



Topik

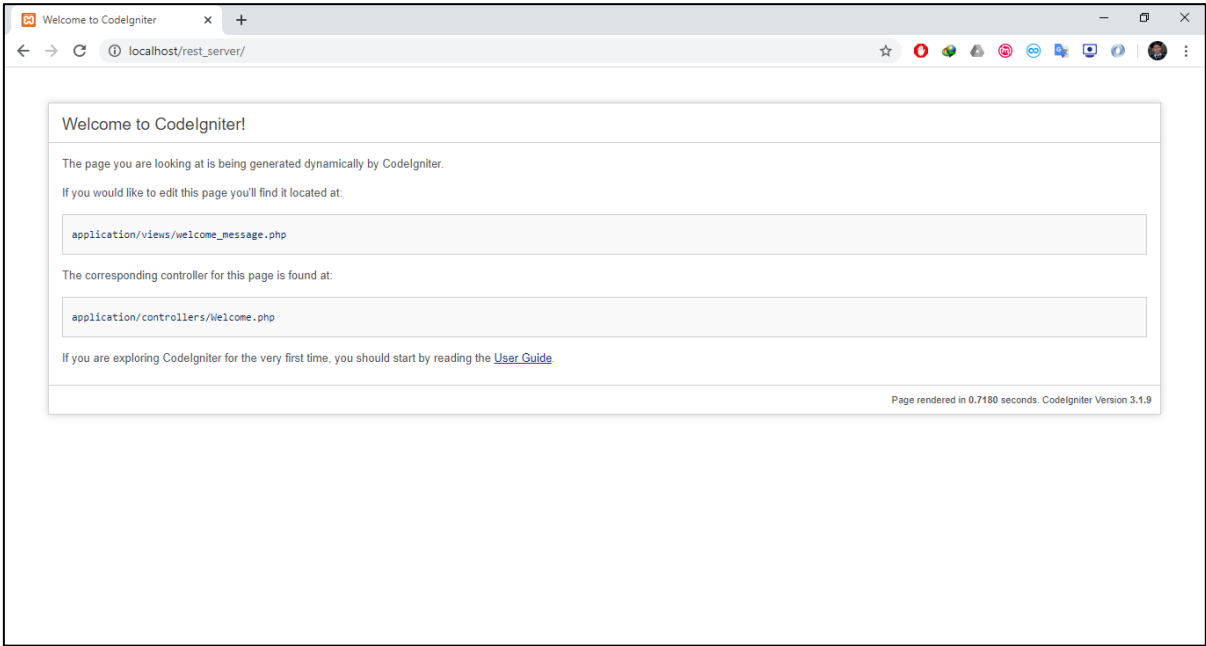
Implementasi Restful API

Tujuan

Mahasiswa diharapkan dapat:

1. Memahami proses membaca data menggunakan Library Rest Server di Codeigniter.
2. Memahami proses menambah data menggunakan Library Rest Server di Codeigniter.
3. Memahami proses mengubah data menggunakan Library Rest Server di Codeigniter.
4. Memahami proses menghapus data menggunakan Library Rest Server di Codeigniter.

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 : https://drive.google.com/drive/folders/10YaCrqrqG2M1Zk0r-trln0hROP9q0hpi?usp=sharing
3	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: 

4

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

```

24  |
25  */
26  $config['base_url'] = '';
27
28  /*
37  */
38  $config['index_page'] = 'index.php';
39
40  /*

```

Menjadi seperti berikut :

```

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' => 'mysql',
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');

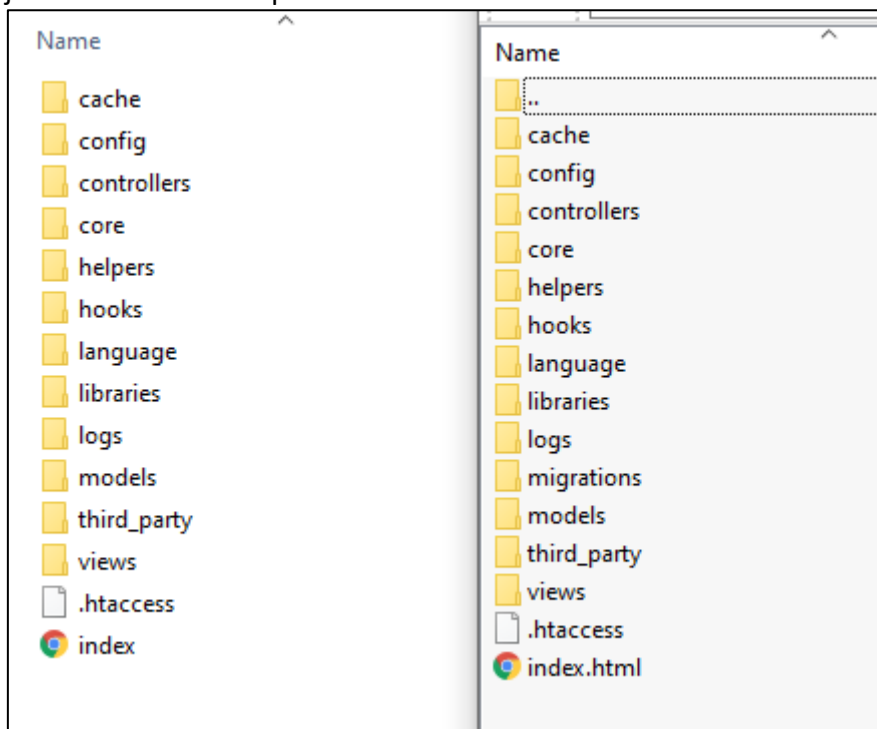
ALTER TABLE `mahasiswa`
  ADD PRIMARY KEY (`id`);

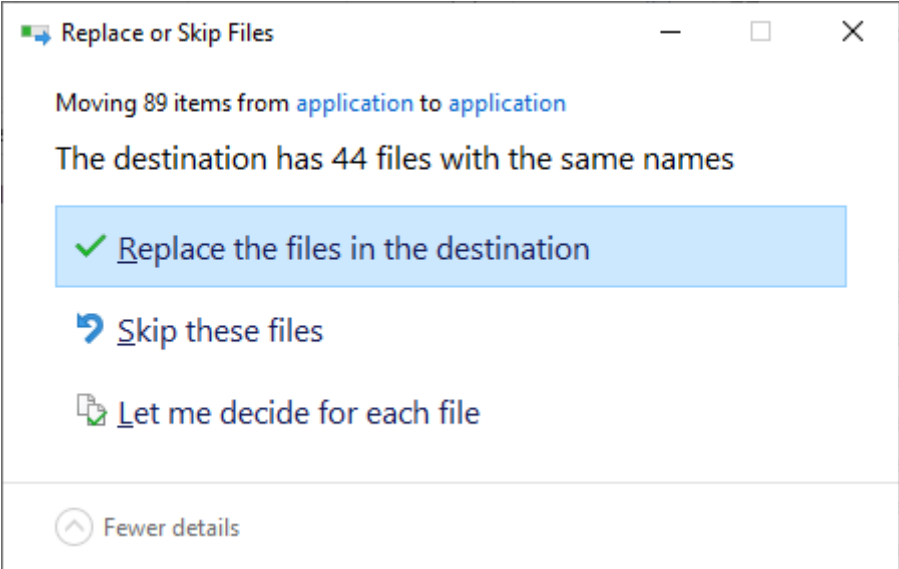
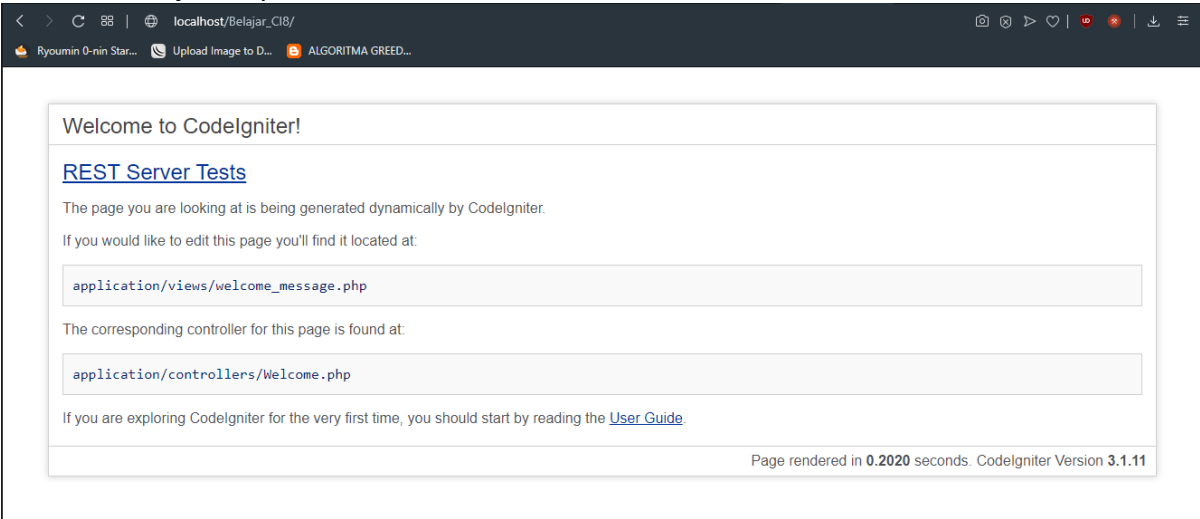
ALTER TABLE `mahasiswa`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=33;

