LAPORAN PRAKTIKUM

PEMROGRAMAN WEB LANJUT

Jobsheet 14 (Rest API)

-

Oleh :

Muhammad Royyan Zamzami

NIM. 1841720053

PROGRAM STUDI TEKNIK INFORMATIKA

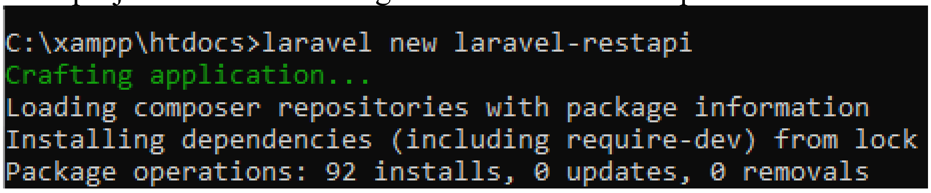
JURUSSAN TEKNOLOGI INFORMASI

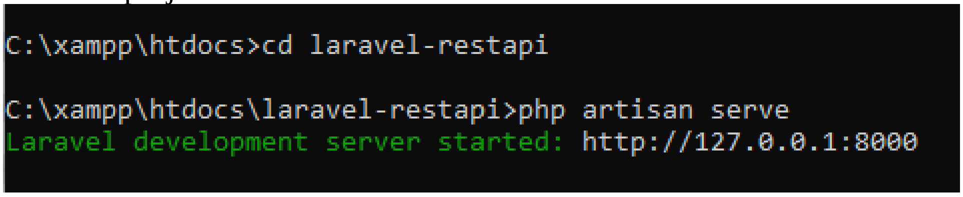
POLITEKNIK NEGERI MALANG

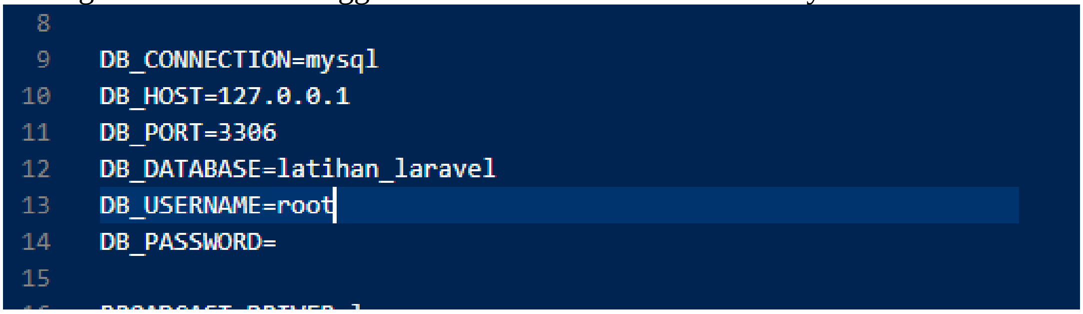
MEI 2020

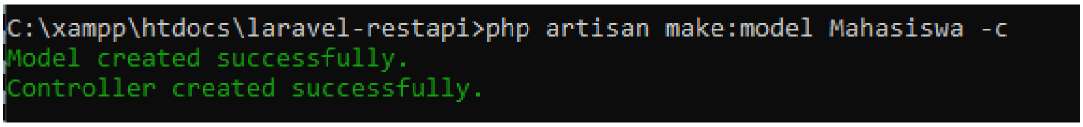
Praktikum: Membuat RESTful API di Laravel

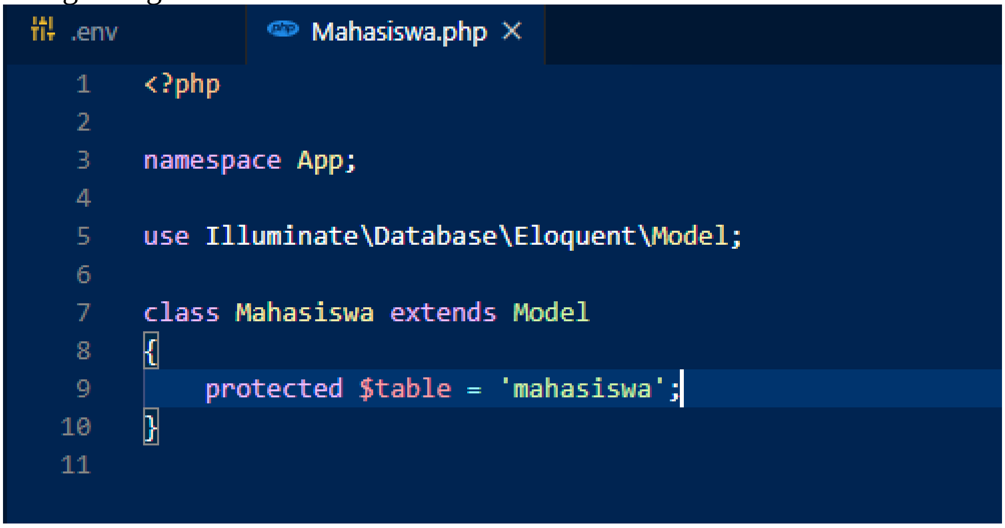
No	Keterangan

1	Buat project laravel baru dengan nama “laravel-restapi”
	
	`C:\xampp\htdocs>laravel new laravel-restapi`
	`Crafting application...`
	`Loading composer repositories with package information`
	`Installing dependencies (including require-dev) from lock`
	`Package operations: 92 installs, 0 updates, 0 removals`

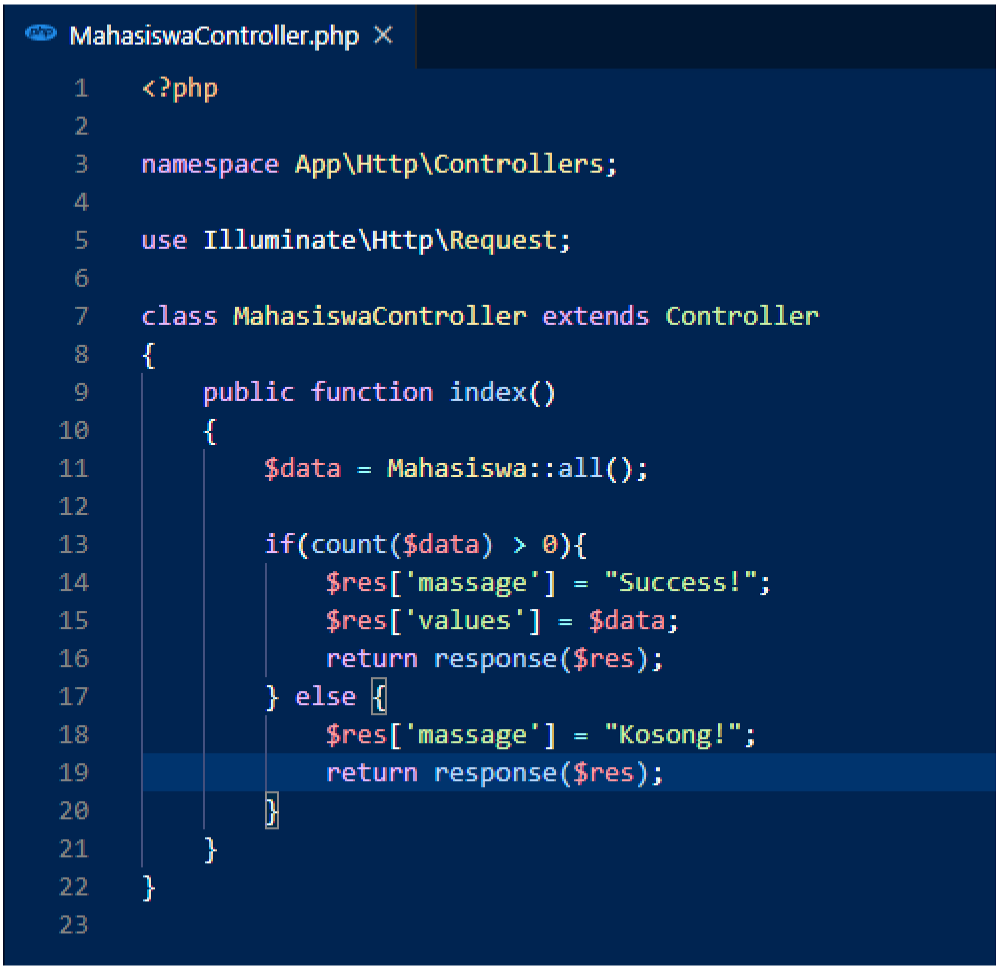
2	Jalankan project
	
	`C:\xampp\htdocs>cd laravel-restapi`
	
	`C:\xampp\htdocs\laravel-restapi>php artisan serve`
	`Laravel development server started: http://127.0.0.1:8000`

3	Konfigurasi database menggunakan database latihan sebelumnya
	
	`DB_CONNECTION=mysql`
	`DB_HOST=127.0.0.1`
	`DB_PORT=3306`
	`DB_DATABASE=latihan_laravel`
	`DB_USERNAME=root`
	`DB_PASSWORD=`

4	baut Model dan Controller Mahasiswa
	
	`C:\xampp\htdocs\laravel-restapi>php artisan make:model Mahasiswa -c`
	`Model created successfully.`
	`Controller created successfully.`

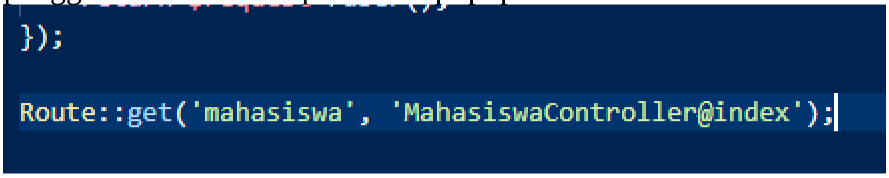
5	Mengkonfigurasi model Mahasiswa
	
	
	`<?php`
	
	`namespace App;`
	
	`use Illuminate\Database\Eloquent\Model;`
	
	`class Mahasiswa extends Model`
	`{`
	`    protected $table = 'mahasiswa';`
	`}`

6	Konfigurasi Controller



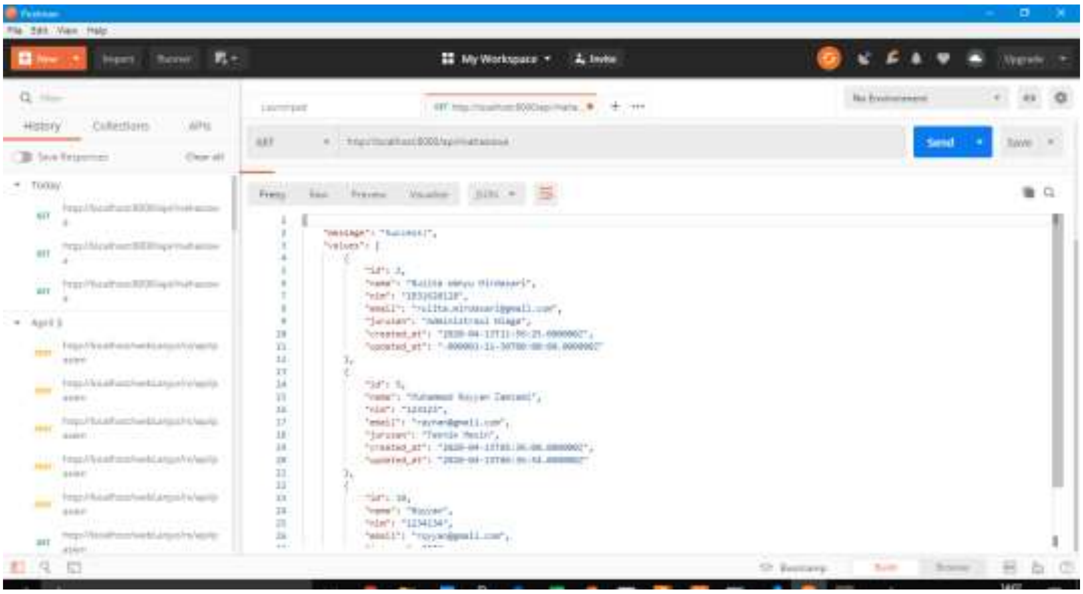
```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class MahasiswaController extends Controller
{
    public function index()
    {
        $data = Mahasiswa::all();

        if(count($data) > 0){
            $res['massage'] = "Success!";
            $res['values'] = $data;
            return response($res);
        } else {
            $res['massage'] = "Kosong!";
            return response($res);
        }
    }
}

```

7 panggil function index pada file api.php



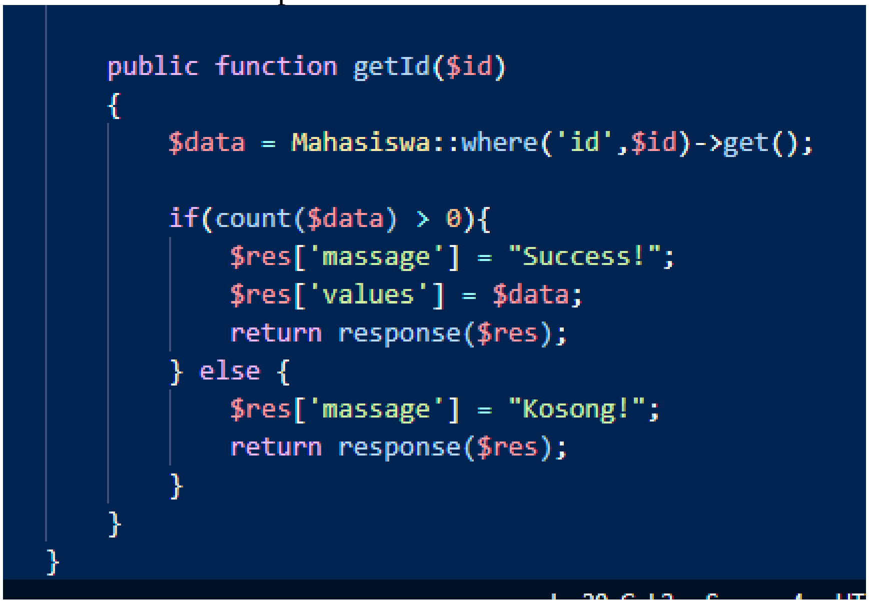
```php
});

Route::get('mahasiswa', 'MahasiswaController@index');
```

8 Jalankan Php artisan serve dan lakukan testing menggunakan postman



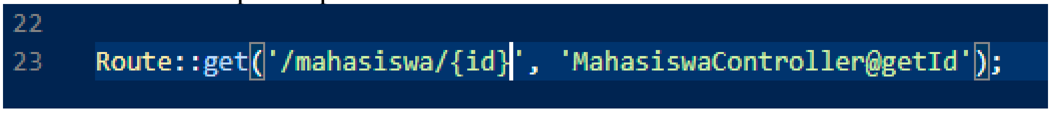
```
{
    "massage": "Success!",
    "values": [
        {
            "id": 3,
            "name": "Rulita Wahyu Wirdasari",
            "nim": "1853620128",
            "email": "rulita.wirdasari@gmail.com",
            "jurusan": "Administrasi Niaga",
            "created_at": "2020-04-13T11:56:21.000000Z",
            "updated_at": "-000001-11-30T00:00:00.000000Z"
        },
        {
            "id": 5,
            "name": "Muhammad Rayyan Zamzami",
            "nim": "123123",
            "email": "rayyan@gmail.com",
            "jurusan": "Teknik Mesin",
            "created_at": "2020-04-13T01:36:08.000000Z",
            "updated_at": "2020-04-13T08:06:42.000000Z"
        },
        {
            "id": 10,
            "name": "Rayyan",
            "nim": "1234134",
            "email": "rayyan@gmail.com",
```

9 Tambahkan function pada controller



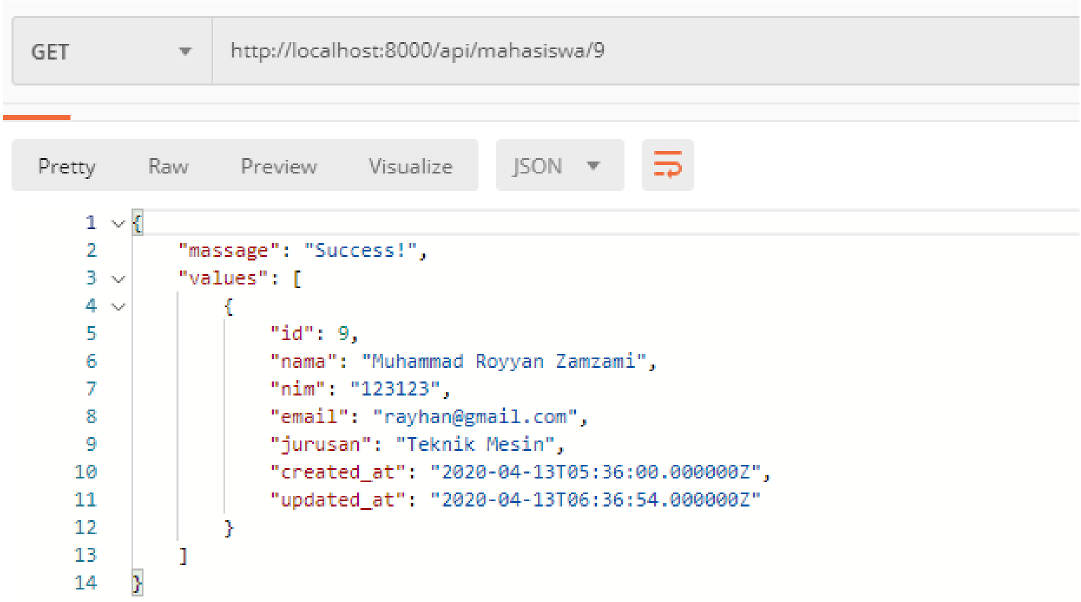
```
    public function getId($id)
    {
        $data = Mahasiswa::where('id',$id)->get();

        if(count($data) > 0){
            $res['massage'] = "Success!";
            $res['values'] = $data;
            return response($res);
        } else {
            $res['massage'] = "Kosong!";
            return response($res);
        }
    }
}
```

10 Tambahkan route pada api



```

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

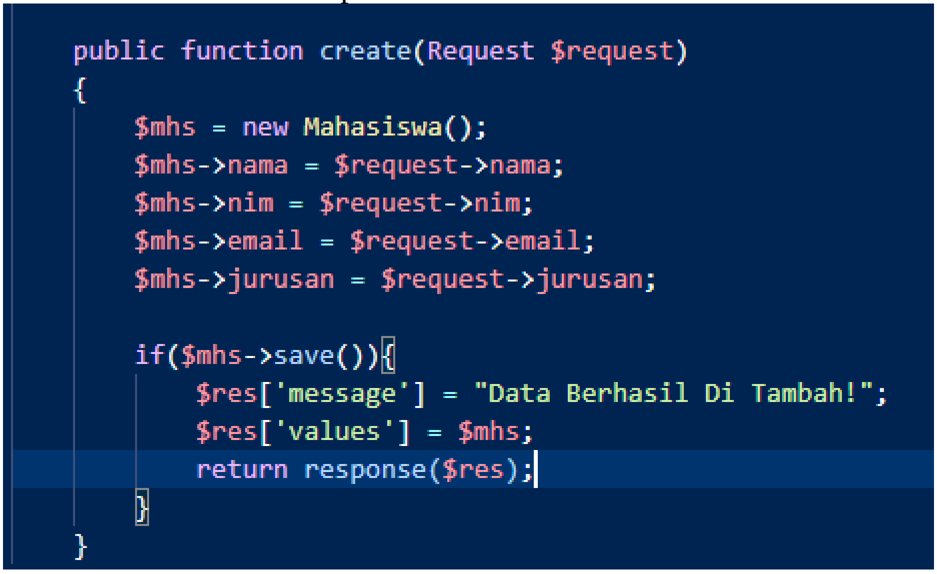
11 Testing di postman “get berdasarkan id”



http://localhost:8000/api/mahasiswa/9

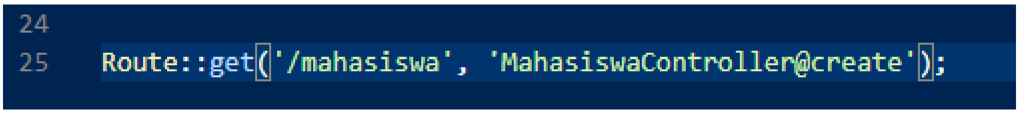
```
{
    "massage": "Success!",
    "values": [
        {
            "id": 9,
            "nama": "Muhammad Royyan Zamzami",
            "nim": "123123",
            "email": "rayhan@gmail.com",
            "jurusan": "Teknik Mesin",
            "created_at": "2020-04-13T05:36:00.000000Z",
            "updated_at": "2020-04-13T06:36:54.000000Z"
        }
    ]
}
```

12 Membuat function Create pada Controller



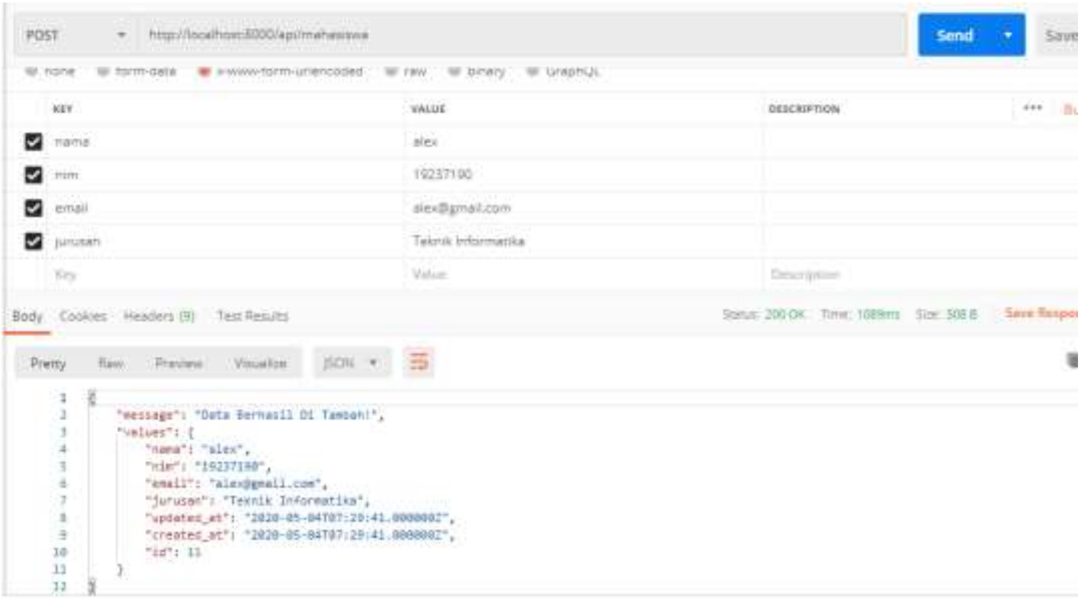
```
    public function create(Request $request)
    {
        $mhs = new Mahasiswa();
        $mhs->nama = $request->nama;
        $mhs->nim = $request->nim;
        $mhs->email = $request->email;
        $mhs->jurusan = $request->jurusan;

        if($mhs->save()){
            $res['message'] = "Data Berhasil Di Tambah!";
            $res['values'] = $mhs;
            return response($res);
        }
    }
```

13 Tambahkan Route create



```

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

14 Coba di postman



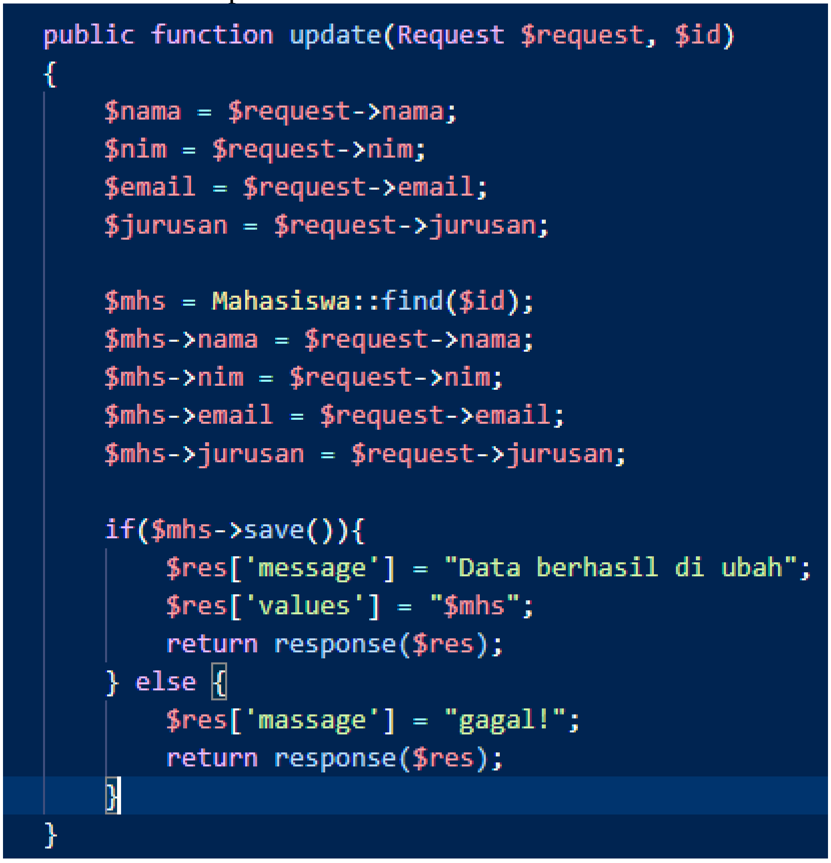
http://localhost:8000/api/mahasiswa

	KEY	VALUE	DESCRIPTION
☑	nama	alex	
☑	nim	19237190	
☑	email	alex@gmail.com	
☑	jurusan	Teknik Informatika	
	Key	Value	Description

Status: 200 OK Time: 1089ms Size: 508 B

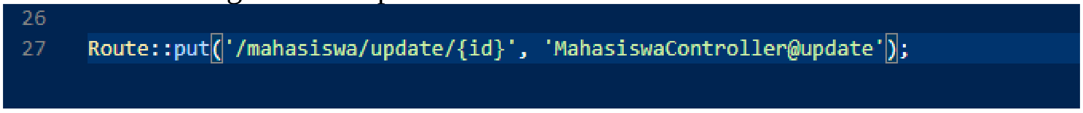
```
{
    "message": "Data Berhasil Di Tambah!",
    "values": {
        "nama": "alex",
        "nim": "19237190",
        "email": "alex@gmail.com",
        "jurusan": "Teknik Informatika",
        "updated_at": "2020-05-04T07:29:41.000000Z",
        "created_at": "2020-05-04T07:29:41.000000Z",
        "id": 11
    }
}
```

15 Buat Controller update



```
public function update(Request $request, $id)
{
    $nama = $request->nama;
    $nim = $request->nim;
    $email = $request->email;
    $jurusan = $request->jurusan;

    $mhs = Mahasiswa::find($id);
    $mhs->nama = $request->nama;
    $mhs->nim = $request->nim;
    $mhs->email = $request->email;
    $mhs->jurusan = $request->jurusan;

    if($mhs->save()){
        $res['message'] = "Data berhasil di ubah";
        $res['values'] = "$mhs";
        return response($res);
    } else {
        $res['massage'] = "gagal!";
        return response($res);
    }
}
```

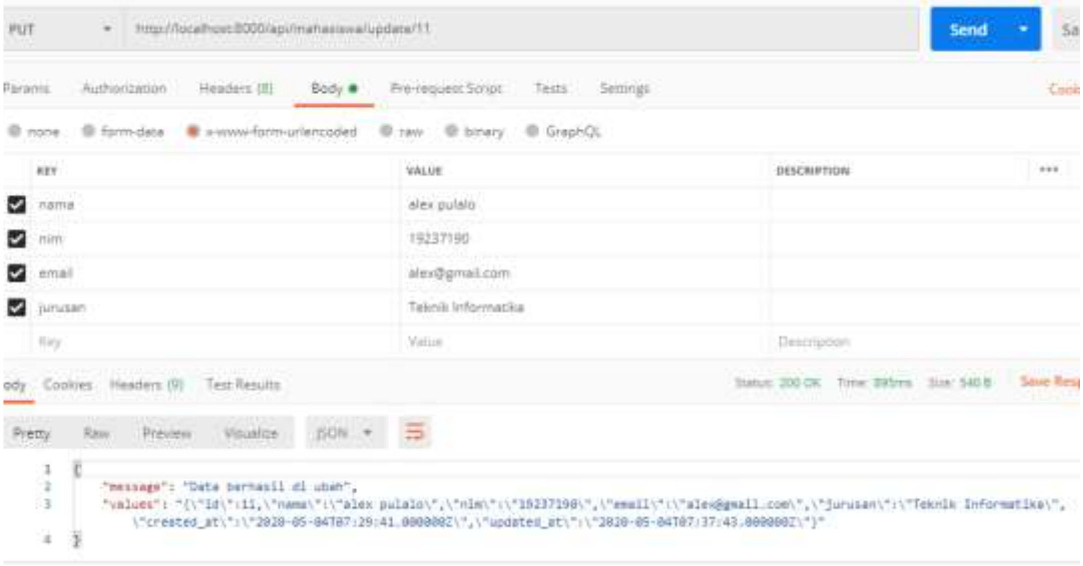
16 Tambahkan di bagian route api



```
Route::put('/mahasiswa/update/{id}', 'MahasiswaController@update');
```

17 Testing di postman

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



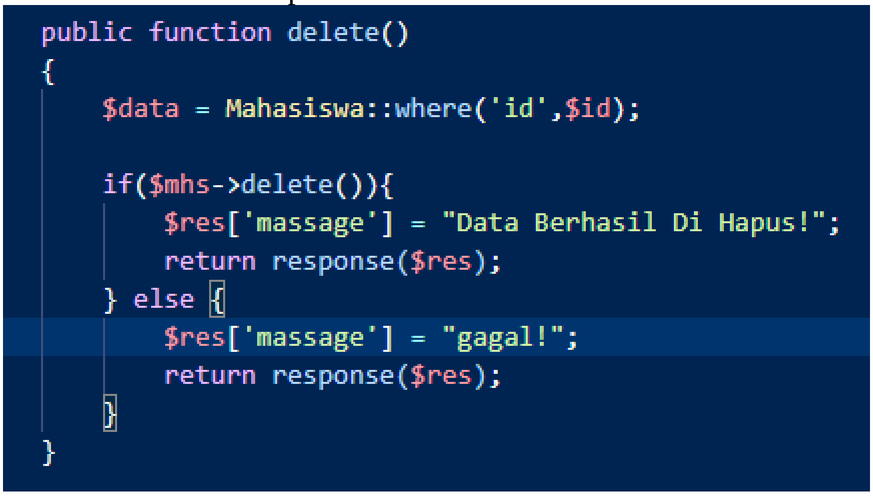
PUT http://localhost:8000/api/mahasiswa/update/11

	KEY	VALUE	DESCRIPTION
☑	nama	alex pulalo	
☑	nim	19237190	
☑	email	alex@gmail.com	
☑	jurusan	Teknik Informatika	
	Key	Value	Description

```
{
    "message": "Data berhasil di ubah",
    "values": "{\"id\":11,\"nama\":\"alex pulalo\",\"nim\":\"19237190\",\"email\":\"alex@gmail.com\",\"jurusan\":\"Teknik Informatika\",
        \"created_at\":\"2020-05-04T07:29:41.000000Z\",\"updated_at\":\"2020-05-04T07:37:43.000000Z\"}"
}
```

</td>
</tr>
<tr>
<td>18</td>
<td>

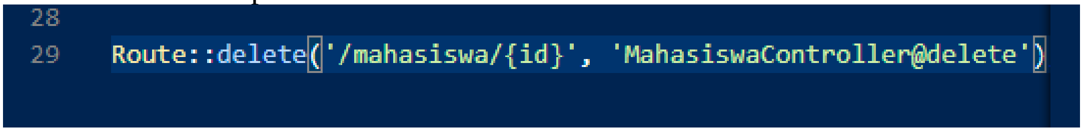
Butat function Delete pada controller



```
public function delete()
{
    $data = Mahasiswa::where('id',$id);

    if($mhs->delete()){
        $res['massage'] = "Data Berhasil Di Hapus!";
        return response($res);
    } else {
        $res['massage'] = "gagal!";
        return response($res);
    }
}
```

</td>
</tr>
<tr>
<td>19</td>
<td>

Tambahkan route api delete

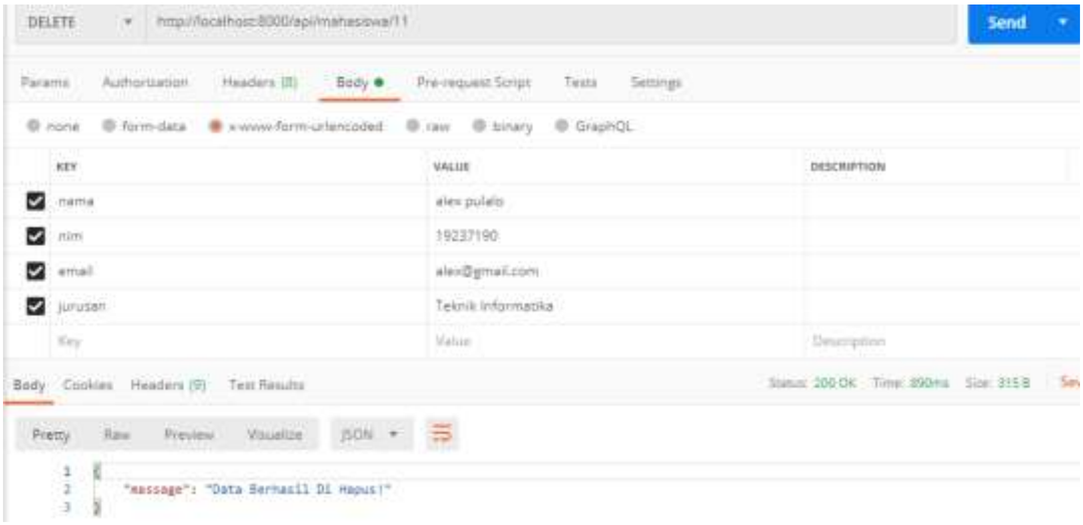


```

Route::delete('/mahasiswa/{id}', 'MahasiswaController@delete')
```

</td>
</tr>
<tr>
<td>20</td>
<td>

Testing di Postman

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

DELETE http://localhost:8000/api/mahasiswa/11

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	nama	alex pulalo	
☑	nim	19237190	
☑	email	alex@gmail.com	
☑	jurusan	Teknik Informatika	
	Key	Value	Description

Status: 200 OK Time: 890ms Size: 315 B

```
{
    "message": "Data Berhasil Di Hapus!"
}
```