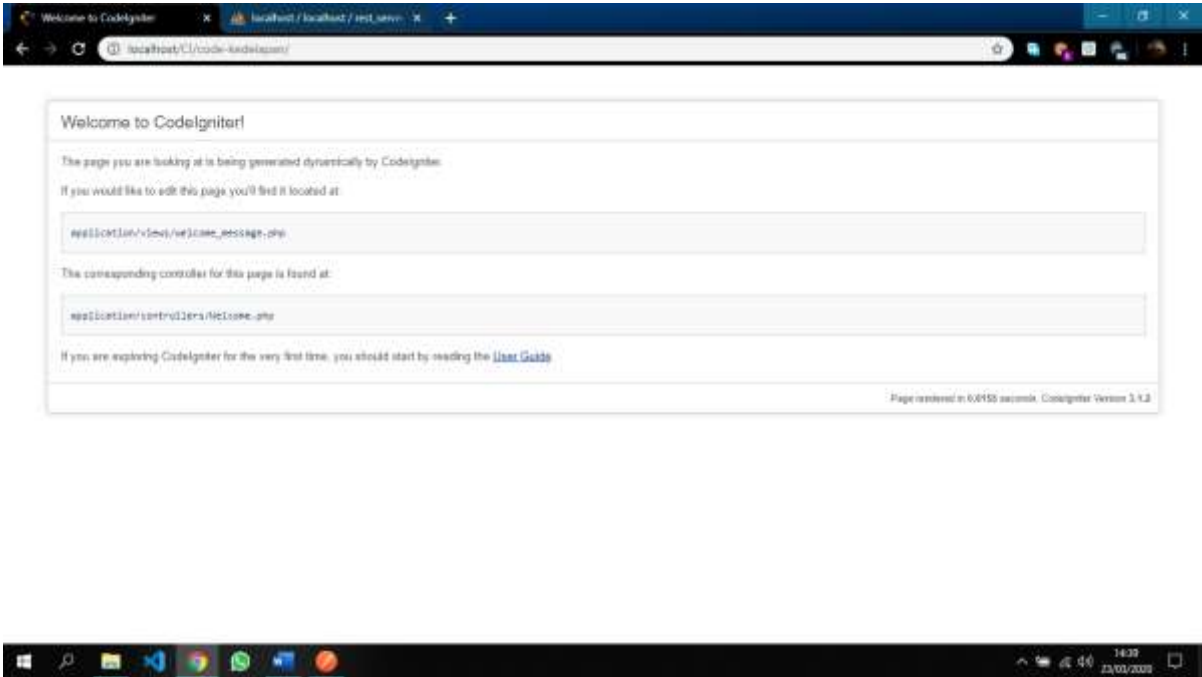




Praktikum – Bagian 1: Instalasi library restful codeigniter

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><code>*/ \$config['base_url'] = ''; ! */ \$config['index_page'] = 'index.php';</code></pre> <p>Menjadi seperti berikut :</p> <pre><code>\$config['base_url'] = 'http://localhost/CI/code-kedelapan/';</code></pre>

```
*/
$config['index_page'] = '';
```

Dan file **autoload.php**

```
$autoload['libraries'] = array();

$autoload['helper'] = array();
```

Menjadi

```
$autoload['libraries'] = array('database');

$autoload['helper'] = array('url');
```

Dan database.php seperti berikut:

```
$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'rest_server',
    'dbdriver' => 'mysqli',
```

5

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

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nrp	char(9)	latin1_swedish_ci		No	None			Change Drop More
3	nama	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
4	email	varchar(250)	latin1_swedish_ci		Yes	NULL			Change Drop More
5	jurusan	varchar(64)	latin1_swedish_ci		No	None			Change Drop More

Options

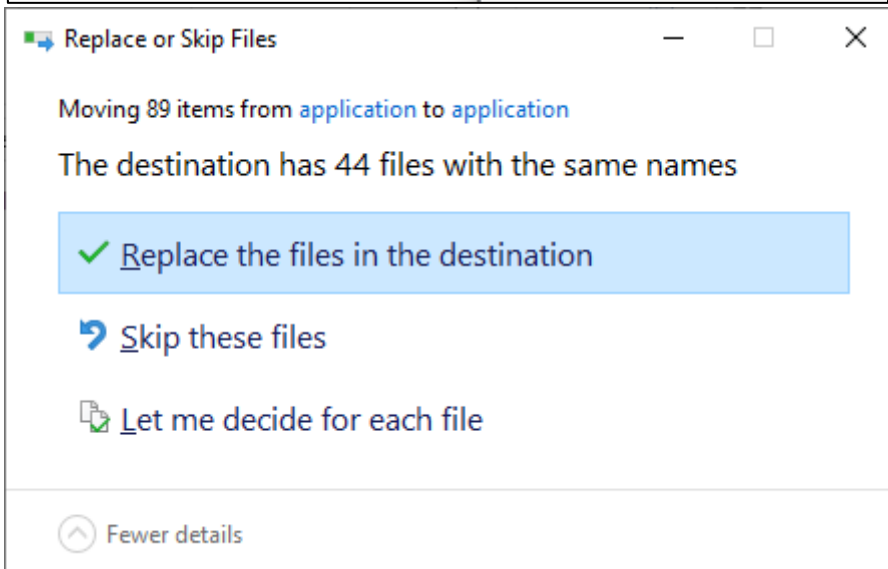
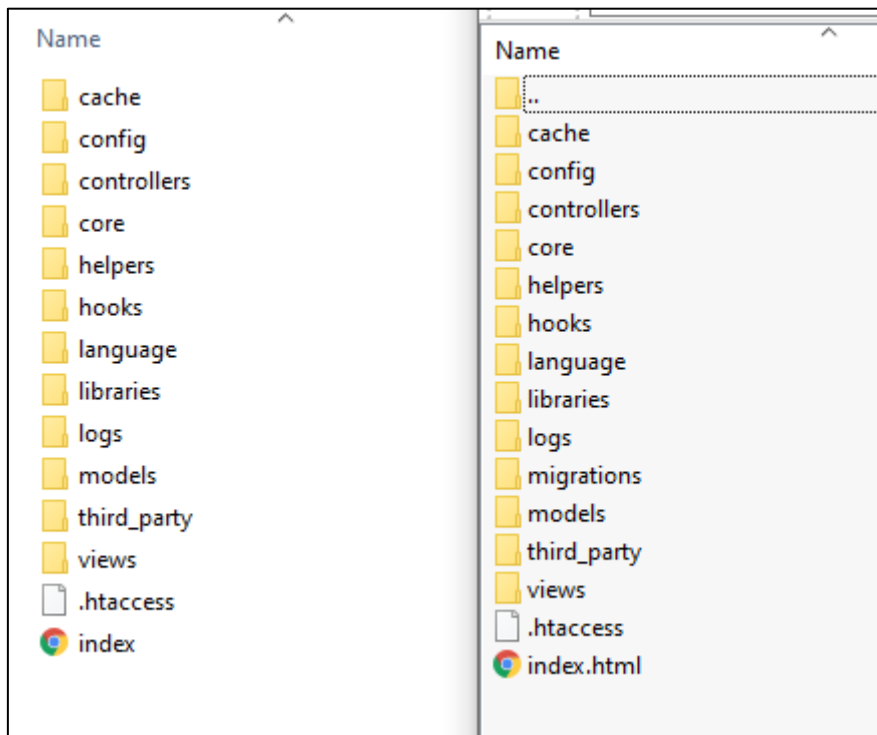
					id	nrp	nama	email	jurusan
					1	043040001	Doddy Ferdiansyah	doy@gmail.com	Teknik Mesin
					2	023040123	Erik	erik@gmail.com	Teknik Industri
					3	043040321	Rommy Fauzi	rommy@gmail.com	Teknik Planologi
					4	033040023	Fajar Darmawan	fajar@yahoo.com	Teknik Informatika
					5	113040321	Ferry Mulyanto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

6

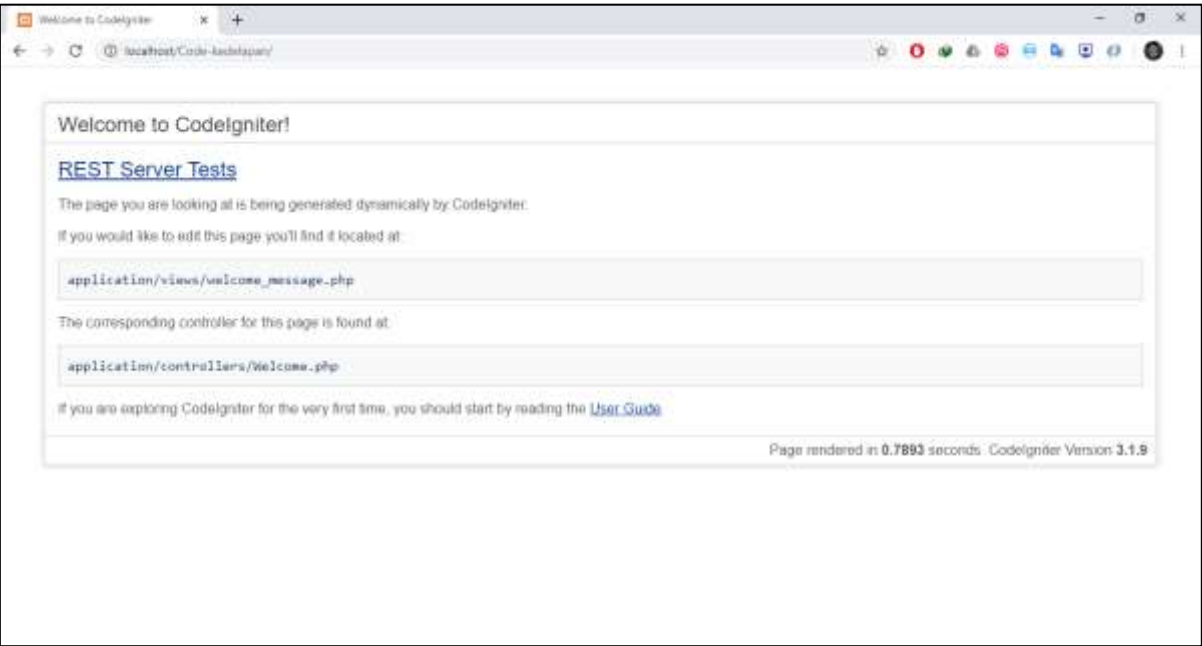
Selanjutnya extract file Library Rest Server, buka folder **application** dan duplikasi file dan folder tersebut kedalam folder **application** pada project **Code-kedelapan**. Klik **Replace files** jika ada notifikasi duplikat.



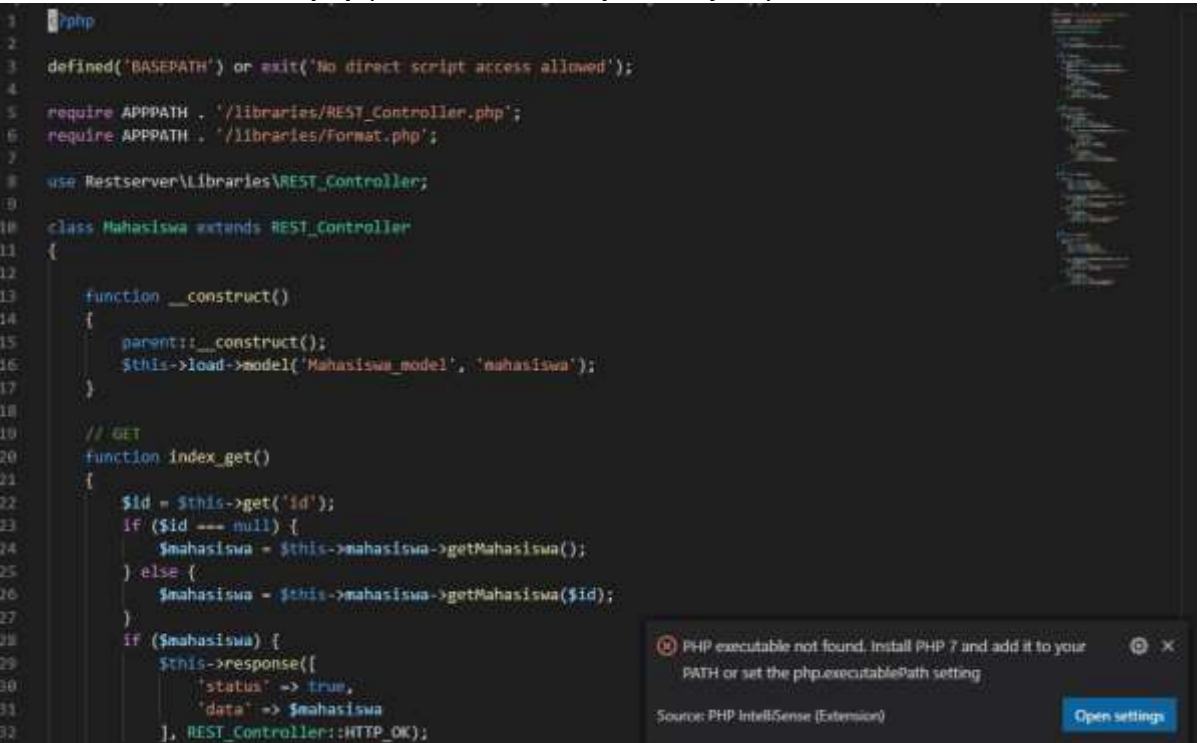
Setelah proses duplikasi file selesai, pastikan file-file yang kita sudah konfigurasi di awal tetap terkonfigurasi.

7

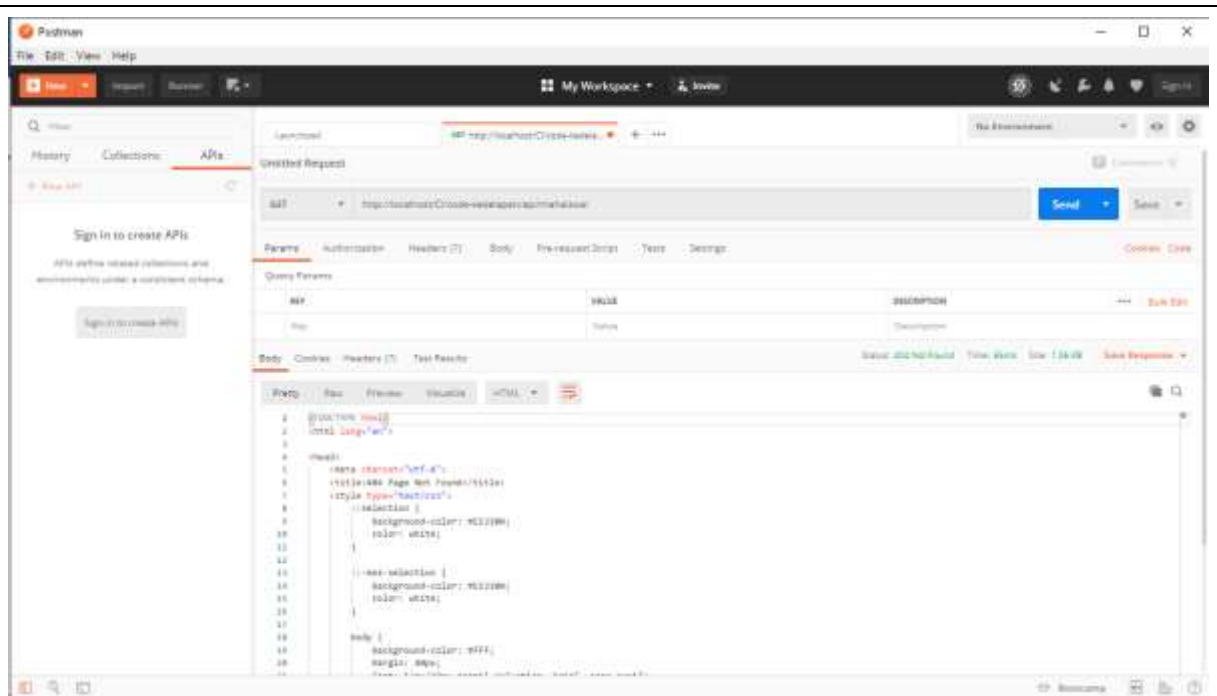
Selanjutnya buka project <http://localhost/Code-kedelapan/>. Jika berhasil, tampilan akan berubah menjadi seperti ini:

	
8	Tahap instalasi Library Rest Server selesai

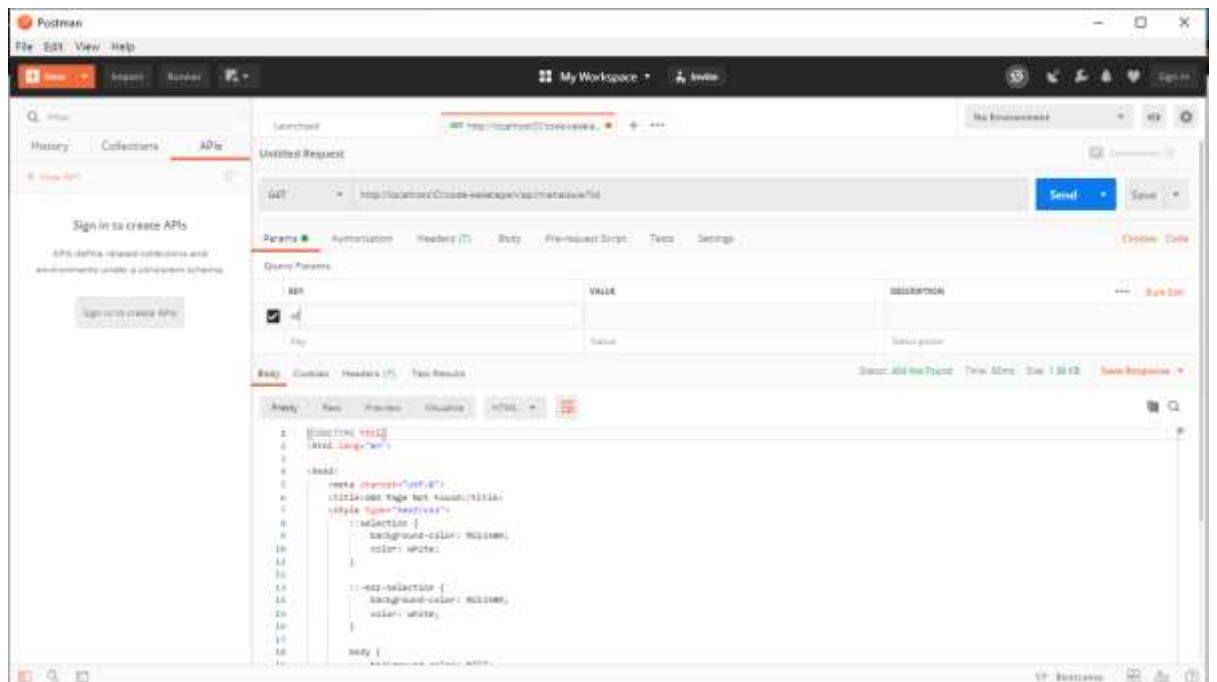
Praktikum – Bagian 2: Membuat Aplikasi CRUD Restful Codeigniter

Langkah	Keterangan
1	<p>Buatlah file Mahasiswa.php pada controller/api. Selanjutnya tuliskan kode berikut:</p> <pre> 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 use Restserver\libraries\REST_Controller; 9 10 class Mahasiswa extends REST_Controller 11 { 12 13 function __construct() 14 { 15 parent::__construct(); 16 \$this->load->model('Mahasiswa_model', 'mahasiswa'); 17 } 18 19 // GET 20 function index_get() 21 { 22 \$id = \$this->get('id'); 23 if (\$id === null) { 24 \$mahasiswa = \$this->mahasiswa->getMahasiswa(); 25 } else { 26 \$mahasiswa = \$this->mahasiswa->getMahasiswa(\$id); 27 } 28 if (\$mahasiswa) { 29 \$this->response([30 'status' => true, 31 'data' => \$mahasiswa 32], REST_Controller::HTTP_OK); </pre> 

	<pre> \$mahasiswa = \$this->mahasiswa->getMahasiswa(); } else { \$mahasiswa = \$this->mahasiswa->getMahasiswa(\$id); } if (\$mahasiswa) { \$this->response(['status' => true, 'data' => \$mahasiswa], REST_Controller::HTTP_OK); } else { \$this->response(['status' => false, 'message' => 'id not found'], REST_Controller::HTTP_NOT_FOUND); } } </pre>
2	<p>Selanjutnya Buatlah file Mahasiswa_model.php pada model. Selanjutnya tuliskan kode berikut:</p> <pre> application > models > Mahasiswa_model.php > ... 1 <?php 2 3 4 defined('BASEPATH') or exit('No direct script access allowed'); 5 6 class Mahasiswa_model extends CI_Model 7 { 8 9 public function getMahasiswa(\$id = null) 10 { 11 if (\$id === null) { 12 return \$this->db->get('mahasiswa')->result_array(); 13 } else { 14 return \$this->db->get_where('mahasiswa', ['id' => \$id])->result_array(); 15 } 16 } 17 18 public function deleteMahasiswa(\$id) 19 { 20 \$this->db->delete('mahasiswa', ['id' => \$id]); 21 return \$this->db->affected_rows(); 22 } 23 24 public function createMahasiswa(\$data) 25 { 26 \$this->db->insert('mahasiswa', \$data); 27 return \$this->db->affected_rows(); 28 } 29 30 public function updateMahasiswa(\$data, \$id) 31 { 32 \$this->db->update('mahasiswa', \$data, ['id' => \$id]); 33 return \$this->db->affected_rows(); 34 } 35 } 36 37 /* End of file Mahasiswa_model.php */ 38 </pre> <p>PHP executable not found PATH or set the php.exec</p>
3	<p>Buka aplikasi postman, masukkan alamat http://localhost/Code-kedelapan/api/mahasiswa dengan metode GET. dan jika berhasil maka tampilannya akan seperti ini:</p>



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**:

```

41 // DELETE
42 function index_delete()
43 {
44     $id = $this->delete('id');
45     if ($id === null) {
46         $this->response([
47             'status' => false,
48             'message' => 'provide an id!'
49         ], REST_Controller::HTTP_BAD_REQUEST);
50     } else {
51         if ($this->mahasiswa->deleteMahasiswa($id) > 0) {
52             //ok
53             $this->response([
54                 'status' => true,
55                 'id' => $id,
56                 'message' => 'deleted.'
57             ], REST_Controller::HTTP_OK);
58         } else {
59             //id not found
60             $this->response([
61                 'status' => false,
62                 'message' => 'provide an id!'
63             ], REST_Controller::HTTP_BAD_REQUEST);
64         }
65     }
66 }
67

```



```

68 // POST
69 public function index_post()
70 {
71     $data = [
72         'nrp' => $this->post('nrp'),
73         'nama' => $this->post('nama'),
74         'email' => $this->post('email'),
75         'jurusan' => $this->post('jurusan'),
76     ];
77
78     if ($this->mahasiswa->createMahasiswa($data) > 0) {
79         $this->response([
80             'status' => true,
81             'message' => 'new mahasiswa has been created'
82         ], REST_Controller::HTTP_CREATED);
83     } else {
84         //id not found
85         $this->response([
86             'status' => false,
87             'message' => 'failed to create new data!'
88         ], REST_Controller::HTTP_BAD_REQUEST);
89     }
90 }
91
92
93 // PUT
94 public function index_put()
95 {
96     $id = $this->put('id');
97     $data = [
98         'nrp' => $this->put('nrp'),
99         'nama' => $this->put('nama'),
100         'email' => $this->put('email'),
101         'jurusan' => $this->put('jurusan'),
102     ];
103
104     if ($this->mahasiswa->updateMahasiswa($data, $id) > 0) {
105         $this->response([
106             'status' => true,
107             'message' => 'data mahasiswa has been updated'
108         ], REST_Controller::HTTP_OK);
109     } else {
110         //id not found
111         $this->response([
112             'status' => false,
113             'message' => 'failed to update data!'
114         ], REST_Controller::HTTP_BAD_REQUEST);
115     }
116 }
117 }
118

```



```

68 // POST
69 public function index_post()
70 {
71     $data = [
72         'nrp' => $this->post('nrp'),
73         'nama' => $this->post('nama'),
74         'email' => $this->post('email'),
75         'jurusan' => $this->post('jurusan'),
76     ];
77
78     if ($this->mahasiswa->createMahasiswa($data) > 0) {
79         $this->response([
80             'status' => true,
81             'message' => 'new mahasiswa has been created'
82         ], REST_Controller::HTTP_CREATED);
83     } else {
84         //id not found
85         $this->response([
86             'status' => false,
87             'message' => 'failed to create new data!'
88         ], REST_Controller::HTTP_BAD_REQUEST);
89     }
90 }
91
92
93 // PUT
94 public function index_put()
95 {
96     $id = $this->put('id');
97     $data = [
98         'nrp' => $this->put('nrp'),
99         'nama' => $this->put('nama'),
100         'email' => $this->put('email'),
101         'jurusan' => $this->put('jurusan'),
102     ];
103
104     if ($this->mahasiswa->updateMahasiswa($data, $id) > 0) {
105         $this->response([
106             'status' => true,
107             'message' => 'data mahasiswa has been updated'
108         ], REST_Controller::HTTP_OK);
109     } else {
110         //id not found
111         $this->response([
112             'status' => false,
113             'message' => 'failed to update data!'
114         ], REST_Controller::HTTP_BAD_REQUEST);
115     }
116 }
117 }
118

```

5

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

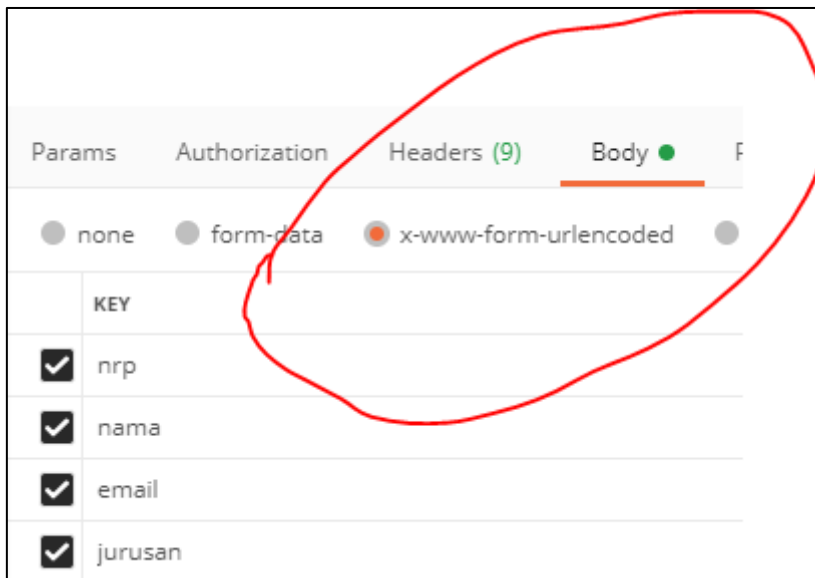
```

17
18     public function deleteMahasiswa($id)
19     {
20         $this->db->delete('mahasiswa', ['id' => $id]);
21         return $this->db->affected_rows();
22     }
23
24     public function createMahasiswa($data)
25     {
26         $this->db->insert('mahasiswa', $data);
27         return $this->db->affected_rows();
28     }
29
30     public function updateMahasiswa($data, $id)
31     {
32         $this->db->update('mahasiswa', $data, ['id' => $id]);
33         return $this->db->affected_rows();
34     }
35 }
36

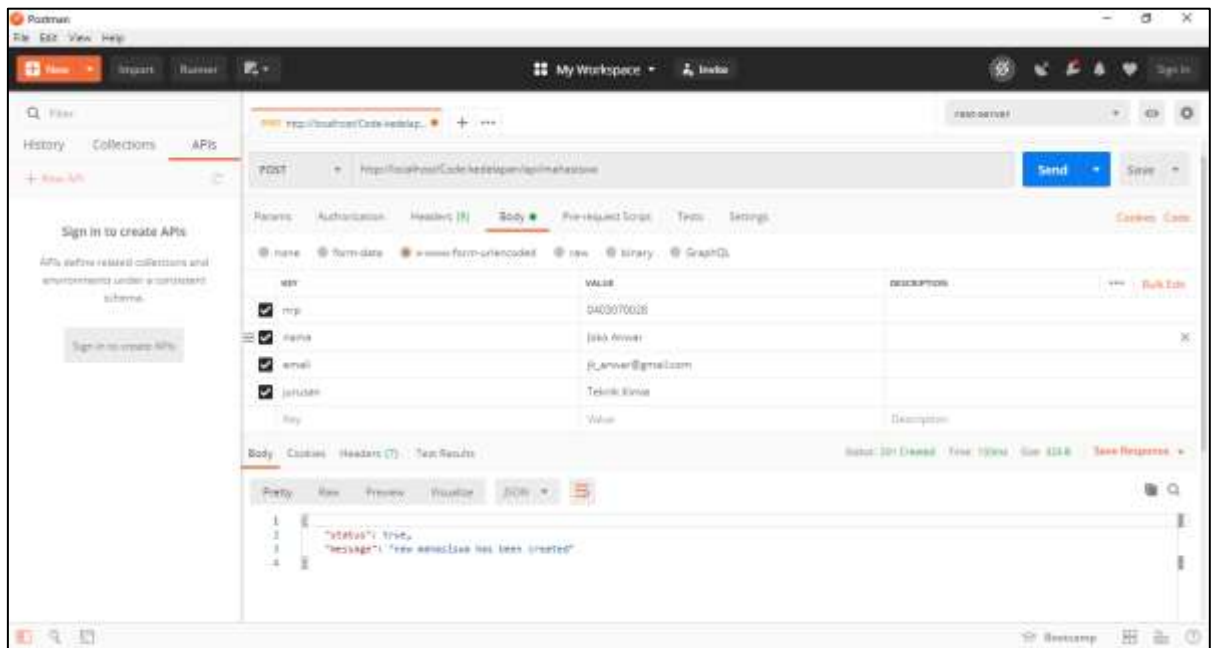
```

6

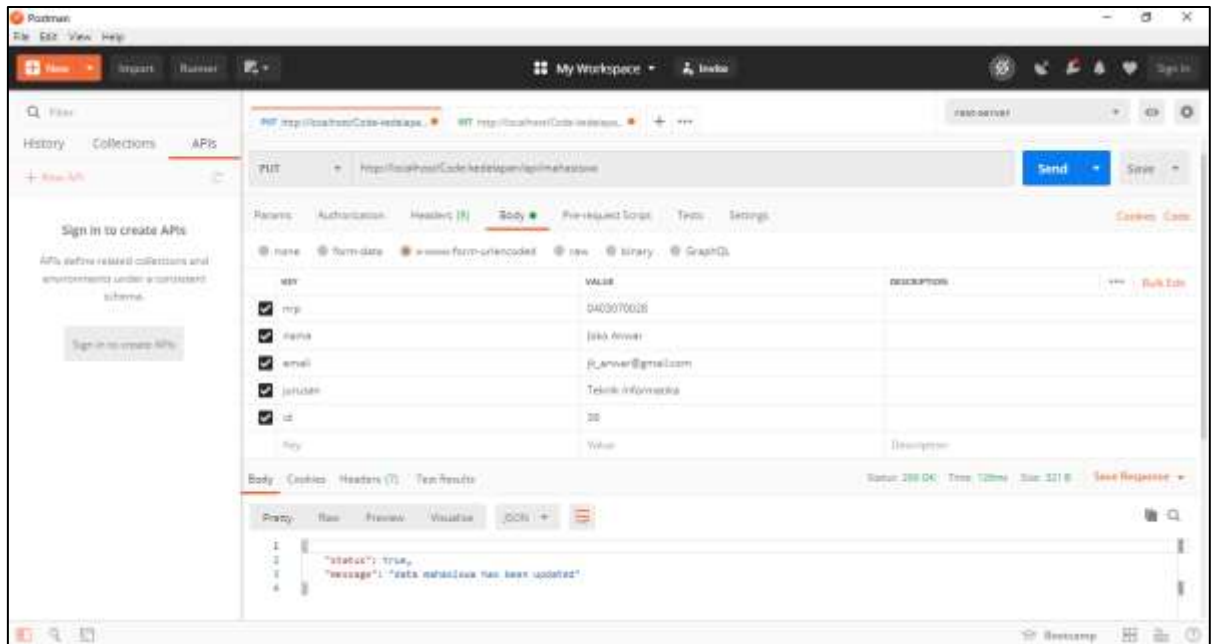
Selanjutnya lakukan operasi **Create** data pada aplikasi postman, untuk operasi **Create**, **Update** dan **Delete**, arahkan parameter ke **Body** dan **x-www-form-urlencoded** seperti berikut:



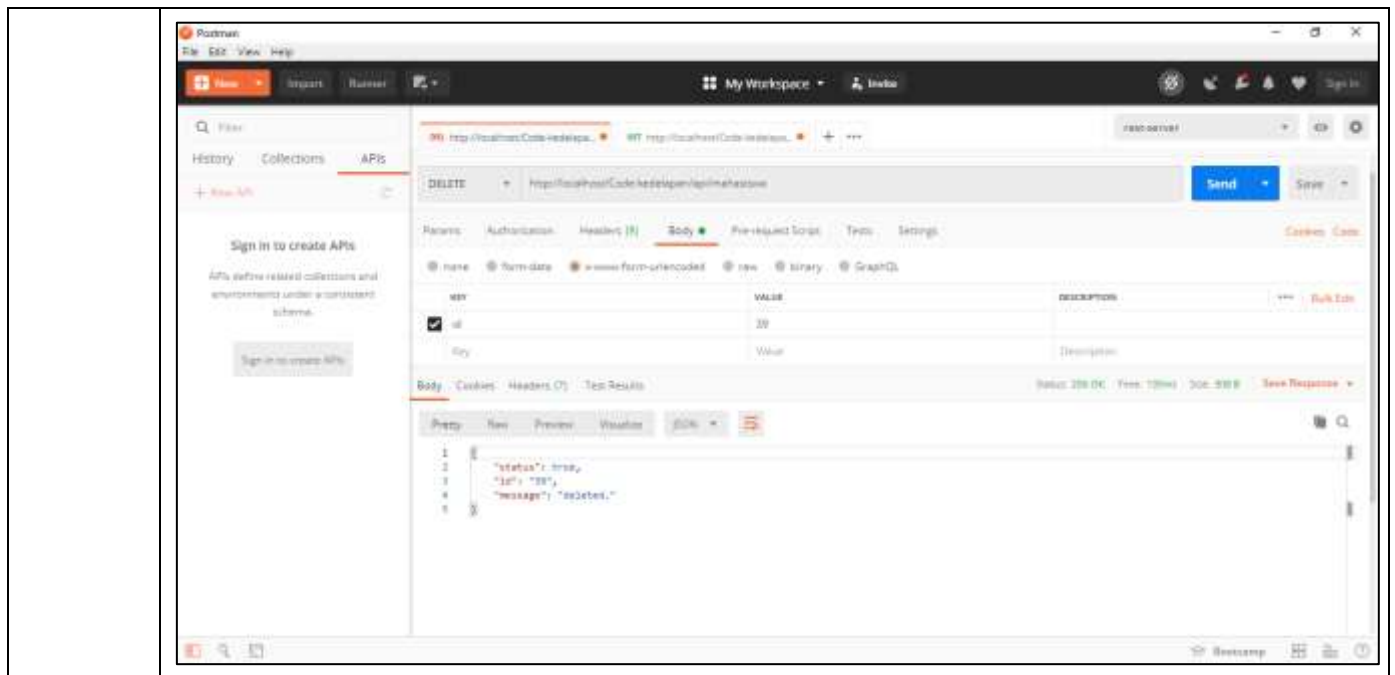
Params		Authorization	Headers (9)	Body ●	F
<input type="radio"/> none <input type="radio"/> form-data <input checked="" type="radio"/> x-www-form-urlencoded <input type="radio"/>					
	KEY				
<input checked="" type="checkbox"/>	nrp				
<input checked="" type="checkbox"/>	nama				
<input checked="" type="checkbox"/>	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 Mengerjakan --