LAPORAN TUGAS PEMROGRAMAN WEB LANJUT

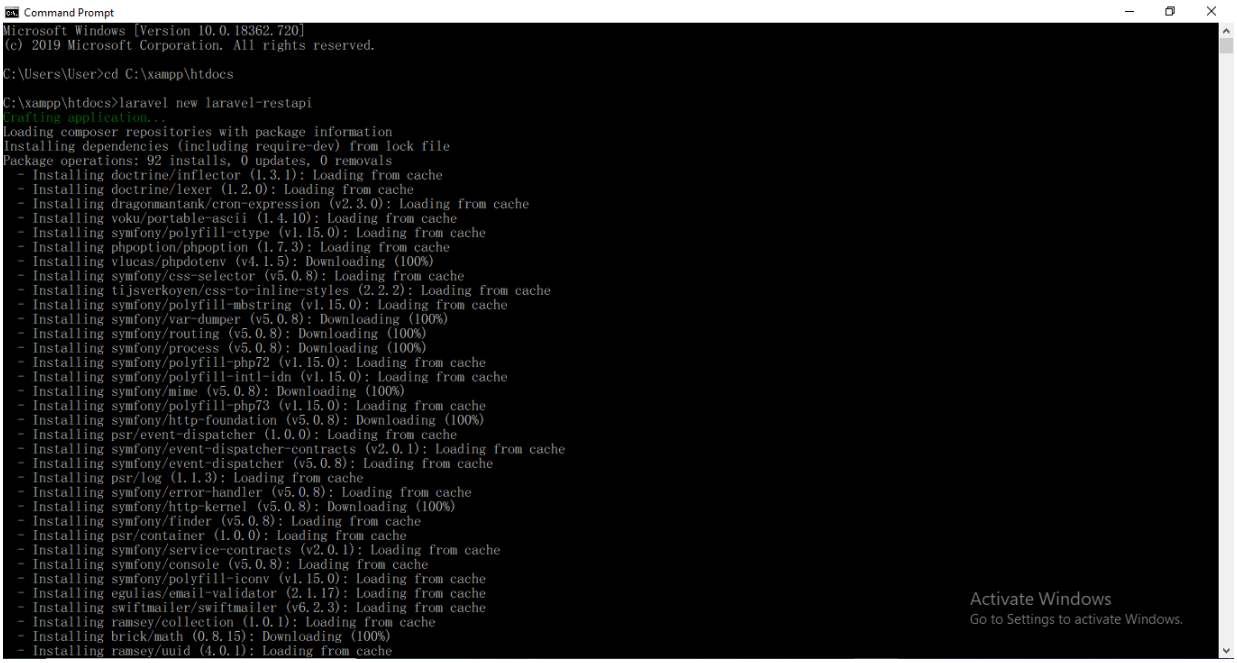
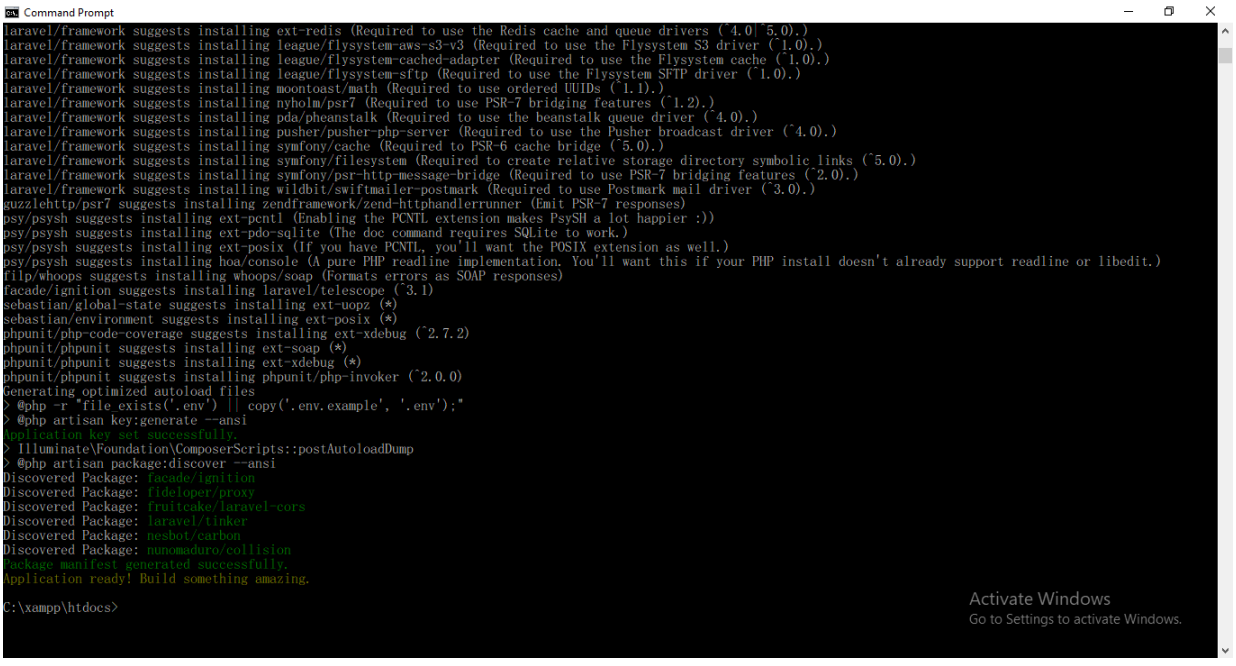
Jobsheet 14
Membuat RESTful API Laravel

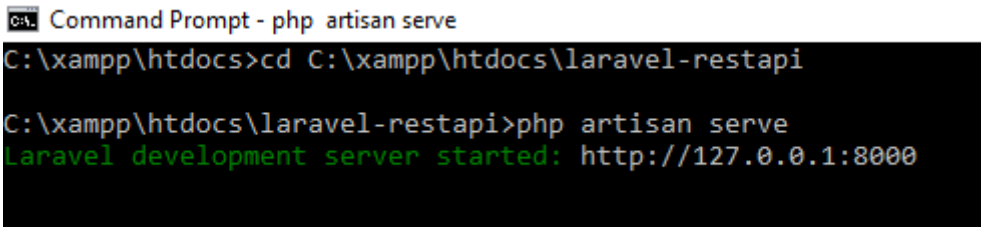
Disusun Oleh :

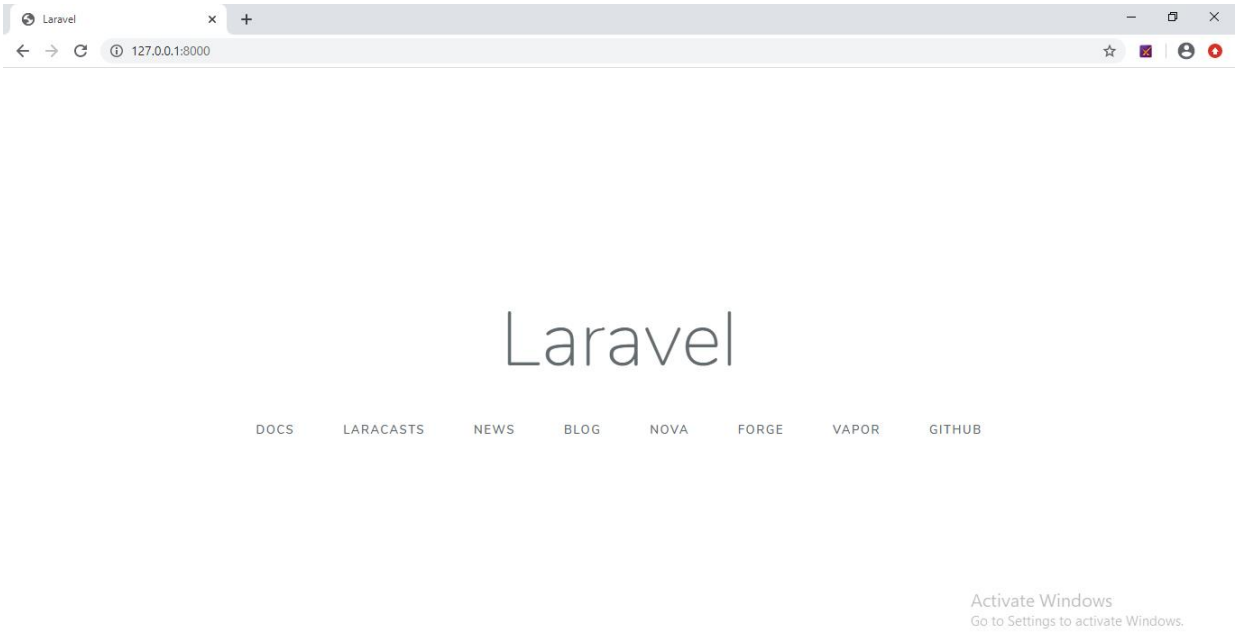
Nama : Hunayn Risatayn
NIM : 1841720148
Kelas : TI-2B

Program Studi D-IV Teknik Informatika
Jurusan Teknologi Informasi
Politeknik Negeri Malang

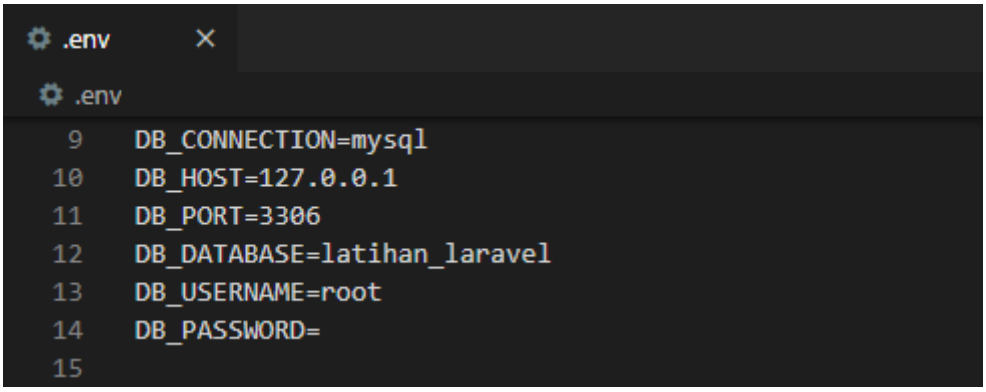
Praktikum – Bagian 1: Membuat RESTful API di Laravel

Langkah	Keterangan
1	Buat project beru dengan nama “laravel-restapi”. Buka command prompt, tuliskan perintah berikut. **cd C:\xampp\htdocs** **laravel new laravel-restapi**   Kita coba jalankan dulu project tersebut. Pada command prompt tulis perintah berikut. **cd C:\laravel-restapi** **php artisan serve** Akan tampil halaman default Laravel seperti di bawah ini.





Kemudian lakukan konfigurasi database pada file .env. Isikan nama database, username, dan password yang akan digunakan. Pada project ini, kita gunakan database dari latihan di minggu-minggu sebelumnya yaitu “latihan_laravel”



Buat model dengan nama Mahasiswa, buat juga controllernya. Untuk membuat model dan controllernya sekaligus tuliskan perintah berikut pada command prompt (terlebih dahulu keluar dari php artisan serve dengan mengetik ctrl+C pada keyboard)

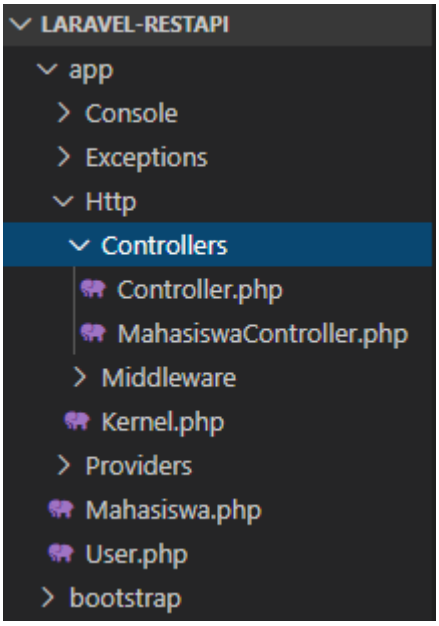
php artisan make:model Mahasiswa -c

Keterangan : • -c merupakan perintah untuk menyertakan pembuatan controller

Sehingga pada project laravel-restapi akan bertambah dua file yaitu model Mahasiswa.php serta controller MahasiswaController.php.

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

C:\xampp\htdocs\laravel-restapi>
```



Selanjutnya ubah isi model Mahasiswa.php seperti berikut ini.

Keterangan:

- Model ini akan mengelola tabel “mahasiswa” yang terdapat pada database latihan_laravel

app > Mahasiswa.php > ...

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Mahasiswa extends Model
{
    protected $table = 'mahasiswa';
}

```

Kemudian kita akan memodifikasi isi dari MahasiswaController.php untuk dapat mengolah data pada tabel ‘mahasiswa’. Pada controller ini, kita akan melakukan operasi untuk menampilkan, menambah, mengubah, dan menghapus data. Pertama, kita akan mengubah fungsi index agar saat fungsi index dipanggil, maka aplikasi akan menampilkan seluruh data dari tabel mahasiswa.

Keterangan:

- Tambahkan line 6 agar model Mahasiswa dapat digunakan pada MahasiswaController
- Line 15-19 digunakan untuk memeriksa apakah data>0 atau data tidak kosong
- Variabel $res[message] digunakan untuk menampilkan pesan apakah ada data atau tidak ada data di tabel mahasiswa
- Variabel $array[values] akan menyimpan semua baris data pada tabel mahasiswa

app > Http > Controllers > MahasiswaController.php > ...

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Mahasiswa;

class MahasiswaController extends Controller
{
    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);
        }
    }
}

```

Tambahkan route untuk memanggil fungsi index pada file routes/api.php (Line 21).

Karena kita ingin menampilkan data, maka perintah yang dipakai adalah 'get'.

routes > api.php > ...

```php
<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

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

```

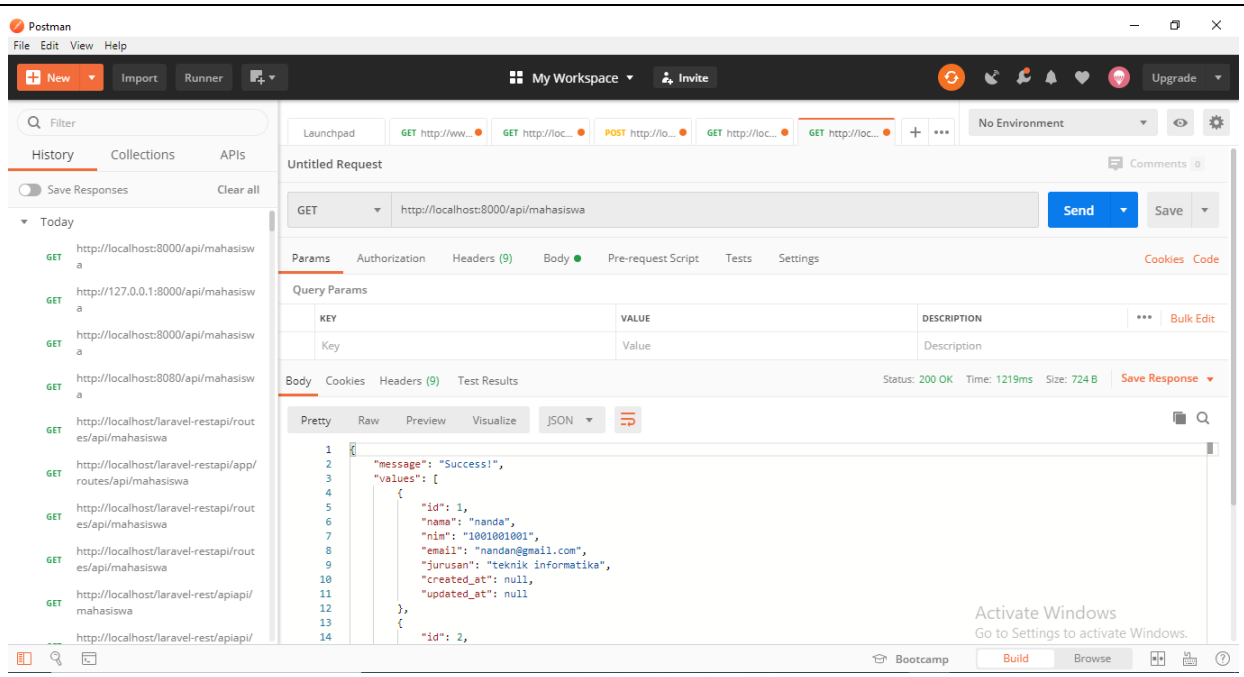
Ketikkan perintah php artisan serve pada command prompt. Lalu kita coba menguji fungsi untuk menampilkan data menggunakan aplikasi Postman.

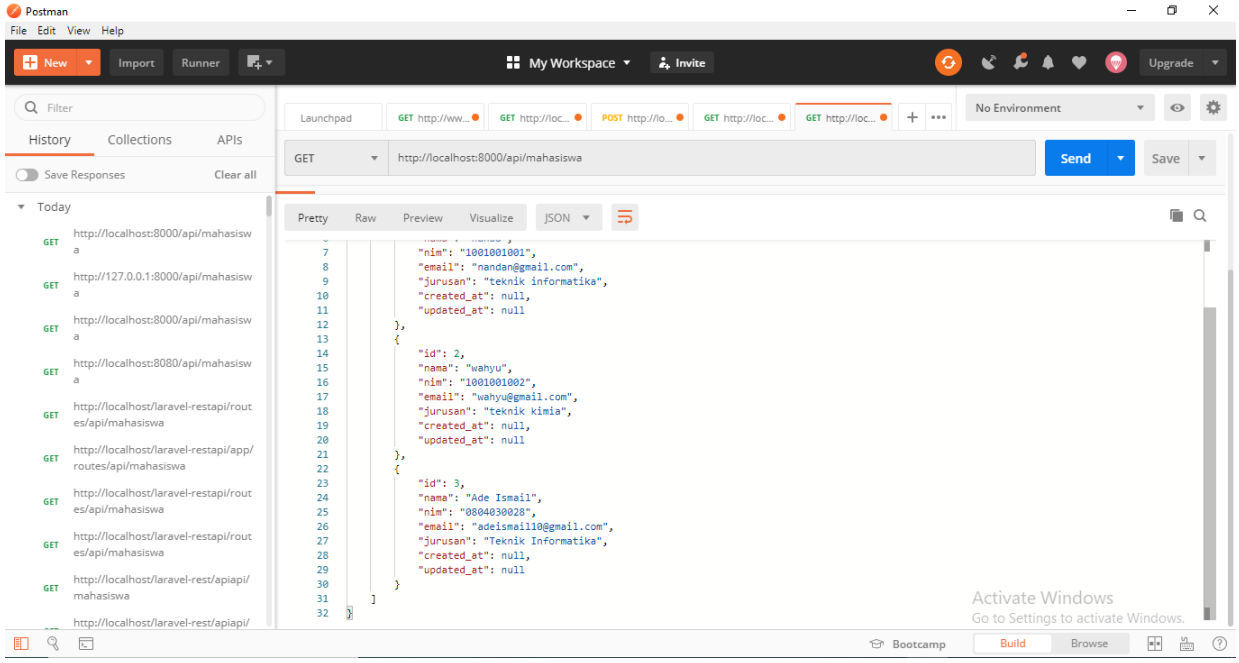
Gunakan perintah GET, isikan url : http://localhost:8000/api/mahasiswa

Berikut adalah tampilan dari aplikasi Postman.

Semua data pada tabel mahasiswa akan tampil, ditampilkan juga pesan sukses.

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





Selanjutnya kita akan menambahkan fungsi untuk melihat suatu data ketika dipilih ID tertentu. Buat fungsi baru yaitu getId pada MahasiswaController.php.

Keterangan:

- Fungsi getId menerima parameter $id yang menunjukkan ID mahasiswa yang dipilih
- Line 30 merupakan pemanggilan model untuk membaca data berdasarkan ID

```php
app > Http > Controllers > MahasiswaController.php > PHP Intelephense > MahasiswaController > index
24      public function getId($id)
25      {
26          $data = Mahasiswa::where('id', $id)->get();
27
28          if (count($data) > 0) {
29              $res['message'] = "Success!";
30              $res['values'] = $data;
31              return response($res);
32          } else {
33              $res['message'] = "Gagal!";
34              return response($res);
35          }
36      }
37  }
38
```

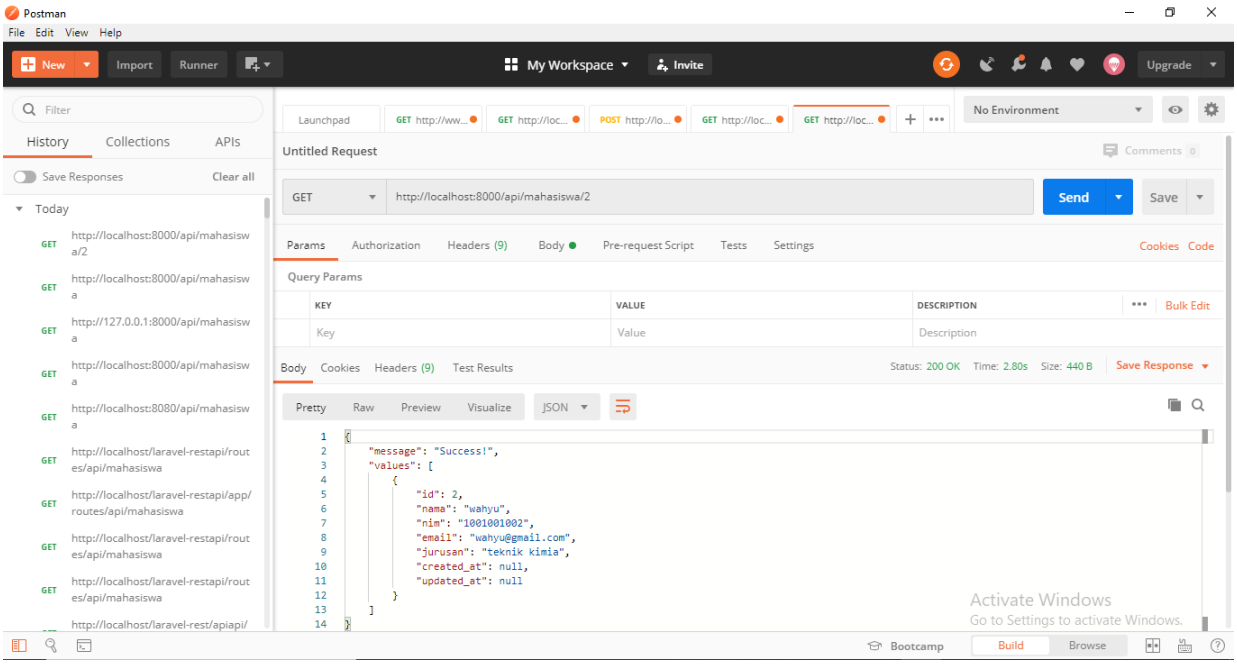
Tambahkan route untuk memanggil fungsi getId pada routes/api.php

Karena kita ingin menampilkan data, maka perintah yang dipakai adalah ‘get’

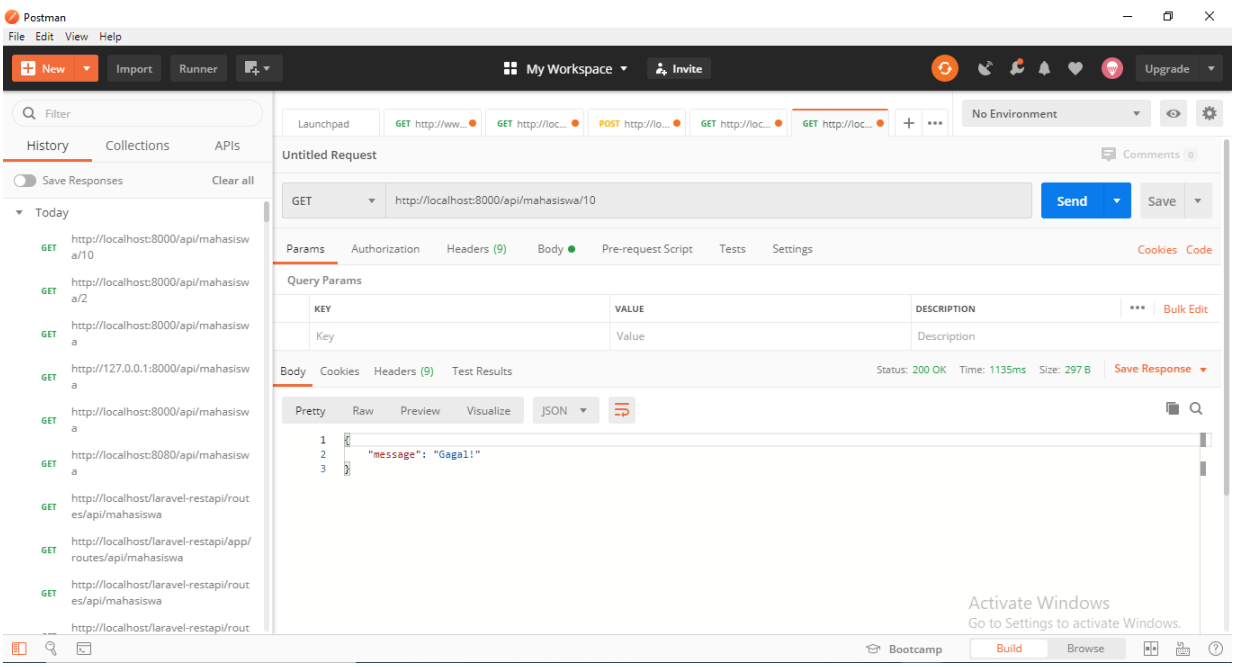
```php
routes > api.php > ...
22
23  Route::get('/mahasiswa/{id}', 'MahasiswaController@getId');
24
```

Sekarang kita coba untuk menampilkan data berdasarkan ID mahasiswa yang dipilih menggunakan Postman. Gunakan perintah GET untuk menampilkan data.

Di bawah ini adalah contoh untuk menampilkan data dengan ID=2, maka url diisi :
http://localhost:8000/api/mahasiswa/2



Ketika mencoba menampilkan ID=10 akan muncul pesan “Gagal”, karena tidak ada data mahasiswa dengan ID tersebut.



Setelah dapat menampilkan data, kita akan membuat fungsi untuk menambahkan data baru ke database dengan nama create pada MahasiswaController.php.

Keterangan:

• Fungsi create menerima parameter Request yang menampung isian data mahasiswa yang akan ditambahkan ke database.

• Line 55-59 : $mhs->save() digunakan untuk menyimpan data ke database, apabila save() berhasil dijalankan maka akan ditampilkan pesan berhasil serta data yang ditambah.

app > Http > Controllers > MahasiswaController.php > PHP Intelephense > MahasiswaController > getId

```
38      public function create(Request $request)
39      {
40          $mhs = new Mahasiswa();
41          $mhs->nama = $request->nama;
42          $mhs->nim = $request->nim;
43          $mhs->email = $request->email;
44          $mhs->jurusan = $request->jurusan;
45
46          if ($mhs->save()) {
47              $res['message'] = "Data berhasil ditambah!";
48              $res['value'] = "$mhs";
49              return response($res);
50          }
51      }
52  }
53
```

Tambahkan route untuk memanggil fungsi create pada routes/api.php

Karena kita ingin menambah data, maka perintah yang dipakai adalah ‘post’.

routes > api.php > ...

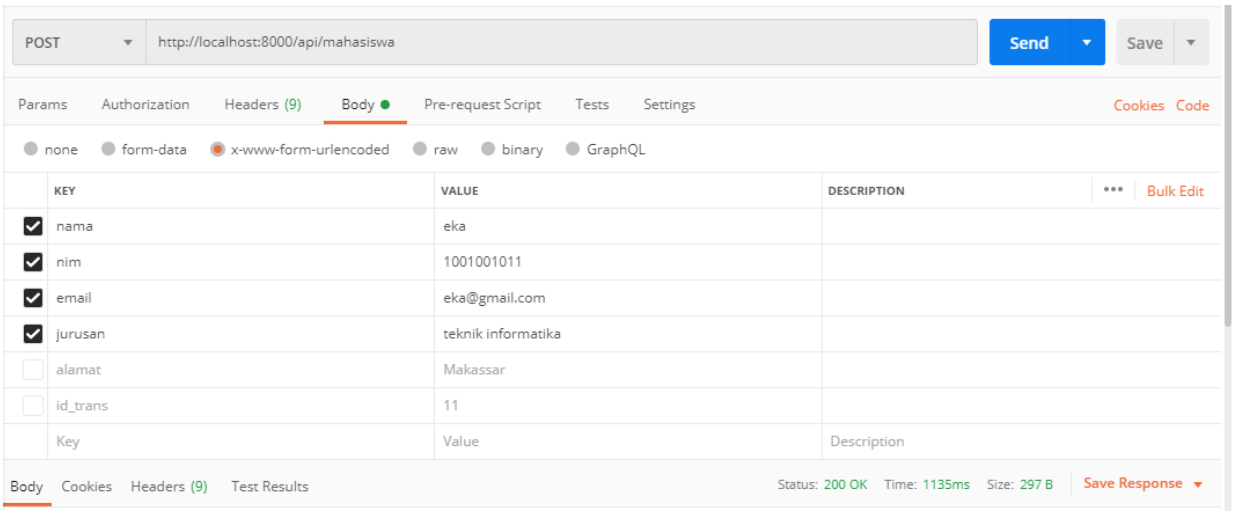
```
24
25  Route::post('/mahasiswa', 'MahasiswaController@create');
26
```

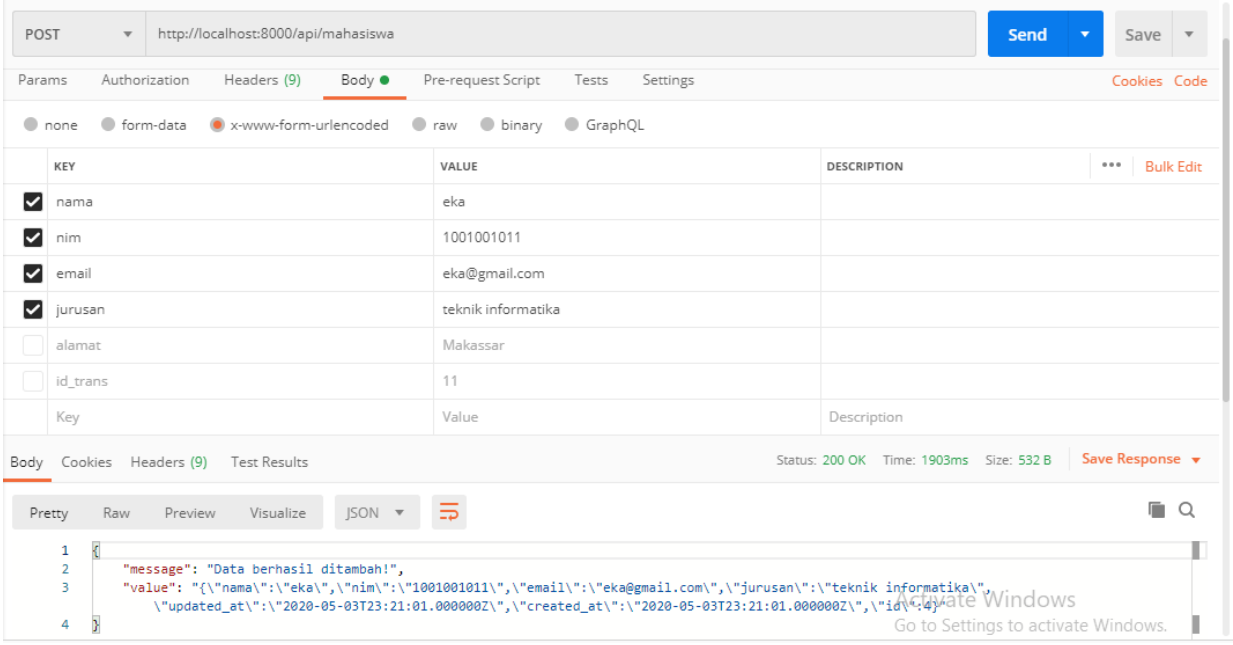
Kita coba untuk menambahkan data melalui Postman.

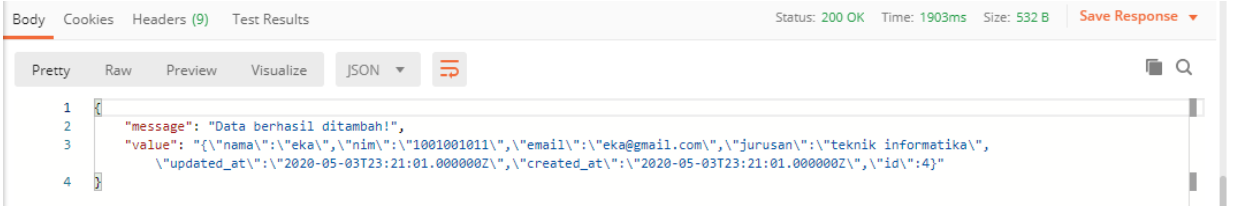
• Isikan url : http://localhost:8000/api/mahasiswa. Karena kita ingin mengirim data ke database, maka

perintah yang dipakai adalah ‘POST’.

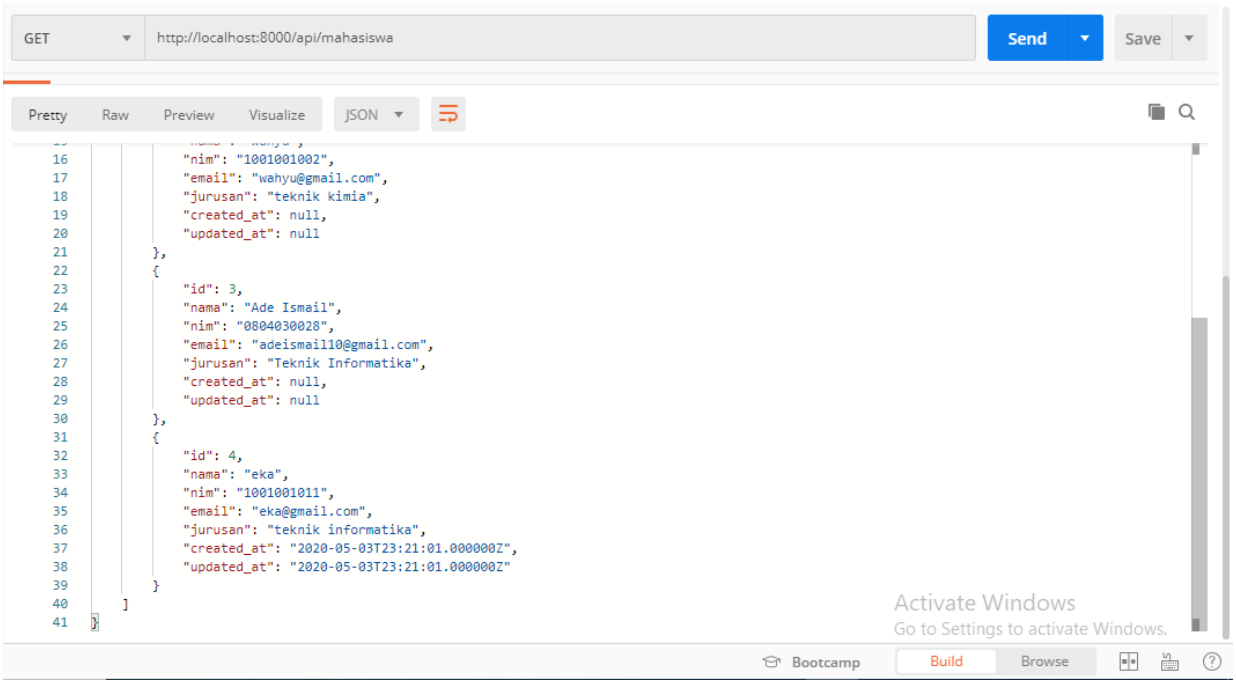
• Pilih tab Body dan pilih radio button x-www-form-urlencode. Isikan nama kolom pada database pada KEY, untuk isian datanya tuliskan pada VALUE.







Kemudian coba untuk tampilkan semua data, kita lihat apakah data baru sudah masuk ke database.



Selanjutnya kita akan menambahkan fungsi untuk mengubah data dari database. Buat fungsi update pada MahasiswaController.php.

Keterangan:

• Fungsi update menerima parameter Request yang menampung isian data mahasiswa yang akan diubah dan parameter id yang menunjukkan ID yang dipilih.

• Line 70 : Mahasiswa::find($id) digunakan untuk pencarian data pada tabel mahasiswa berdasarkan $id.
• Line 76-80 : $mhs->save() digunakan untuk menyimpan perubahan data ke database, apabila save() berhasil dijalankan maka akan ditampilkan pesan berhasil serta data yang diubah.

app > Http > Controllers > MahasiswaController.php > PHP Intelephense > MahasiswaController > create

```
53      public function update(Request $request, $id)
54      {
55          $nama = $request->nama;
56          $nim = $request->nim;
57          $email = $request->email;
58          $jurusan = $request->jurusan;
59
60          $mhs = Mahasiswa::find($id);
61          $mhs->nama = $nama;
62          $mhs->nim = $nim;
63          $mhs->email = $email;
64          $mhs->jurusan = $jurusan;
65
66          if ($mhs->save()) {
67              $res['message'] = "Data berhasil diubah!";
68              $res['value'] = "$mhs";
69              return response($res);
70          } else {
71              $res['message'] = "Gagal!";
72              return response($res);
73          }
74      }
75  }
76
```

Tambahkan route untuk memanggil fungsi update pada routes/api.php

Karena kita ingin memasukkan perubahan data, maka perintah yang dipakai adalah ‘put’

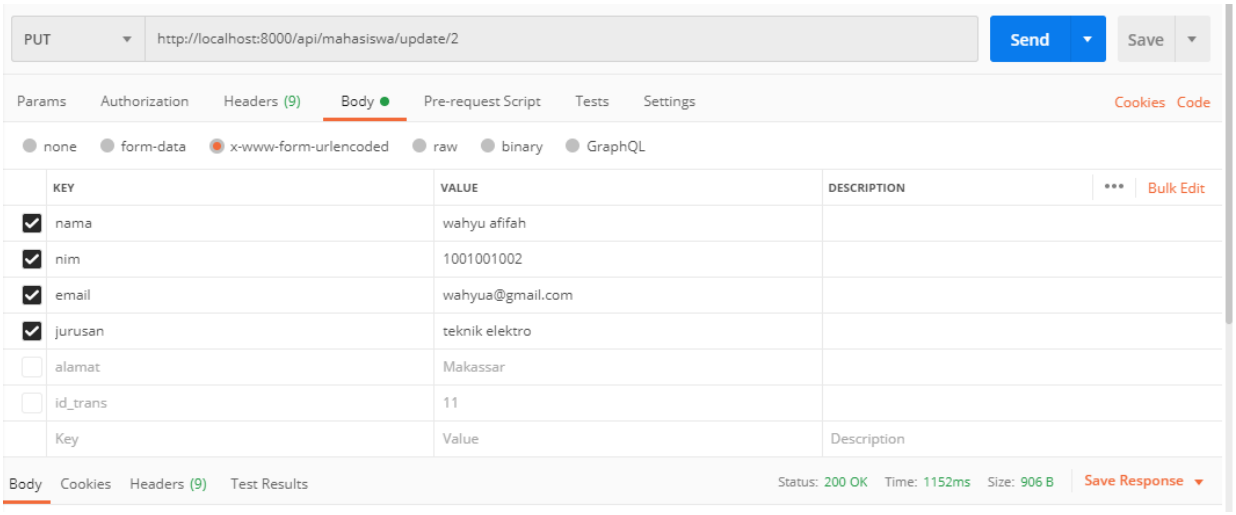
routes > api.php > ...

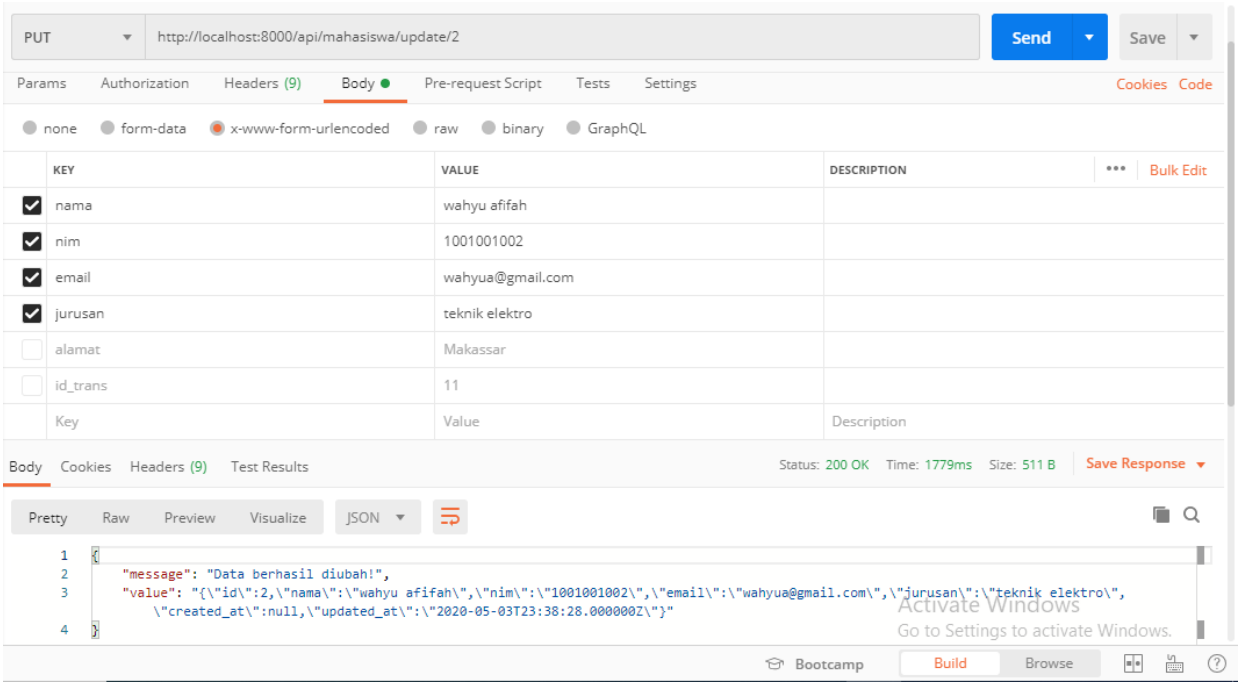
```
26
27    Route::put('/mahasiswa/update/{id}', 'MahasiswaController@update');
28
```

Sekarang kita coba untuk mengubah data berdasarkan ID mahasiswa yang dipilih menggunakan Postman. Gunakan perintah PUT untuk mengubah data.

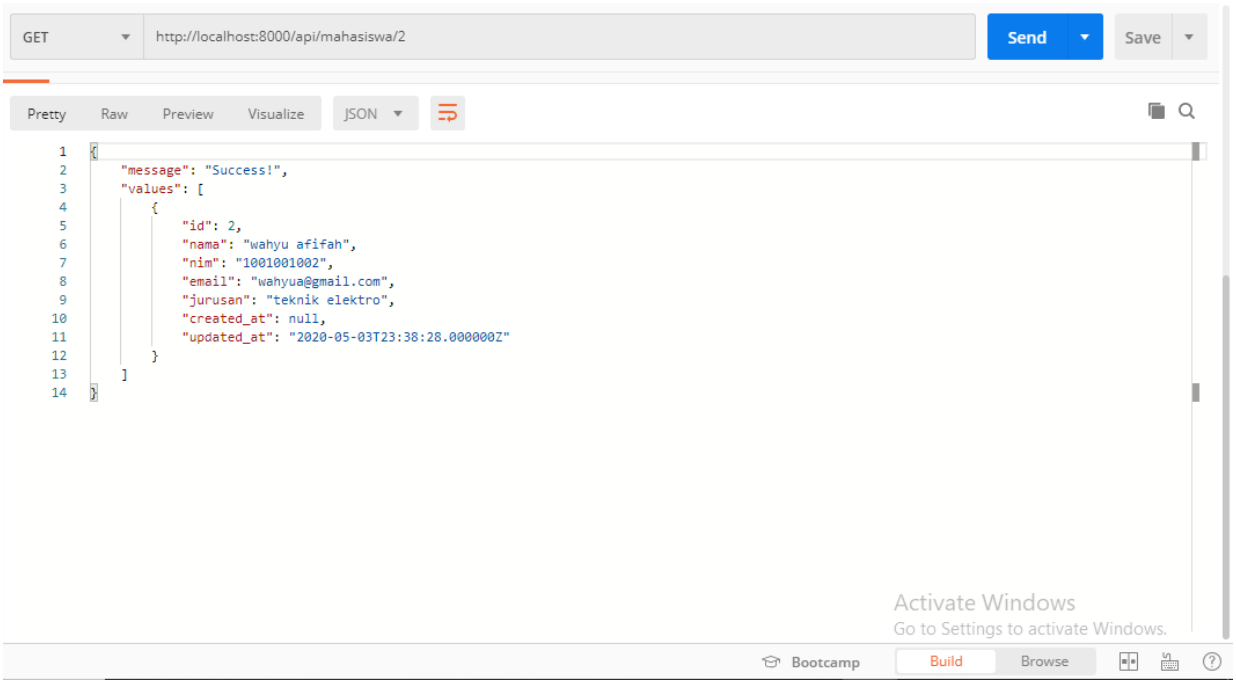
Berikut adalah contoh untuk mengubah data dengan ID=2, maka url diisi : http://localhost:8000/api/mahasiswa/update/2. Pilih tab Body dan pilih radio button xwww-form-urlencode. Isikan nama kolom pada database pada KEY, untuk isian data yang diubah tuliskan pada VALUE.

Akan muncul pesan berhasil serta perubahan data dari ID=2.





Kemudian coba untuk menampilkan data dengan ID=2 untuk melihat apakah data sudah ter-update.



Terakhir kita akan membuat fungsi untuk menghapus data dengan nama delete di MahasiswaController.php.

Keterangan:

• Fungsi delete menerima parameter id yang menunjukkan ID yang dipilih.

• Line 92-99 : $mhs->delete() digunakan untuk menghapus data dari database, apabila delete() berhasil dijalankan maka akan ditampilkan pesan berhasil.

app > Http > Controllers > MahasiswaController.php > PHP Intelephense > MahasiswaController > update

```
76      public function delete($id)
77      {
78          $mhs = Mahasiswa::where('id', $id);
79
80          if ($mhs->delete()) {
81              $res['message'] = "Data berhasil dihapus!";
82              return response($res);
83          } else {
84              $res['message'] = "Gagal!";
85              return response($res);
86          }
87      }
88  }
89
```

Tambahkan route untuk memanggil fungsi delete pada routes/api.php

Karena kita ingin menghapus data, maka perintah yang dipakai adalah ‘delete.

routes > api.php > ...

```
28
29  Route::delete('/mahasiswa/{id}', 'MahasiswaController@delete');
30
```

Buka Postman untuk mencoba menghapus data dari database. Gunakan perintah DELETE untuk mengubah data.

Berikut adalah contoh untuk menghapus data dengan ID=10, maka url diisi :

http://localhost:8000/api/mahasiswa /10

Muncul pesan berhasil ketika data terhapus dari database.