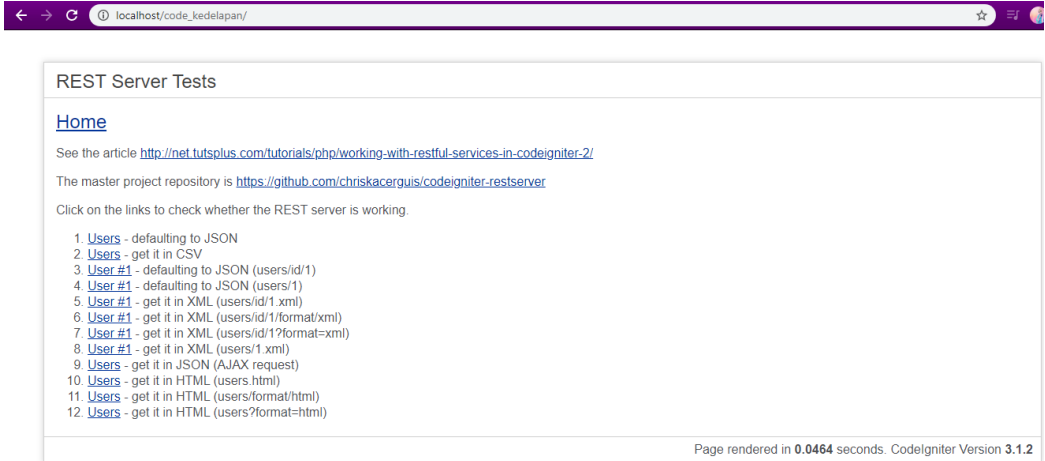




LAPORAN PRAKTIKUM PEMROGRAMAN DASAR WEB

NAMA : NUR HANIFAH
NIM : 1841720062
Kelas : TI 2B
Isi Laporan : Jobsheet8 - Implementasi Restful API

Praktikum – Bagian 1: Instalasi library restful codeigniter

Langkah	Keterangan
1	Silahkan unduh aplikasi Postman terlebih dahulu.
2	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 :
3	<p>Extract codeigniter yang sudah di unduh dan letakkan pada folder httdocs. Beri nama Code-kedelapan dan coba akses di http://localhost/Code-kedelapan/. Pastikan tampilan seperti gambar dibawah ini:</p> 
4	<p>Selanjutnya lakukan konfigurasi dasar pada config.php seperti berikut:</p> <pre>24 25 */ 26 \$config['base_url'] = ''; 27 28 /*</pre>

```

37  */
38  $config['index_page'] = 'index.php';
39
40  /*

```

Menjadi seperti berikut :

```

24  |
25  */
26  $config['base_url'] = 'http://localhost/rest_server/';
27
28  /*

```

```

37  */
38  $config['index_page'] = '';
39
40  /*

```

Dan file **autoload.php**

```

60  */
61  $autoload['libraries'] = array();
62
63  /*

```

```

91  */
92  $autoload['helper'] = array();
93
94  /*

```

Menjadi

```

60  */
61  $autoload['libraries'] = array('database');
62
63  /*

```

```

91  */
92  $autoload['helper'] = array('url');
93
94  /*

```

Dan database.php seperti berikut:

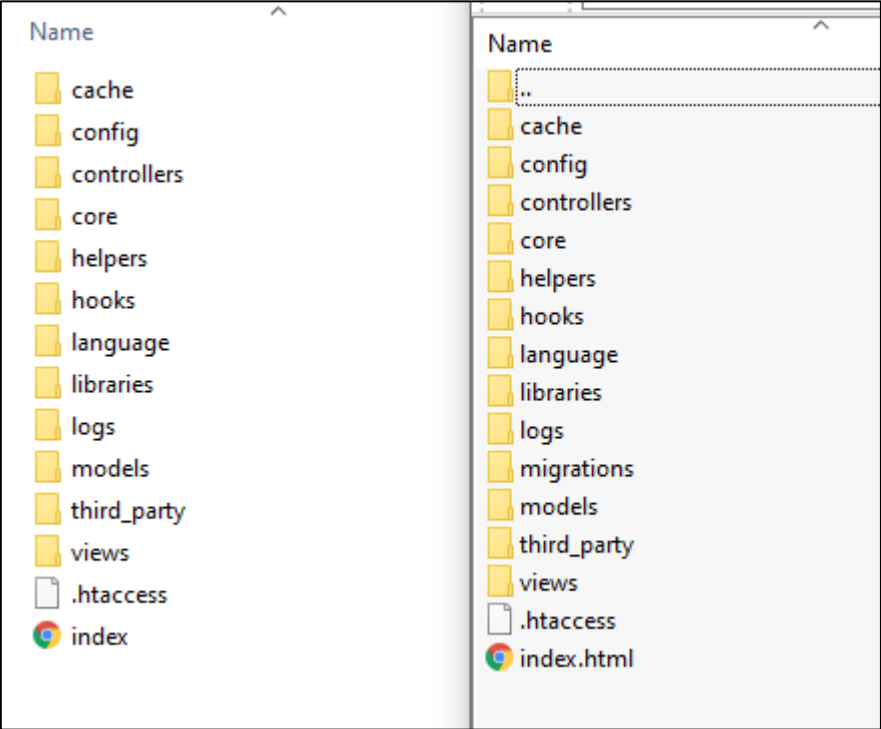
```

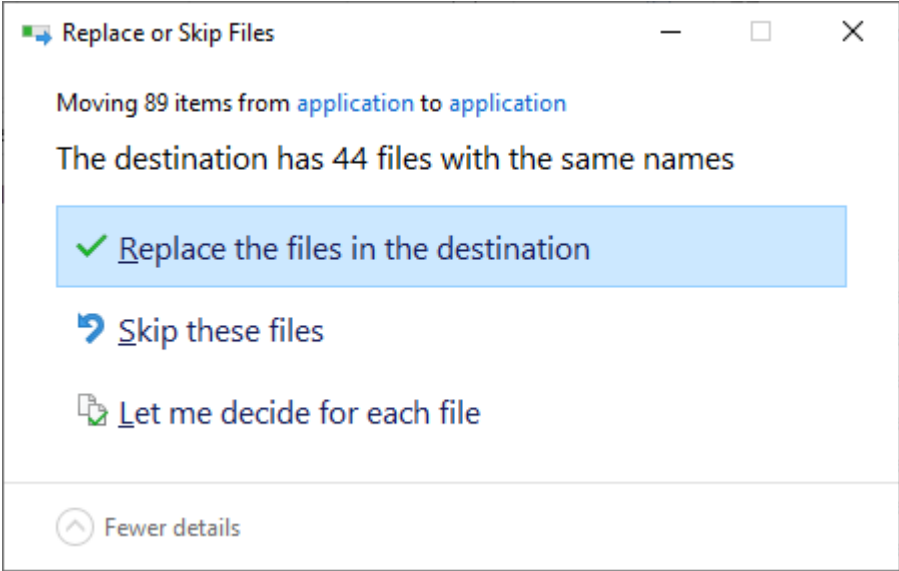
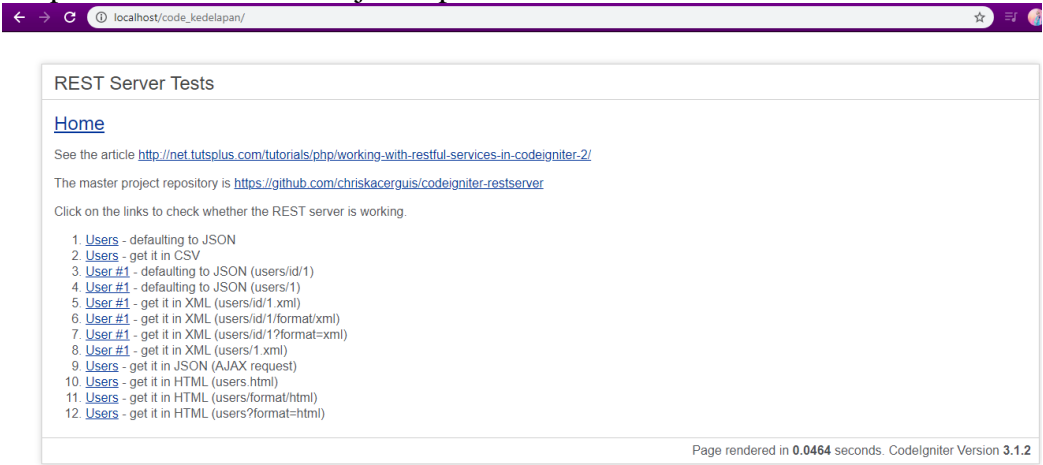
76  ▾ $db['default'] = array(
77      'dsn' => '',
78      'hostname' => 'localhost',
79      'username' => 'root',
80      'password' => '',
81      'database' => 'rest_server',
82      'dbdriver' => 'mysqli',

```

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

```
CREATE TABLE `mahasiswa` (
```

	<pre> `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; </pre>
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	Buatlah file Mahasiswa.php pada controller/api . Selanjutnya tuliskan kode berikut:

```

application > controllers > api > Mahasiswa.php > ...
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  use Restserver\Libraries\REST_Controller;
8
9  class Mahasiswa extends REST_Controller {
10
11     function __construct($config = 'rest') {
12         parent::__construct($config);
13         $this->load->model('Mahasiswa_model', 'mahasiswa');
14     }
15
16     function index_get() {
17         $id = $this->get('id');
18         if ($id === null) {
19             $mahasiswa = $this->mahasiswa->getMahasiswa();
20         } else {
21             $this->db->where('id', $id);
22             $mahasiswa = $this->mahasiswa->getMahasiswa($id);
23         }
24
25         if ($mahasiswa) {
26             $this->response([
27                 'status' => true,
28                 'data' => $mahasiswa
29             ], REST_Controller::HTTP_OK);
30         } else {
31             $this->response([
32                 'status' => true,
33                 'data' => 'id not found'
34             ], REST_Controller::HTTP_NOT_FOUND);
35         }
36     }
37

```

2

Selanjutnya Buatlah file **Mahasiswa_model.php** pada model. Selanjutnya tuliskan kode berikut:

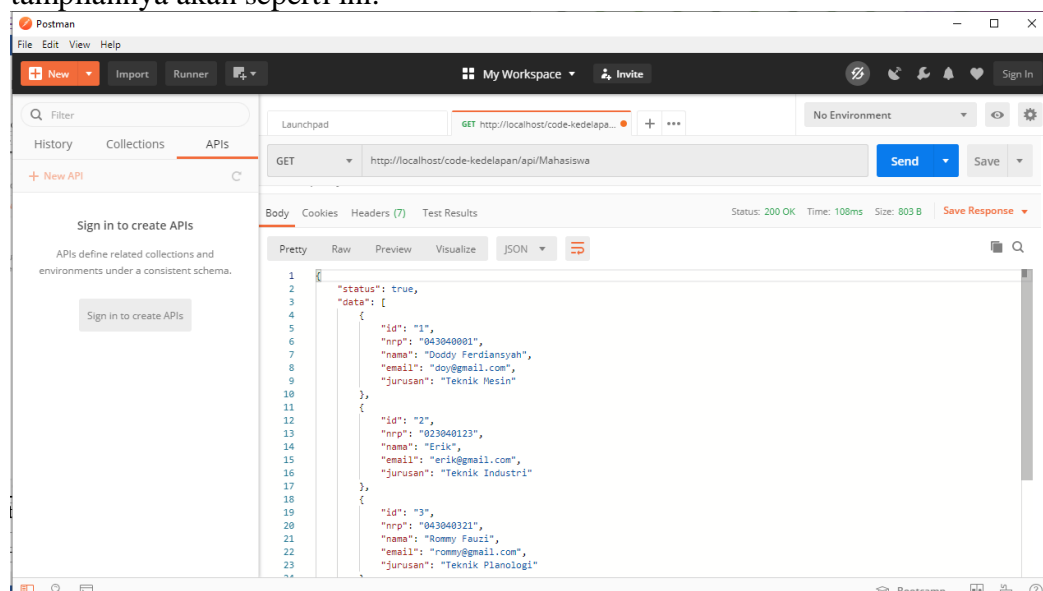
```

application > models > Mahasiswa_model.php > ...
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class Mahasiswa_model extends CI_Model {
6
7      public function getMahasiswa($id=null){
8          if ($id === null) {
9              return $this->db->get('mahasiswa')->result_array();
10         } else {
11             return $this->db->get_where('mahasiswa', ['id' => $id])->result_array();
12         }
13     }
14
15     public function deleteMahasiswa($id){
16         $this->db->delete('mahasiswa', ['id' => $id]);
17         return $this->db->affected_rows();
18     }
19
20
21     public function createMahasiswa($data){
22         $this->db->insert('mahasiswa', $data);
23         return $this->db->affected_rows();
24     }
25
26
27     public function updateMahasiswa($data, $id){
28         $this->db->update('mahasiswa', $data, ['id' => $id]);
29         return $this->db->affected_rows();
30     }
31
32
33 }

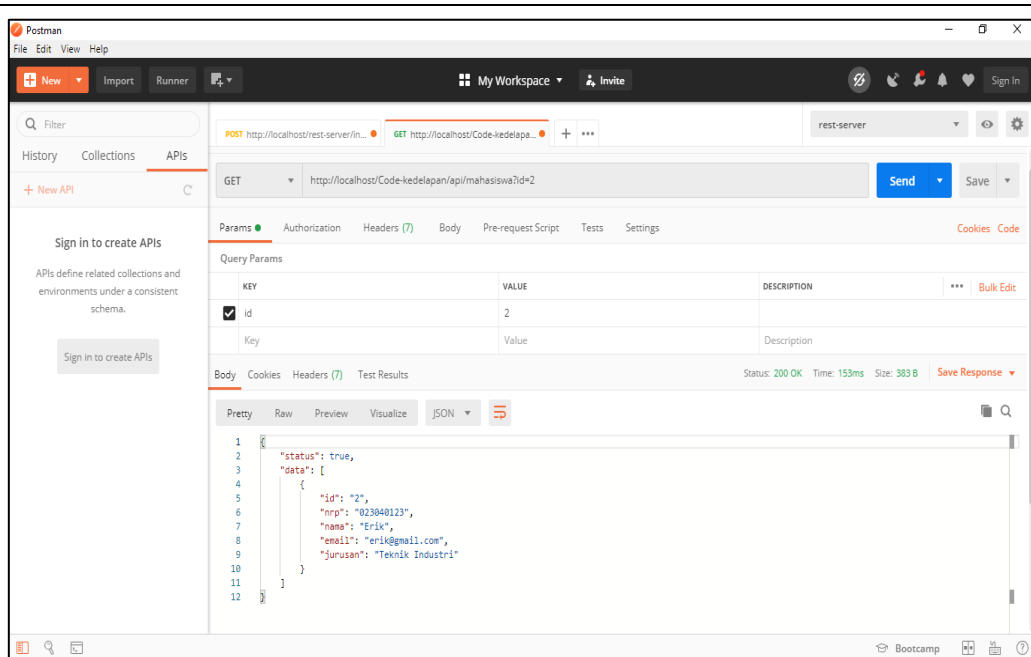
```

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**:

```

application > controllers > api > Mahasiswa.php
66     public function index_post()
67     {
68         $data = [
69             'nrp' => $this->post('nrp'),
70             'nama' => $this->post('nama'),
71             'email' => $this->post('email'),
72             'jurusan' => $this->post('jurusan'),
73         ];
74     };
75
76     if ($this->mahasiswa->createMahasiswa($data) > 0) {
77         $this->response([
78             'status' => true,
79             'message' => 'new mahasiswa has been created'
80         ], REST_Controller::HTTP_CREATED);
81     } else {
82         // id not found
83         $this->response([
84             'status' => false,
85             'message' => 'failde to create new data!'
86         ], REST_Controller::HTTP_BAD_REQUEST);
87     }
88 }
89
90 }
91

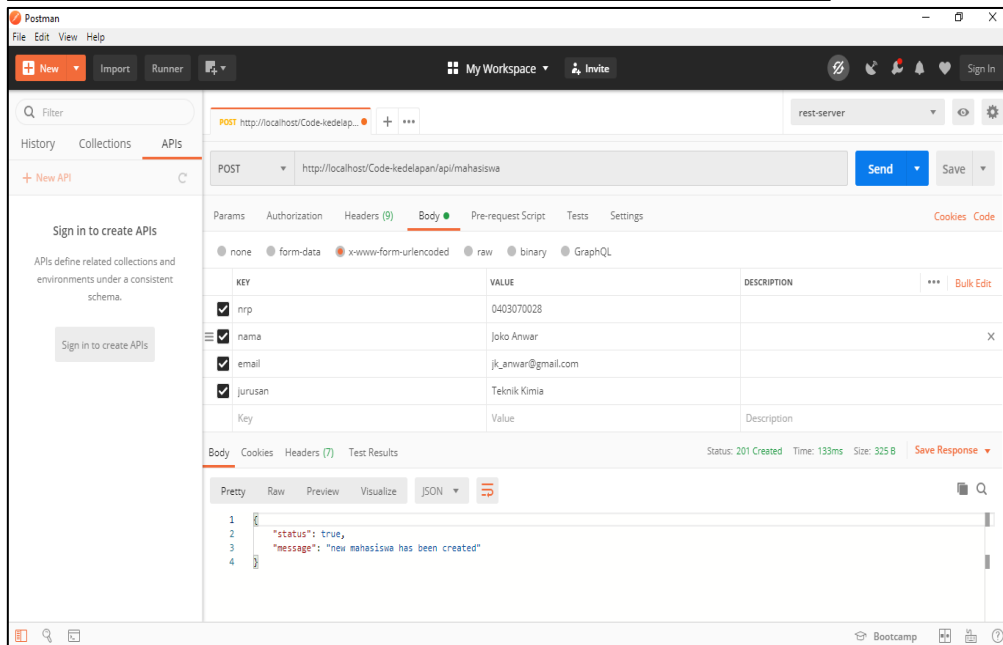
```

	<pre> application > controllers > api > Mahasiswa.php 92 public function index_put() 93 { 94 \$id = \$this->put('id'); 95 \$data = [96 'nrp' => \$this->put('nrp'), 97 'nama' => \$this->put('nama'), 98 'email' => \$this->put('email'), 99 'jurusan' => \$this->put('jurusan'), 100 101]; 102 103 if (\$this->mahasiswa->updateMahasiswa(\$data, \$id) > 0) { 104 \$this->response([105 'status' => true, 106 'message' => 'data mahasiswa has been updated' 107], REST_Controller::HTTP_OK); 108 } else { 109 // id not found 110 \$this->response([111 'status' => false, 112 'message' => 'failde to update data!' 113], REST_Controller::HTTP_BAD_REQUEST); 114 } 115 } 116 } 117] </pre>
5	<p>Dan lanjutkan kode pada file Mahasiswa_model.php:</p> <pre> application > models > Mahasiswa_model.php 15 16 public function deleteMahasiswa(\$id) 17 { 18 \$this->db->delete('mahasiswa',['id'=> \$id]); 19 return \$this->db->affected_rows(); 20 } 21 22 public function createMahasiswa(\$data) 23 { 24 \$this->db->insert('mahasiswa', \$data); 25 return \$this->db->affected_rows(); 26 } 27 28 public function updateMahasiswa(\$data, \$id) 29 { 30 \$this->db->update('mahasiswa', \$data, ['id' => \$id]); 31 return \$this->db->affected_rows(); 32 } 33 } </pre>
6	<p>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:</p>

Params Authorization Headers (9) **Body** F

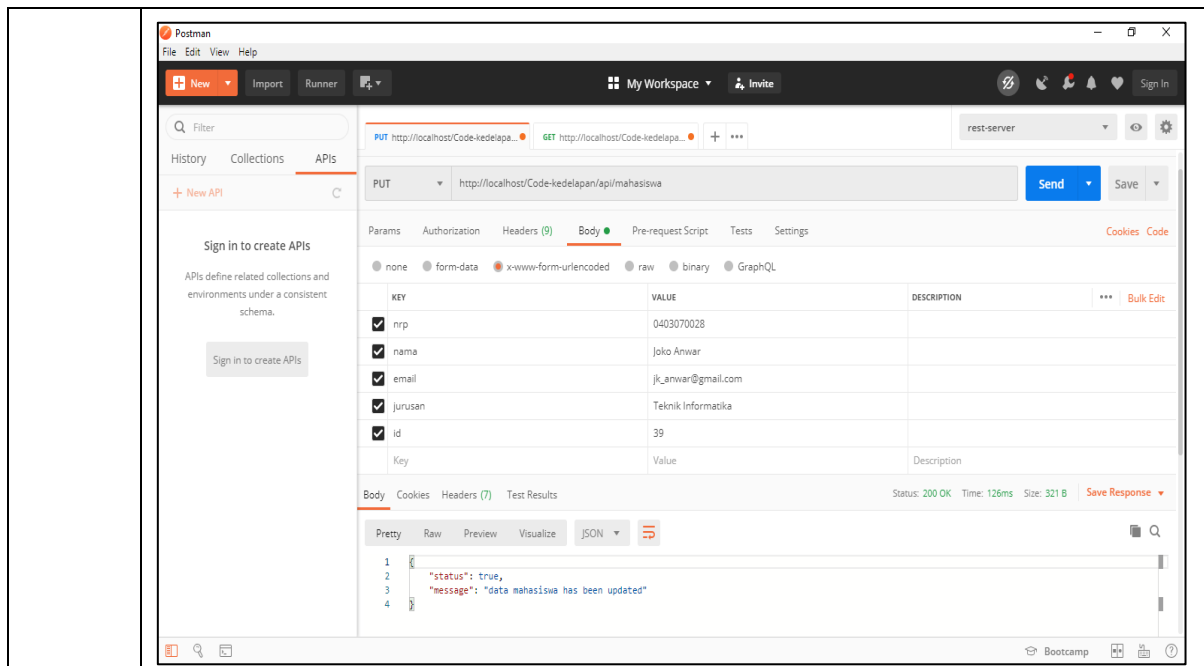
☐ none ☐ form-data ☒ x-www-form-urlencoded ☐

KEY
<input checked="" type="checkbox"/> nrp
<input checked="" type="checkbox"/> nama
<input checked="" type="checkbox"/> email
<input checked="" type="checkbox"/> jurusan



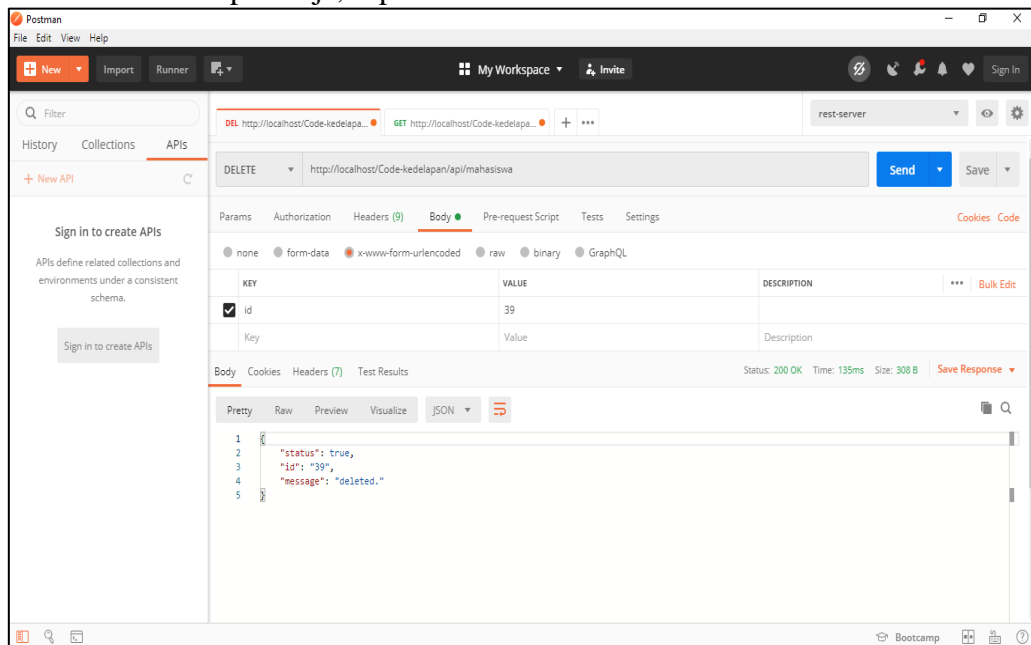
7

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:



-- Selamat Mengunjungi --