



Jurusan Teknologi Informasi Politeknik Negeri Malang  
**Jobsheet-8: Implementasi Restful API**  
**Mata Kuliah Pemrograman Web Lanjut**  
**Pengampu:** Tim Ajar Pemrograman Web Lanjut  
*Februari 2020*

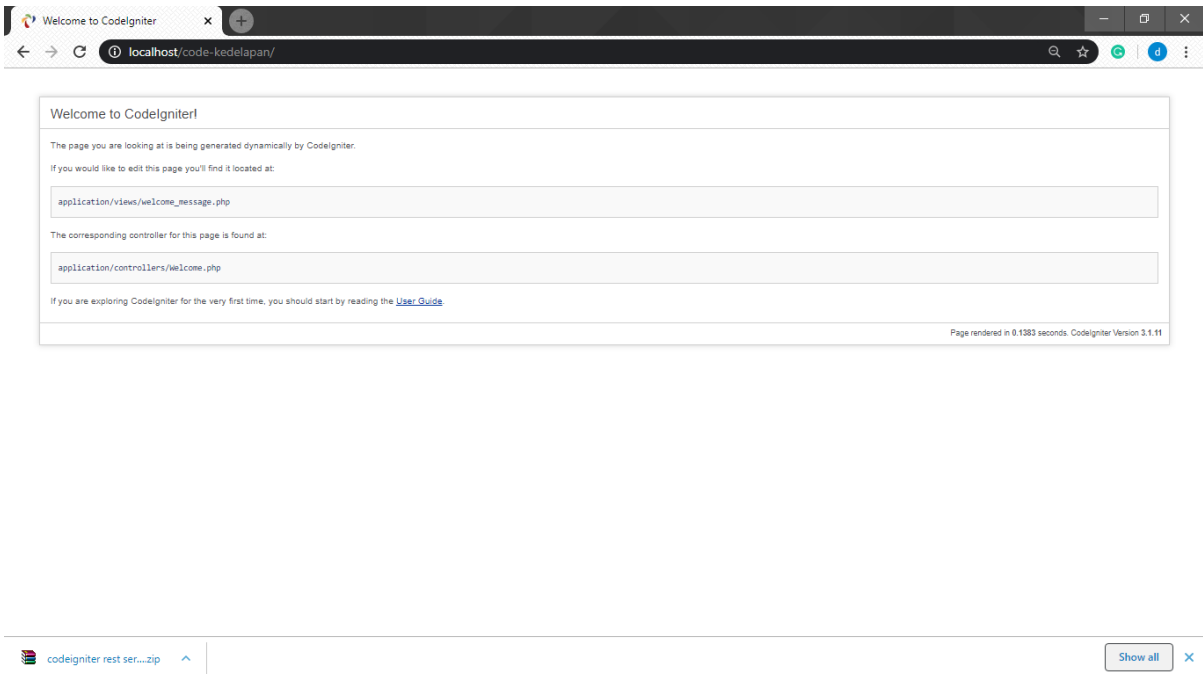
---

Nama : Darin Zahira Aflah

Kelas : TI-2A/09

NIM : 1841720033

**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 : <a href="https://drive.google.com/drive/folders/10YaCrqrqG2M1Zk0r-trln0hROP9q0hpi?usp=sharing">https://drive.google.com/drive/folders/10YaCrqrqG2M1Zk0r-trln0hROP9q0hpi?usp=sharing</a>
3	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: 
4	Selanjutnya lakukan konfigurasi dasar pada <b>config.php</b> seperti berikut:

```

24 |
25 */
26 $config['base_url'] = '';
27
28 /*
37 */
38 $config['index_page'] = 'index.php';
39
40 /*

```

Menjadi seperti berikut :

```

25 */
26 $config['base_url'] = 'http://localhost/rest_server/';
27
28 /*
29 |-----
30 | Index File
31 |-----
32 |
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 |
37 */
38 $config['index_page'] = '';

```

Dan file **autoload.php**

```

60 */
61 $autoload['libraries'] = array();
62
63 /*

```

```

91 */
92 $autoload['helper'] = array();
93
94 /*

```

Menjadi

```

57 | In the controller:
58 |
59 | $autoload['libraries'] = array('user_agent' => 'ua');
60 */
61 $autoload['libraries'] = array('database');
62

```

```

89 |
90 | $autoload['helper'] = array('url', 'file');
91 */
92 $autoload['helper'] = array('url');
93
94 /*
95

```

Dan database.php seperti berikut:

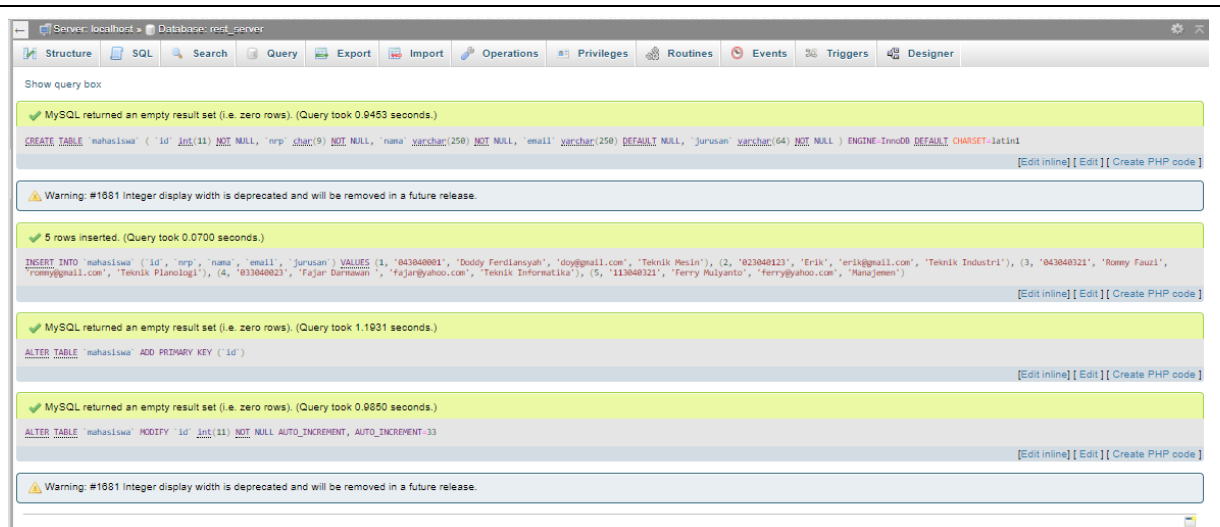
```
76 $db['default'] = array(  
77     'dsn'      => '',  
78     'hostname' => 'localhost',  
79     'username' => 'root',  
80     'password' => '',  
81     'database' => 'rest_server',  
82     'dbdriver' => 'mysqli',
```

```
75  
76 $db['default'] = array(  
77     'dsn'      => '',  
78     'hostname' => 'localhost',  
79     'username' => 'root',  
80     'password' => '',  
81     'database' => 'rest_server',  
82     'dbdriver' => 'mysqli',  
83     'dbprefix' => '',  
84     'pconnect' => FALSE,
```

5

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

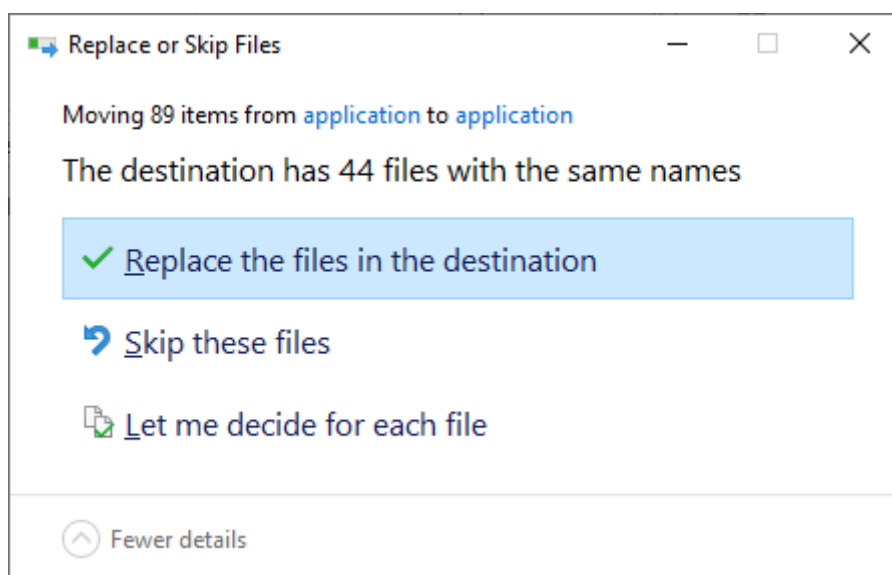
```
CREATE TABLE `mahasiswa` (  
    `id` int(11) NOT NULL,  
    `nrp` char(9) NOT NULL,  
    `nama` varchar(250) NOT NULL,  
    `email` varchar(250) DEFAULT NULL,  
    `jurusan` varchar(64) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
INSERT INTO `mahasiswa` (`id`, `nrp`, `nama`, `email`, `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;
```



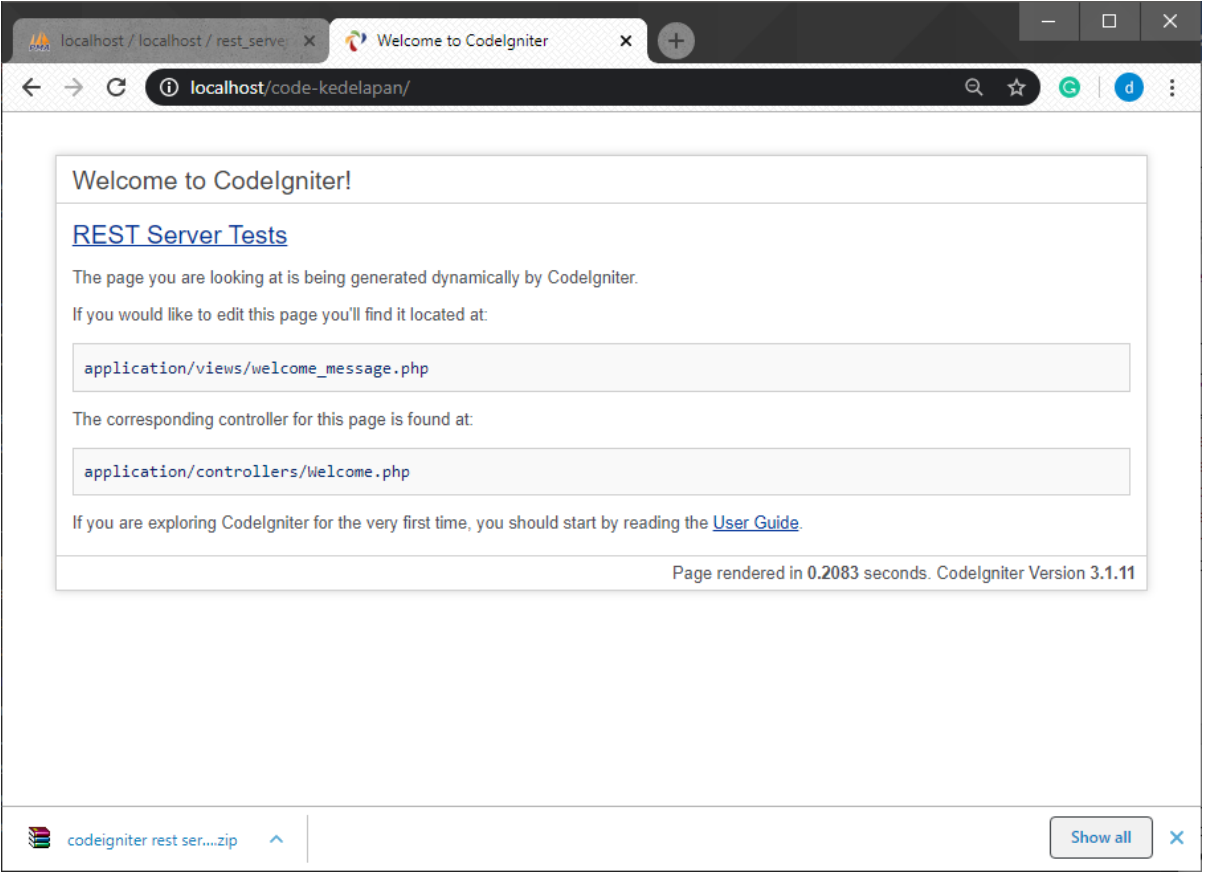
6

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.

Name	Date modified	Type	Size
application	24/03/2020 12:20	File folder	
system	19/09/2019 5:08	File folder	
user_guide	19/09/2019 5:08	File folder	
.editorconfig	19/09/2019 5:08	EDITORCONFIG File	1 KB
.gitignore	19/09/2019 5:08	Text Document	1 KB
composer.json	19/09/2019 5:08	JSON File	1 KB
contributing.md	19/09/2019 5:08	MD File	7 KB
index.php	19/09/2019 5:08	PHP File	11 KB
license	19/09/2019 5:08	Text Document	2 KB
readme.rst	19/09/2019 5:08	RST File	3 KB



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

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> 
8	Tahap instalasi Library Rest Server selesai

