

# **LAPORAN JOBSHEET 8**

**(Implementasi Restful API)**

**Disusun sebagai**

**MATA KULIAH :**

**Pemrograman Web Lanjut**



**TI-2B**

**OLEH**

**WILDAN DAWAM BASH (1841720166)**

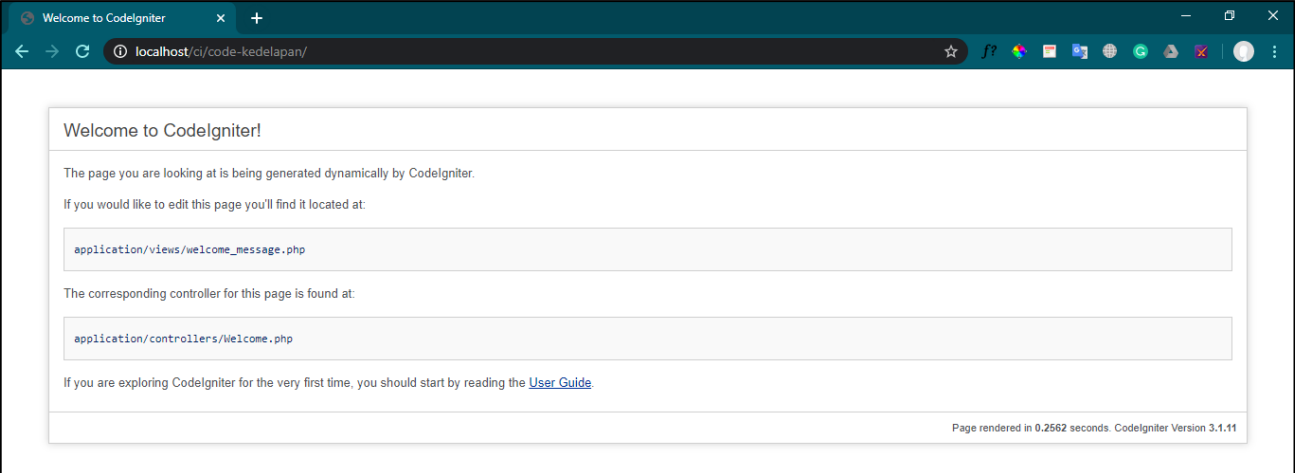
**PROGAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2020**

## Bagian 1 – Instalasi library restful codeigniter

| No | Keterangan   |
|----|--|
| 1  | Silahkan unduh aplikasi Postman terlebih dahulu.   |
| 2  | <p>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 :</p> <p><a href="https://github.com/dimas1984/CI/tree/master/Belajar-rest">https://github.com/dimas1984/CI/tree/master/Belajar-rest</a></p>                                    |
| 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  | <p>Selanjutnya lakukan konfigurasi dasar pada <b>config.php</b> seperti berikut:</p> <pre>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>25  */ 26  \$config['base_url'] = 'http://localhost/ci/code-kedelapan/'; 27 28  /*</pre> <pre>36  / 37  */ 38  \$config['index_page'] = ''; 39 40  /*</pre> |

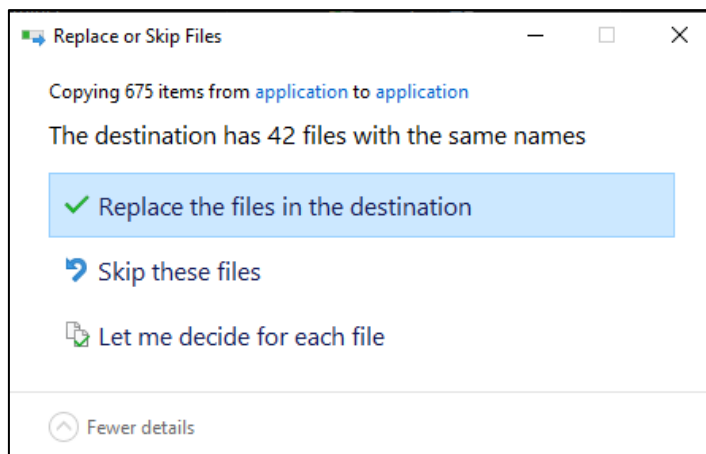
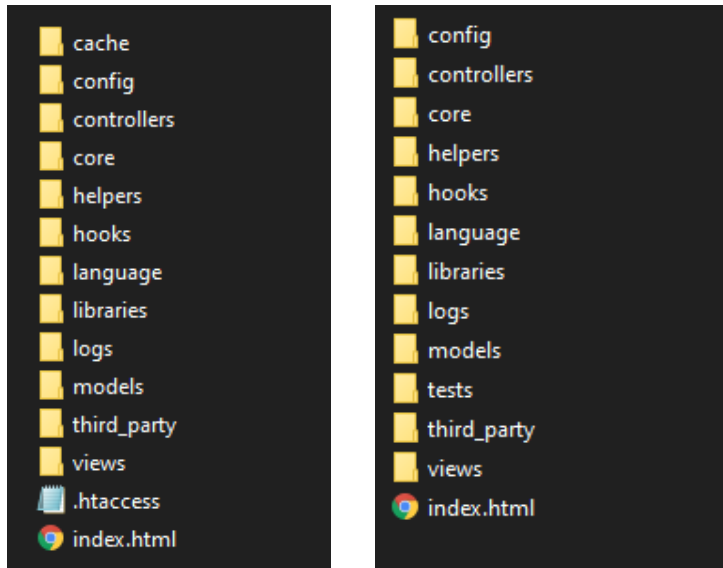
```
60 */
61 $autoload['libraries'] = array();
62
63 /*
64
65 | $autoload['helper'] = array(
66 */
67 $autoload['helper'] = array();
68
69 /*
```

```
60  */
61  $autoload['libraries'] = array('database');
62
63  /*
64
65  91  */
92  $autoload['helper'] = array('url');
93
94  /*
```

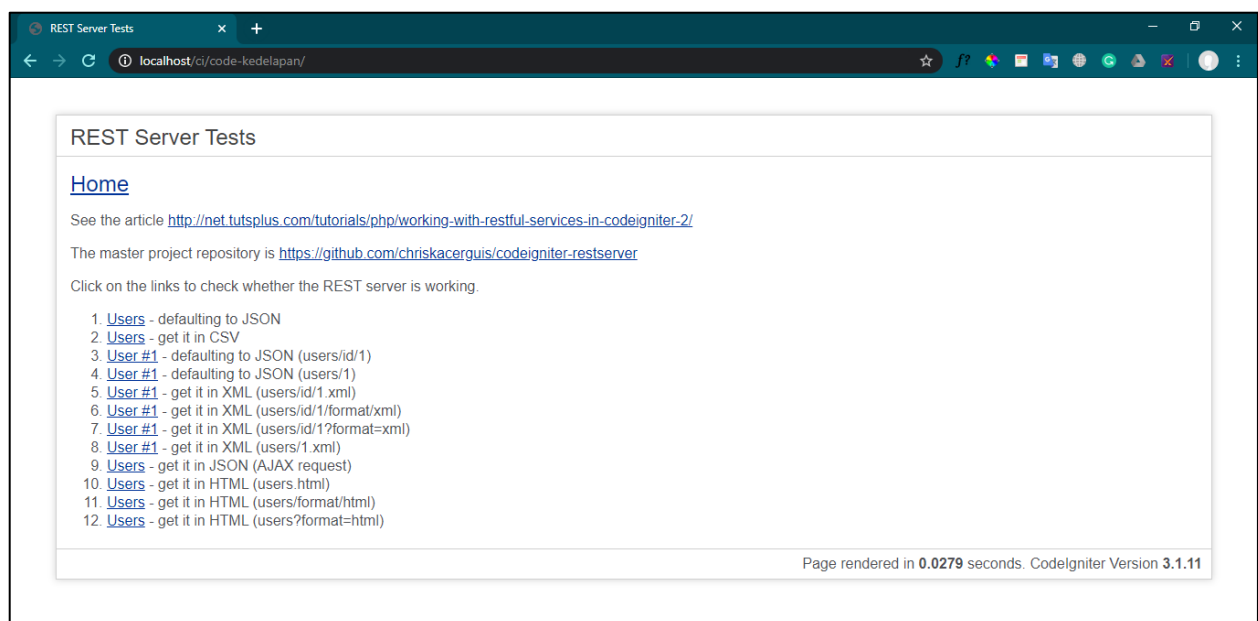
```
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'rest_server',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
);
```

```
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.



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

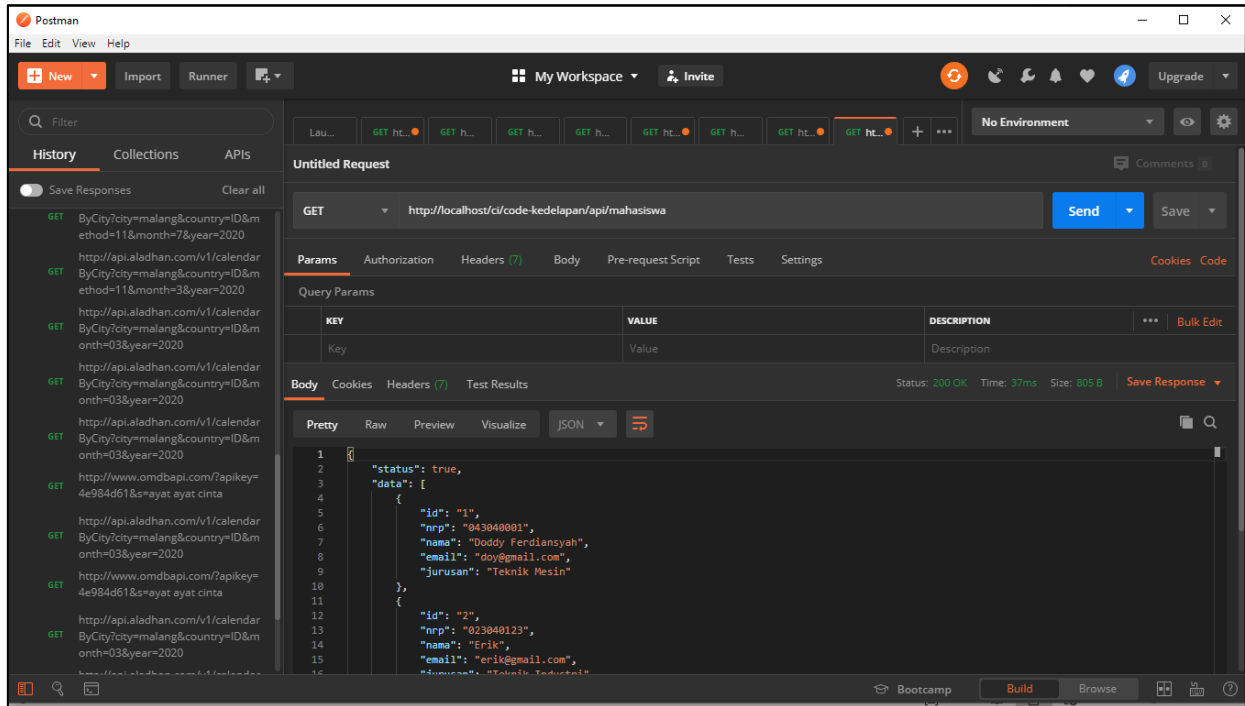


- 8 Tahap instalasi Library Rest Server selesai

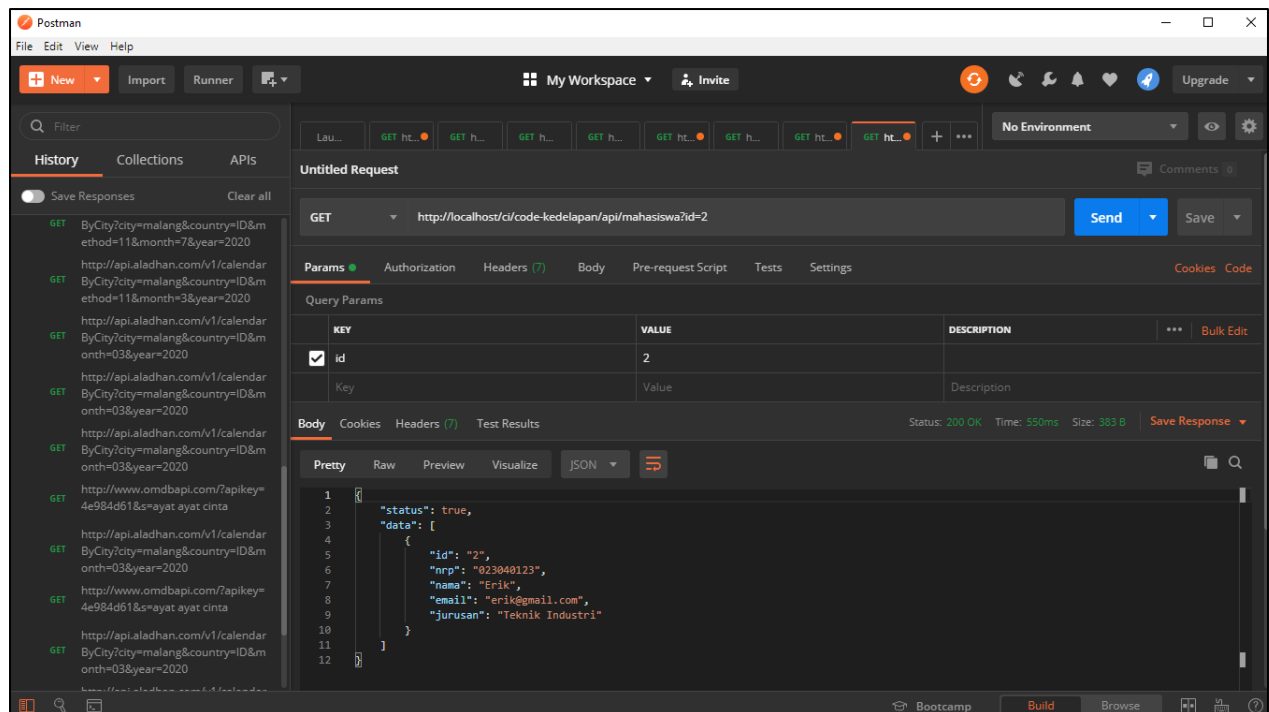
## Bagian 2 – Membuat Aplikasi CRUD Restful Codeigniter

| No | Keterangan  |
|----|---|
| 1  | <p>Buatlah file <b>Mahasiswa.php</b> pada <b>controller/api</b>. Selanjutnya tuliskan kode berikut:</p> <pre data-bbox="185 271 1091 1420">&lt;?php defined('BASEPATH') OR exit('No direct script access allowed');  use Restserver\Libraries\REST_Controller;  require APPPATH . 'libraries/Rest_Controller.php'; require APPPATH . 'libraries/Format.php';  class Mahasiswa extends REST_Controller {      public function __construct()     {         parent::__construct();         \$this-&gt;load-&gt;model('Mahasiswa_model','mahasiswa');     }      public function index_get()     {         \$id = \$this-&gt;get('id');         if(\$id === null){             \$mahasiswa = \$this-&gt;mahasiswa-&gt;getMahasiswa();         }else{             \$mahasiswa = \$this-&gt;mahasiswa-&gt;getMahasiswa(\$id);         }          if(\$mahasiswa){             \$this-&gt;response([                 'status' =&gt; true,                 'data' =&gt; \$mahasiswa             ], REST_Controller::HTTP_OK);         }else{             \$this-&gt;response([                 'status' =&gt; false,                 'message' =&gt; 'id not found'             ], REST_Controller::HTTP_NOT_FOUND);         }     }  }</pre> |
| 2  | <p>Selanjutnya Buatlah file <b>Mahasiswa_model.php</b> pada model. Selanjutnya tuliskan kode berikut:</p> <pre data-bbox="185 1518 1347 2013">&lt;?php  defined('BASEPATH') OR exit('No direct script access allowed');  class Mahasiswa_model extends CI_Model {      public function getMahasiswa(\$id = null){         if(\$id == null){             return \$this-&gt;db-&gt;get('mahasiswa')-&gt;result_array();         }else{             return \$this-&gt;db-&gt;get_where('mahasiswa', ['id'=&gt;\$id])-&gt;result_array();         }     }  }</pre>  |

- 3 Buka aplikasi postman, masukkan alamat <http://localhost/ci/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**:

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

```
62     public function index_post(){
63         $data = [
64             'nrp' => $this->post('nrp'),
65             'nama' => $this->post('nama'),
66             'email' => $this->post('email'),
67             'jurusan' => $this->post('jurusan'),
68         ];
69
70         if($this->mahasiswa->createMahasiswa($data) > 0){
71             $this->response([
72                 'status' => true,
73                 'message' => 'new mahasiswa has been created'
74             ], REST_Controller::HTTP_CREATED);
75         }else{
76             $this->response([
77                 'status' => false,
78                 'message' => 'failed to create new data'
79             ], REST_Controller::HTTP_BAD_REQUEST);
80         }
81     }
82 }
```

```

83     public function index_put(){
84         $id = $this->put('id');
85         $data = [
86             'nrp' => $this->put('nrp'),
87             'nama' => $this->put('nama'),
88             'email' => $this->put('email'),
89             'jurusan' => $this->put('jurusan'),
90         ];
91
92         if($this->mahasiswa->updateMahasiswa($data, $id) > 0){
93             $this->response([
94                 'status' => true,
95                 'message' => 'data mahasiswa has been updated'
96             ], REST_Controller::HTTP_OK);
97         }else{
98             $this->response([
99                 'status' => false,
100                 'message' => 'failed to update data'
101             ], REST_Controller::HTTP_BAD_REQUEST);
102         }
103     }
104

```

5 Dan lanjutkan kode pada file **Mahasiswa\_model.php**:

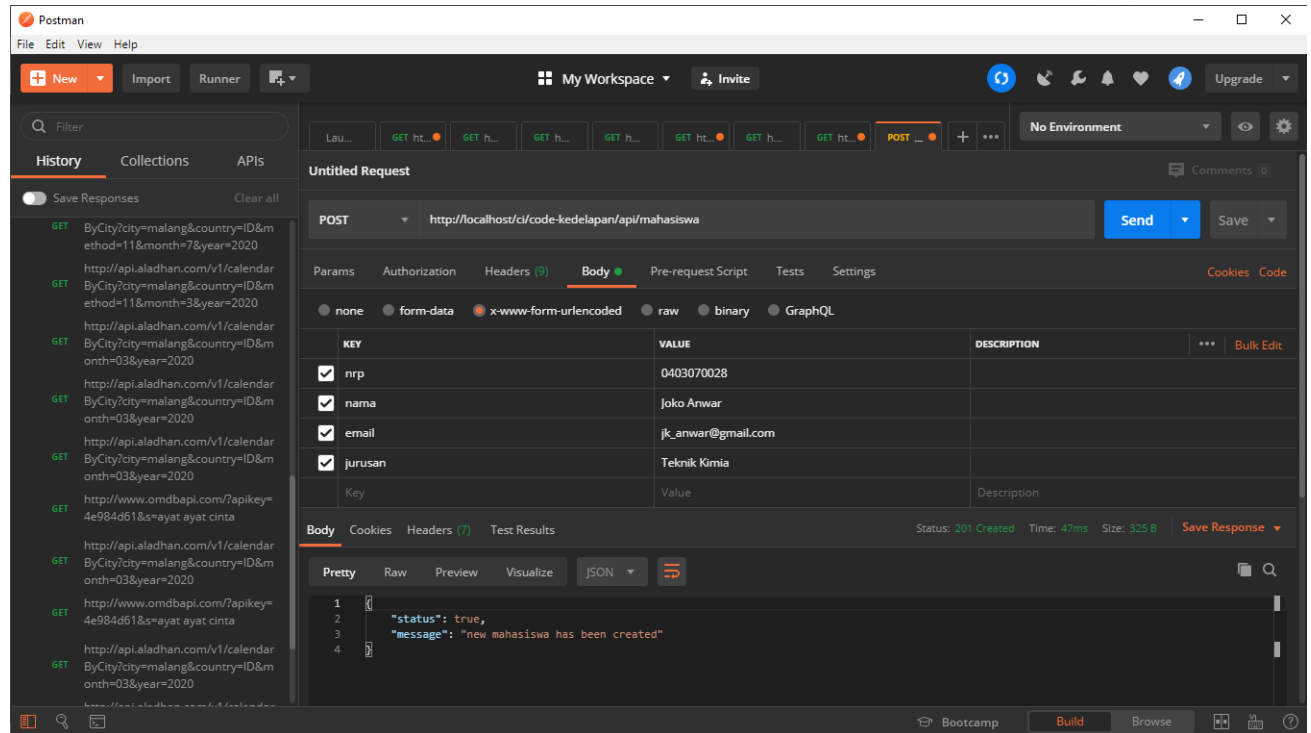
```

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

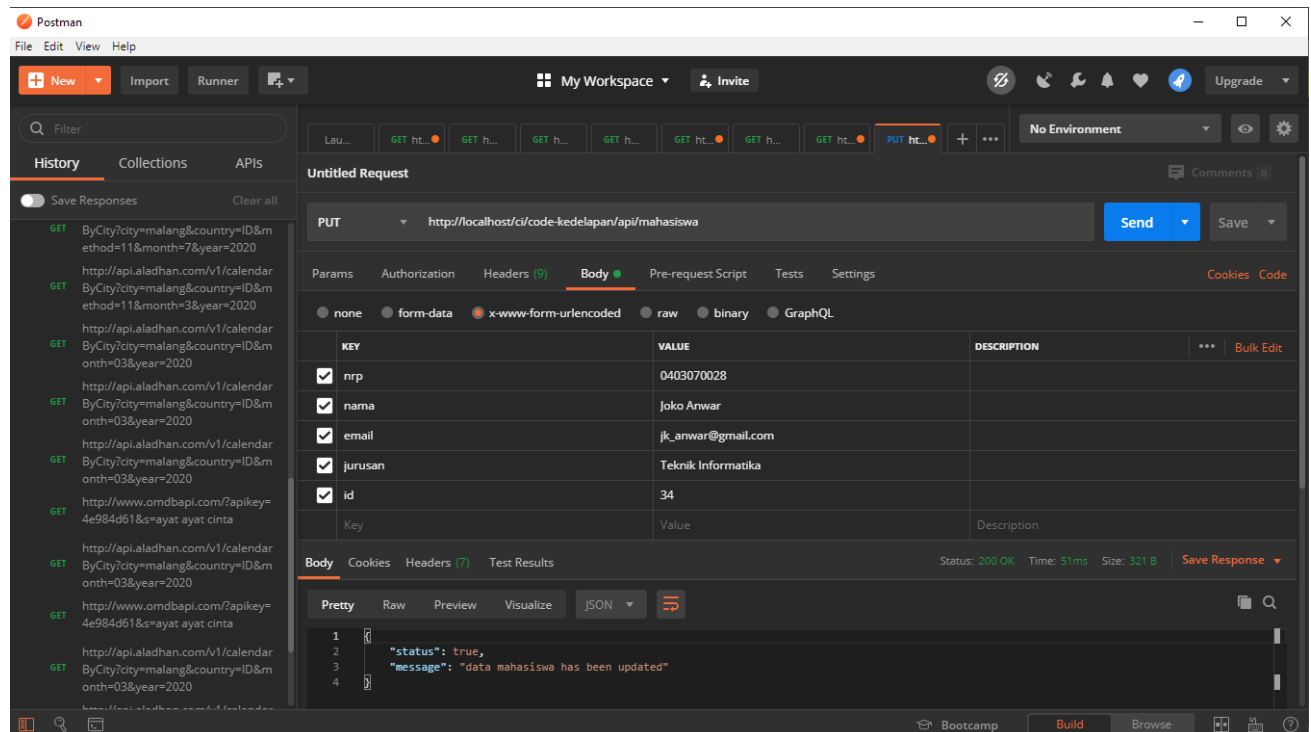
```



- 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:

