**Jobsheet-10: Membuat CRUD menggunakan Laravel**

**Mata Kuliah Pemrograman Web Lanjut**

****

**Oleh :**

**M AQSAL ZILDJIAN/ NIM : 1841720191**

**KELAS TI 2A**

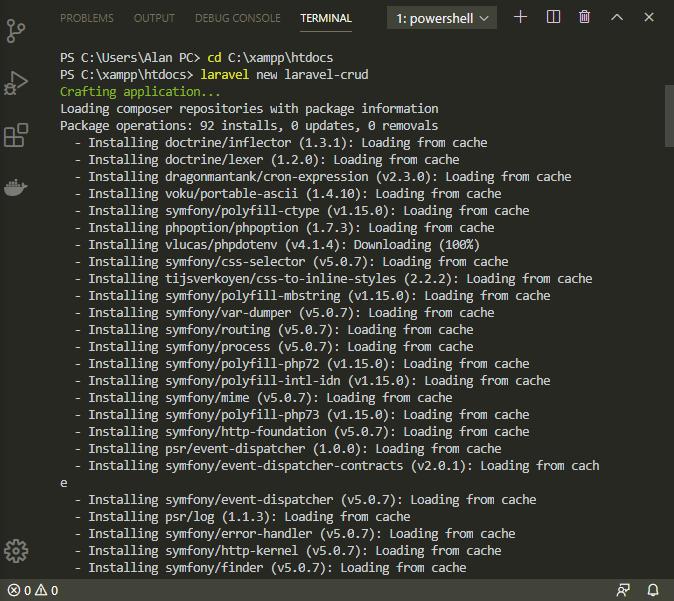
**PROGRAM STUDI TEKNIK INFORMATIKA JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI MALANG**

**2020**

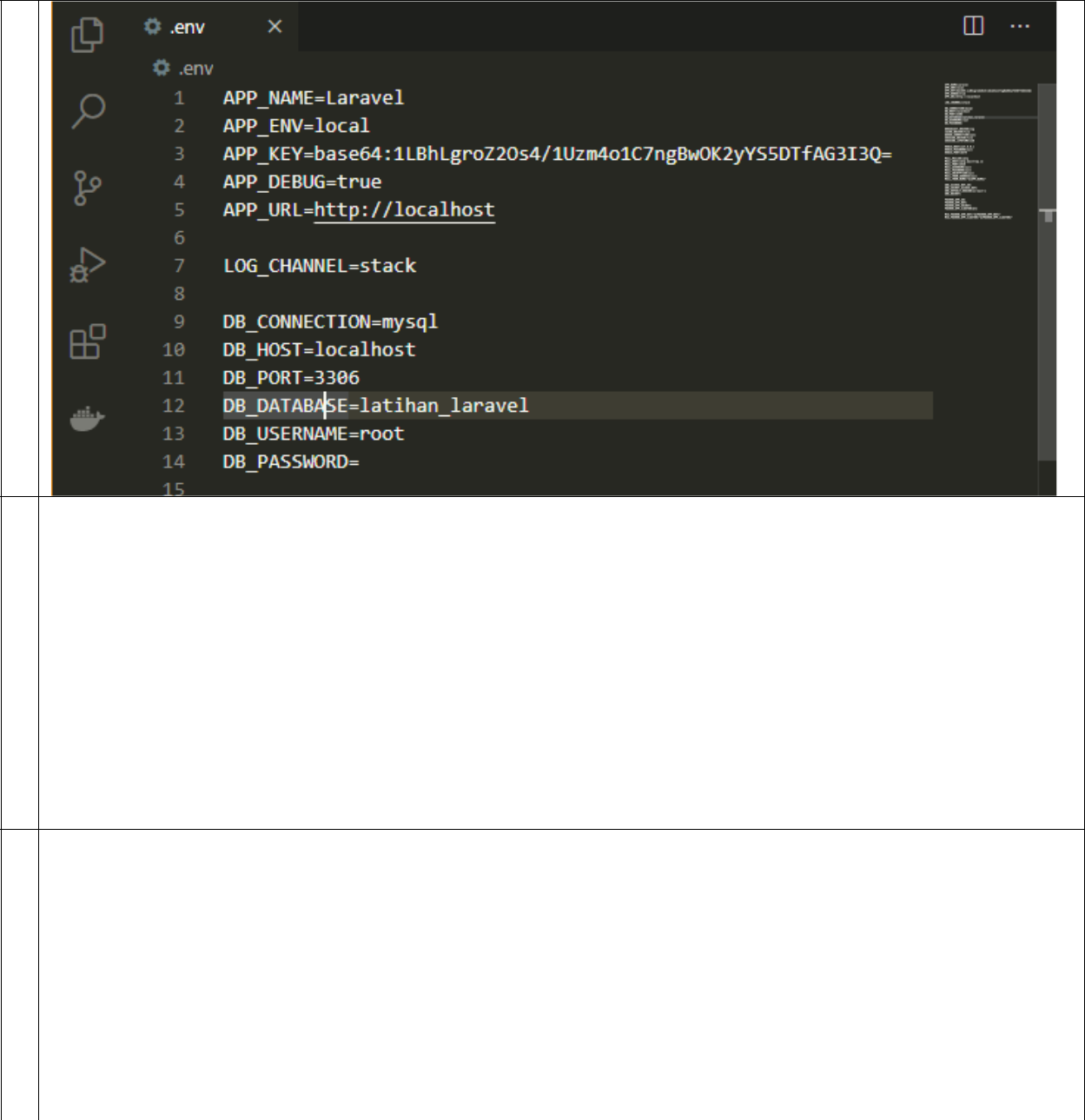
**Praktikum – Bagian 1: Membuat CRUD di Laravel menggunakan Query Builder**

1. Buatlah project Laravel baru dengan nama laravel-crud. Buka command prompt, tuliskan perintah berikut.

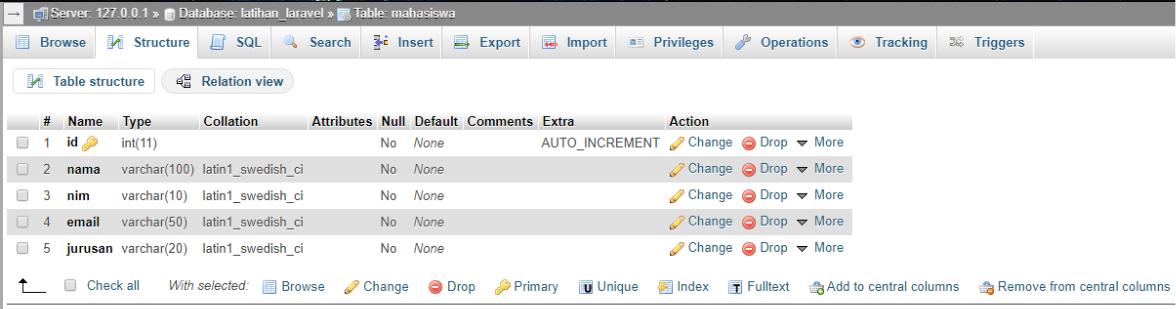
cd C:\xampp\htdocs laravel new laravel-crud



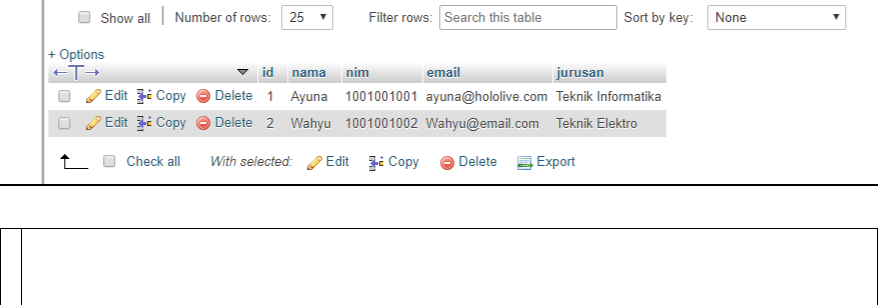
Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.



Jalankan xampp, selanjutnya buat tabel dengan nama mahasiswa di mysql pada database latihan\_laravel.



Isilah beberapa data pada tabel mahasiswa tersebut.

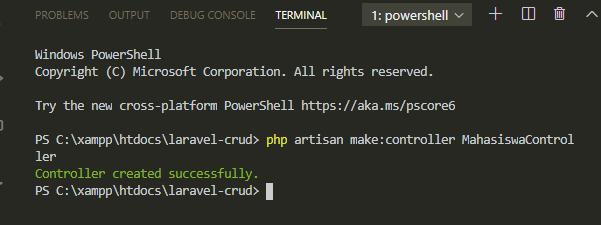


Setelah kita memiliki beberapa data pada tabel mahasiswa, kita akan mencoba untuk menampilkan data tersebut ketika project dijalankan. Pertama, buatlah route pada routes/web.php sehingga ketika pertama kali project dijalankan akan terbuka halaman yang menampilkan data.

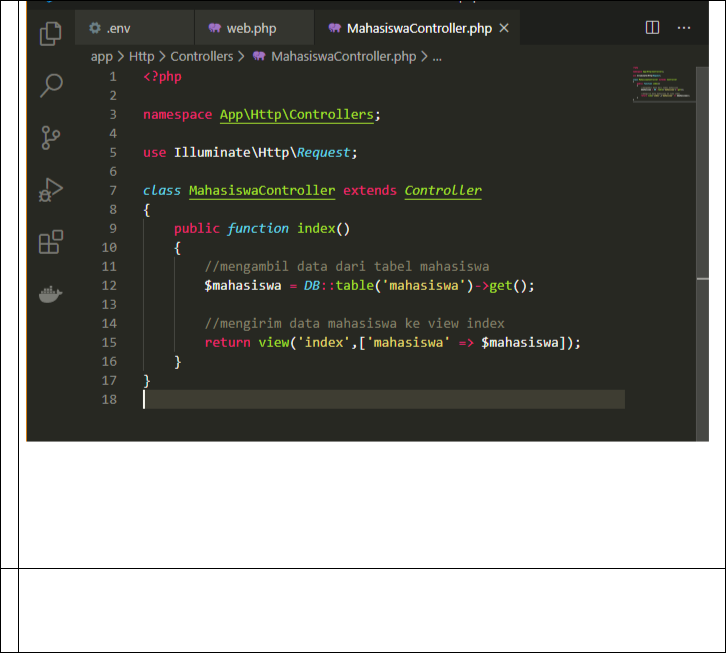


Buat controller baru menggunakan command prompt yaitu MahasiswaController menggunakan php artisan

cd laravel-crud php artisan make:controller MahasiswaController

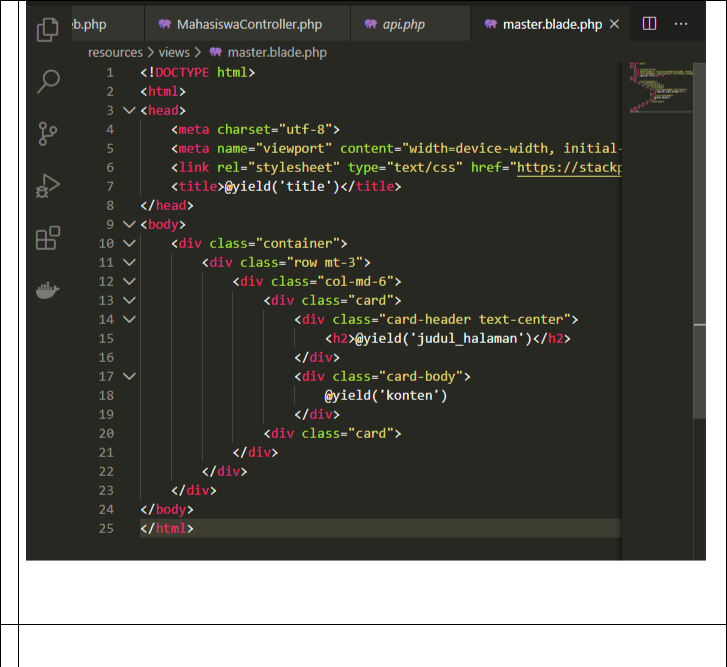


Buat method index pada MahasiswaController.php pada folder app/Http/Controllers



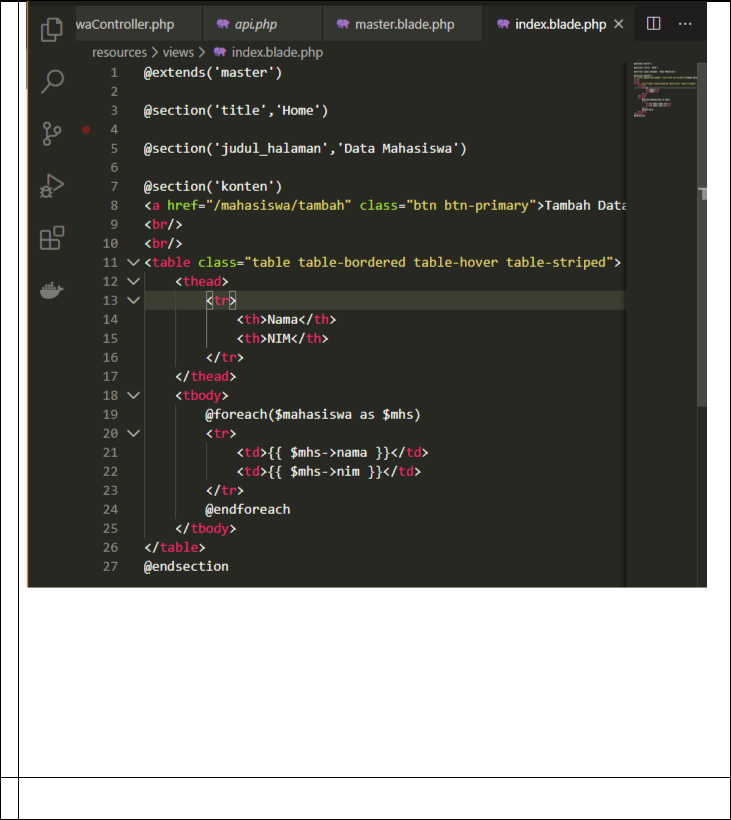
Keterangan: - Tambahkan ‘use Illuminate\Support\Facades\DB;’ (line 6) agar query builder dapat digunakan - Line 13 untuk mengambil data dari tabel mahasiswa menggunakan query builder laravel dan akan disimpan di variabel $mahasiswa - Line 16 : data akan dikirim ke blade view bernama index

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.



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

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



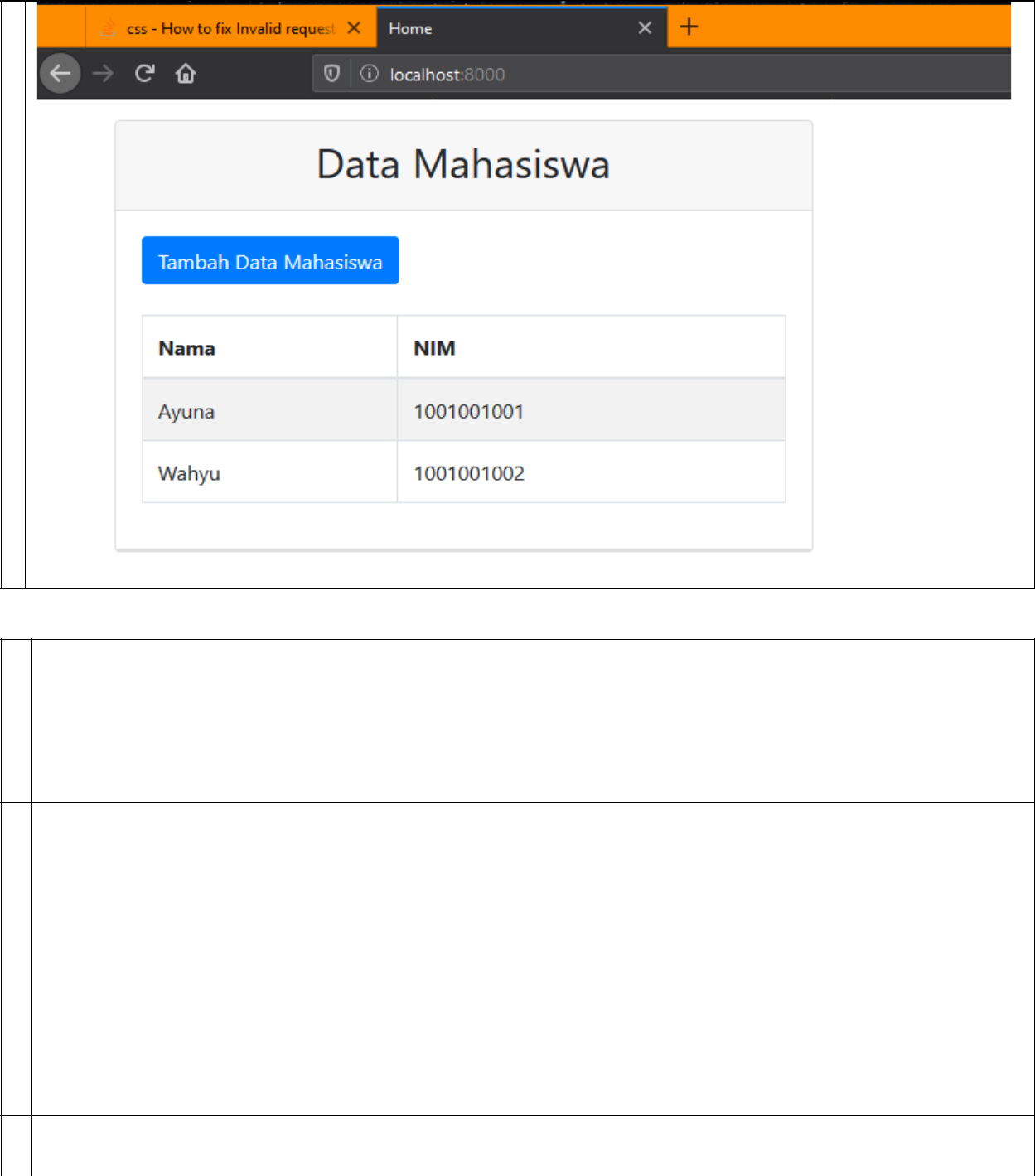
Keterangan: - Line 1 : @extends menunjukkan bahwa file index.blade.php menerapkan blade lain yaitu master.blade.php

* Line 4 : @section(‘title’, ‘Home’) berarti bahwa file index mengisi @yield(‘title’) pada master dengan ‘Home’
  + Line 7 : @section(‘judul\_halaman’, ‘Data Mahasiswa) berarti bahwa file index mengisi @yield(‘judul\_halaman) pada master dengan ‘Home’
* Line 10-30 : mengisi yield(@konten), karena terdapat banyak baris pada diawali dengan

@section(‘konten’) dan diakhiri dengan @endsection

* Line 24-25 menampilkan data mahasiswa dengan kolom nama dan nim

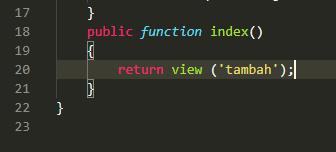
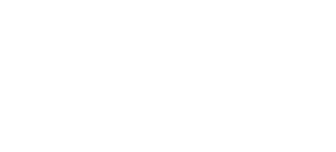
Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud php artisan serve Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut



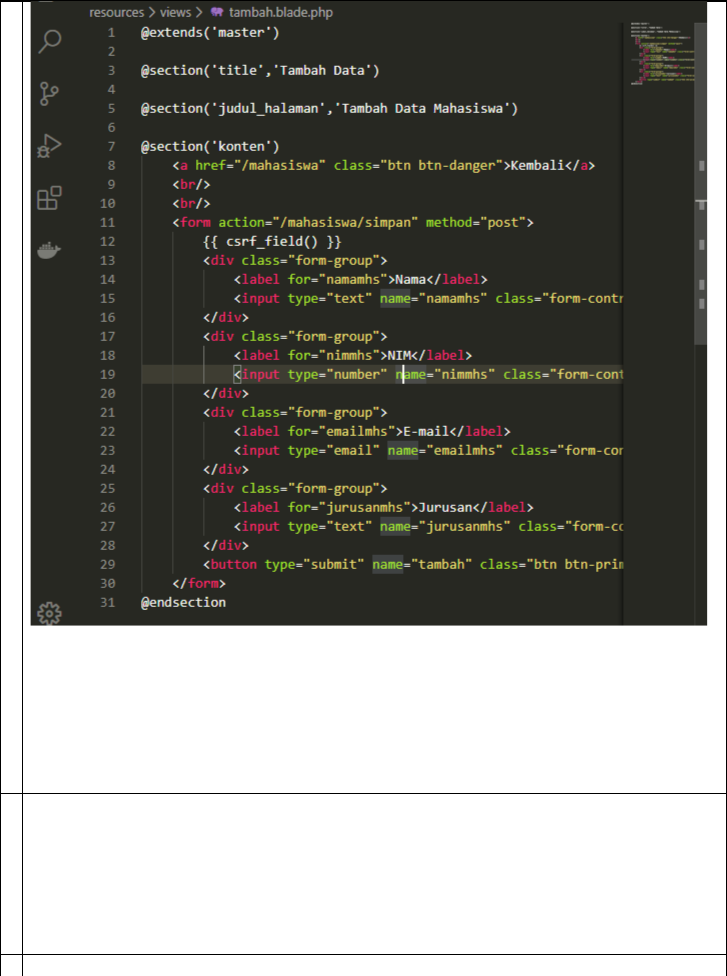
Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController



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



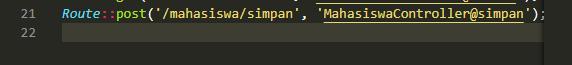
Kemudian buatlah view tambah.blade.php yang berisi form untuk memasukkan data baru pada folder resources/views. View tambah juga mengaplikasikan master.blade.php



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.

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



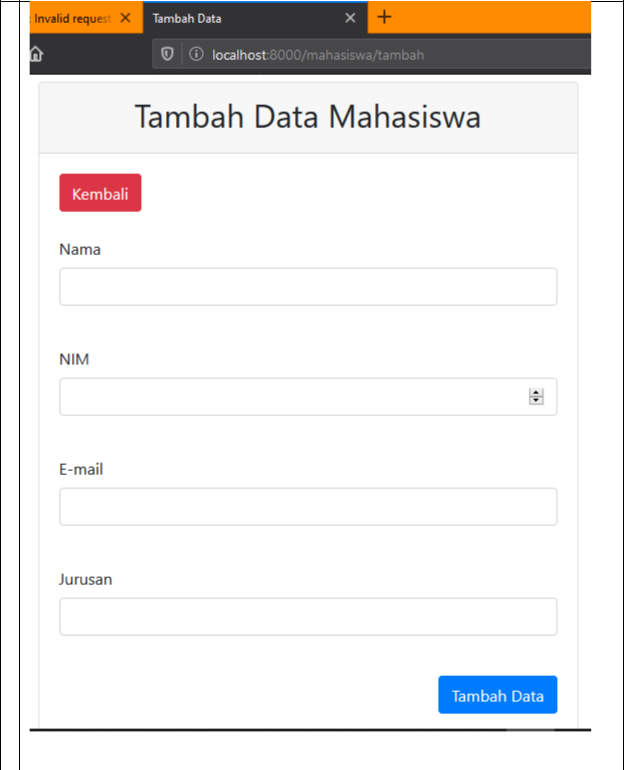
Keterangan: - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php

Buat method simpan pada MahasiswaController untuk menyimpan data ke database.

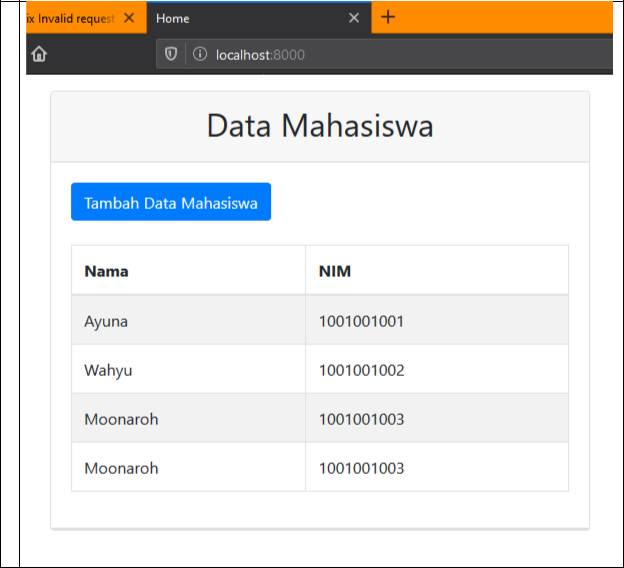


Keterangan: - Variabel $request untuk menerima data yang akan ditambahkan ke database - Line 28-33 merupakan query builder untuk insert data ke tabel mahasiswa

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.

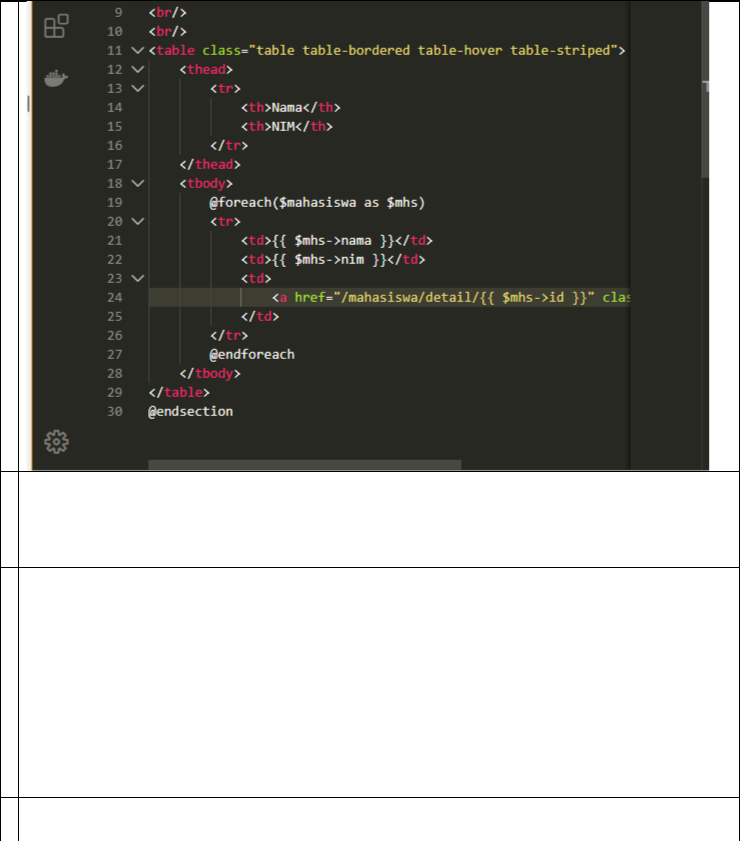


Setelah kita klik tombol tambah data, maka data baru akan ditampilkan pada halaman index.



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

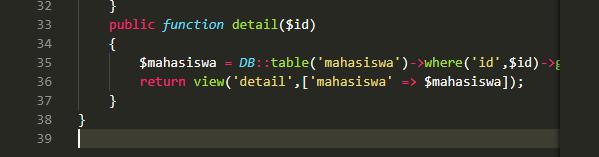
Buatlah tombol untuk melihat detail data (Line 28) pada file index.blade.php di folder resources/views



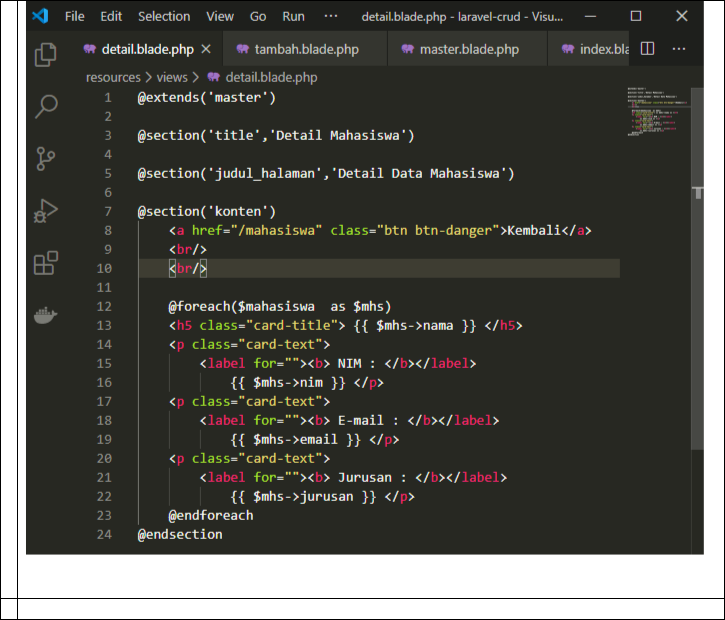
Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController



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

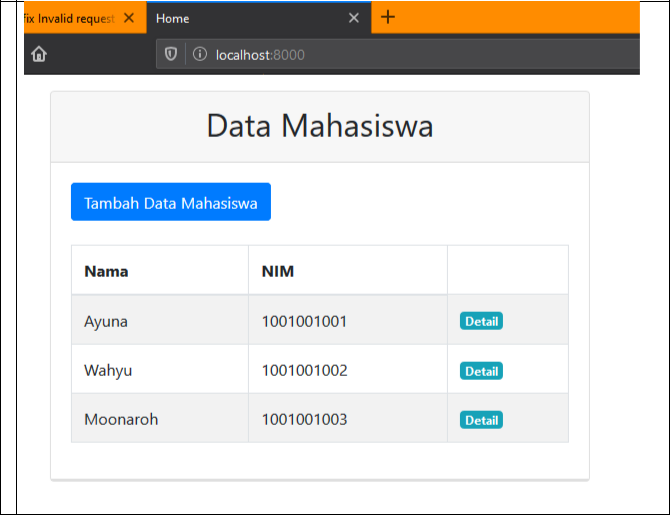


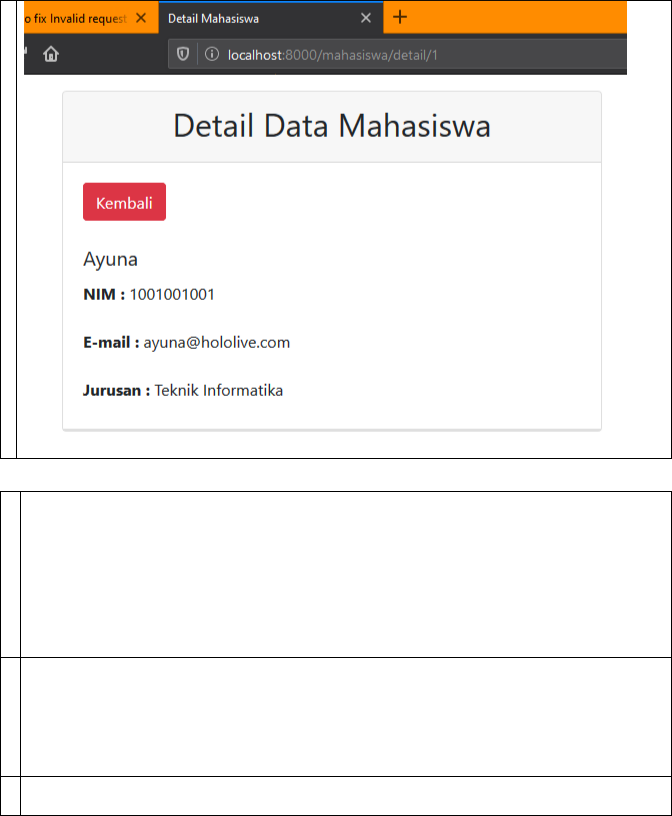
Kemudian buatlah view detail.blade.php yang menampilkan detail data mahasiswa pada folder resources/views. View detail juga mengaplikasikan master.blade.php.



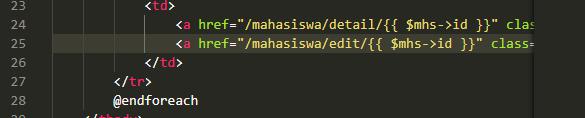
Keterangan: - Line 16-27 digunakan untuk menampilkan data mahasiswa berupa nama, nim, email, dan jurusan

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

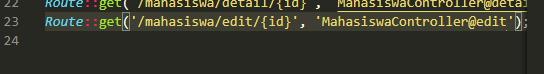




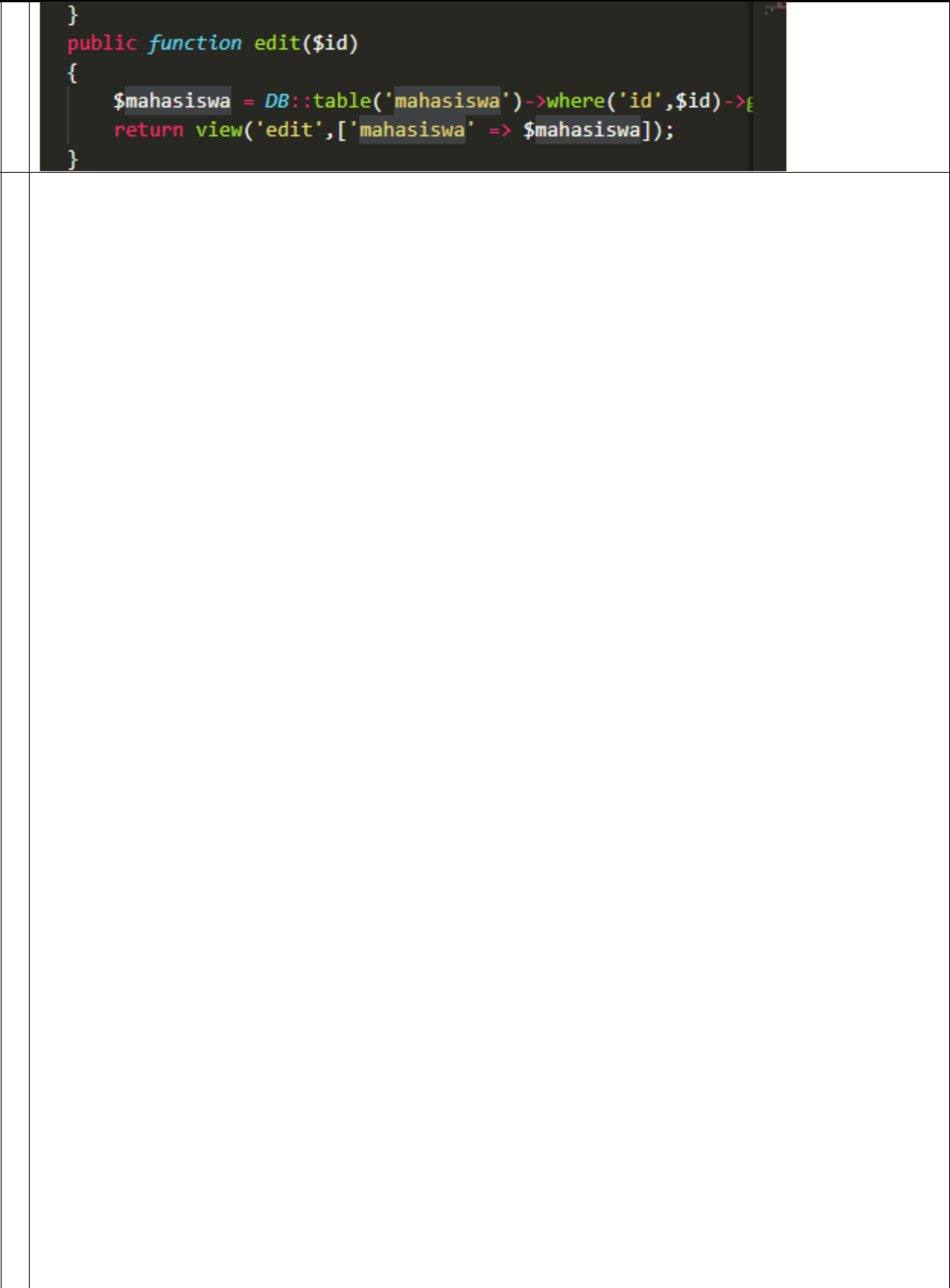
Buatlah tombol untuk mengubah data (Line 29) pada file index.blade.php di folder resources/views]



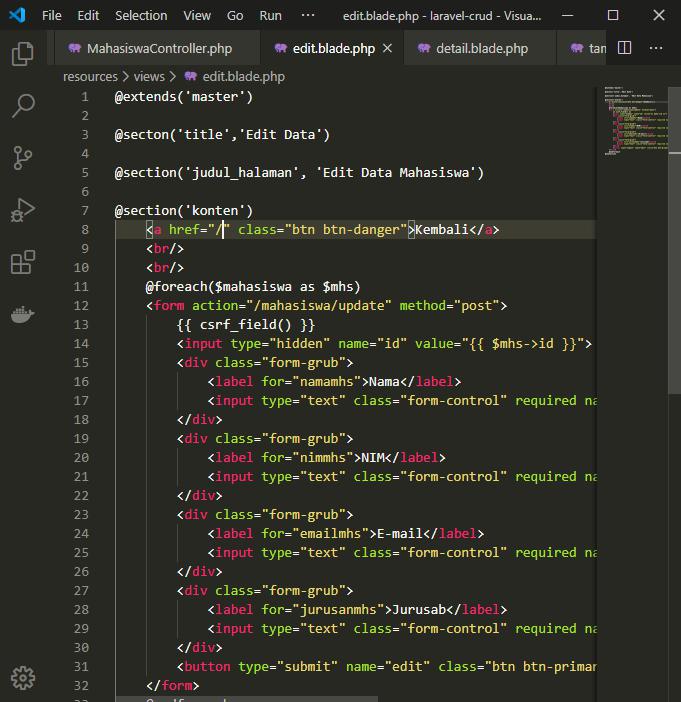
Buatlah route baru pada routes/web.php dengan nama /mahasiswa/edit yang akan menjalankan fungsi edit pada MahasiswaController



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



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



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

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



Keterangan:

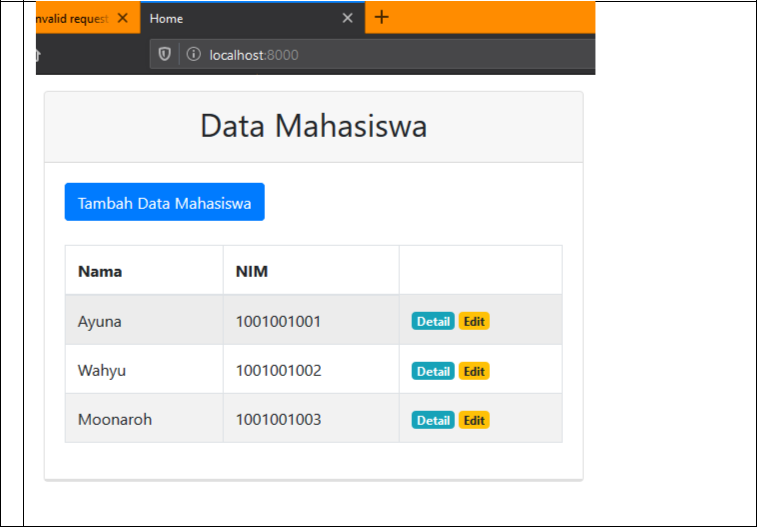
* Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php

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

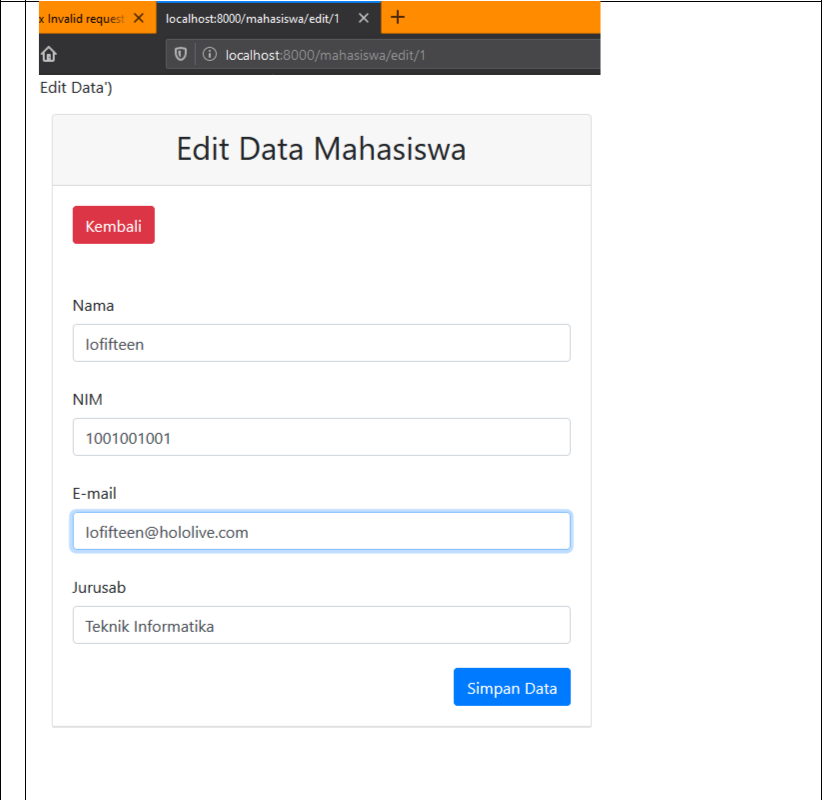


Keterangan: - Variabel $request untuk menerima data yang akan ditambahkan ke database - Line 58-63 merupakan query builder untuk update data ke tabel mahasiswa

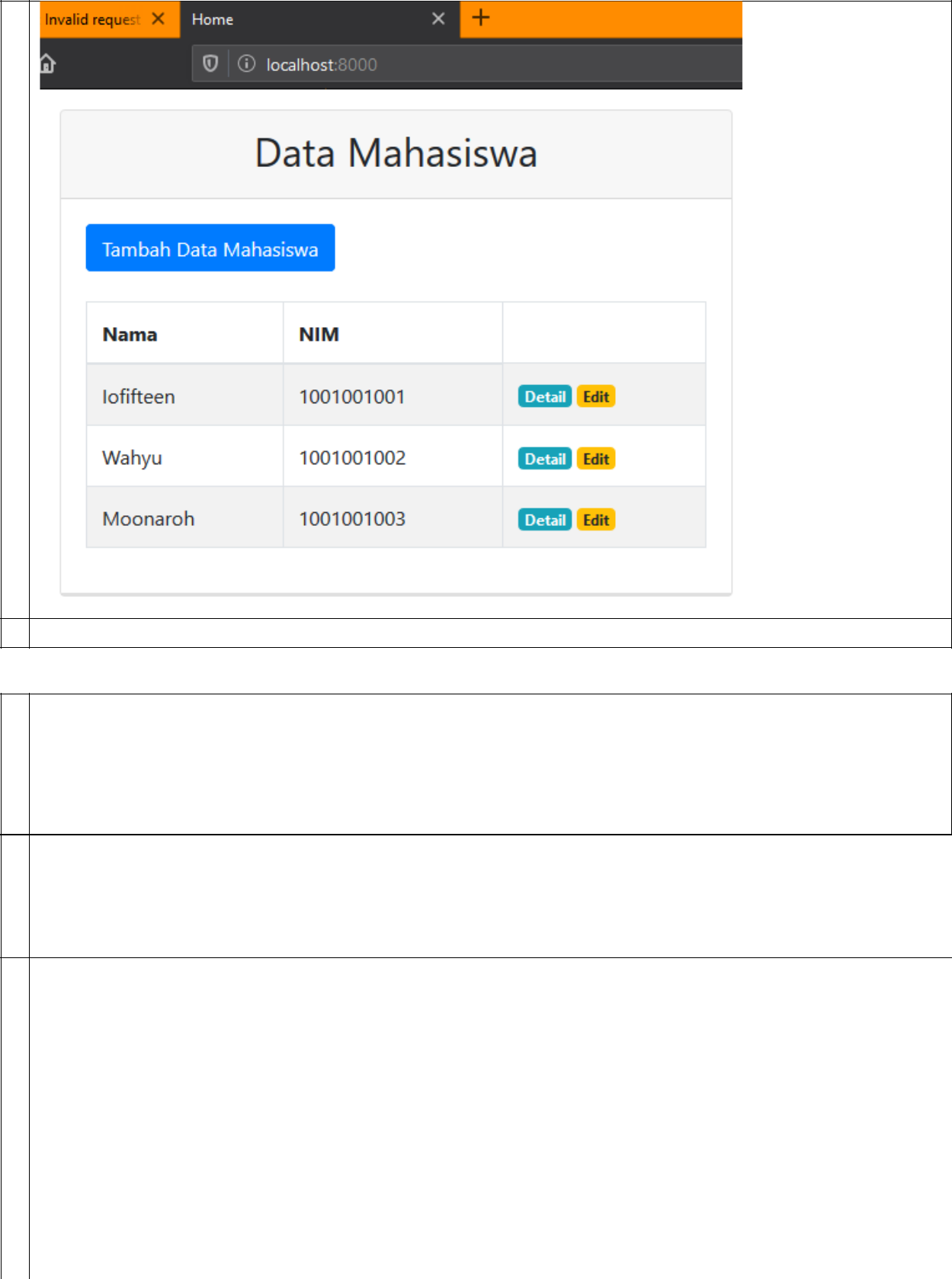
Jalankan localhost:8000 dan pilih tombol ‘Edit’, maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru.



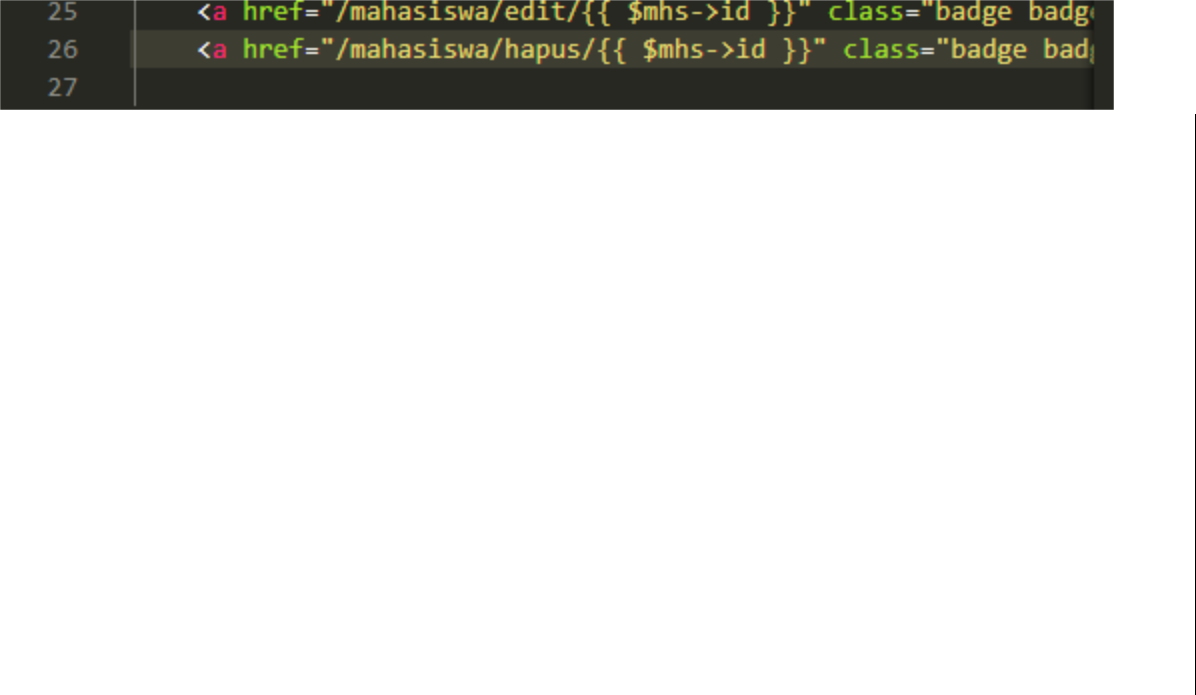
Klik edit di data pertama, kita akan mengubah namanya.



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



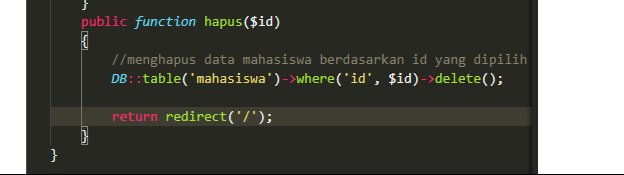
Buatlah tombol untuk menghapus data (Line 30) pada file index.blade.php di folder resources/views



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



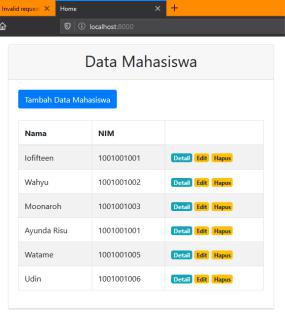
Buat method hapus pada MahasiswaController.php di folder app/Http/Controllers yang akan menjalankan fungsi hapus.



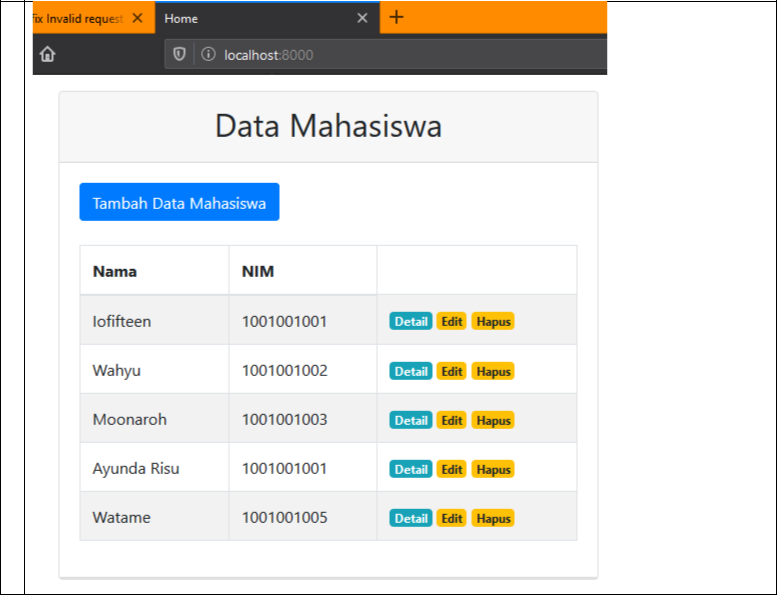
Keterangan : - Line 67 : query builder untuk menghapus data mahasiswa berdasarkan id yang dipilih

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

menghapus data terakhir.

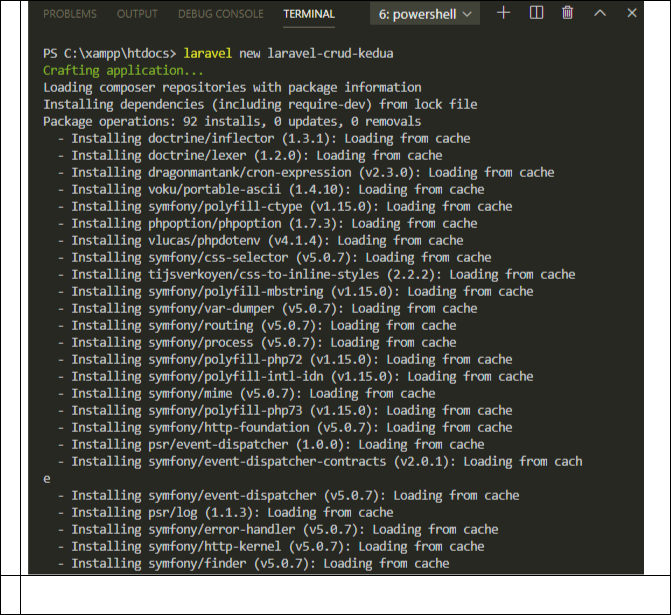


Setelah menghpaus data terakhir

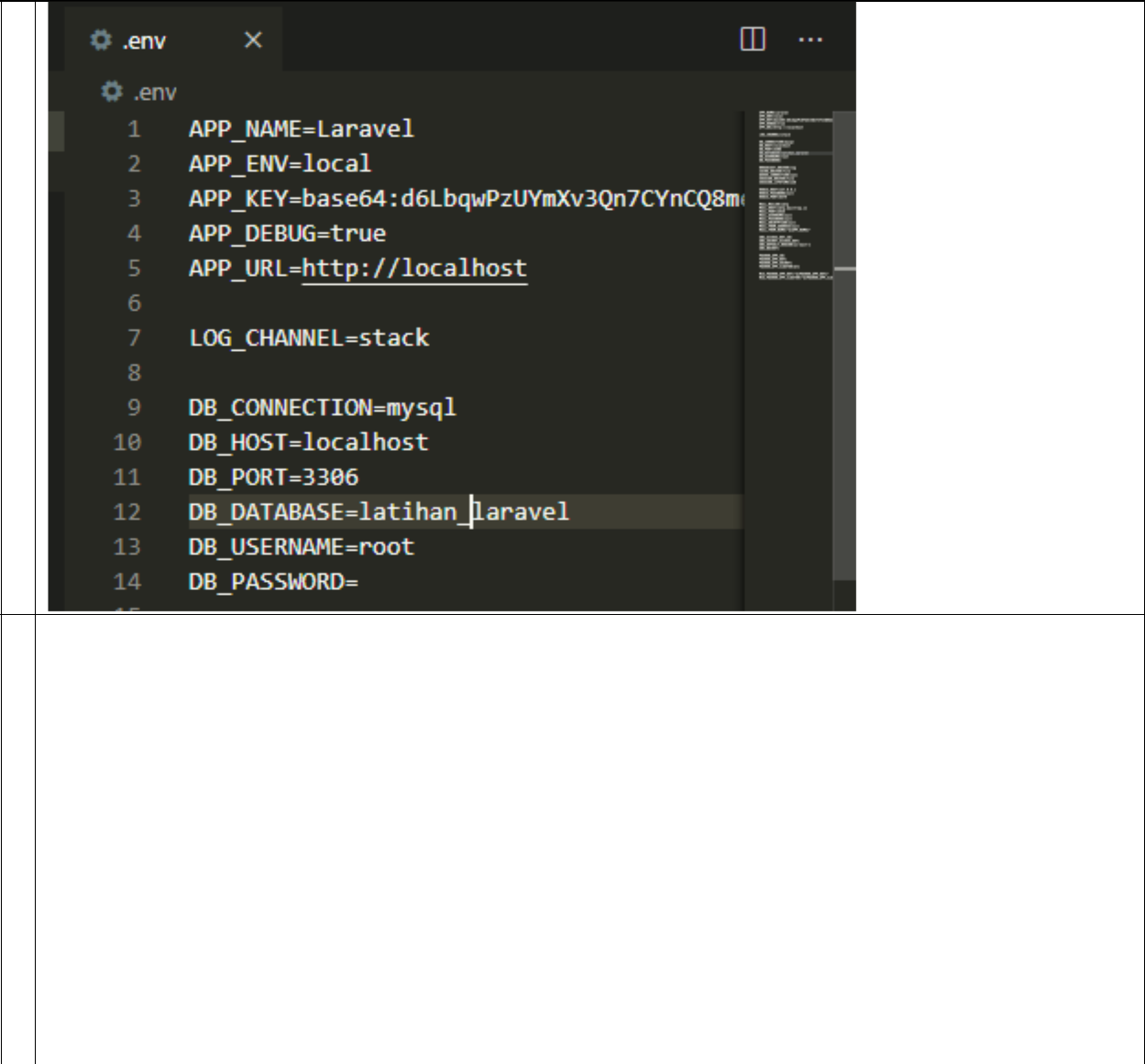


**Praktikum – Bagian 2: Membuat CRUD di Laravel menggunakan Eloquent**

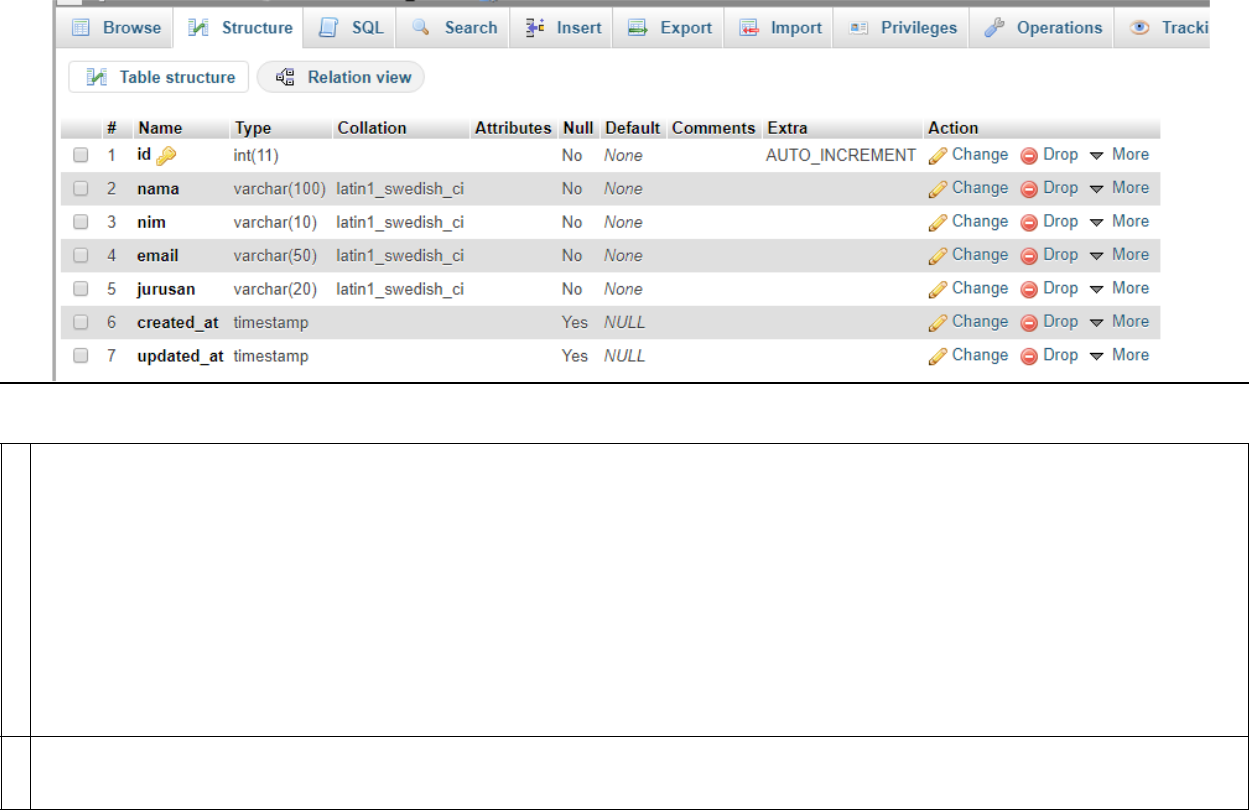
Buatlah project Laravel baru dengan nama laravel-crud-kedua. Buka command prompt, tuliskan perintah berikut. cd C:\xampp\htdocs laravel new laravel-crud-kedua



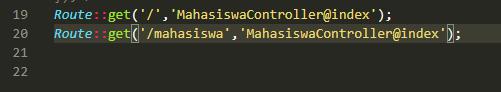
Selanjutnya kita lakukan konfigurasi database di Laravel. Untuk melakukan konfigurasi database, bukalah file .env pada project laravel-crud. Ubah seperti di bawah ini.



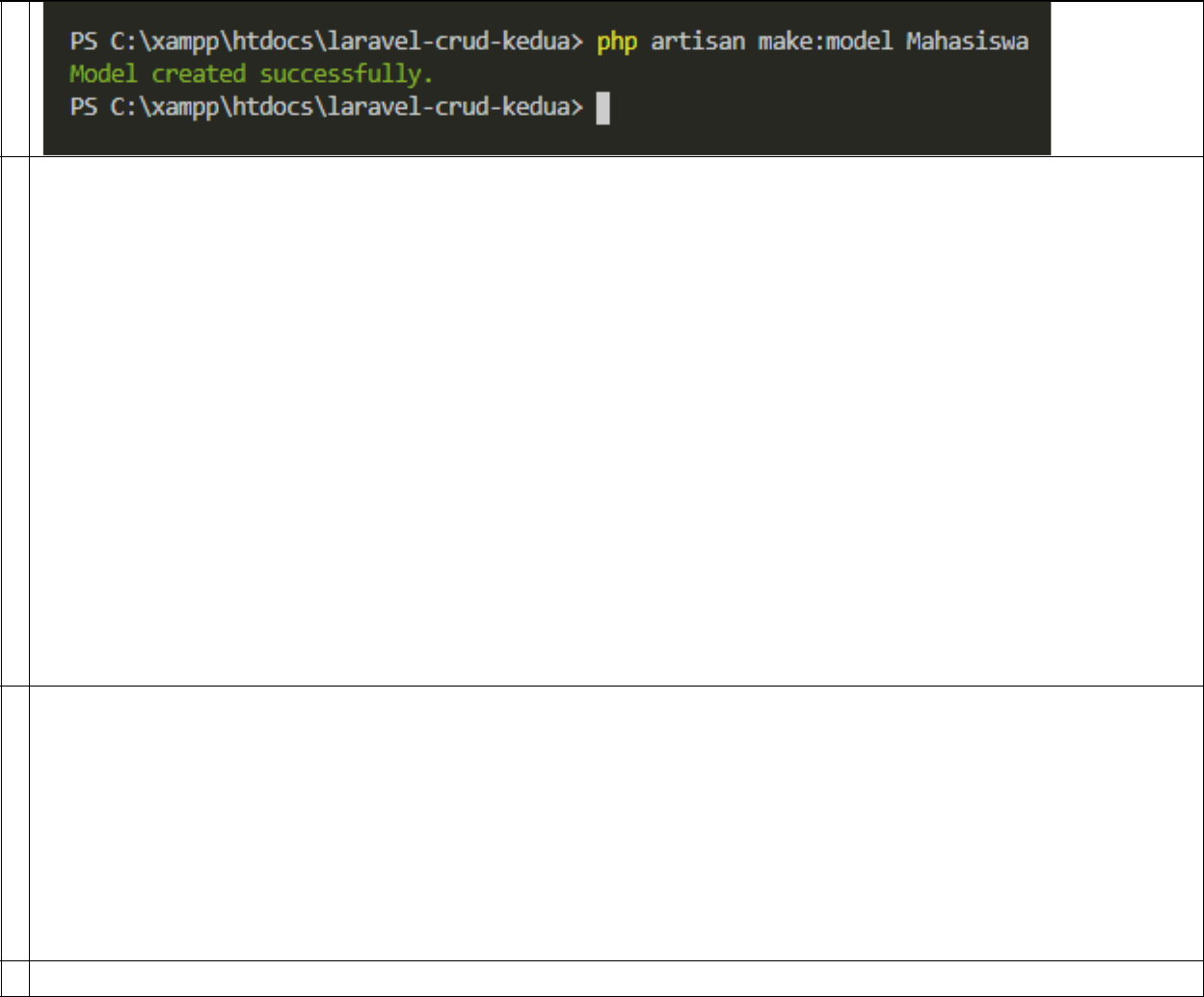
Pada project ini kita gunakan database dan tabel yang sebelumnya digunakan pada Praktikum Bagian 1. Tambahkan kolom created\_at dan updated\_at yang bertipe TIMESTAMP dan default nilainya NULL



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.



Buat model menggunakan command prompt dengan nama Mahasiswa menggunakan php artisan cd laravel-crud-kedua php artisan make:model Mahasiswa

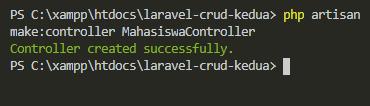


Ubah model Mahasiswa.php pada folder App menjadi seperti berikut.

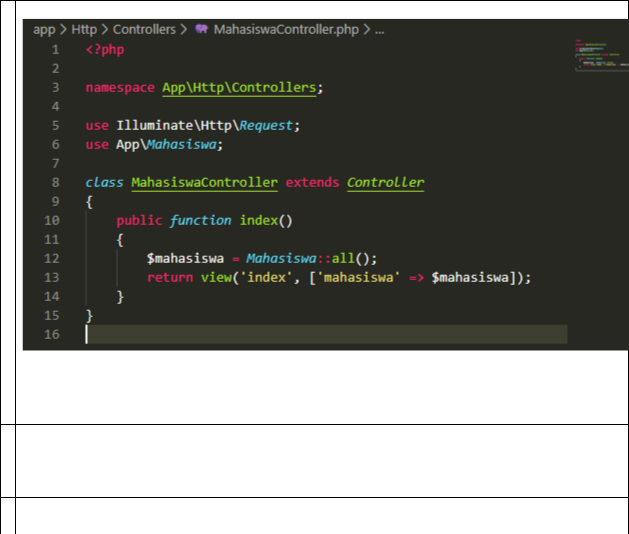


Keterangan: - Model Mahasiswa akan menangani tabel mahasiswa

Buat controller baru yaitu MahasiswaController menggunakan php artisan php artisan make:controller MahasiswaController



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

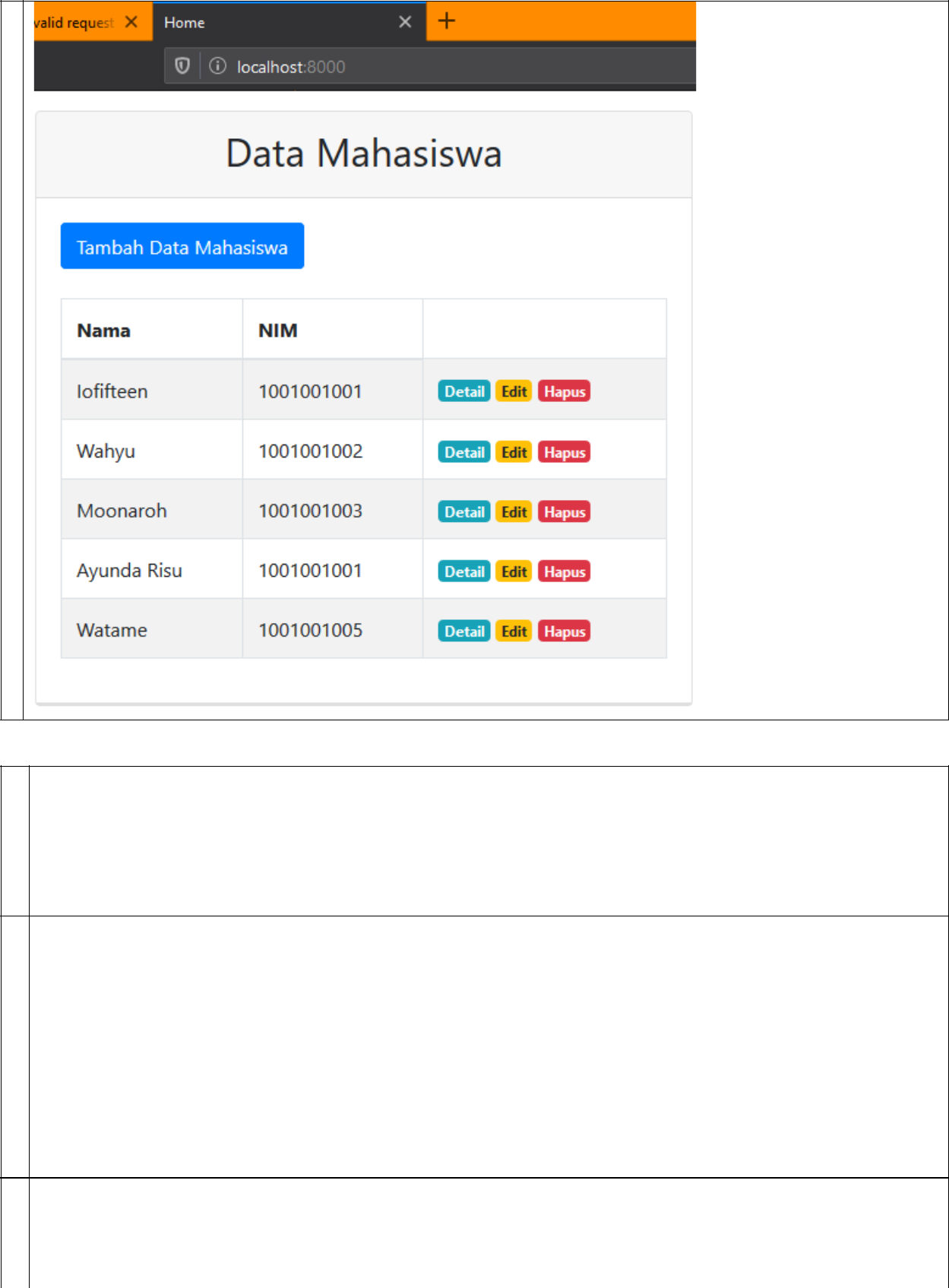


Keterangan:

* 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

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)

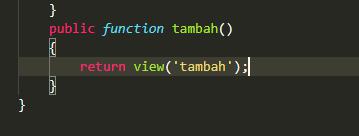
Jalankan command prompt, tuliskan perintah untuk menjalankan project laravel-crud php artisan serve Buka browser dan ketikkan localhost:8000, maka akan tampil sebagai berikut



Buatlah route baru pada routes/web.php dengan nama /mahasiswa/tambah yang akan menjalankan fungsi tambah pada MahasiswaController ketika tombol Tambah Data Mahasiswa ditekan.



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



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

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



Keterangan: - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method simpan di MahasiswaController.php

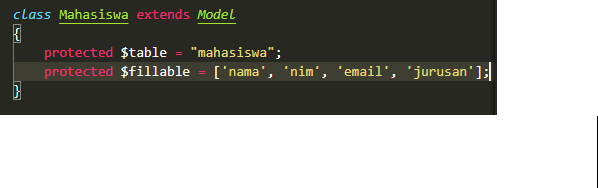
Buat method simpan pada MahasiswaController untuk menyimpan data ke database.



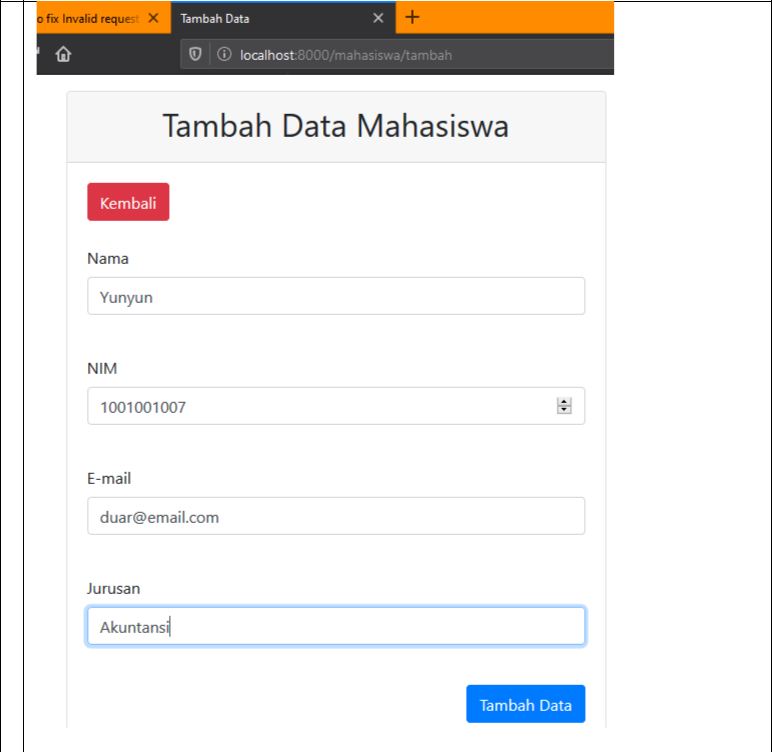
Keterangan: - Variabel $request untuk menerima data yang akan ditambahkan ke database

* Line 23-28 merupakan fitur eloquent menggunakan fungsi create() untuk insert data ke tabel mahasiswa

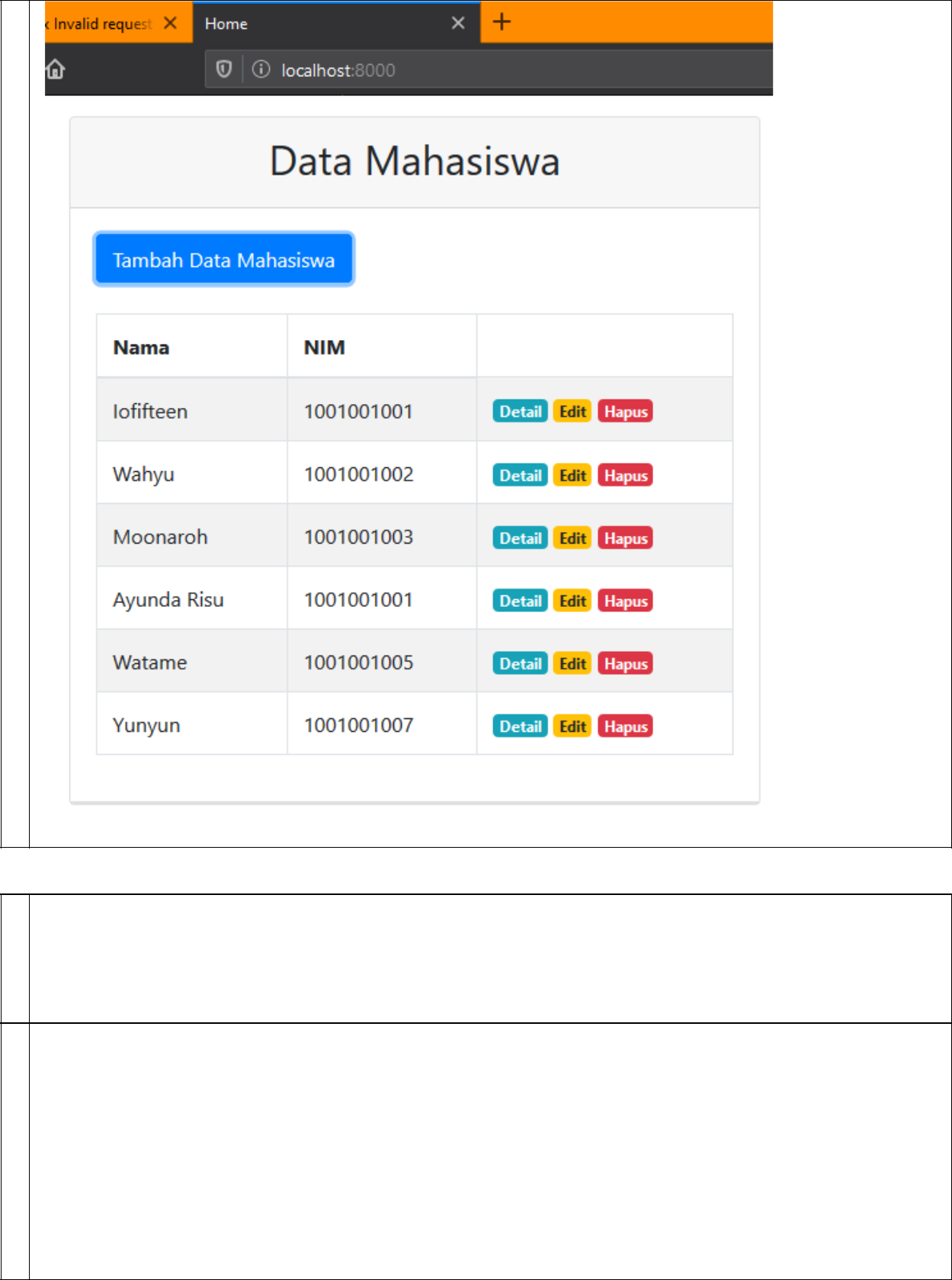
Karena kita menggunakan fungsi create pada Controller,maka butuh ditambahkan code pada Line 10 yang disebut Mass Assignment pada model Mahasiswa.php. Mass Assignment digunakan untuk memfilter kolom mana yang boleh dan tidak boleh diinput.



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.



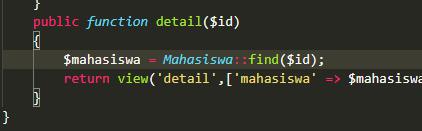
Setelah kita klik tombol tambah data, maka hasilnya sebagai berikut.



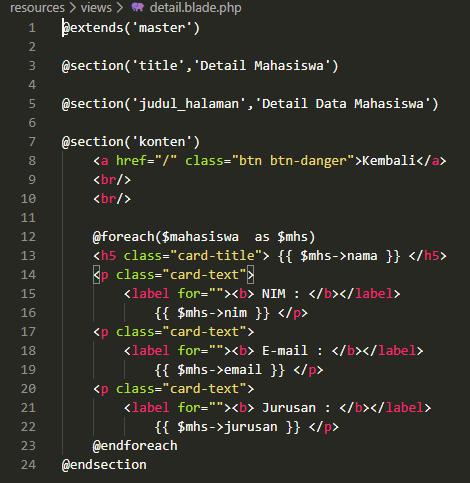
Buatlah route baru pada routes/web.php dengan nama /mahasiswa/detail yang akan menjalankan fungsi detail pada MahasiswaController



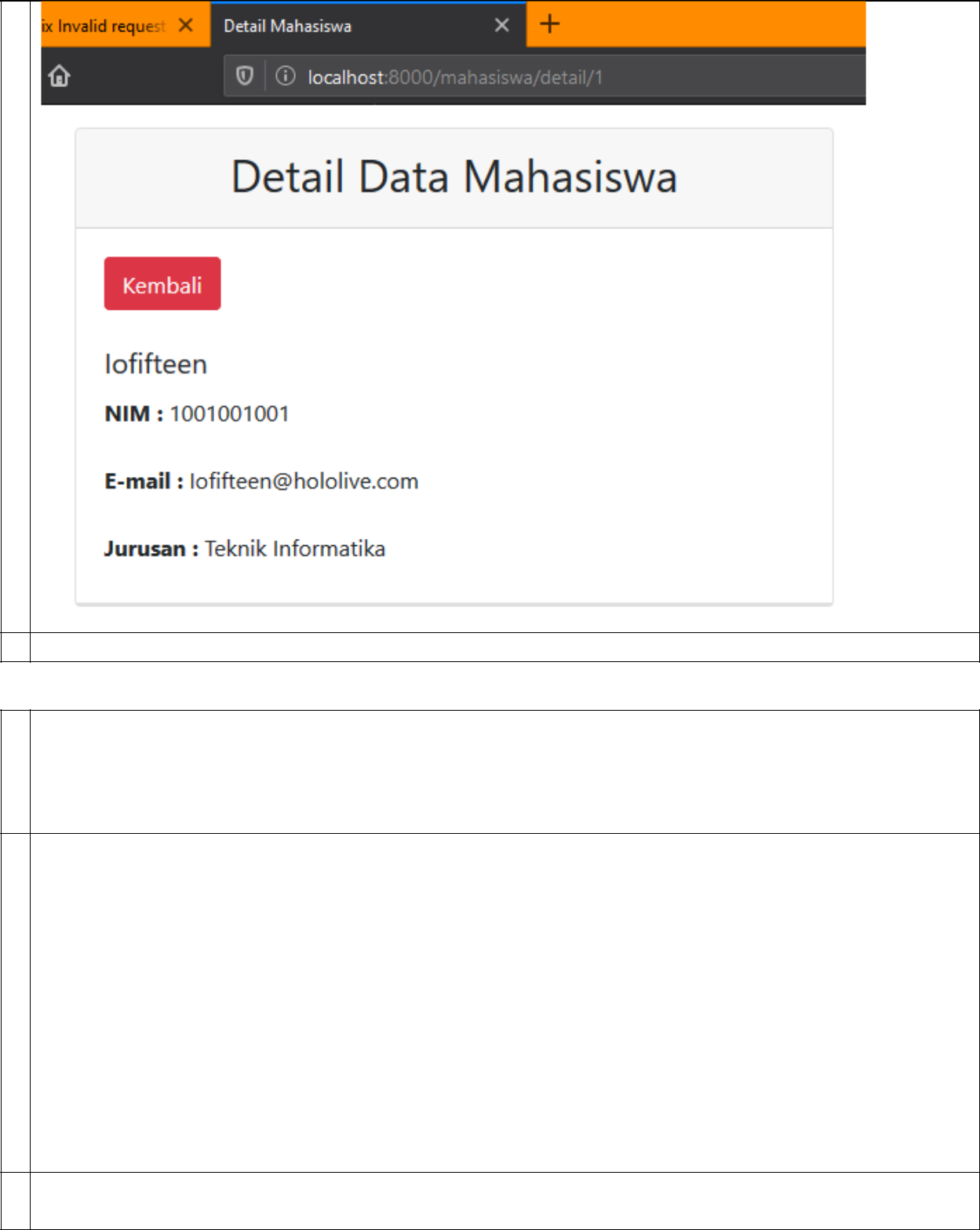
Buat method detail pada MahasiswaController.php di folder app/Http/Controllers yang akan menampilkan data mahasiswa pada view detail.blade.php.



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



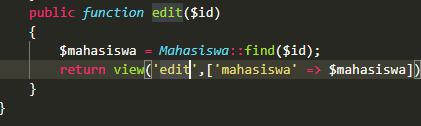
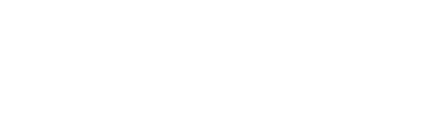
Jalankan localhost:8000 dan pilih tombol ‘Detail’ di suatu data yang ingin kita lihat detailnya.



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



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



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

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

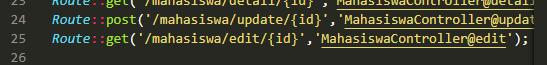


Keterangan: - Line 11-31 merupakan form untuk memasukkan data mahasiswa berupa nama, nim, email, dan jurusan

* Line 11 terdapat action=”/mahasiswa/update/{{$mahasiswa->id)}}” yang menunjukkan routes

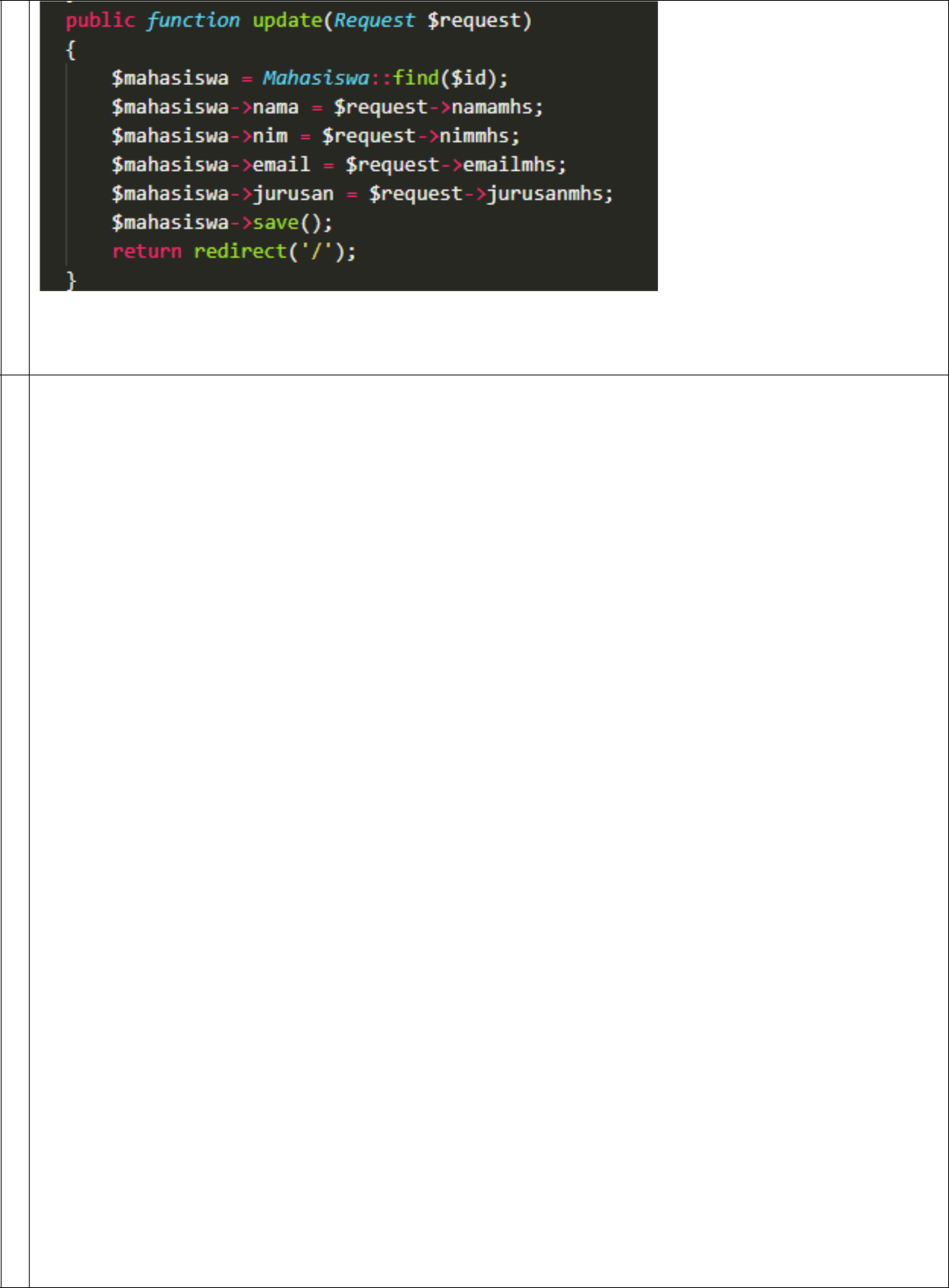
/mahasiswa/update/{id} dimana data pada form tersebut akan dikirimkan ke fungsi update pada controller MahasiswaController

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



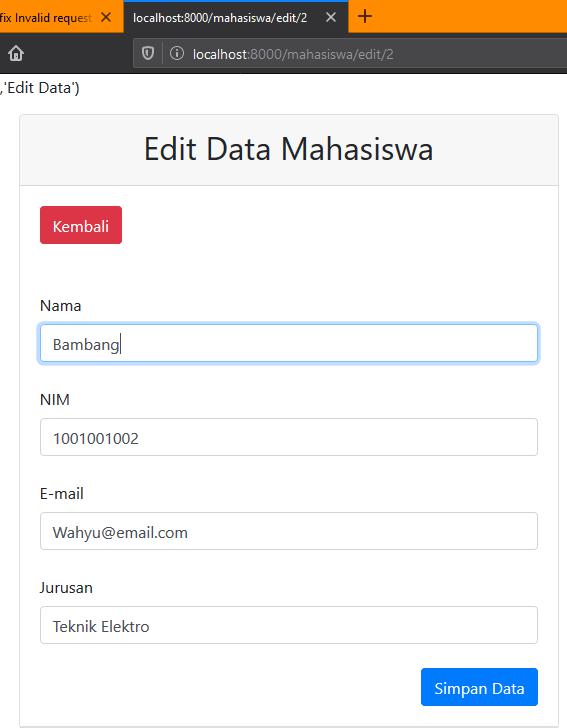
Keterangan: - Pada route ini menggunakan metode post karena data mahasiswa dari form akan dikirim ke method update di MahasiswaController.php

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



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

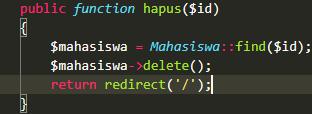
Jalankan localhost:8000 dan pilih tombol ‘Edit’, maka akan ditampilkan halaman edit yang berisi form untuk mengubah data baru. Cobalah untuk mengedit data.



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



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



Keterangan : - Line 59 :fungsi eloquent untuk menghapus data mahasiswa berdasarkan id yang dipilih

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

