

# **LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT**



**Oleh:**

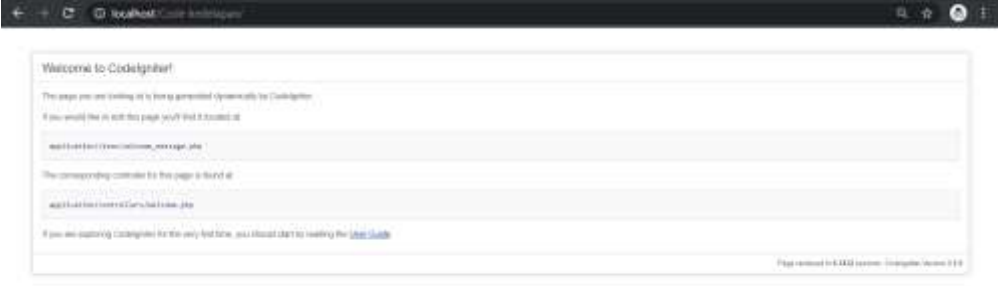
**ADIE BAGUS SATRIO**

**TI-2A**

**NIM : 1841720186**

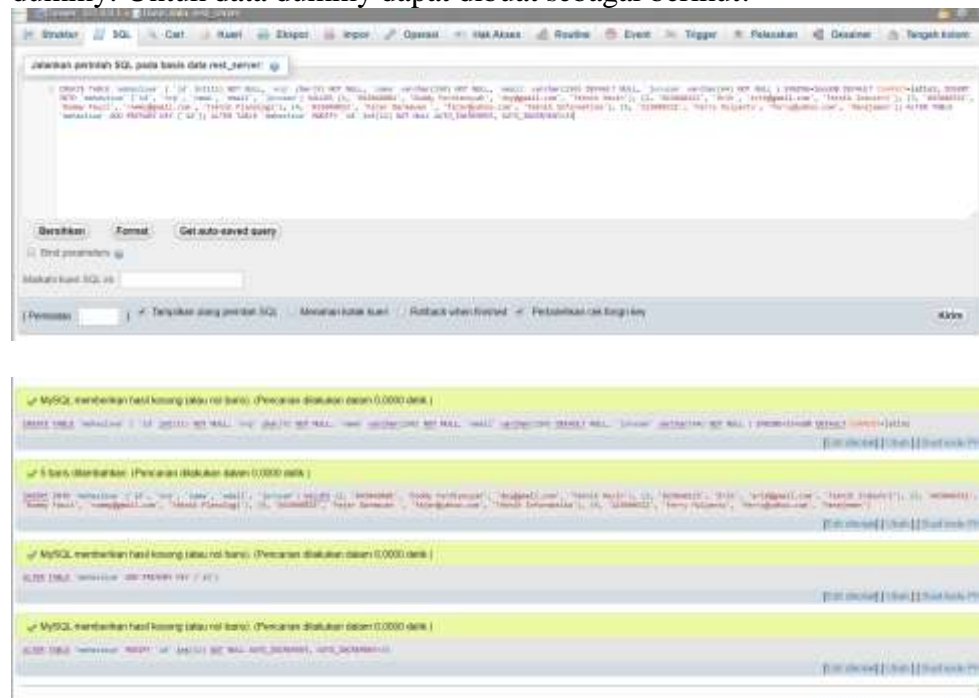
**PROGRAM STUDI TEKNIK  
INFORMATIKA JURUSAN TEKNOLOGI  
INFORMASI POLITEKNIK NEGERI MALANG**

