LAPORAN PRAKTIKUM

PEMROGRAMAN WEB LANJUT

Jobsheet 8 (Implementasi Resful API)

-

Oleh :

Muhammad Royyan Zamzami

NIM. 1841720053

PROGRAM STUDI TEKNIK INFORMATIKA

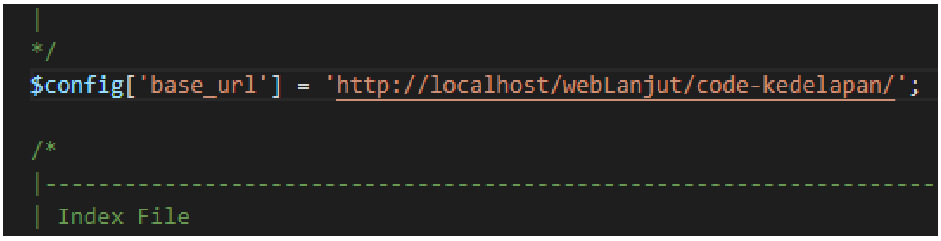
JURUSSAN TEKNOLOGI INFORMASI

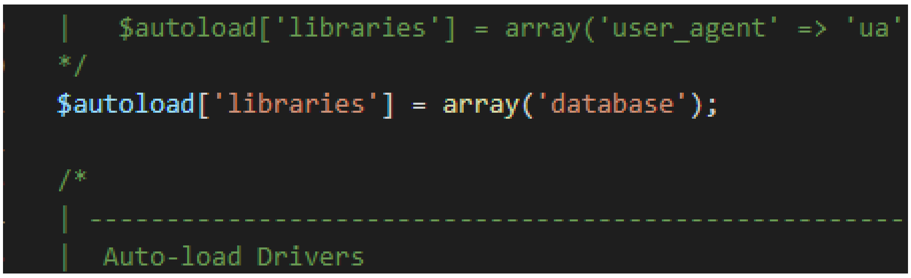
POLITEKNIK NEGERI MALANG

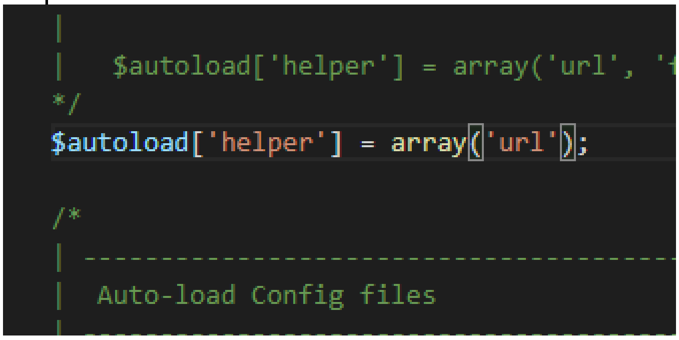
FEBRUARI 2020

Bagian 1 : Instalasi Library Restful codeigniter

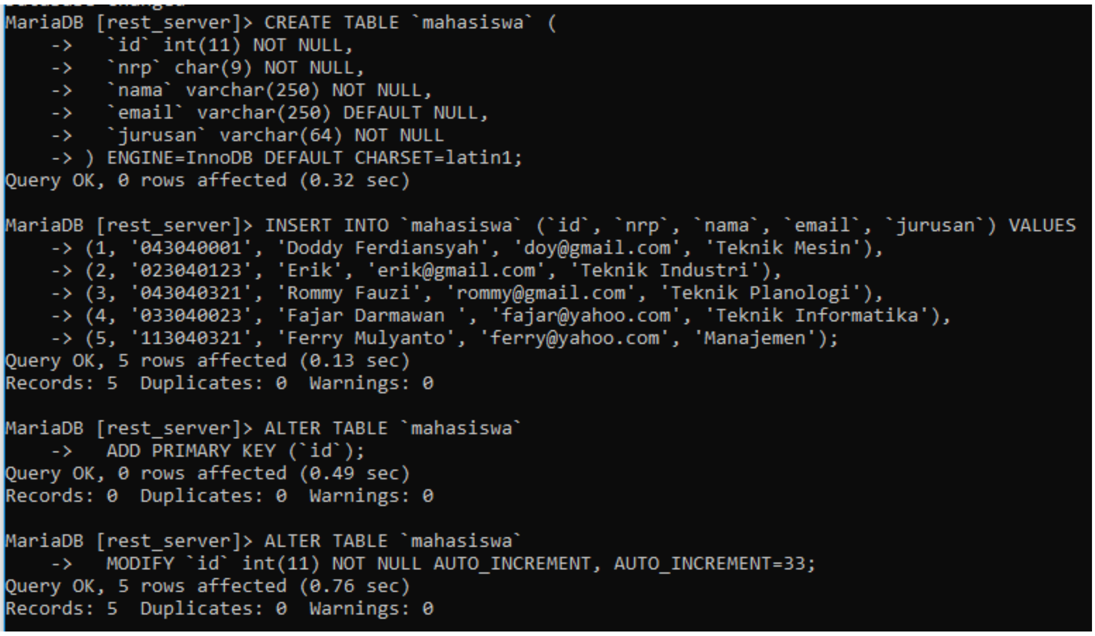
<table>
<tr><th>No</th><th>Keterangan</th></tr>
<tr><td>1</td><td>Unduh postman

Get Postman for Windows
Join 10 million developers and download the **ONLY** complete API Development Environment.
Download
Version 7.20.1 | RELEASE NOTES | PRODUCT ROADMAP</td></tr>
<tr><td>2</td><td>Setting config
- Base URL

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

/*
|--
| Index File</td></tr>
<tr><td>3</td><td>Setting Auto load
- Libraries

| $autoload['libraries'] = array('user_agent' => 'ua'
*/
$autoload['libraries'] = array('database');

/*
| ---
| Auto-load Drivers
- helper

|
| $autoload['helper'] = array('url', '
*/
$autoload['helper'] = array('url');

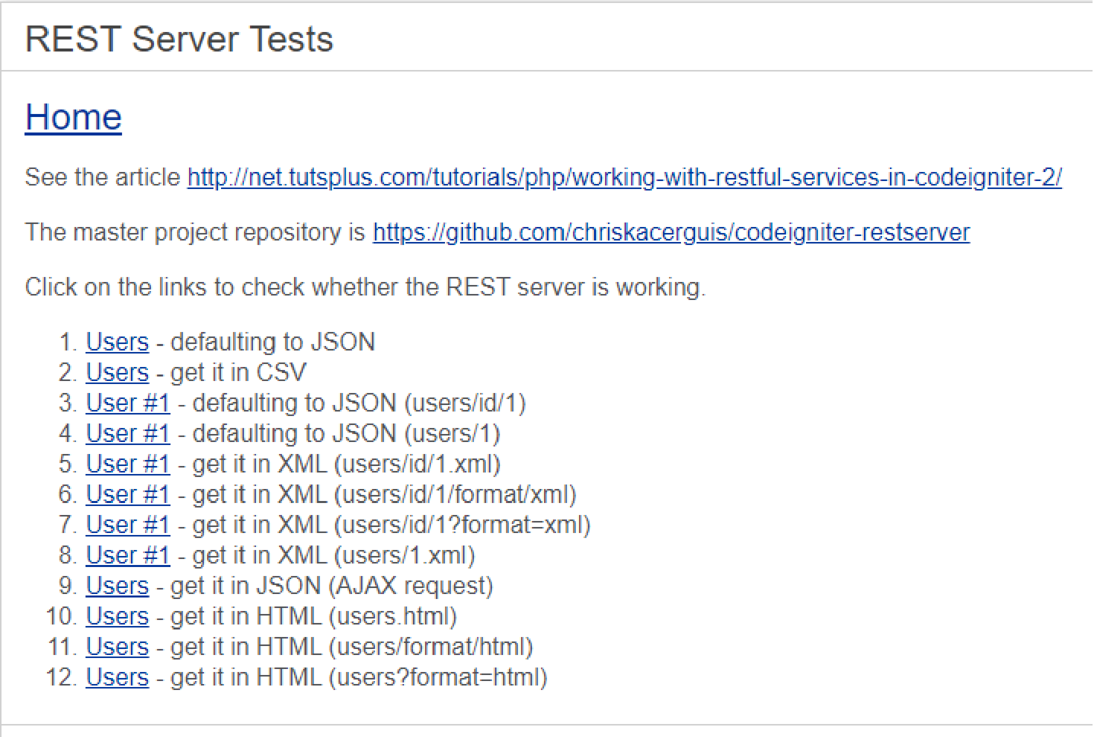
/*
| --
| Auto-load Config files</td></tr>
<tr><td>4</td><td>Buat database “rest_server” dan buat table</td></tr>
</table>



```
MariaDB [rest_server]> 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;
Query OK, 0 rows affected (0.32 sec)

MariaDB [rest_server]> 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');
Query OK, 5 rows affected (0.13 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [rest_server]> ALTER TABLE `mahasiswa`
    ->   ADD PRIMARY KEY (`id`);
Query OK, 0 rows affected (0.49 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [rest_server]> ALTER TABLE `mahasiswa`
    ->   MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=33;
Query OK, 5 rows affected (0.76 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

5 Replace ke folder aplication

6 Menampilkan “ localhost/webLanjut/code-kedelapan/rest_server “



REST Server Tests

Home

See the article http://net.tutsplus.com/tutorials/php/working-with-restful-services-in-codeigniter-2/

The master project repository is https://github.com/chriskacerguis/codeigniter-restserver

Click on the links to check whether the REST server is working.

1. Users - defaulting to JSON
2. Users - get it in CSV
3. User #1 - defaulting to JSON (users/id/1)
4. User #1 - defaulting to JSON (users/1)
5. User #1 - get it in XML (users/id/1.xml)
6. User #1 - get it in XML (users/id/1/format/xml)
7. User #1 - get it in XML (users/id/1?format=xml)
8. User #1 - get it in XML (users/1.xml)
9. Users - get it in JSON (AJAX request)
10. Users - get it in HTML (users.html)
11. Users - get it in HTML (users/format/html)
12. Users - get it in HTML (users?format=html)

Bagian 2 : Membuat Aplikasi CRUD Restful Codeigniter

<table>
<tr><th>No</th><th>Keterangan</th></tr>
<tr><td>1</td><td>

Membuat folder API di dalam controler dan membuat file Mahasiswa.php

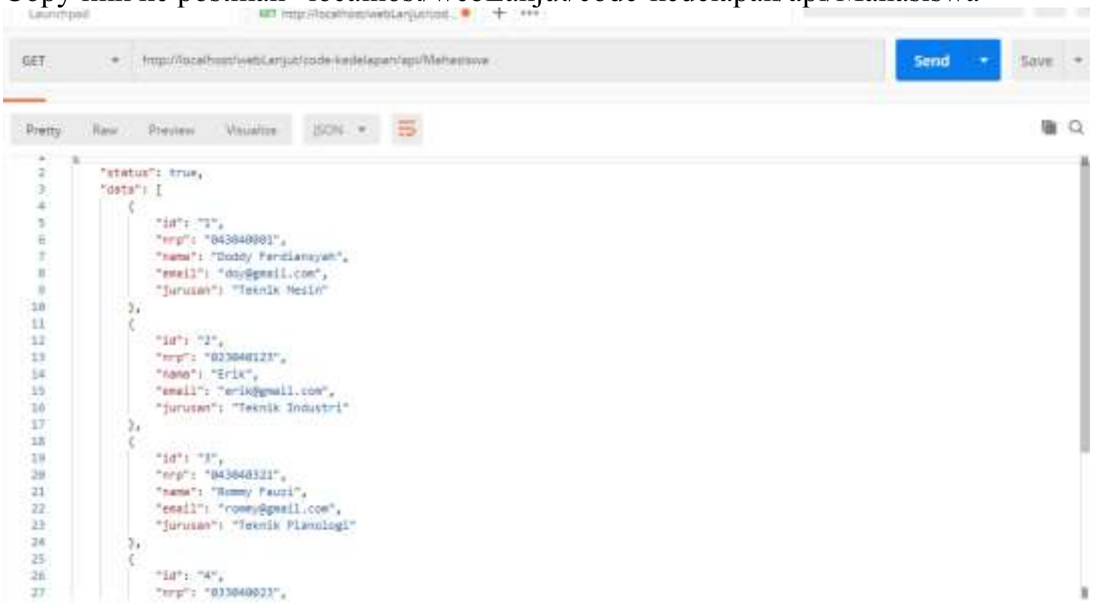
```php
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . 'libraries/REST_Controller.php';
require APPPATH . 'libraries/Format.php';

class Mahasiswa extends REST_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('Mahasiswa_model','mahasiswa');
    }

    public function index_get()
    {
        $id = $this->get('id');
        if ($id == null){
            $mahasiswa = $this->mahasiswa->getMahasiswa();
        }else {
            $mahasiswa = $this->mahasiswa->getMahasiswa($id);
        }

        if($mahasiswa){
            $this->response([
                'status' => true,
                'data' => $mahasiswa
            ], REST_Ccontroller::HTTP_OK);
        } else {
            $this->response([
                'status' => false,
                'message' => 'id not found'
            ], REST_Controller::HTTP_NOT_FOUND);
        }
    }
}
```

</td></tr>
</table>

2 Membuat file Mahasiswa_model.php

```
<?php

class Mahasiswa_model extends CI_model
{
    public function getMahasiswa($id = null)
    {
        if($id === null){
            return $this->db->get ('mahasiswa')->result_array();
        } else {
            return $this->db->get_where('mahasiswa', ['id'=>
$id])->result_array();
        }
    }
}
```

3 Copy link ke postman “localhost/webLanjut/code-kedelapan/api/Mahasiswa”

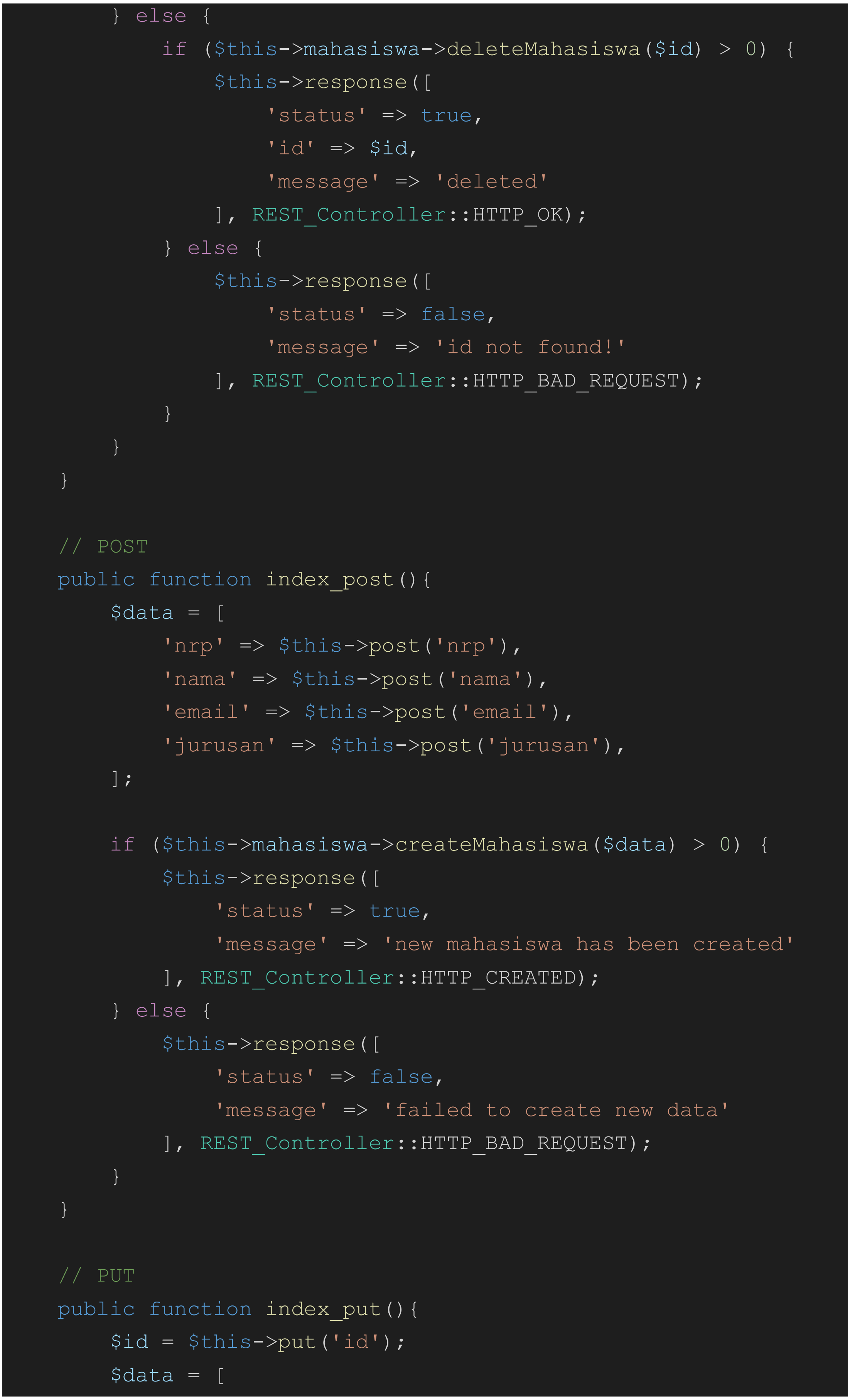


http://localhost/webLanjut/code-kedelapan/api/Mahasiswa

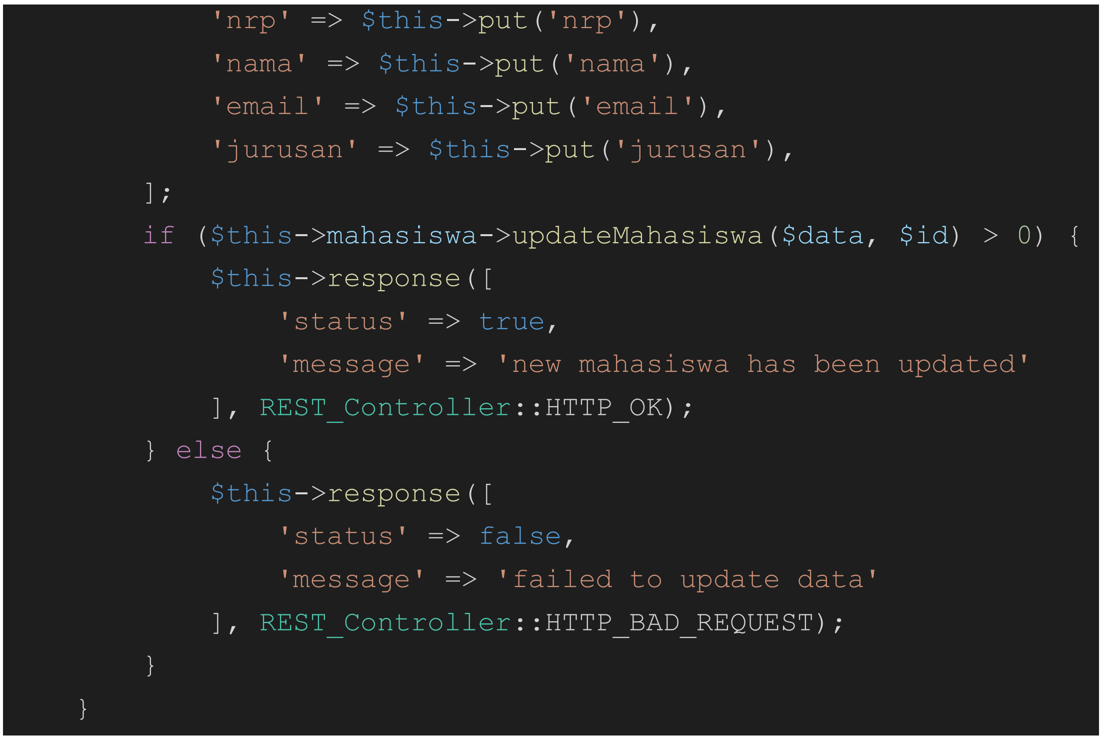
```
{
    "status": true,
    "data": [
        {
            "id": "1",
            "nrp": "043040001",
            "nama": "Doddy Ferdiansyah",
            "email": "doy@gmail.com",
            "jurusan": "Teknik Mesin"
        },
        {
            "id": "2",
            "nrp": "023040123",
            "nama": "Erik",
            "email": "erik@gmail.com",
            "jurusan": "Teknik Industri"
        },
        {
            "id": "3",
            "nrp": "043040321",
            "nama": "Rommy Fauzi",
            "email": "rommy@gmail.com",
            "jurusan": "Teknik Planologi"
        },
        {
            "id": "4",
            "nrp": "033040023",
```

4 Menambahkan fungsi Create, Update, dan Delete

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

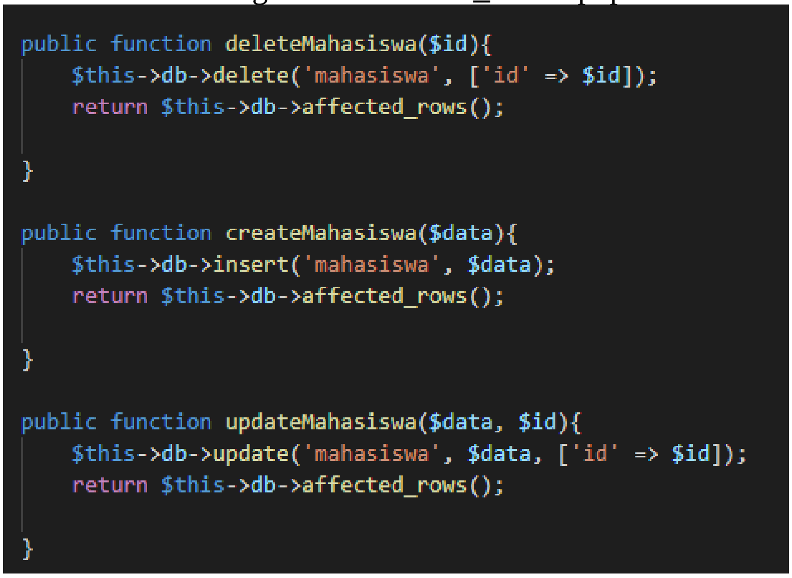


```php
        } else {
            if ($this->mahasiswa->deleteMahasiswa($id) > 0) {
                $this->response([
                    'status' => true,
                    'id' => $id,
                    'message' => 'deleted'
                ], REST_Controller::HTTP_OK);
            } else {
                $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);
        }
    }

    // PUT
    public function index_put(){
        $id = $this->put('id');
        $data = [
```



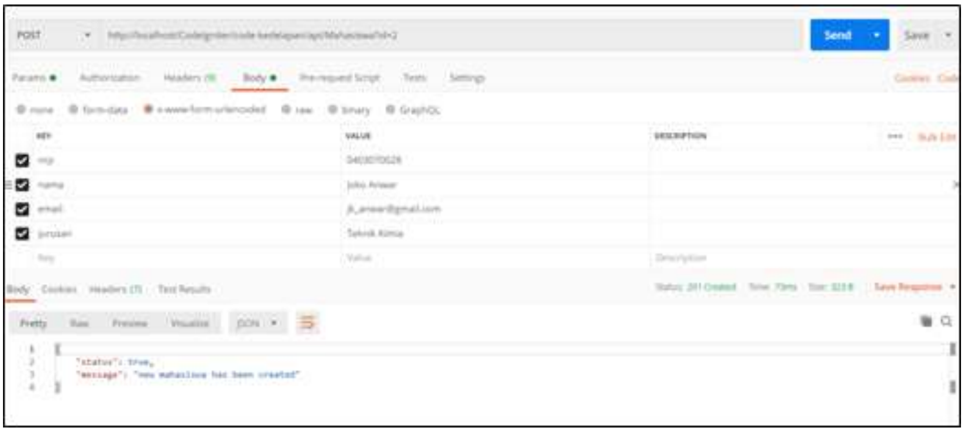
```
            'nrp' => $this->put('nrp'),
            'nama' => $this->put('nama'),
            'email' => $this->put('email'),
            'jurusan' => $this->put('jurusan'),
        ];
        if ($this->mahasiswa->updateMahasiswa($data, $id) > 0) {
            $this->response([
                'status'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

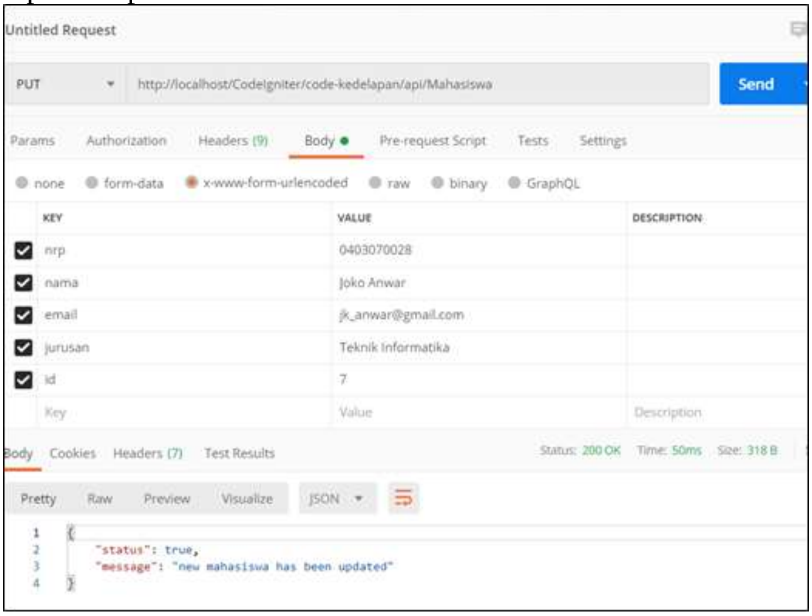
5 Menambahkan Fungsi di Mahasiswa_model.php



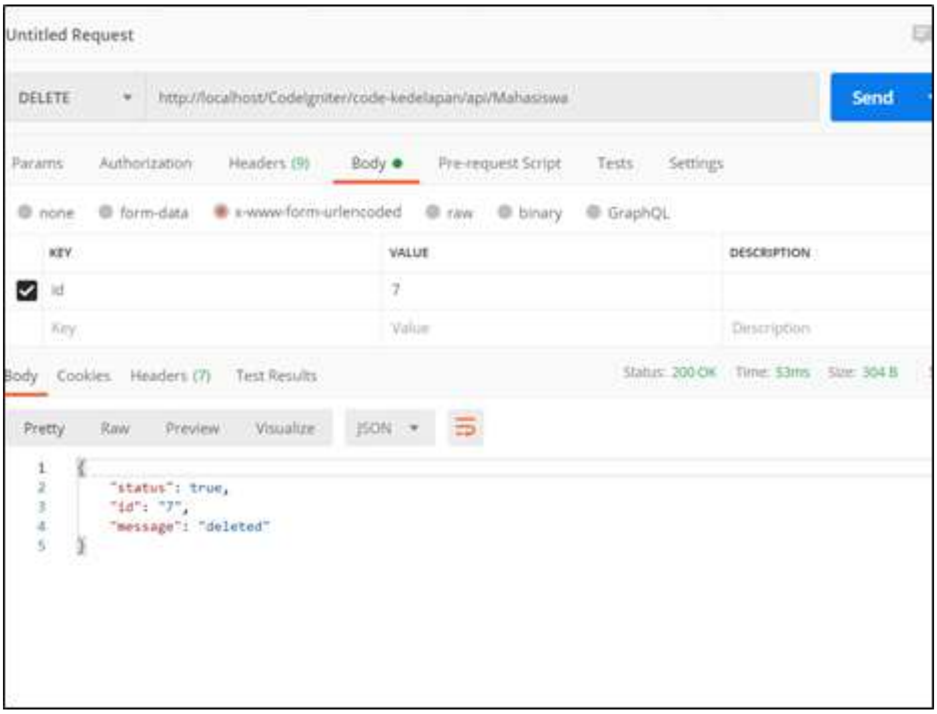
```
public function deleteMahasiswa($id){
    $this->db->delete('mahasiswa', ['id' => $id]);
    return $this->db->affected_rows();

}

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

	
	POST http://localhost/CodeIgniter/code-kedelapan/api/Mahasiswa?id=2
	Body: x-www-form-urlencoded
	☑ nrp 0403070028
	☑ nama Joko Anwar
	☑ email jk_anwar@gmail.com
	☑ jurusan Teknik Kimia
	Status: 201 Created Time: 73ms Size: 323 B
	{
	"status": true,
	"message": "new mahasiswa has been created"
	}

7	Operasi update
	
	Untitled Request
	PUT http://localhost/CodeIgniter/code-kedelapan/api/Mahasiswa
	Body: x-www-form-urlencoded
	☑ nrp 0403070028
	☑ nama Joko Anwar
	☑ email jk_anwar@gmail.com
	☑ jurusan Teknik Informatika
	☑ id 7
	Status: 200 OK Time: 50ms Size: 318 B
	{
	"status": true,
	"message": "new mahasiswa has been updated"
	}

8	Operasi Delete

Untitled Request

DELETE http://localhost/CodeIgniter/code-kedelapan/api/Mahasiswa

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	id	7	
	Key	Value	Description

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 53ms Size: 304 B

Pretty Raw Preview Visualize JSON

```
{
    "status": true,
    "id": "7",
    "message": "deleted"
}
```