

LAPORAN PRAKTIKUM Pemrograman Web Lanjut

PRAKTIKUM 10



Oleh :

**MARTIN AMANU KHUSNA / NIM : 1841720138
KELAS : TI-2B**

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

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

a. Konfigurasi Database dan Pembuatan Tabel di MySQL

NO	STEP																																																						
1.	<p>Buatlah project Laravel baru dengan nama laravel-crud. Buka command prompt, tuliskan perintah berikut.</p> <pre>C:\xampp\htdocs>laravel new laravel-crud Crafting application... Loading composer repositories with package information Installing dependencies (including require-dev) from lock file Package operations: 92 installs, 0 updates, 0 removals Discovered Package: nunomaduro/collision Package manifest generated successfully. Application ready! Build something amazing.</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:ENoes8M2LMMEyv8oAxRtacrpfOWrLv/8gYGObMgWMD4= 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=</pre>																																																						
3.	<p>Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database latihan_laravel.</p> <table><tr><th>#</th><th>Nama</th><th>Jenis</th><th>Penyortiran</th><th>Atribut</th><th>Tak Ternilai</th><th>Bawaan</th><th>Komentar</th><th>Ekstra</th></tr><tr><td><input type="checkbox"/> 1</td><td>id</td><td>int(11)</td><td></td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td>AUTO_INCREMENT</td></tr><tr><td><input type="checkbox"/> 2</td><td>nama</td><td>varchar(100)</td><td>latin1_swedish_ci</td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td></td></tr><tr><td><input type="checkbox"/> 3</td><td>nim</td><td>varchar(10)</td><td>latin1_swedish_ci</td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td></td></tr><tr><td><input type="checkbox"/> 4</td><td>email</td><td>varchar(50)</td><td>latin1_swedish_ci</td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td></td></tr><tr><td><input type="checkbox"/> 5</td><td>jurusan</td><td>varchar(20)</td><td>latin1_swedish_ci</td><td></td><td>Tidak</td><td>Tidak ada</td><td></td><td></td></tr></table>	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	<input type="checkbox"/> 1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	<input type="checkbox"/> 2	nama	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada			<input type="checkbox"/> 3	nim	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada			<input type="checkbox"/> 4	email	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada			<input type="checkbox"/> 5	jurusan	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada		
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra																																															
<input type="checkbox"/> 1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT																																															
<input type="checkbox"/> 2	nama	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada																																																	
<input type="checkbox"/> 3	nim	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada																																																	
<input type="checkbox"/> 4	email	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada																																																	
<input type="checkbox"/> 5	jurusan	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada																																																	
4.	<p>Isilah beberapa data pada tabel mahasiswa tersebut.</p> <table><tr><th></th><th></th><th></th><th></th><th>id</th><th>nama</th><th>nim</th><th>email</th><th>jurusan</th></tr><tr><td><input type="checkbox"/></td><td>Ubah</td><td>Salin</td><td>Hapus</td><td>1</td><td>Nanda</td><td>1001001001</td><td>nandan@gmail.com</td><td>teknik informatika</td></tr><tr><td><input type="checkbox"/></td><td>Ubah</td><td>Salin</td><td>Hapus</td><td>2</td><td>wahyu</td><td>1001001002</td><td>wahyu@gmail.com</td><td>teknik elektro</td></tr></table>					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																											
				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																																															

b. Menampilkan Data dari Database

NO	STEP
1.	<p>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.</p> <pre data-bbox="280 436 1130 1087">routes > web.php 1 <?php 2 3 use Illuminate\Support\Facades\Route; 4 5 /* 6 ----- 7 Web Routes 8 ----- 9 10 Here is where you can register web routes for your application. These 11 routes are loaded by the RouteServiceProvider within the routes/web.php 12 file and are grouped within the "web" middleware group. Now create 13 your first web route: 14 15 Route::get('/', function () { 16 return view('welcome'); 17 }); 18 */ 19 20 Route::get('/', 'MahasiswaController@index');</pre> <p>Ketika / maka dijalankan method index pada controller mahasiswacontroller.</p>
2.	<p>Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan.</p> <pre data-bbox="280 1220 1401 1297">C:\xampp\htdocs\laravel-crud>php artisan make:controller MahasiswaController Controller created successfully.</pre>

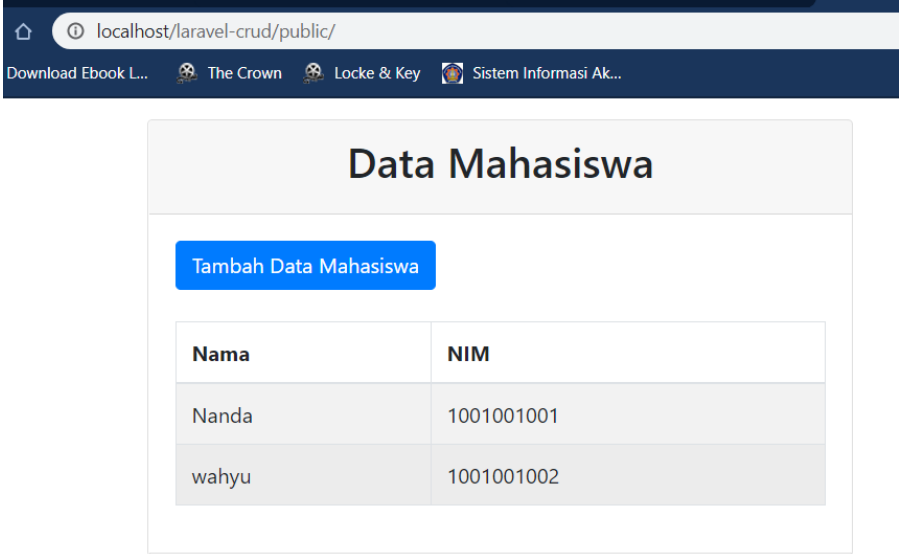
3. Buat method index pada MahasiswaController.php pada folder app/Http/Controllers.

```
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         //mengambil data dari tabel mahasiswa
13         $mahasiswa = DB::table('mahasiswa')->get();
14
15         //mengirim data mahasiswa ke view index
16         return view('index',['mahasiswa' => $mahasiswa]);
17     }
18 }
```

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 href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/boo
6      <title> @yield('title') </title>
7  </head>
8  <body>
9      <div class="container">
10         <div class="row mt-3">
11             <div class="col-md-6">
12                 <div class="card">
13                     <div class="card-header text-center">
14                         <!-- bagian judul halaman -->
15                         <h2> @yield('judul_halaman') </h2>
16                     </div>
17                     <div class="card-body">
18                         <!-- bagian konten blog -->
19                         @yield('konten')
20                     </div>
21                 </div>
22             </div>
23         </div>
24     </div>
25 </body>
26 </html>
```

	<p>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</p>
5.	<p>Sekarang buat file index.blade.php yang menerapkan template master.blade.php dan akan digunakan untuk menampilkan data.</p> <pre> resources > views > index.blade.php 1 @extends('master') 2 3 <!-- isi title --> 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 Tambah Data Mahasi 12
 13
 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 </pre> <p>Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan blade lain yaitu master.blade.php</p> <p>- Line 4 : @section('title', 'Home') berarti bahwa file index mengisi @yield('title') pada master dengan 'Home'</p> <p>- Line 7 : @section('judul_halaman', 'Data Mahasiswa') berarti bahwa file index mengisi @yield('judul_halaman') pada master dengan 'Home'</p> <p>- Line 10-30 : mengisi yield(@konten), karena terdapat banyak baris pada diawali dengan @section('konten') dan diakhiri dengan @endsection</p> <p>- Line 24-25 menampilkan data mahasiswa dengan kolom nama dan nim</p>

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

c. Memasukkan Data (Create) ke Database

NO	STEP
1.	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController.</p> <pre> 15 16 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 17 </pre>
2.	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> } public function tambah(){ //memanggil view 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 <!-- isi title --> 3 @section('title', 'Tambah Data') 4 5 <!-- isi bagian judul halaman --> 6 @section('judul_halaman', 'Tambah Data Mahasiswa') 7 8 <!-- isi bagian konten --> 9 @section('konten') 10 Kembali 11
 12
 13 <form action="/mahasiswa/simpan" method="post"> 14 {{ csrf_field() }} 15 <div class="form-group"> 16 <label for="namamhs">Nama</label> 17 <input type="text" class="form-control" required="required" name="namamhs"> 18 </div> 19 <div class="form-group"> 20 <label for="nimhs">NIM</label> 21 <input type="number" class="form-control" required="required" name="nimhs"> 22 </div> 23 <div class="form-group"> 24 <label for="emailmhs">E-mail</label> 25 <input type="email" class="form-control" required="required" name="emailmhs"> 26 </div> 27 <div class="form-group"> 28 <label for="jurusanmhs">Jurusan</label> 29 <input type="text" class="form-control" required="required" name="jurusanmhs"> 30 </div> 31 <button type="submit" name="tambah" class="btn btn-primary float-right">Simpan</button> 32 </form> 33 @endsection</pre>
4.	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>19 20 Route::get('/mahasiswa/simpan', 'MahasiswaController@simpan'); 21</pre> <p>Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php</p>
5.	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre>18 public function simpan(Request \$request){ 19 //insert data ke table mahasiswa 20 DB::table('mahasiswa')->insert([21 'nama' => \$request->namamhs, 22 'nim' => \$request->nimmhs, 23 'email' => \$request->emailmhs, 24 'jurusan' => \$request->jurusanmhs 25]); 26 27 }</pre> <p>Variabel \$request untuk menerima data yang akan ditambahkan ke database</p>

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.

Tambah Data Mahasiswa

[Kembali](#)

Nama

dinda

NIM

1001001003

E-mail

dinda@gmail.com

Jurusan

teknik industri

[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
dinda	1001001003

d. Melihat Detail Data dari Database

N O	STEP
1.	<p>Buatlah tombol untuk melihat detail data pada file index.blade.php di folder resources/views.</p> <pre data-bbox="277 359 1365 447"><td>id }}" class="badge badge-info">Detail</td></tr></pre>
2.	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre data-bbox="277 541 1365 653">21 22 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 23</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 data-bbox="277 758 1276 1041">public function detail(\$id){ //mengambil data mahasiswa berdasarkan id yang dipilih \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); //kirim data mahasiswa yang diambil ke view edit.blade.php return view('detail',['mahasiswa' => \$mahasiswa]); }</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 data-bbox="277 1136 995 1860">resources > views > detail.blade.php 1 @extends('master') 2 <!-- isi title --> 3 @section('title', 'Detail Mahasiswa') 4 5 <!-- isi bagian judul halaman --> 6 @section('judul_halaman', 'Detail Data Mahasiswa') 7 8 <!-- isi bagian konten --> 9 @section('konten') 10 Kembali 11 12
 13
 14 15 @foreach(\$mahasiswa as \$mhs) 16 <h5 class="card-title"> {{ \$mhs->nama }} </h5> 17 <p class="card-text"> 18 <label for=""> NIM : </label> 19 {{ \$mhs->nim }} </p> 20 <p class="card-text"> 21 <label for=""> E-mail : </label> 22 {{ \$mhs->email }} </p> 23 <p class="card-text"> 24 <label for=""> Jurusan : </label> 25 {{ \$mhs->jurusan }} </p> 26 @endforeach 27 @endsection</pre>

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	Detail
wahyu	1001001002	Detail
dinda	1001001003	Detail

Detail Data Mahasiswa

Kembali

dinda

NIM : 1001001003

E-mail : dinda@gmail.com

Jurusan : teknik industri

e. Mengubah Data (Update) dari Database

NO	STEP
1.	<p>Buatlah tombol untuk mengubah data (Line 28) pada file index.blade.php di folder resources/views.</p> <pre>26 <td> 27 id }}" class="badge badge-info">Detail 28 id }}" class="badge badge-warning">Edit 29 </td></pre>
2.	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p>

3.	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p> <pre> 40 41 public function edit(\$id){ 42 //mengambil data mahasiswa berdasarkan id yang dipilih 43 \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); 44 //kirim data mahasiswa yang diambil ke view edit.blade.php 45 return view('edit',['mahasiswa' => \$mahasiswa]); 46 } 47 </pre> <p>Line 43 : query builder untuk mengambil data mahasiswa berdasarkan id yang dipilih</p>
4.	<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 > views > edit.blade.php 1 @extends('master') 2 <!-- isi title --> 3 @section('title', 'edit Data') 4 5 <!-- isi bagian judul halaman --> 6 @section('judul_halaman', 'Edit Data Mahasiswa') 7 8 <!-- isi bagian konten --> 9 @section('konten') 10 Kembali 11
 12
 13 @foreach(\$mahasiswa as \$mhs) 14 <form action="/mahasiswa/simpan" method="post"> 15 {{ csrf_field() }} 16 <input type="hidden" name="id" value="{{ \$mhs->id }}">
 17 <div class="form-group"> 18 <label for="namamhs">Nama</label> 19 <input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mhs->nama }}">
 20 </div> 21 <div class="form-group"> 22 <label for="nimhs">NIM</label> 23 <input type="number" class="form-control" required="required" name="nimhs" value="{{ \$mhs->nim }}">
 24 </div> 25 <div class="form-group"> 26 <label for="emailmhs">E-mail</label> 27 <input type="email" class="form-control" required="required" name="emailmhs" value="{{ \$mhs->email }}">
 28 </div> 29 <div class="form-group"> 30 <label for="jurusanmhs">Jurusan</label> 31 <input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mhs->jurusan }}">
 32 </div> 33 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button> 34 </form> 35 @endforeach 36 @endsection </pre>
5.	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 25 26 Route::post('/mahasiswa/update/{id}','MahasiswaController@update'); 27 </pre> <p>Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php</p>
6.	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p>

```

47
48 public function update(Request $request){
49     //update data mahasiswa
50     DB::table('mahasiswa')->where('id',$request->id)->update([
51         'nama' => $request->namamhs,
52         'nim' => $request->nimmhs,
53         'email' => $request->emailmhs,
54         'jurusan' => $request->jurusanmhs
55     ]);
56     return redirect('/mahasiswa');
57 }
58

```

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

Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM	
Nanda	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
dinda	1001001003	Detail Edit

Klik edit di data pertama, kita akan mengubah Namanya

Edit Data Mahasiswa

[Kembali](#)

Nama

dinda katrina k

NIM

1001001003

E-mail

dinda@gmail.com

Jurusan

teknik industri

[Simpan Data](#)

Setelah kita klik simpan Data, maka data pertama akan berubah

Data Mahasiswa

[Tambah Data Mahasiswa](#)

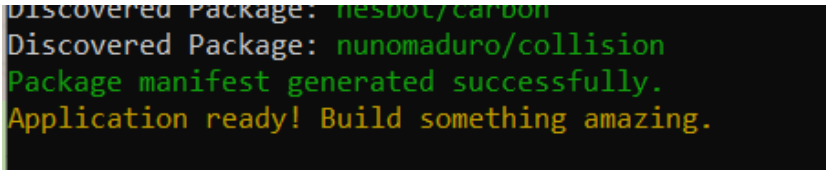
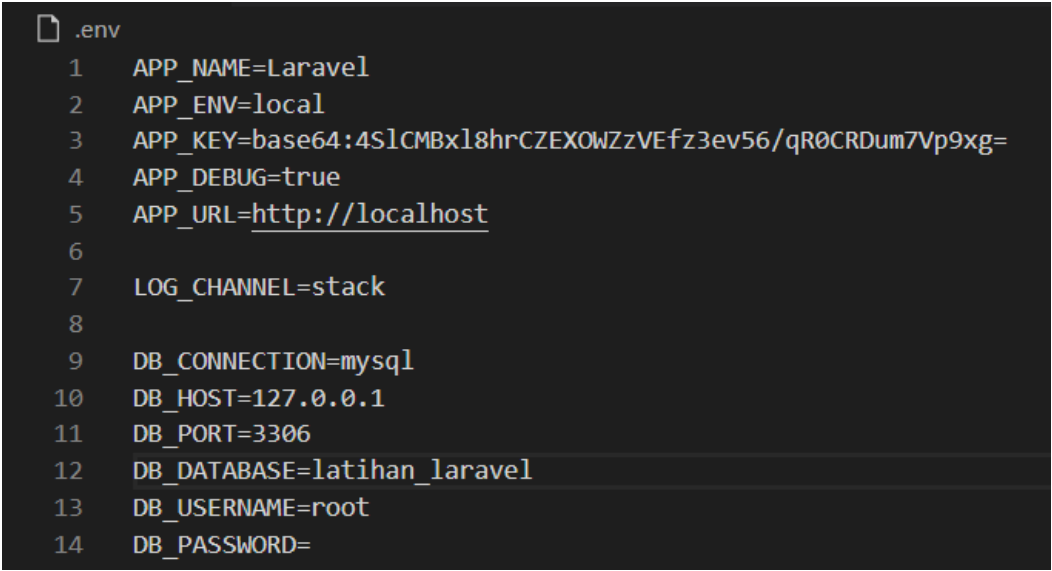
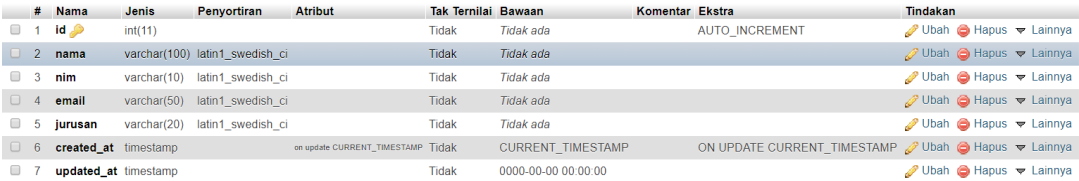
Nama	NIM	
Nanda	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
dinda katrina k	1001001003	Detail Edit Hapus

f. Menghapus Data (Delete) dari Database

NO	STEP																											
1.	<p>Buatlah tombol untuk menghapus data (Line 29) pada file index.blade.php di folder resources/views.</p> <pre>27 id }}" class="badge badge-info">Detail 28 id }}" class="badge badge-warning">Edit 29 id }}" class="badge badge-danger">Hapus</pre>																											
2.	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController.</p> <pre>27 28 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 29</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){ //menghapus data mahasiswa berdasarkan id yang dipilih DB::table('mahasiswa')->where('id',\$id)->delete(); return redirect('/mahasiswa'); }</pre>																											
4.	<p>Jalankan localhost:8000 dan pilih tombol ‘Hapus’, maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir.</p> <div><div><p>Data Mahasiswa</p><p>Tambah Data Mahasiswa</p><table><tr><th>Nama</th><th>NIM</th><th></th></tr><tr><td>Nanda</td><td>1001001001</td><td>Detail Edit Hapus</td></tr><tr><td>wahyu</td><td>1001001002</td><td>Detail Edit Hapus</td></tr><tr><td>dinda</td><td>1001001003</td><td>Detail Edit Hapus</td></tr><tr><td>dinda katrina</td><td>1001001003</td><td>Detail Edit Hapus</td></tr></table></div><div><p>Data Mahasiswa</p><p>Tambah Data Mahasiswa</p><table><tr><th>Nama</th><th>NIM</th><th></th></tr><tr><td>Nanda</td><td>1001001001</td><td>Detail Edit Hapus</td></tr><tr><td>wahyu</td><td>1001001002</td><td>Detail Edit Hapus</td></tr><tr><td>dinda katrina</td><td>1001001003</td><td>Detail Edit Hapus</td></tr></table></div></div>	Nama	NIM		Nanda	1001001001	Detail Edit Hapus	wahyu	1001001002	Detail Edit Hapus	dinda	1001001003	Detail Edit Hapus	dinda katrina	1001001003	Detail Edit Hapus	Nama	NIM		Nanda	1001001001	Detail Edit Hapus	wahyu	1001001002	Detail Edit Hapus	dinda katrina	1001001003	Detail Edit Hapus
Nama	NIM																											
Nanda	1001001001	Detail Edit Hapus																										
wahyu	1001001002	Detail Edit Hapus																										
dinda	1001001003	Detail Edit Hapus																										
dinda katrina	1001001003	Detail Edit Hapus																										
Nama	NIM																											
