

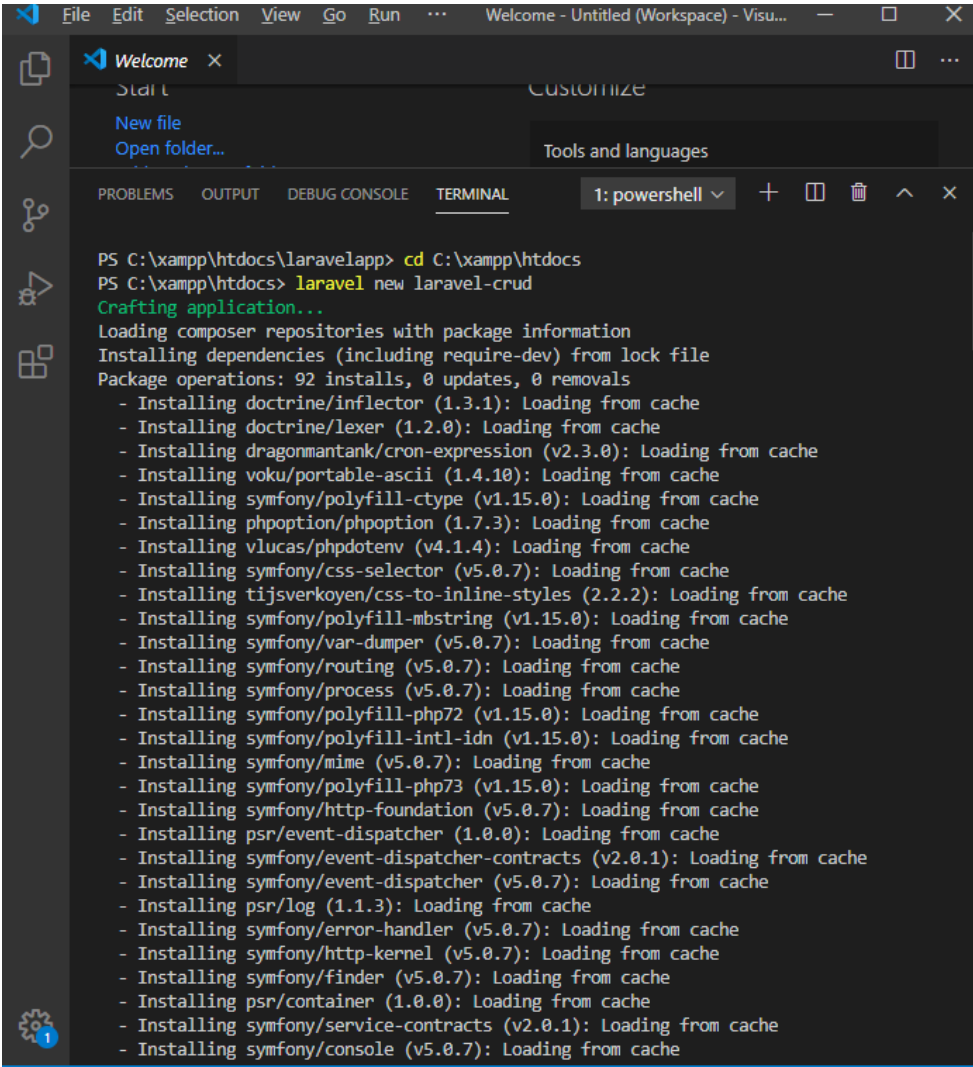


Nama : Agsal Fairrohmad A.P
No : 04/TI2A

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> 

2

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

```

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:KOQ+JFm1Mm7s4PpYHT8rZh0Ygyf
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
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=127.0.0.1
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
25
26 MAIL_MAILER=smtp
27 MAIL_HOST=smtp.mailtrap.io
28 MAIL_PORT=2525
29 MAIL_USERNAME=null
30 MAIL_PASSWORD=null
31 MAIL_ENCRYPTION=null
32 MAIL_FROM_ADDRESS=null
33 MAIL_FROM_NAME="${APP_NAME}"
34
35 AWS_ACCESS_KEY_ID=
36 AWS_SECRET_ACCESS_KEY=
  
```

Keterangan: Nama database yang akan digunakan adalah latihan_laravel dengan username root.

3

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

The screenshot shows the phpMyAdmin interface with the following details:

- Database:** latihan_laravel
- Table:** mahasiswa
- Columns:**

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar
1	id	int(11)			Tidak	Tidak ada	
2	nama	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada	
3	nim	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada	
4	email	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada	
5	jurusan	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada	
- Primary Key:** id
- Indexes:** PRIMARY (BTREE, Ya, Tidak, id, 0, A)
- Partitions:** No partitioning defined!

4

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

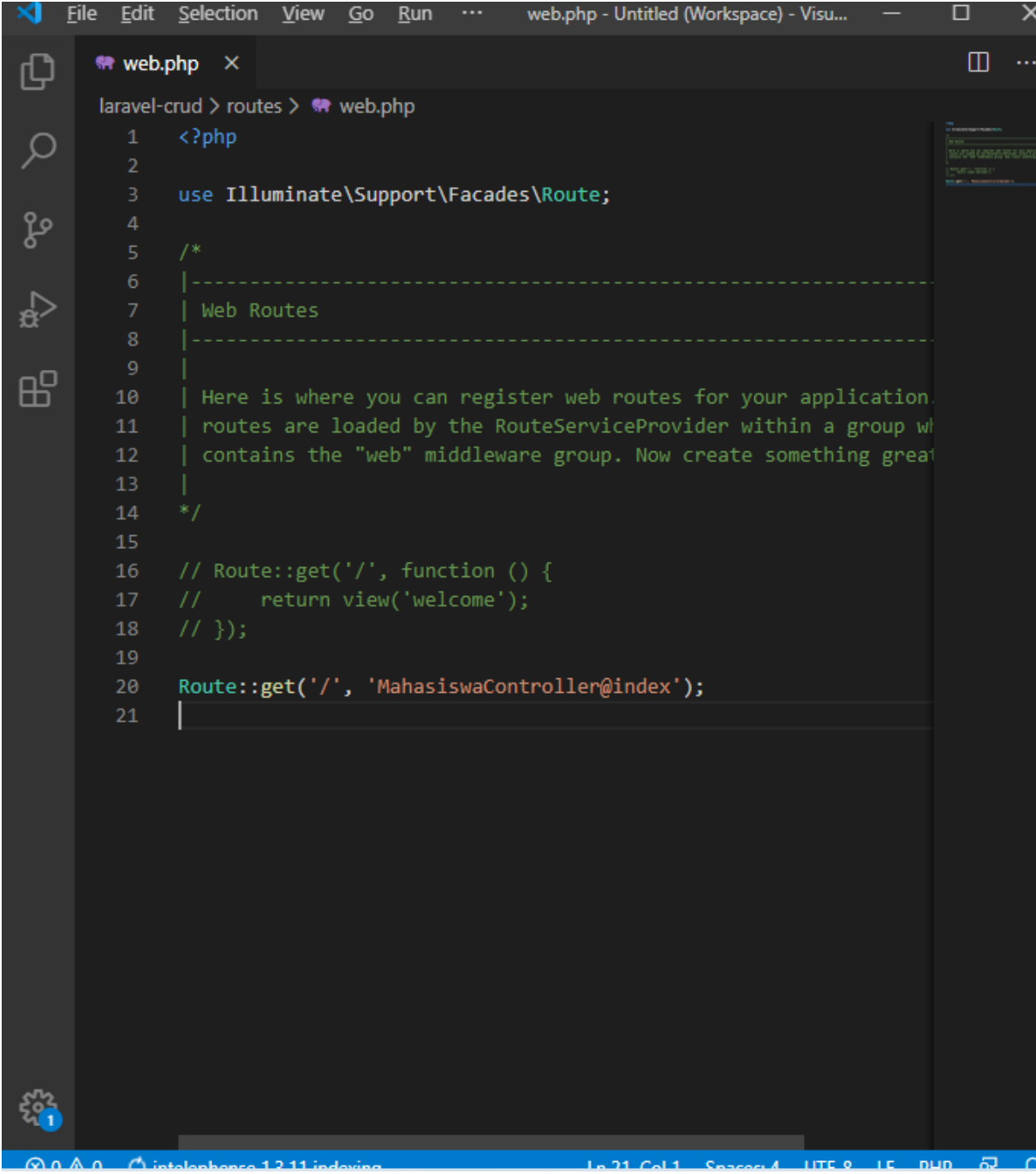
The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure with the following items: Baru, biodata, dbperpus, ecommerce, finland, information_schema, jne_database, kuliah_ci, latihan, latihan_ci, latihan_laravel, mahasiswa, mysql, pajak, pemesanan, performance_schema, pesanhotel, phpmyadmin, praktikumdb, rest_server, and test. The 'mahasiswa' table is selected under the 'latihan' database.

The main panel shows the 'mahasiswa' table with the following data:

id	nama	nim	email	jurusan
1	Agsal	1001001001	agsal@gmail.com	teknik informatika
2	wahyu	1001001002	wahyu@gmail.com	teknik elektro

The interface also includes a SQL query editor with the query `SELECT * FROM 'mahasiswa'` and a console at the bottom.

b. Menampilkan Data dari Database

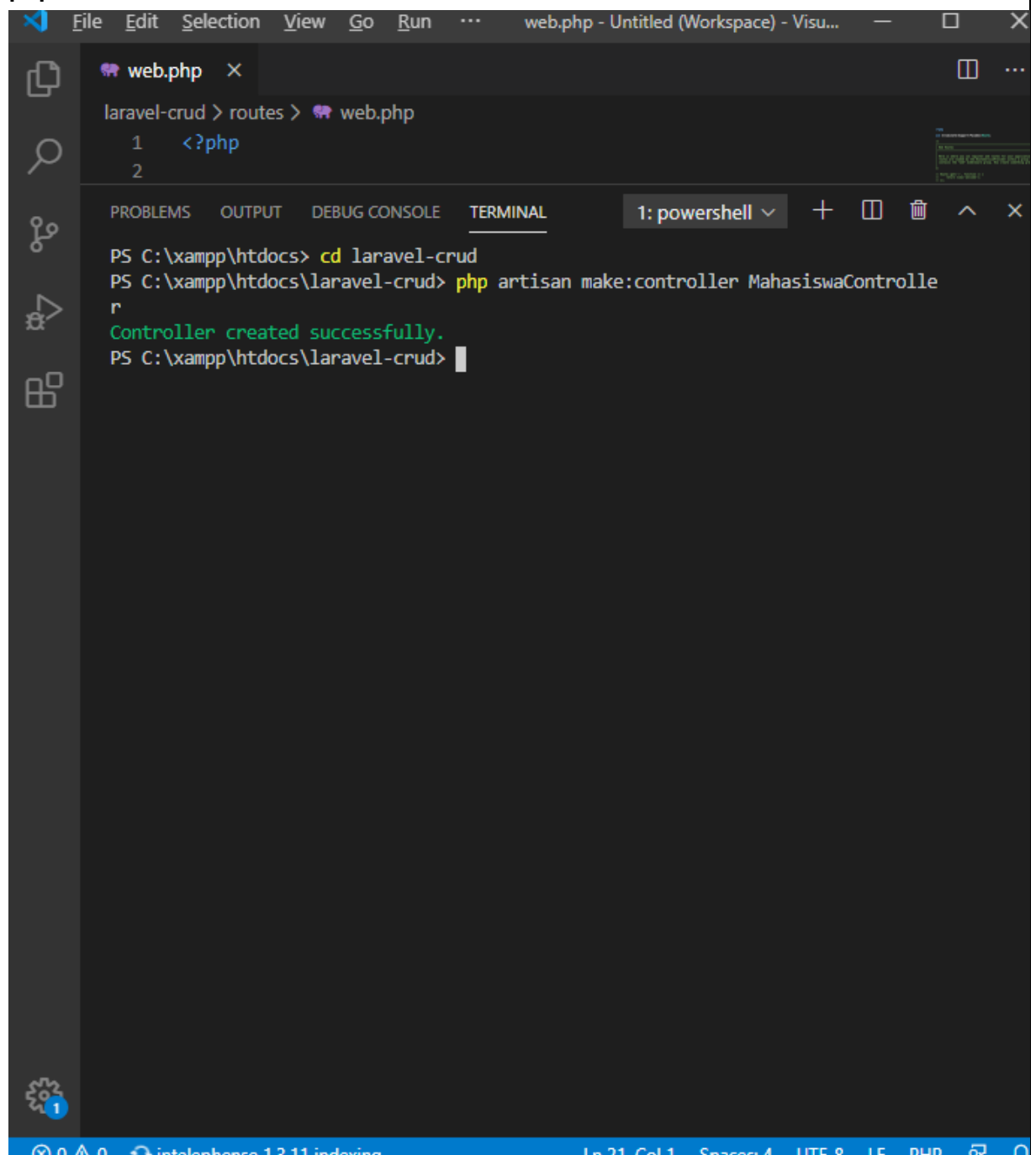
Langkah	Keterangan
1	<p>Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan.</p> <p>Pertama, buatlah <i>route</i> pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.</p>  <p>Keterangan:</p> <ul style="list-style-type: none"> - 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

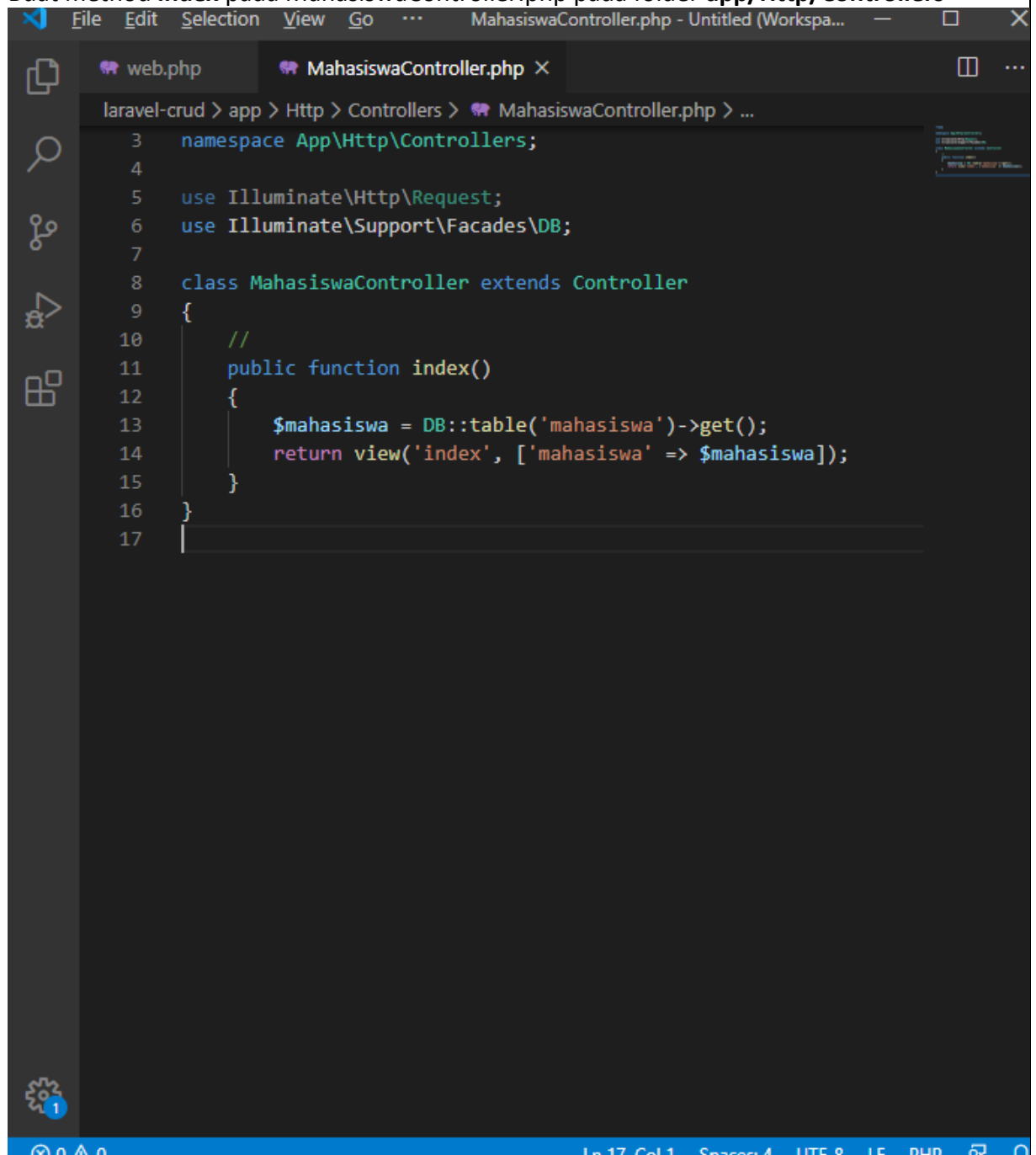


The screenshot shows the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, and a dropdown menu. The file explorer on the left shows a project named 'laravel-crud' with a 'routes' folder containing a 'web.php' file. The editor window displays the 'web.php' file with two lines of code: `<?php` and a blank line. The terminal window at the bottom shows the following commands and output:

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

The status bar at the bottom indicates the current file is 'web.php' and the language is 'PHP'.

3

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

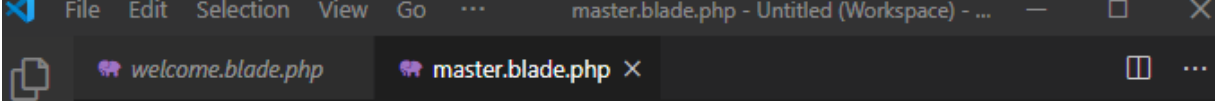
```
File Edit Selection View Go ... MahasiswaController.php - Untitled (Workspa...
web.php MahasiswaController.php X
laravel-crud > app > Http > Controllers > MahasiswaController.php > ...
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class MahasiswaController extends Controller
9 {
10     //
11     public function index()
12     {
13         $mahasiswa = DB::table('mahasiswa')->get();
14         return view('index', ['mahasiswa' => $mahasiswa]);
15     }
16 }
17
```

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**.



```
<html>

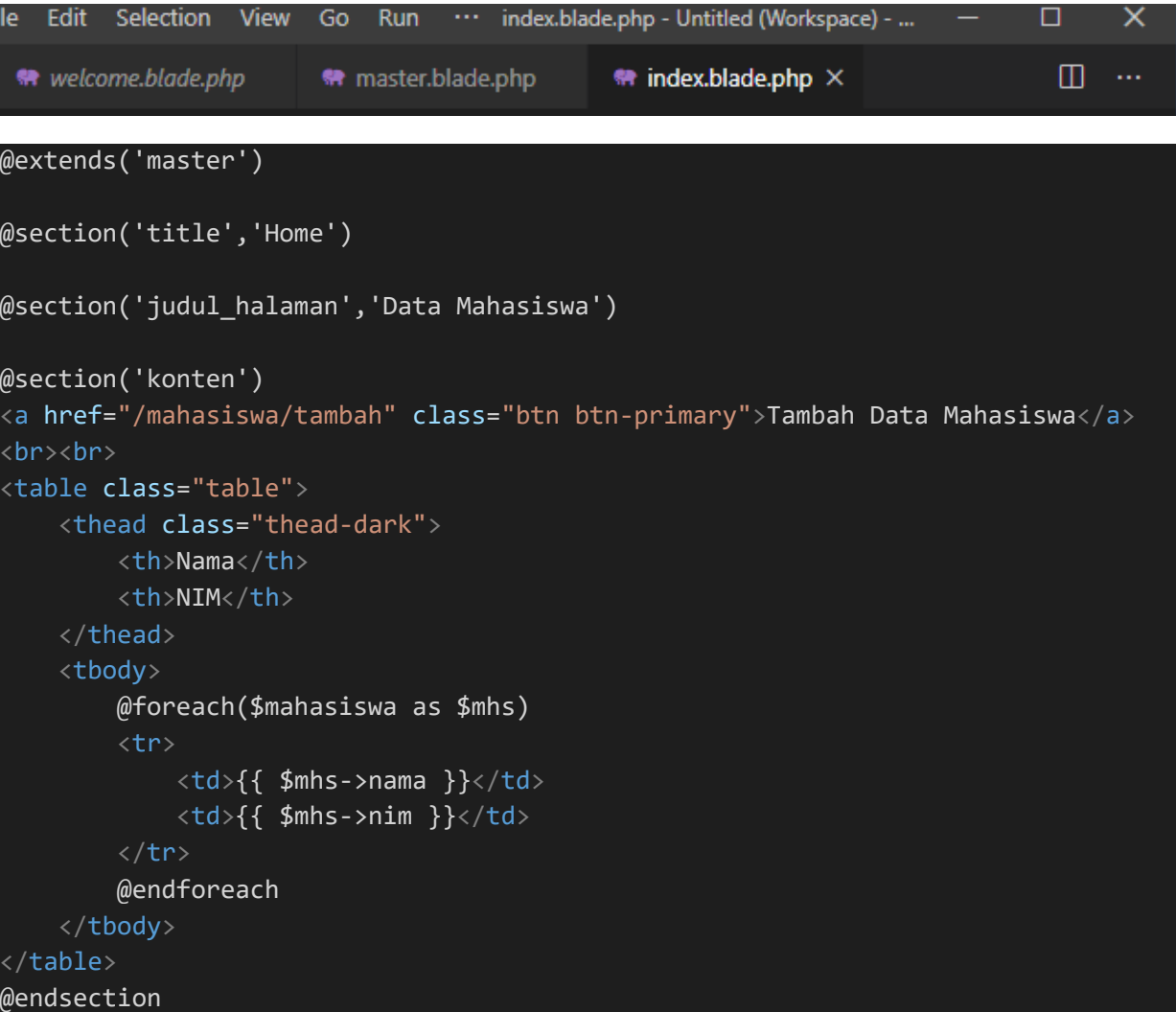
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
    <title> @yield('title') </title>
</head>

<body>
    <div class="container">
        <div class="row mt-3">
            <div class="col-md-6">
                <div class="card-header text-center">
                    <!-- Judul Halaman -->
                    <h2> @yield('judul_halaman') </h2>
                </div>
                <div class="card-body">
                    <!-- Konten Halaman -->
                    @yield('konten')
                </div>
            </div>
        </div>
    </div>
</body>

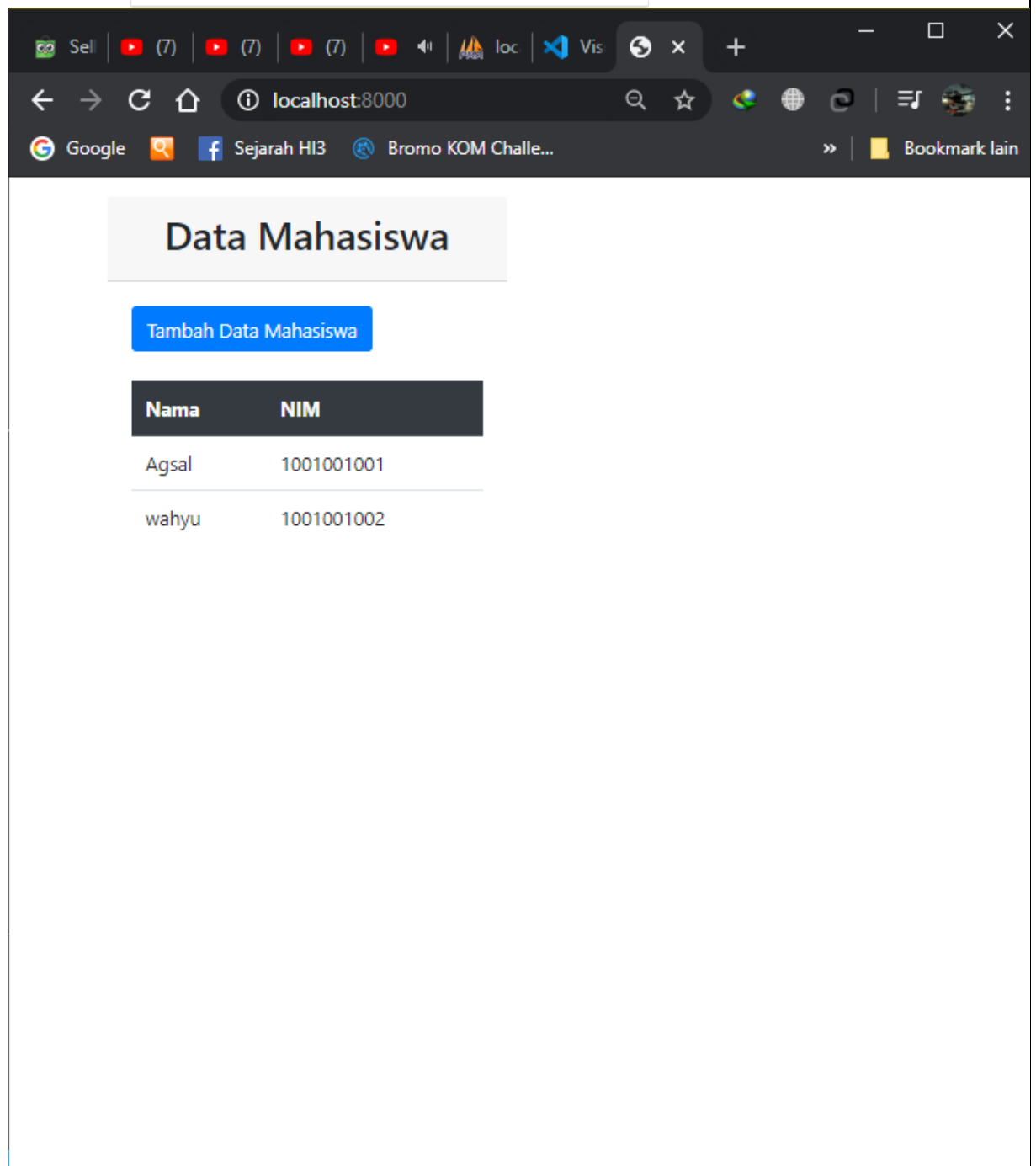
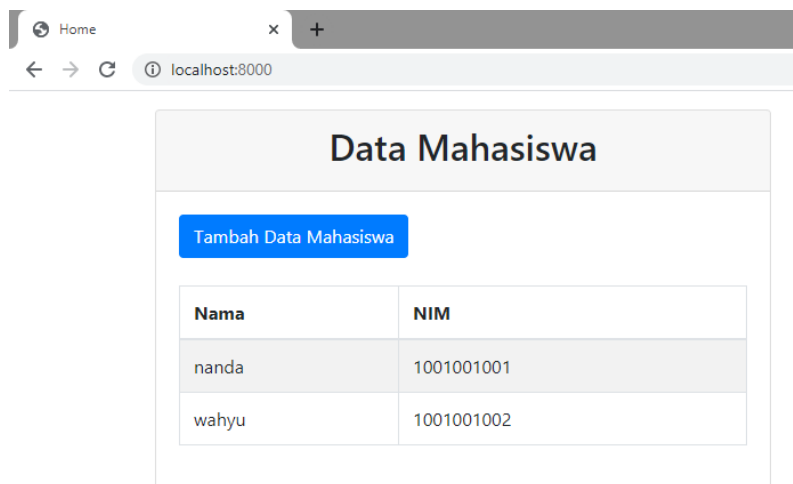
</html>
```

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	<p>Sekarang buat file index.blade.php yang menerapkan <i>template</i> master.blade.php dan akan digunakan untuk menampilkan data.</p>  <pre> @extends('master') @section('title','Home') @section('judul_halaman','Data Mahasiswa') @section('konten') Tambah Data Mahasiswa

 <table class="table"> <thead class="thead-dark"> <th>Nama</th> <th>NIM</th> </thead> <tbody> @foreach(\$mahasiswa as \$mhs) <tr> <td>{{ \$mhs->nama }}</td> <td>{{ \$mhs->nim }}</td> </tr> @endforeach </tbody> </table> @endsection </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan blade lain yaitu master.blade.php - Line 4 : @section('title', 'Home') berarti bahwa file index mengisi @yield('title') pada master dengan 'Home' - Line 7 : @section('judul_halaman', 'Data Mahasiswa') berarti bahwa file index mengisi @yield('judul_halaman') pada master dengan 'Home' - Line 10-30 : mengisi yield(@konten), karena terdapat banyak baris pada diawali dengan @section('konten') dan diakhiri dengan @endsection - Line 24-25 menampilkan data mahasiswa dengan kolom nama dan nim
6	<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');</pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre> public function tambah() { # memanggil file tambah return view('tambah'); }</pre>
3	<p>Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. View tambah juga mengaplikasikan master.blade.php.</p> <pre> @extends('master') @section('title','Tambah Data') @section('judul_halaman','Tambah Data Mahasiswa') @section('konten') Kembali

 <form action="/mahasiswa/simpan" method="POST"> {{ csrf_field() }} <div class="form-group"> <label>Nama</label> <input type="text" class="form- control" required="required" name="namamhs">
 </div> <div class="form-group"> <label>NIM</label> <input type="number" class="form- control" required="required" name="nimmmhs">
 </div> <div class="form-group"> <label>Email</label> <input type="email" class="form- control" required="required" name="emailmhs">
 </div> <div class="form-group"> <label>Jurusan</label> <input type="text" class="form-</pre>

```

control" required="required" name="jurusanmhs"><br>
</div>
<input type="submit" class="btn btn-primary float-
right" value="Simpan Data">
</form>
@endsection

```

```

File Edit Selection View Go ... tambah.blade.php - Untitled (Workspace) - ...
EXPLORER
OPEN EDITORS
web.php laravel-crud *...
MahasiswaController....
X tambah.blade.php lar...
UNTITLED (WORKSPACE)
> Middleware
Kernel.php
> Providers
User.php
> bootstrap
> config
> database
> public
resources
> js
> lang
> sass
views
index.blade.php
master.blade.php
tambah.blade.php
welcome.blade.php
> routes
> storage
> tests
> vendor
.editorconfig
.env
.env.example
OUTLINE
NPM SCRIPTS

laravel-crud > resources > views > tambah.blade.php > ...
8 <a href="/" class="btn btn-danger">Kembali</a>
9 <br>
10 <br>
11 <form action="/mahasiswa/simpan" method="POST">
12     {{ csrf_field() }}
13     <div class="form-group">
14         <label>Nama</label>
15         <input type="text" class="form-control">
16     </div>
17     <div class="form-group">
18         <label>NIM</label>
19         <input type="number" class="form-control">
20     </div>
21     <div class="form-group">
22         <label>Email</label>
23         <input type="email" class="form-control">
24     </div>
25     <div class="form-group">
26         <label>Jurusan</label>
27         <input type="text" class="form-control">
28     </div>
29     <input type="submit" class="btn btn-primary" value="Simpan Data">
30 </form>
31 @endsection

```

```

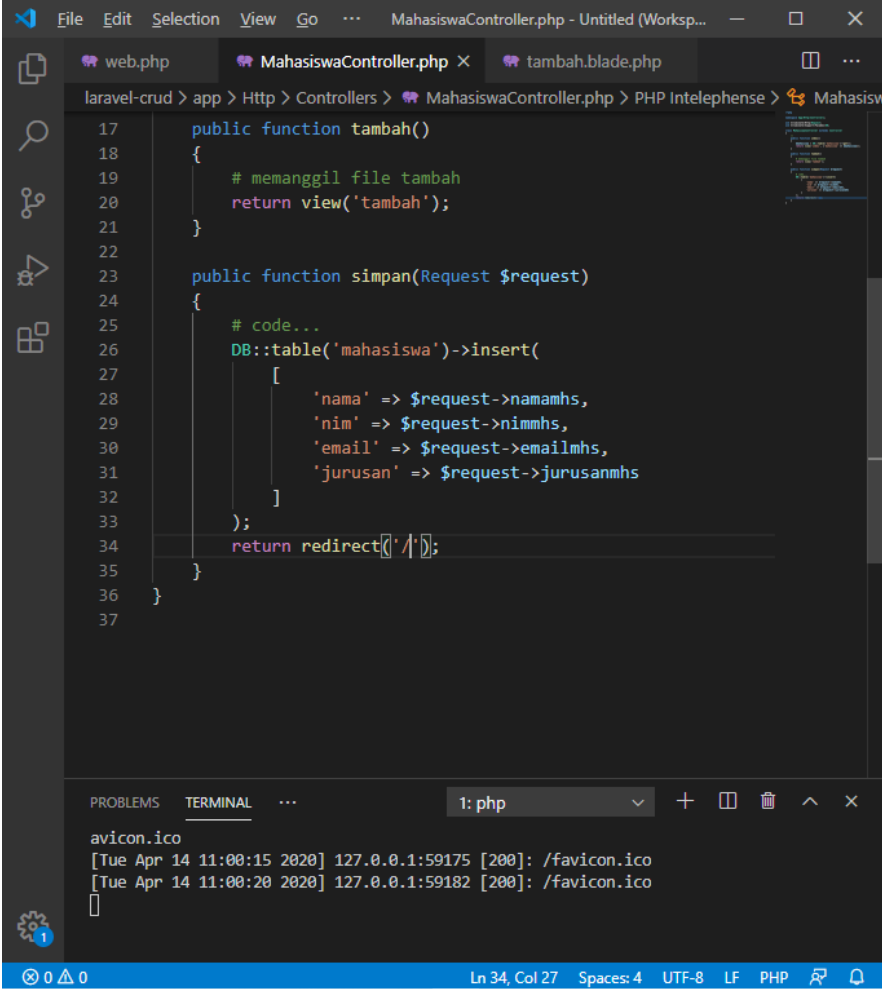
TERMINAL
1: powershell
000
[Tue Apr 14 10:28:31 2020] 127.0.0.1:58764 [200]: /fav
icon.ico
[Tue Apr 14 10:28:52 2020] 127.0.0.1:58771 [200]: /fav
icon.ico
PS C:\xampp\htdocs\laravel-crud>

```

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	<p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php
5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p>  <pre> public function simpan(Request \$request) { # code... DB::table('mahasiswa')->insert(['nama' => \$request->namamhs, 'nim' => \$request->nimmhs, 'email' => \$request->emailmhs, 'jurusan' => \$request->jurusanmhs]); return redirect('/'); } </pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Variabel \$request untuk menerima data yang akan ditambahkan ke database <p>Line 28-33 merupakan query builder untuk insert data ke tabel mahasiswa</p>

6

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

The screenshot shows a web browser window with the address bar displaying 'localhost:8000/mahasiswa/tambah'. The page title is 'Tambah Data Mahasiswa'. The form contains the following fields and buttons:

- Kembali** (Red button)
- Nama**: Input field with value 'Sugoy'
- NIM**: Input field with value '1001001005'
- Email**: Input field with value 'sugoy@Yuhuu.com'
- Jurusan**: Input field with value 'Teknologi Informasi'
- Simpan Data** (Blue button)

At the bottom of the browser window, a status bar shows 'Menunggu localhost...'.

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

Tambah Data Mahasiswa

Nama	NIM
Agsal	1001001001
wahyu	1001001002
Arcgis	1001001003
Mirai	1001001004
Sugoy	1001001005

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> @extends('master') @section('title','Home') @section('judul_halaman','Data Mahasiswa') @section('konten') Tambah Data Mahasiswa

 <table class="table"> <thead class="thead-dark"> <th>Nama</th> <th>NIM</th> <th>Aksi</th> </thead> <tbody> @foreach(\$mahasiswa as \$mhs) <tr> <td>{{ \$mhs->nama }}</td> <td>{{ \$mhs->nim }}</td> <td> id }}" class="badge badge-info">Detail </td> </tr> @endforeach </tbody> </table> @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> Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); </pre> <pre> 25 26 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); </pre>
3	<p>Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.</p>

	<pre> public function detail(\$id) { \$mahasiswa = DB::table('mahasiswa')->where('id', \$id)->get(); return view('detail', ['mahasiswa' => \$mahasiswa]); } </pre>
4	<p>Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php.</p> <pre> @extends('master') @section('title','Detail Mahasiswa') @section('judul_halaman','Detail Data Mahasiswa') @section('konten') Kembali

 @foreach(\$mahasiswa as \$mhs) <h5 class="card-title">{{ \$mhs->nama }}</h5> <p class="card-text"> <label for="">NIM : </label> {{ \$mhs->nim }} </p> <p class="card-text"> <label for="">Email : </label> {{ \$mhs->email }} </p> <p class="card-text"> <label for="">Jurusan : </label> {{ \$mhs->jurusan }} </p> @endforeach @endsection </pre>

```

1  @extends('master')
2
3  @section('title','Detail Mahasiswa')
4
5  @section('judul_halaman','Detail Data Mahasiswa')
6
7  @section('konten')
8      <a href="/" class="btn btn-danger">Kembali</a>
9
10     <br>
11     <br>
12
13     @foreach($mahasiswa as $mhs)
14         <h5 class="card-title">{{ $mhs->nama }}</h5>
15         <p class="card-text">
16             <label for=""><b>NIM : </b></label>
17             {{ $mhs->nim }}
18         </p>
19         <p class="card-text">
20             <label for=""><b>Email : </b></label>
21             {{ $mhs->email }}
22         </p>
23         <p class="card-text">
24             <label for=""><b>Jurusan : </b></label>
25             {{ $mhs->jurusan }}
26         </p>
27     @endforeach
28 @endsection

```

PROBLEMS TERMINAL ... 1: php

[Tue Apr 14 11:35:43 2020] 127.0.0.1:59376 [200]: /favicon.ico

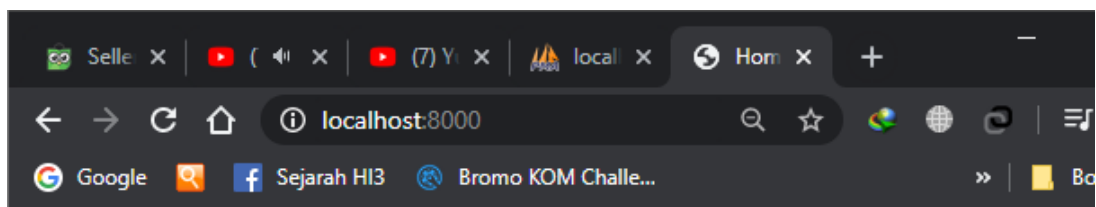
0 0 Tab Moves Focus Ln 26, Col 5 Spaces: 4 UTF-8 CRLF PHP

Keterangan:

- Line 13-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan

5

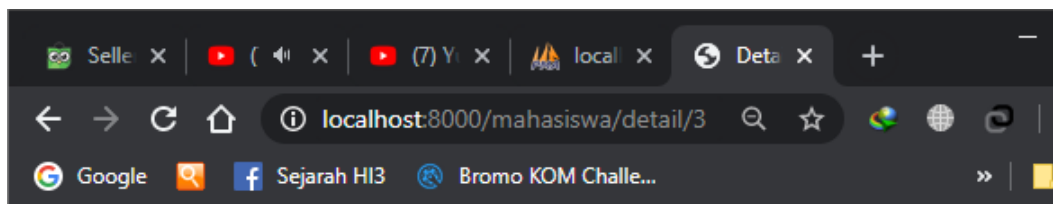
Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.



Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	Action
Agsal	1001001001	Detail
wahyu	1001001002	Detail
Arcgis	1001001003	Detail
Mirai	1001001004	Detail
Sugoy	1001001005	Detail



Detail Data Mahasiswa

Kembali

Arcgis

NIM : 1001001003

Email : Arcgis@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> <pre> @extends('master') @section('title','Home') @section('judul_halaman','Data Mahasiswa') @section('konten') Tambah Data Mahasiswa

 <table class="table"> <thead class="thead-dark"> <th>Nama</th> <th>NIM</th> <th>Action</th> </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> @endsection </pre>

```

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

```

PROBLEMS TERMINAL ... 1: php

```

[Tue Apr 14 11:42:00 2020] 127.0.0.1:58166 [200]: /favicon.ico
[Tue Apr 14 11:42:25 2020] 127.0.0.1:58174 [200]: /favicon.ico

```

0 0 Tab Moves Focus Ln 18, Col 13 Spaces: 4 UTF-8 CRLF PHP

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

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

3

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

```

20     return view('tambah');
21 }
22
23 public function simpan(Request $request)
24 {
25     # code...
26     DB::table('mahasiswa')->insert(
27         [
28             'nama' => $request->namamhs,
29             'nim' => $request->nimmhs,
30             'email' => $request->emailmhs,
31             'jurusan' => $request->jurusanmhs
32         ]
33     );
34     return redirect('/');
35 }
36 public function detail($id)
37 {
38     $mahasiswa = DB::table('mahasiswa')->where('id', $id)->get();
39     return view('detail', ['mahasiswa' => $mahasiswa]);
40 }
41 public function edit($id)
42 {
43     $mahasiswa = DB::table('mahasiswa')->where('id', $id)->get();
44     return view('edit', ['mahasiswa' => $mahasiswa]);
45 }
46
47

```

PROBLEMS TERMINAL ... 1: php

[Tue Apr 14 11:42:25 2020] 127.0.0.1:58174 [200]: /favicon.ico

0 0 Tab Moves Focus Ln 36, Col 32 Spaces: 4 UTF-8 LF PHP

```

public function edit($id)
{
    $mahasiswa = DB::table('mahasiswa')->where('id', $id)->get();
    return view('edit', ['mahasiswa' => $mahasiswa]);
}

```

Keterangan :

- Line 43 : 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.

```
@extends('master')

@section('title','Edit Data')

@section('judul_halaman','Edit Data Mahasiswa')

@section('konten')
<a href="/" class="btn btn-danger">Kembali</a>
<br>
<br>
<form action="/mahasiswa/update" method="POST">
    {{ csrf_field() }}
    @foreach($mahasiswa as $mhs)
        <input type="hidden" name="id" value="{{ $mhs->id }}">
        <div class="form-group">
            <label>Nama</label>
            <input type="text" class="form-control" value="{{ $mhs-
>nama }}" required="required" name="namamhs"><br>
        </div>
        <div class="form-group">
            <label>NIM</label>
            <input type="number" class="form-control" value="{{ $mhs-
>nim }}" required="required" name="nimmmhs"><br>
        </div>
        <div class="form-group">
            <label>Email</label>
            <input type="email" class="form-control" value="{{ $mhs-
>email }}" required="required" name="emailmhs"><br>
        </div>
        <div class="form-group">
            <label>Jurusan</label>
            <input type="text" class="form-control" value="{{ $mhs-
>jurusan }}" required="required" name="jurusanmhs"><br>
        </div>
        <input type="submit" class="btn btn-primary float-
right" value="Update Data">
    @endforeach
</form>
@endsection
```

```

6
7 @section('konten')
8 <a href="/" class="btn btn-danger">Kembali</a>
9 <br>
10 <br>
11 <form action="/mahasiswa/update" method="POST">
12     {{ csrf_field() }}
13     @foreach($mahasiswa as $mhs)
14         <input type="hidden" name="id" value="{{ $mhs->id }}">
15         <div class="form-group">
16             <label>Nama</label>
17             <input type="text" class="form-control" value="{{ $mhs->nama }}">
18         </div>
19         <div class="form-group">
20             <label>NIM</label>
21             <input type="number" class="form-control" value="{{ $mhs->nim }}">
22         </div>
23         <div class="form-group">
24             <label>Email</label>
25             <input type="email" class="form-control" value="{{ $mhs->email }}">
26         </div>
27         <div class="form-group">
28             <label>Jurusan</label>
29             <input type="text" class="form-control" value="{{ $mhs->jurusan }}">
30         </div>
31         <input type="submit" class="btn btn-primary float-right" value="Simpan">
32     @endforeach
33 </form>
34 @endsection

```

PROBLEMS TERMINAL ... 1: php

```

[Tue Apr 14 12:12:49 2020] 127.0.0.1:58305 [200]: /favicon.ico
[Tue Apr 14 12:13:03 2020] 127.0.0.1:58313 [200]: /favicon.ico

```

0 0 0 Tab Moves Focus Ln 27, Col 29 Spaces: 4 UTF-8 CRLF PHP

Keterangan:

- Line 14-31 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan
- Line 12 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.

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

29

```
30 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.

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

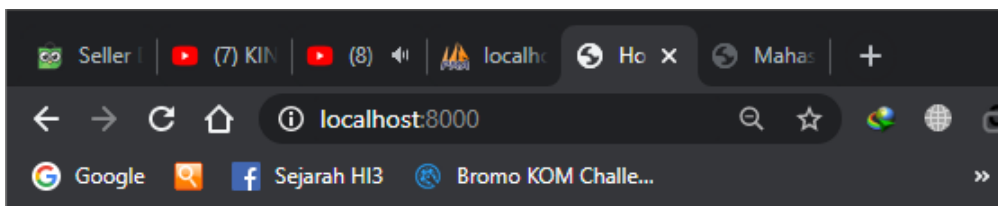
Keterangan:

- Variabel \$request untuk menerima data yang akan ditambahkan ke database
- Line 48-53 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.

Klik edit di data pertama, kita akan mengubah namanya.



Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	Action
Agsal	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
Arcgis	1001001003	Detail Edit
Mirai	1001001004	Detail Edit
Sugoy	1001001005	Detail Edit

Edit Data Mahasiswa

[Kembali](#)

Nama
Agsal F.A.P

NIM
1001001001

Email
agsal@gmail.com

Jurusan
teknik informatika

[Update Data](#)

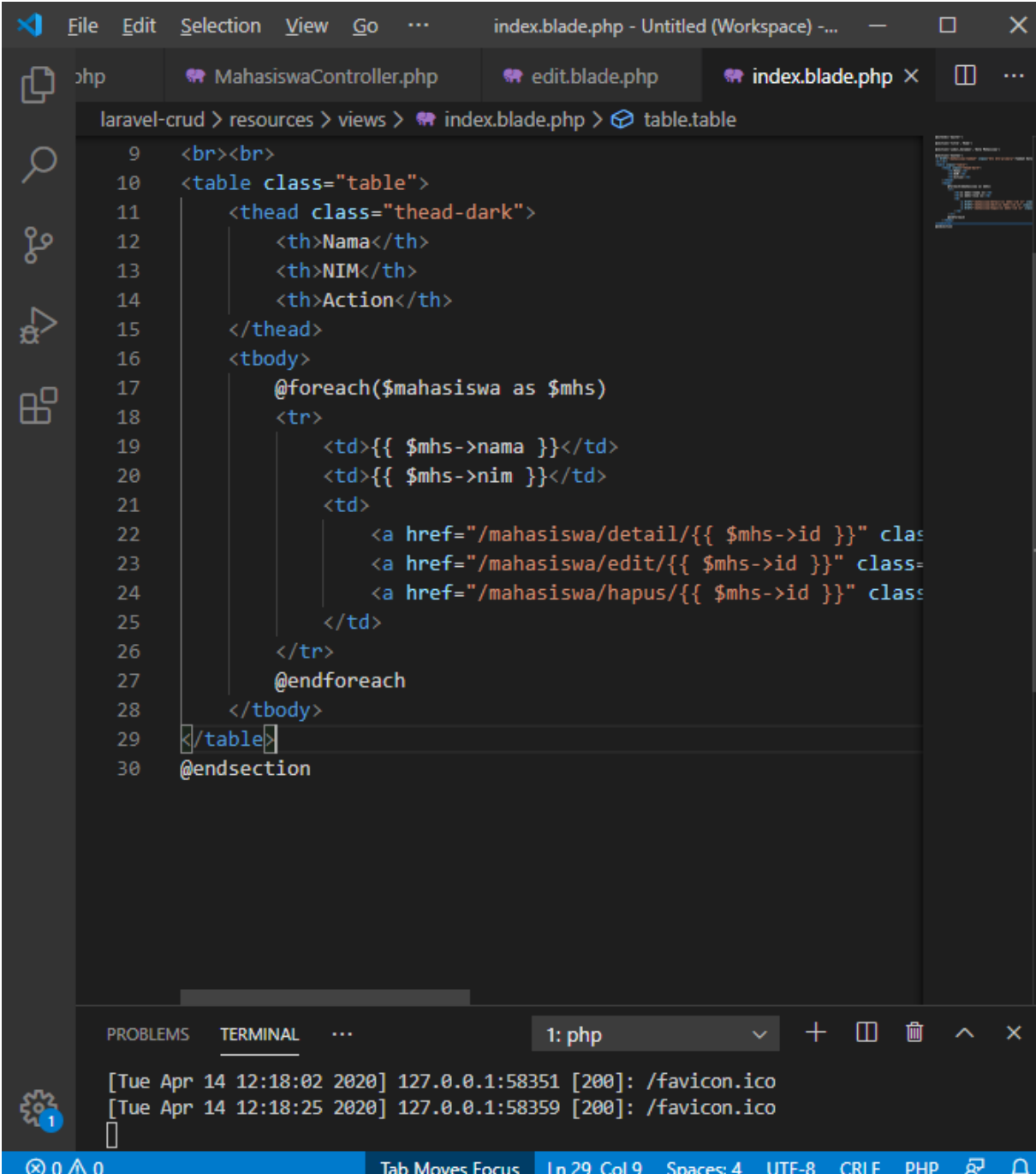
Setelah kita klik simpan Data, maka data pertama akan berubah.

Data Mahasiswa

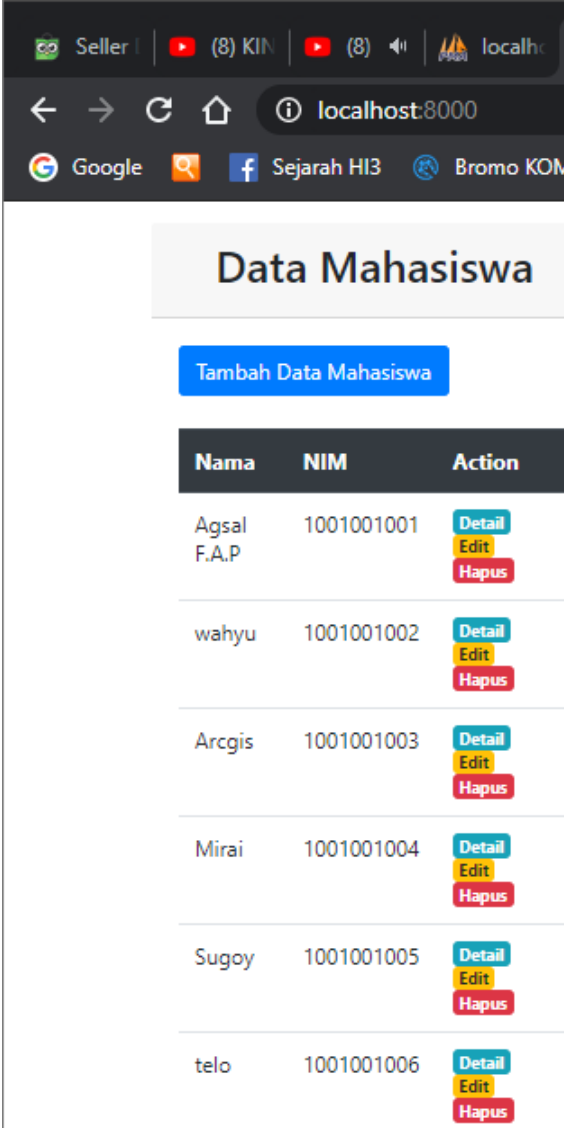
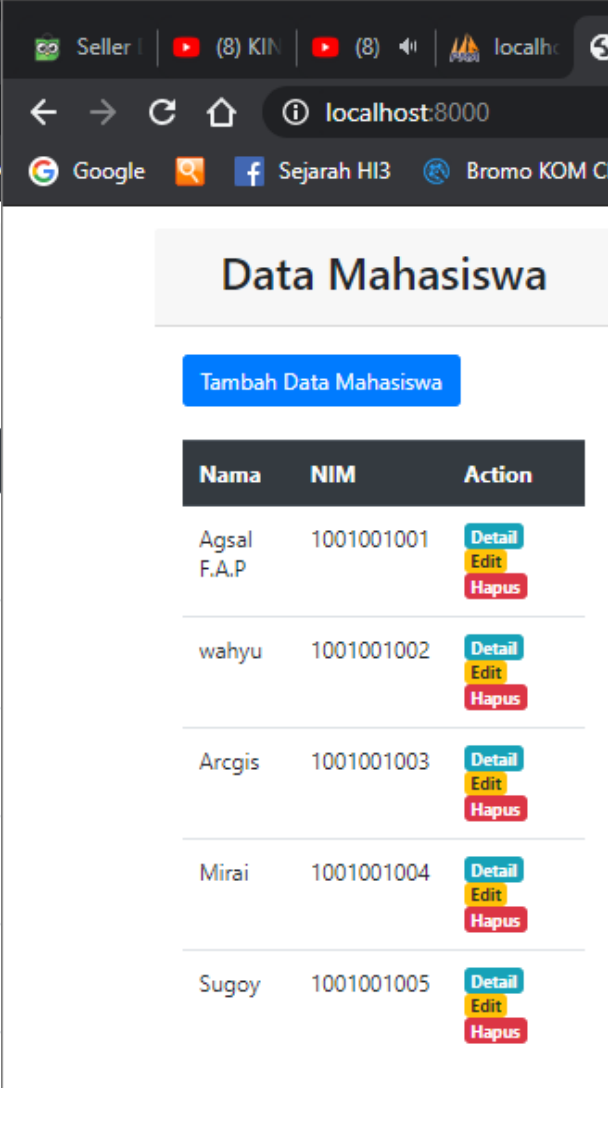
[Tambah Data Mahasiswa](#)

Nama	NIM	Action
Agsal F.A.P	1001001001	Detail Edit
wahyu	1001001002	Detail Edit
Arcgis	1001001003	Detail Edit
Mirai	1001001004	Detail Edit
Sugoy	1001001005	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> 9

 10 <table class="table"> 11 <thead class="thead-dark"> 12 <th>Nama</th> 13 <th>NIM</th> 14 <th>Action</th> 15 </thead> 16 <tbody> 17 @foreach(\$mahasiswa as \$mhs) 18 <tr> 19 <td>{{ \$mhs->nama }}</td> 20 <td>{{ \$mhs->nim }}</td> 21 <td> 22 id }}" class="btn btn-primary">Detail 23 id }}" class="btn btn-warning">Edit 24 id }}" class="btn btn-danger">Hapus 25 </td> 26 </tr> 27 @endforeach 28 </tbody> 29 </table> 30 @endsection </pre> <p>PROBLEMS TERMINAL ... 1: php</p> <pre> [Tue Apr 14 12:18:02 2020] 127.0.0.1:58351 [200]: /favicon.ico [Tue Apr 14 12:18:25 2020] 127.0.0.1:58359 [200]: /favicon.ico </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>Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus');</pre> <pre> 31 32 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>public function hapus(\$id) { // hapus data mahasiswa by id DB::table('mahasiswa')->where('id', \$id)->delete(); return redirect('/'); }</pre> <p>Keterangan :</p> <ul style="list-style-type: none"> - Line 67 : query builder untuk menghapus data mahasiswa berdasarkan id yang dipilih
4	<p>Jalankan localhost:8000 dan pilih tombol ‘Hapus’, maka data yang terpilih akan dihapus. Contoh: menghapus data terakhir.</p> <div>   </div>

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
2	Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini. APP_NAME=Laravel APP_ENV=local APP_KEY=base64:VNVBgoflRkjCg+oYSoTt322r/GZgz8F/3GFFY8ctipc= 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= Keterangan: - 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

Seller | (8) | loc x | Home | Mahas | web.p | +

localhost/phpmyadmin/sql...

Google | Sejarah HI3 | Bromo KOM Challe... | Bookmark lain

Server: 127.0.0.1 » Basis data: latihan_laravel » Tabel: mahasiswa

Jelajahi | Struktur | SQL | Cari | Tambahkan | Lainnya

Terbaru | Favorit

Baru

- biodata
- dbperpus
- ecommerce
- finland
- information_schema
- jne_database
- kuliah_ci
- latihan
 - latihan_ci
 - latihan_laravel
 - Baru
 - mahasiswa
- mysql
- pajak
- pemesanan
- performance_schema
- pesanhotel
- phpmyadmin
- praktikumdb
- rest_server
- test

Selesaikan dalam 0,0000 detik.)

[Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

Saring baris: Cari di tabel ini Urut berdasarkan kunci: Tidak ada

id	nama	email	jurusan	created_at	updated_at
001001001	agsal@gmail.com	teknik informatika	2020-04-14 12:37:57	0000-00-00 00:00:00	
001001002	wahyu@gmail.com	teknik elektro	2020-04-14 12:37:57	0000-00-00 00:00:00	
001001003	Arogis@gmail.com	Teknik Informatika	2020-04-14 12:37:57	0000-00-00 00:00:00	
001001004	Mirai@yourmail.com	Teknik Informatika	2020-04-14 12:37:57	0000-00-00 00:00:00	
001001005	sugoy@Yuhuu.com	Teknologi Informasi	2020-04-14 12:37:57	0000-00-00 00:00:00	

[Ubah] [Salin] [Hapus] [Ekspor]

Saring baris: Cari di tabel ini Urut berdasarkan kunci: Tidak ada

Selesaikan bagan [Buat tampilan]

semua pengguna untuk mengakses markah ini

Markahi kueri SQL ini

Konsol

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');</pre>
2	<p>Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan</p> <pre> cd laravel-crud-kedua php artisan make:model Mahasiswa</pre>
3	<p>Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.</p>  <pre> laravel-crud-kedua > app > Mahasiswa.php > ... 1 <?php 2 3 namespace App; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class Mahasiswa extends Model 8 { 9 protected \$table = "mahasiswa"; 10 11 } 12</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</p> <pre> php artisan make:controller MahasiswaController</pre>

5

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

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** The file explorer on the left shows the project structure. The path `laravel-crud-kedua > app > Http > Controllers > MahasiswaController.php` is highlighted.
- EDITOR:** The main editor window shows the code for `MahasiswaController.php`. The code is as follows:


```

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     //
11     public function index()
12     {
13         $mahasiswa = Mahasiswa::all();
14         return view('index', ['mahasiswa' => $mahasiswa]);
15     }
16 }
17

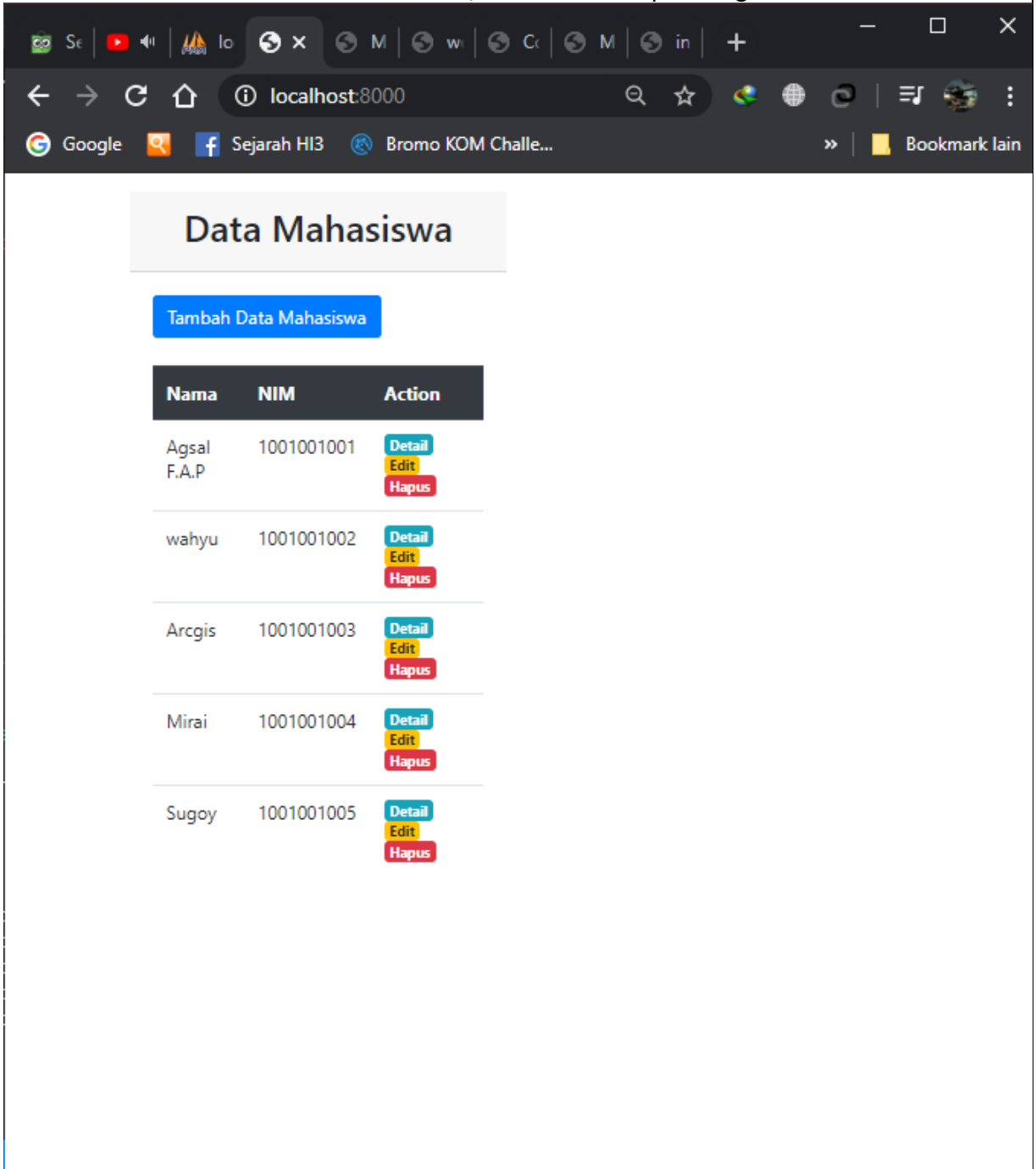
```
- TERMINAL:** The terminal at the bottom shows the execution of Laravel commands:


```

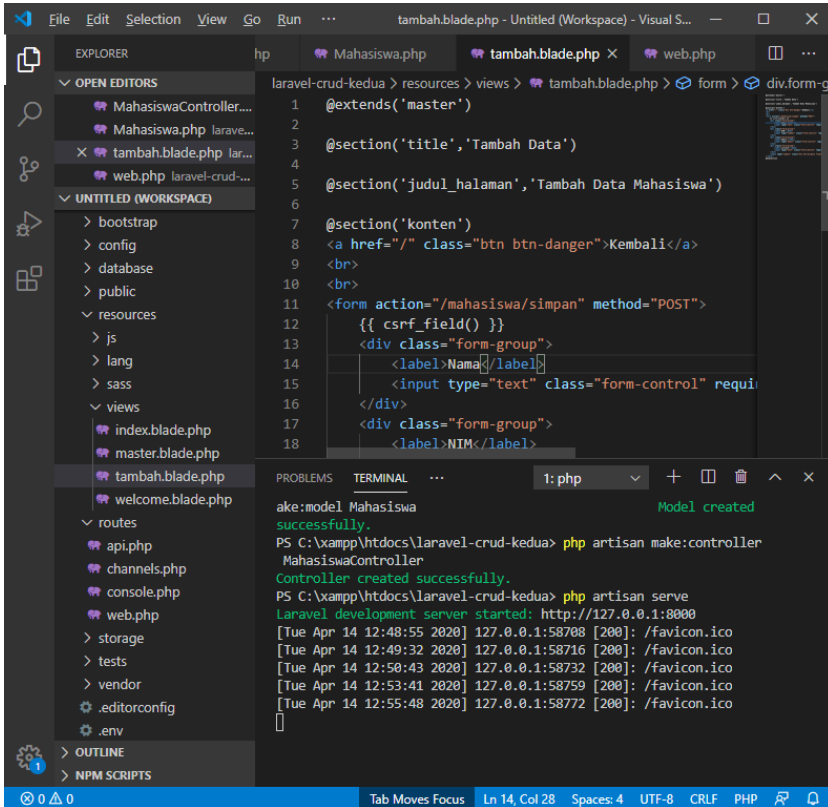
ake:model Mahasiswa
successfully.
PS C:\xampp\htdocs\laravel-crud-kedua> php artisan make:controller
MahasiswaController
Controller created successfully.
PS C:\xampp\htdocs\laravel-crud-kedua> php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Tue Apr 14 12:48:55 2020] 127.0.0.1:58708 [200]: /favicon.ico
[Tue Apr 14 12:49:32 2020] 127.0.0.1:58716 [200]: /favicon.ico
[Tue Apr 14 12:50:43 2020] 127.0.0.1:58732 [200]: /favicon.ico
[Tue Apr 14 12:53:41 2020] 127.0.0.1:58759 [200]: /favicon.ico

```

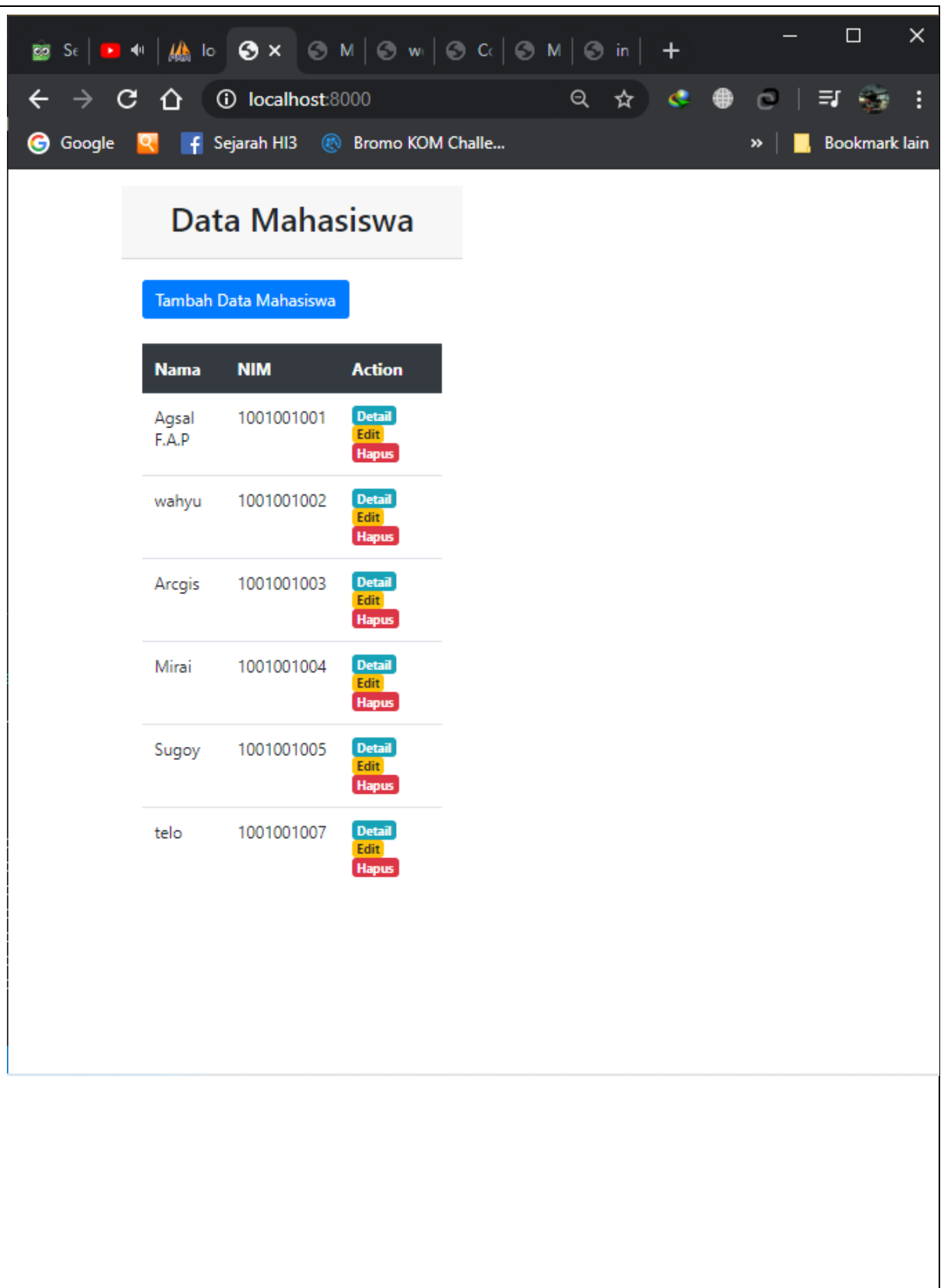
Keterangan:

	<ul style="list-style-type: none">- Tambahkan 'use App\Mahasiswa' (line 6) untuk menggunakan model Mahasiswa- Line 12 untuk mengambil semua data dari model/tabel Mahasiswa dan akan disimpan di variabel \$mahasiswa- Line 16 : data akan dikirim ke blade view bernama index																		
6	Selanjutnya kita akan membuat view untuk menampilkan data mahasiswa dengan nama index.blade.php. Tetapi agar pembuatan view selanjutnya menjadi lebih mudah, terlebih dahulu kita akan membuat template blade (seperti pada Bagian 1). Pada bagian view kita buat seperti bagian 1 (copy-paste dari bagian 1)																		
7	<p>Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud php artisan serve</p> <p>Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut</p>  <p>The screenshot shows a web browser window with the address bar set to 'localhost:8000'. The page content is as follows:</p> <h3>Data Mahasiswa</h3> <p>Tambah Data Mahasiswa</p> <table><tr><th>Nama</th><th>NIM</th><th>Action</th></tr><tr><td>Agsal F.A.P</td><td>1001001001</td><td>Detail Edit Hapus</td></tr><tr><td>wahyu</td><td>1001001002</td><td>Detail Edit Hapus</td></tr><tr><td>Arcgis</td><td>1001001003</td><td>Detail Edit Hapus</td></tr><tr><td>Mirai</td><td>1001001004</td><td>Detail Edit Hapus</td></tr><tr><td>Sugoy</td><td>1001001005</td><td>Detail Edit Hapus</td></tr></table>	Nama	NIM	Action	Agsal F.A.P	1001001001	Detail Edit Hapus	wahyu	1001001002	Detail Edit Hapus	Arcgis	1001001003	Detail Edit Hapus	Mirai	1001001004	Detail Edit Hapus	Sugoy	1001001005	Detail Edit Hapus
Nama	NIM	Action																	
Agsal F.A.P	1001001001	Detail Edit Hapus																	
wahyu	1001001002	Detail Edit Hapus																	
Arcgis	1001001003	Detail Edit Hapus																	
Mirai	1001001004	Detail Edit Hapus																	
Sugoy	1001001005	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>Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah');</pre>
2	<p>Buat method tambah pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view tambah.</p> <pre>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>
	<p>Kita lakukan copy paste dari tambah.blade.php di Bagian 1</p>  <p>Ketika tombol simpan ditekan, akan dipanggil routes /mahasiswa/simpan. Oleh karena itu, kita buat terlebih dahulu route tersebut.</p> <pre>Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php

5	<p>Buat method simpan pada MahasiswaController untuk menyimpan data ke database.</p> <pre> public function simpan(Request \$request) { // insert data ke table mahasiswa Mahasiswa::create(['nama' => \$request->namamhs, 'nim' => \$request->nimmhs, 'email' => \$request->emailmhs, 'jurusan' => \$request->jurusanmhs]); return redirect('/'); } </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
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> 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>

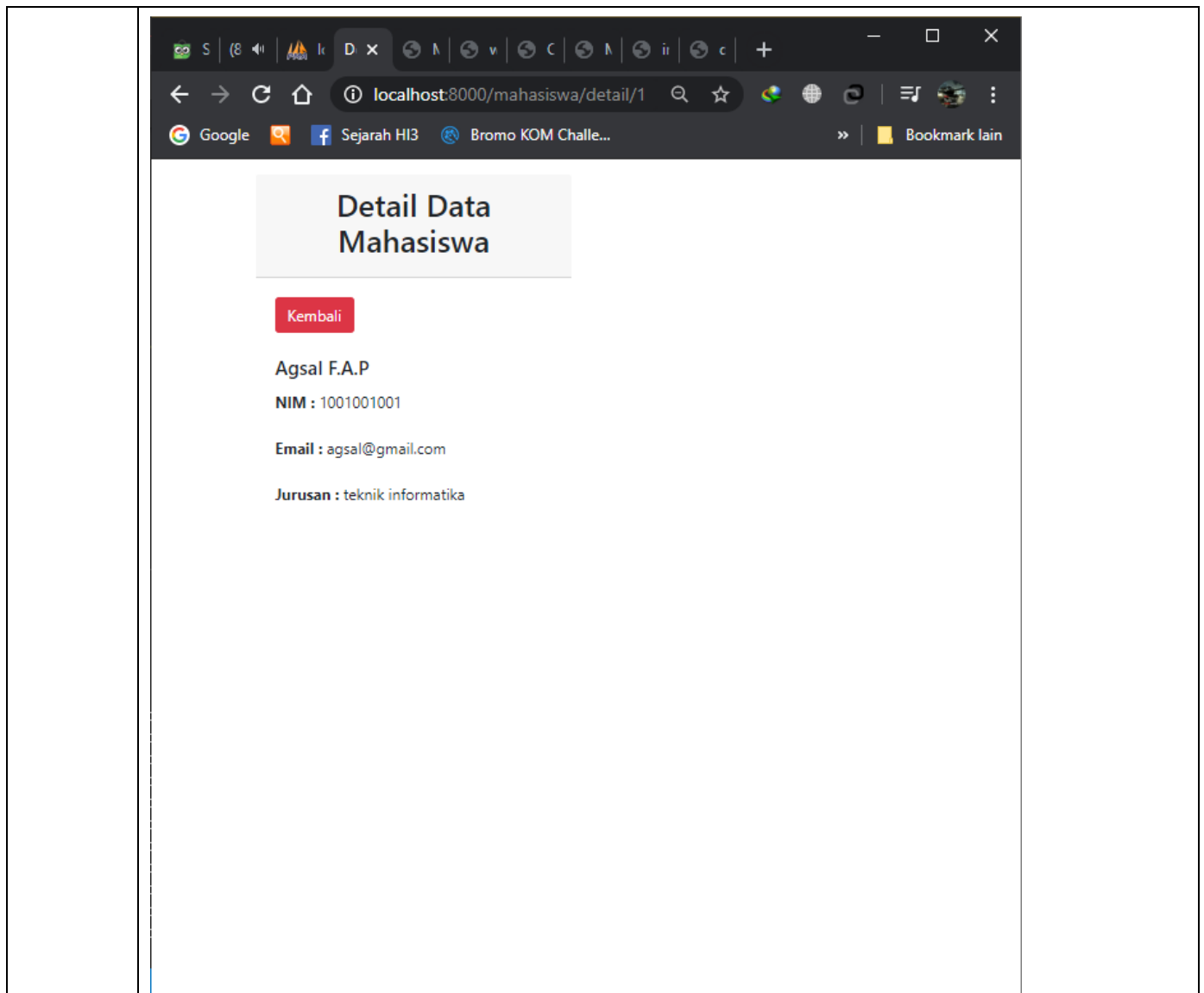


d. Melihat Detail Data dari Database

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

	<pre>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>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>@extends('master') @section('title','Detail Mahasiswa') @section('judul_halaman','Detail Data Mahasiswa') @section('konten') Kembali

 <h5 class="card-title">{{ \$mahasiswa->nama }}</h5> <p class="card-text"> <label for="">NIM : </label> {{ \$mahasiswa->nim }} </p> <p class="card-text"> <label for="">Email : </label> {{ \$mahasiswa->email }} </p> <p class="card-text"> <label for="">Jurusan : </label> {{ \$mahasiswa->jurusan }} </p> @endsection</pre>
5	<p>Jalankan localhost:8000 dan pilih tombol 'Detail' di suatu data yang ingin kita lihat detailnya.</p>



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>Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit');</pre>
2	<p>Buat method edit pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan view edit.</p>

	<pre> public function edit(\$id) { \$mahasiswa = Mahasiswa::find(\$id); return view('edit', ['mahasiswa' => \$mahasiswa]); } </pre>
	<p>Keterangan :</p> <ul style="list-style-type: none"> - Line 41 : fungsi eloquent untuk mengambil data mahasiswa berdasarkan id yang dipilih
3	<p>Kemudian buatlah view edit.blade.php yang berisi form untuk mengubah data pada folder resources/views. View edit juga mengaplikasikan master.blade.php.</p> <pre> @extends('master') @section('title','Edit Data') @section('judul_halaman','Edit Data Mahasiswa') @section('konten') Kembali

 <form action="/mahasiswa/update" method="POST"> {{ csrf_field() }} <input type="hidden" name="id" value="{{ \$mahasiswa->id }}"> <div class="form-group"> <label>Nama</label> <input type="text" class="form-control" value="{{ \$mahasiswa- >nama }}" required="required" name="namamhs">
 </div> <div class="form-group"> <label>NIM</label> <input type="number" class="form-control" value="{{ \$mahasiswa- >nim }}" required="required" name="nimmhs">
 </div> <div class="form-group"> <label>Email</label> <input type="email" class="form-control" value="{{ \$mahasiswa- >email }}" required="required" name="emailmhs">
 </div> <div class="form-group"> <label>Jurusan</label> <input type="text" class="form-control" value="{{ \$mahasiswa- >jurusan }}" required="required" name="jurusanmhs">
 </div> <input type="submit" class="btn btn-primary float- right" value="Update Data"> </form> @endsection </pre>

```

4
5 @section('judul_halaman','Edit Data Mahasiswa')
6
7 @section('konten')
8 <a href="/" class="btn btn-danger">Kembali</a>
9 <br>
10 <br>
11 <form action="/mahasiswa/update" method="POST">
12     {{ csrf_field() }}
13     <input type="hidden" name="id" value="{{ $mahasiswa->id }}" />
14     <div class="form-group">
15         <label>Nama</label>
16         <input type="text" class="form-control" value="{{ $mahasiswa->nama }}" />
17     </div>
18     <div class="form-group">
19         <label>NIM</label>
20         <input type="number" class="form-control" value="{{ $mahasiswa->nim }}" />
21     </div>
22     <div class="form-group">
23         <label>Email</label>
24         <input type="email" class="form-control" value="{{ $mahasiswa->email }}" />
25     </div>
26     <div class="form-group">
27         <label>Jurusan</label>
28         <input type="text" class="form-control" value="{{ $mahasiswa->jurusan }}" />
29     </div>
30     <input type="submit" class="btn btn-primary float-right" value="Simpan" />
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

4

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

```
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

5

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

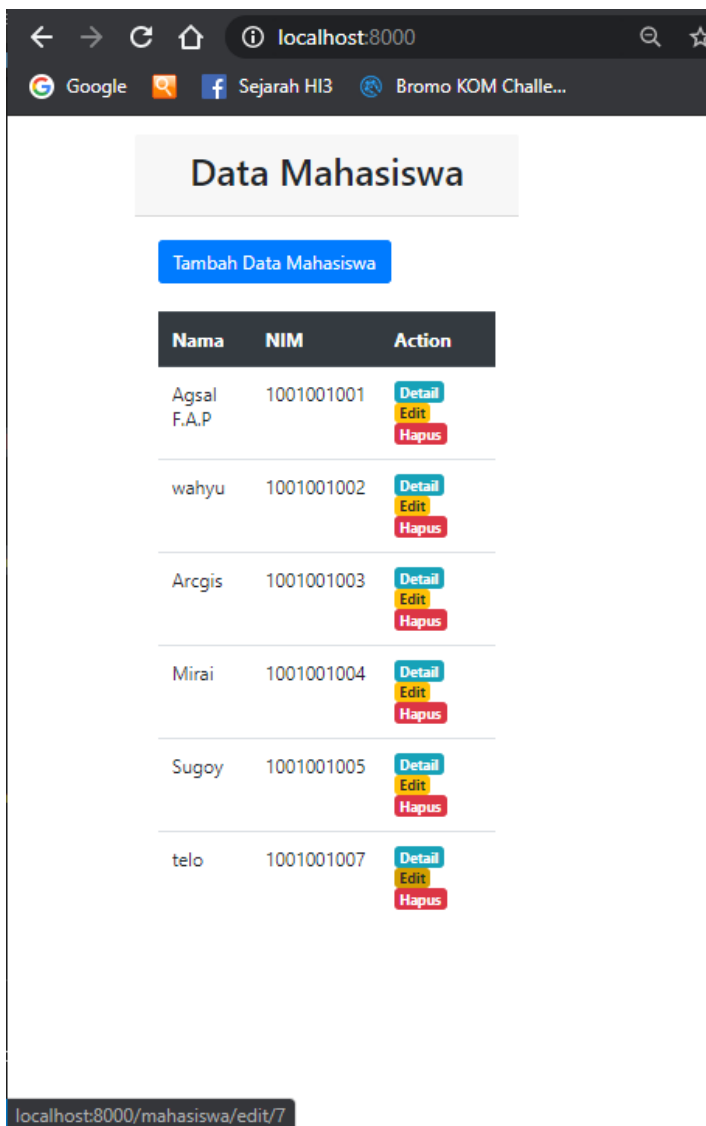

```

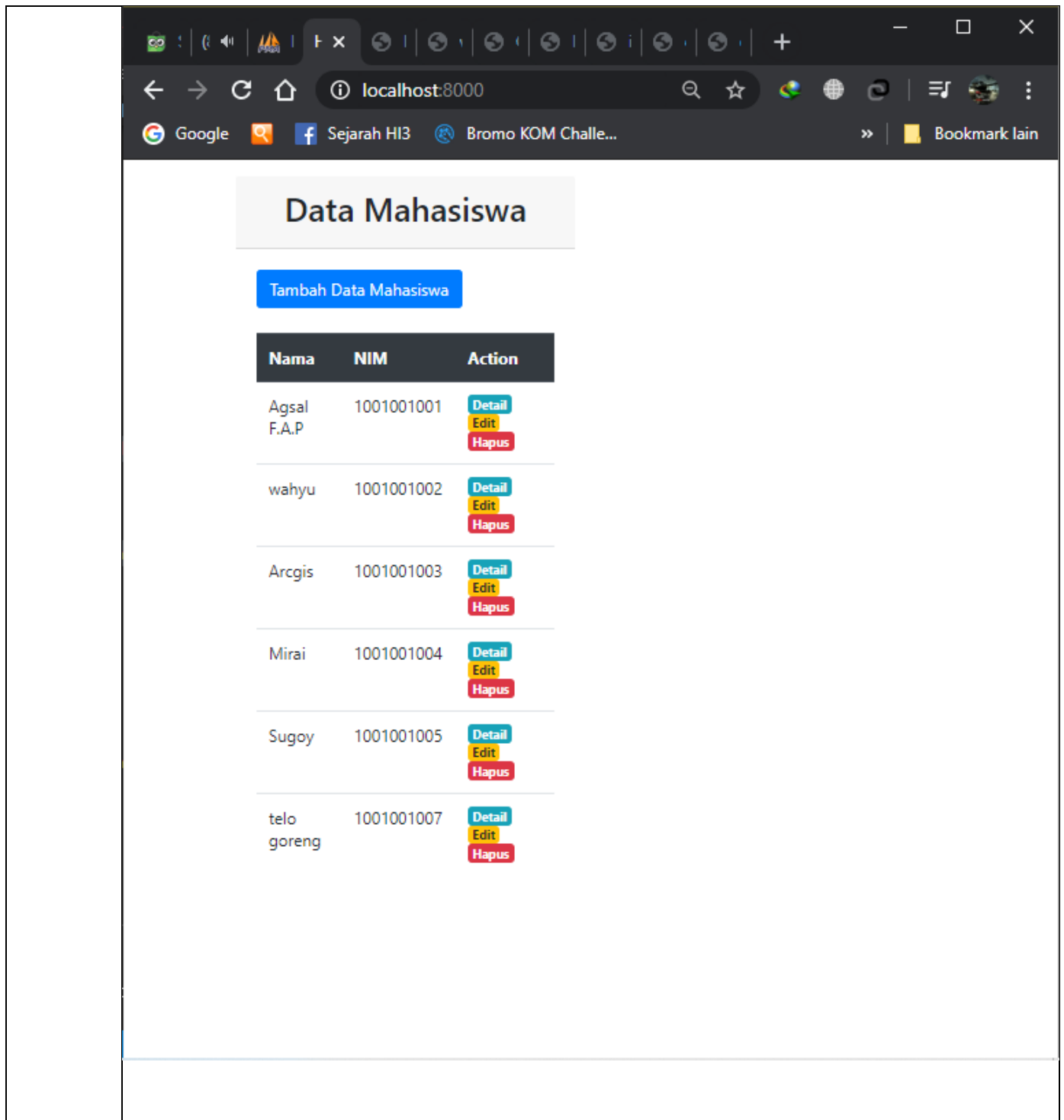
public function update(Request $request)
{
    // update data mahasiswa
    $mahasiswa = Mahasiswa::find($request->id);
    $mahasiswa->nama = $request->namamhs;
    $mahasiswa->nim = $request->nimmhs;
    $mahasiswa->email = $request->emailmhs;
    $mahasiswa->jurusan = $request->jurusanmhs;
    $mahasiswa->save();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

```

6

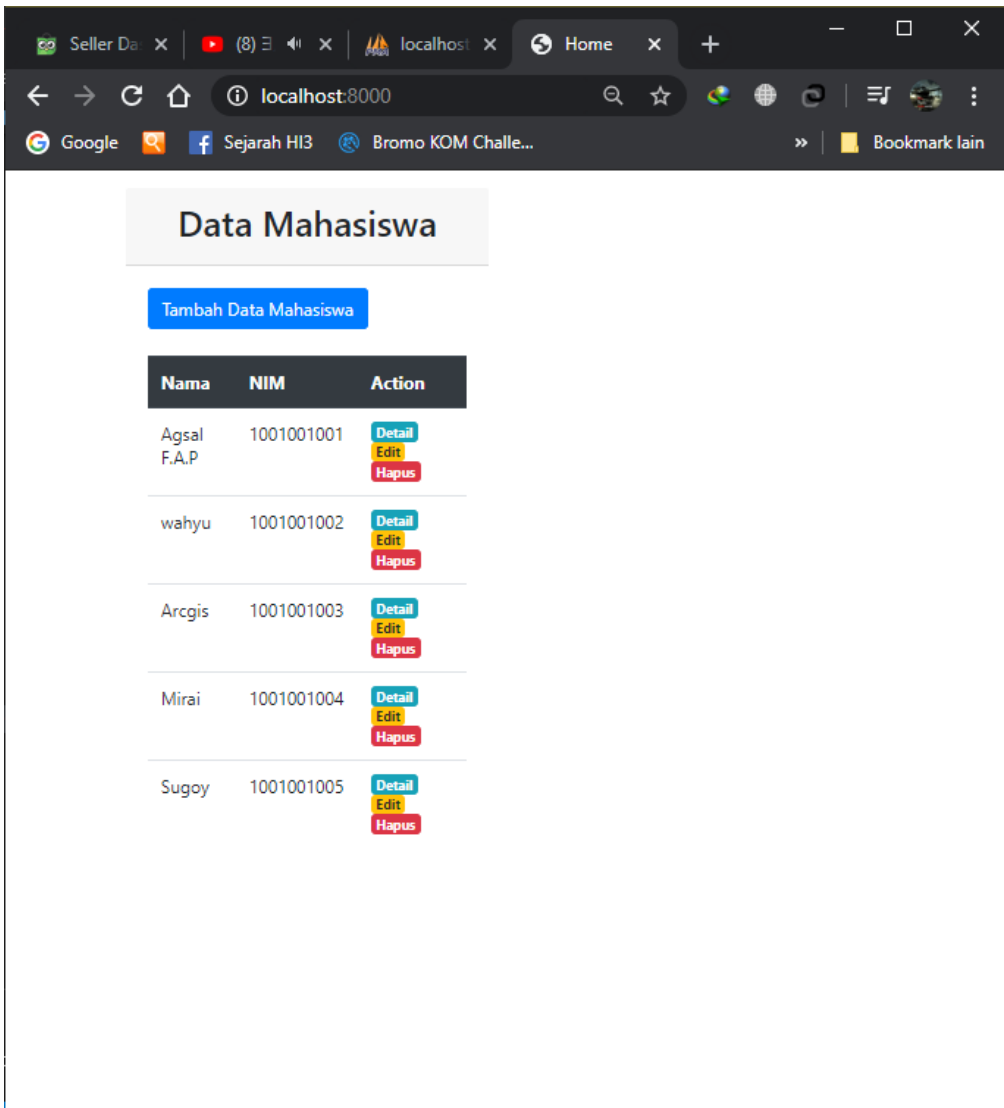
Jalankan localhost:8000 dan pilih tombol 'Edit', maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru.
Cobalah untuk mengedit data.





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>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>public function hapus(\$id) { // hapus data mahasiswa by id \$mahasiswa = Mahasiswa::find(\$id); //menghapus data berdasarkan id \$mahasiswa->delete(); return redirect('/'); }</pre>
	<p>Keterangan :</p> <ul style="list-style-type: none">- fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih
3	<p>Jalankan localhost:8000 dan pilih tombol 'Hapus', maka data yang terpilih akan dihapus.</p> 

-- Selamat Mengerjakan --