LAPORAN PRAKTIKUM

PEMROGRAMAN WEB LANJUT

CRUD Laravel

Oleh:

MAHENDRA FIRMANSYAH

NIM. 1841720027

TI-2A

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK NEGERI MALANG

a. Konfigurasi Database dan Pembuatan Tabel di MySQL

Langkah	Keterangan

1	Buatlah project Laravel baru dengan nama **laravel-crud**. Buka command prompt, tuliskan perintah berikut.
	`cd C:\xampp\htdocs`
	`laravel new laravel-crud`
	`C:\Users\Mahendra>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/phpoption (1.7.3): Loading from cache`
	`  - Installing vlucas/phpdotenv (v4.1.4): Downloading (100%)`

2	Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file **.env** pada project laravel-crud. Ubah seperti di bawah ini.
	EXPLORER
	OPEN EDITORS
	LARAVEL-CRUD
	app
	bootstrap
	config
	database
	public
	resources
	routes
	storage
	tests
	vendor
	.editorconfig
	.env
	.env
	`1 APP_NAME=Laravel`
	`2 APP_ENV=local`
	`3 APP_KEY=base64:aJdRpfTYHHBEPVavj5S8Cw8zF11FhxNtyJww3FYjjAE=`
	`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.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:zzLbguV/97GiybLQUgd7mCojFDQUJKWwe68p0uznfwQ=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=latihan_laravel
DB_USERNAME=root
DB_PASSWORD=
```

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

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
	1	**id**	int(11)			No	*None*		AUTO_INCREMENT
	2	**nama**	varchar(100)	utf8mb4_general_ci		No	*None*		
	3	**nim**	varchar(10)	utf8mb4_general_ci		No	*None*		
	4	**email**	varchar(50)	utf8mb4_general_ci		No	*None*		
	5	**jurusan**	varchar(20)	utf8mb4_general_ci		No	*None*		

HASIL :

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
	1	**id**	int(50)			No	*None*		AUTO_INCREMENT	Change Drop More
	2	**nama**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
	3	**nim**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
	4	**email**	varchar(250)	utf8mb4_general_ci		No	*None*			Change Drop More
	5	**jurusan**	int(250)			No	*None*			Change Drop More

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

+ Options

←T→			id	nama	nim	email	jurusan
Edit	Copy	Delete	1	nanda	1001001001	nandan@gmail.com	teknik informatika
Edit	Copy	Delete	2	wahyu	1001001002	wahyu@gmail.com	teknik elektro

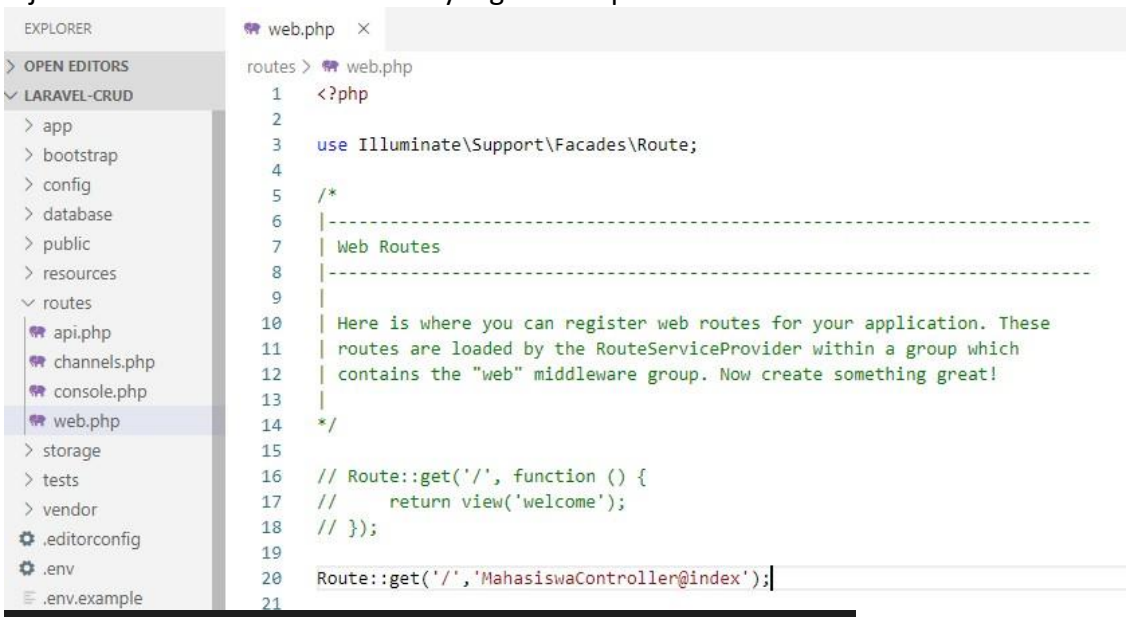
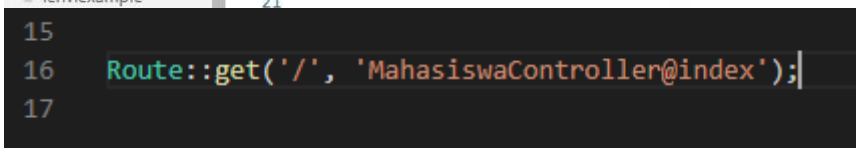
HASIL :

+ Options

←T→			id	nama	nim	email	jurusan
Edit	Copy	Delete	1	nanda	1001001001	nandan@gmail.com	Teknik Informatika
Edit	Copy	Delete	2	wahyu	1001001002	wahyu@gmail.com	Teknik Elektro

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.   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**

EXPLORER | web.php | MahasiswaController.php

app > Http > Controllers > MahasiswaController.php > MahasiswaController > index

```
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
```

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     // mengambil data dari tabel mahasiswa
11     $mahasiswa = DB::table('mahasiswa')=>get();
12
13     // mengirim data mahasiswa ke view index
14     return view('index',['mahasiswa' => $mahasiswa]);
15 }
16
```

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

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**.

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

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-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 class="card">
22             </div>
23         </div>
24     </div>
25 </body>
26 </html>
```

resources > views > master.blade.php > html > body > div.container > div.row.mt-3 >

```
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/
6  </head>
7  <body>
8      <div class="container">
9          <div class="row mt-3">
10             <div class="col-md-6">
11                 <div class="card">
12                     <div class="card-header text-center">
13                         <!-- bagian judul halaman -->
14                         <h2> @yield('judul_halaman') </h2>
15                     </div>
16                     <div class="card-body">
17                         <!-- bagian konten blog -->
18                         @yield('konten')
19                     </div>
20                 <div class="card">
21                 </div>
22             </div>
23         </div>
24 </body>
25 </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

5

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

EXPLORER | master.blade.php | index.blade.php

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
```

resources > views > index.blade.php > table.table.table-bordered.table-hover.table-striped > thead

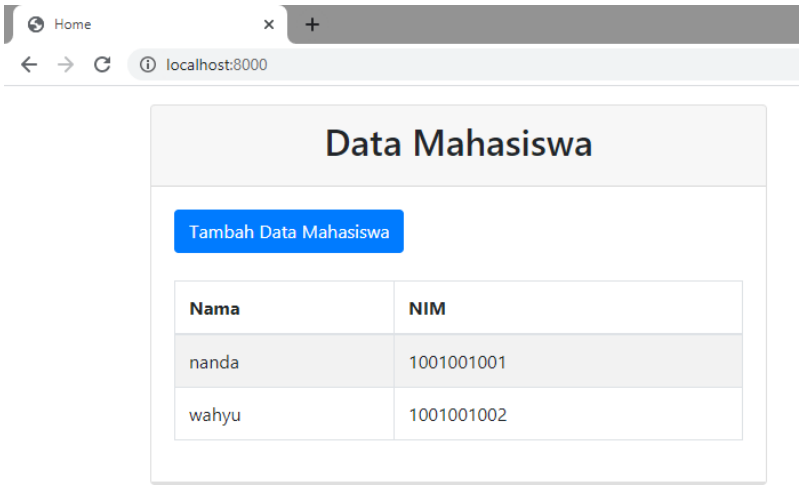
```
1  @extends('master')
2
3  <!-- isi title -->
4  @section('title', 'Home')
5
6  <!-- isi bagian judul halaman -->
7  @section('judul_halaman', 'Data Mahasiswa')
8
9  <!-- isi bagian konten -->
10 @section('konten')
11     <a href="/mahasiswa/tambah" class="btn btn-primary">Tambah Data Mahasiswa
12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

Keterangan:

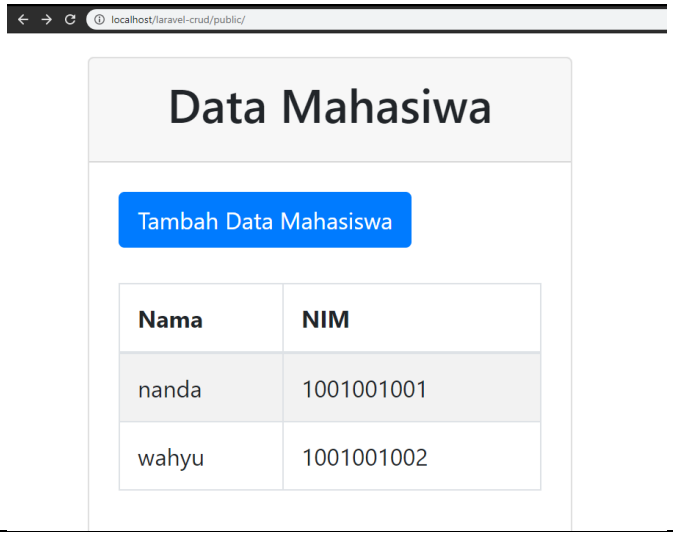
- Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan bladelain yaitu master.blade.php

	- 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	Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud
	`php artisan serve`
	`C:\xampp\htdocs\laravel-crud>php artisan serve`
	`Laravel development server started: http://127.0.0.1:8000`
	`[Tue Apr 14 00:15:02 2020] PHP 7.4.3 Development Server (http://127.0.0.1:8000) started`
	Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut



HASIL :

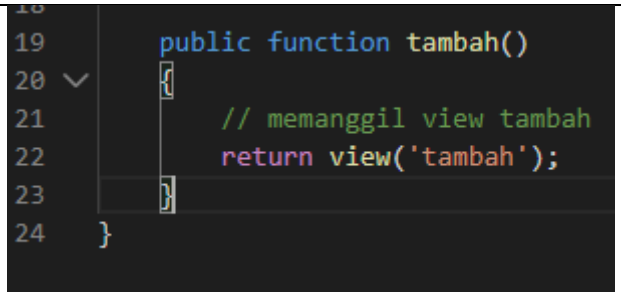


c. Memasukkan Data (Create) ke Database

Langkah	Keterangan

1	Buatlah *route* baru pada **routes/web.php** dengan nama **/mahasiswa/tambah** yang akan menjalankan fungsi tambah pada MahasiswaController
	`21`
	`22 Route::get('/mahasiswa/tambah','MahasiswaController@tambah');`
	`15`
	`16 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');`
	`17`

2	Buat method **tambah** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menampilkan view tambah.
	`19 public function tambah()`
	`20 {`
	`21 // memanggil view tambah`
	`22 return view('tambah');`
	`23 }`
	`24`

	 19 public function tambah() 20 { 21 // memanggil view tambah 22 return view('tambah'); 23 } 24 }
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.

```
EXPLORER
OPEN EDIT... 2 UNSAVED
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
OUTLINE
NPM SCRIPTS
```

index.blade.php | web.php ● | MahasiswaController.php | tambah.blade.php ●

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     <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
11     <br/>
12     <br/>
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"> <br/>
18         </div>
19         <div class="form-group">
20             <label for="nimmhs">NIM</label>
21             <input type="number" class="form-control" required="required" name="nimmhs"> <br/>
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"> <br/>
26         </div>
27         <div class="form-group">
28             <label for="jurusanmhs">Jurusan</label>
29             <input type="text" class="form-control" required="required" name="jurusanmhs"> <br/>
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 -->
9  @section('konten')
10     <a href="/mahasiswa" class="btn btn-danger">Kembali</a>
11     <br/>
12     <br/>
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"> <br/>
18     </div>
19     <div class="form-group">
20         <label for="nimmhs">NIM</label>
21         <input type="number" class="form-control" required="required" name="nimmhs"> <br/>
22     </div>
23     <div class="form-group">
24         <label for="jurusanmhs">Jurusan</label>
25         <input type="text" class="form-control" required="required" name="jurusanmhs"> <br/>
26     </div>
27     <button type="submit" name="tambah" class="btn btn-primary float-right">Tambah Data</button>
28 </form>
29 @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 pada controller MahasiswaController
- Line 14 terdapat {{ csrf_field() }} yang digunakan untuk menerapkan fitur laravel yaitu csrf protection. csrf protection adalah fitur keamanan untuk mencegah penginputan dari luar aplikasi, csrf akan men-generate kode token otomatis yang dibuat dalam bentuk *form hidden*.

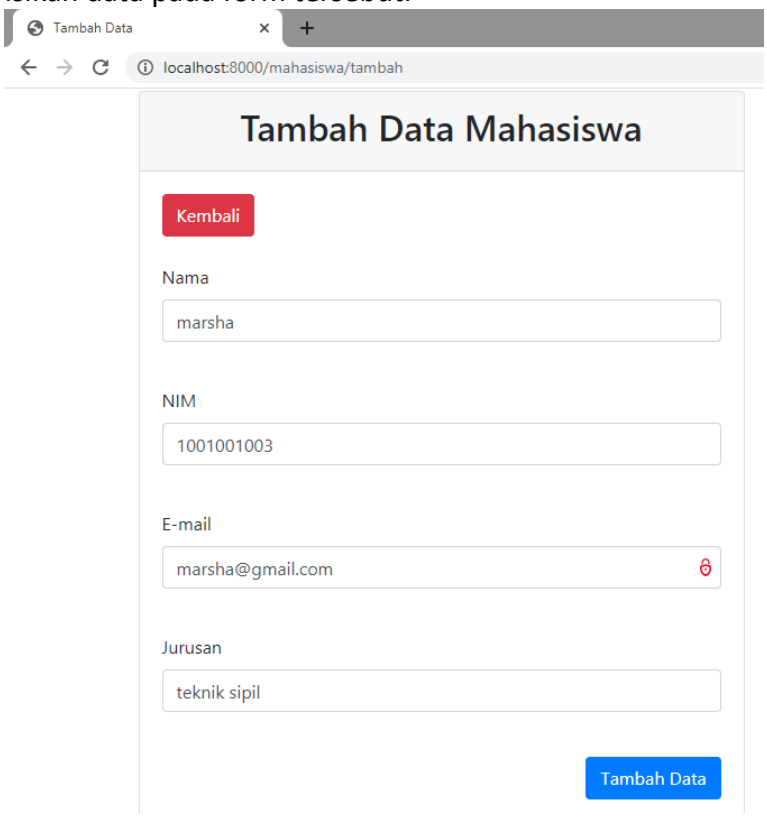
4	Ketika tombol simpan ditekan, akan dipanggil routes **/mahasiswa/simpan**. Oleh karena itu, kita buat terlebih dahulu route tersebut. 23 24 `Route::post('/mahasiswa/simpan','MahasiswaController@simpan');` 20 `Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');` 21 Keterangan: - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php
5	Buat method **simpan** pada **MahasiswaController** untuk menyimpan data ke database.

```
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  }
```

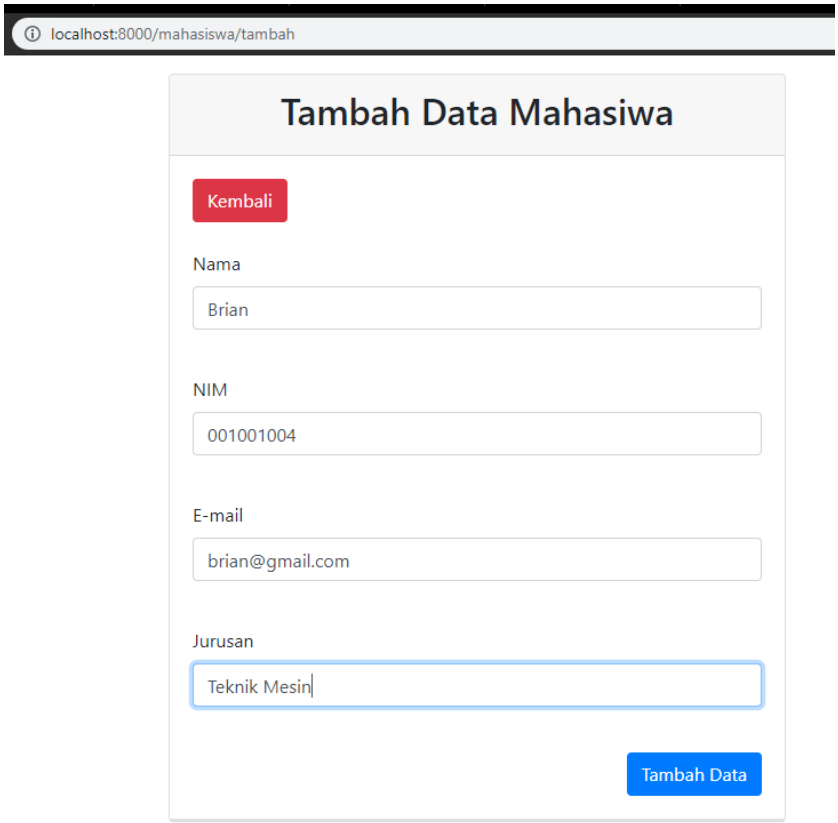
Keterangan:

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

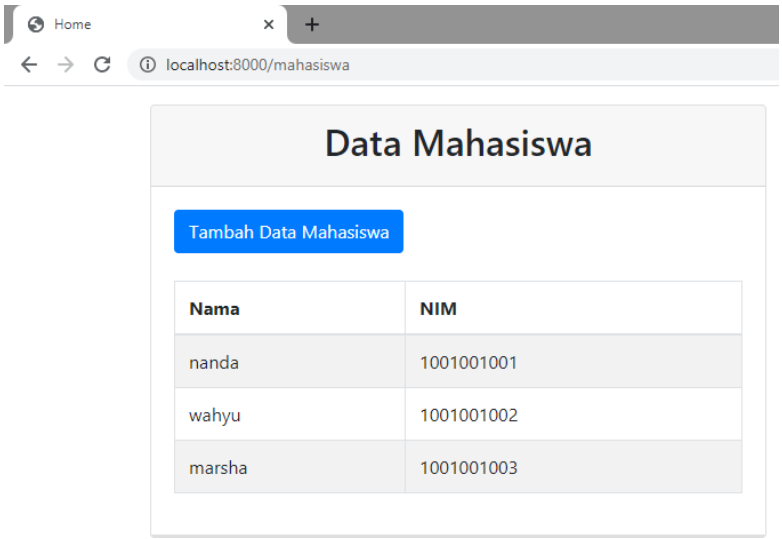
6	Kembali coba jalankan localhost:8000 dan pilih tombol 'Tambah Data Mahasiswa', maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.



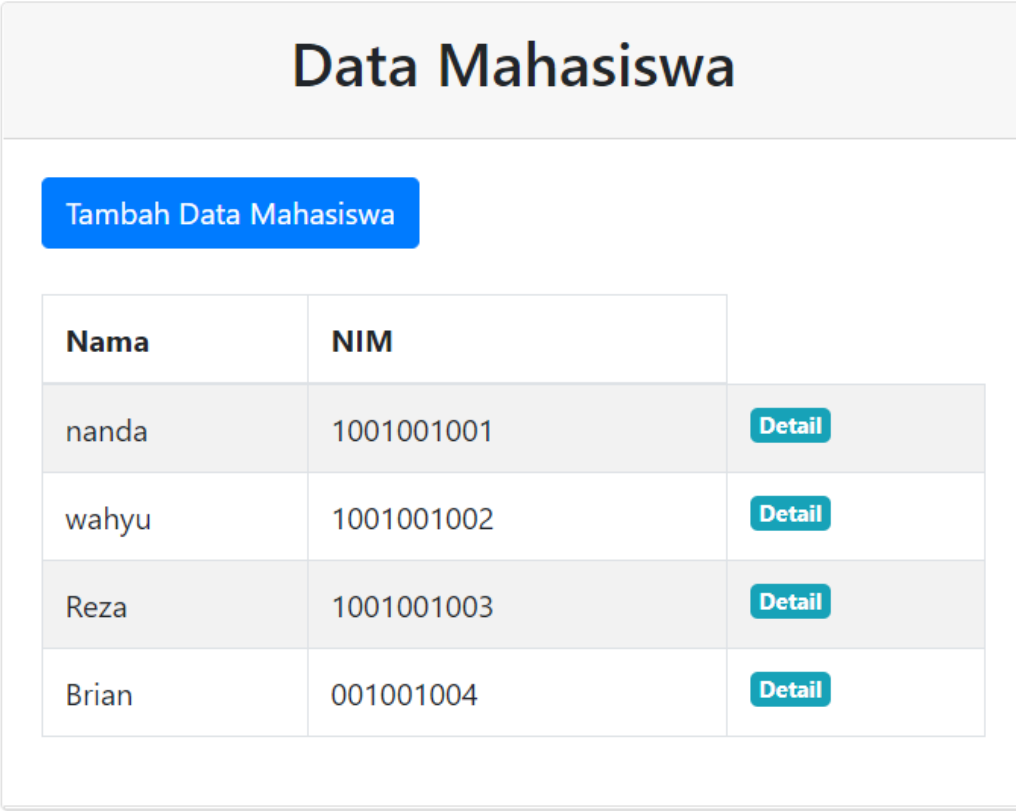
HASIL : :



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



HASIL :



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

d. Melihat Detail Data dari Database

Langkah	Keterangan
1	Buatlah tombol untuk melihat detail data (Line 28) pada file **index.blade.php** di folder **resources/views**
2	Buatlah *route* baru pada **routes/web.php** dengan nama **/mahasiswa/detail** yang akan menjalankan fungsi detail pada MahasiswaController
3	Buat method **detail** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menampilkan data mahasiswa pada view detail.blade.php.

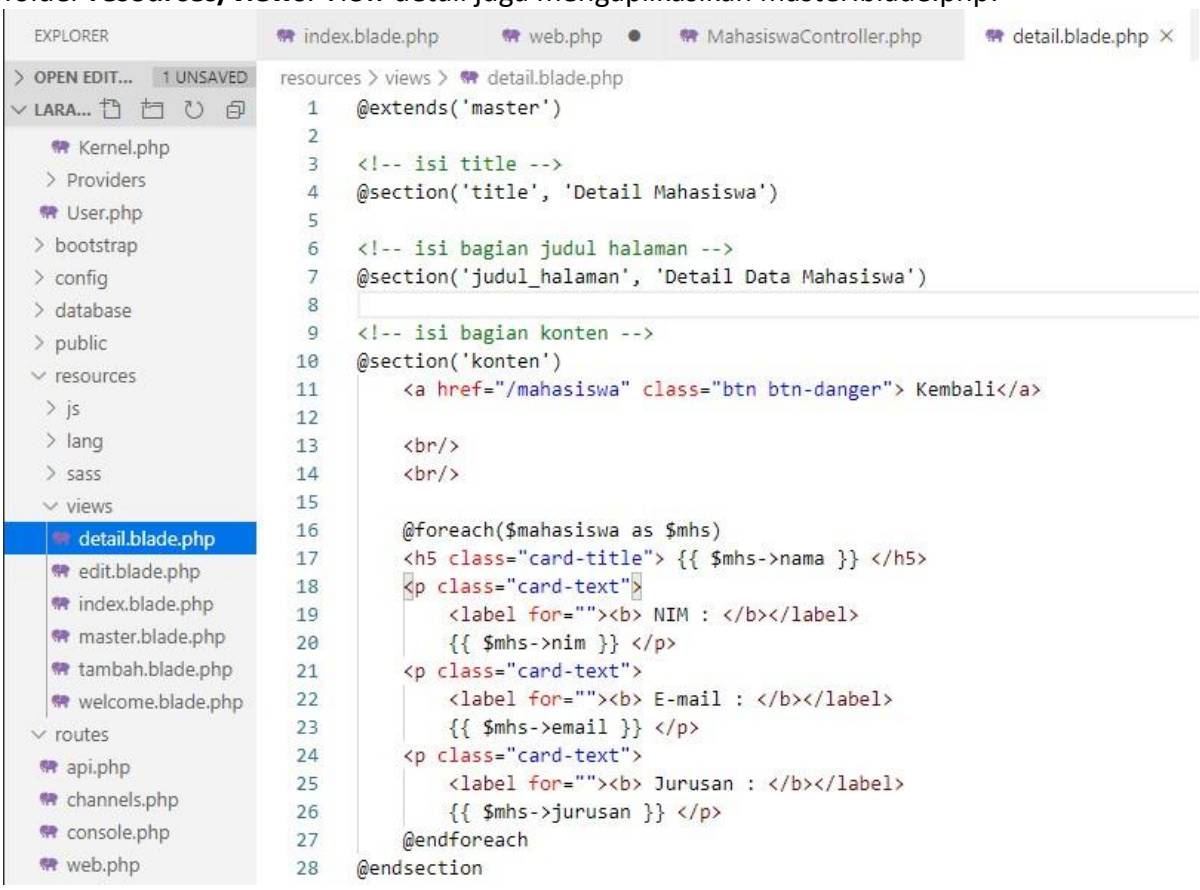
```
<td>
    <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a>
</td>
```

```
25
26    Route::get('/mahasiswa/detail/{id}','MahasiswaController@detail');
```

```
Route::get('/', 'MahasiswaController@index');
Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');
Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');
Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');
```

```php
    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]);
    }
}
```

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



<table>
<tr>
<td></td>
<td>

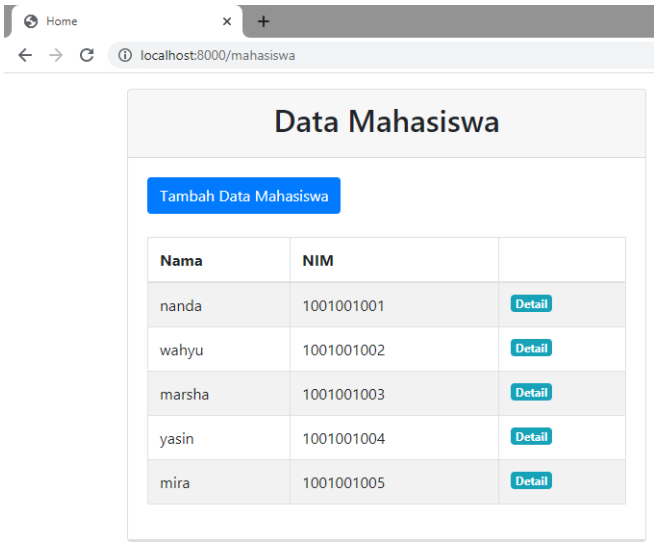
resources > views > detail.blade.php > ...

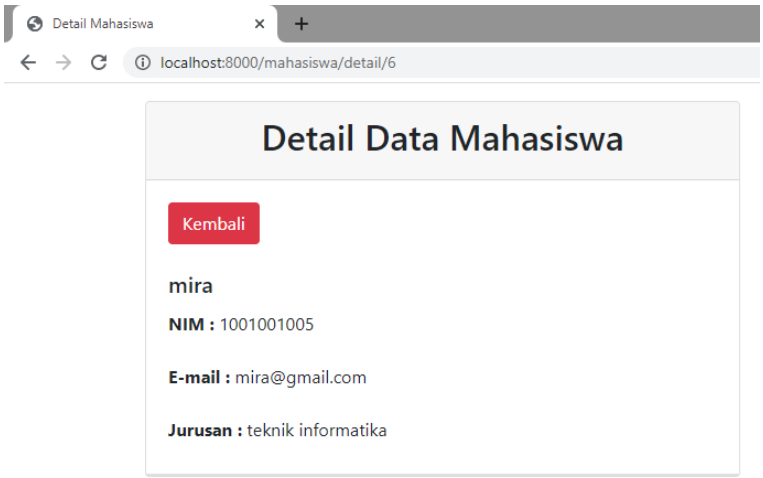
```
1   @extends('master')
2
3   <!-- isi title -->
4   @section('title', 'Detail Mahasiswa')
5
6   <!-- isi bagian judul halaman -->
7   @section('judul_halaman', 'Detail Data Mahasiswa')
8
9   <!-- isi bagian konten -->
10  @section('konten')
11      <a href="/mahasiswa" class="btn btn-danger">Kembali</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=""> E-mail : </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:

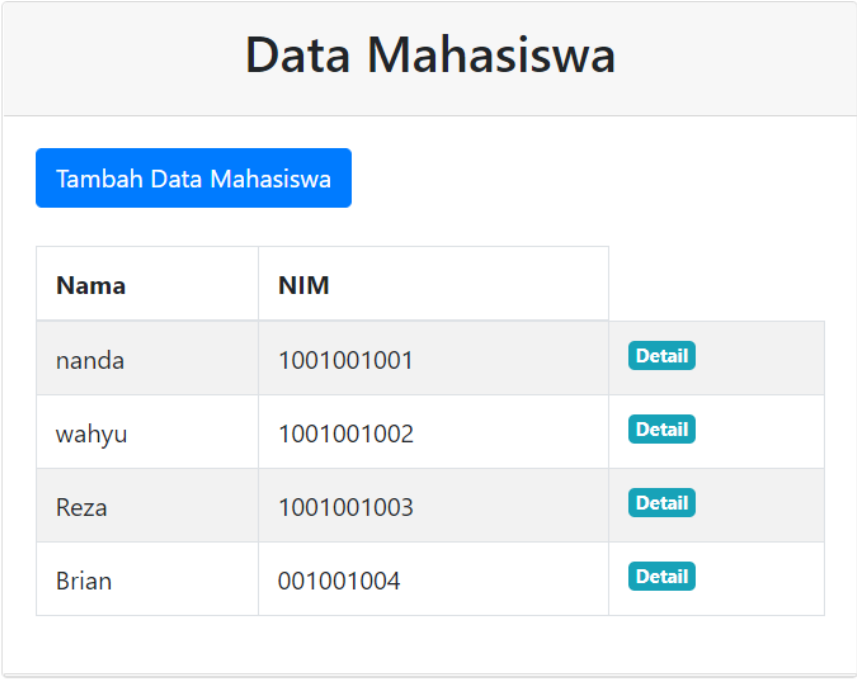
- Line 16-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan

</td>
</tr>
<tr>
<td>5</td>
<td>Jalankan localhost:8000 dan pilih tombol ‘Detail’ di suatu data yang ingin kita lihat detailnya.</td>
</tr>
</table>


Home
localhost:8000/mahasiswa
Data Mahasiswa
Tambah Data Mahasiswa
Nama
NIM
nanda
1001001001
Detail
wahyu
1001001002
Detail
marsha
1001001003
Detail
yasin
1001001004
Detail
mira
1001001005
Detail

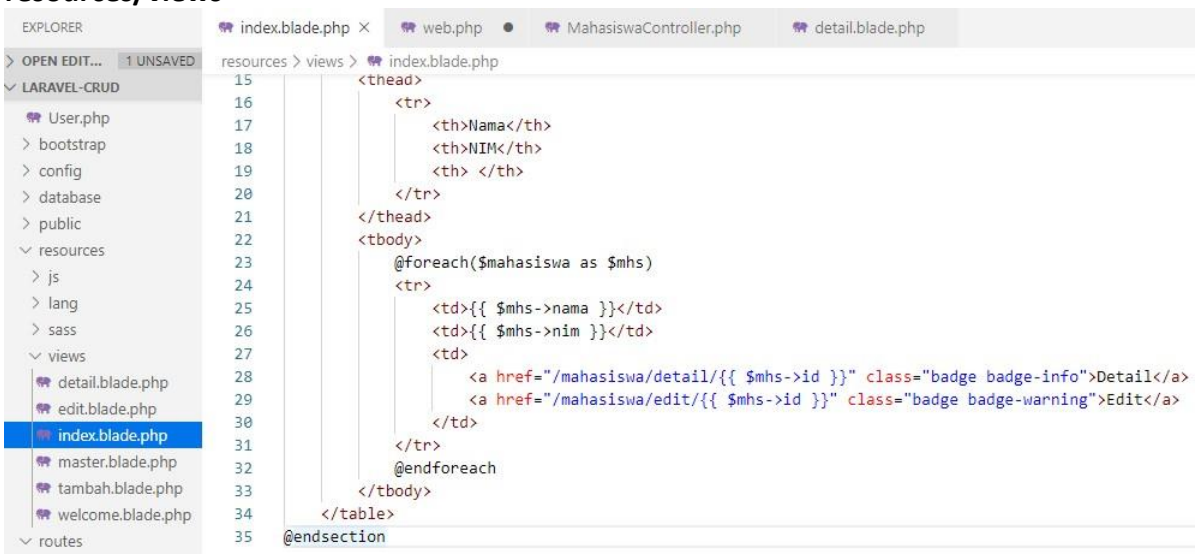

Detail Mahasiswa
localhost:8000/mahasiswa/detail/6
Detail Data Mahasiswa
Kembali
mira
NIM : 1001001005
E-mail : mira@gmail.com
Jurusan : teknik informatika

HASIL :


Data Mahasiswa
Tambah Data Mahasiswa
Nama
NIM
nanda
1001001001
Detail
wahyu
1001001002
Detail
Reza
1001001003
Detail
Brian
001001004
Detail

e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	Buatlah tombol untuk mengubah data (Line 29) pada file **index.blade.php** di folder **resources/views**



	<td> <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a> <a href="/mahasiswa/edit/{{ $mhs->id }}" class="badge badge-warning">Edit</a> </td>
2	Buatlah *route* baru pada **routes/web.php** dengan nama **/mahasiswa/edit** yang akan menjalankan fungsi edit pada MahasiswaController

```
27
28  Route::get('/mahasiswa/edit/{id}','MahasiswaController@edit');
```

```
20  Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');
```

3

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

```
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      }
```

```
45      public function edit($id)
46      {
47          // mengambil data mahasiswa berdasarakan id yang di pilih
48          $mahasiswa = DB::table('mahasiswa')->where('id',$id)->get();
49          //kirim data mahasiswa yang diambil ke view edit.blade.php
50          return view('edit', ['mahasiswa' => $mahasiswa]);
51      }
52
```

Keterangan :

- Line 49 : query builder untuk mengambil data mahasiswa berdasarkan id yang dipilih

4

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

```
EXPLORER
OPEN EDIT... 2 UNSAVED
LARAVEL-CRUD
  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
  storage
  tests
  vendor
OUTLINE
NPM SCRIPTS
```

index.blade.php | web.php | MahasiswaController.php | edit.blade.php

resources > views > edit.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     <br/>
14     @foreach($mahasiswa as $mhs)
15     <form action="/mahasiswa/update" method="post">
16         {{ csrf_field() }}
17         <input type="hidden" name="id" value="{{ $mhs->id }}"> <br/>
18         <div class="form-group">
19             <label for="namamhs">Nama</label>
20             <input type="text" class="form-control" required="required" name="namamhs" value="{{ $mhs->nama }}"> <br/>
21         </div>
22         <div class="form-group">
23             <label for="nimmhs">NIM</label>
24             <input type="number" class="form-control" required="required" name="nimmhs" value="{{ $mhs->nim }}"> <br/>
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="{{ $mhs->email }}"> <br/>
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="{{ $mhs->jurusan }}"> <br/
33         </div>
34         <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button>
35     </form>
36     @endforeach
37 @endsection
```

```
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="/" class="btn btn-danger">Kembali</a>
12     <br/>
13     <br/>
14     @foreach($mahasiswa as $mhs)
15     <form action="/mahasiswa/update" method="post">
16         {{ csrf_field() }}
17         <input type="hidden" name="id" value="{{ $mhs->id }}"> <br/>
18         <div class="form-group">
19             <label for="namamhs"> Nama </label>
20             <input type="text" class="form-control" required="required" name="namamhs" value="{{ $mhs->nama }}"> <br/>
21         </div>
22         <div class="form-group">
23             <label for="nimmhs"> NIM </label>
24             <input type="number" class="form-control" required="required" name="nimmhs" value="{{ $mhs->nim }}"> <br/>
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="{{ $mhs->email }}"> <br/>
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="{{ $mhs->jurusan }}"> <br/>
33         </div>
34         <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button>
35             </form>
36             @endforeach
37         @endsection
```

Keterangan:

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

5	Ketika tombol simpan ditekan, akan dipanggil routes **/mahasiswa/update**. Oleh karena itu, kita buat terlebih dahulu route tersebut. 29 30 `Route::post('/mahasiswa/update','MahasiswaController@update');` 21 `Route::post('/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.

```
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    }
```

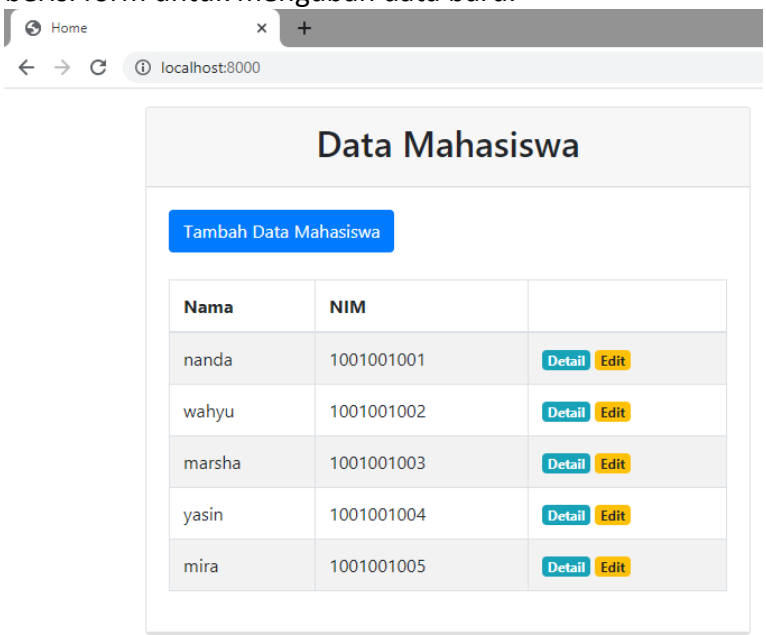
```
53    public function update(Request $request)
54    {
55        // update data mahasiswa
56        DB::table('mahasiswa')->where('id', $request->id)->update([
57            'nama' => $request->namamhs,
58            'nim' => $request->nimmhs,
59            'email' => $request->emailmhs,
60            'jurusan' => $request->jurusanmhs
61        ]);
62        return redirect('/');
63    }
```

Keterangan:

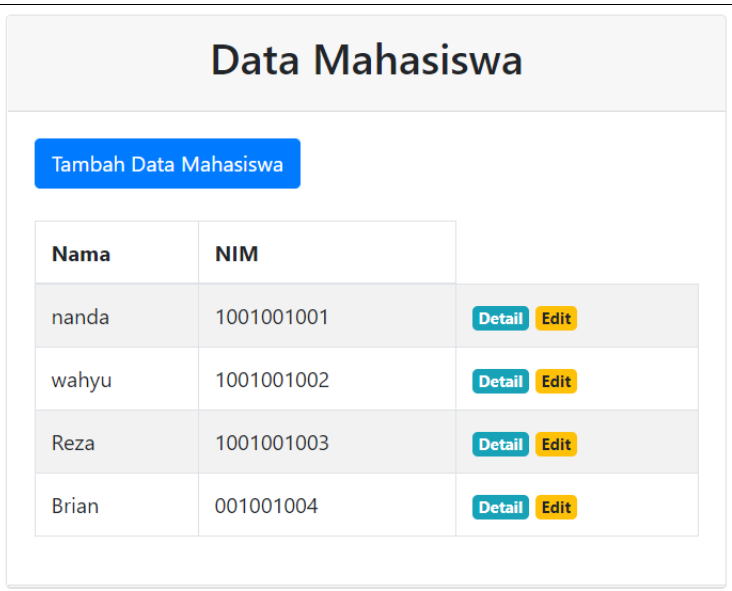
- Variabel $request untuk menerima data yang akan ditambahkan ke database
- Line 58-63 merupakan query builder untuk update data ke tabel mahasiswa

7

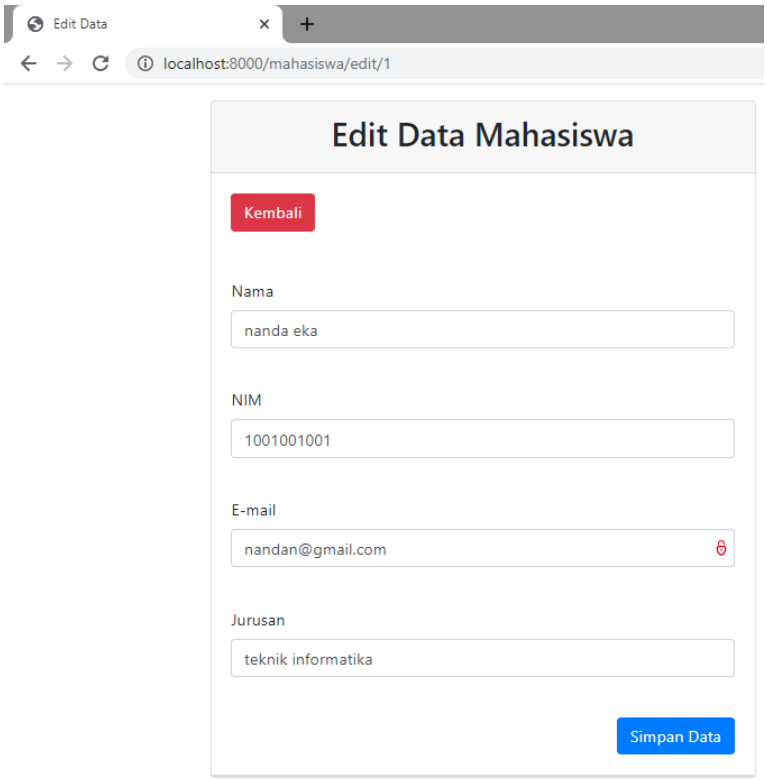
Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru.



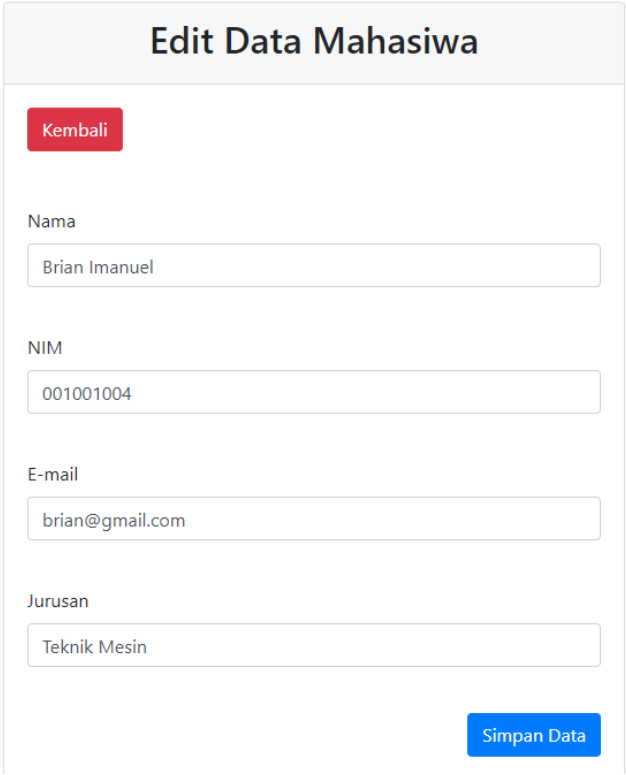
HASIL :



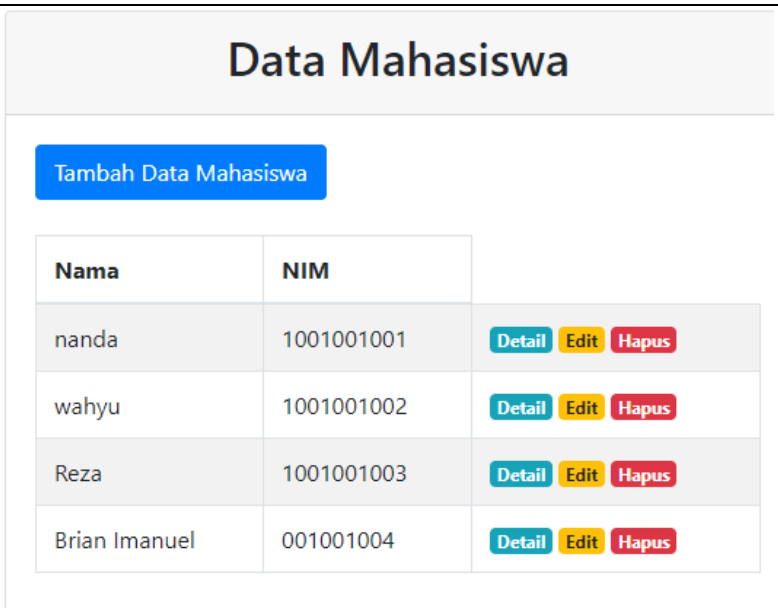
Klik edit di data pertama, kita akan mengubah namanya.



HASIL :



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



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**

```
EXPLORER
OPEN EDIT... 3 UNSAVED
LARAVEL-CRUD
User.php
bootstrap
config
database
public
resources
js
lang
sass
views
detail.blade.php
edit.blade.php
index.blade.php
master.blade.php
```

index.blade.php ● | web.php ● | MahasiswaController.php | edit.blade.php ●

resources > views > index.blade.php

```
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                <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a>
29                <a href="/mahasiswa/edit/{{ $mhs->id }}" class="badge badge-warning">Edit</a>
30                <a href="/mahasiswa/hapus/{{ $mhs->id }}" class="badge badge-danger">Hapus</a>
31            </td>
32        </tr>
33        @endforeach
```

```
<td>
    <a href="/mahasiswa/detail/{{ $mhs->id }}" class="badge badge-info">Detail</a>
    <a href="/mahasiswa/edit/{{ $mhs->id }}" class="badge badge-warning">Edit</a>
    <a href="/mahasiswa/hapus/{{ $mhs->id }}" class="badge badge-danger">Hapus</a>
</td>
```

2

Buatlah *route* baru pada **routes/web.php** dengan nama **/mahasiswa/hapus** yang akan menjalankan fungsi hapus pada MahasiswaController

```
31
32    Route::get('/mahasiswa/hapus/{id}','MahasiswaController@hapus');
```

```
22    Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');
```

3

Buat method **hapus** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menjalankan fungsi hapus.

```
EXPLORER
OPEN EDIT... 3 UNSAVED
LARAVEL-CRUD
app
Console
Exceptions
Http
Controllers
Controller.php
MahasiswaContro...
Middleware
Kernel.php
```

index.blade.php ● | web.php ● | MahasiswaController.php × | edit.blade.php ●

app > Http > Controllers > MahasiswaController.php > MahasiswaController > tambah

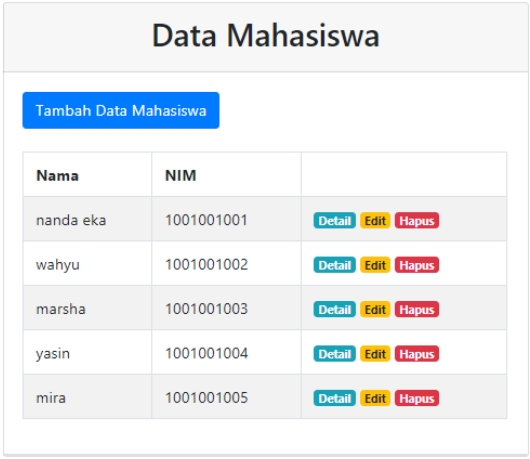
```
65        }
66
67        public function hapus($id)
68        {
69            // menghapus data mahasiswa berdasarkan id yang dipilih
70            DB::table('mahasiswa')->where('id',$id)->delete();
71
72            return redirect('/mahasiswa');
73        }
74    }
75
```

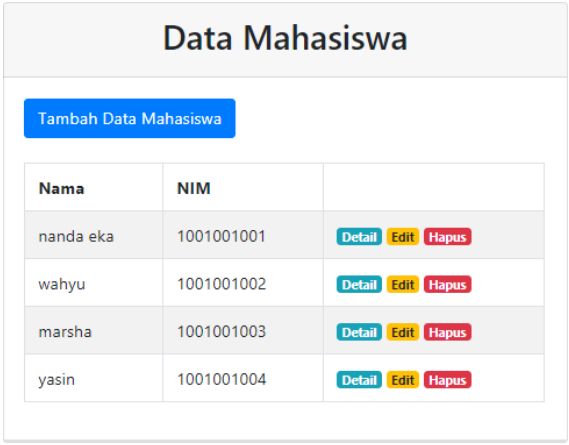
```
65        public function hapus($id)
66        {
67            // mengahpus data mahasiswa berdasarkan id yang dipilih
68            DB::table('mahasiswa')->where('id',$id)->delete();
69
70            return redirect('/');
71        }
```

Keterangan :

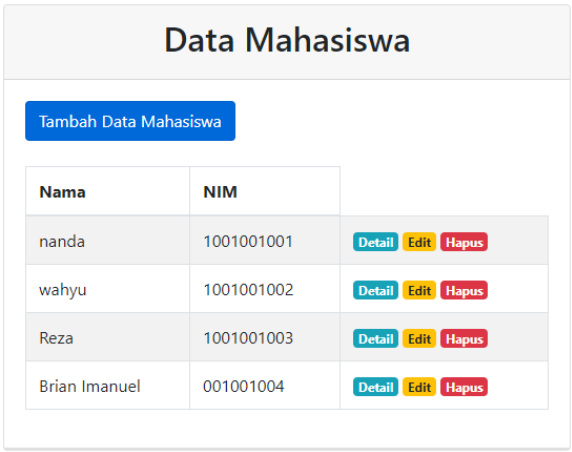
- Line 67 : query builder untuk menghapus data mahasiswa berdasarkan id yang dipilih

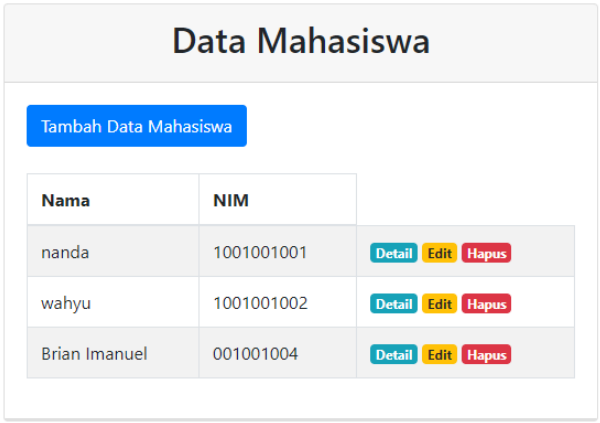
4	Jalankan localhost:8000 dan pilih tombol ‘Hapus’, maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir.





HASIL :





Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent

Eloquent adalah sebuah fitur untuk mengelola data yang ada pada database. Eloquent ORM menyediakan fungsi-fungsi active record, atau fungsi-fungsi query sql untuk mengelola data pada database. Dengan menggunakan Eloquent pada Laravel, kita bisa mengelola data pada database dari hanya satu buah model.

a. Konfigurasi Database

Langkah	Keterangan

1	Buatlah project Laravel baru dengan nama **laravel-crud-kedua**. Buka command prompt, tuliskan perintah berikut.
	`cd C:\xampp\htdocs`
	`laravel new laravel-crud-kedua`
	
	`C:\xampp\htdocs>laravel new laravel-crud-kedua`
	`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`

2	Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file **.env** pada project laravel-crud. Ubah seperti di bawah ini.
	EXPLORER
	OPEN EDITORS
	LARAVEL-CRUD
	app
	bootstrap
	config
	database
	public
	resources
	routes
	storage
	tests
	vendor
	.editorconfig
	.env
	
	.env
	`1 APP_NAME=Laravel`
	`2 APP_ENV=local`
	`3 APP_KEY=base64:aJdRpfTYHHBEPVavj5S8Cw8zF11FhxNtyJww3FYjjAE=`
	`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`

<table>
<tr>
<td></td>
<td>

.env.example

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=latihan_laravel
DB_USERNAME=root
DB_PASSWORD=
```

Keterangan:

- Nama database yang akan digunakan adalah latihan_laravel dengan username root.
</td>
</tr>
<tr>
<td>3</td>
<td>

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

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

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**id**	int(50)			No	*None*		AUTO_INCREMENT	Change Drop More
2	**nama**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
3	**nim**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
4	**email**	varchar(250)	utf8mb4_general_ci		No	*None*			Change Drop More
5	**jurusan**	varchar(250)	utf8mb4_general_ci		No	*None*			Change Drop More
6	**created_at**	timestamp			Yes	*NULL*			Change Drop More
7	**updated_at**	timestamp			Yes	*NULL*			Change Drop More

</td>
</tr>
</table>

b. Menampilkan Data dari Database

Langkah	Keterangan

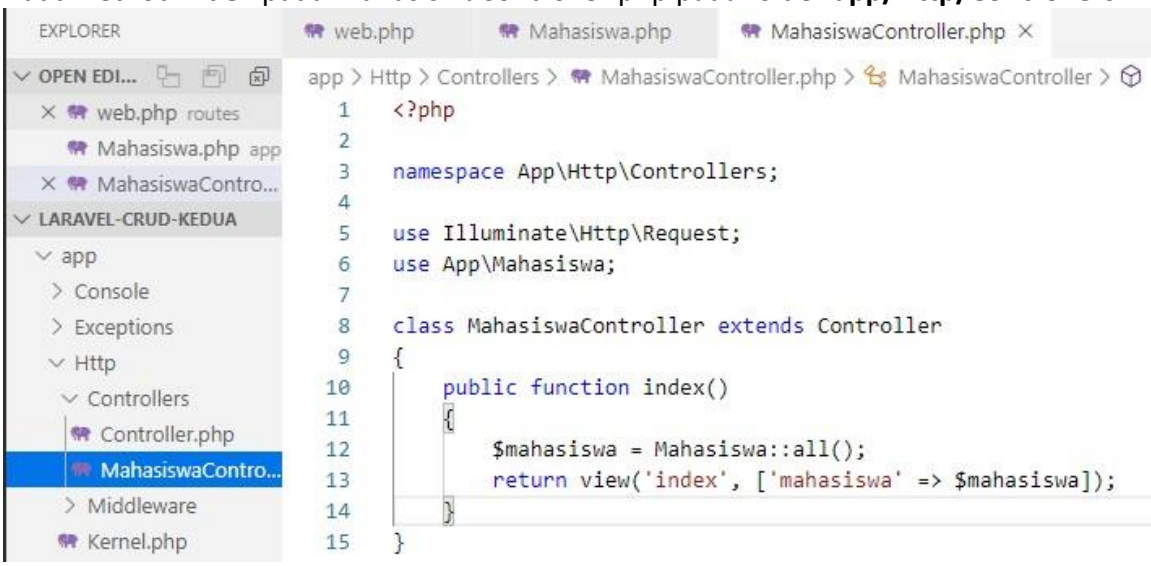
1	Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan.
	Pertama, buatlah *route* pada **routes/web.php** sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.
	`21 Route::get('/','MahasiswaController@index');`
	`22 Route::get('/mahasiswa','MahasiswaController@index');`
	`22 Route::get('/', 'MahasiswaController@index');`
	`23 Route::get('/mahasiswa', 'MahasiswaController@index');`

2	Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan
	`cd laravel-crud-kedua`
	`php artisan make:model Mahasiswa`
	`C:\xampp\htdocs>cd laravel-crud-kedua`
	`C:\xampp\htdocs\laravel-crud-kedua>php artisan make:model Mahasiswa`
	`Model created successfully.`
	`C:\xampp\htdocs\laravel-crud-kedua>`

3	Ubah model **Mahasiswa.php** pada folder **App** menjadi seperti berikut.
	EXPLORER / web.php / Mahasiswa.php
	OPEN EDITORS
	web.php routes
	Mahasiswa.php app
	LARAVEL-CRUD-KEDUA
	app
	Console
	Exceptions
	Http
	Providers
	app > Mahasiswa.php > Mahasiswa
	`1 <?php`
	`2`
	`3 namespace App;`
	`4`
	`5 use Illuminate\Database\Eloquent\Model;`
	`6`
	`7 class Mahasiswa extends Model`
	`8 {`
	`9 protected $table = "mahasiswa";`
	Mahasiswa.php ●
	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 protected $fillable = ['nama', 'nim', 'email', 'jurusan'];`
	`11 }`
	`12`

	Keterangan: - Model Mahasiswa akan menangani tabel mahasiswa

4	Buat controller baru yaitu MahasiswaController menggunakan php artisan
	`php artisan make:controller MahasiswaController`
	
	`C:\xampp\htdocs\laravel-crud-kedua>php artisan make:controller MahasiswaController`
	`Controller created successfully.`

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

app > Http > Controllers > MahasiswaController.php > PHP Intelephense > MahasiswaController > edit

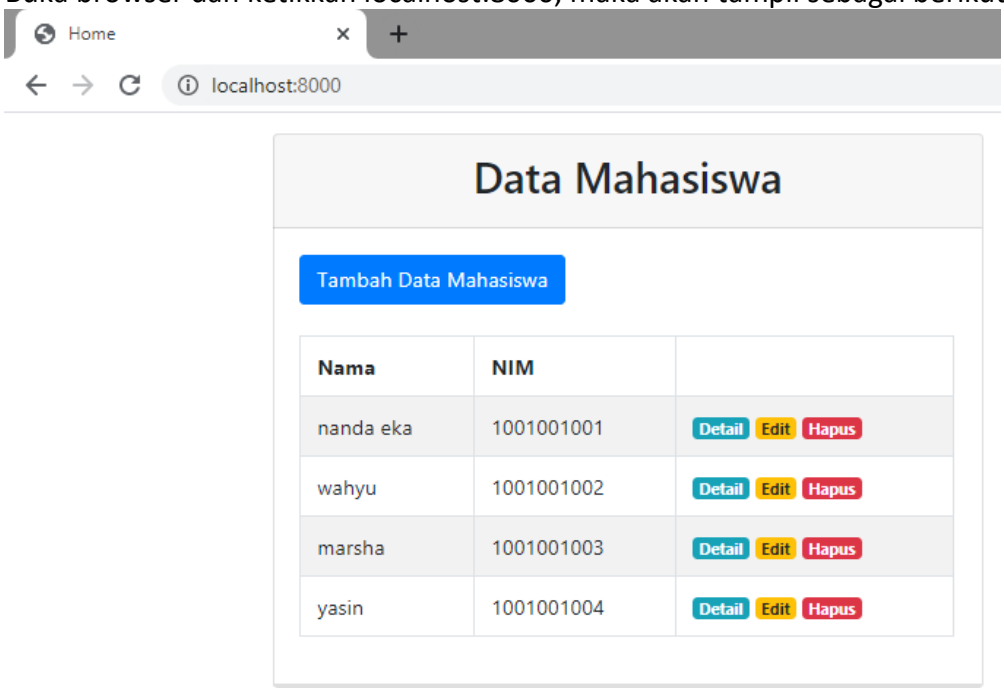
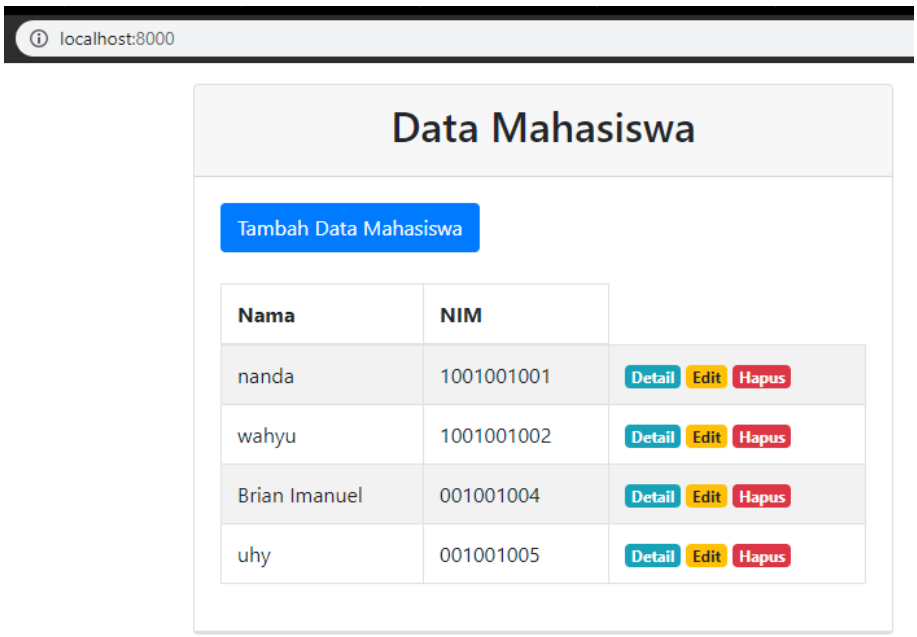
```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Mahasiswa;

class MahasiswaController extends Controller
{
    public function index()
    {
        $mahasiswa = Mahasiswa::all();
        return view('index', ['mahasiswa' => $mahasiswa]);
    }
```

Keterangan:

	- 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	Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud **`php artisan serve`** Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut  **HASIL :** 

c. Memasukkan Data (Create) ke Database

Langkah	Keterangan

1	Buatlah *route* baru pada **routes/web.php** dengan nama **/mahasiswa/tambah** yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.
	`22`
	`23 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');`
	`17`
	`18 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');`
	`19`

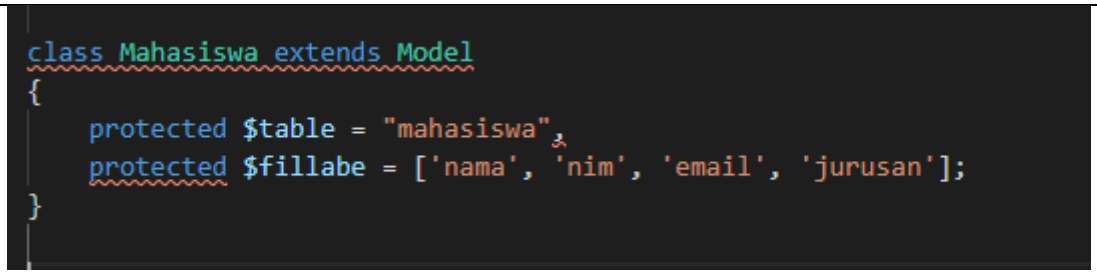
2	Buat method **tambah** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menampilkan view tambah.
	`16 public function tambah()`
	`17 {`
	`18 return view('tambah');`
	`19 }`
	`public function tambah()`
	`{`
	`// memanggil view tambah`
	`return view('tambah');`
	`}`

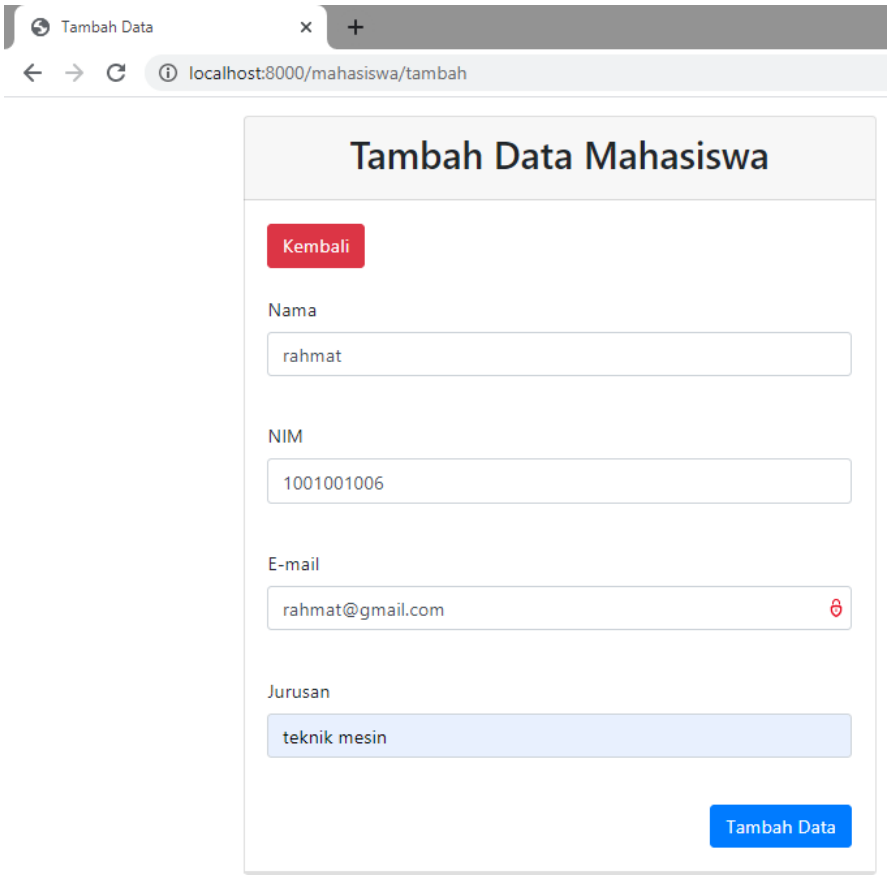
3	Kemudian buatlah view **tambah.blade.php** yang berisi form untuk memasukkan data baru pada folder **resources/views.**

	Kita lakukan copy paste dari tambah.blade.php di Bagian 1
4	Ketika tombol simpan ditekan, akan dipanggil routes **/mahasiswa/simpan**. Oleh karena itu, kita buat terlebih dahulu route tersebut. 24 25 `Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');` 20 `Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');` 21 Keterangan: - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php

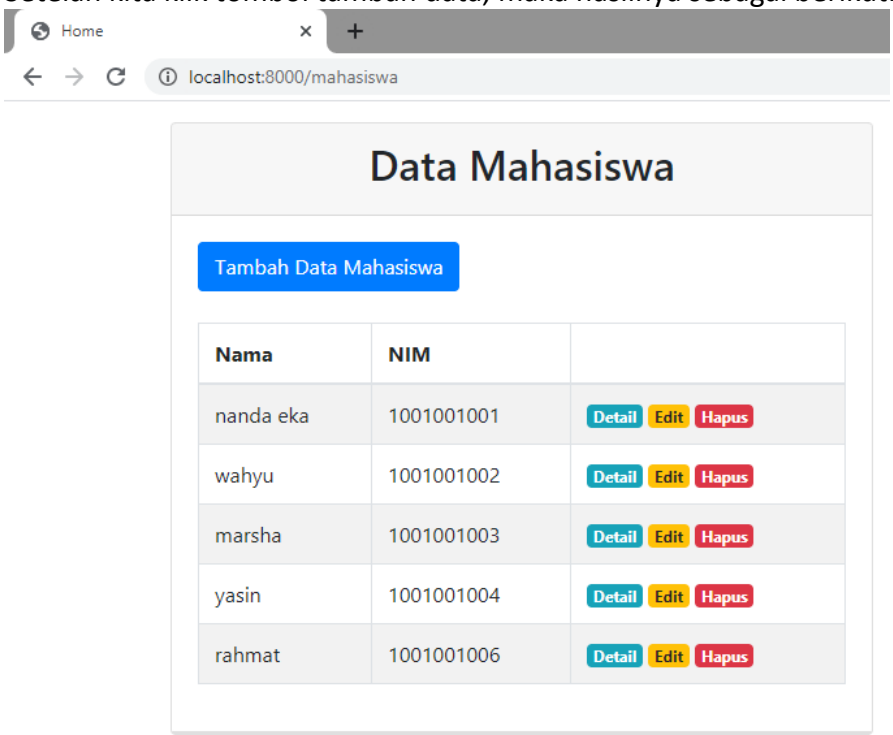
5	Buat method **simpan** pada **MahasiswaController** untuk menyimpan data ke database.
	21 `public function simpan(Request $request)`
	22 `{`
	23 `Mahasiswa::create([`
	24 `'nama' => $request->namamhs,`
	25 `'nim' => $request->nimmhs,`
	26 `'email' => $request->emailmhs,`
	27 `'jurusan' => $request->jurusanmhs`
	28 `]);`
	29
	30 `return redirect('/mahasiswa');`
	31 `}`
	`public function simpan(Request $request)`
	`{`
	`Mahasiswa::Create([`
	`'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 23-28 merupakan fitur eloquent menggunakan fungsi create() untuk insert data ke tabel mahasiswa

6	Karena kita menggunakan fungsi create pada Controller,maka butuh ditambahkan code pada Line 10 yang disebut Mass Assignment pada model Mahasiswa.php. Mass Assignment digunakan untuk memfilter kolom mana yang boleh dan tidak boleh diinput. 7 `class Mahasiswa extends Model` 8 `{` 9 `protected $table = "mahasiswa";` 10 `protected $fillable = ['nama','nim','email','jurusan'];` 11 `}`

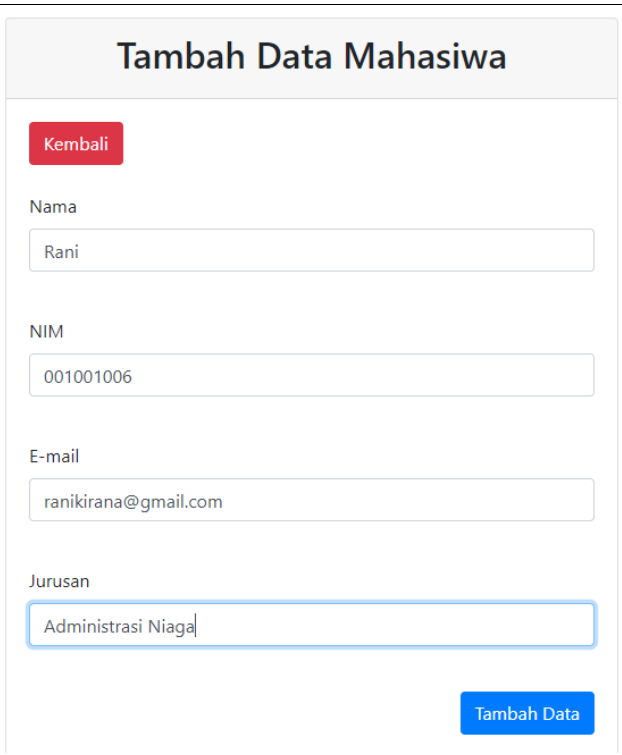
	
7	Kembali coba jalankan localhost:8000 dan pilih tombol 'Tambah Data Mahasiswa', maka akan ditampilkan halaman tambah yang berisi form untuk memasukkan data baru. Kita coba isikan data pada form tersebut.

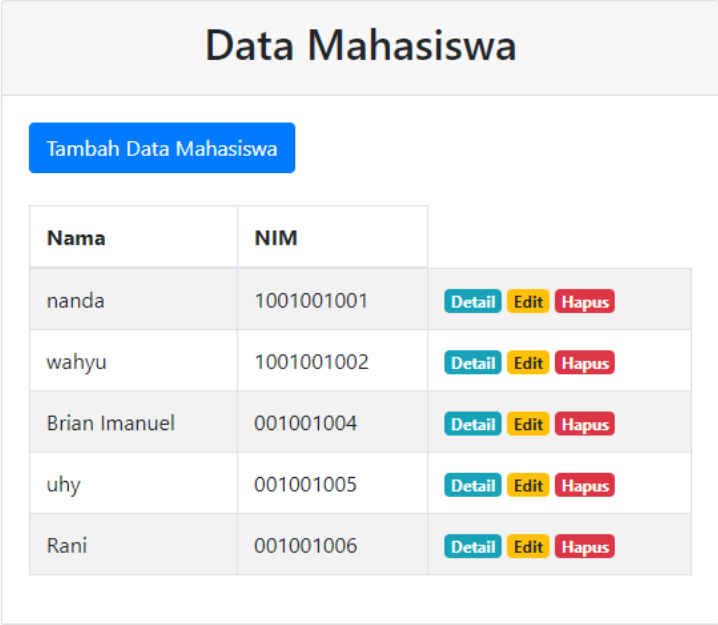


Setelah kita klik tombol tambah data, maka hasilnya sebagai berikut.



HASIL :





d. Melihat Detail Data dari Database

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

```
27
28  Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');
```

```
19  Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail');
```

3

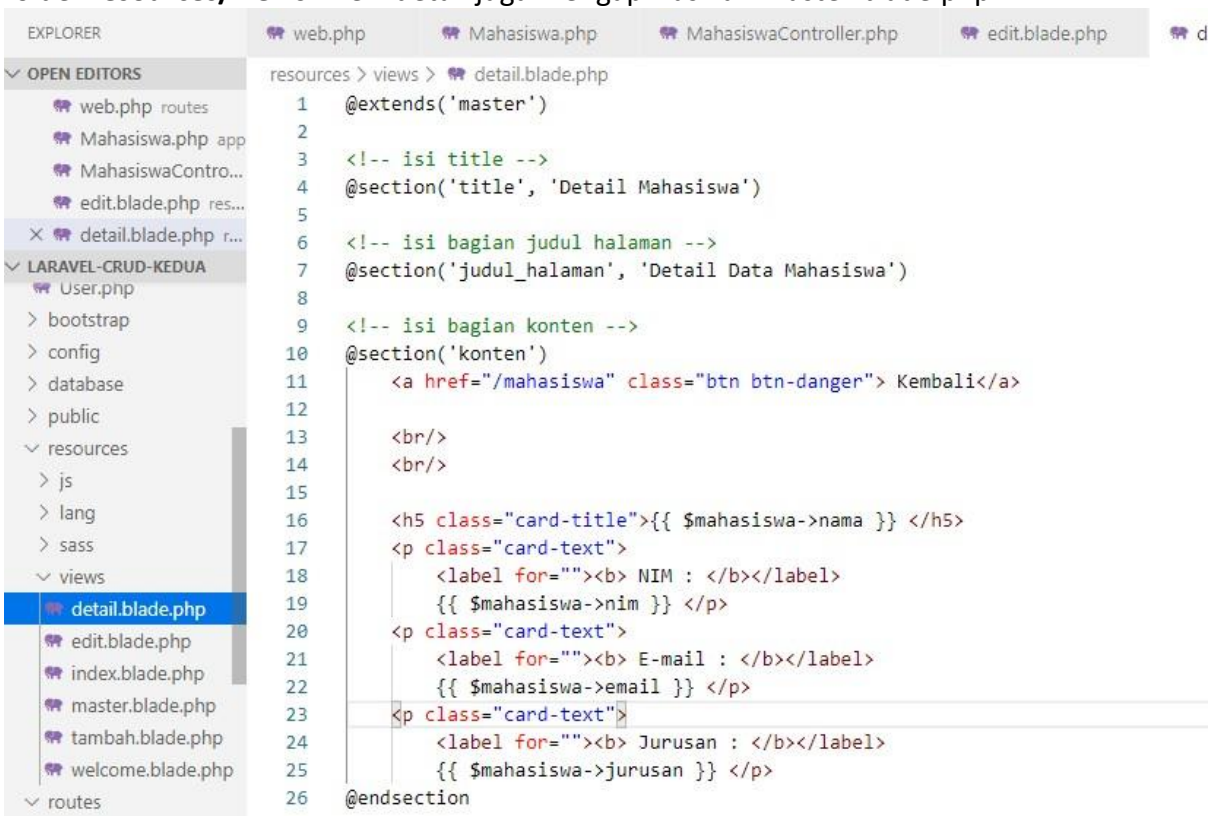
Buat method **detail** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menampilkan data mahasiswa pada view detail.blade.php.

```
33      public function detail($id)
34      {
35          $mahasiswa = Mahasiswa::find($id);
36          return view('detail', ['mahasiswa' => $mahasiswa ]);
37      }
```

```
37      public function detail($id)
38      {
39          $mahasiswa = Mahasiswa::find($id);
40          return view('detail', ['mahasiswa' => $mahasiswa]);
41      }
```

4

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



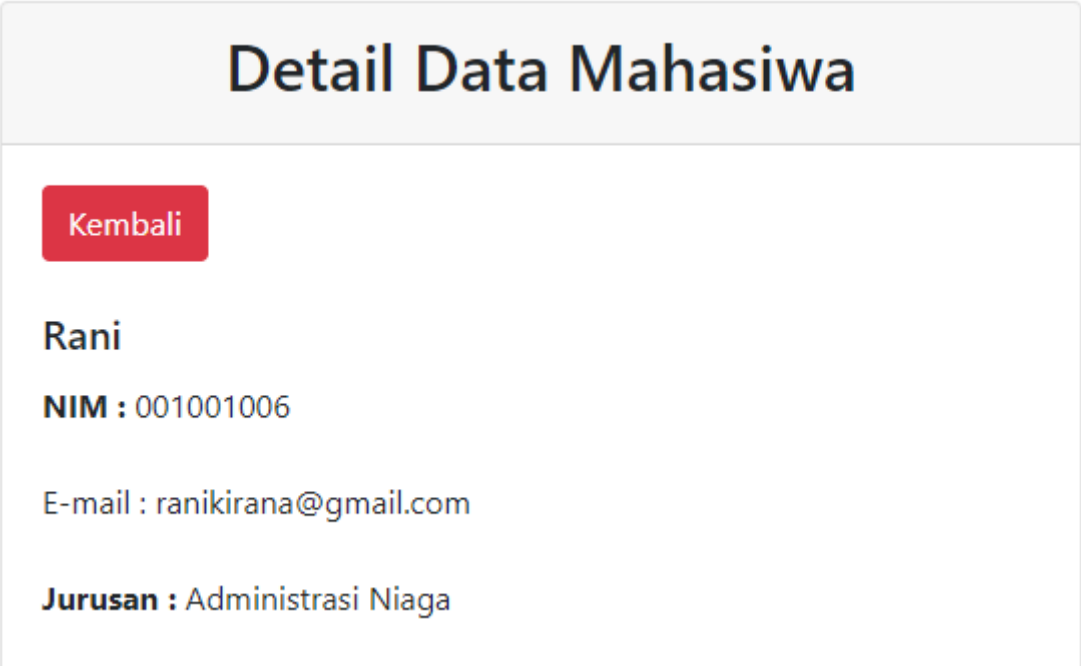
resources > views > detail.blade.php > ...

```
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=""> E-mail : </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
```

5

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

HASIL :



e. Mengubah Data (Update) dari Database

Langkah	Keterangan
1	Buatlah *route* baru pada **routes/web.php** dengan nama **/mahasiswa/edit** yang akan menjalankan fungsi edit pada MahasiswaController 29 30 `Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');` 20 `Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');`
2	Buat method **edit** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menampilkan view edit.

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

```
43    public function edit($id)
44    {
45        $mahasiswa = Mahasiswa::find($id);
46        return view('edit', ['mahasiswa' => $mahasiswa]);
47    }
```

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.

```
resources > views > edit.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      <br/>
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         <br/></div>
18         <div class="form-group">
19             <label for="nimmhs">NIM</label>
20             <input type="number" class="form-control" required="required" name="nimmhs" value="{{ $mahasiswa->nim }}">
21         <br/></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         <br/></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         <br/></div>
30         <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button>
31     </form>
32 @endsection
```

```
resources > views > edit.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 Mahasiwa')
8
9  <!-- isi bagian konten -->
10 @section('konten')
11     <a href="/" class="btn btn-danger">Kembali</a>
12     <br/>
13     <br/>
14     @foreach($mahasiswa as $mhs)
15     <form action="/mahasiswa/update" method="post">
16         {{ csrf_field() }}
17         <input type="hidden" name="id" value="{{ $mhs->id }}"> <br/>
18         <div class="form-group">
19             <label for="namamhs"> Nama </label>
20             <input type="text" class="form-control" required="required" name="namamhs" value="{{ $mhs->nama }}"> <br/>
21         </div>
22         <div class="form-group">
23             <label for="nimmhs"> NIM </label>
24             <input type="number" class="form-control" required="required" name="nimmhs" value="{{ $mhs->nim }}"> <br/>
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="{{ $mhs->email }}"> <br/>
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="{{ $mhs->jurusan }}"> <br/>
33         </div>
34         <button type="submit" name="edit" class="btn btn-primary float-right">Simpan Data</button>
35             </form>
36             @endforeach
37         @endsection
```

<table>
<tr><td></td><td>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</td></tr>
<tr><td>4</td><td>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/update/{id}. Oleh karena itu, kita buat terlebih dahulu route tersebut.
31
32 Route::post('/mahasiswa/update/{id}', 'MahasiswaController@update');
21 Route::post('/mahasiswa/update', 'MahasiswaController@update');
Keterangan:
- Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php</td></tr>
<tr><td>5</td><td>Buat method update pada MahasiswaController untuk menyimpan data yang diubah ke database.</td></tr>
</table>

```
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    }
```

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

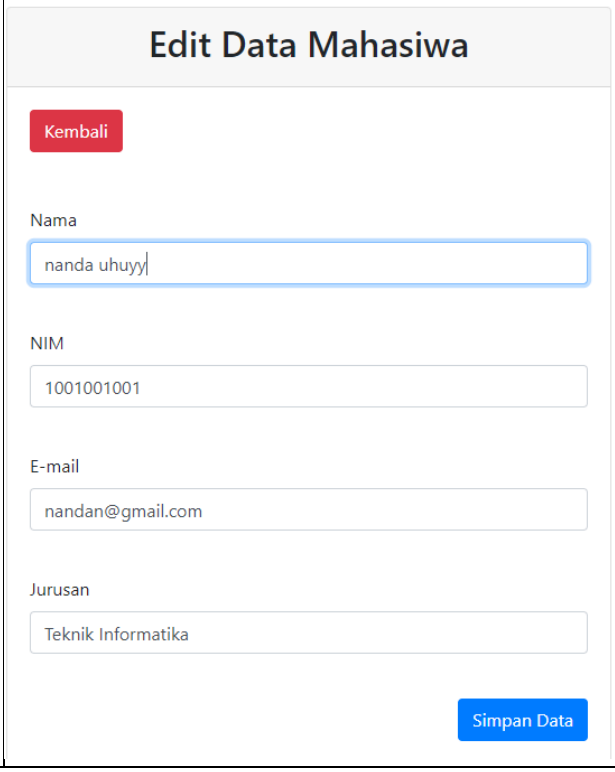
Keterangan:

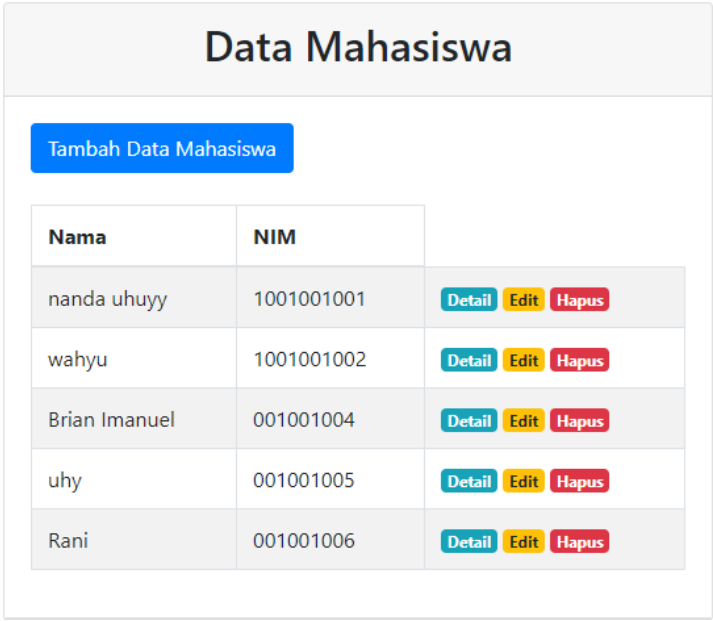
- Variabel $request untuk menerima data yang akan ditambahkan ke database
- Line 48-52 merupakan fungsi eloquent untuk update data ke tabel mahasiswa

6

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

HASIL :





f. Menghapus Data (Delete) dari Database

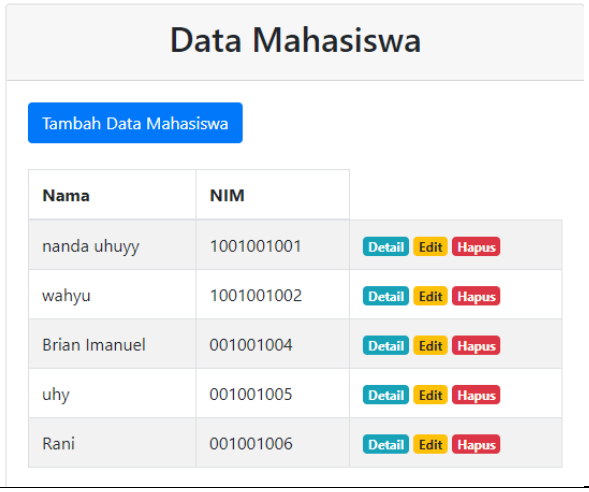
Langkah	Keterangan

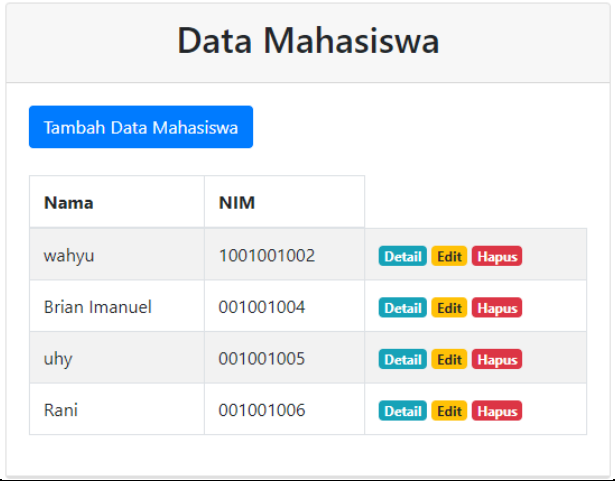
1	Buatlah *route* baru pada **routes/web.php** dengan nama **/mahasiswa/hapus** yang akan menjalankan fungsi hapus pada MahasiswaController
	`33`
	`34 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');`
	`22 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');`

2	Buat method **hapus** pada **MahasiswaController.php** di folder **app/Http/Controllers** yang akan menjalankan fungsi hapus.
	`56 public function hapus($id)`
	`57 {`
	`58 $mahasiswa = Mahasiswa::find($id);`
	`59 $mahasiswa->delete();`
	`60 return redirect('/mahasiswa');`
	`61 }`
	`60 public function hapus($id)`
	`61 {`
	`62 $mahasiswa = Mahasiswa::find($id);`
	`63 $mahasiswa->delete();`
	`64 return redirect('/');`
	`65 }`
	Keterangan :
	- Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih

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

HASIL :





-- Selamat Mengerjakan --