# Jobsheet-8: Implementasi Restful API

Langkah	Keterangan
	<b>Praktikum – Bagian 1: Instalasi library restful codeigniter</b>
1.	<p>Extract codeigniter yang sudah di unduh dan letakkan pada folder htdocs. 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> 
2.	<p>Selanjutnya lakukan konfigurasi dasar pada config.php seperti berikut:</p> <pre data-bbox="359 981 1279 1294"> 25 */ 26 \$config['base_url'] = 'http://localhost/rest_server/'; 27 28 /* 29 ..... 30 Index File 31 ..... 32 Typically this will be your index.php file, unless you've renamed it to 33 something else. If you are using mod_rewrite to remove the page set this 34 variable so that it is blank. 35 36 */ 37 \$config['index_page'] = ''; 38 39 </pre> <p>Dan file autoload.php</p> <pre data-bbox="359 1370 1168 1527"> 60 */ 61 \$autoload['libraries'] = array('database'); 62 63 /* 64 ..... 65 Auto-load Drivers </pre> <pre data-bbox="359 1563 1168 1729"> 91 */ 92 \$autoload['helper'] = array('url'); 93 94 /* 95 ..... 96 Auto-load Config files 97 ..... </pre> <p>Dan database.php seperti berikut:</p> <pre data-bbox="359 1803 833 2011"> 76 \$db['default'] = array( 77     'dsn' =&gt; '', 78     'hostname' =&gt; 'localhost', 79     'username' =&gt; 'root', 80     'password' =&gt; '', 81     'database' =&gt; 'rest_server', 82     'dbdriver' =&gt; 'mysqli', 83     'dbprefix' =&gt; '' </pre>

- 5.

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



- 
- 6.

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



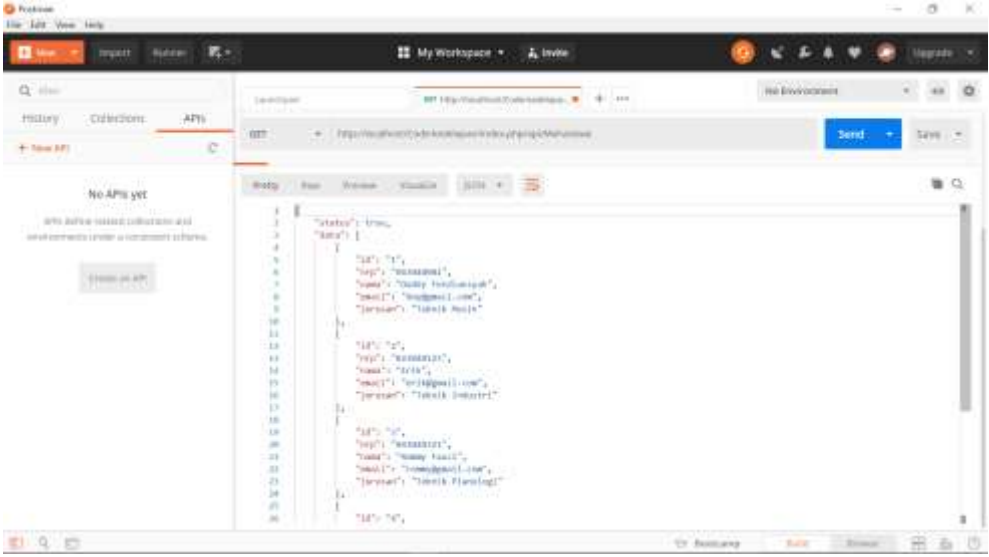
- 7.

Tahap instalasi Library Rest Server selesai.

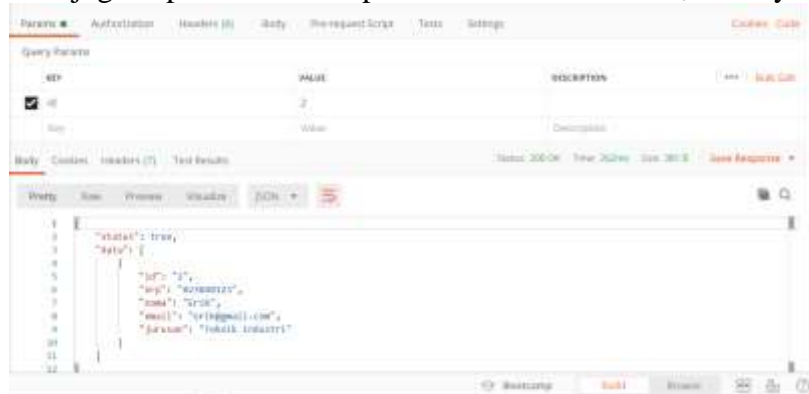
## Praktikum – Bagian 2: Membuat Aplikasi CRUD Restful Codeigniter

- 
- 1.

Buatlah file Mahasiswa.php pada controller/api. Selanjutnya tuliskan kode berikut:

	<pre> 1 &lt;?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 { 10     public function __construct() 11     { 12         parent::__construct(); 13         \$this-&gt;load_model('Mahasiswa_model', 'mahasiswa'); 14     } 15 16     public function index_get() 17     { 18         \$id = \$this-&gt;get('id'); 19         if (\$id === null) { 20             \$mahasiswa = \$this-&gt;mahasiswa-&gt;getMahasiswa(); 21         } else { 22             \$mahasiswa = \$this-&gt;mahasiswa-&gt;getMahasiswa(\$id); 23         } 24 25         if(\$mahasiswa) { 26             \$this-&gt;response([ 27                 'status' =&gt; true, 28                 'data' =&gt; \$mahasiswa 29             ], REST_Controller::HTTP_OK); 30         } else { 31             \$this-&gt;response([ 32                 'status' =&gt; false, </pre>
2.	<p>Selanjutnya Buatlah file Mahasiswa_model.php pada model. Selanjutnya tuliskan kode berikut:</p> <pre> 1 &lt;?php 2 3 class Mahasiswa_model extends CI_model 4 { 5     public function getMahasiswa(\$id = null) 6     { 7         if (\$id === null) { 8 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])-&gt;result_array(); 12         } 13     } 14 } 15 </pre>
3.	<p>Buka aplikasi postman, masukkan alamat <b><i>http://localhost/Code-kedelapan/api/mahasiswa</i></b> dengan metode GET. dan jika berhasil maka tampilannya akan seperti ini:</p>  <p>The screenshot shows the Postman interface with the following details:</p> <ul style="list-style-type: none"> <li><b>URL:</b> http://localhost/Code-kedelapan/api/mahasiswa</li> <li><b>Method:</b> GET</li> <li><b>Response:</b> <pre> {   "status": true,   "data": [     {       "id": "1",       "nama": "Mahasiswa",       "email": "Mahasiswa@gmail.com",       "password": "Mahasiswa"     },     {       "id": "2",       "nama": "Mahasiswa",       "email": "Mahasiswa@gmail.com",       "password": "Mahasiswa"     },     {       "id": "3",       "nama": "Mahasiswa",       "email": "Mahasiswa@gmail.com",       "password": "Mahasiswa"     }   ] } </pre> </li> </ul>

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) {
            //ok
            $this->response([
                'status' => true,
                'id' => $id,
                'message' => 'deleted.'
            ], REST_Controller::HTTP_OK);
        }else {
            // id not found
            $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'),
    ];
}
```

	<pre>         if(\$this-&gt;mahasiswa-&gt;createMahasiswa(\$data) &gt; 0) {             \$this-&gt;response([                 'status' =&gt; true,                 'message' =&gt; 'New mahasiswa has been created'             ], REST_Controller::HTTP_CREATED);         }else {             // id not found             \$this-&gt;response([                 'status' =&gt; false,                 'message' =&gt; 'Failed to create new data!'             ], REST_Controller::HTTP_BAD_REQUEST);         }     }     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; 'Data mahasiswa has been updated'             ], REST_Controller::HTTP_CREATED);         }else {             // id not found             \$this-&gt;response([                 'status' =&gt; false,                 'message' =&gt; 'Failed to update data!'             ], REST_Controller::HTTP_BAD_REQUEST);         }     } } </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); } </pre>

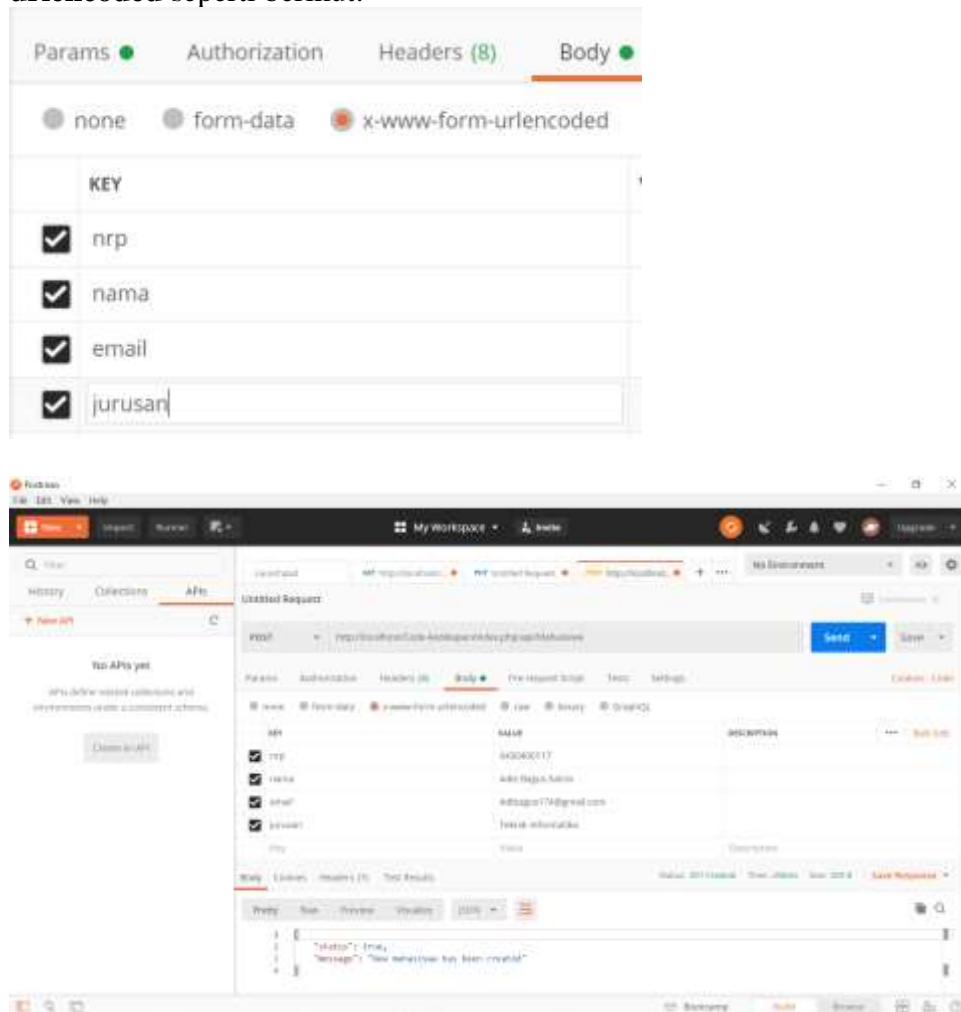
```

        return $this->db->affected_rows();
    }

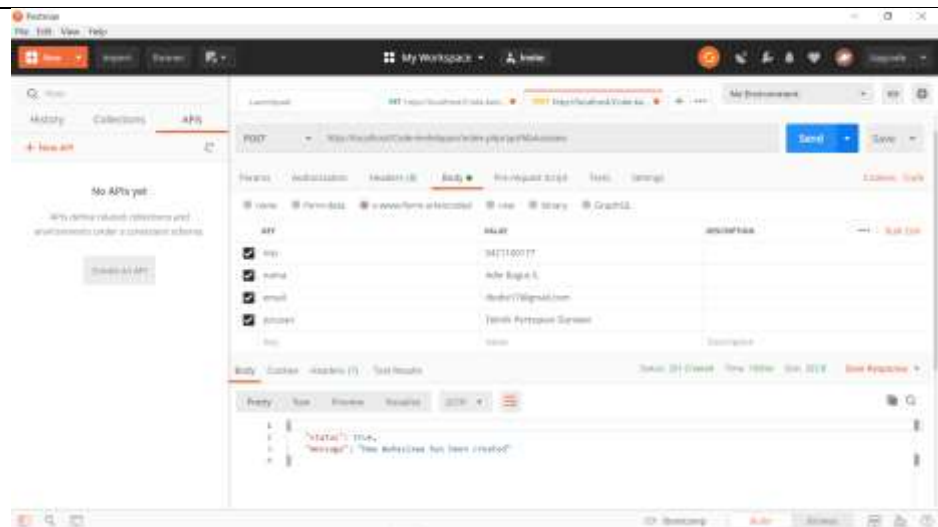
    public function updateMahasiswa($data, $id)
    {
        $this->db->update('mahasiswa', $data, ['id' => $id]);
        return $this->db->affected_rows();
    }

```

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:

