

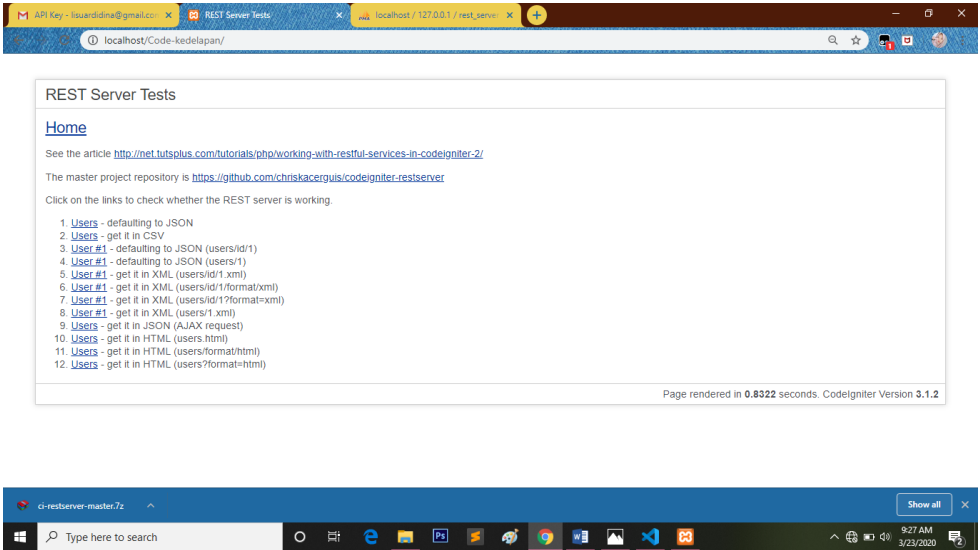


## LAPORAN PRAKTIKUM

### Web Lanjut

**NAMA** : Dina Lisuardi  
**NIM** : 1941723004  
**Kelas** : TI 2B  
**Isi Laporan** : Jobsheet-8 : Implementasi Restful Api

#### Bagian 1

| Langka<br>h | 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 <a href="http://localhost/Code-kedelapan/">http://localhost/Code-kedelapan/</a>. Pastikan tampilan seperti gambar dibawah ini:</p>  |
| 4           | Selanjutnya lakukan konfigurasi dasar pada <b>config.php</b> seperti berikut:   |

```

*/
$config['base_url'] = 'http://localhost/Code-kedelapan/';
/*
|-----

```

#### Autoload

```

$autoload['libraries'] = array('database');
*/
$autoload['helper'] = array('url');

```

#### Database

```

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'rest_server',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

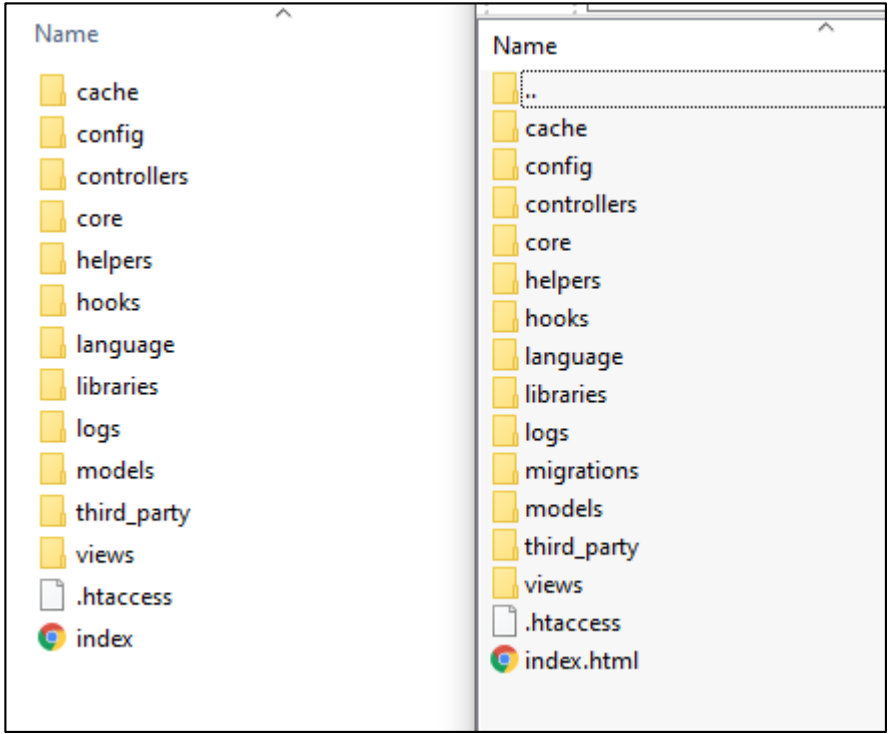
- 5 Selanjutnya buatlah database **rest\_server** pada phpmyadmin dan masukkan data dummy. Untuk data dummy dapat dibuat sebagai berikut:

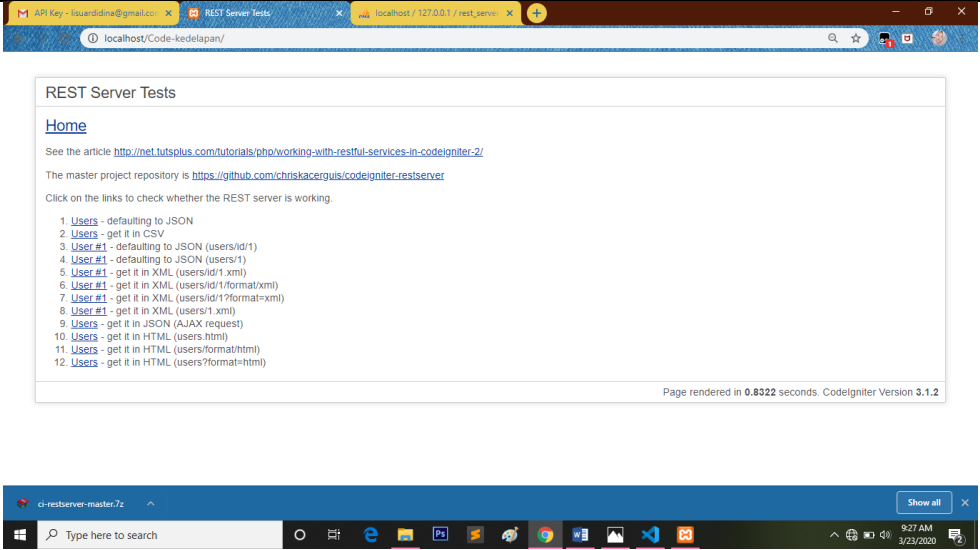
```

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

```

|   |   |
|---|---|
|   | <pre> (1, '043040001', 'Doddy Ferdiansyah', 'doy@gmail.com', 'Teknik Me sin'), (2, '023040123', 'Erik', 'erik@gmail.com', 'Teknik Industri'), (3, '043040321', 'Rommy Fauzi', 'rommy@gmail.com', 'Teknik Planol ogi'), (4, '033040023', 'Fajar Darmawan ', 'fajar@yahoo.com', 'Teknik In formatika'), (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 <b>application</b> dan duplikasi file dan folder tersebut kedalam folder <b>application</b> pada project <b>Code-kedelapan</b>. Klik <b>Replace</b> files jika ada notifikasi duplikat.</p>   |
| 7 | <p>Selanjutnya buka project <a href="http://localhost/Code-kedelapan/">http://localhost/Code-kedelapan/</a>. Jika berhasil, tampilan akan berubah menjadi seperti ini:</p>  |

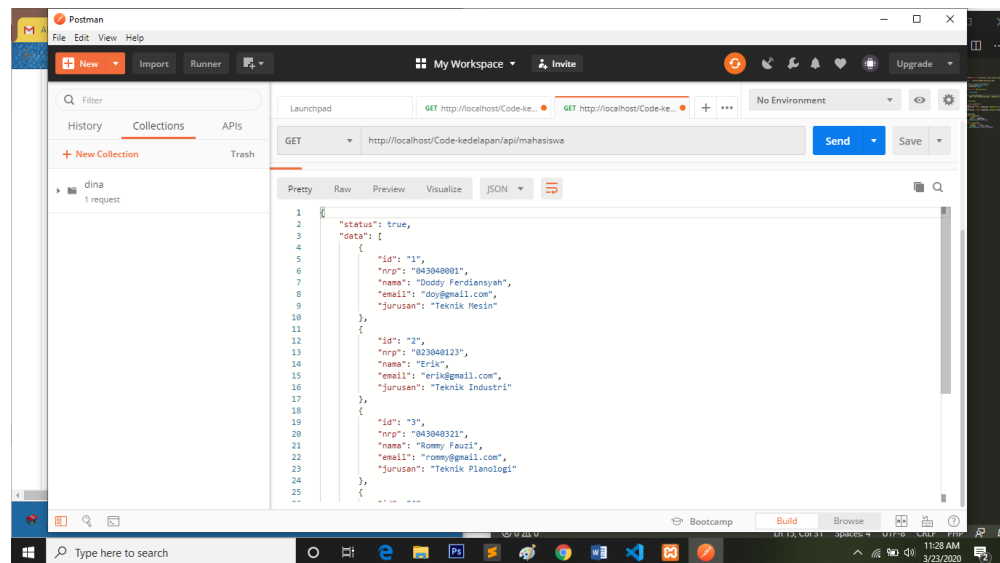
|  |  |
|--|--|
|  |  <p>REST Server Tests</p> <p><a href="#">Home</a></p> <p>See the article <a href="http://net.tutsolus.com/tutorials/php/working-with-restful-services-in-codeigniter-2/">http://net.tutsolus.com/tutorials/php/working-with-restful-services-in-codeigniter-2/</a></p> <p>The master project repository is <a href="https://github.com/chrisKacerguis/codeigniter-restserver">https://github.com/chrisKacerguis/codeigniter-restserver</a></p> <p>Click on the links to check whether the REST server is working.</p> <ol style="list-style-type: none"><li>1. <a href="#">Users</a> - defaulting to JSON</li><li>2. <a href="#">Users</a> - get it in CSV</li><li>3. <a href="#">User #1</a> - defaulting to JSON (users/id/1)</li><li>4. <a href="#">User #1</a> - defaulting to JSON (users/1)</li><li>5. <a href="#">User #1</a> - get it in XML (users/id/1.xml)</li><li>6. <a href="#">User #1</a> - get it in XML (users/id/1?format=xml)</li><li>7. <a href="#">User #1</a> - get it in XML (users/id/1?format=xml)</li><li>8. <a href="#">User #1</a> - get it in XML (users/1.xml)</li><li>9. <a href="#">Users</a> - get it in JSON (AJAX request)</li><li>10. <a href="#">Users</a> - get it in HTML (users.html)</li><li>11. <a href="#">Users</a> - get it in HTML (users?format=html)</li><li>12. <a href="#">Users</a> - get it in HTML (users?format=html)</li></ol> <p>Page rendered in 0.8322 seconds. CodeIgniter Version 3.1.2</p> <p>8 Tahap instalasi Library Rest Server selesai</p> |
|--|--|

## Bagian 2

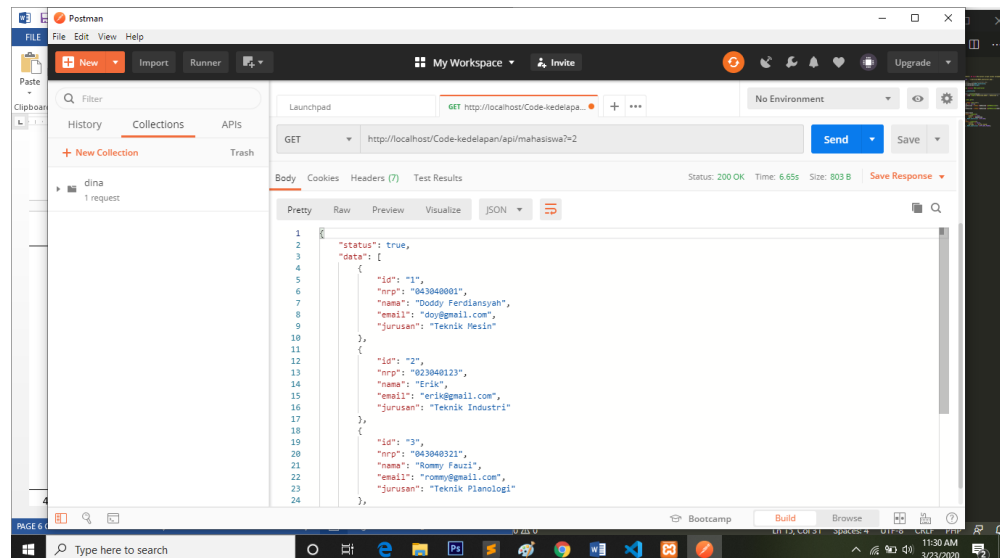
| Langkah | Keterangan  |
|---------|---|
| 1       | <p>Buatlah file <b>Mahasiswa.php</b> pada <b>controller/api</b>. Selanjutnya tuliskan kode berikut:</p> <pre> 1  &lt;?php 2 3  defined('BASEPATH') OR exit('No direct script access allowed'); 4 5  require APPPATH . '/libraries/REST_Controller.php'; 6 7  //require APPPATH . '/libraries/Format.php'; 8  use Restserver\Libraries\REST_Controller; 9  use Restserver\Libraries\Format; 10 11 class Mahasiswa extends REST_Controller 12 { 13     function __construct() 14     { 15         parent::__construct(); 16         \$this-&gt;load-&gt;model('Mahasiswa_model','mahasiswa'); 17     } 18 19     function index_get() 20     { 21         \$id = \$this-&gt;get('id'); 22         if (\$id === null){ 23             \$mahasiswa = \$this-&gt;mahasiswa-&gt;getMahasiswa(); 24         }else{ 25             \$mahasiswa = \$this-&gt;mahasiswa-&gt;getMahasiswa(\$id); 26         } 27 28         if(\$mahasiswa){ 29             \$this-&gt;response([ 30                 'status' =&gt; true, 31                 'data' =&gt; \$mahasiswa 32             ], REST_Controller::HTTP_OK); 33         }else{ 34             \$this-&gt;response([ 35                 'status' =&gt; false, 36                 'message' =&gt; 'id not found' 37             ], REST_Controller::HTTP_NOT_FOUND); 38         } 39     } 40 </pre> |
| 2       | <p>Selanjutnya Buatlah file <b>Mahasiswa_model.php</b> pada model. Selanjutnya tuliskan kode berikut:</p> <pre> 1  &lt;?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-&gt;db-&gt;get('mahasiswa')-&gt;result_array(); 10             } else { 11                 return \$this-&gt;db-&gt;get_where('mahasiswa', ['id' =&gt; \$id]) 12                     -&gt;result_array(); 13             } 14         } 15     } 16 </pre>  |

3

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



Berdasarkan id



4

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

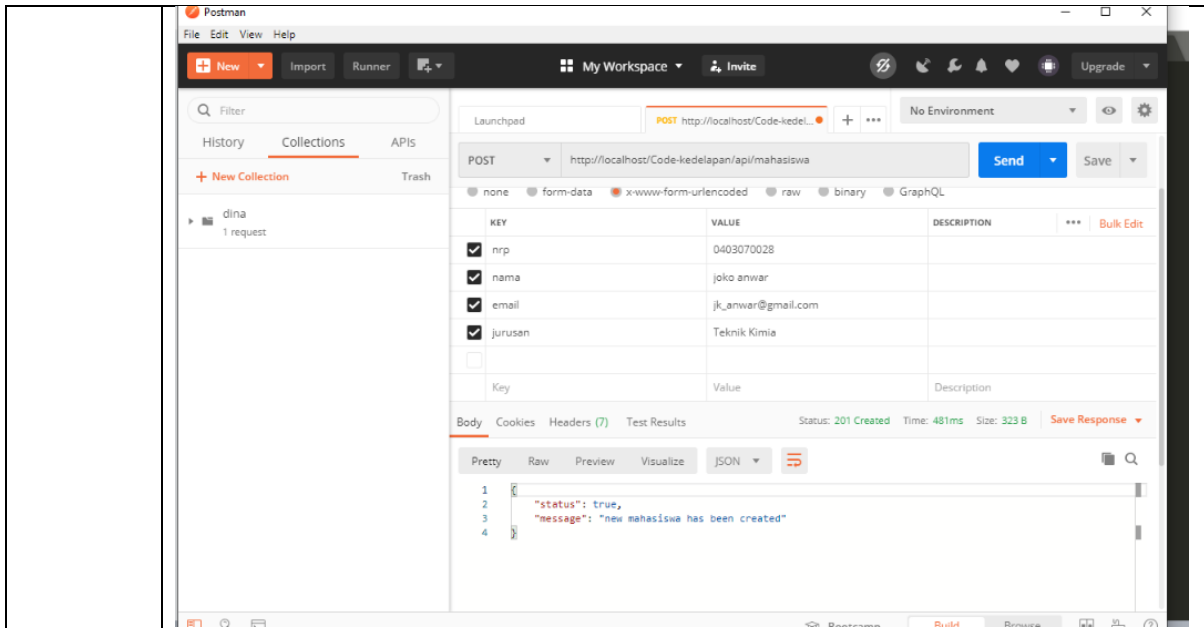
```
// DELETE
function index_delete() {
    $id = $this->delete('id');
    if ($id === null) {
        $this->response([
            'status' => false,
            'message' => 'provide an id'
        ], REST_Controller::HTTP_BAD_REQUEST);
    } else {
        if ($this->mahasiswa->deleteMahasiswa($id) > 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);
        }
    }
}
}
```

```
// POST
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);
    }
}
}
```

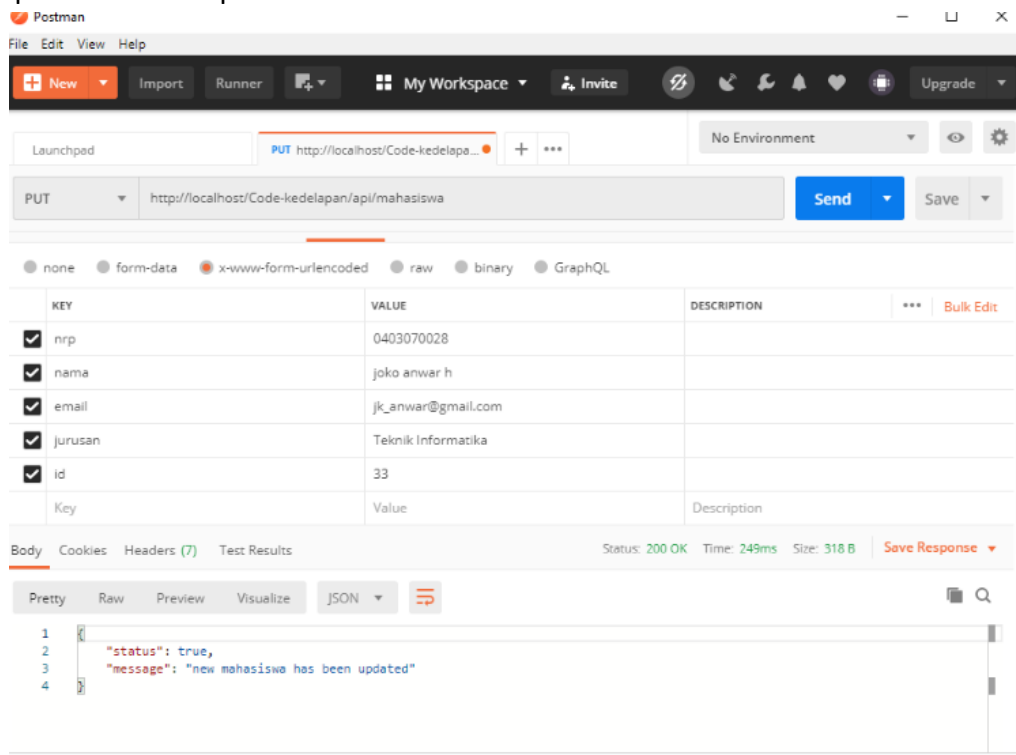
|   |  |
|---|--|
|   | <pre> // PUT public function index_put(){     \$id = \$this-&gt;put('id');     \$data = [         'nrp' =&gt; \$this-&gt;put('nrp'),         'nama' =&gt; \$this-&gt;put('nama'),         'email' =&gt; \$this-&gt;put('email'),         'jurusan' =&gt; \$this-&gt;put('jurusan'),     ];     if (\$this-&gt;mahasiswa-&gt;updateMahasiswa(\$data, \$id) &gt; 0) {         \$this-&gt;response([             'status' =&gt; true,             'message' =&gt; 'new mahasiswa has been updated'         ], REST_Controller::HTTP_OK);     } else {         \$this-&gt;response([             'status' =&gt; false,             'message' =&gt; 'failed to update data'         ], REST_Controller::HTTP_BAD_REQUEST);     } } } ?&gt; </pre> |
| 5 | <p>Dan lanjutkan kode pada file <b>Mahasiswa_model.php</b></p> <pre> public function deleteMahasiswa(\$id){     \$this-&gt;db-&gt;delete('mahasiswa', ['id' =&gt; \$id]);     return \$this-&gt;db-&gt;affected_rows(); }  public function createMahasiswa(\$data){     \$this-&gt;db-&gt;insert('mahasiswa', \$data);     return \$this-&gt;db-&gt;affected_rows(); }  public function updateMahasiswa(\$data, \$id){     \$this-&gt;db-&gt;update('mahasiswa', \$data, ['id' =&gt; \$id]);     return \$this-&gt;db-&gt;affected_rows(); } } </pre>  |
| 6 | <p>Selanjutnya lakukan operasi <b>Create</b> data pada aplikasi postman, untuk operasi <b>Create</b>, <b>Update</b> dan <b>Delete</b>, arahkan parameter ke <b>Body</b> dan <b>x-www-form-urlencoded</b> seperti berikut:</p>  |





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:

