

LAPORAN PRAKTIKUM
WEB LANJUT

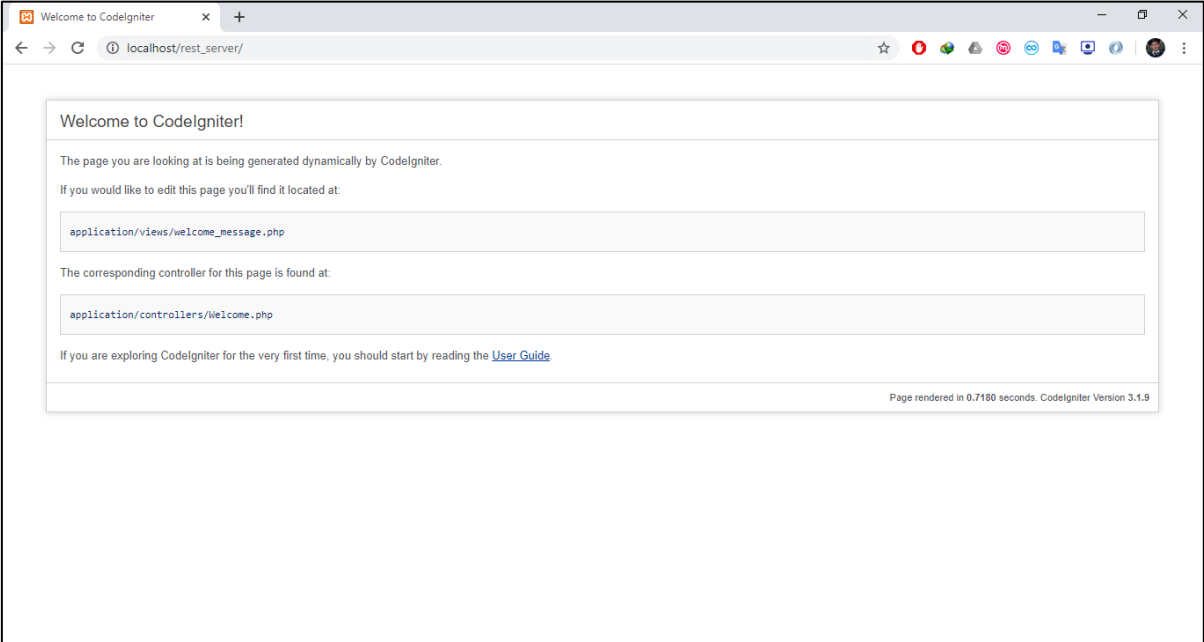
Implementasi Restful API



Disusun oleh:
MAHENDRA FIRMANSYAH
NIM : 1841720027

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

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	<p>Extract codeigniter yang sudah di unduh dan letakkan pada folder htdocs. 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>24 25 */ 26 \$config['base_url'] = ''; 27 28 /*</pre> <pre>37 */ 38 \$config['index_page'] = 'index.php'; 39 40 /*</pre> <p>Menjadi seperti berikut :</p> <pre>24 25 */ 26 \$config['base_url'] = 'http://localhost/rest_server/'; 27 28 /*</pre>

```

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' => '',
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;

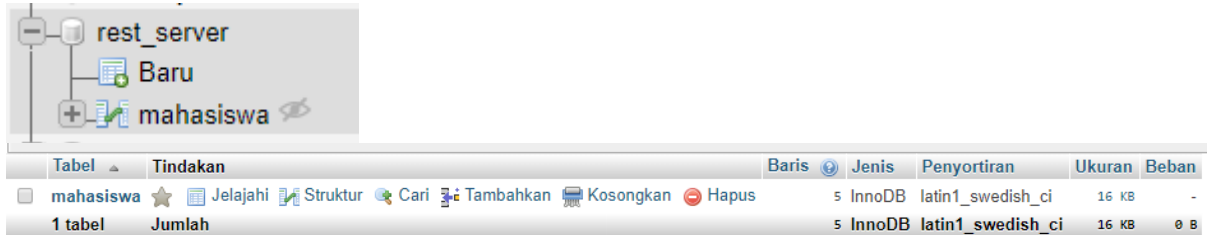
INSERT INTO `mahasiswa` (`id`, `nrp`, `nama`, `email`, `jurusan`) VALUES
(1, '043040001', 'Doddy Ferdiansyah', 'doy@gmail.com', 'Teknik Mesin'),
(2, '023040123', 'Erik', 'erik@gmail.com', 'Teknik Industri'),
(3, '043040321', 'Rommy Fauzi', 'rommy@gmail.com', 'Teknik Planologi'),
(4, '033040023', 'Fajar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

```

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

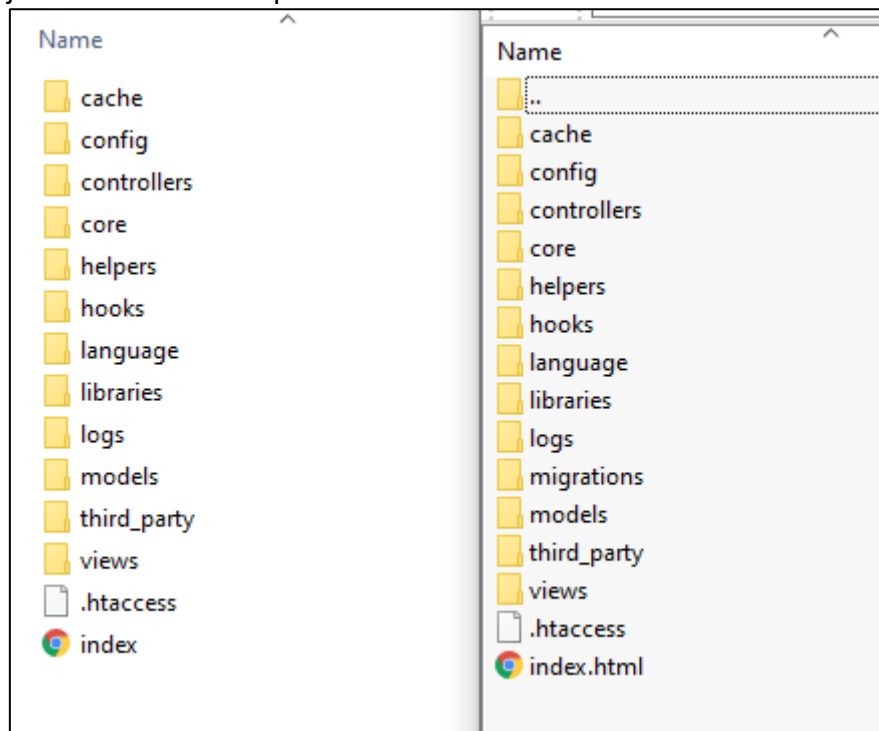
ALTER TABLE `mahasiswa`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=33;
```

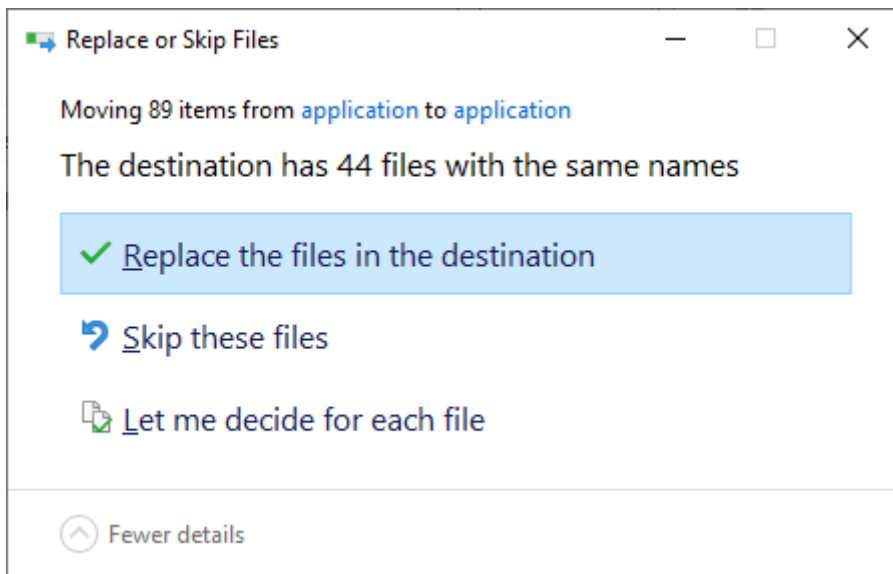
HASIL :



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.

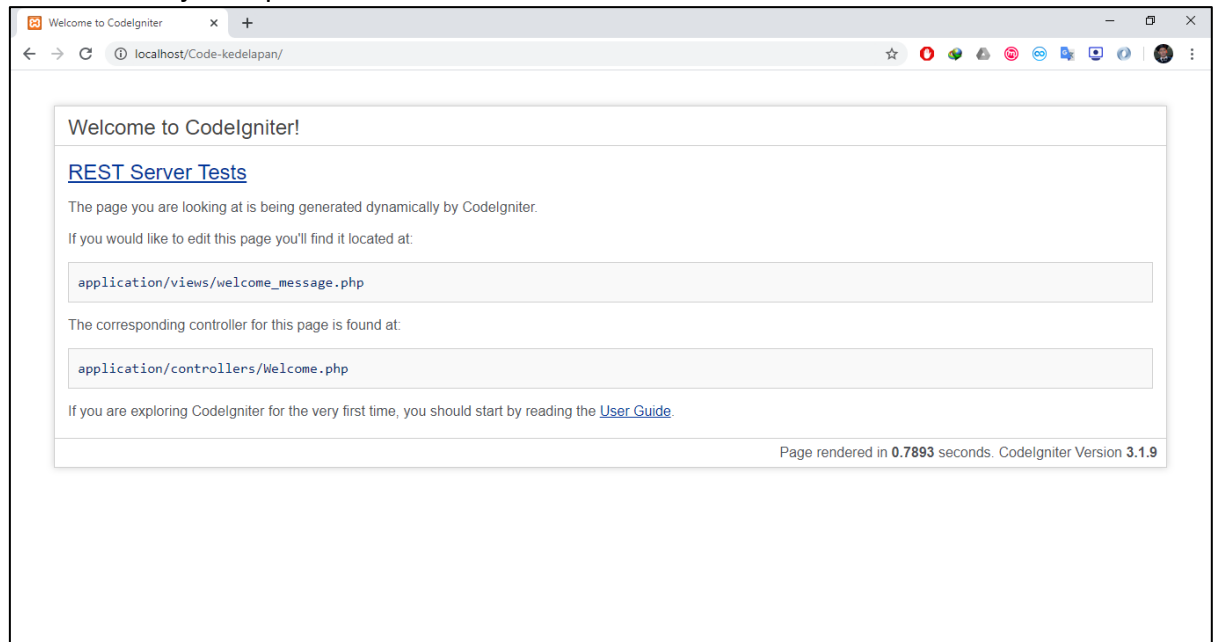




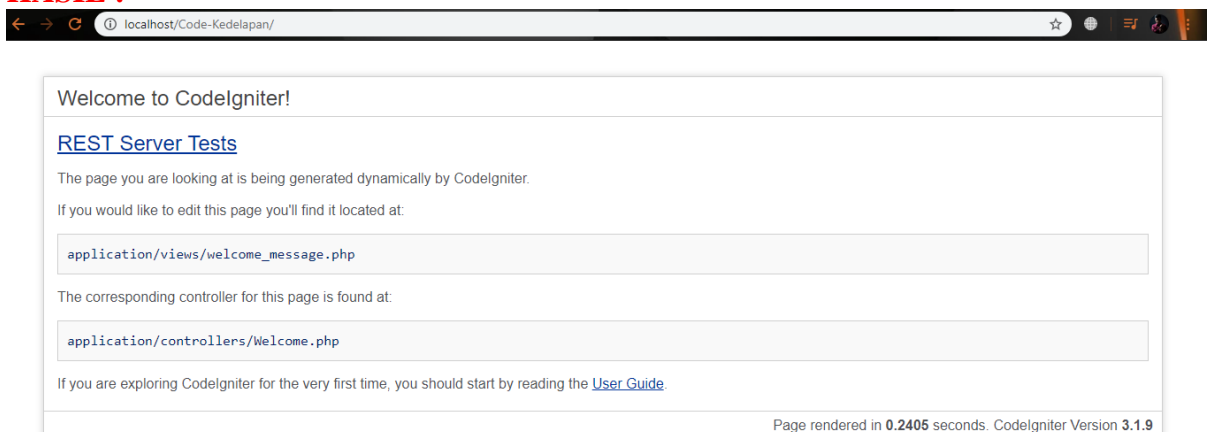
Setelah proses duplikasi file selesai, pastikan file-file yang kita sudah konfigurasi di awal tetap terkonfigurasi.

7

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



HASIL :



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> application > controllers > api > 📄 Mahasiswa.php 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 class Mahasiswa extends REST_Controller 9 { 10 public function __construct() 11 { 12 parent::__construct(); 13 \$this->load->model('Mahasiswa_model','mahasiswa'); 14 } 15 16 public function index_get() 17 { 18 \$id = \$this->get('id'); 19 if (\$id === null) { 20 \$mahasiswa = \$this->mahasiswa->getMahasiswa(); 21 } else { 22 \$mahasiswa = \$this->mahasiswa->getMahasiswa(\$id); 23 } 24 25 if(\$mahasiswa) { 26 \$this->response([27 'status' => true, 28 'data' => \$mahasiswa 29], REST_Controller::HTTP_OK); 30 } else { 31 \$this->response([32 'status' => false, 33 'message' => 'id not found' 34], REST_Controller::HTTP_NOT_FOUND); 35 } 36 } 37 38 } </pre> <p>HASIL :</p> <pre> <?php defined('BASEPATH') OR exit('No direct script access allowed'); </pre>

```

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_Controller::HTTP_OK);
        }else {
            $this->response([
                'status' => false,
                'message' => 'id not found'
            ], REST_Controller::HTTP_NOT_FOUND);
        }
    }
    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'),
    ];

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

```

2

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


```

application > models > Mahasiswa_model.php
1  <?php
2  |
3  class Mahasiswa_model extends CI_model
4  {
5      public function getMahasiswa($id = null)
6      {
7          if ($id === null) {
8
9
10         return $this->db->get ('mahasiswa')->result_array();
11     } else {
12         return $this->db->get_where('mahasiswa', ['id'=> $id])->result_array();
13     }
14 }
15 }

```

HASIL :

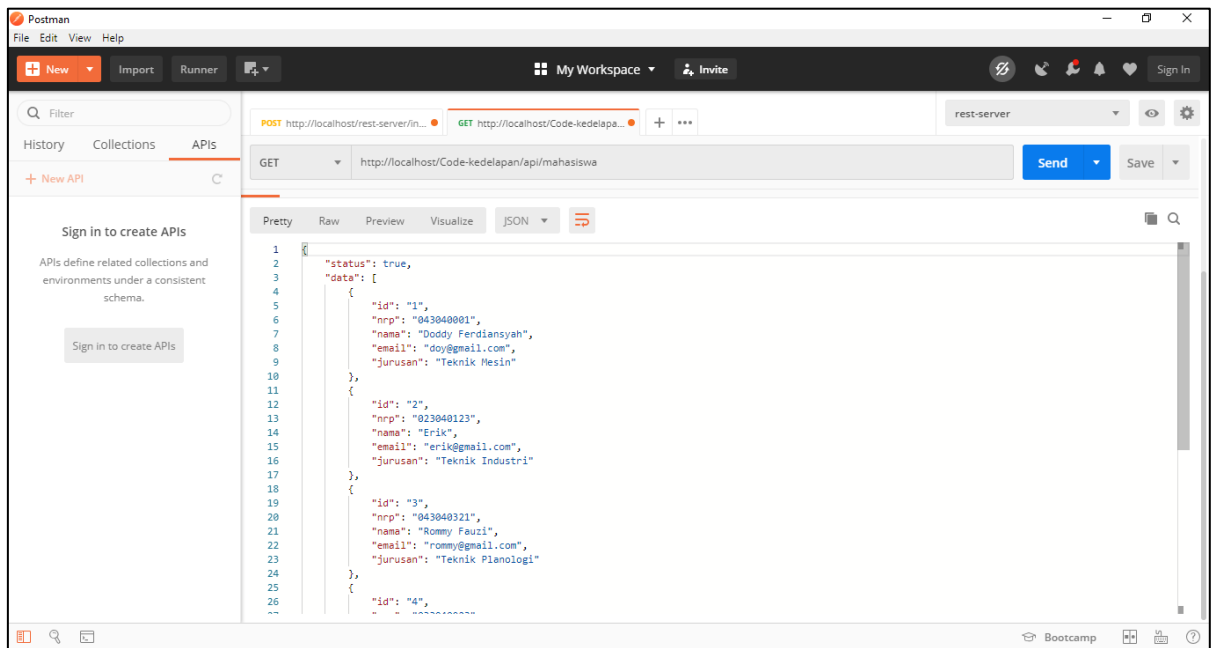
```

<?php

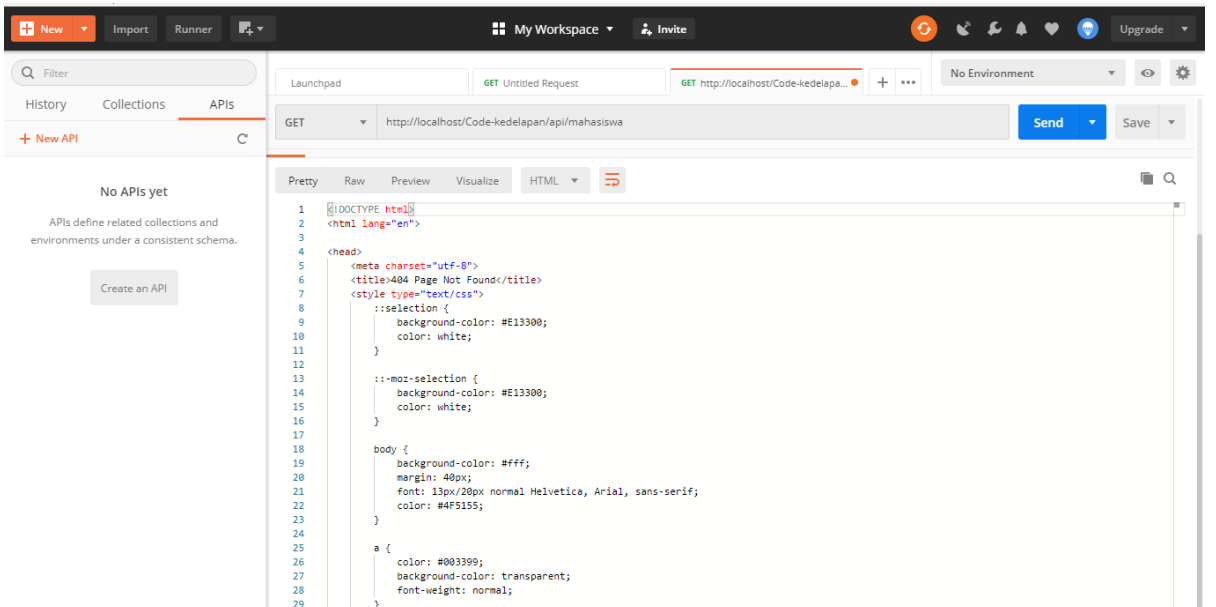
class Mahasiswa_model extends CI_Model {
    public function getMahasiswa($id = null) {
        if($id == null) {
            return $this->db->get('mahasiswa')->result_array();
        } else {
            return $this->db->get_where('mahasiswa', ['id'=>$id])->
>result_array();
        }
    }
    public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

```

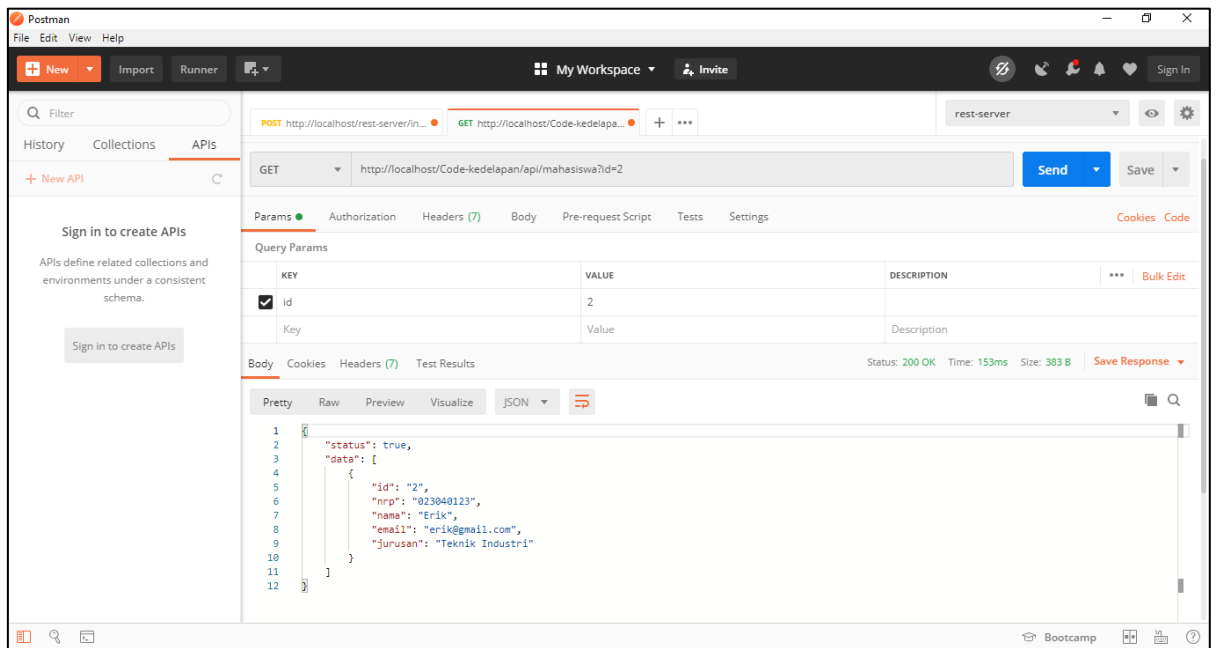
- 3 Buka aplikasi postman, masukkan alamat <http://localhost/Code-kedelapan/api/mahasiswa> dengan metode **GET**. dan jika berhasil maka tampilannya akan seperti ini:



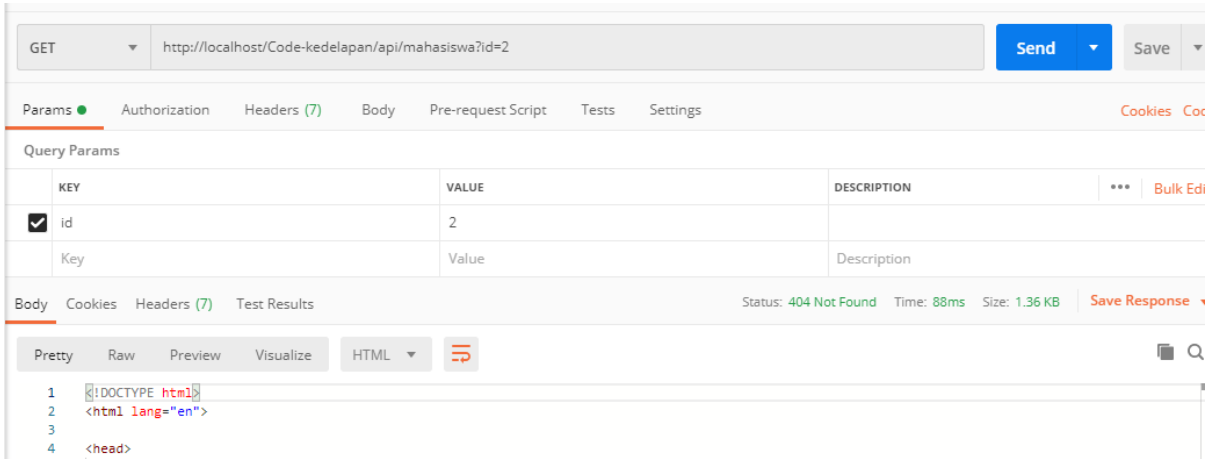
HASIL :



Kita juga dapat melakukan pencarian berdasarkan id, hasilnya akan seperti ini:



HASIL :



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

application > controllers > api > Mahasiswa.php

```
37
38     public function index_delete() {
39         $id=$this->delete('id');
40
41         if ($id===null) {
42             $this->response([
43                 'status' => false,
44                 'message' => 'provide an id!'
45             ], REST_Controller::HTTP_BAD_REQUEST);
46
47         } else {
48             if ($this->mahasiswa->deleteMahasiswa($id) > 0) {
49                 //ok
50                 $this->response([
51                     'status' => true,
52                     'id' => $id,
53                     'message' => 'deleted.'
54                 ], REST_Controller::HTTP_OK);
55             } else {
56                 // id not found
57                 $this->response([
58                     'status' => false,
59                     'message' => 'id not found!'
60                 ], REST_Controller::HTTP_BAD_REQUEST);
61             }
62         }
63     }
64 }
65
```

application > controllers > api > Mahasiswa.php

```
66     public function index_post()
67     {
68         $data = [
69             'nrp' => $this->post('nrp'),
70             'nama' => $this->post('nama'),
71             'email' => $this->post('email'),
72             'jurusan' => $this->post('jurusan'),
73
74         ];
75
76         if ($this->mahasiswa->createMahasiswa($data) > 0) {
77             $this->response([
78                 'status' => true,
79                 'message' => 'new mahasiswa has been created'
80             ], REST_Controller::HTTP_CREATED);
81         } else {
82             // id not found
83             $this->response([
84                 'status' => false,
85                 'message' => 'failde to create new data!'
86             ], REST_Controller::HTTP_BAD_REQUEST);
87         }
88     }
89
90 }
91
```

application > controllers > api > Mahasiswa.php

```
92     public function index_put()
93     {
94         $id = $this->put('id');
95         $data = [
96             'nrp' => $this->put('nrp'),
97             'nama' => $this->put('nama'),
98             'email' => $this->put('email'),
99             'jurusan' => $this->put('jurusan'),
100
101         ];
102
103         if ($this->mahasiswa->updateMahasiswa($data, $id) > 0) {
104             $this->response([
105                 'status' => true,
106                 'message' => 'data mahasiswa has been updated'
107             ], REST_Controller::HTTP_OK);
108         } else {
109             // id not found
110             $this->response([
111                 'status' => false,
112                 'message' => 'failde to update data!'
113             ], REST_Controller::HTTP_BAD_REQUEST);
114         }
115     }
116 }
117 }
```

HASIL :

```
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'),
        ];

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

```

5

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

```

application > models > Mahasiswa_model.php
15
16     public function deleteMahasiswa($id)
17     {
18         $this->db->delete('mahasiswa', ['id' => $id]);
19         return $this->db->affected_rows();
20     }
21
22     public function createMahasiswa($data)
23     {
24         $this->db->insert('mahasiswa', $data);
25         return $this->db->affected_rows();
26     }
27
28     public function updateMahasiswa($data, $id)
29     {
30         $this->db->update('mahasiswa', $data, ['id' => $id]);
31         return $this->db->affected_rows();
32     }
33 }

```

HASIL :

```

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:

Params Authorization Headers (9) Body F

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐

KEY
<input checked="" type="checkbox"/> nrp
<input checked="" type="checkbox"/> nama
<input checked="" type="checkbox"/> email
<input checked="" type="checkbox"/> jurusan

Postman

File Edit View Help

New Import Runner My Workspace Invite

rest-server

Filter

History Collections APIs

+ New API

Sign in to create APIs

APIs define related collections and environments under a consistent schema.

Sign in to create APIs

POST http://localhost/Code-kedelapan/api/mahasiswa

Send Save

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

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nrp	0403070028	
<input checked="" type="checkbox"/> nama	Joko Anwar	
<input checked="" type="checkbox"/> email	jk_anwar@gmail.com	
<input checked="" type="checkbox"/> jurusan	Teknik Kimia	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 201 Created Time: 133ms Size: 325 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": true,
3   "message": "new mahasiswa has been created"
4 }

```

Bootcamp

HASIL :

GET http://localhost/Code-kedelapan/api/maha...

Params Authorization Headers (7) Body

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐

KEY
<input checked="" type="checkbox"/> nrp
<input checked="" type="checkbox"/> nama
<input checked="" type="checkbox"/> email
<input checked="" type="checkbox"/> jurusan

POST http://localhost/Code-kedelapan/api/mahasiswa

Send Save

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

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

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> nrp	00098080			
<input checked="" type="checkbox"/> nama	Mahendra Firmansyah			
<input checked="" type="checkbox"/> email	@mahendrafir69@gmail.com			
<input checked="" type="checkbox"/> jurusan	Teknik Informatika			
..	...	-		

7

Lakukan juga operasi **Update** data pada aplikasi postman, masukkan parameter id seperti berikut:

Postman

File Edit View Help

New Import Runner My Workspace Invite

PUT http://localhost/Code-kedelapan/api/mahasiswa

Send Save

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

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

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> nrp	0403070028			
<input checked="" type="checkbox"/> nama	Joko Anwar			
<input checked="" type="checkbox"/> email	jk_anwar@gmail.com			
<input checked="" type="checkbox"/> jurusan	Teknik Informatika			
<input checked="" type="checkbox"/> id	39			
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 126ms Size: 321 B Save Response

Pretty Raw Preview Visualize JSON

```

1
2  "status": true,
3  "message": "data mahasiswa has been updated"
4

```

HASIL :

PUT http://localhost/Code-kedelapan/api/mahasiswa

Send Save

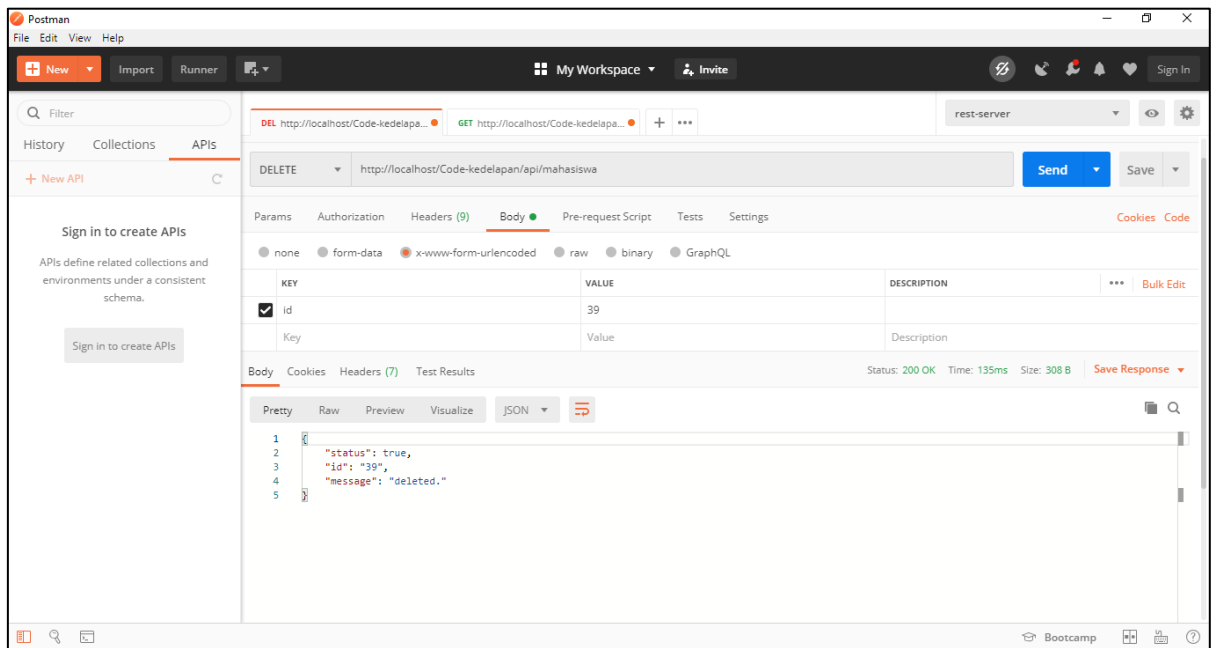
Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

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

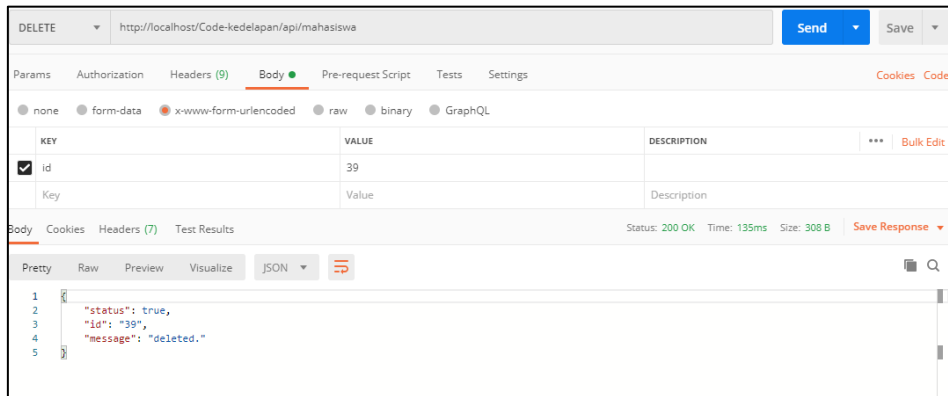
KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> nrp	00098080			
<input checked="" type="checkbox"/> nama	Mahendra Firmansyah			
<input checked="" type="checkbox"/> email	@mahendrafir69@gmail.com			
<input checked="" type="checkbox"/> jurusan	Teknik Informatika			
<input checked="" type="checkbox"/> id	12			
Key	Value	Description		

8

Lakukan juga operasi **Delete** data pada aplikasi postman, parameter yang dimasukkan cukup id saja, seperti berikut:



HASIL :



-- Selamat Mengerjakan --