

LAPORAN PRAKTIKUM PEMROGRAMAN WEBSITE LANJUT

TUGAS PRAKTIKUM

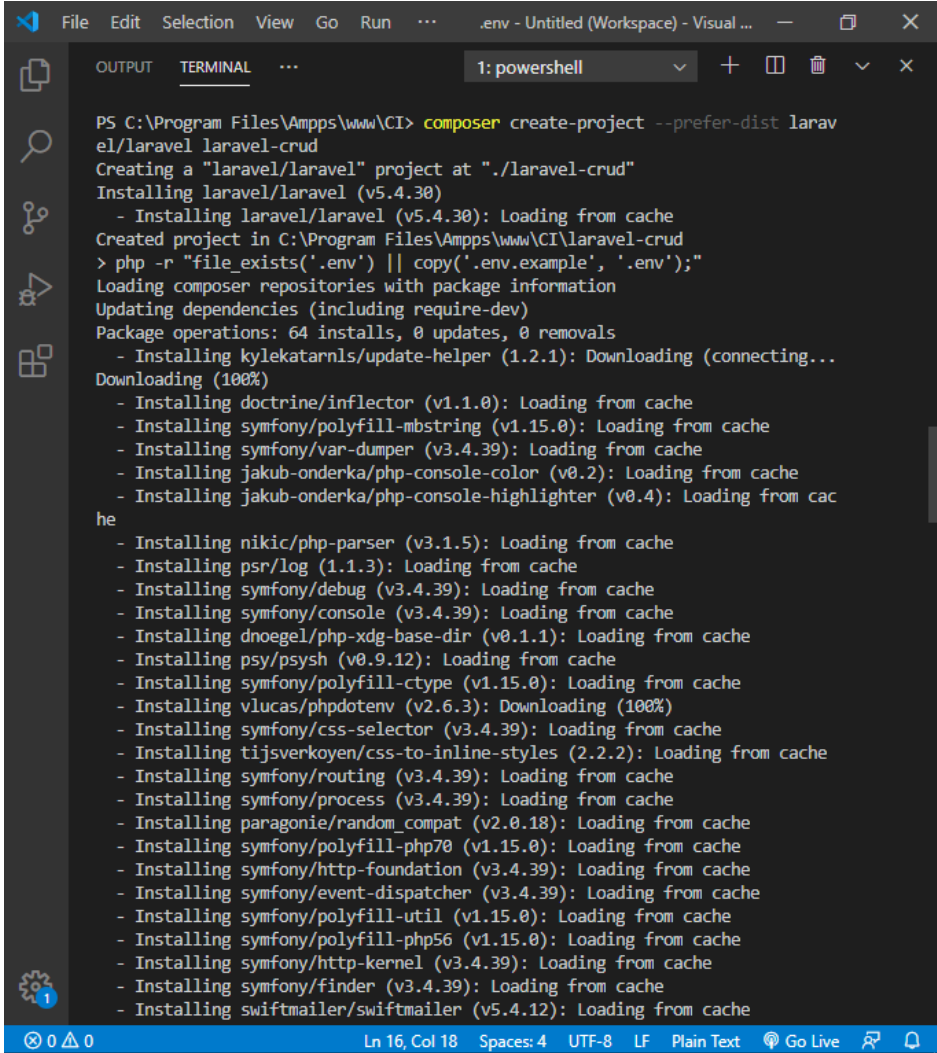
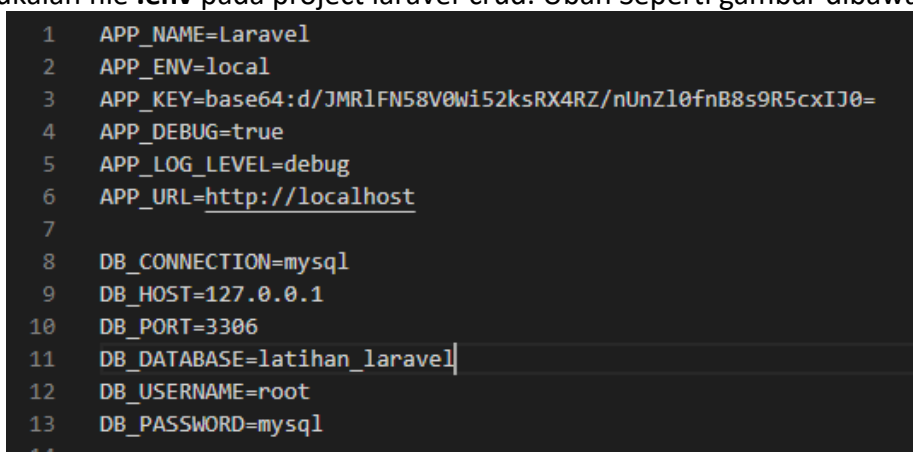


Oleh :

**M. WAHYU PRASETYO / NIM : 1841720177
KELAS TI 2B**

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

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 composer create-project --prefer-dist laravel/laravel 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 gambar dibawah ini.</p>  <pre>1 APP_NAME=Laravel 2 APP_ENV=local 3 APP_KEY=base64:d/JMRlFN58V0Wi52ksRX4RZ/nUnZl0fnB8s9R5cxIJ0= 4 APP_DEBUG=true 5 APP_LOG_LEVEL=debug 6 APP_URL=http://localhost 7 8 DB_CONNECTION=mysql 9 DB_HOST=127.0.0.1 10 DB_PORT=3306 11 DB_DATABASE=latihan_laravel 12 DB_USERNAME=root 13 DB_PASSWORD=mysql</pre> <p>Keterangan: Nama database yang akan digunakan adalah latihan_laravel dengan username root.</p>
3	<p>Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database latihan_laravel.</p>

JelajahiStrukturSQLCariTambahkanEksporImporHak AksesOperasiTrigger

Struktur tabelTampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	UbahHapusLainnya
2	nama	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			UbahHapusLainnya
3	nim	varchar(10)	utf8mb4_general_ci		Tidak	Tidak ada			UbahHapusLainnya
4	email	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			UbahHapusLainnya
5	jurusan	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			UbahHapusLainnya

Pilih SemuaDengan pilihan: JelajahiUbahHapusUtamaUnikIndeksTeks penuh

4Isilah beberapa data pada tabel mahasiswa tersebut.

+ Opsi

idnama_nim_email_jurusan

UbahSalinHapus

1nanda1001001001nanda@gmail.comTeknik Informatika

UbahSalinHapus

2wahyu1001001002wahyu@gmail.comTeknik Elektro

Pilih SemuaDengan pilihan: UbahSalinHapusEkspor

b. Menampilkan Data dari Database

1Setelah 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.

laravel-crud > 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 which
12 | contains the "web" middleware group. Now create something great here.
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.

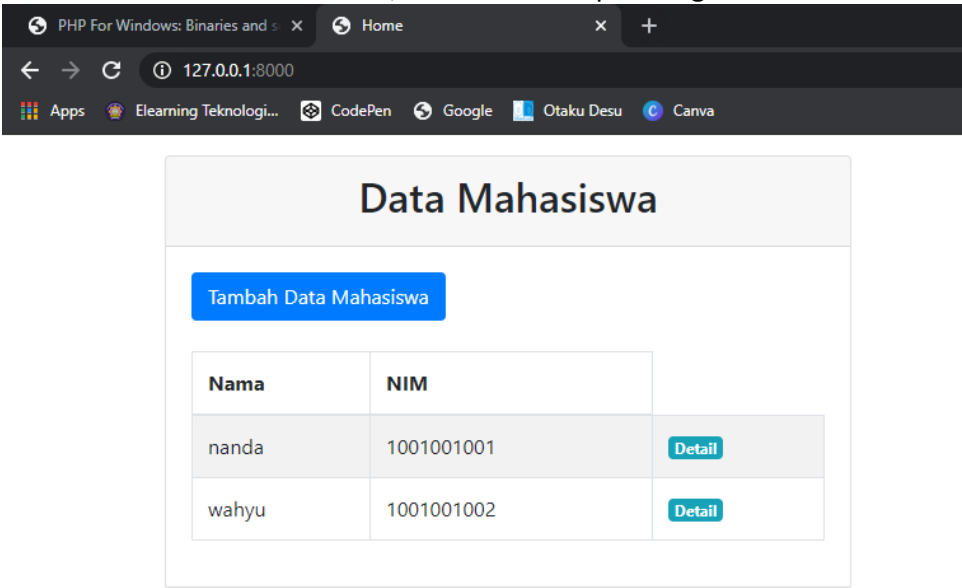
2Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan
cd laravel-crud
php artisan make:controller MahasiswaController

	<div><div>OUTPUTTERMINALDEBUG CONSOLEPROBLEMS1: powershell + □ ✕</div><div>Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved. PS C:\Program Files\Amp\www\CI\laravel-crud> php artisan make:controller MahasiswaController Controller created successfully. PS C:\Program Files\Amp\www\CI\laravel-crud> </div></div>
3	<div>Buat method index pada MahasiswaController.php pada folder app/Http/Controllers</div> <div><div>laravel-crud > app > Http > Controllers > MahasiswaController.php > ...</div><pre>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 ke view index 16 return view('index',['mahasiswa' => \$mahasiswa]); 17 } 18 }</pre></div>
4	<div>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 <i>header-footer</i> pada pembahasan CI) dengan nama master.blade.php pada folder resources/views.</div> <div><div>laravel-crud > resources > views > master.blade.php > html > body > div.container > div.row.mt-3 > div.col-md-6 > div.card ></div><pre>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 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"> 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 <!-- judul halaman --> 16 <h2>@yield('judul_halaman')</h2> 17 </div> 18 <div class="card-body"> 19 <!-- bagian konten --> 20 @yield('konten') 21 </div> 22 </div> 23 </div> 24 </div> 25 </div> 26 </body> 27 </html></pre></div> <div>Keterangan:</div> <div><ul style="list-style-type: none">- 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</div>
5	<div>Sekarang buat file index.blade.php yang menerapkan template master.blade.php dan akan digunakan untuk menampilkan data.</div>

```
laravel-crud > resources > views > index.blade.php > ...
1  @extends('master')
2
3  <!-- title -->
4  @section('title','Home')
5
6  <!-- judul halaman -->
7  @section('judul_halaman','Data Mahasiswa')
8
9  <!-- isi konten -->
10 $section('konten')
11     <a href="/mahasiswa/tambah" class="btn btn-primary">Tambah Data Mahasiswa</a>
12     <br/>
13     <br/>
14     <table class="table table-border table-hover table-striped">
15         <thead>
16             <th>Nama</th>
17             <th>NIM</th>
18         </thead>
19         <tbody>
20             @foreach($mahasiswa as $mhs)
21                 <tr>
22                     <td>{{ $mhs->nama }}</td>
23                     <td>{{ $mhs->nim }}</td>
24                 </tr>
25             @endforeach
26         </tbody>
27     </table>
28 @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. Buatlah *route* baru pada **routes/web.php** dengan nama **/mahasiswa/tambah** yang akan menjalankan fungsi tambah pada MahasiswaController

```
laravel-crud > 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 a group which
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');
21 Route::get('/', 'MahasiswaController@tambah');
```

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

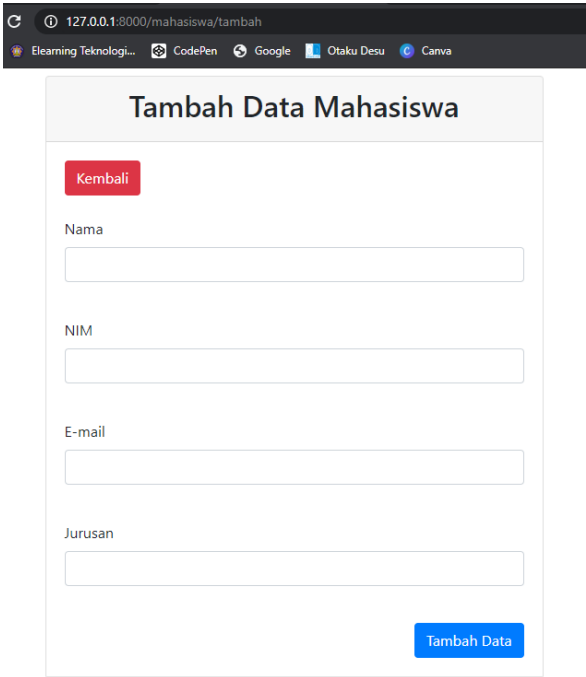
```
public function tambah(){
    //memanggil view tambah
    return view('tambah');
}
```

3. Kemudian buatlah view **tambah.blade.php** yang berisi form untuk memasukkan data baru pada folder **resources/views**. View tambah juga mengaplikasikan master.blade.php.

```
laravel-crud > resources > views > tambah.blade.php > ...
1  @extends('master')
2
3  @section('title','Tambah Data')
4
5  @section('judul_halaman','Tambah Data Mahasiswa')
6
7  @section('konten')
8      <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
9      <br/>
10     <br/>
11     <form action="/mahasiswa/simpan" method="post">
12         {{ csrf_field() }}
13         <div class="form-group">
14             <label for="namamhs">Nama</label>
15             <input type="text" name="namamhs" class="form-control" required="required"> <br/>
16         </div>
17         <div class="form-group">
18             <label for="nimmhs">NIM</label>
19             <input type="number" name="nimmhs" class="form-control" required="required"> <br/>
20         </div>
21         <div class="form-group">
22             <label for="emailmhs">E-mail</label>
23             <input type="email" name="emailmhs" class="form-control" required="required"> <br/>
24         </div>
25         <div class="form-group">
26             <label for="jurusanmhs">Jurusan</label>
27             <input type="text" name="jurusanmhs" class="form-control" required="required"> <br/>
28         </div>
29         <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button>
30     </form>
31 @endsection
```

Keterangan:

- Line 13-32 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan

	<p>- Line 13 terdapat <code>action="/mahasiswa/simpan"</code> yang menunjukkan routes <code>/mahasiswa/simpan</code> dimana data pada form tersebut akan dikirimkan ke fungsi <code>simpan</code> pada controller <code>MahasiswaController</code></p> <p>- Line 14 terdapat <code>{{ csrf_field() }}</code> yang digunakan untuk menerapkan fitur laravel yaitu <code>csrf protection</code>. <code>csrf protection</code> adalah fitur keamanan untuk mencegah penginputan dari luar aplikasi, <code>csrf</code> akan men-generate kode token otomatis yang dibuat dalam bentuk <i>form hidden</i>.</p>
4	<p>Ketika tombol <code>simpan</code> ditekan, akan dipanggil routes <code>/mahasiswa/simpan</code>. 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">- Pada route ini menggunakan metode <code>post</code> karena data mahasiswa dari form akan dikirim ke method <code>simpan</code> di <code>MahasiswaController.php</code>
5	<p>Buat method <code>simpan</code> pada <code>MahasiswaController</code> untuk menyimpan data ke database.</p> <pre>public function simpan(Request \$request) { //insert data ke table mahasiswa DB::table('mahasiswa')->insert(['nama' => \$request->namamhs, 'nim' => \$request->nimmhs, 'email' => \$request->emailmhs, 'jurusan' => \$request->jurusanmhs]); return redirect('/mahasiswa'); }</pre> <p>Keterangan:</p> <ul style="list-style-type: none">- Variabel <code>\$request</code> untuk menerima data yang akan ditambahkan ke database- Line 28-33 merupakan query builder untuk insert data ke tabel mahasiswa
6	<p>Kembali coba jalankan <code>localhost:8000</code> dan pilih tombol <code>'Tambah Data Mahasiswa'</code>, maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.</p> <div></div> <p>Setelah kita klik tombol tambah data, maka data baru akan ditampilkan pada halaman index.</p>

127.0.0.1:8000/mahasiswa

AppsElearning Teknologi...CodePenGoogleOtaku DesuCanva

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda	1001001001	Detail
wahyu	1001001002	Detail
M. Wahyu Prasetyo	1841720177	Detail

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**

```
laravel-crud > 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                 <td>
24                     <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a>
25                 </td>
26             </tr>
27         @endforeach
28     </tbody>
29 </table>
30 @endsection
```

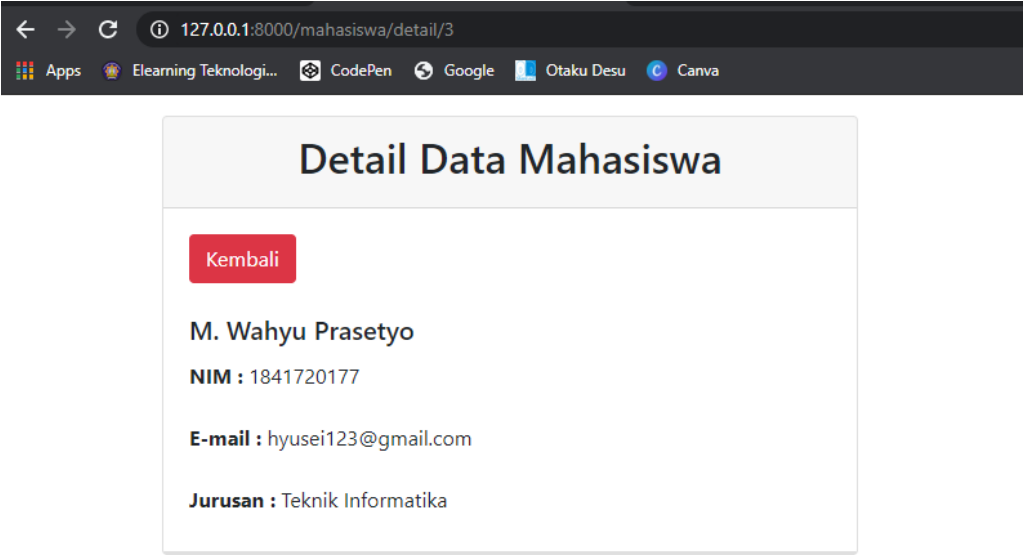
2

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');
```

3

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

	<pre>public function detail(\$id) { \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); 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>laravel-crud > 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 @foreach(\$mahasiswa as \$mhs) 13 <h5 class="card-title"> {{ \$mhs->nama }} </h5> 14 <p class="card-text"> 15 <label for=""> NIM : </label> 16 {{ \$mhs->nim }} </p> 17 <p class="card-text"> 18 <label for=""> E-mail : </label> 19 {{ \$mhs->email }} </p> 20 <p class="card-text"> 21 <label for=""> Jurusan : </label> 22 {{ \$mhs->jurusan }} </p> 23 @endforeach 24 @endsection</pre> <p>Keterangan: - Line 16-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan</p>	
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>laravel-crud > resources > views > index.blade.php > table.table-bordered.table-hover.table-striped > tbody > tr > td 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 </tr> 17 </thead> 18 <tbody> 19 @foreach(\$mahasiswa as \$mhs) 20 <tr> 21 <td>{{ \$mhs->nama }}</td> 22 <td>{{ \$mhs->nim }}</td> 23 <td> 24 id }}" class="badge badge-info">Detail 25 id }}" class="badge badge-warning">Edit 26 </td> 27 </tr> 28 @endforeach 29 </tbody> 30 </table> 31 @endsection</pre>
2	<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>
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> <p>Keterangan :</p> <ul style="list-style-type: none">- Line 49 : query builder untuk mengambil data mahasiswa berdasarkan id yang dipilih
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>laravel-crud > resources > views > edit.blade.php > ... 5 @section('judul_halaman','Edit Data Mahasiswa') 6 7 @section('konten') 8 Kembali 9
 10
 11 @foreach(\$mahasiswa as \$mhs) 12 <form action="/mahasiswa/update" method="post"> 13 {{ csrf_field() }} 14 <input type="hidden" name="id" value="{{ \$mhs->id }}">
 15 <div class="form-grub"> 16 <label for="namamhs">Nama</label> 17 <input type="text" class="form-control" required name="namamhs" value="{{ \$mhs->nama }}">
 18 </div> 19 <div class="form-grub"> 20 <label for="nimhs">NIM</label> 21 <input type="text" class="form-control" required name="nimhs" value="{{ \$mhs->nim }}">
 22 </div> 23 <div class="form-grub"> 24 <label for="emailhs">E-mail</label> 25 <input type="text" class="form-control" required name="emailhs" value="{{ \$mhs->email }}">
 26 </div> 27 <div class="form-grub"> 28 <label for="jurusanhs">Jurusan</label> 29 <input type="text" class="form-control" required name="jurusanhs" value="{{ \$mhs->jurusan }}">
 30 </div> 31 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button> 32 </form> 33 @endforeach 34 @endsection</pre>

5

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

```
28 Route::get('/mahasiswa/update','MahasiswaController@update');
```

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.

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

Keterangan:

- 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.

127.0.0.1:8000/mahasiswa

Elearning Teknologi...

CodePen

Google

Otaku Desu

Canva

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda	1001001001	<div>DetailEdit</div>
wahyu	1001001002	<div>DetailEdit</div>
M. Wahyu Prasetyo	1841720177	<div>DetailEdit</div>