

LAPORAN TUGAS PEMROGRAMAN WEB LANJUT

“JOBSHEET 13”



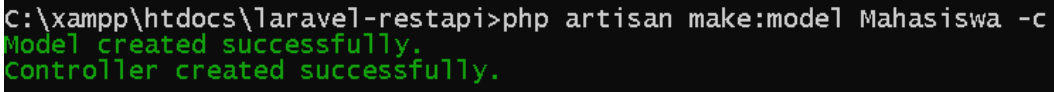
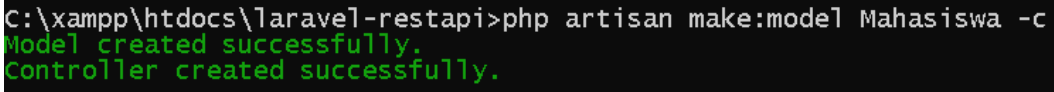
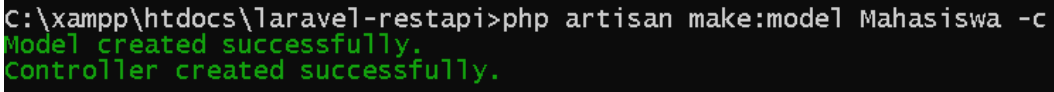
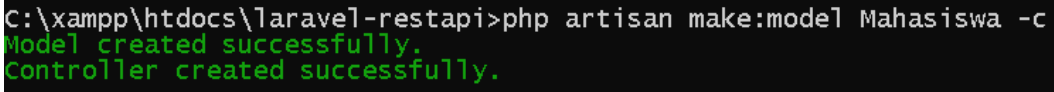
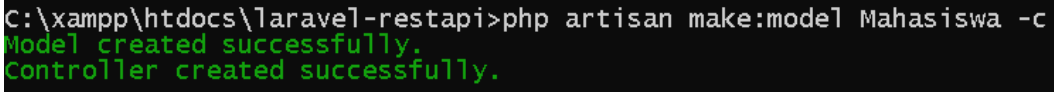
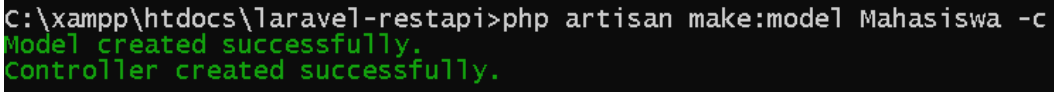
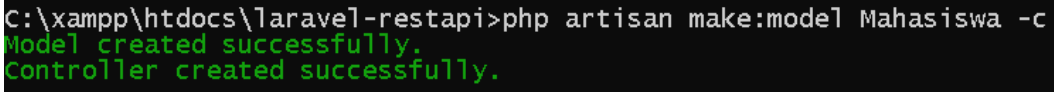
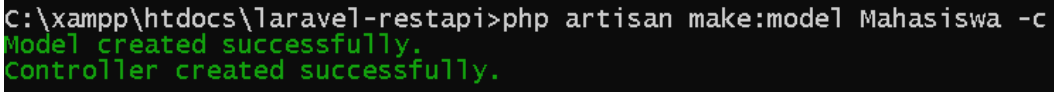
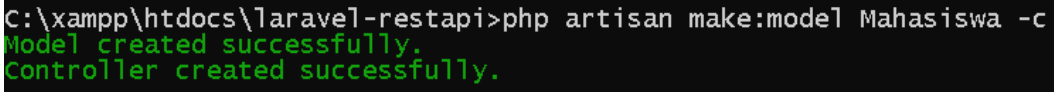
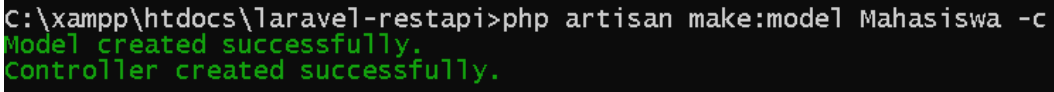
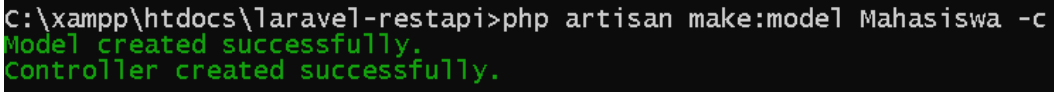
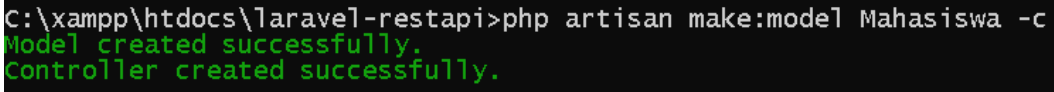
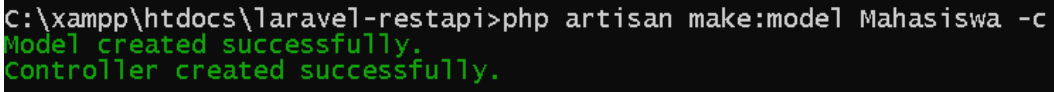
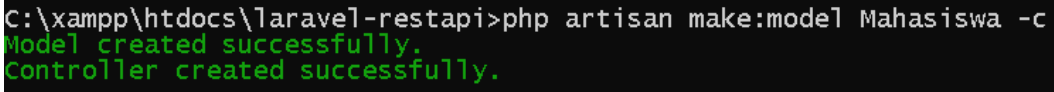
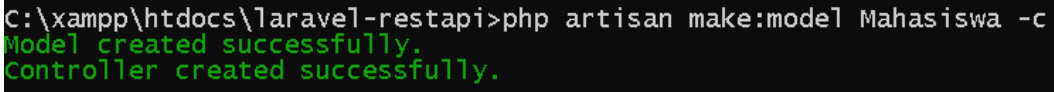
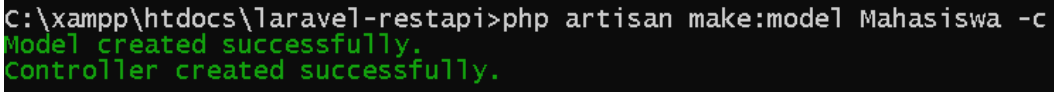
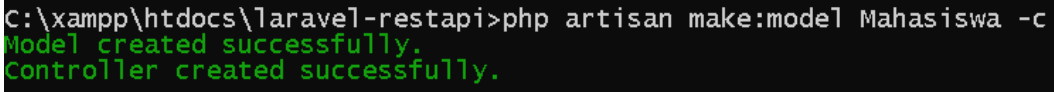
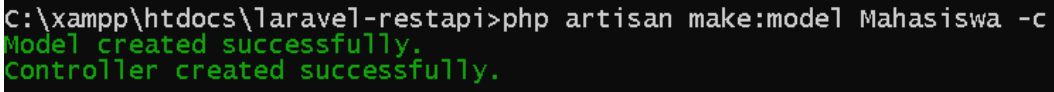
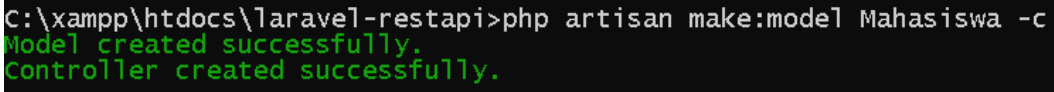
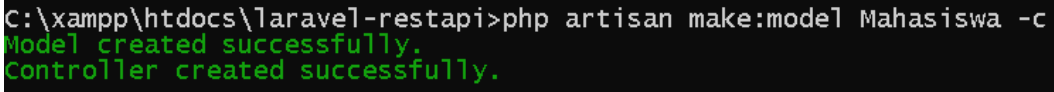
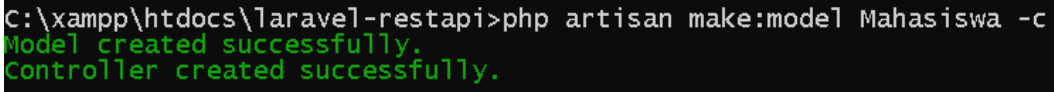
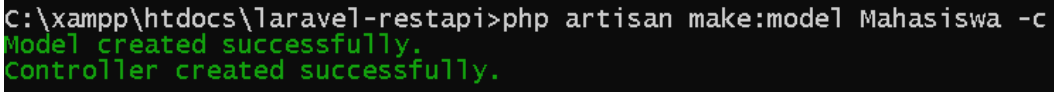
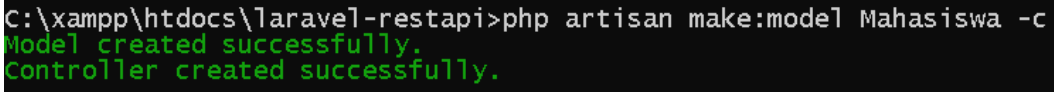
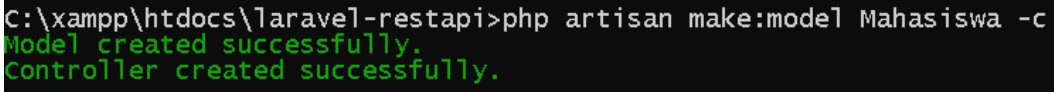
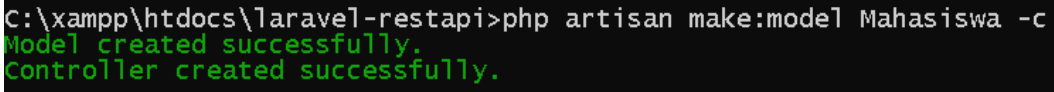
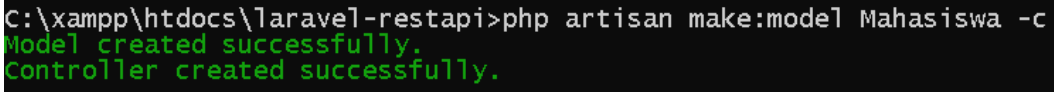
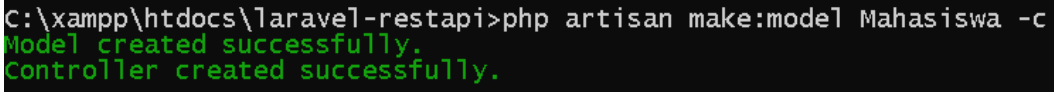
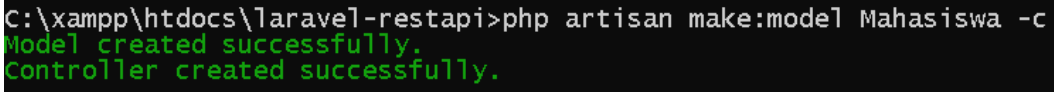
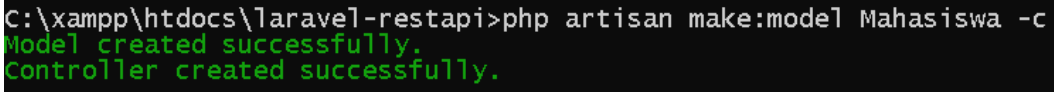
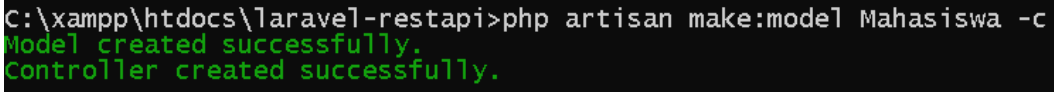
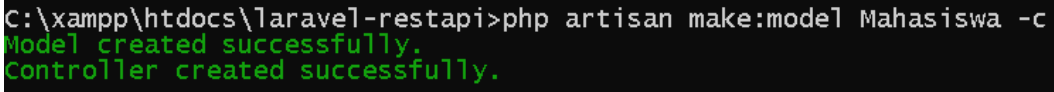
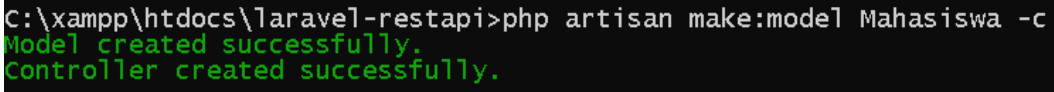
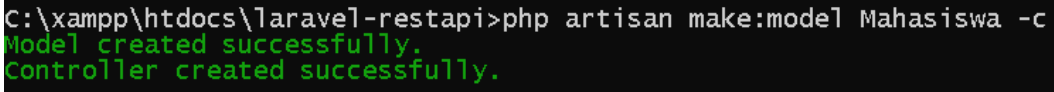
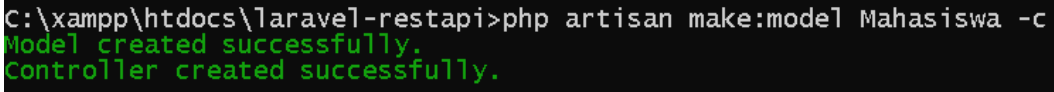
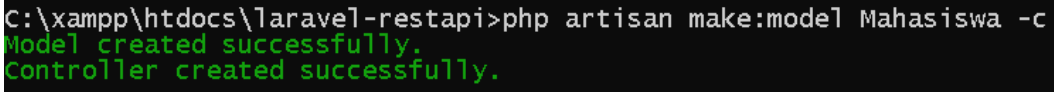
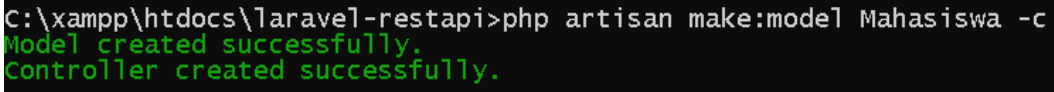
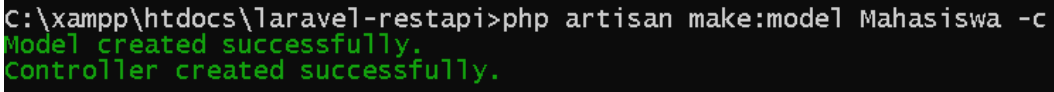
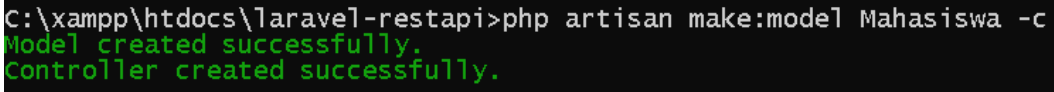
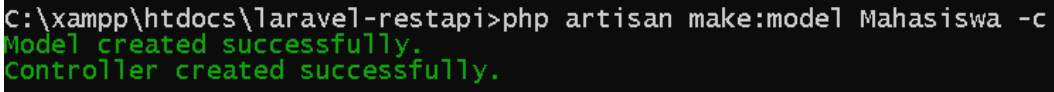
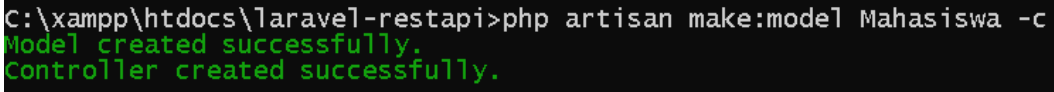
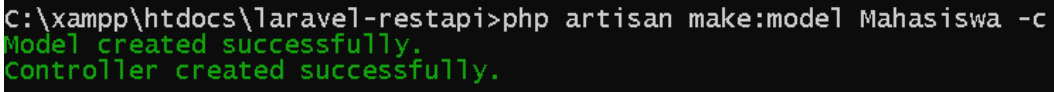
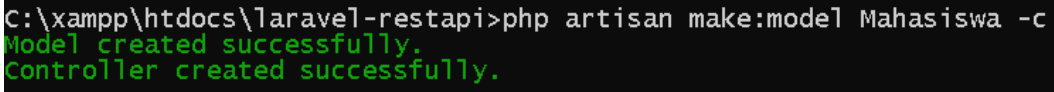
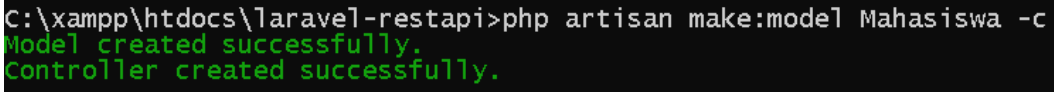
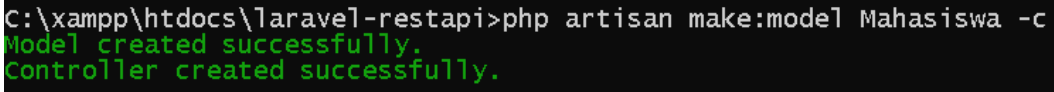
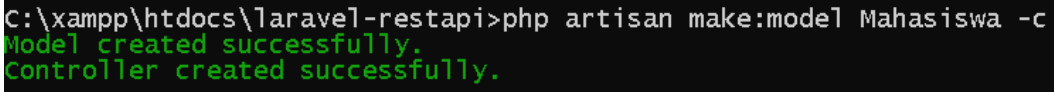
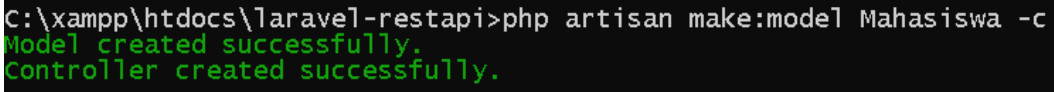
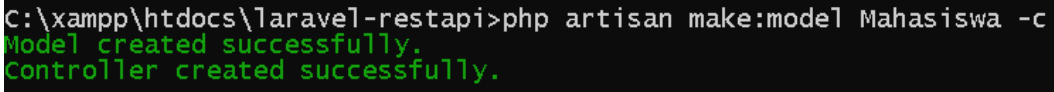
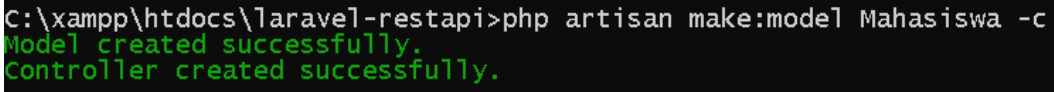
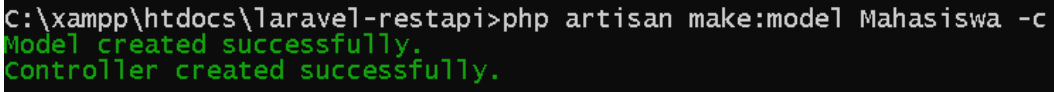
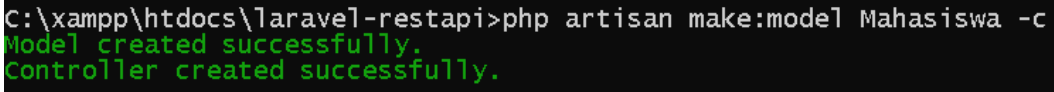
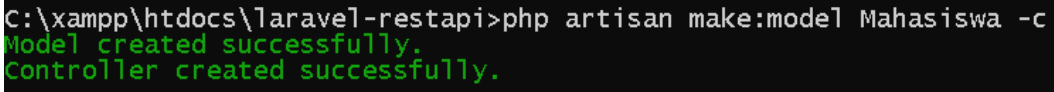
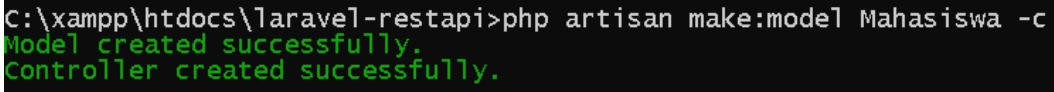
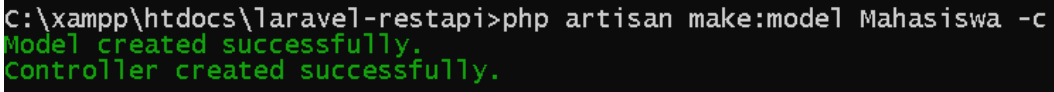
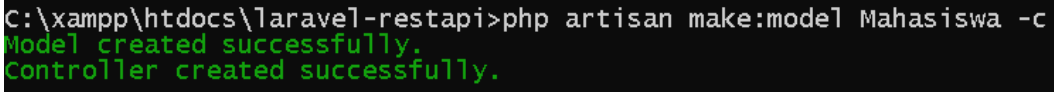
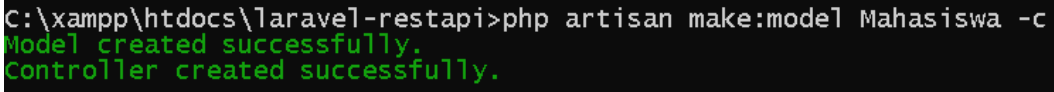
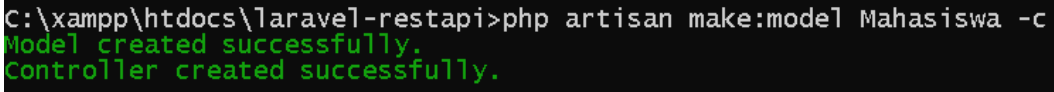
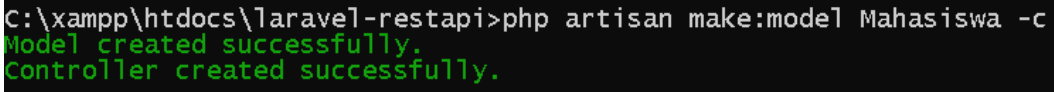
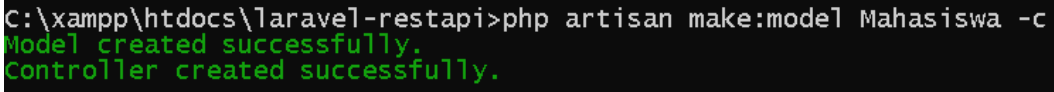
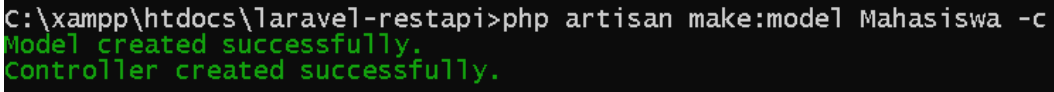
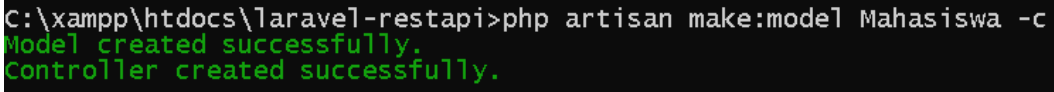
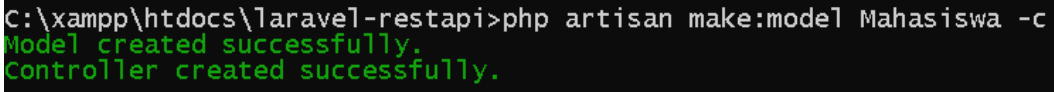
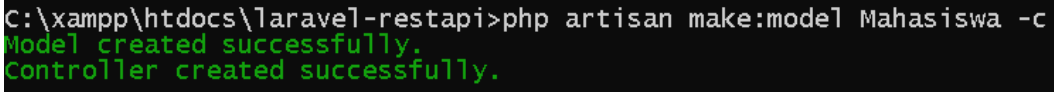
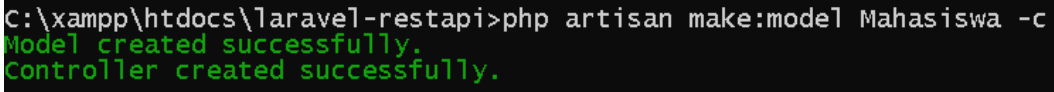
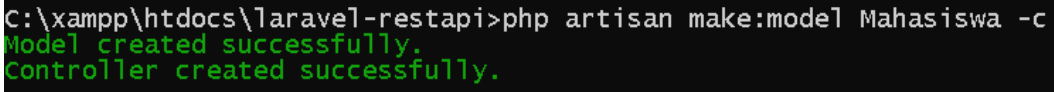
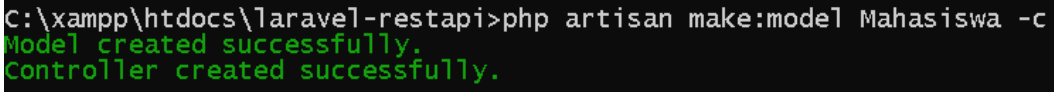
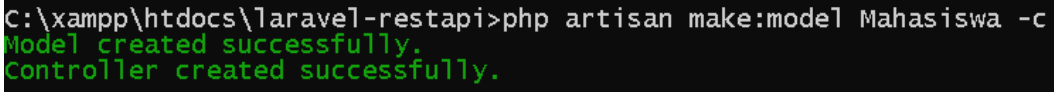
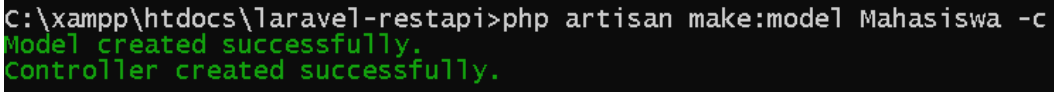
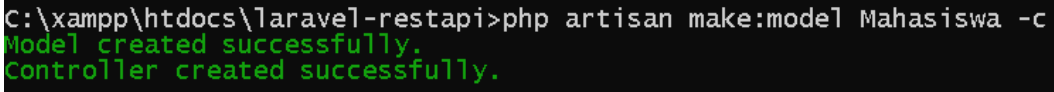
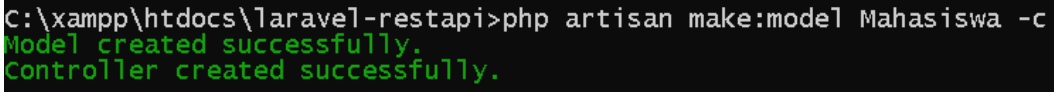
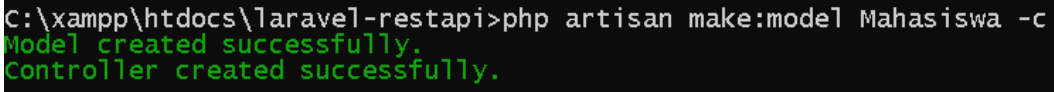
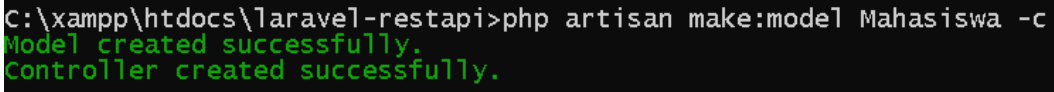
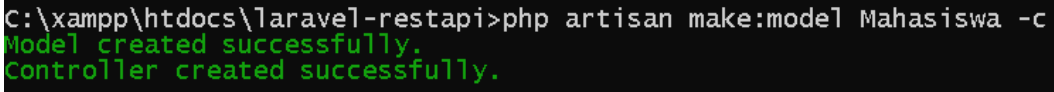
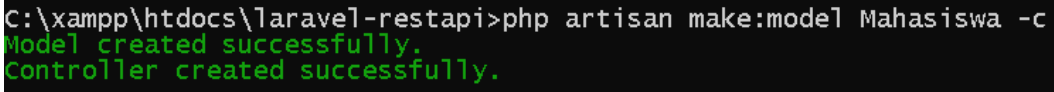
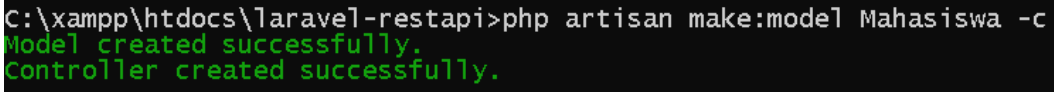
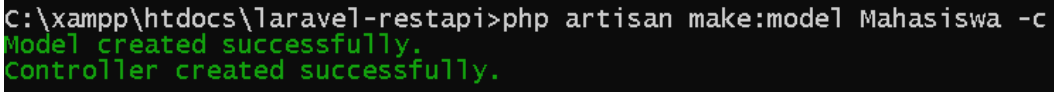
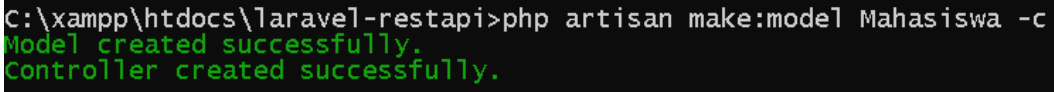
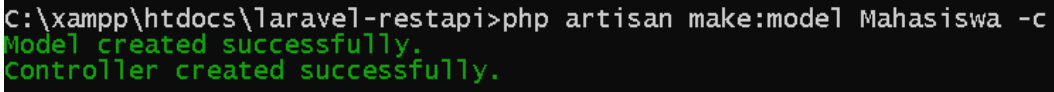
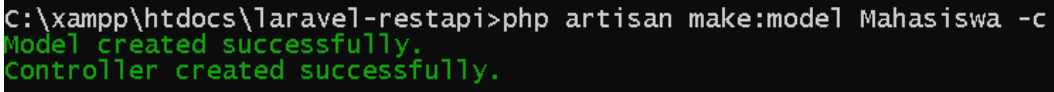
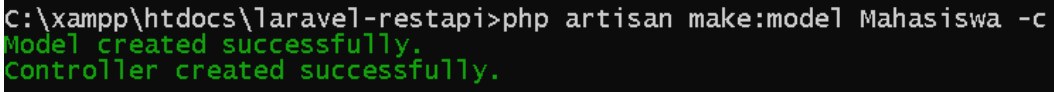
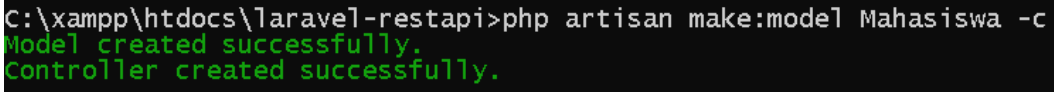
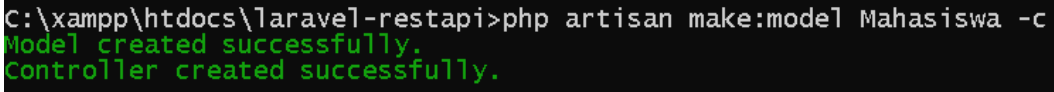
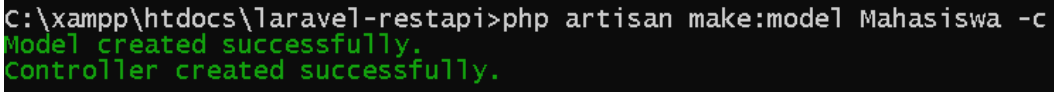
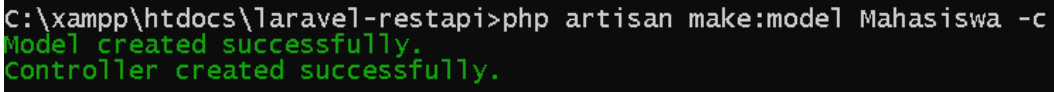
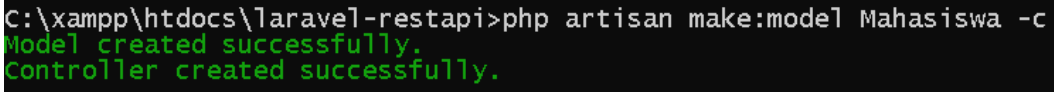
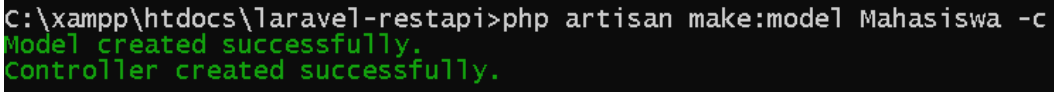
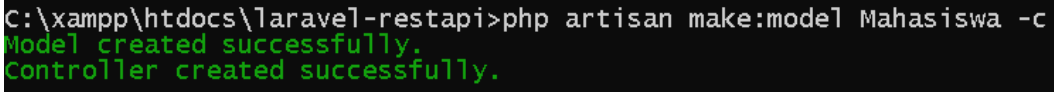
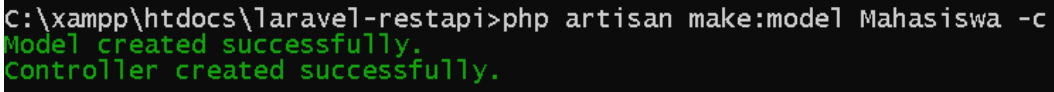
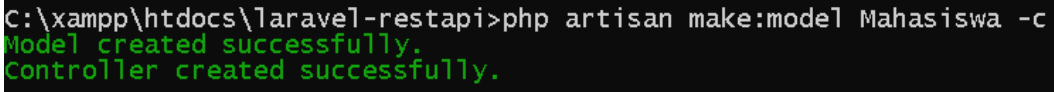
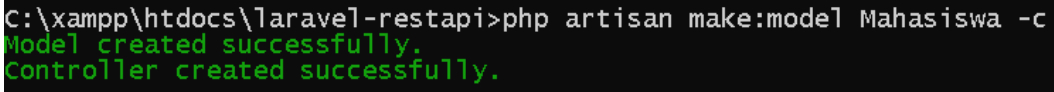
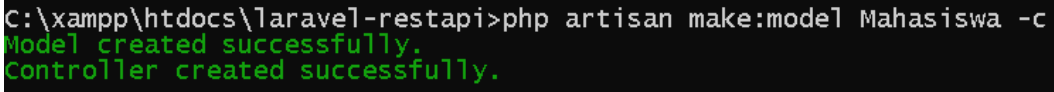
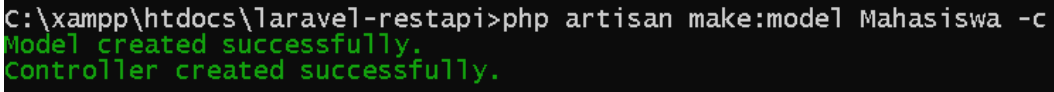
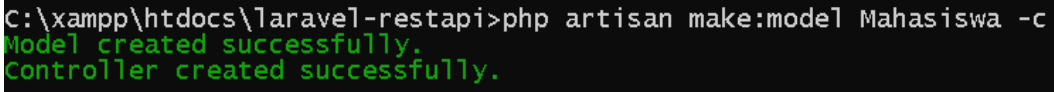
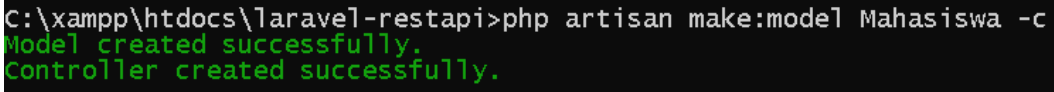
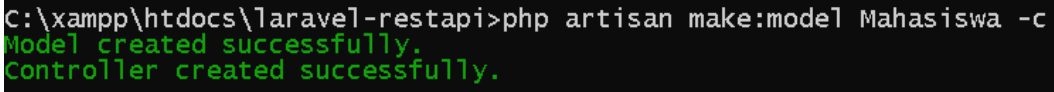
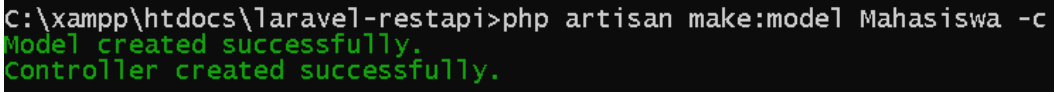
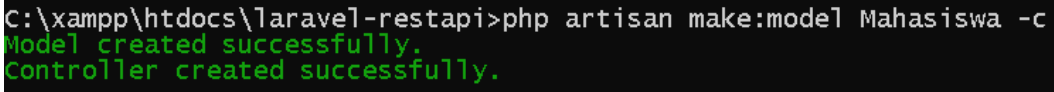
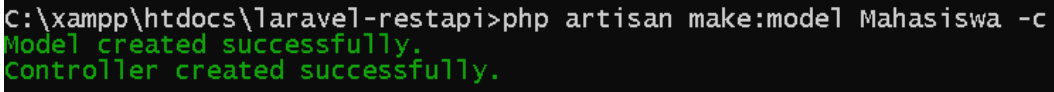
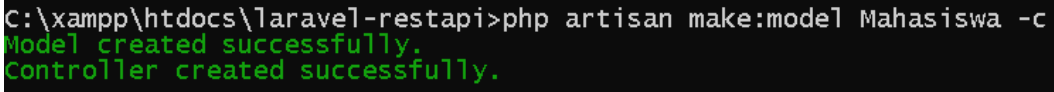
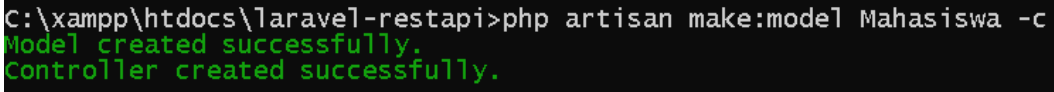
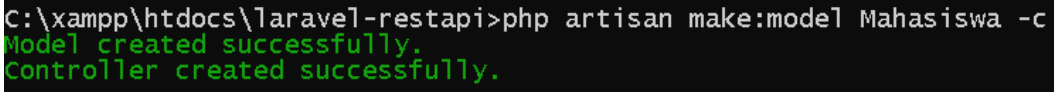
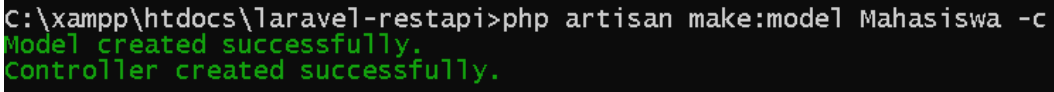
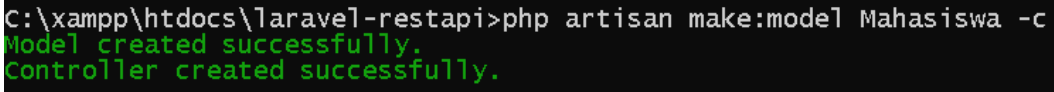
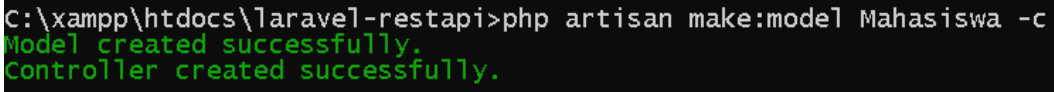
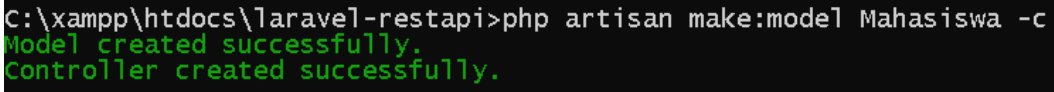
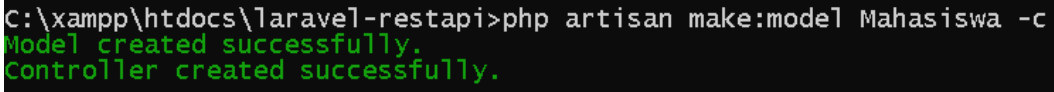
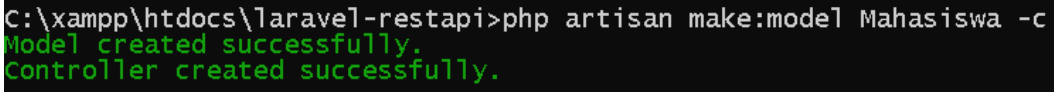
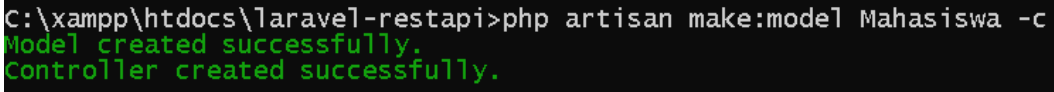
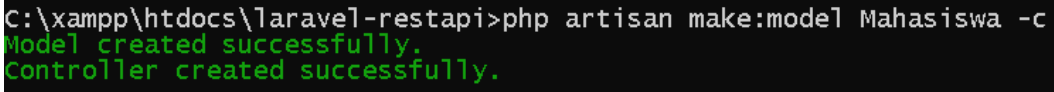
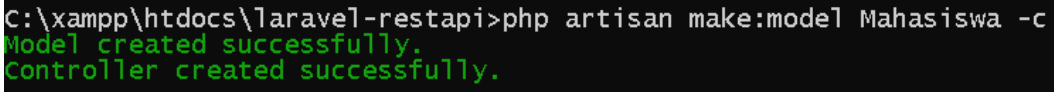
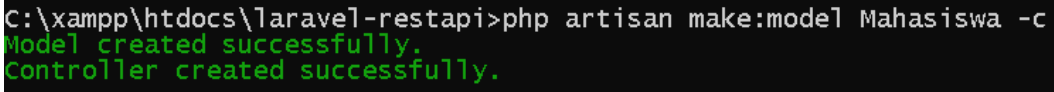
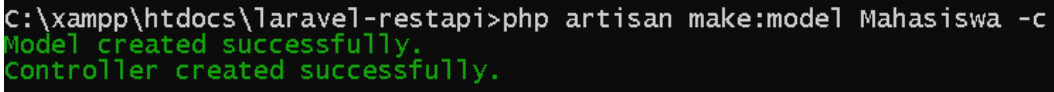
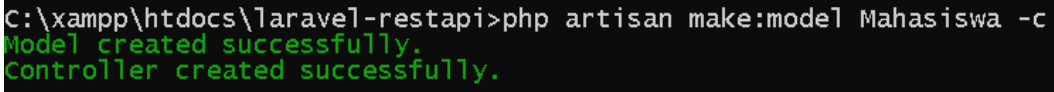
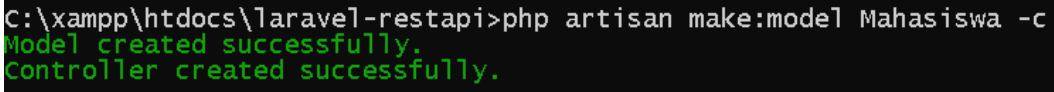
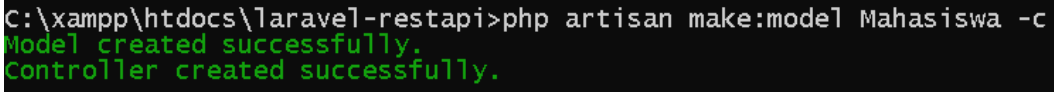
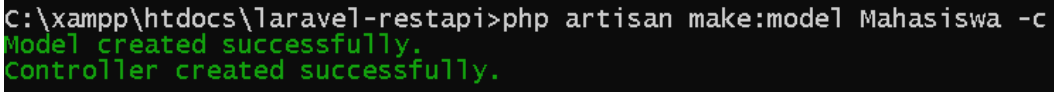
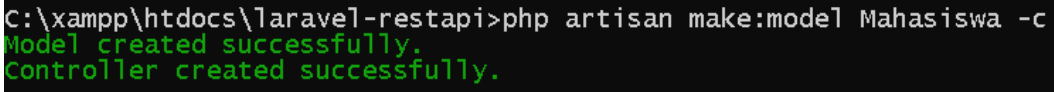
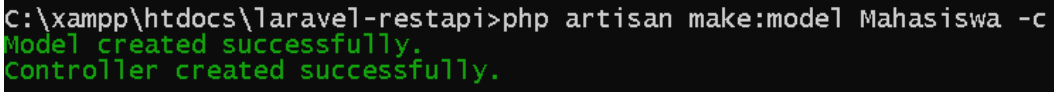
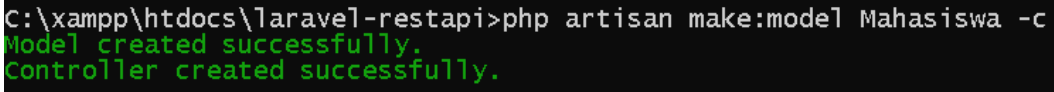
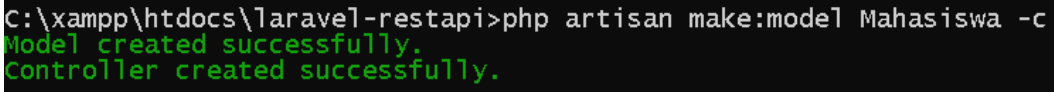
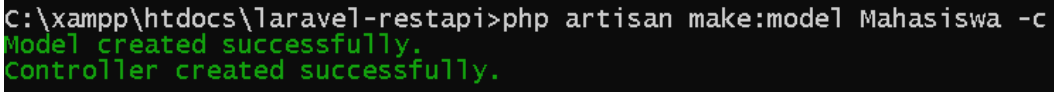
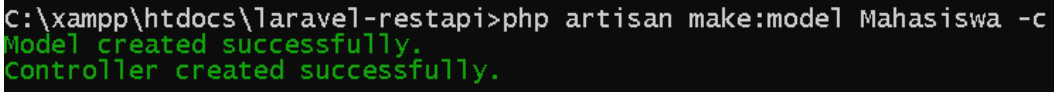
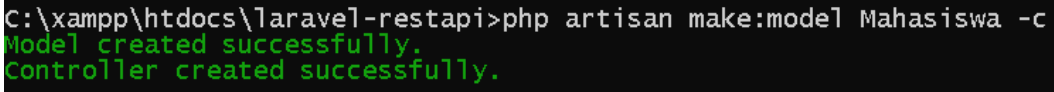
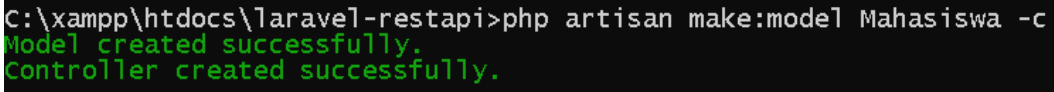
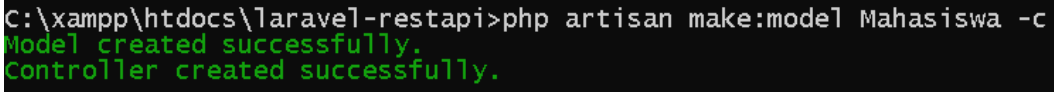
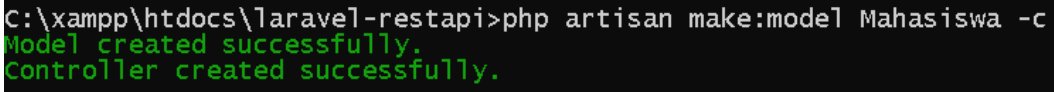
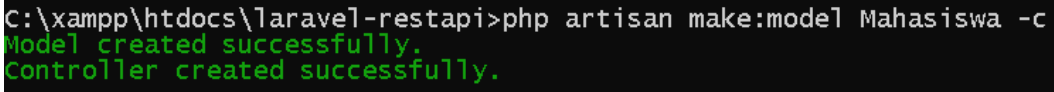
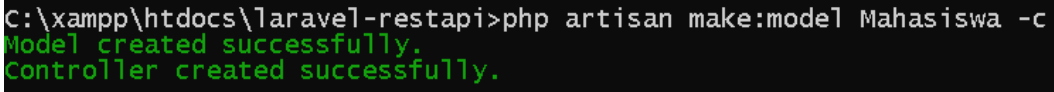
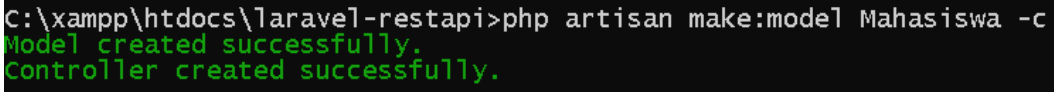
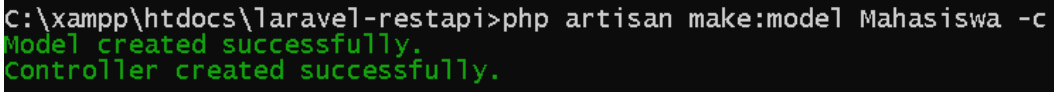
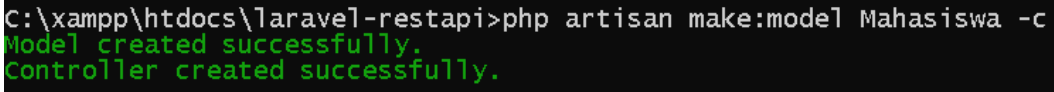
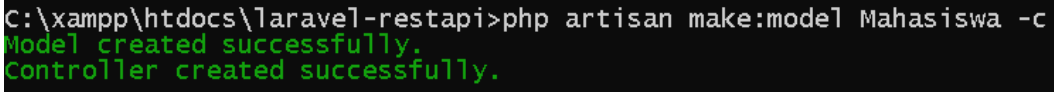
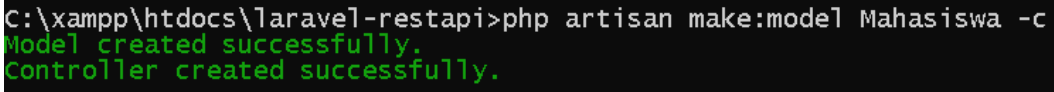
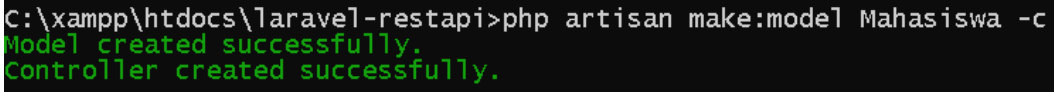
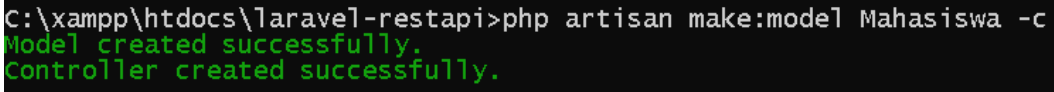
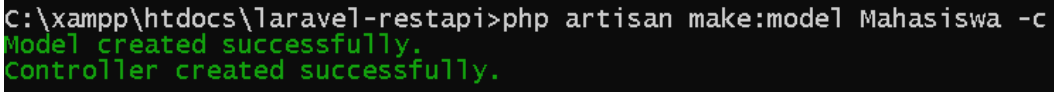
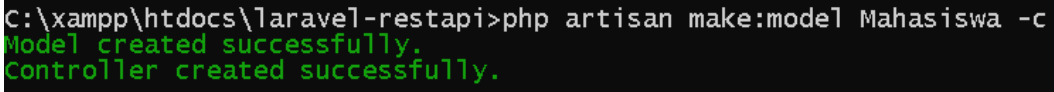
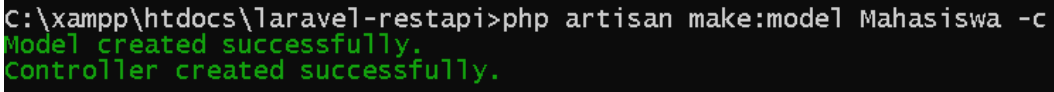
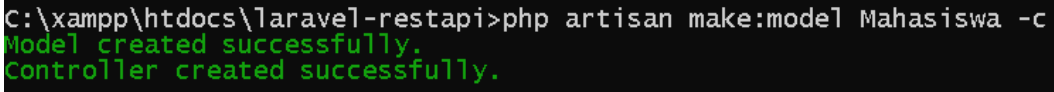
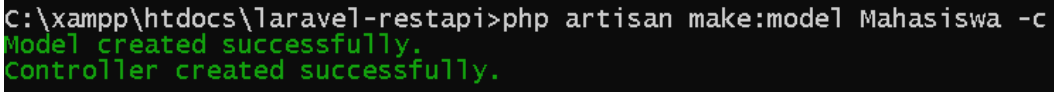
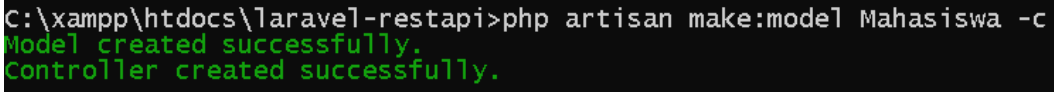
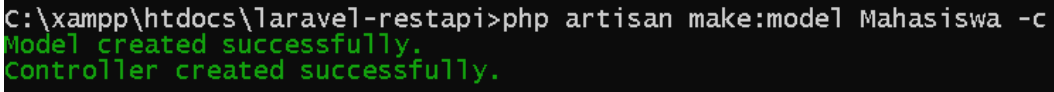
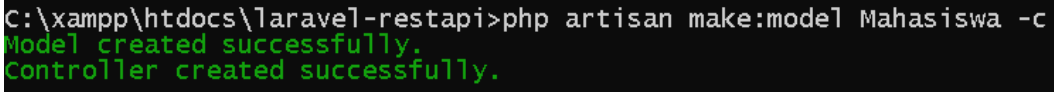
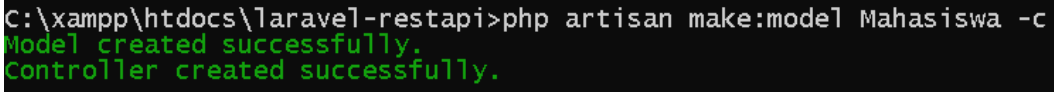
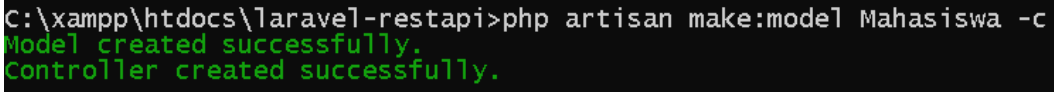
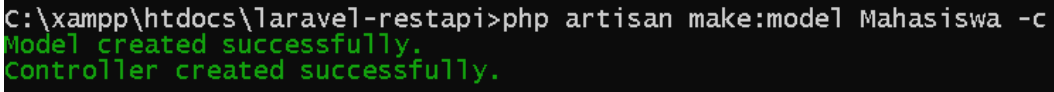
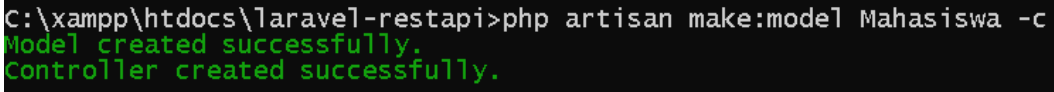
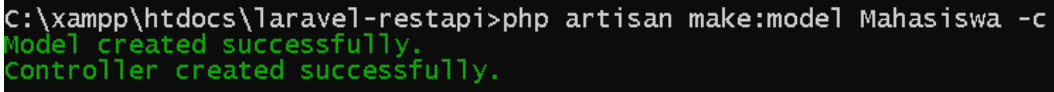
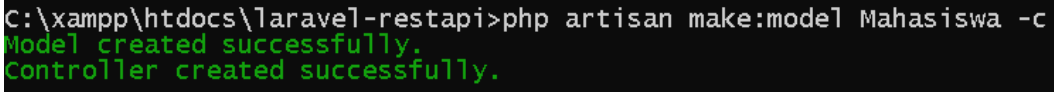
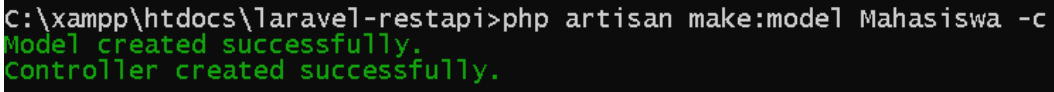
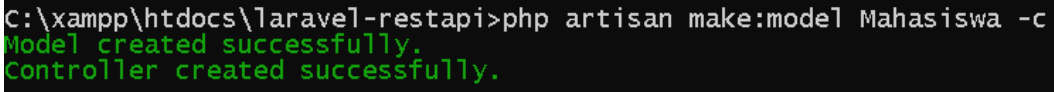
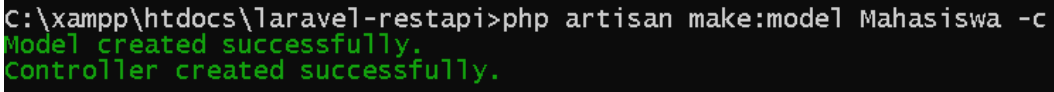
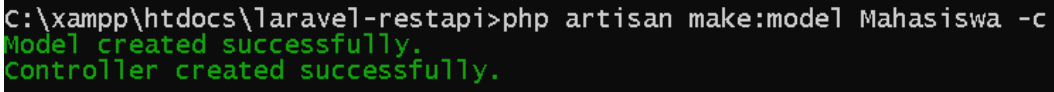
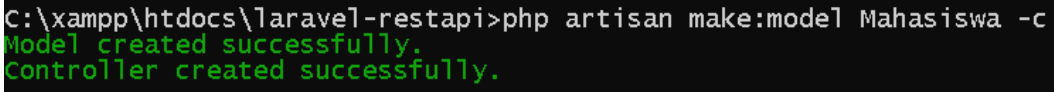
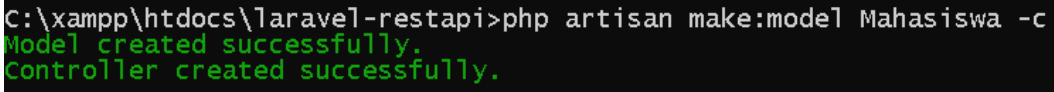
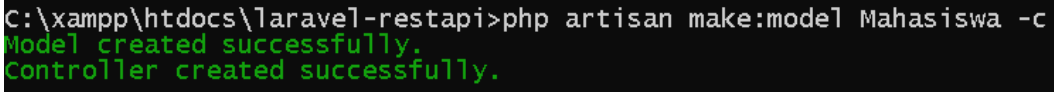
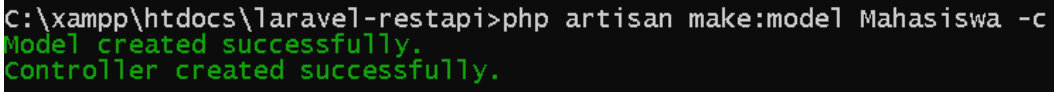
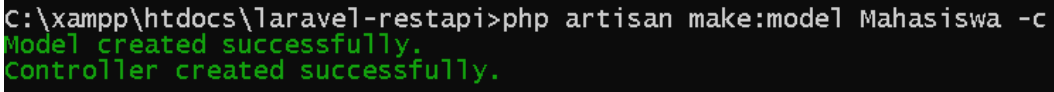
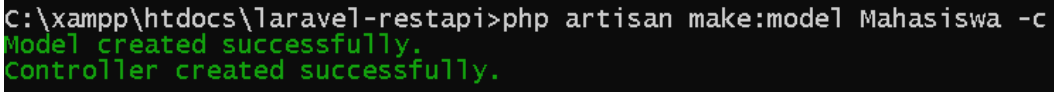
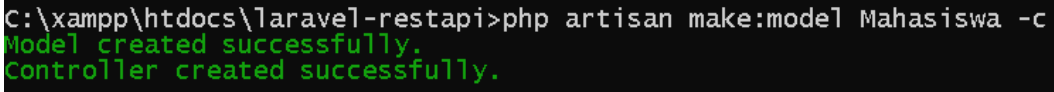
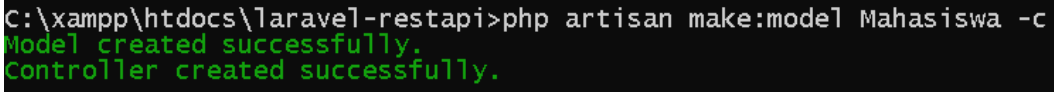
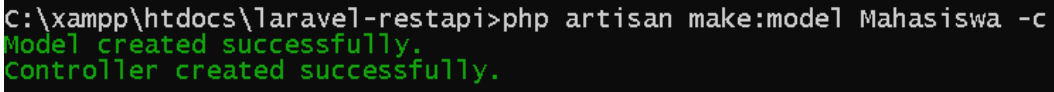
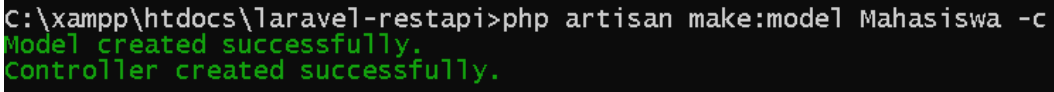
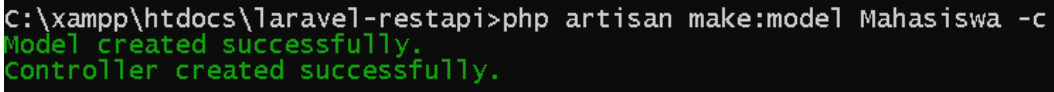
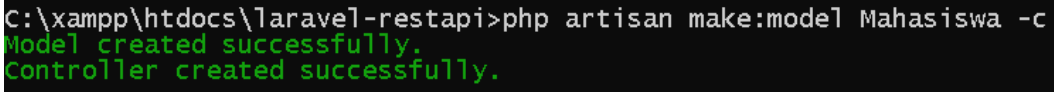
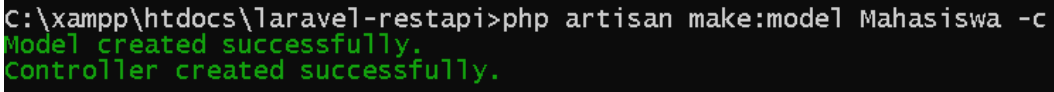
Oleh :

NAMA : ARDAN ANJUNG KUSUMA

NIM : 1841720041

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020**

Praktikum: Membuat RESTful API di Laravel

1	<p>Membuat project baru bernama laravel-restapi</p> <pre>C:\xampp\htdocs>laravel new laravel-restapi Creating 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 Application key set successfully. Discovered Package: facade/ignition Discovered Package: fideloper/proxy Discovered Package: fruitcake/laravel-cors Discovered Package: laravel/tinker Discovered Package: nesbot/carbon Discovered Package: nunomaduro/collision Package manifest generated successfully. - 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</pre>
2	<p>Ganti file .env dan edit nama database menjadi latihan laravel</p> <pre>DB_CONNECTION=mysql DB_HOST=127.0.0.1 DB_PORT=3306 DB_DATABASE=latihan_laravel DB_USERNAME=root DB_PASSWORD=</pre>
3	<p>Membuat model mahasiswa</p> <pre>C:\xampp\htdocs\laravel-restapi>php artisan make:model Mahasiswa -c Model created successfully. Controller created successfully.</pre>                                                                                                                                                                     

```

class MahasiswaController extends Controller
{
    //fungsi index digunakan untuk menampilkan semua data mahasiswa
    public function index()
    {
        $data = Mahasiswa::all();

        //cek data apakah kosong atau tidak
        if (count($data) > 0) {
            $res['message'] = "Success";
            $res['values'] = $data;
            return response($res);
        } else {
            $res['message'] = "Data is Empty";
            return response($data);
        }
    }
}

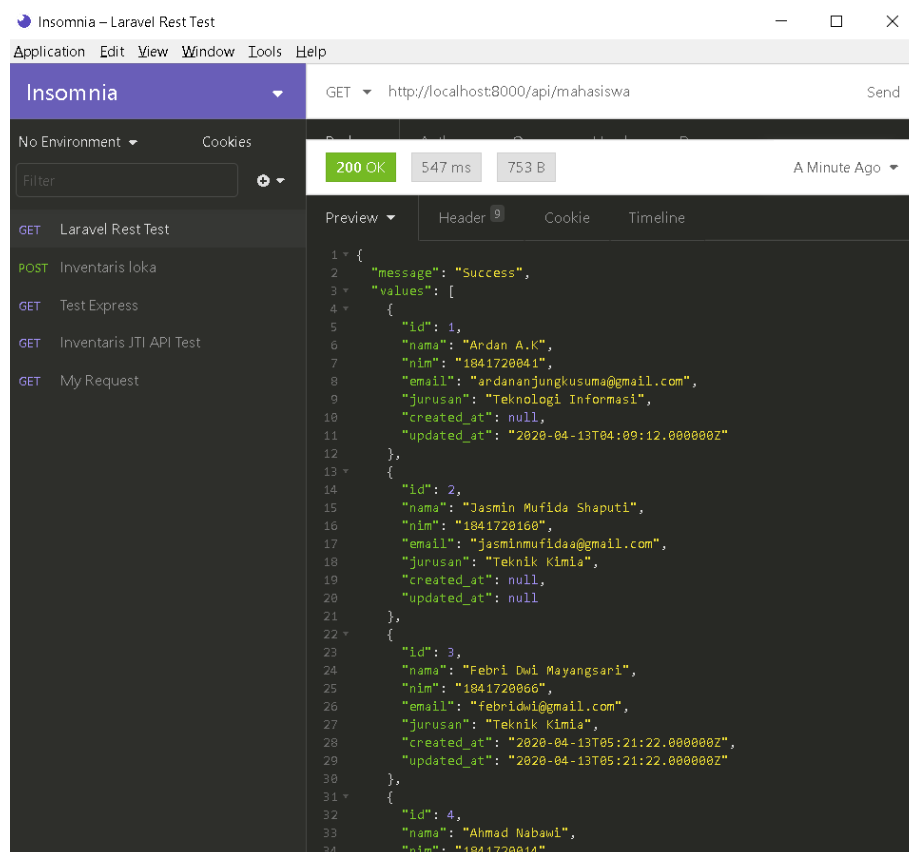
```

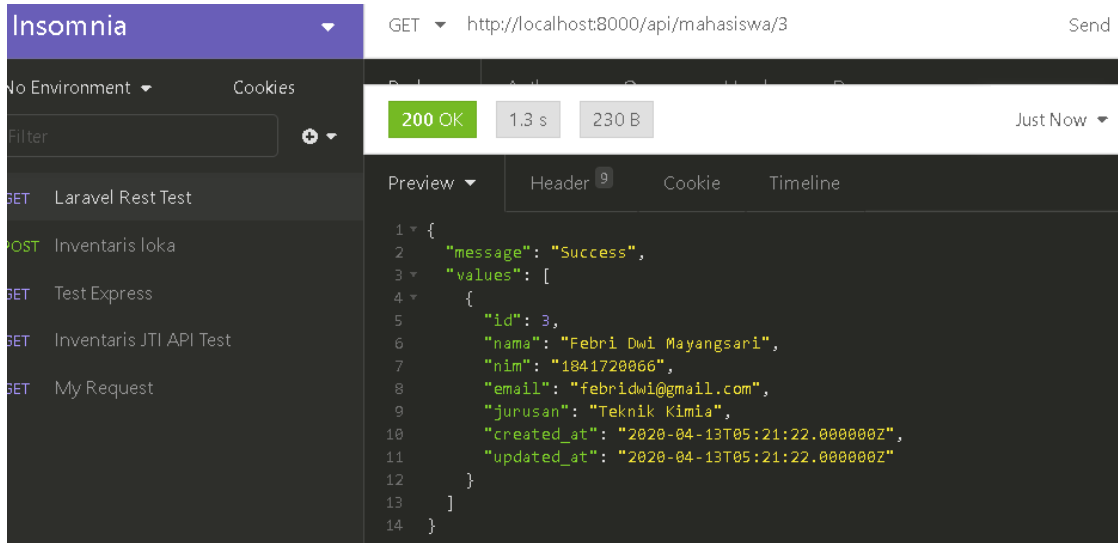
7 Tambahkan route untuk memanggil fungsi index pada file routes/api.php

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

Karena kita ingin menampilkan data maka perintah yang dipakai get

8 Gunakan insomnia untuk melakukan request get



9	<p>Tambahkan function getId pada controller mahasiswa</p> <pre> public function getId(\$id) { \$data = Mahasiswa::where('id', \$id)->get(); //cek jika data ditemukan if (count(\$data) > 0) { \$res['message'] = "Success"; \$res['values'] = \$data; return response(\$res); } else { \$res['message'] = "Data is Empty"; return response(\$data); } } </pre>
10	<p>Tambahkan kode berikut pada routes/api.php, untuk menjalankan function get Mahasiswa By Id</p> <pre> Route::get('/mahasiswa/{id}', 'MahasiswaController@getId'); </pre>
11	<p>Mencoba get data by id menggunakan insomnia</p>  <p>The screenshot shows the Insomnia client interface. The top bar indicates a GET request to <code>http://localhost:8000/api/mahasiswa/3</code> with a status of 200 OK, a response time of 1.3 s, and a body size of 230 B. The response body is a JSON object:</p> <pre> { "message": "Success", "values": [{ "id": 3, "nama": "Febri Dwi Mayangsari", "nim": "1841720066", "email": "febridwi@gmail.com", "jurusan": "Teknik Kimia", "created_at": "2020-04-13T05:21:22.000000Z", "updated_at": "2020-04-13T05:21:22.000000Z" }] } </pre>
12	<p>Tambahkan function create untuk post data pada API. Tambahkan script berikut pada controller mahasiswa</p> <pre> public function create(Request \$request) { \$mhs = new Mahasiswa(); \$mhs->nama = \$request->nama; \$mhs->nim = \$request->nim; \$mhs->email = \$request->email; \$mhs->jurusan = \$request->jurusan; } </pre>

```

        if ($mhs->save()) {
            $res['message'] = "Data has been added";
            $res['value'] = "$mhs";
            return response($res);
        }
    }
}

```

13 Tambahkan pula script berikut pada routes/api.php

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

14 Mencoba post melalui insomnia

The screenshot shows the Insomnia REST client interface. The top bar indicates a POST request to `http://localhost8000/api/mahasiswa`. The left sidebar lists several requests, including 'Laravel Rest Test' and 'Inventaris Ioka'. The main panel shows the 'Form' tab with the following data:

Field	Value
nama	Mentari
nim	1841720069
email	mentariallya@gmail.com
jurusan	Teknologi Informasi
New name	New value

Below the form, the response status is **200 OK**, with a response time of **1.3 s** and a size of **265 B**. The 'Preview' tab shows the JSON response:

```

1 {
2   "message": "Data has been added",
3   "value": {
4     "nama": "Mentari", "nim": "1841720069", "email": "mentariallya@gmail.com", "jurusan": "Teknologi Informasi", "updated_at": "2020-05-04T01:57:27.000000Z", "created_at": "2020-05-04T01:57:27.000000Z", "id": 5
5   }
6 }

```

Melihat hasil yang sudah kita post dengan get

Insomnia

GET http://localhost:8000/api/mahasiswa

Form 4

nama	Mentari
nim	1841720069
email	mentariallya@gmail.com
jurusan	Teknologi Informasi
New name	New value

200 OK 1.17 s 949 B Just Now

Preview

```
31 {
32   "id": 4,
33   "nama": "Ahmad Nabawi",
34   "nim": "1841720014",
35   "email": "ahmadnaba@gmail.com",
36   "jurusan": "Teknik Sipil",
37   "created_at": "2020-04-27T01:52:07.000000Z",
38   "updated_at": "2020-04-27T01:52:07.000000Z"
39 },
40 {
41   "id": 5,
42   "nama": "Mentari",
43   "nim": "1841720069",
44   "email": "mentariallya@gmail.com",
45   "jurusan": "Teknologi Informasi",
46   "created_at": "2020-05-04T01:57:27.000000Z",
47   "updated_at": "2020-05-04T01:57:27.000000Z"
48 }
49 }
```

15 Selanjutnya menambahkan function update pada controller Mahasiswa

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

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

    if ($mhs->save()) {
        $res['message'] = "Data has been updated";
        $res['value'] = "$mhs";
        return response($res);
    }
}
```

```

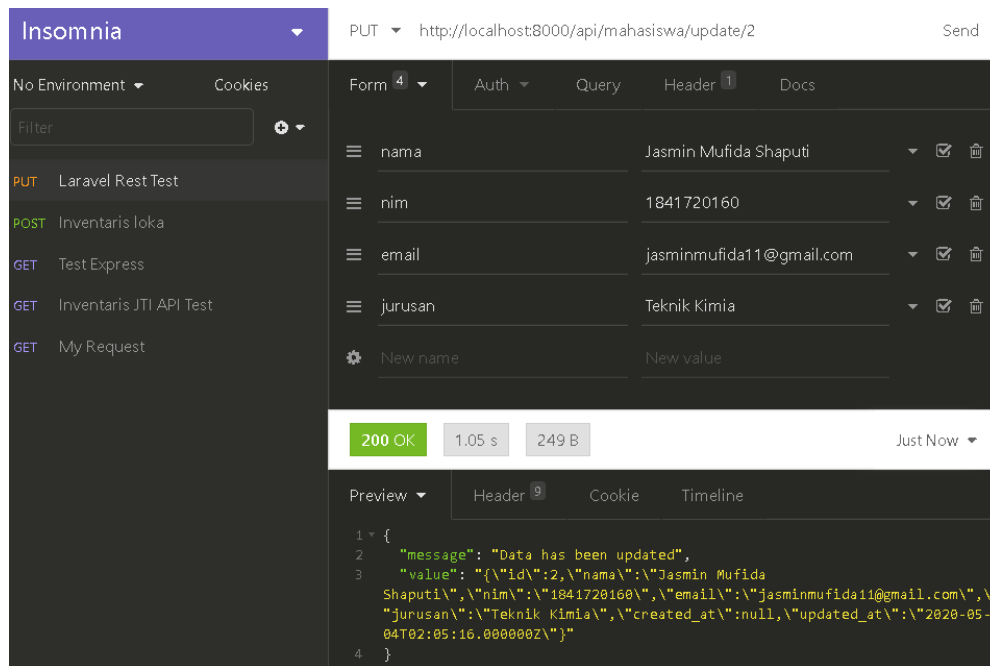
    } else {
        $res['message'] = "Data failed to changed";
        return response($res);
    }
}

```

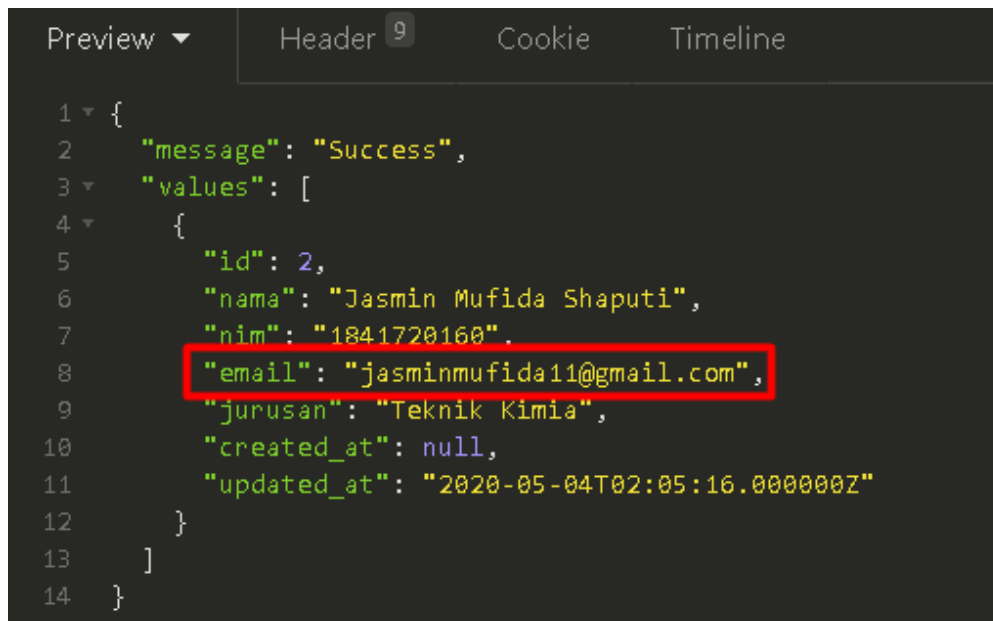
16 Tambahkan script berikut pada routes.php, menggunakan put karena ingin update data

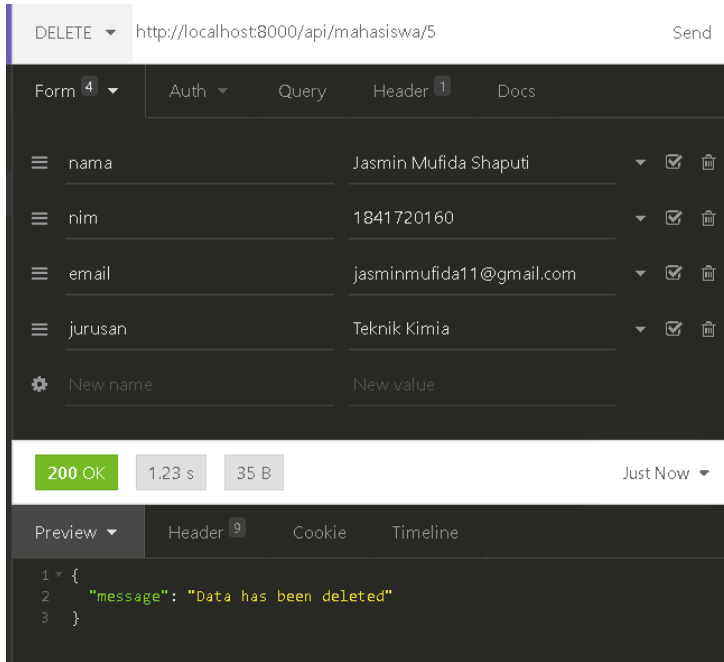
```
Route::put('/mahasiswa/update/{id}', 'MahasiswaController@update');
```

17 Menggunakan put pada insomnia untuk mengubah email



Data sudah berhasil diupdate



18	<p>Tambahkan script berikut untuk delete data pada mahasiswa controller</p> <pre> public function delete(\$id) { \$mhs = Mahasiswa::where('id', \$id); if (\$mhs->delete()) { \$res['message'] = "Data has been deleted"; return response(\$res); } else { \$res['message'] = "Gagal"; return response(\$res); } } </pre>										
19	<p>Tambahkan kode berikut pada routes/api.php</p> <pre> Route::delete('/mahasiswa/{id}', 'MahasiswaController@delete'); </pre>										
20	<p>Hapus data dan sukses</p>  <p>The screenshot shows a REST client interface with the following details:</p> <ul style="list-style-type: none"> Method: DELETE URL: http://localhost:8000/api/mahasiswa/5 Form Data: <table border="1"> <tr><td>nama</td><td>Jasmin Mufida Shaputi</td></tr> <tr><td>nim</td><td>1841720160</td></tr> <tr><td>email</td><td>jasminmufida11@gmail.com</td></tr> <tr><td>jurusan</td><td>Teknik Kimia</td></tr> <tr><td>New name</td><td>New value</td></tr> </table> Status: 200 OK Time: 1.23 s Size: 35 B Response Body: <pre> 1 { 2 "message": "Data has been deleted" 3 } </pre> 	nama	Jasmin Mufida Shaputi	nim	1841720160	email	jasminmufida11@gmail.com	jurusan	Teknik Kimia	New name	New value
nama	Jasmin Mufida Shaputi										
nim	1841720160										
email	jasminmufida11@gmail.com										
jurusan	Teknik Kimia										
New name	New value										