



Nama : Alfian Noufal Nasruddin

Kelas : TI-2A

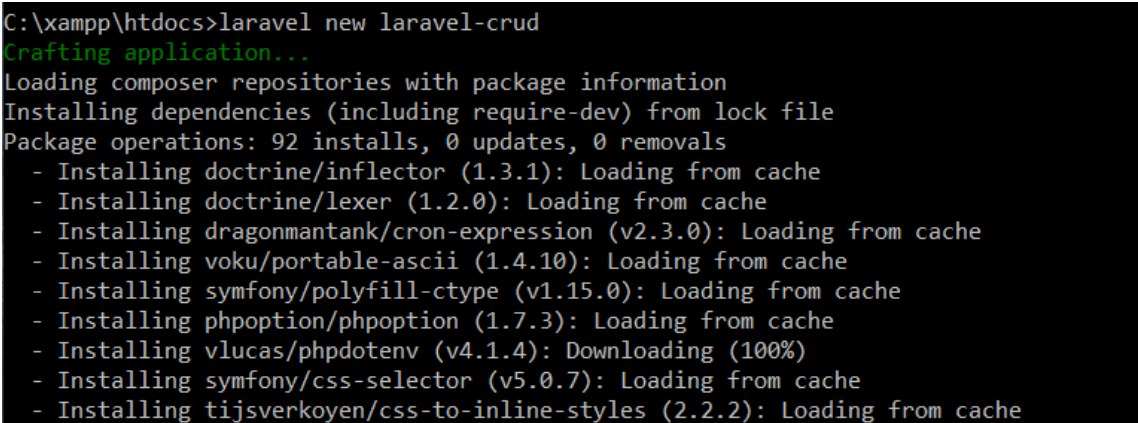
NIM : 1841720059

No.absen : 07

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

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

a. Konfigurasi Database dan Pembuatan Tabel di MySQL

Langkah	Keterangan
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>  <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 - Installing doctrine/inflector (1.3.1): Loading from cache - Installing doctrine/lexer (1.2.0): Loading from cache - Installing dragonmantank/cron-expression (v2.3.0): Loading from cache - Installing voku/portable-ascii (1.4.10): Loading from cache - Installing symfony/polyfill-ctype (v1.15.0): Loading from cache - Installing phpoption/phpooption (1.7.3): Loading from cache - Installing vlucas/phpdotenv (v4.1.4): Downloading (100%) - Installing symfony/css-selector (v5.0.7): Loading from cache - Installing tijsverkoyen/css-to-inline-styles (2.2.2): Loading from cache</pre>

2

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



The screenshot shows a code editor with a sidebar on the left and a main editor area on the right. The sidebar has a section 'EXPLORER' with a sub-section 'LARAVEL-CRUD' containing a list of folders: app, bootstrap, config, database, public, resources, routes, storage, tests, and vendor. Below this is a file named '.env'. The main editor area shows the contents of the '.env' file, which is a configuration file for a Laravel application. The file contains the following lines:

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:aJdRpTYHHBEPVavj5S8Cw8zF11FhxNtyJww3FYjjAE=
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.

```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:yIeQ00ZYUclZ7Sby0Iys/j87n7ba4WzwZrru5Pm1/T0=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=latihan_laravel
13 DB_USERNAME=root
14 DB_PASSWORD=
15
```

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

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 nama	varchar(100)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	3 nim	varchar(10)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	4 email	varchar(50)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	5 jurusan	varchar(20)	utf8mb4_general_ci		No	None		

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id	int(11)		Tidak	Tidak ada			AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2 nama	varchar(100)	utf8_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	3 nim	varchar(10)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	4 email	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	5 jurusan	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya

- 4 Isilah beberapa data pada tabel **mahasiswa** tersebut.

+ Options

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

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Urut berdasarkan kunci: Tidak ada

+ Opsi

	id	nama	nim	email	jurusan
<input type="checkbox"/>	1	alfan	1841720059	alfannoufal@gmail.com	Teknik Informatika
<input type="checkbox"/>	2	Bianca	1841623011	bianca@gmail.com	Teknik Kimia

b. Menampilkan Data dari Database

Langkah	Keterangan
---------	------------

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.

```

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

```

```

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 wh
12 | contains the "web" middleware group. Now create something great
13 |
14 */
15
16 Route::get('/', 'MahasiswaController@index');
17

```

Keterangan:

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

2

Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan

cd laravel-crud

php artisan make:controller MahasiswaController

```

C:\xampp\htdocs\laravel-crud>php artisan make:controller MahasiswaController
Controller created successfully.

```

3

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

```

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 }

```

Keterangan:

- Tambahkan 'use Illuminate\Support\Facades\DB;' (line 6) agar *query builder* dapat digunakan
- Line 13 untuk mengambil data dari tabel mahasiswa menggunakan *query builder* laravel dan akan disimpan di variabel \$mahasiswa
- Line 16 : data akan dikirim ke blade view bernama index

```

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         //send data mahasiswa ke view index
16         return view('index', ['mahasiswa' => $mahasiswa]);
17     }
18 }
19

```

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**.

```

EXPLORER
> OPEN EDITORS
  LARA...
  > app
  > bootstrap
  > config
  > database
  > public
  > resources
    > js
    > lang
    > sass
    > views
      detail.blade.php
      edit.blade.php
      index.blade.php
      master.blade.php
      tambah.blade.php
      welcome.blade.php
  > routes
  > storage
  > tests
  > vendor
  .editorconfig
  .env

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/bootstrap.min.css" rel="stylesheet">
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>

```

Keterangan:

- Fungsi `@yield` pada line 6(title), 15(judul_halaman), dan 19(konten) berfungsi sebagai penanda bagian-bagian pada master blade. Nantinya bagian `@yield` ini akan diisi sesuai dengan halaman view yang menerapkan master.blade.php

```

resources > views > master.blade.php
1  <html>
2  <head>
3      <meta charset="utf-8">
4      <meta name="viewport" content="width=device-width, initial-s
5      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
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 class="card">
22                     </div>
23             </div>
24         </div>
25     </body>
26 </html>

```


5

Sekarang buat file **index.blade.php** yang menerapkan *template* master.blade.php dan akan digunakan untuk menampilkan data.

```

EXPLORER
  > OPEN EDITORS
  > LARAVEL-CRUD
    > app
    > bootstrap
    > config
    > database
    > public
    > resources
      > js
      > lang
      > sass
      > views
        detail.blade.php
        edit.blade.php
        index.blade.php
        master.blade.php
        tambah.blade.php
        welcome.blade.php
    > routes
    > storage
    > tests
    > vendor
    .editorconfig
    .env
    .env.example
    .gitattributes
    .gitignore

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     <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
  
```

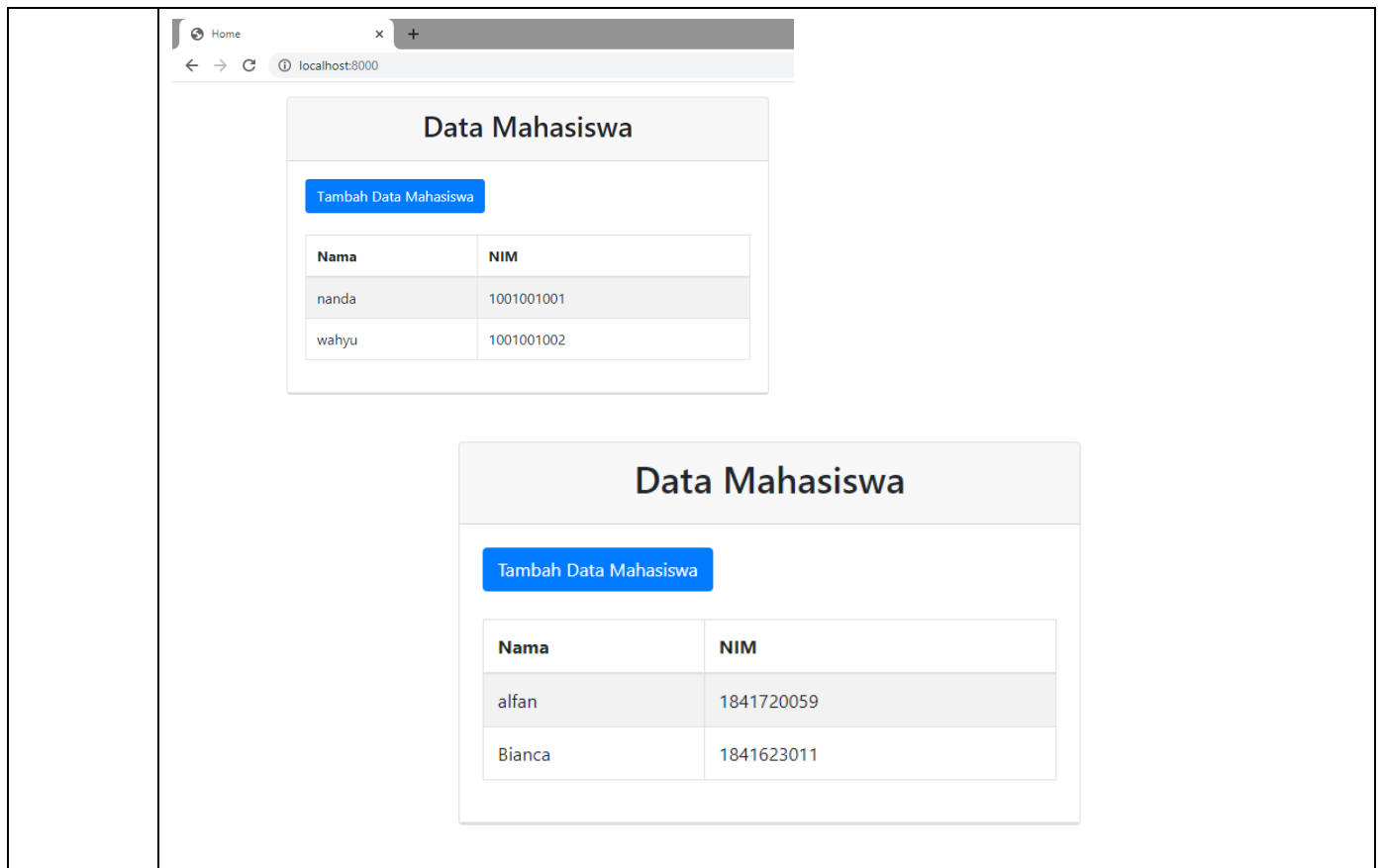
```

.env
web.php
MahasiswaController.php
master.blade.php
index.blade

resources > views > index.blade.php
1  @extends('master')
2
3  <!-- Isi Judul -->
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:

	<ul style="list-style-type: none"> - Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan bladelain yaitu master.blade.php - Line 4 : @section('title', 'Home') berarti bahwa file index mengisi @yield('title') pada master dengan 'Home' - Line 7 : @section('judul_halaman', 'Data Mahasiswa) berarti bahwa file index mengisi @yield('judul_halaman) pada master dengan 'Home' - Line 10-30 : mengisi yield(@konten), karena terdapat banyak baris pada diawali dengan @section('konten') dan diakhiri dengan @endsection - Line 24-25 menampilkan data mahasiswa dengan kolom nama dan nim
6	<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

Langkah	Keterangan
1	<p>Buatlah <i>route</i> 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'); 17 18 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 19 </pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> 19 public function tambah() 20 { 21 // memanggil view tambah 22 return view('tambah'); 23 } 24 </pre> <pre> public function tambah() { //call view add 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>

EXPLORER

index.blade.php

web.php

MahasiswaController.php

tambah.blade.php

OPEN EDIT...

2 UNSAVED

resources > views > tambah.blade.php

LARA...

app

Console

Exceptions

Http

Controllers

Controller.php

MahasiswaContro...

Middleware

Kernel.php

Providers

User.php

bootstrap

config

database

public

resources

js

lang

sass

views

detail.blade.php

edit.blade.php

index.blade.php

master.blade.php

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="nimmhs">NIM</label>

21 <input type="number" class="form-control" required="required" name="nimmhs">

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">Tambah Data</button>

32 </form>

33 @endsection

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 halaman -->

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="nimmhs">NIM</label>

21 <input type="number" class="form-control" required="required" name="nimmhs">

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">Tambah Data</button>

32 </form>

33 @endsection

Keterangan:

- Line 13-32 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 13 terdapat action="/mahasiswa/simpan" yang menunjukkan routes /mahasiswa/simpan dimana data pada form tersebut akan dikirimkan ke fungsi simpan

	<p>pada controller MahasiswaController</p> <ul style="list-style-type: none"> - Line 14 terdapat {{ csrf_field() }} yang digunakan untuk menerapkan fitur laravel yaitu csrf protection. csrf protection adalah fitur keamanan untuk mencegah penginputan dari luar aplikasi, csrf akan men-generate kode token otomatis yang dibuat dalam bentuk <i>form hidden</i>.
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');</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php <pre>19 20 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');</pre>
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p>

```

24
25 public function simpan(Request $request)
26 {
27     // insert data ke table mahasiswa
28     DB::table('mahasiswa')->insert([
29         'nama' => $request->namamhs,
30         'nim' => $request->nimmhs,
31         'email' => $request->emailmhs,
32         'jurusan' => $request->jurusanmhs
33     ]);
34     return redirect('/mahasiswa');
35 }
36

```

Keterangan:

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

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

Tambah Data Mahasiswa

[Kembali](#)

Nama
marsha

NIM
1001001003

E-mail
marsha@gmail.com

Jurusan
teknik sipil

[Tambah Data](#)

Tambah Data Mahasiswa

Kembali

Nama

Alfan Gantenk

NIM

1841720059

E-mail

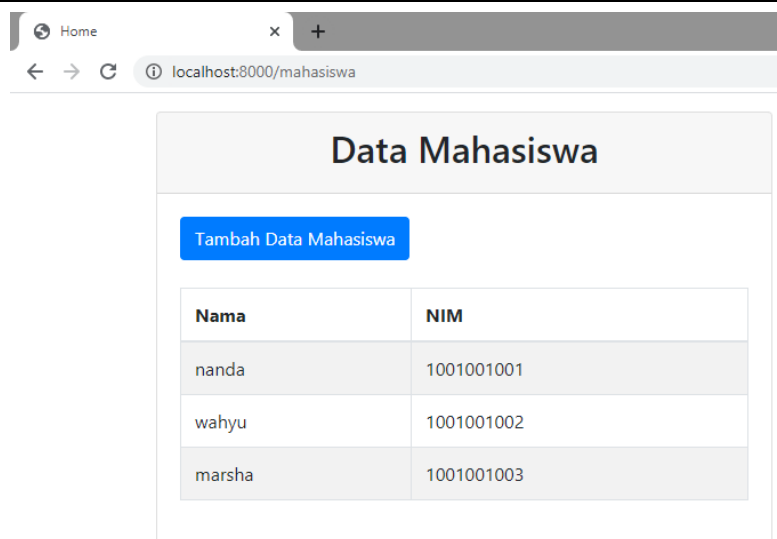
alfanganteng@gmail.com

Jurusan

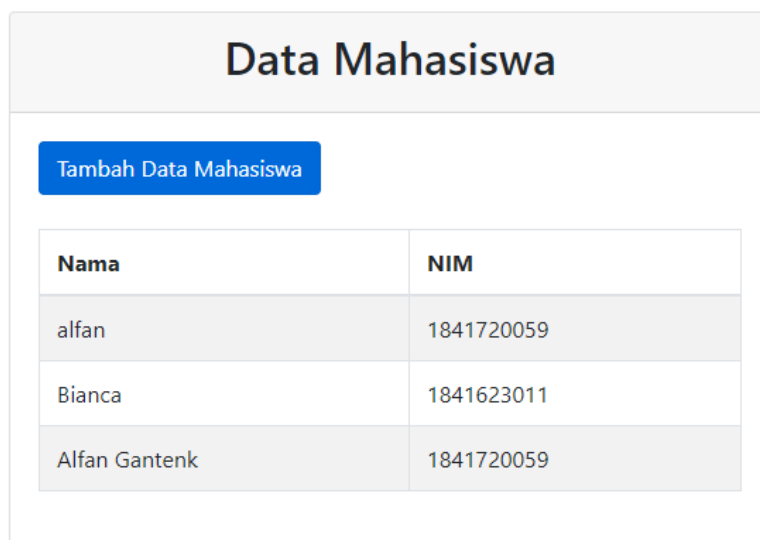
Teknik Informatika

Tambah Data

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




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



d. Melihat Detail Data dari Database

Langkah	Keterangan
---------	------------

1	<p>Buatlah tombol untuk melihat detail data (Line 28) pada file index.blade.php di folder resources/views</p>  <pre> 14 <table class="table table-bordered table-hover table-striped"> 15 <thead> 16 <tr> 17 <th>Nama</th> 18 <th>NIM</th> 19 <th></th> 20 </tr> 21 </thead> 22 <tbody> 23 @foreach(\$mahasiswa as \$mhs) 24 <tr> 25 <td>{{ \$mhs->nama }}</td> 26 <td>{{ \$mhs->nim }}</td> 27 <td> 28 id }}" class="badge badge-info">Detail 29 </td> 30 </tr> 31 @endforeach 32 </tbody> 33 </table> 34 @endsection </pre>
2	<p>Buatlah <i>route</i> 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'); 21 22 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>

EXPLORER index.blade.php web.php MahasiswaController.php tambah.blade.p

> OPEN EDIT... 2 UNSAVED app > Http > Controllers > MahasiswaController.php > MahasiswaController > tambah

▼ LARAVEL-CRUD

▼ app

> Console

> Exceptions

▼ Http

▼ Controllers

Controller.php

MahasiswaContro...

```

36
37 public function detail($id)
38 {
39     // mengambil data mahasiswa berdasarkan id yang dipilih
40     $mahasiswa = DB::table('mahasiswa')->where('id',$id)->get();
41     // kirim data mahasiswa yang diambil ke view edit.blade.php
42     return view('detail',['mahasiswa' => $mahasiswa]);
43 }
44
45

```

```

public function detail($id)
{
    //put data mahasiswa berdasar id yg dipilih
    $mahasiswa = DB::table('mahasiswa')->where('id',$id)->get();
    //send data mahasiswa yg diambil ke view edit.blade.php
    return view('detail',['mahasiswa' => $mahasiswa]);
}

```

4

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

EXPLORER index.blade.php web.php MahasiswaController.php detail.blade.php

> OPEN EDIT... 1 UNSAVED resources > views > detail.blade.php

▼ LARA... Kernel.php

> Providers

User.php

> bootstrap

> config

> database

> public

▼ resources

> js

> lang

> sass

▼ views

detail.blade.php

edit.blade.php

index.blade.php

master.blade.php

tambah.blade.php

welcome.blade.php

▼ routes

api.php

channels.php

console.php

web.php

```

1 @extends('master')
2
3 <!-- isi title -->
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
13     <br/>
14     <br/>
15
16     @foreach($mahasiswa as $mhs)
17         <h5 class="card-title"> {{ $mhs->nama }} </h5>
18         <p class="card-text">
19             <label for=""><b> NIM : </b></label>
20             {{ $mhs->nim }} </p>
21         <p class="card-text">
22             <label for=""><b> E-mail : </b></label>
23             {{ $mhs->email }} </p>
24         <p class="card-text">
25             <label for=""><b> Jurusan : </b></label>
26             {{ $mhs->jurusan }} </p>
27     @endforeach
28 @endsection

```

Keterangan:

- Line 16-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan

	<pre> resources / views / detail.blade.php 1 @extends('master') 2 3 <!-- isi title --> 4 @section('title','Detail Mahasiswa') 5 6 <!-- isi bagian judul halaman --> 7 @section('judul_halaman','Detail Data Mahasiswa') 8 9 <!-- isi bagian konten halaman --> 10 @section('konten') 11 Kembali 12 13
 14
 15 16 @foreach(\$mahasiswa as \$mhs) 17 <h5 class="card-title">{{\$mhs->nama}}</h5> 18 <p class="card-text"> 19 <label for=""> NIM : </label> 20 {{\$mhs->nim}} </p> 21 <p class="card-text"> 22 <label for=""> E-mail : </label> 23 {{\$mhs->email}} </p> 24 <p class="card-text"> 25 <label for=""> Jurusan : </label> 26 {{\$mhs->jurusan}} </p> 27 @endforeach 28 @endsection </pre>
5	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p>

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda	1001001001	Detail
wahyu	1001001002	Detail
marsha	1001001003	Detail
yasin	1001001004	Detail
mira	1001001005	Detail

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
alfan	1841720059	Detail
Bianca	1841623011	Detail
Alfan Gantenk	1841720059	Detail

Detail Data Mahasiswa

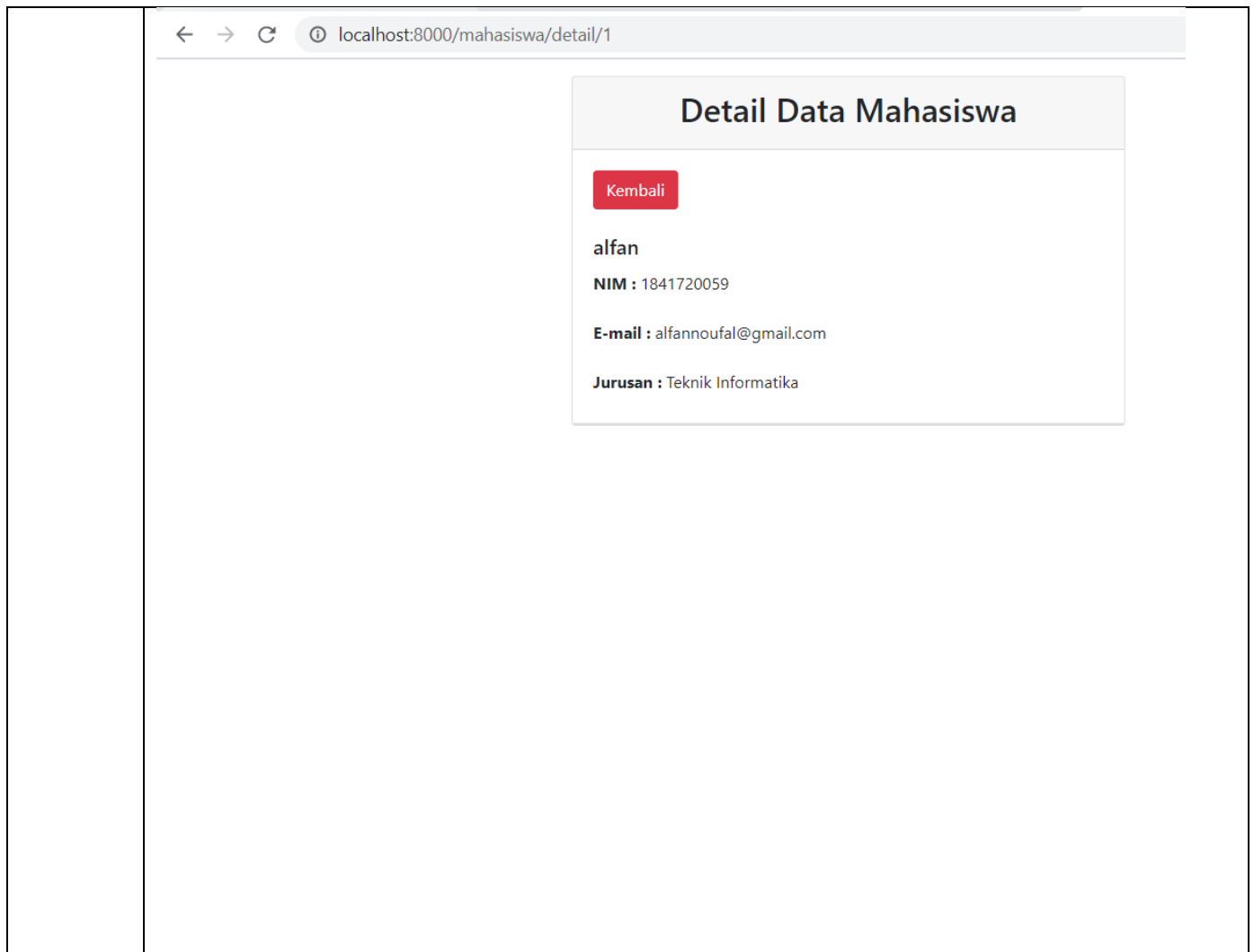
Kembali

mira

NIM : 1001001005

E-mail : mira@gmail.com

Jurusan : teknik informatika



e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views</p> <div><div><div>EXPLORER</div><div><div>OPEN EDIT... 1 UNSAVED</div><div>LARAVEL-CRUD</div><div><div>User.php</div><div>bootstrap</div><div>config</div><div>database</div><div>public</div><div>resources</div><div><div>js</div><div>lang</div><div>sass</div><div>views</div></div><div><div>detail.blade.php</div><div>edit.blade.php</div><div>index.blade.php</div><div>master.blade.php</div><div>tambah.blade.php</div><div>welcome.blade.php</div></div><div>routes</div></div></div><div><div>index.blade.php x web.php MahasiswaController.php detail.blade.php</div><div>resources > views > index.blade.php</div><div><pre>15 <thead> 16 <tr> 17 <th>Nama</th> 18 <th>NIM</th> 19 <th> </th> 20 </tr> 21 </thead> 22 <tbody> 23 @foreach(\$mahasiswa as \$mhs) 24 <tr> 25 <td>{{ \$mhs->nama }}</td> 26 <td>{{ \$mhs->nim }}</td> 27 <td> 28 id }}" class="badge badge-info">Detail 29 id }}" class="badge badge-warning">Edit 30 </td> 31 </tr> 32 @endforeach 33 </tbody> 34 </table> 35 @endsection</pre></div></div></div></div>

	<pre> <table class="table table-bordered table-hover table-striped"> <thead> <tr> <th>Nama</th> <th>NIM</th> <th> </th> </tr> </thead> <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 </td> </tr> @endforeach </tbody> </table> </pre>
2	<p>Buatlah <i>route</i> 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'); 23 24 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> 46 public function edit(\$id) 47 { 48 // mengambil data mahasiswa berdasarkan id yang dipilih 49 \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); 50 // kirim data mahasiswa yang diambil ke view edit.blade.php 51 return view('edit',['mahasiswa' => \$mahasiswa]); 52 } 53</pre> <pre> public function edit(\$id) { //put data mahasiswa berdasar id yg dipilih \$mahasiswa = DB::table('mahasiswa')->where('id',\$id)->get(); //send data mahasiswa yg diambil ke view edit.blade.php 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

Kemudian buatlah view **edit.blade.php** yang berisi form untuk mengubah data pada folder **resources/views**. View edit juga mengaplikasikan master.blade.php.

```

1  @extends('master')
2
3  <!-- isi title -->
4  @section('title', 'Edit Data')
5
6  <!-- isi bagian judul halaman -->
7  @section('judul_halaman', 'Edit Data Mahasiswa')
8
9  <!-- isi bagian konten -->
10 @section('konten')
11     <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
12     <br/>
13     @foreach($mahasiswa as $mhs)
14         <form action="/mahasiswa/update" 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="nimmhs">NIM</label>
23                 <input type="number" class="form-control" required="required" name="nimmhs" 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

```

Keterangan:

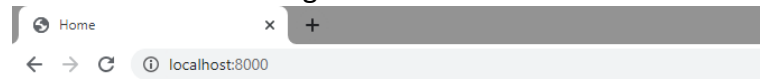
- Line 14-36 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 15 terdapat action="/mahasiswa/update" yang menunjukkan routes /mahasiswa/update dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController

	<pre> resources > views > edit.blade.php 1 @extends('master') 2 3 <!-- isi title --> 4 @section('title','Detail Mahasiswa') 5 6 <!-- isi bagian judul halaman --> 7 @section('judul_halaman','Edit Data Mahasiswa') 8 9 <!-- isi bagian konten halaman --> 10 @section('konten') 11 Kembali 12 13
 14
 15 @foreach(\$mahasiswa as \$mhs) 16 <form action="/mahasiswa/update" method="post"> 17 {{ csrf_field() }} 18 <input type="hidden" name="id" value="{{ \$mhs->id }}">
 19 <div class="form-group"> 20 <label for="namamhs">Nama</label> 21 <input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mhs->nama }}">
 22 </div> 23 <div class="form-group"> 24 <label for="nimhs">NIM</label> 25 <input type="number" class="form-control" required="required" name="nimhs" value="{{ \$mhs->nim }}">
 26 </div> 27 <div class="form-group"> 28 <label for="emailmhs">E-mail</label> 29 <input type="email" class="form-control" required="required" name="emailmhs" value="{{ \$mhs->email }}">
 30 </div> 31 <div class="form-group"> 32 <label for="jurusanmhs">Jurusan</label> 33 <input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mhs->jurusan }}">
 34 </div> 35 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button> 36 </form> 37 @endforeach 38 @endsection </pre>
5	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 29 30 Route::post('/mahasiswa/update','MahasiswaController@update'); </pre> <p>Keterangan:</p>

	<ul style="list-style-type: none"> - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php <pre> 25 26 Route::post('/mahasiswa/update', 'MahasiswaController@update');</pre>
6	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p> <pre> 55 public function update(Request \$request) 56 { 57 // update data mahasiswa 58 DB::table('mahasiswa')->where('id',\$request->id)->update([59 'nama' => \$request->namamhs, 60 'nim' => \$request->nimmhs, 61 'email' => \$request->emailmhs, 62 'jurusan' => \$request->jurusanmhs 63]); 64 return redirect('/mahasiswa'); 65 }</pre> <pre> public function update(Request \$request) { // insert data ke table mahasiswa DB::table('mahasiswa')->where('id',\$request->id)->update(['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 \$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.



Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
nanda	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
marsha	1001001003	Detail Edit
yasin	1001001004	Detail Edit
mira	1001001005	Detail Edit

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
alfan	1841720059	Detail Edit
Bianca	1841623011	Detail Edit
Alfan Gantenk	1841720059	Detail Edit

Klik edit di data pertama, kita akan mengubah namanya.

Edit Data

localhost:8000/mahasiswa/edit/1

Edit Data Mahasiswa

[Kembali](#)

Nama
nanda eka

NIM
1001001001

E-mail
nandan@gmail.com

Jurusan
teknik informatika

[Simpan Data](#)

localhost:8000/mahasiswa/edit/3

Edit Data Mahasiswa

[Kembali](#)

Nama
Alfan Noufal Nasruddin

NIM
184172006d

E-mail
alfanganteng@gmail.com

Jurusan
Teknik Informatika

[Tambah Data](#)

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

Home

localhost:8000/mahasiswa

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
marsha	1001001003	Detail Edit
yasin	1001001004	Detail Edit
mira	1001001005	Detail Edit

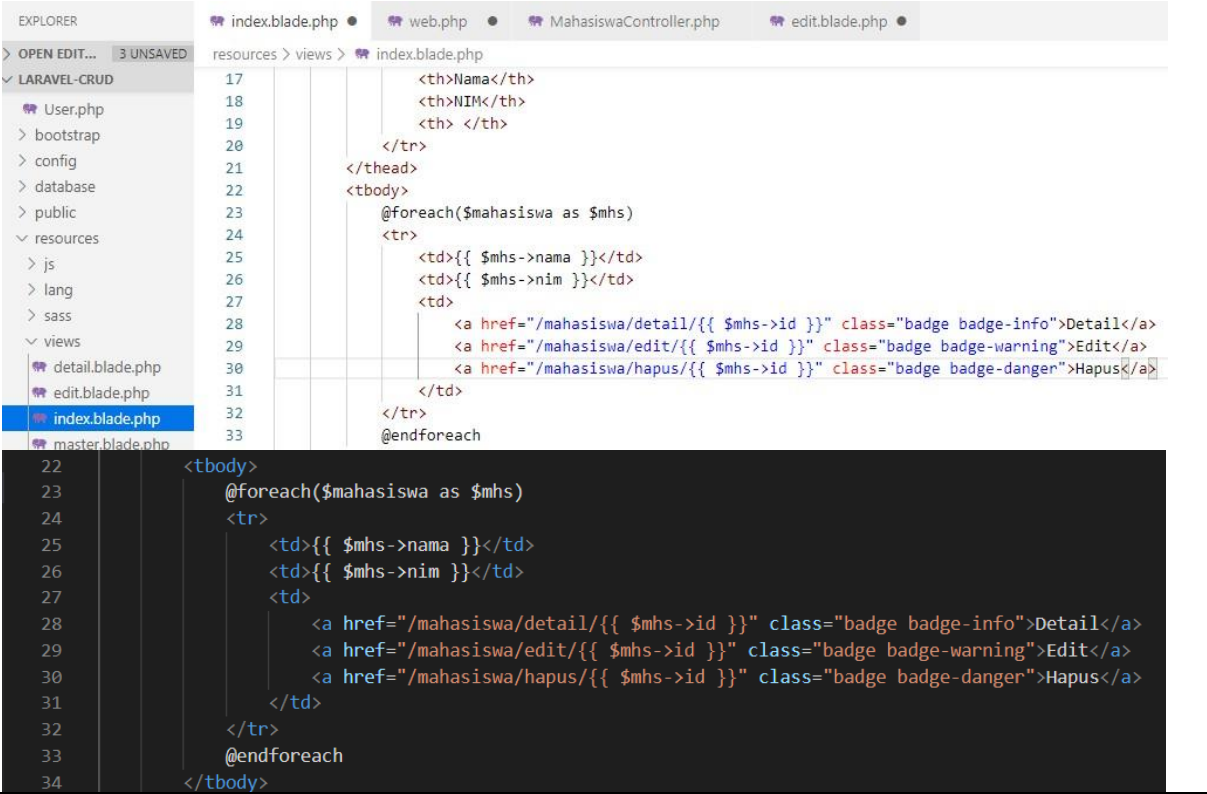

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
alfan	1841720059	Detail Edit
Bianca	1841623011	Detail Edit
Alfan Noufal Nasruddin	1841720060	Detail Edit

f. Menghapus Data (Delete) dari Database

Langkah	Keterangan
1	Buatlah tombol untuk menghapus data (Line 30) pada file index.blade.php di folder resources/views

	 <pre> 17 <th>Nama</th> 18 <th>NIM</th> 19 <th></th> 20 </tr> 21 </thead> 22 <tbody> 23 @foreach(\$mahasiswa as \$mhs) 24 <tr> 25 <td>{{ \$mhs->nama }}</td> 26 <td>{{ \$mhs->nim }}</td> 27 <td> 28 id }}" class="badge badge-info">Detail 29 id }}" class="badge badge-warning">Edit 30 id }}" class="badge badge-danger">Hapus 31 </td> 32 </tr> 33 @endforeach 34 </tbody> </pre>
2	<p>Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> 31 32 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 27 28 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); </pre>
3	<p>Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.</p>  <pre> 65 } 66 67 public function hapus(\$id) 68 { 69 // menghapus data mahasiswa berdasarkan id yang dipilih 70 DB::table('mahasiswa')->where('id',\$id)->delete(); 71 72 return redirect('/mahasiswa'); 73 } 74 75 </pre> <pre> public function hapus(\$id) { //menghapus data mhs berdasarkan id yg diplh DB::table('mahasiswa')->where('id',\$id)->delete(); return redirect('/mahasiswa'); } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - 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	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
yasin	1001001004	Detail Edit Hapus
mira	1001001005	Detail Edit Hapus

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
yasin	1001001004	Detail Edit Hapus

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
alfan	1841720059	Detail Edit Hapus
Bianca	1841623011	Detail Edit Hapus
Alfan Noufal Nasruddin	1841720060	Detail Edit Hapus

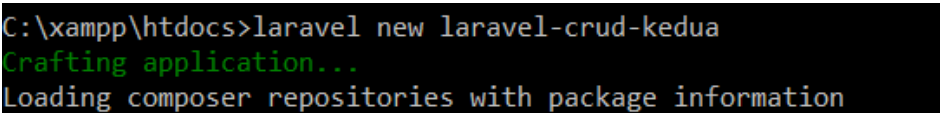
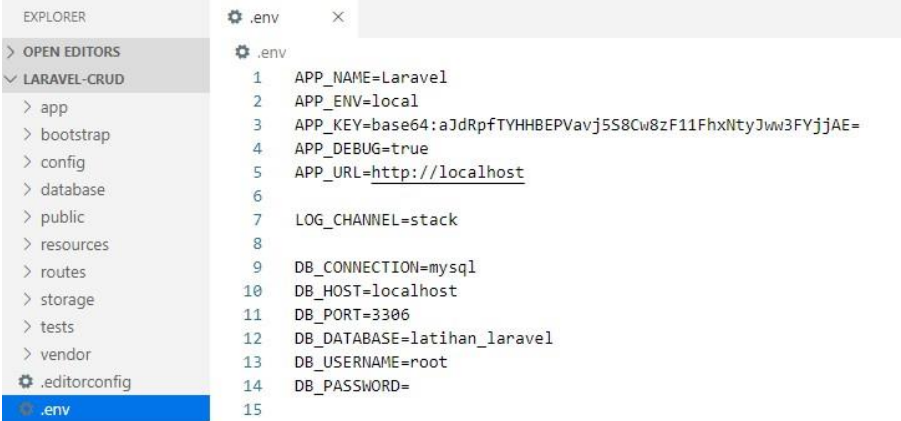
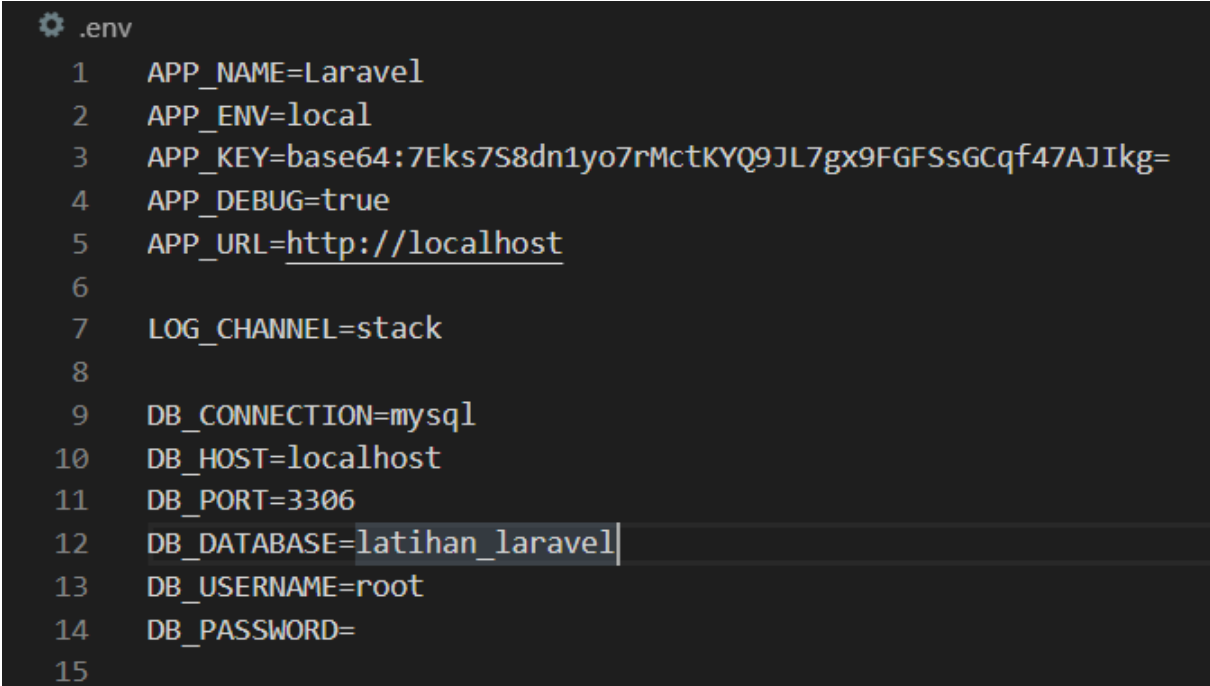
Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
alfan	1841720059	Detail Edit Hapus
Bianca	1841623011	Detail Edit Hapus

Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

a. Konfigurasi Database

Langkah	Keterangan
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>  <p>Keterangan:</p> <ul style="list-style-type: none">- Nama database yang akan digunakan adalah latihan_laravel dengan username root. 

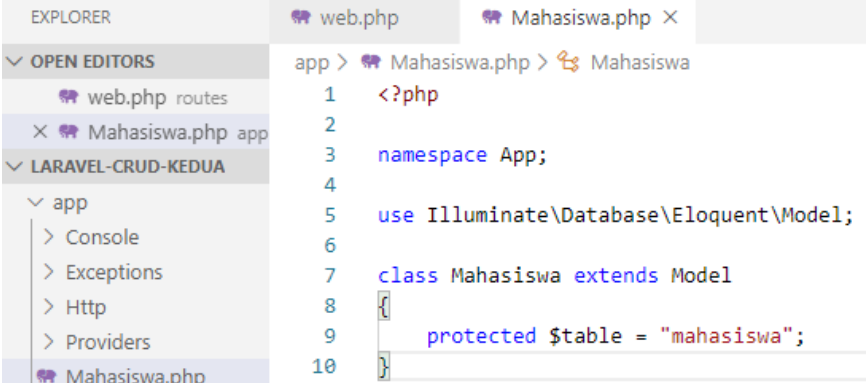
3

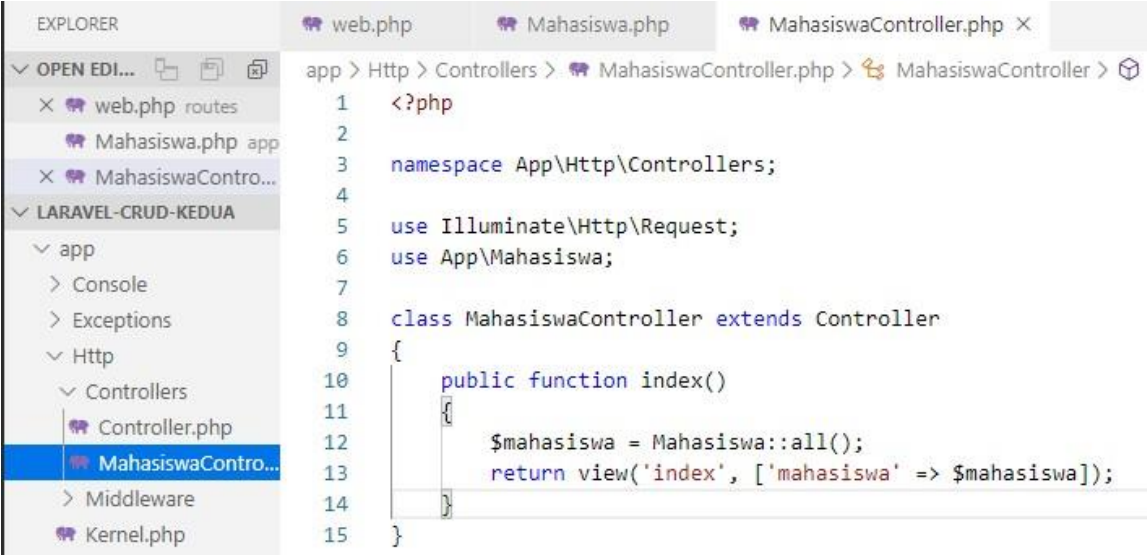
Pada project ini kita gunakan database dan tabel yang sebelumnya digunakan pada Praktikum Bagian 1. Tambahkan kolom `created_at` dan `updated_at` yang bertipe `TIMESTAMP` dan default nilainya `NULL`

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	nama	varchar(100)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	3	nim	varchar(10)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	4	email	varchar(50)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	5	jurusan	varchar(20)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	6	created_at	timestamp		Yes	NULL		
<input type="checkbox"/>	7	updated_at	timestamp		Yes	NULL		

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	id	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2	nama	varchar(100)	utf8_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3	nim	varchar(10)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4	email	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	5	jurusan	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6	created_at	timestamp		Ya	NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	7	updated_at	timestamp		Ya	NULL			Ubah Hapus Lainnya

b. Menampilkan Data dari Database

Langkah	Keterangan
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan.</p> <p>Pertama, buatlah <i>route</i> 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'); 20 Route::get('/', 'MahasiswaController@index'); 21 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 php artisan make:model Mahasiswa</p> <pre> C:\xampp\htdocs>cd laravel-crud-kedua 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 namespace App; 3 use Illuminate\Database\Eloquent\Model; 4 class Mahasiswa extends Model 5 { 6 protected \$table = "mahasiswa"; 7 } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Model Mahasiswa akan menangani tabel mahasiswa

4	<p>Buat controller baru yaitu MahasiswaController menggunakan php artisan php artisan make:controller MahasiswaController</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> 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 }</pre> <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 } 15</pre> <p>Keterangan:</p>

	<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	<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).</p> <pre> resources > views > index.blade.php 1 @extends('master') 2 3 <!-- Isi Judul --> 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 Mahasiswa 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 <th></th> 20 </tr> 21 </thead> 22 <tbody> 23 @foreach(\$mahasiswa as \$mhs) 24 <tr> 25 <td>{{ \$mhs->nama }}</td> 26 <td>{{ \$mhs->nim }}</td> 27 <td> 28 id }}" class="badge badge-info">Detail 29 id }}" class="badge badge-warning">Edit 30 id }}" class="badge badge-danger">Hapus 31 </td> 32 </tr> 33 @endforeach 34 </tbody> 35 </table> 36 @endsection </pre> <p>Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)</p>

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



Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
yasin	1001001004	Detail Edit Hapus

Data Mahasiswa		
Tambah Data Mahasiswa		
Nama	NIM	
alfan	1841720059	Detail Edit Hapus
Bianca	1841623011	Detail Edit Hapus

c. Memasukkan Data (Create) ke Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.</p> <pre> 22 23 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');</pre>

	<pre> 22 23 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 24 </pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> 16 public function tambah() 17 { 18 return view('tambah'); 19 } </pre> <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.</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 halaman --> 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="nimmhs">NIM</label> 21 <input type="number" class="form-control" required="required" name="nimmhs">
 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">Tambah Data</button> 32 </form> 33 @endsection </pre>

	Kita lakukan copy paste dari tambah.blade.php di Bagian 1
4	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre> 24 25 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php <pre> 25 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 26 </pre>
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre> 21 public function simpan(Request \$request) 22 { 23 Mahasiswa::create([24 'nama' => \$request->namamhs, 25 'nim' => \$request->nimmhs, 26 'email' => \$request->emailmhs, 27 'jurusan' => \$request->jurusanmhs 28]); 29 30 return redirect('/mahasiswa'); 31 } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Variabel \$request untuk menerima data yang akan ditambahkan ke database - Line 23-28 merupakan fitur eloquent menggunakan fungsi create() untuk insert data ke tabel mahasiswa <pre> public function simpan(Request \$request) { Mahasiswa::create(['nama' => \$request->namamhs, 'nim' => \$request->nimmhs, 'email' => \$request->emailmhs, 'jurusan' => \$request->jurusanmhs]); return redirect('/mahasiswa'); } </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> 7 class Mahasiswa extends Model 8 { 9 protected \$table = "mahasiswa"; 10 protected \$fillable = ['nama', 'nim', 'email', 'jurusan']; 11 } </pre>

	<pre>class Mahasiswa extends Model { protected \$table = "mahasiswa"; protected \$fillable = ['nama', 'nim', 'email', 'jurusan']; }</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>

Tambah Data

localhost:8000/mahasiswa/tambah

Tambah Data Mahasiswa

Kembali

Nama

rahmat

NIM

1001001006

E-mail

rahmat@gmail.com

Jurusan

teknik mesin

Tambah Data

Tambah Data Mahasiswa

[Kembali](#)

Nama

Monica Alvira Putri

NIM

1831720057

E-mail

monica@gmail.com

Jurusan

Teknik Sipil

[Tambah Data](#)

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

Home

localhost:8000/mahasiswa

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
nanda eka	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
marsha	1001001003	Detail Edit Hapus
yasin	1001001004	Detail Edit Hapus
rahmat	1001001006	Detail Edit Hapus

Data Mahasiswa

Tambah Data Mahasiswa

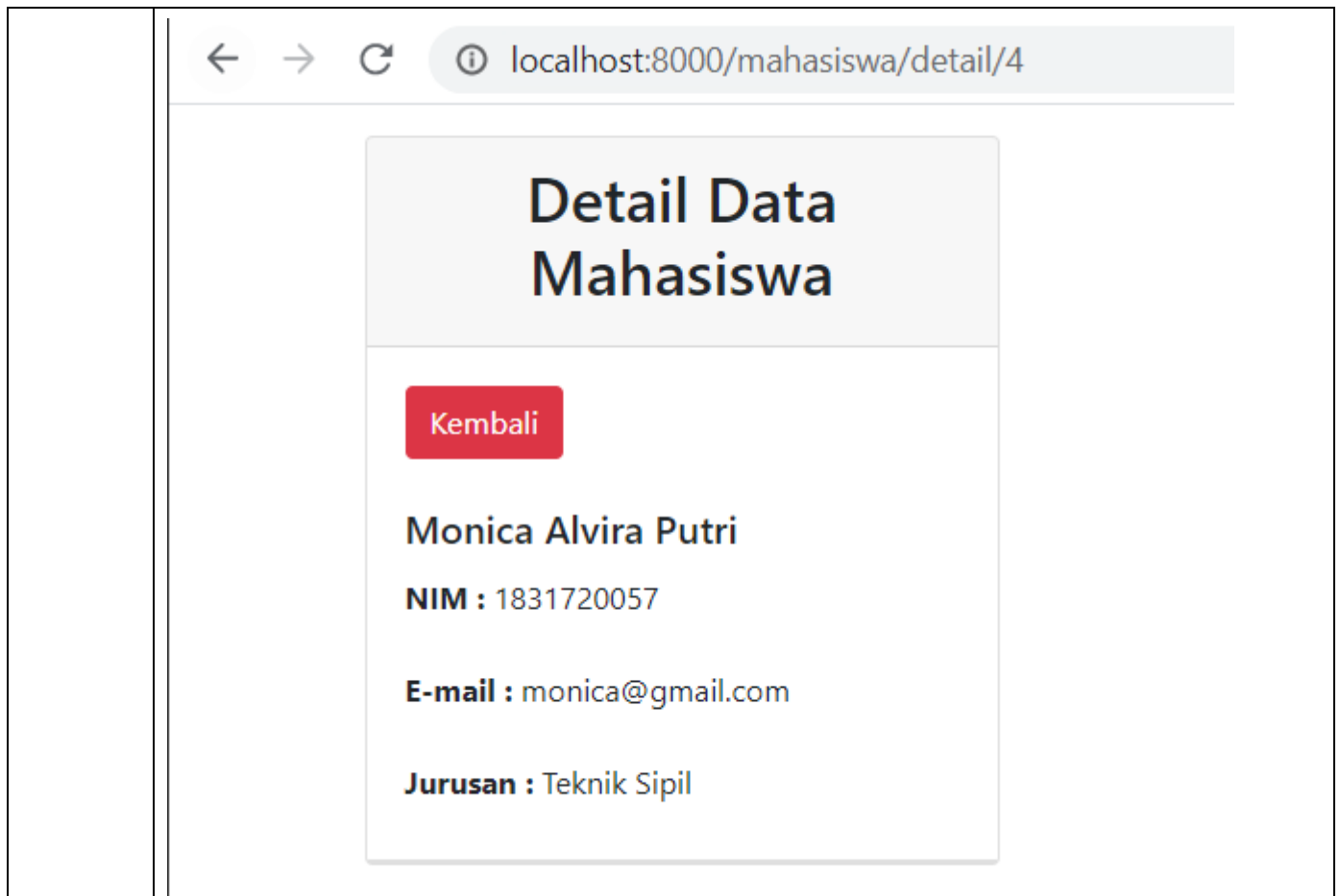
Nama	NIM	
alfan	1841720059	Detail Edit Hapus
Bianca	1841623011	Detail Edit Hapus
Monica Alvira Putri	1831720057	Detail Edit Hapus

d. Melihat Detail Data dari Database

Langkah	Keterangan
1	Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController

	<pre> 27 28 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 26 27 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> 33 public function detail(\$id) 34 { 35 \$mahasiswa = Mahasiswa::find(\$id); 36 return view('detail', ['mahasiswa' => \$mahasiswa]); 37 } </pre> <pre> public function detail(\$id) { \$mahasiswa = Mahasiswa::find(\$id); 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> EXPLORER web.php routes Mahasiswa.php app MahasiswaContro... edit.blade.php res... detail.blade.php r... LARAVEL-CRUD-KEDUA User.php bootstrap config database public resources js lang sass views detail.blade.php edit.blade.php index.blade.php master.blade.php tambah.blade.php welcome.blade.php routes resources > views > detail.blade.php 1 @extends('master') 2 3 <!-- isi title --> 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 Kembali 12 13
 14
 15 16 <h5 class="card-title">{{ \$mahasiswa->nama }} </h5> 17 <p class="card-text"> 18 <label for=""> NIM : </label> 19 {{ \$mahasiswa->nim }} </p> 20 <p class="card-text"> 21 <label for=""> E-mail : </label> 22 {{ \$mahasiswa->email }} </p> 23 <p class="card-text"> 24 <label for=""> Jurusan : </label> 25 {{ \$mahasiswa->jurusan }} </p> 26 @endsection </pre>

	<pre> resources > views > 🐛 detail.blade.php 1 @extends('master') 2 3 <!-- isi title --> 4 @section('title','Detail Mahasiswa') 5 6 <!-- isi bagian judul halaman --> 7 @section('judul_halaman','Detail Data Mahasiswa') 8 9 <!-- isi bagian konten halaman --> 10 @section('konten') 11 Kembali 12 13
 14
 15 16 <h5 class="card-title">{{\$mahasiswa->nama}}</h5> 17 <p class="card-text"> 18 <label for=""> NIM : </label> 19 {{\$mahasiswa->nim}} </p> 20 <p class="card-text"> 21 <label for=""> E-mail : </label> 22 {{\$mahasiswa->email}} </p> 23 <p class="card-text"> 24 <label for=""> Jurusan : </label> 25 {{\$mahasiswa->jurusan}} </p> 26 @endsection </pre>
5	Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.



e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController</p> <pre> 29 30 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 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>


```

39     public function edit($id)
40     {
41         $mahasiswa = Mahasiswa::find($id);
42         return view('edit', ['mahasiswa' => $mahasiswa]);
43     }

```

```

public function edit($id)
{
    $mahasiswa = Mahasiswa::find($id);
    return view('edit', ['mahasiswa' => $mahasiswa]);
}

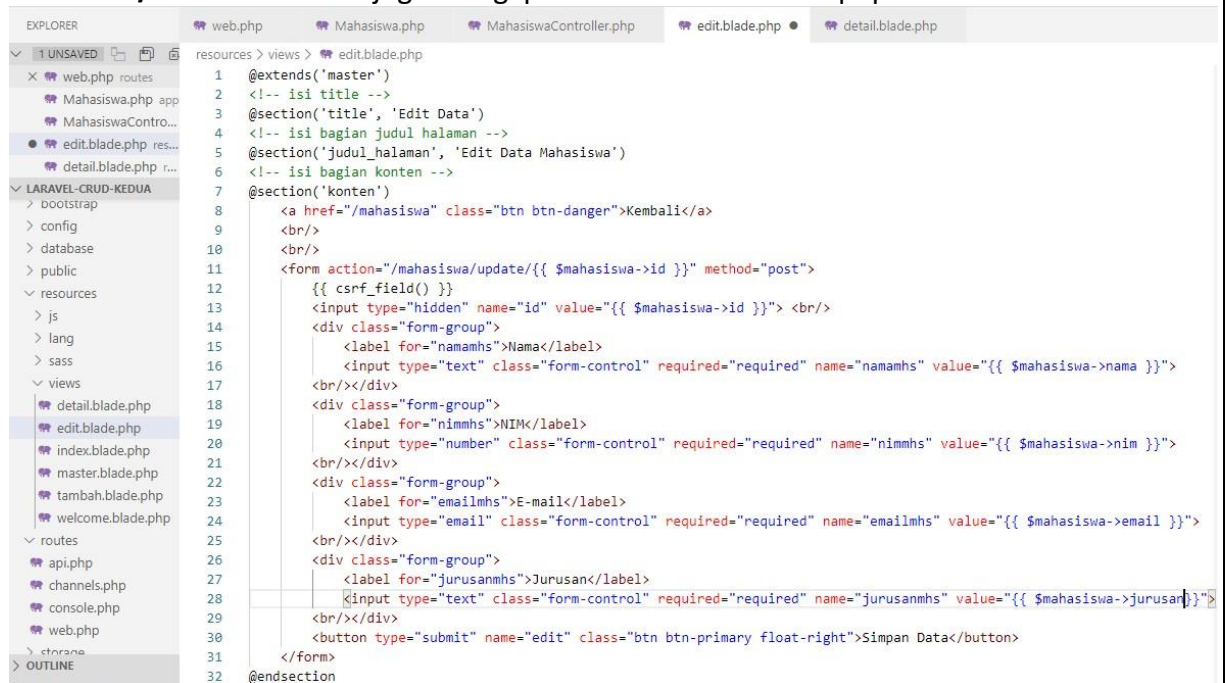
```

Keterangan :

- Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih

3

Kemudian buatlah view **edit.blade.php** yang berisi form untuk mengubah data pada folder **resources/views**. View edit juga mengaplikasikan master.blade.php.



```

1 @extends('master')
2 <!-- isi title -->
3 @section('title', 'Edit Data')
4 <!-- isi bagian judul halaman -->
5 @section('judul_halaman', 'Edit Data Mahasiswa')
6 <!-- isi bagian konten -->
7 @section('konten')
8     <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
9
10    <br/>
11    <form action="/mahasiswa/update/{{ $mahasiswa->id }}" method="post">
12        {{ csrf_field() }}
13        <input type="hidden" name="id" value="{{ $mahasiswa->id }}"> <br/>
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 @endsection

```

Keterangan:

- Line 11-31 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 11 terdapat action="/mahasiswa/update/{{ \$mahasiswa->id }}" yang menunjukkan routes /mahasiswa/update/{id} dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController

	<pre> resources > views > edit.blade.php 1 @extends('master') 2 3 <!-- isi title --> 4 @section('title','Detail Mahasiswa') 5 6 <!-- isi bagian judul halaman --> 7 @section('judul_halaman','Edit Data Mahasiswa') 8 9 <!-- isi bagian konten halaman --> 10 @section('konten') 11 Kembali 12 13
 14
 15 <form action="/mahasiswa/update" method="post"> 16 {{ csrf_field() }} 17 <input type="hidden" name="id" value="{{ \$mahasiswa->id }}">
 18 <div class="form-group"> 19 <label for="namamhs">Nama</label> 20 <input type="text" class="form-control" required="required" name="namamhs" value="{{ \$mahasiswa->nama }}">
 21 </div> 22 <div class="form-group"> 23 <label for="nimmhs">NIM</label> 24 <input type="number" class="form-control" required="required" name="nimmhs" value="{{ \$mahasiswa->nim }}">
 25 </div> 26 <div class="form-group"> 27 <label for="emailmhs">E-mail</label> 28 <input type="email" class="form-control" required="required" name="emailmhs" value="{{ \$mahasiswa->email }}">
 29 </div> 30 <div class="form-group"> 31 <label for="jurusanmhs">Jurusan</label> 32 <input type="text" class="form-control" required="required" name="jurusanmhs" value="{{ \$mahasiswa->jurusan }}">
 33 </div> 34 <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button> 35 </form> 36 @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 32 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update'); </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php <pre> 30 31 Route::post('/mahasiswa/update', 'MahasiswaController@update'); </pre>
5	<p>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</p>

```

45     public function update($id, Request $request)
46     {
47         $mahasiswa = Mahasiswa::find($id);
48         $mahasiswa->nama = $request->namamhs;
49         $mahasiswa->nim = $request->nimmhs;
50         $mahasiswa->email = $request->emailmhs;
51         $mahasiswa->jurusan = $request->jurusanmhs;
52         $mahasiswa->save();
53         return redirect('/mahasiswa');
54     }

```

```

46     public function update(Request $request)
47     {
48         $mahasiswa = Mahasiswa::find($request->id);
49         $mahasiswa->nama = $request->namamhs;
50         $mahasiswa->nim = $request->nimmhs;
51         $mahasiswa->email = $request->emailmhs;
52         $mahasiswa->jurusan = $request->jurusanmhs;
53         $mahasiswa->save();
54         return redirect(['/mahasiswa']);
55     }

```

Keterangan:

- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 48-52 merupakan fungsi eloquent untuk update data ke tabel 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.

Edit Data Mahasiswa

Kembali

Nama

NIM

E-mail

Jurusan

Simpan Data

<h2 style="text-align: center;">Data Mahasiswa</h2> <div style="text-align: center; margin-bottom: 10px;"> Tambah Data Mahasiswa </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 35%;">Nama</th> <th style="width: 35%;">NIM</th> <th style="width: 30%;"></th> </tr> </thead> <tbody> <tr> <td>alfan</td> <td>1841720059</td> <td style="text-align: center;"> Detail Edit Hapus </td> </tr> <tr> <td>Bianca</td> <td>1841623011</td> <td style="text-align: center;"> Detail Edit Hapus </td> </tr> <tr> <td>Monica Alvira Putri Nasruddin</td> <td>1831720058</td> <td style="text-align: center;"> Detail Edit Hapus </td> </tr> </tbody> </table>			Nama	NIM		alfan	1841720059	Detail Edit Hapus	Bianca	1841623011	Detail Edit Hapus	Monica Alvira Putri Nasruddin	1831720058	Detail Edit Hapus
Nama	NIM													
alfan	1841720059	Detail Edit Hapus												
Bianca	1841623011	Detail Edit Hapus												
Monica Alvira Putri Nasruddin	1831720058	Detail Edit Hapus												

f. Menghapus Data (Delete) dari Database

Langkah	Keterangan
1	<p>Buatlah <i>route</i> baru pada routes/web.php dengan nama /mahasiswa/hapus yang akan menjalankan fungsi hapus pada MahasiswaController</p> <pre> 33 34 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 32 33 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> 56 public function hapus(\$id) 57 { 58 \$mahasiswa = Mahasiswa::find(\$id); 59 \$mahasiswa->delete(); 60 return redirect('/mahasiswa'); 61 } </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih <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	
alfan	1841720059	Detail Edit Hapus
Bianca	1841623011	Detail Edit Hapus
Monica Alvira Putri Nasruddin	1831720058	Detail Edit Hapus

Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	
alfan	1841720059	Detail Edit Hapus
Bianca	1841623011	Detail Edit Hapus

Link source code: <https://github.com/alfan06/Laravel-CRUD>

-- Selamat Mengerjakan --