```

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.



	 <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:

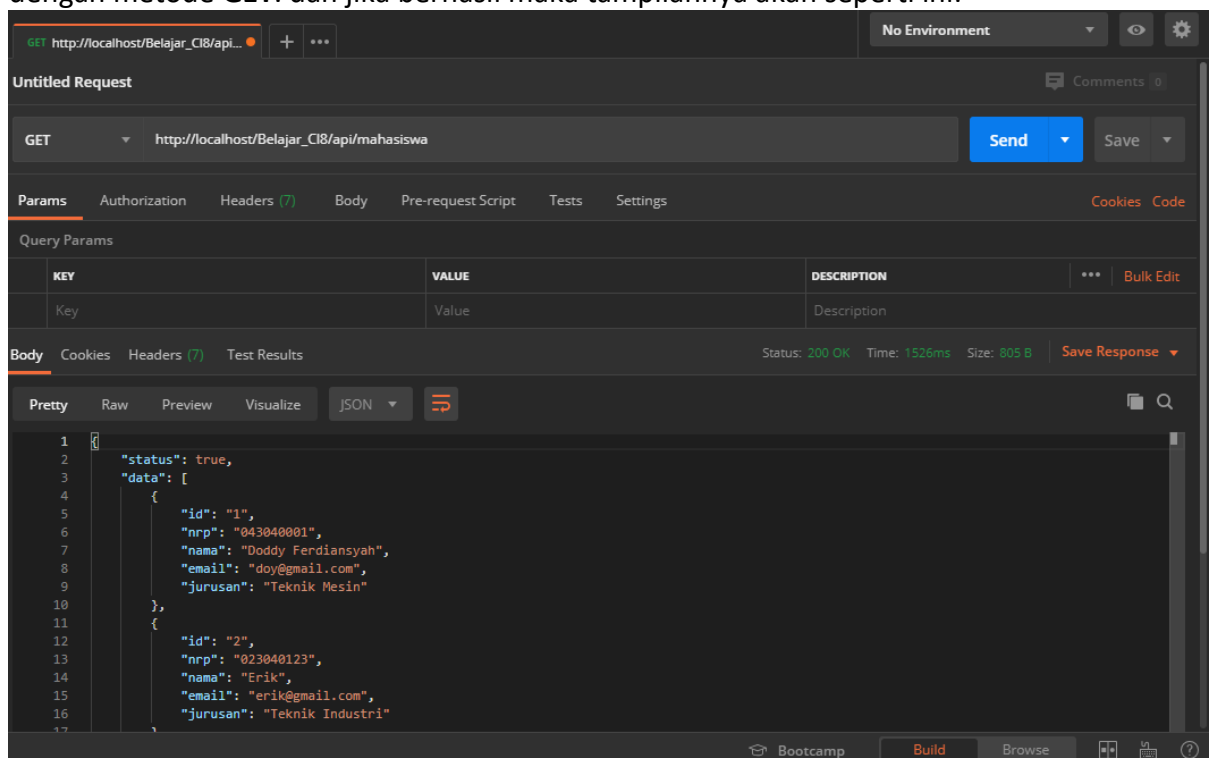
```
Mahasiswa.php X Mahasiswa_model.php
application > controllers > api > Mahasiswa.php > ...
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  require APPPATH . 'libraries/REST_Controller.php';
5  require APPPATH . 'libraries/Format.php';
6
7  class Mahasiswa extends REST_Controller {
8
9
10     public function __construct()
11     {
12         parent::__construct();
13         $this->load->model('Mahasiswa_model','mahasiswa');
14     }
15
16
17     public function index_get()
18     {
19         $id = $this->get('id');
20         if ($id === null) {
21             $mahasiswa = $this->mahasiswa->getMahasiswa();
22         } else {
23             $mahasiswa = $this->mahasiswa->getMahasiswa($id);
24         }
25
26         if ($mahasiswa) {
27             $this->response([
28                 'status' => true,
29                 'data' => $mahasiswa
30             ], REST_Controller::HTTP_OK);
31         } else {
32             $this->response([
33                 'status' => false,
34                 'data' => 'id not found'
35             ], REST_Controller::HTTP_NOT_FOUND);
36         }
37     }
38 }
39 /* End of file Mahasiswa.php */
40 ?>
```

2

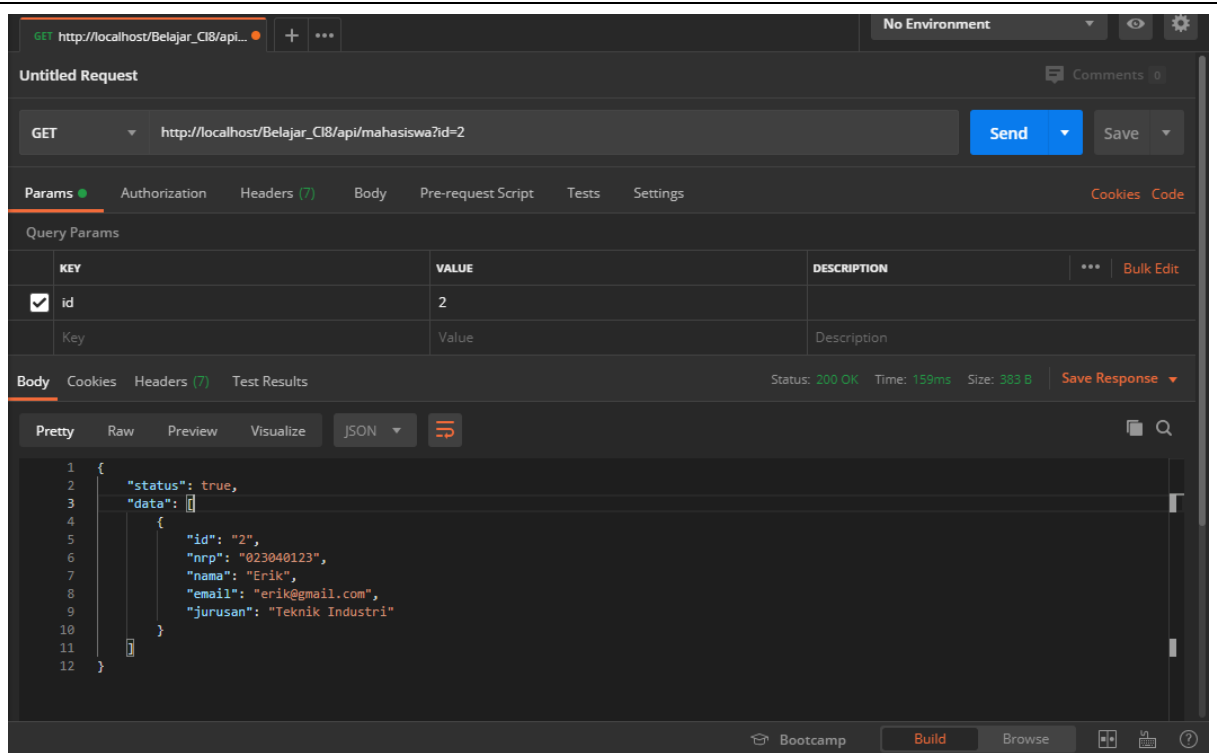
Selanjutnya Buatlah file **Mahasiswa_model.php** pada model. Selanjutnya tuliskan kode berikut:

```
Mahasiswa.php Mahasiswa_model.php X
application > models > Mahasiswa_model.php > PHP Intelephense > Mahasiswa_model > getMahasiswa
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class Mahasiswa_model extends CI_Model {
6      public function getMahasiswa($id = null)
7      {
8          if ($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
16 /* End of file Mahasiswa_model.php */
17
18
19 ?>
```

- 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**:

```

public function index_delete()
{
    $id = $this->delete('id');

    if ($id === null) {
        $this->response([
            'status' => false,
            'message' => 'provide an id!'
        ], REST_Controller::HTTP_BAD_REQUEST);
    } else {
        if ($this->mahasiswa->deleteMahasiswa($id)>0) {
            $this->response([
                'status' => true,
                'id' => $id,
                'message' => 'deleted.'
            ], REST_Controller::HTTP_OK);
        } else {
            $this->response([
                'status' => false,
                'message' => 'id not found'
            ], REST_Controller::HTTP_BAD_REQUEST);
        }
    }
}

```

```

public function index_post()
{
    $data = [
        'nrp' => $this->post('nrp'),
        'nama' => $this->post('nama'),
        'email' => $this->post('email'),
        'jurusan' => $this->post('jurusan'),
    ];

    if ($this->mahasiswa->createMahasiswa($data)>0) {
        $this->response([
            'status' => true,
            'message' => 'new mahasiswa has been created'
        ], REST_Controller::HTTP_CREATED);
    } else {
        $this->response([
            'status' => false,
            'message' => 'failed to create new data!'
        ], REST_Controller::HTTP_BAD_REQUEST);
    }
}

```

```

public function index_put()
{
    $id = $this->put('id');
    $data = [
        'nrp' => $this->put('nrp'),
        'nama' => $this->put('nama'),
        'email' => $this->put('email'),
        'jurusan' => $this->put('jurusan'),
    ];

    if ($this->mahasiswa->updateMahasiswa($data, $id)>0) {
        $this->response([
            'status' => true,
            'message' => 'data mahasiswa has been updated'
        ], REST_Controller::HTTP_OK);
    } else {
        $this->response([
            'status' => false,
            'message' => 'failed to update data!'
        ], REST_Controller::HTTP_BAD_REQUEST);
    }
}

```

5

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


```

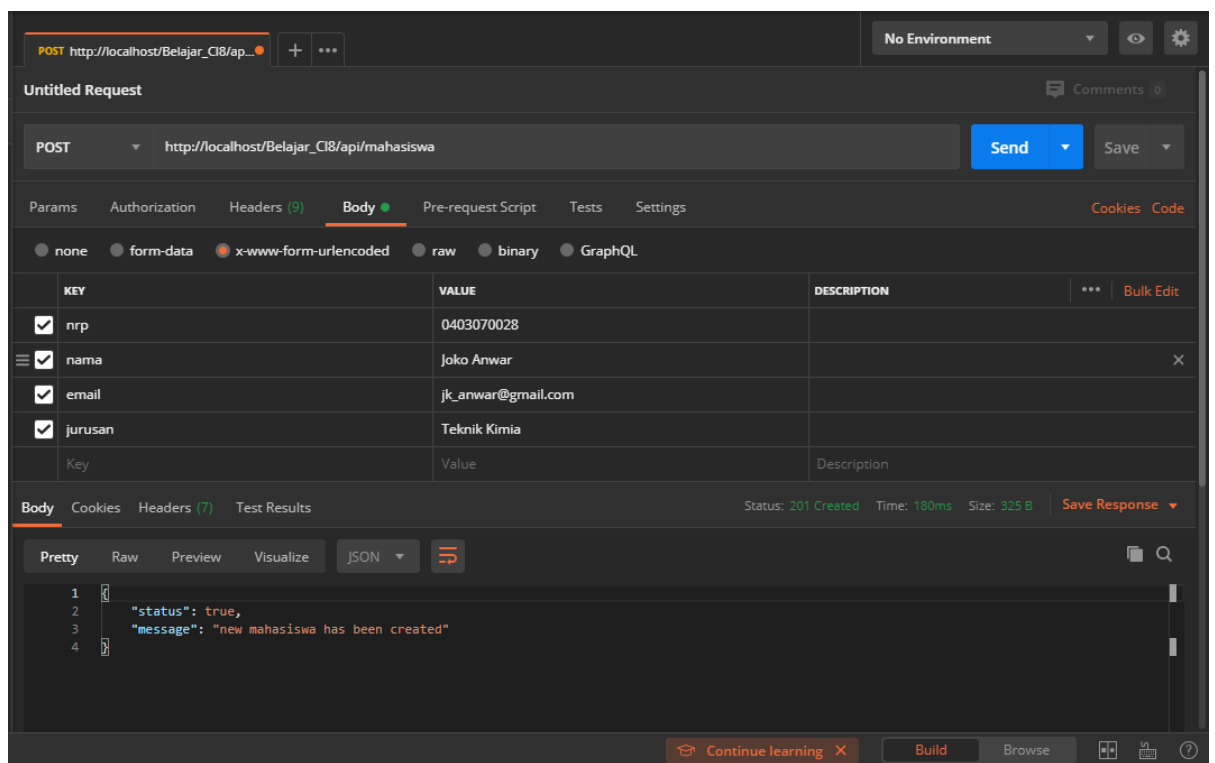
public function deleteMahasiswa($id)
{
    $this->db->delete('mahasiswa',['id' => $id]);
    return $this->db->affected_rows();
}

public function createMahasiswa($data)
{
    $this->db->insert('mahasiswa', $data);
    return $this->db->affected_rows();
}

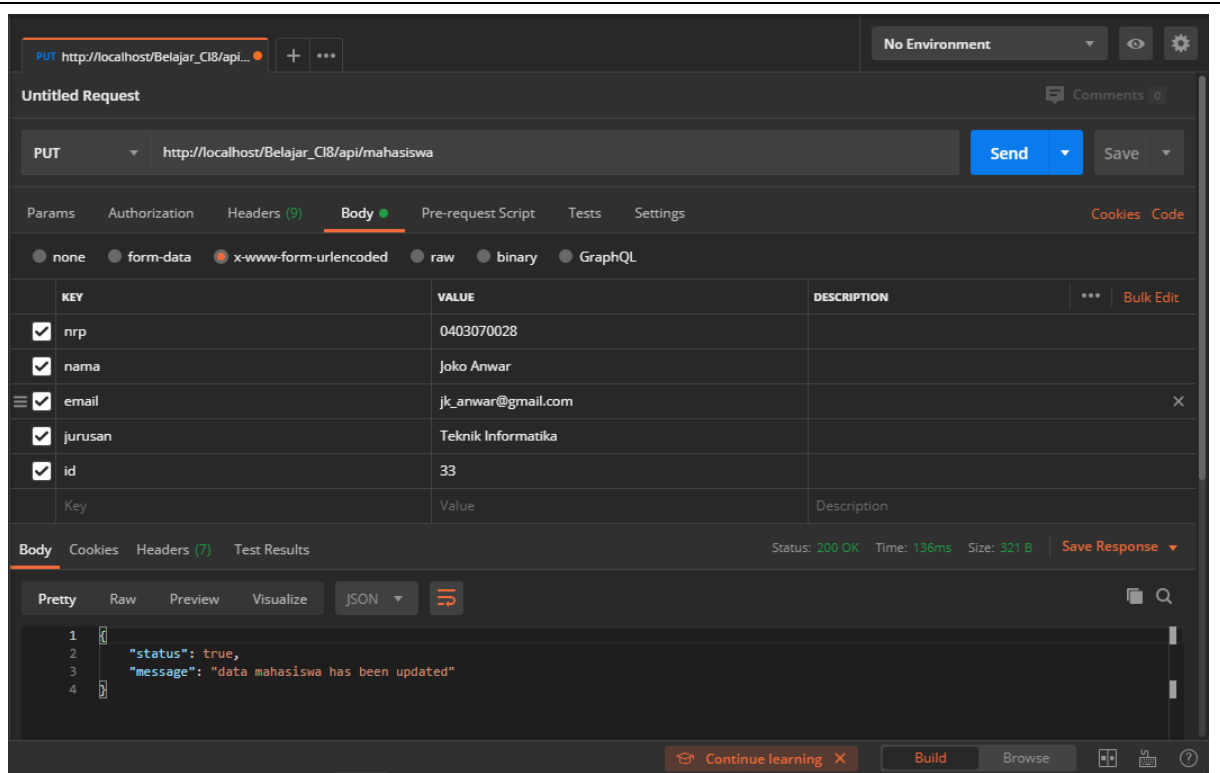
public function updateMahasiswa($data, $id)
{
    $this->db->update('mahasiswa', $data, ['id'=>$id]);
    return $this->db->affected_rows();
}

```

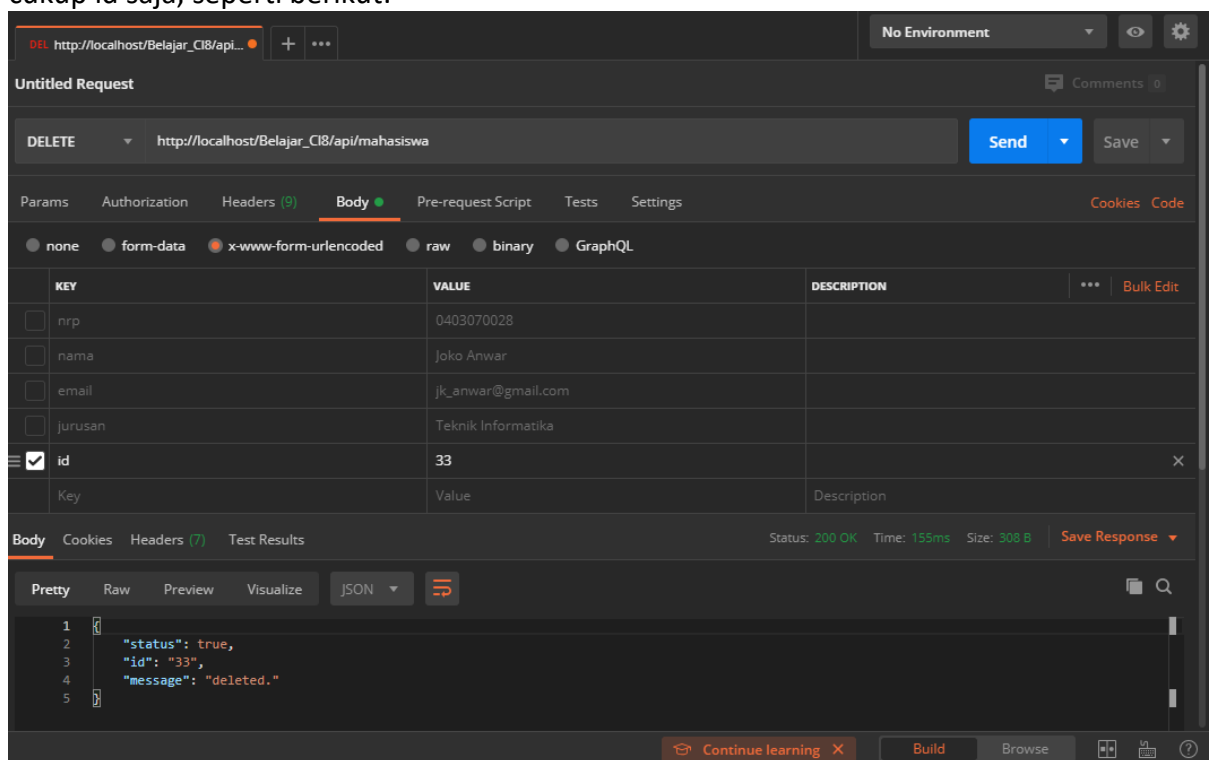
- 6 Selanjutnya lakukan operasi **Create** data pada aplikasi postman, untuk operasi **Create**, **Update** dan **Delete**, arahkan parameter ke **Body** dan **x-www-form-urlencoded** seperti berikut:



- 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:



-- Selamat Mengerjakan --