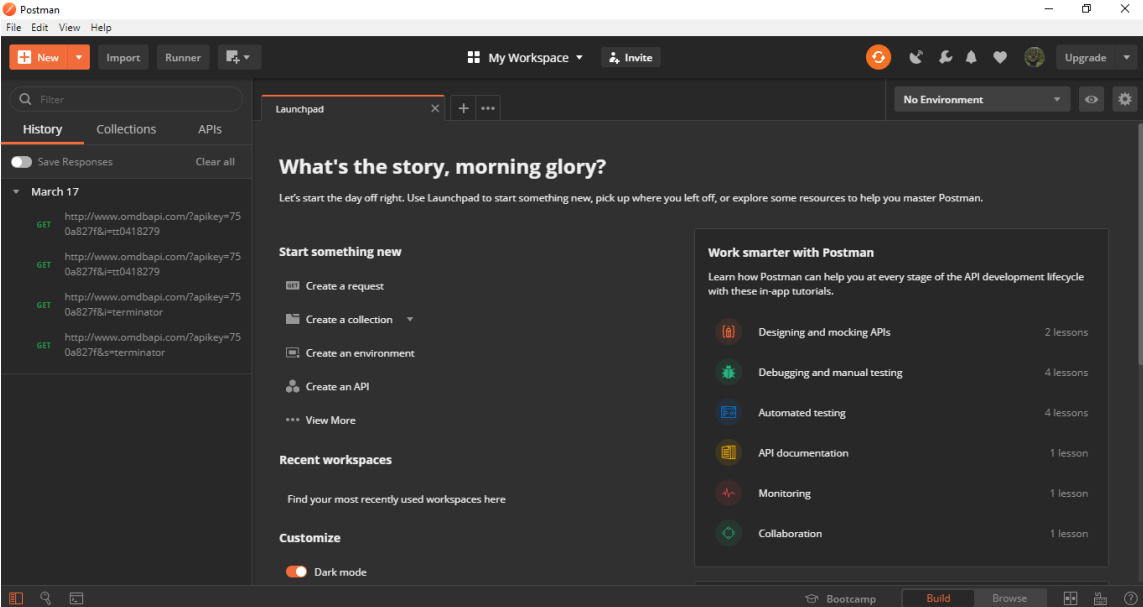
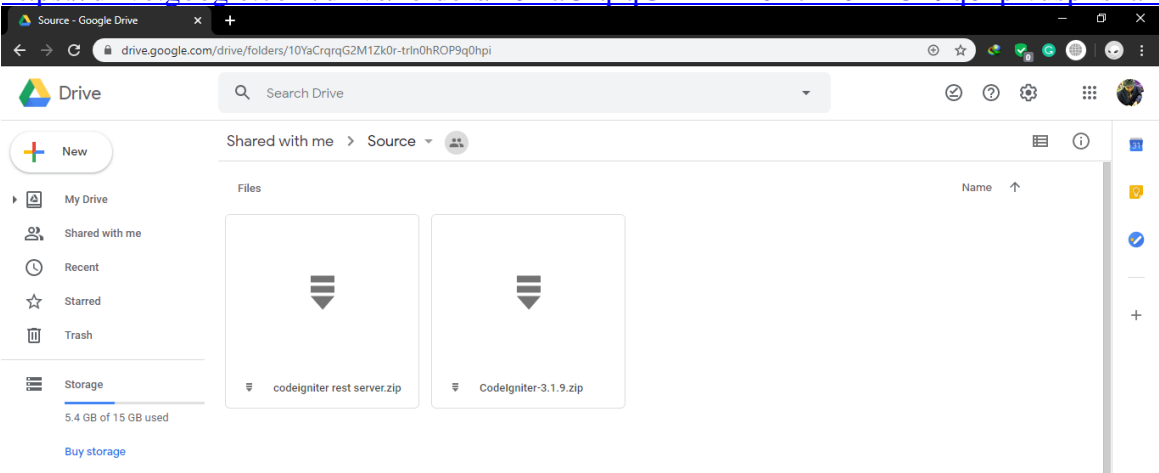




Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-8: Implementasi Restful API
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
Maret 2020

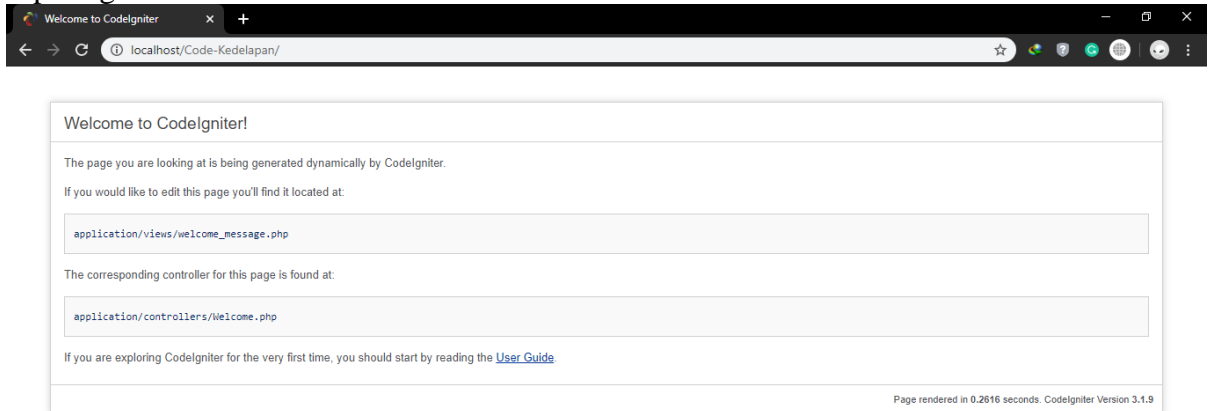
Nama : Septianda Reza Maulana
Kelas : TI-2A
NIM : 1841720135

Praktikum – Bagian 1: Instalasi library restful codeigniter

Langkah	Keterangan
1	<p>Silahkan unduh aplikasi Postman terlebih dahulu.</p> 
2	<p>Pada praktikum kali ini kita akan menggunakan source codeigniter yang baru dan Library Rest Server. Untuk versi codeigniter yang digunakan adalah versi 3.1.9. sedangkan untuk Library Rest Server dapat di unduh pada link berikut :</p> <p>https://drive.google.com/drive/folders/10YaCrqrqG2M1Zk0r-trln0hROP9q0hpi?usp=sharing</p> 

3

Extract codeigniter yang sudah di unduh dan letakkan pada folder Ampmps/www. Beri nama Code-Kedelapan dan coba akses di <http://localhost/Code-Kedelapan/>. Pastikan tampilan seperti gambar dibawah ini:



4

Selanjutnya lakukan konfigurasi dasar pada **config.php** seperti berikut:

```
config.php ×
Code-Kedelapan > application > config > config.php > ...
26 $config['base_url'] = '';

config.php ×
Code-Kedelapan > application > config > config.php > ...
38 $config['index_page'] = 'index.php';
```

Menjadi seperti berikut :

```
config.php ×
Code-Kedelapan > application > config > config.php > ...
26 $config['base_url'] = 'http://localhost/rest_server/';

config.php ×
Code-Kedelapan > application > config > config.php > ...
38 $config['index_page'] = '';
```

Dan file **autoload.php**

```
autoload.php ×
Code-Kedelapan > application > config > autoload.php > ...
61 $autoload['libraries'] = array();

autoload.php ×
Code-Kedelapan > application > config > autoload.php > ...
92 $autoload['helper'] = array();
```

Menjadi :

```
autoload.php ×
Code-Kedelapan > application > config > autoload.php > ...
61 $autoload['libraries'] = array('database');
```

```
autoload.php X
Code-Kedelapan > application > config > autoload.php > ...
92 $autoload['helper'] = array('url');
```

Dan database.php seperti berikut :

```
database.php X
Code-Kedelapan > application > config > database.php > ...

76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'rest_server',
82     'dbdriver' => 'mysqli',
```

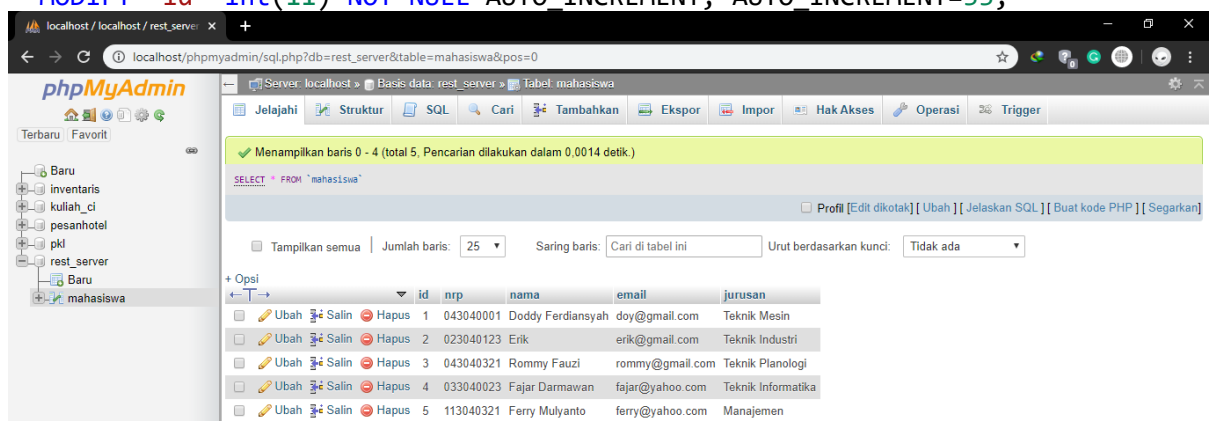
- 5 Selanjutnya buatlah database **rest_server** pada phpmyadmin dan masukkan data dummy. Untuk data dummy dapat dibuat sebagai berikut:

```
CREATE TABLE `mahasiswa` (
  `id` int(11) NOT NULL,
  `nrp` char(9) NOT NULL,
  `nama` varchar(250) NOT NULL,
  `email` varchar(250) DEFAULT NULL,
  `jurusan` varchar(64) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `mahasiswa` (`id`, `nrp`, `nama`, `email`, `jurusan`) VALUES
(1, '043040001', 'Doddy Ferdiansyah', 'doy@gmail.com', 'Teknik Mesin'),
(2, '023040123', 'Erik', 'erik@gmail.com', 'Teknik Industri'),
(3, '043040321', 'Rommy Fauzi', 'rommy@gmail.com', 'Teknik Planologi'),
(4, '033040023', 'Fajar Darmawan ', 'fajar@yahoo.com', 'Teknik Informatika'),
(5, '113040321', 'Ferry Mulyanto', 'ferry@yahoo.com', 'Manajemen');

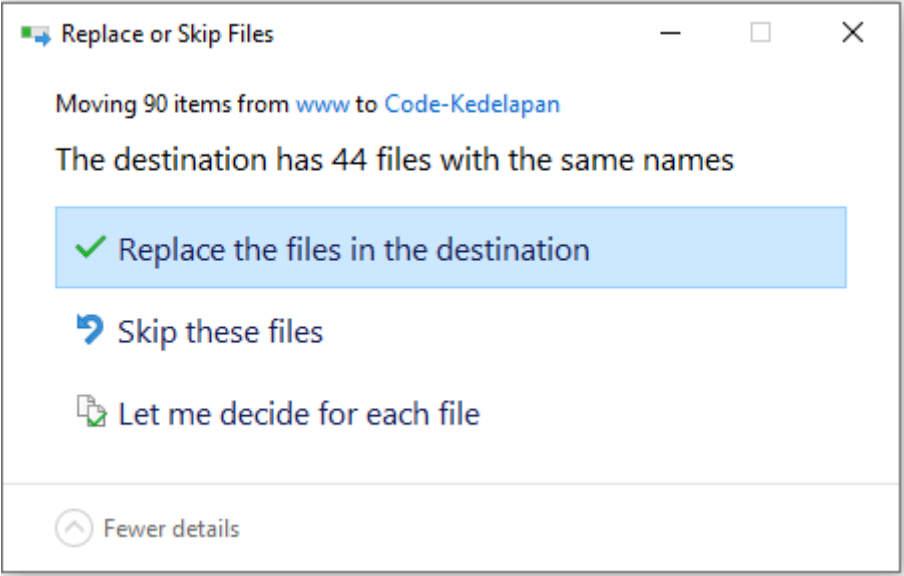
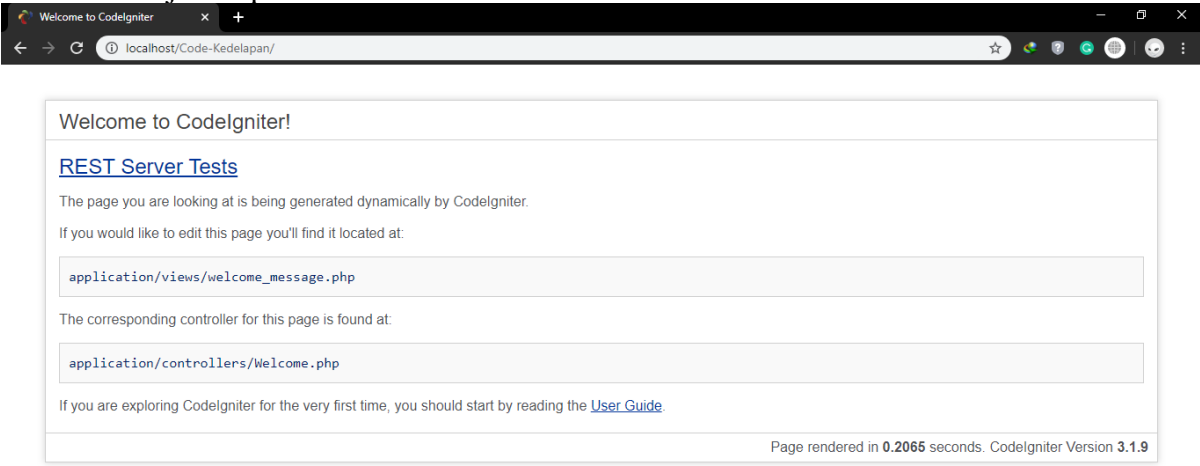
ALTER TABLE `mahasiswa`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `mahasiswa`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=33;
```



The screenshot shows the phpMyAdmin interface for the 'rest_server' database. The 'mahasiswa' table is selected, and the data is displayed in a table view. The table has 5 rows of data, each with columns for id, nrp, nama, email, and jurusan.

	id	nrp	nama	email	jurusan
	1	043040001	Doddy Ferdiansyah	doy@gmail.com	Teknik Mesin
	2	023040123	Erik	erik@gmail.com	Teknik Industri
	3	043040321	Romy Fauzi	rommy@gmail.com	Teknik Planologi
	4	033040023	Fajar Darmawan	fajar@yahoo.com	Teknik Informatika
	5	113040321	Ferry Mulyanto	ferry@yahoo.com	Manajemen

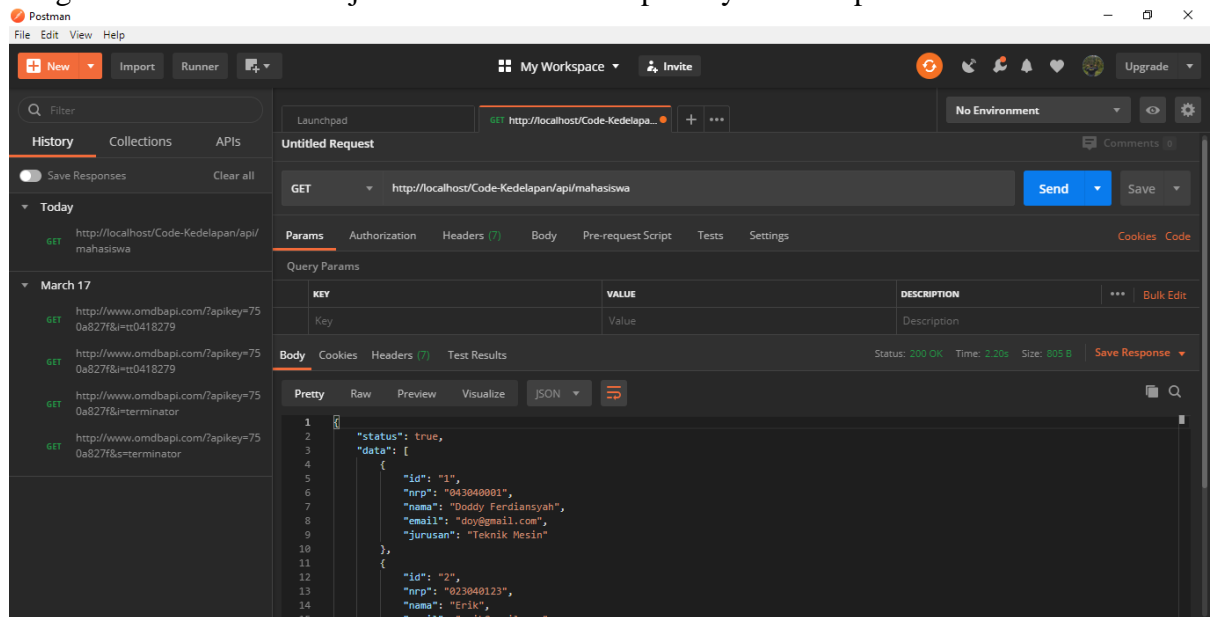
6	<p>Selanjutnya extract file Library Rest Server, buka folder application dan duplikasi file dan folder tersebut kedalam folder application pada project Code-Kedelapan. Klik Replace files jika ada notifikasi duplikat.</p>  <p>Setelah proses duplikasi file selesai, pastikan file-file yang kita sudah konfigurasi di awal tetap terkonfigurasi.</p>
7	<p>Selanjutnya buka project http://localhost/Code-Kedelapan/. Jika berhasil, tampilan akan berubah menjadi seperti ini:</p> 
8	Tahap instalasi Library Rest Server selesai

Praktikum – Bagian 2: Membuat Aplikasi CRUD Restful Codeigniter

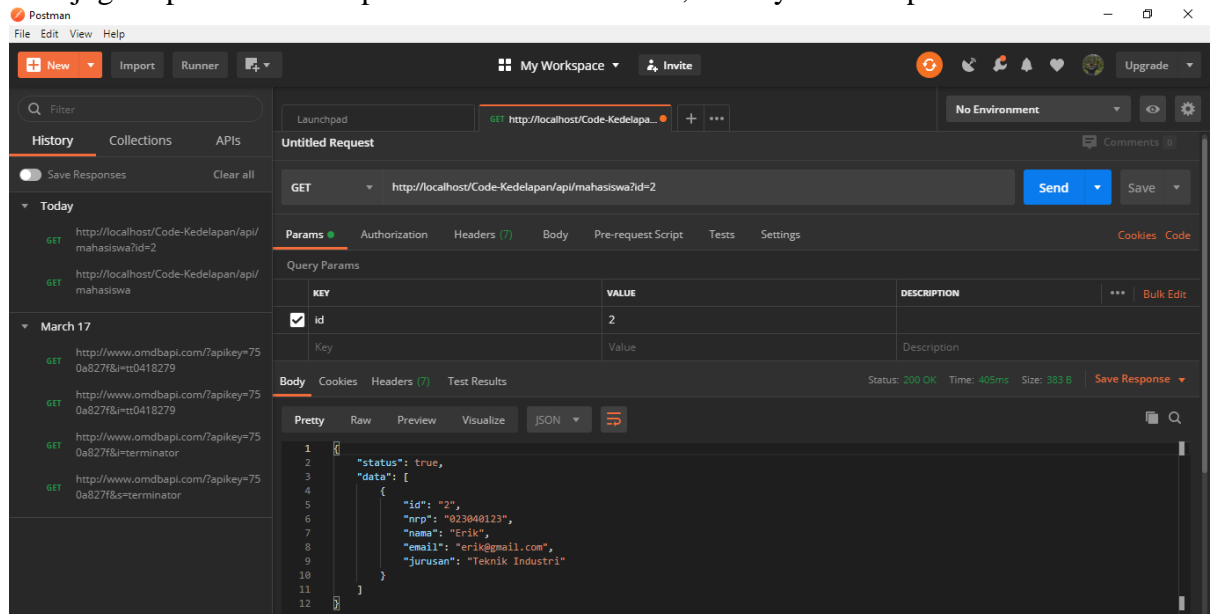
Langkah	Keterangan
1	<p>Buatlah file Mahasiswa.php pada controller/api. Selanjutnya tuliskan kode berikut:</p> <pre> Mahasiswa.php X Code-Kedelapan > application > controllers > api > Mahasiswa.php > Mahasiswa 1 <?php 2 3 defined('BASEPATH') OR exit('No direct script access allowed'); 4 5 require APPPATH . 'libraries/REST_Controller.php'; 6 require APPPATH . 'libraries/Format.php'; 7 8 class Mahasiswa extends REST_Controller { 9 public function __construct() { 10 parent::__construct(); 11 \$this->load->model('Mahasiswa_model', 'mahasiswa'); 12 } 13 public function index_get() { 14 \$id = \$this->get('id'); 15 if(\$id === null) { 16 \$mahasiswa = \$this->mahasiswa->getMahasiswa(); 17 }else { 18 \$mahasiswa = \$this->mahasiswa->getMahasiswa(\$id); 19 } 20 21 if(\$mahasiswa) { 22 \$this->response([23 'status' => true, 24 'data' => \$mahasiswa 25], REST_Controller::HTTP_OK); 26 }else { 27 \$this->response([28 'status' => false, 29 'message' => 'id not found' 30], REST_Controller::HTTP_NOT_FOUND); 31 } 32 } 33 }</pre>
2	<p>Selanjutnya Buatlah file Mahasiswa_model.php pada model. Selanjutnya tuliskan kode berikut:</p> <pre> Mahasiswa_model.php X Code-Kedelapan > application > models > Mahasiswa_model.php > Mahasiswa_model 1 <?php 2 3 class Mahasiswa_model extends CI_model { 4 public function getMahasiswa(\$id = null) { 5 if(\$id === null) { 6 return \$this->db->get('mahasiswa')->result_array(); 7 }else { 8 return \$this->db->get_where('mahasiswa', ['id'=> \$id])->result_array(); 9 } 10 } 11 }</pre>

3

Buka aplikasi postman, masukkan alamat <http://localhost/Code-Kedelapan/api/mahasiswa> dengan metode **GET**. dan jika berhasil maka tampilannya akan seperti ini:



Kita juga dapat melakukan pencarian berdasarkan id, hasilnya akan seperti ini:



4

Untuk fungsi lain **Create**, **Update** dan **Delete**. Lanjutkan kode berikut pada file **mahasiswa.php**:

```

Mahasiswa.php X
Code-Kedelapan > application > controllers > api > Mahasiswa.php > index_delete
33 public function index_delete() {
34     $id = $this->delete('id');
35
36     if($id === null) {
37         $this->response([
38             'status' => false,
39             'message' => 'provide an id!'
40         ], REST_Controller::HTTP_BAD_REQUEST);
41     }else {
42         if($this->mahasiswa->deleteMahasiswa($id)>0) {
43             //ok
44             $this->response([
45                 'status' => true,
46                 'id' => $id,
47                 'message' => 'deleted.'
48             ], REST_Controller::HTTP_OK);
49         }else {
50             // id not found
51             $this->response([
52                 'status' => false,
53                 'message' => 'id not found!'
54             ], REST_Controller::HTTP_BAD_REQUEST);
55         }
56     }
57 }

Mahasiswa.php X
Code-Kedelapan > application > controllers > api > Mahasiswa.php > Mahasiswa
58 public function index_post() {
59     $data = [
60         'nrp' => $this->post('nrp'),
61         'nama' => $this->post('nama'),
62         'email' => $this->post('email'),
63         'jurusan' => $this->post('jurusan'),
64     ];
65
66     if($this->mahasiswa->createMahasiswa($data) > 0) {
67         $this->response([
68             'status' => true,
69             'message' => 'new mahasiswa has been created'
70         ], REST_Controller::HTTP_CREATED);
71     }else {
72         // id not found
73         $this->response([
74             'status' => false,
75             'message' => 'failed to create new data!'
76         ], REST_Controller::HTTP_BAD_REQUEST);
77     }
78 }
79

```

```
Mahasiswa.php X
Code-Kedelapan > application > controllers > api > Mahasiswa.php > Mahasiswa

79 public function index_put() {
80     $id = $this->put('id');
81     $data = [
82         'nrp' => $this->put('nrp'),
83         'nama' => $this->put('nama'),
84         'email' => $this->put('email'),
85         'jurusan' => $this->put('jurusan'),
86     ];
87
88     if($this->mahasiswa->updateMahasiswa($data, $id) > 0) {
89         $this->response([
90             'status' => true,
91             'message' => 'Data mahasiswa has been updated'
92         ], REST_Controller::HTTP_CREATED);
93     }else {
94         // id not found
95         $this->response([
96             'status' => false,
97             'message' => 'Failed to update data!'
98         ], REST_Controller::HTTP_BAD_REQUEST);
99     }
100 }
101 }
```

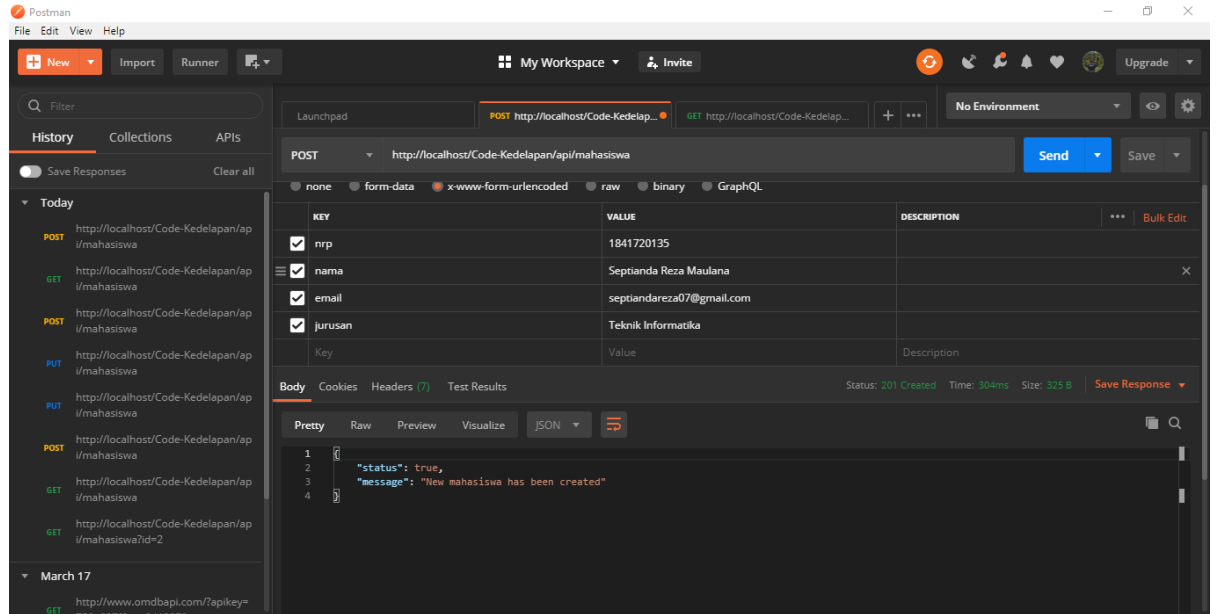
5 Dan lanjutkan kode pada file **Mahasiswa_model.php**:

```
Mahasiswa_model.php X
Code-Kedelapan > application > models > Mahasiswa_model.php > Mahasiswa_model

11 public function deleteMahasiswa($id) {
12     $this->db->delete('mahasiswa', ['id' => $id]);
13     return $this->db->affected_rows();
14 }
15 public function createMahasiswa($data) {
16     $this->db->insert('mahasiswa', $data);
17     return $this->db->affected_rows();
18 }
19 public function updateMahasiswa($data, $id) {
20     $this->db->update('mahasiswa', $data, ['id' => $id]);
21     return $this->db->affected_rows();
22 }
23 }
```

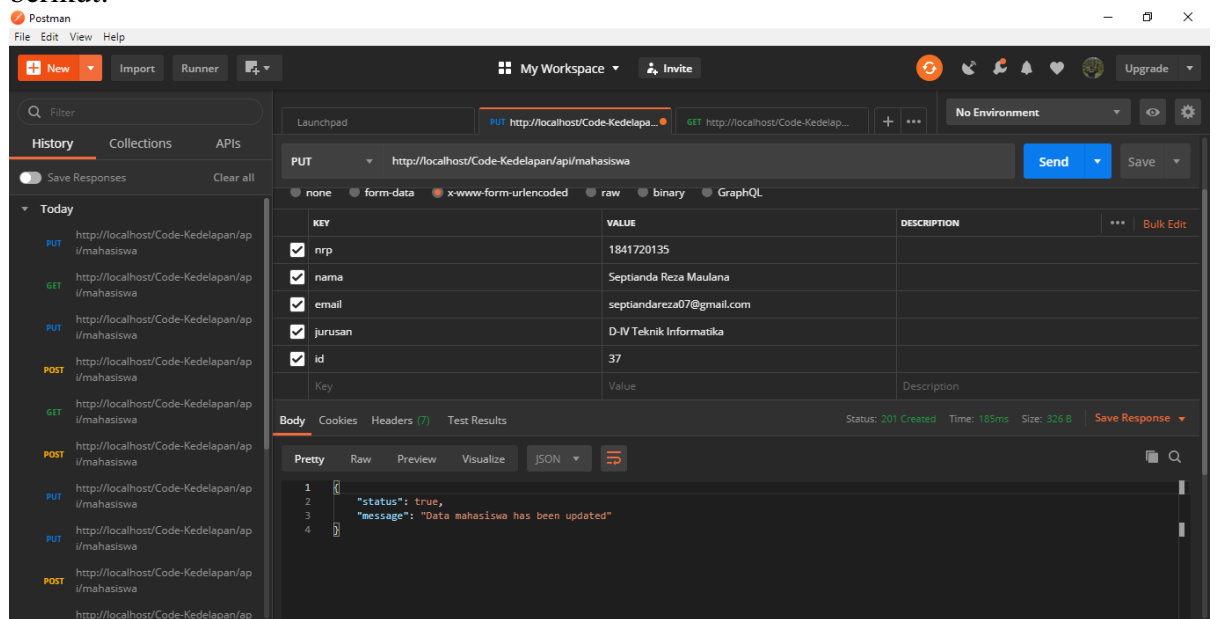

6

Selanjutnya lakukan operasi **Create** data pada aplikasi postman, untuk operasi **Create**, **Update** dan **Delete**, arahkan parameter ke **Body** dan **x-www-form-urlencoded** seperti berikut:



7

Lakukan juga operasi **Update** data pada aplikasi postman, masukkan parameter id seperti berikut:



8

Lakukan juga operasi **Delete** data pada aplikasi postman, parameter yang dimasukkan cukup id saja, seperti berikut:

