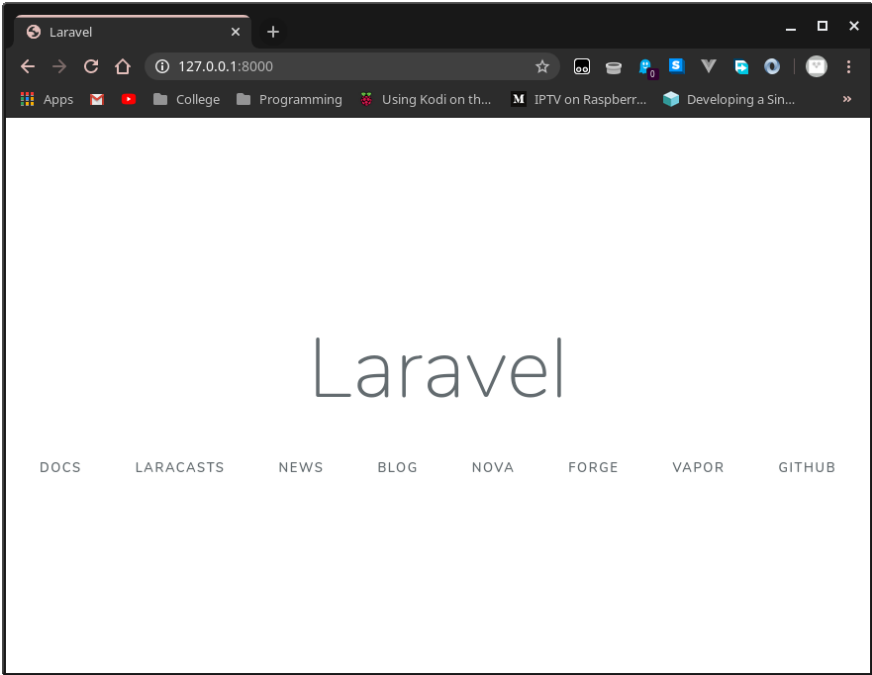
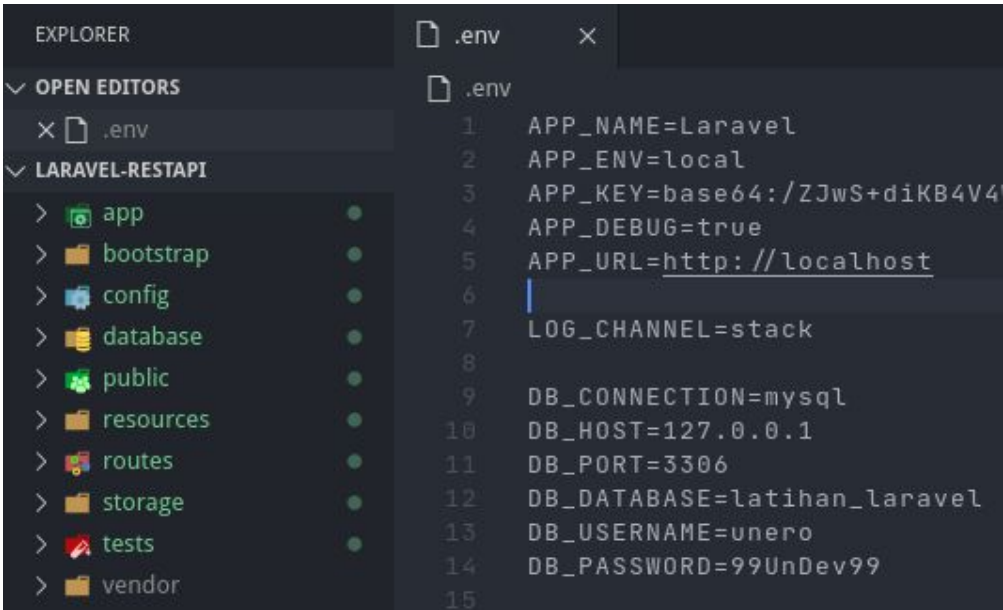


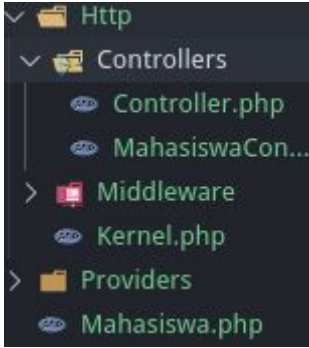
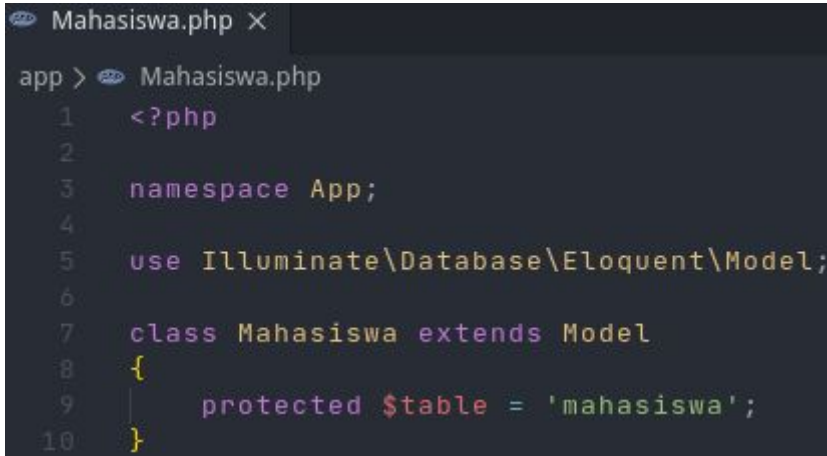
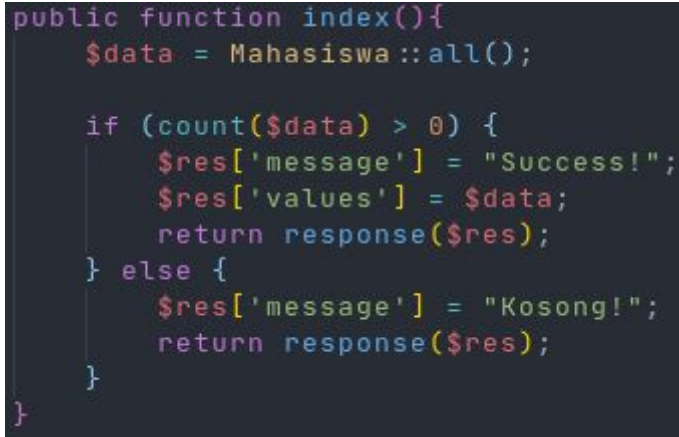
LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT
JOBSHEET 14: MEMBUAT RESTFUL API LARAVEL



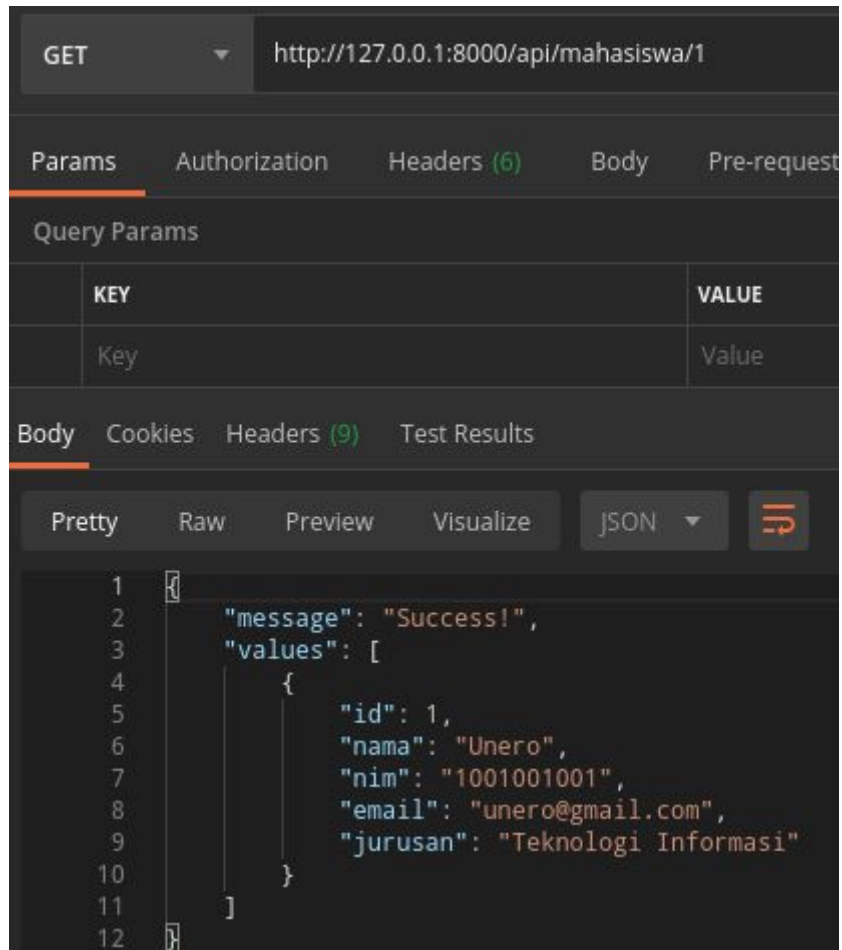
Oleh:
Unero Bhagaskara Ramadhan Marbun
TI 2B - 1841720173

TEKNOLOGI INFORMASI
TEKNIK INFORMATIKA
POLITEKNIK NEGERI MALANG
2020

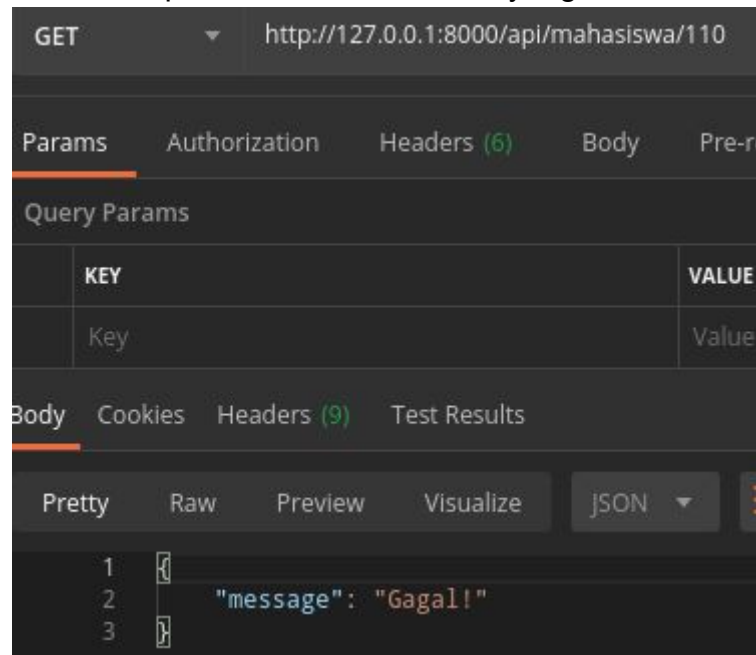
1	<p>php artisan serve</p> 
2	<p>Konfigurasi .env</p> 
3	<p>Membuat Model dan Controller Mahasiswa</p> <pre>[rothetux@ideapads340 laravel-restapi]\$ php artisan make:model Mahasiswa -c Model created successfully. Controller created successfully.</pre> <p>List Folder</p>

	
4	<p>Mengubah isi Model Mahasiswa</p>  <pre> 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 } </pre>
5	<p>Menambahkan fungsi index pada Controller Mahasiswa</p>  <pre> public function index(){ \$data = Mahasiswa::all(); if (count(\$data) > 0) { \$res['message'] = "Success!"; \$res['values'] = \$data; return response(\$res); } else { \$res['message'] = "Kosong!"; return response(\$res); } } </pre>
6	<p>Menambahkan Route API</p> <pre>Route::get('mahasiswa', 'MahasiswaController@index');</pre>
7	<p>Mencoba API dengan POSTMAN</p>

	<div><div>GET ▼ http://127.0.0.1:8000/api/mahasiswa Send ▼</div><div><div>Params Authorization Headers (6) Body Pre-request Script Tests Settings</div><div>Query Params</div><table><thead><tr><th>KEY</th><th>VALUE</th><th>DESCRIPTION</th></tr></thead><tbody><tr><td>Key</td><td>Value</td><td>Description</td></tr></tbody></table><div>Body Cookies Headers (9) Test Results Status: 200 OK Time: 167 ms Size: 605 B Save</div><div><div>Pretty Raw Preview Visualize JSON ▼ </div><div><pre>1 { 2 "message": "Success!", 3 "values": [4 { 5 "id": 1, 6 "nama": "Unero", 7 "nim": "1001001001", 8 "email": "unero@gmail.com", 9 "jurusan": "Teknologi Informasi" 10 }, 11 { 12 "id": 2, 13 "nama": "Denny", 14 "nim": "1001001003", 15 "email": "denny@gmail.com", 16 "jurusan": "Jaringan" 17 }, 18 { 19 "id": 4, 20 "nama": "Unero Tambah"</pre></div></div></div></div>	KEY	VALUE	DESCRIPTION	Key	Value	Description
KEY	VALUE	DESCRIPTION					
Key	Value	Description					
8	<div><div>Fungsi getId pada Controller Mahasiswa</div><div><pre>public function getId(\$id){ \$data = Mahasiswa::where('id', \$id)→get(); if (count(\$data) > 0) { \$res['message'] = "Success!"; \$res['values'] = \$data; return response(\$res); } else { \$res['message'] = "Kosong!"; return response(\$res); } }</pre></div></div>						
9	<div><div>Route getId</div><div><pre>Route::get('mahasiswa/{id}', 'MahasiswaController@getId');</pre></div></div>						
10	<div><div>Mencoba API getId</div></div>						



Tampilan ketika tidak ada Id yang cocok



```

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 ditambah";
        $res['values'] = "$mhs";
        return response($res);
    }
}

```

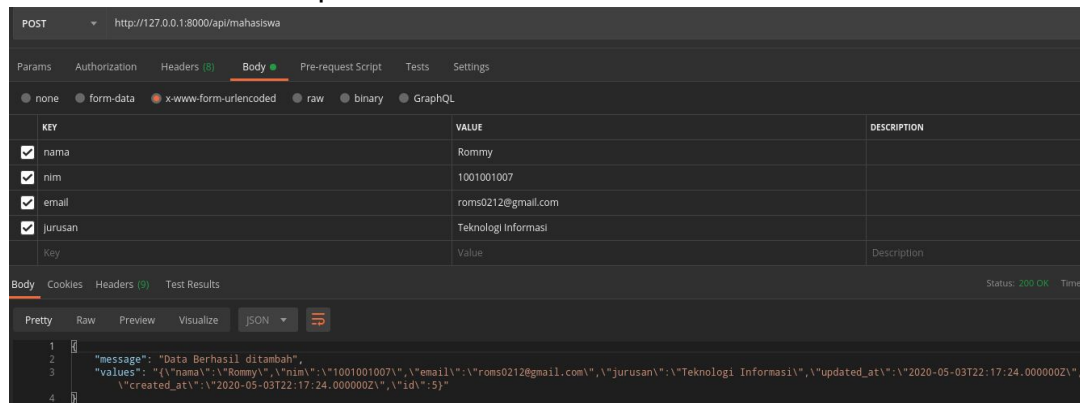
12

Route Create API

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

13

Tampilan ketika create lewat POSTMAN



Tampilan data yang sudah dicreate

```

31 {
32   "id": 5,
33   "nama": "Rommy",
34   "nim": "1001001007",
35   "email": "roms0212@gmail.com",
36   "jurusan": "Teknologi Informasi",
37   "updated_at": "2020-05-03T22:17:24.000000Z",
38   "created_at": "2020-05-03T22:17:24.000000Z"
39 }
40 ]
41

```

14

Fungsi 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 Berhasil ditambah";
        $res['values'] = "$mhs";
        return response($res);
    } else {
        $res['message'] = "Gagal!";
        return response($res);
    }
}

```

15

Route PUT dari API

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

16

Update data dengan id = 5, lewat POSTMAN

PUT ▼ http://127.0.0.1:8000/api/mahasiswa/update/5

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	nama	Romy Rafael	
<input checked="" type="checkbox"/>	nim	1001001007	
<input checked="" type="checkbox"/>	email	roms0212@gmail.com	
<input checked="" type="checkbox"/>	jurusan	Teknologi Informasi	
	Key	Value	Description

Body Cookies Headers (9) Test Results Status: 200 OK Time: 168 ms

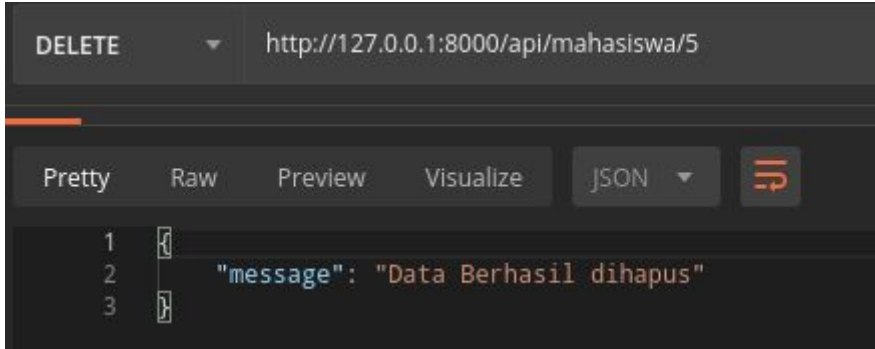
Pretty Raw Preview Visualize JSON ↻

```

1  [
2    "message": "Data Berhasil diubah",
3    "values": "{\"id\":\"5\",\"nama\":\"Romy Rafael\",\"nim\":\"1001001007\",\"email\":\"roms0212@",
4    \"jurusan\":\"Teknologi Informasi\",\"updated_at\":\"2020-05-03T22:28:37.000000Z\",
    \"created_at\":\"2020-05-03T22:17:24.000000Z\"}"

```

Hasil perubahan data

	<pre> { "id": 5, "nama": "Romy Rafael", "nim": "1001001007", "email": "roms0212@gmail.com", "jurusan": "Teknologi Informasi", "updated_at": "2020-05-03T22:28:37.000000Z", "created_at": "2020-05-03T22:17:24.000000Z" } </pre> <p>Rommy -> Romy Rafael</p>
17	<p>Membuat fungsi delete pada Controller Mahasiswa</p> <pre> public function delete(\$id){ \$mhs = Mahasiswa::where('id', \$id); if (\$mhs->delete()) { \$res['message'] = "Data Berhasil dihapus"; return response(\$res); } else { \$res['message'] = "Gagal!"; return response(\$res); } } </pre>
18	<p>Route delete pada API</p> <pre> Route :: delete('mahasiswa/{id}', 'MahasiswaController@delete'); </pre>
19	<p>Menghapus data lewat POSTMAN</p> 

== SELESAI ==