- 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

No	Keterangan
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>public function tambah(){ return view('tambah'); }</pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. View tambah juga mengaplikasikan master.blade.php.</p> <pre>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 Kembali 9
 10
 11 <form action="/mahasiswa/simpan" method="POST"> 12 {{ csrf() }} 13 <div class="form-group"> 14 <label for="namamhs">Nama</label> 15 <input type="text" class="form-control" required="required" name="namamhs">
 16 </div> 17 <div class="form-group"> 18 <label for="nimmhs">NIM</label> 19 <input type="number" class="form-control" required="required" name="nimmhs">
 20 </div> 21 <div class="form-group"> 22 <label for="emailmhs">E-mail</label> 23 <input type="email" class="form-control" required="required" name="emailmhs">
 24 </div> 25 <div class="form-group"> 26 <label for="jurusanmhs">Jurusan</label> 27 <input type="text" class="form-control" required="required" name="jurusanmhs">
 28 </div> 29 <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button> 30 </form> 31 @endsection 32</pre>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>23 24 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 25</pre>

- 5 Buat method simpan pada MahasiswaController untuk menyimpan data ke database.

```
public function simpan(Request $request){
    DB::table('mahasiswa')->insert([
        'nama' => $request->namamhs,
        'nim' => $request->nimmhs,
        'email' => $request->emailmhs,
        'jurusan' => $request->jurusanmhs,
    ]);
    return redirect('/mahasiswa');
}
```

- 6 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
Wildan Dawam Bash

NIM
1841720166

E-mail
wildandawambash@gmail.com

Jurusan
Teknik Informatika

Tambah Data

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

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM
nanda	1001001001
wahyu	1001001002
Wildan Dawam Bash	1841720166

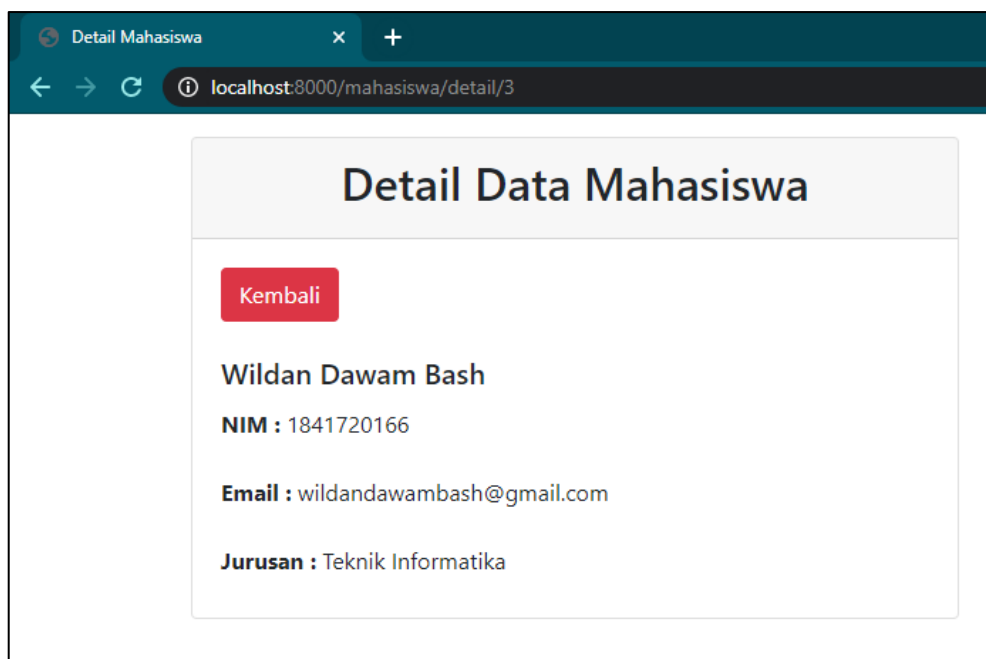
d. Melihat Detail dari Database

No	Keterangan
1	<p>Buatlah tombol untuk melihat detail data (Line 25) pada file index.blade.php di folder resources/views</p> <pre>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 Tambah Data Mahasiswa 9
 10
 11 <table class="table table-bordered table-hover table-striped"> 12 <thead> 13 <tr> 14 <th>Nama</th> 15 <th>NIM</th> 16 <th></th> 17 </tr> 18 </thead> 19 <tbody> 20 @foreach (\$mahasiswa as \$mhs) 21 <tr> 22 <td>{{ \$mhs->nama }}</td> 23 <td>{{ \$mhs->nim }}</td> 24 <td> 25 id }}" class="badge badge-info">Detail 26 </td> 27 </tr> 28 @endforeach 29 </tbody> 30 </table> 31 @endsection 32</pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre>25 26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');</pre>
3	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p> <pre>30 31 public function detail(\$id){ 32 \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); 33 34 return view('detail', ['mahasiswa' => \$mahasiswa]); 35 }</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 > ...
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> Email : </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
29
30
```

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



e. Mengubah Data(Update) dari Database

No	Keterangan
1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views</p> <pre>resources > views > index.blade.php > ... 7 @section('konten') 8 Tambah Data Mahasiswa 9
 10
 11 <table class="table table-bordered table-hover table-striped"> 12 <thead> 13 <tr> 14 <th>Nama</th> 15 <th>NIM</th> 16 <th></th> 17 </tr> 18 </thead> 19 <tbody> 20 @foreach (\$mahasiswa as \$mhs) 21 <tr> 22 <td>{{ \$mhs->nama }}</td> 23 <td>{{ \$mhs->nim }}</td> 24 <td> 25 id }}" class="badge badge-info">Detail 26 id }}" class="badge badge-warning">Edit 27 </td> 28 </tr> 29 @endforeach 30 </tbody> 31 </table> 32 @endsection 33</pre>
2	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre>27 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> <pre>public function edit(\$id){ \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); return view('edit', ['mahasiswa' => \$mahasiswa]); }</pre>

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

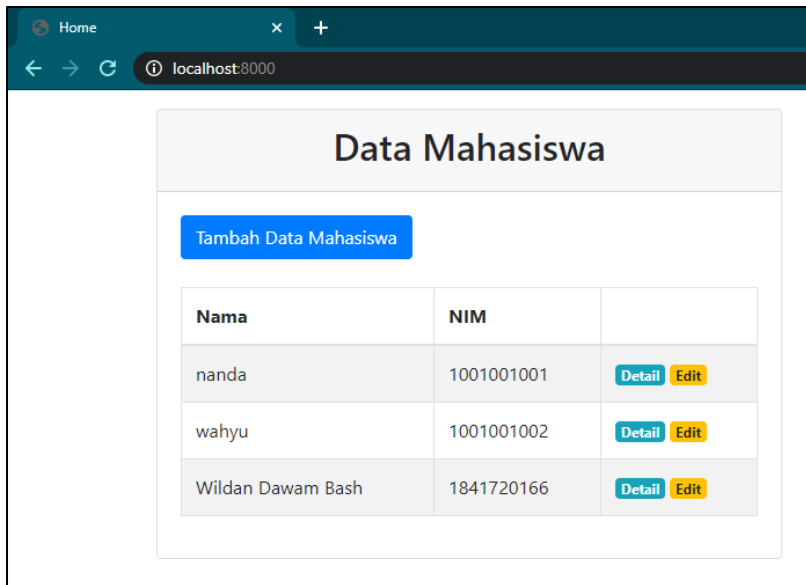
- 5 Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update. Oleh karena itu, kita buat terlebih dahulu route tersebut.

```
29
30 Route::post('/mahasiswa/update', 'MahasiswaController@update');
31
```

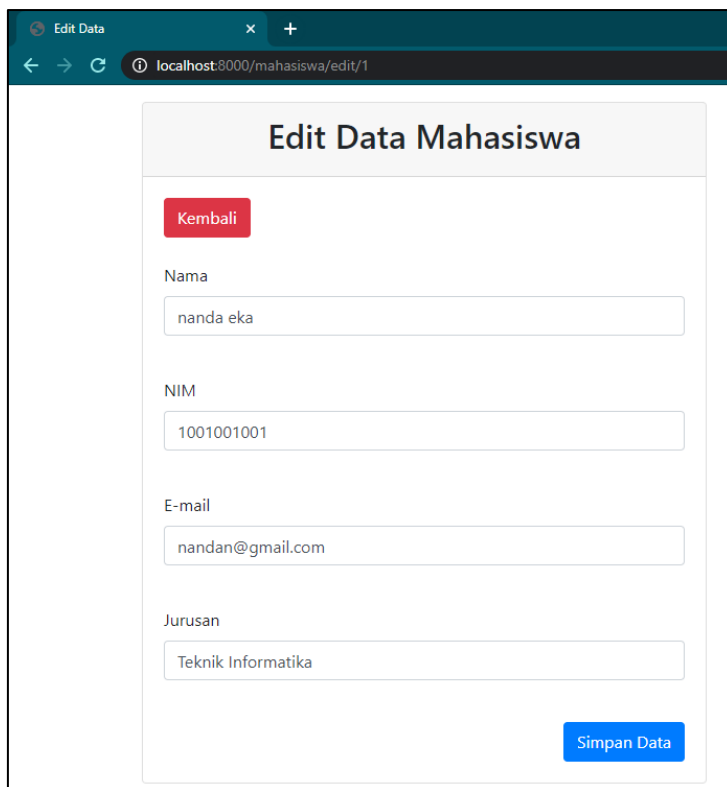
- 6 Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.

```
42
43 public function update(Request $request){{
44     DB::table('mahasiswa')->where('id', $request->id)->update([
45         'nama' => $request->namamhs,
46         'nim' => $request->nimmhs,
47         'email' => $request->emailmhs,
48         'jurusan' => $request->jurusanmhs,
49     ]);
50     return redirect('/mahasiswa');
51 }
52
```

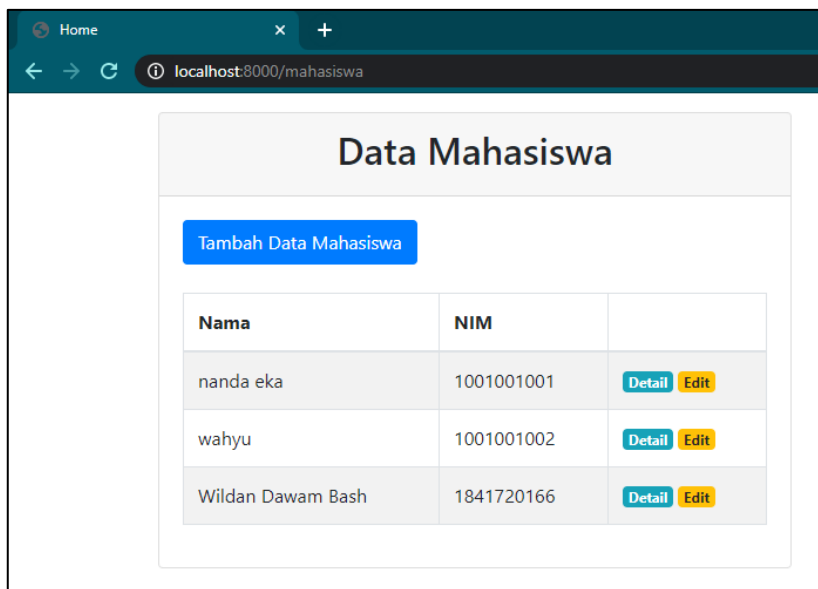
- 7 Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru.



Klik edit di data pertama, kita akan mengubah namanya.



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



f. Menghapus Data(Delete) dari database

No	Keterangan
1	<p>Buatlah tombol untuk menghapus data (Line 30) pada file index.blade.php di folder resources/views</p> <pre> <tbody> @foreach (\$mahasiswa as \$mhs) <tr> <td>{{ \$mhs->nama }}</td> <td>{{ \$mhs->nim }}</td> <td> id }}" class="badge badge-info">Detail id }}" class="badge badge-warning">Edit id }}" class="badge badge-danger">Hapus </td> </tr> @endforeach </tbody> </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> <pre> public function hapus(\$id){ DB::table('mahasiswa')->where('id', \$id)->delete(); return redirect('/mahasiswa'); } </pre>

- 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	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
Wildan Dawam Bash	1841720166	Detail Edit Hapus
Mira	1001001003	Detail Edit Hapus

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
Wildan Dawam Bash	1841720166	Detail Edit Hapus

Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

a. Konfigurasi Database

No	Keterangan
1	<p>Buatlah project Laravel baru dengan nama laravel-crud-kedua.</p> 
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> 
3	<p>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</p> 

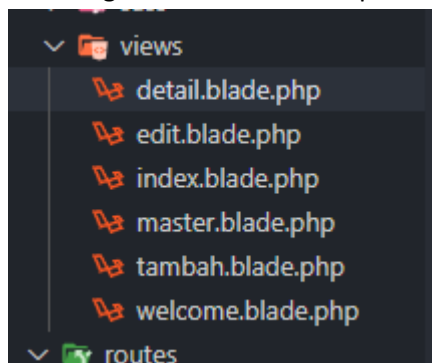
b. Menampilkan Data dari Database

No	Keterangan
1	<p>buatlah route pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p> <pre>21 Route::get('/', 'MahasiswaController@index'); 22 Route::get('/mahasiswa', 'MahasiswaController@index');</pre>
2	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan</p> <p>cd laravel-crud-kedua</p> <p>php artisan make:model Mahasiswa</p> <pre>PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved. Try the new cross-platform PowerShell https://aka.ms/pscore6 PS C:\laragon\www\laravel-crud-kedua> php artisan make:model Mahasiswa Model created successfully. PS C:\laragon\www\laravel-crud-kedua></pre>
3	<p>Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.</p> <pre>web.php Mahasiswa.php X app > Mahasiswa.php > ... 1 <?php 2 3 namespace App; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class Mahasiswa extends Model 8 { 9 protected \$table = "mahasiswa"; 10 } 11</pre>
4	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan</p> <p>php artisan make:controller MahasiswaController</p> <pre>PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PS C:\laragon\www\laravel-crud-kedua> php artisan make:controller MahasiswaController Controller created successfully. PS C:\laragon\www\laravel-crud-kedua></pre>

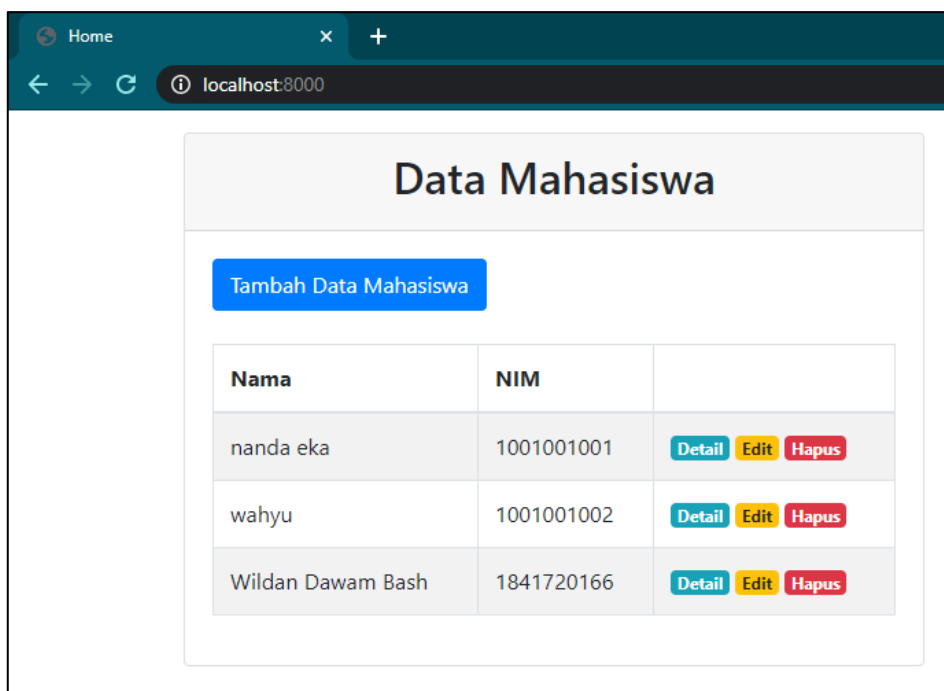
- 5 Buat method index pada MahasiswaController.php pada folder **app/Http/Controllers**

```
web.php  Mahasiswa.php  MahasiswaController.php X
app > Http > Controllers > MahasiswaController.php > ...
1  <?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' => $mahasiswa]);
14     }
15 }
16
```

- 6 buatlah 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



c. Memasukkan Data (Create) ke Database

No	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre data-bbox="199 309 1157 387"> 23 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 data-bbox="215 533 566 633"> public function tambah(){ return view('tambah'); } </pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views.</p> <pre data-bbox="180 795 1220 1491"> 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 Kembali 9
 10
 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" required="required" name="namamhs">
 16 </div> 17 <div class="form-group"> 18 <label for="nimmhs">NIM</label> 19 <input type="number" class="form-control" required="required" name="nimmhs">
 20 </div> 21 <div class="form-group"> 22 <label for="emailmhs">E-mail</label> 23 <input type="email" class="form-control" required="required" name="emailmhs">
 24 </div> 25 <div class="form-group"> 26 <label for="jurusanmhs">Jurusan</label> 27 <input type="text" class="form-control" required="required" name="jurusanmhs">
 28 </div> 29 <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button> 30 </form> 31 @endsection 32 </pre>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre data-bbox="172 1630 1201 1686"> 25 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); </pre>
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre data-bbox="180 1780 837 2114"> public function simpan(Request \$request){ Mahasiswa::create(['nama' => \$request->namamhs, 'nim' => \$request->nimmhs, 'email' => \$request->emailmhs, 'jurusan' => \$request->jurusanmhs,]); return redirect('/mahasiswa'); } </pre>

- 6 Karena kita menggunakan fungsi create pada Controller, maka butuh ditambahkan code pada Line 10 yang disebut Mass Assignment pada model Mahasiswa.php. Mass Assignment digunakan untuk memfilter kolom mana yang boleh dan tidak boleh diinput.

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

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

Tambah Data Mahasiswa

Kembali

Nama
rahmat

NIM
1001001005

E-mail
rahmat@gmail.com

Jurusan
Teknik Geofisika

Tambah Data

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
Wildan Dawam Bash	1841720166	Detail Edit Hapus
rahmat	1001001005	Detail Edit Hapus

d. Melihat Detail Data Dari Database

No	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre>26 27 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 28</pre>
2	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p> <pre>public function detail(\$id){ \$mahasiswa = Mahasiswa::find(\$id); return view('detail', ['mahasiswa' => \$mahasiswa]); }</pre>
3	<p>Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php.</p> <pre>web.php MahasiswaController.php detail.blade.php X Mahasiswa.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 Kembali 9 10
 11
 12 13 <h5 class="card-title">{{ \$mahasiswa->nama }}</h5> 14 <p class="card-text"> 15 <label for=""> NIM : </label> 16 {{ \$mahasiswa->nim }} 17 </p> 18 <p class="card-text"> 19 <label for=""> Email : </label> 20 {{ \$mahasiswa->email }} 21 </p> 22 <p class="card-text"> 23 <label for=""> Jurusan : </label> 24 {{ \$mahasiswa->jurusan }} 25 </p> 26 27 @endsection 28</pre>
4	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p> 

e. Mengubah Data (Update) dari Database

No	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre>28 29 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 30</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' => \$mahasiswa]); }</pre>
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>7 @section('konten') 8 Kembali 9
 10
 11 <form action="/mahasiswa/update/{{ \$mahasiswa->id }}" method="POST"> 12 {{ csrf_field() }} 13 <input type="hidden" name="id" value="{{ \$mahasiswa->id }}"> 14 <div class="form-group"> 15 <label for="namamhs">Nama</label> 16 <input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mahasiswa->nama }}">
 17 </div> 18 <div class="form-group"> 19 <label for="nimhs">NIM</label> 20 <input type="number" class="form-control" required="required" name="nimhs" value="{{ \$mahasiswa->nim }}">
 21 </div> 22 <div class="form-group"> 23 <label for="emailmhs">E-mail</label> 24 <input type="email" class="form-control" required="required" name="emailmhs" value="{{ \$mahasiswa->email }}">
 25 </div> 26 <div class="form-group"> 27 <label for="jurusanmhs">Jurusan</label> 28 <input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mahasiswa->jurusan }}">
 29 </div> 30 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button> 31 </form> 32 33 @endsection</pre>
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update/{id}. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>31 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update'); 32</pre>

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

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

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

Sebelum update

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
Wildan Dawam Bash	1841720166	Detail Edit Hapus
rahmat	1001001005	Detail Edit Hapus

Proses update

Edit Data Mahasiswa	
Kembali	
Nama	<input type="text" value="rahmat reki"/>
NIM	<input type="text" value="1001001005"/>
E-mail	<input type="text" value="rahmat@gmail.com"/>
Jurusan	<input type="text" value="Teknik Geofisika"/>
Simpan Data	

Selesai Update data nama

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
Wildan Dawam Bash	1841720166	Detail Edit Hapus
rahmat reki	1001001005	Detail Edit Hapus

f. Menghapus Data(Delete) dari Database

No	Keterangan
1	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre>32 33 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 34</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->delete(); return redirect('/mahasiswa'); }</pre>

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

Data sebelum dihapus

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
Wildan Dawam Bash	1841720166	Detail Edit Hapus
rahmat reki	1001001005	Detail Edit Hapus

Data setelah dihapus

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
Wildan Dawam Bash	1841720166	Detail Edit Hapus