Nanda	1001001001	Detail Edit Hapus																										
wahyu	1001001002	Detail Edit Hapus																										
dinda katrina	1001001003	Detail Edit Hapus																										

Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

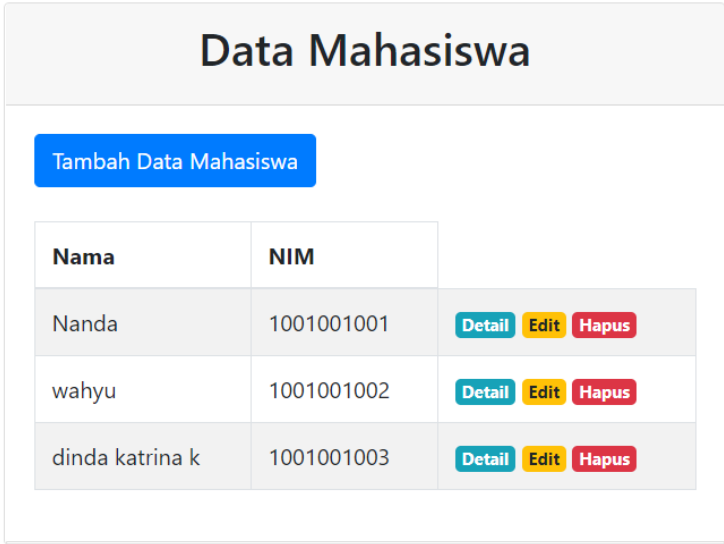
a. Konfigurasi Database

NO	STEP
1.	<p>Buatlah project Laravel baru dengan nama laravel-crud-kedua. Buka command prompt, tuliskan perintah berikut.</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	STEP
1.	<p>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.</p>

	<pre> 15 16 Route::get('/', 'MahasiswaController@index'); 17 Route::get('/mahasiswa', 'MahasiswaController@index'); 18 </pre>
2.	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan</p> <pre> C:\xampp\htdocs\laravel-crud-kedua>php artisan make:model Mahasiswa Model created successfully. </pre>
3.	<p>Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.</p> <pre> 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> <p>Model Mahasiswa akan menangani tabel mahasiswa</p>
4.	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan</p> <pre> C:\xampp\htdocs\laravel-crud-kedua>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 > Http > Controllers > MahasiswaController.php 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class MahasiswaController extends Controller 8 { 9 public function index() 10 { 11 \$mahasiswa = Mahasiswa::all(); 12 return view('index', ['mahasiswa' => \$mahasiswa]); 13 } 14 } </pre> <ul style="list-style-type: none"> - Tambahkan 'use App\Mahasiswa' (line 6) untuk menggunakan model Mahasiswa - Line 12 untuk mengambil semua data dari model/tabel Mahasiswa dan akan disimpan di variabel \$mahasiswa - Line 16 : data akan dikirim ke blade view bernama index

6.	Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat template blade (seperti pada Bagian 1). Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)
7.	<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

NO	STEP
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> 18 19 Route::get('/mahasiswa/tambah','MahasiswaController@tambah');</pre>
2.	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> 14 15 public function tambah(){ 16 return view('tambah'); 17 } 18 19</pre>
3.	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. Lakukan copy pada praktikum 1.</p>

4.	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 20 21 Route::post('/mahasiswa/simpan','MahasiswaController@simpan'); </pre> <p>Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php</p>
5.	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre> 18 19 public function simpan(Request \$request){ 20 Mahasiswa::create([21 'nama' => \$request->namamhs, 22 'nim' => \$request->nimmhs, 23 'email' => \$request->emailmhs, 24 'jurusan' => \$request->jurusanmhs 25]); 26 return redirect('/mahasiswa'); 27 } 28 </pre>
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> 6 7 class Mahasiswa extends Model 8 { 9 protected \$table = "mahasiswa"; 10 protected \$fillable = ['nama','nim','email','jurusan']; 11 } 12 </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.

Tambah Data Mahasiswa

[Kembali](#)

Nama

lince

NIM

1001001005

E-mail

lince@gmail.com

Jurusan

teknik mesin

[Tambah Data](#)

Data Mahasiswa

[Tambah Data Mahasiswa](#)

Nama	NIM	
Nanda	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
dinda katrina k	1001001003	Detail Edit Hapus
lince	1001001005	Detail Edit Hapus

d. Melihat Detail Data dari Database

NO	STEP
1.	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController</p> <pre>22 23 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');</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>28 29 public function detail(\$id){ 30 \$mahasiswa = Mahasiswa::find(\$id); 31 return view('detail', ['mahasiswa' => \$mahasiswa]); 32 } 33</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>resources > views > detail.blade.php 1 @extends('master') 2 <!-- isi title --> 3 @section('title', 'Detail Mahasiswa') 4 5 <!-- isi bagian judul halaman --> 6 @section('judul_halaman', 'Detail Data Mahasiswa') 7 8 <!-- isi bagian konten --> 9 @section('konten') 10 Kembali 11 12
 13
 14 15 <h5 class="card-title"> {{ \$mahasiswa->nama }} </h5> 16 <p class="card-text"> 17 <label for=""> NIM : </label> 18 {{ \$mahasiswa->nim }} </p> 19 <p class="card-text"> 20 <label for=""> E-mail : </label> 21 {{ \$mahasiswa->email }} </p> 22 <p class="card-text"> 23 <label for=""> Jurusan : </label> 24 {{ \$mahasiswa->jurusan }} </p> 25 @endsection</pre>

4.	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p> <div> <h2>Detail Data Mahasiswa</h2> <div>Kembali</div> <p>dinda katrina k</p> <p>NIM : 1001001003</p> <p>E-mail : dinda@gmail.com</p> <p>Jurusan : teknik industri</p> </div>
----	---

e. Mengubah Data (Update) dari Database

NO	STEP
1.	<p>Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController.</p> <pre> 24 25 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> 33 34 public function edit(\$id){ 35 \$mahasiswa = Mahasiswa::find(\$id); 36 return view('edit', ['mahasiswa' => \$mahasiswa]); 37 } 38 }</pre> <p>Line 35 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih</p>

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

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

```
26
27 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update');
```

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->nimhs;
    $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.

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
Nanda	1001001001	Detail Edit Hapus
dinda katrina	1001001003	Detail Edit Hapus
lince	1001001005	Detail Edit Hapus

Edit Data Mahasiswa

Kembali

Nama

dinda katrina din

NIM

1001001003

E-mail

dinda@gmail.com

Jurusan

teknik industri

Simpan Data

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
Nanda	1001001001	Detail Edit Hapus
dinda katrina din	1001001003	Detail Edit Hapus
lince	1001001005	Detail Edit Hapus

f. Menghapus Data (Delete) dari Database

NO	STEP
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@uhapus');</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.	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.</p>

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
Nanda	1001001001	<div>DetailEditHapus</div>
wahyu	1001001002	<div>DetailEditHapus</div>
dinda katrina k	1001001003	<div>DetailEditHapus</div>
lince	1001001005	<div>DetailEditHapus</div>

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
Nanda	1001001001	<div>DetailEditHapus</div>
dinda katrina k	1001001003	<div>DetailEditHapus</div>
lince	1001001005	<div>DetailEditHapus</div>