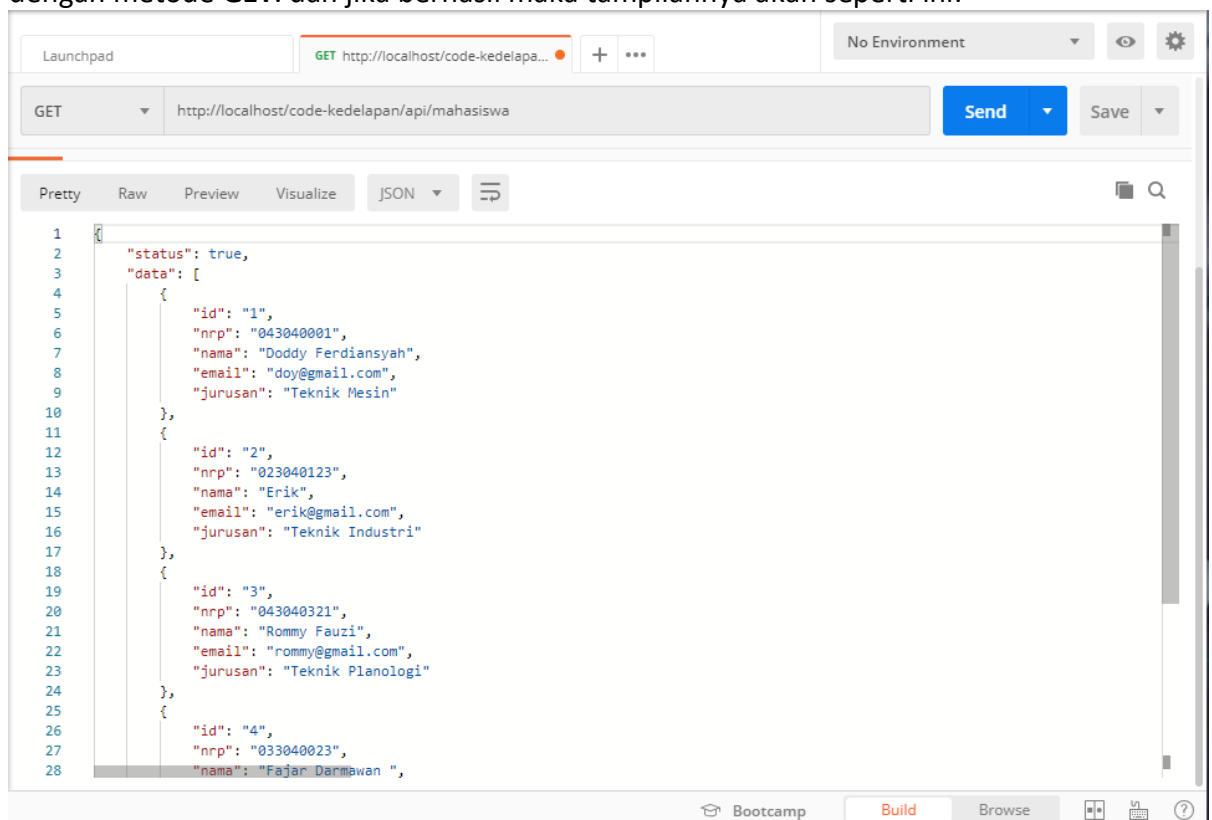
### Praktikum – Bagian 2: Membuat Aplikasi CRUD Restful Codeigniter

Langkah	Keterangan
1	Buatlah file <b>Mahasiswa.php</b> pada <b>controller/api</b> . Selanjutnya tuliskan kode berikut:

	<div> <div> Mahasiswa.php × autoload.php Mahasiswa_model.php </div> <div> application &gt; controllers &gt; api &gt; Mahasiswa.php </div> <div> <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-&gt;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, 33                 'message' =&gt; 'id not found' 34             ], REST_Controller::HTTP_NOT_FOUND); 35         } 36     } 37 38 } 39 40 /* End of file Mahasiswa.php */ 41 42 ?&gt; </pre> </div> </div>
2	<p>Selanjutnya Buatlah file <b>Mahasiswa_model.php</b> pada model. Selanjutnya tuliskan kode berikut:</p>

```
Mahasiswa.php  Mahasiswa_model.php X
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              return $this->db->get('mahasiswa')->result_array();
9          } else {
10             return $this->db->get_where('mahasiswa', ['id'=>$id])->result_array();
11         }
12     }
13
14 }
15
16 /* End of file Mahasiswa_model.php */
17
18 ?>
```

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

Launchpad GET http://localhost/code-kedelapa... No Environment

Untitled Request Comments 0

GET http://localhost/code-kedelapan/api/mahasiswa?id=2 Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	2			
	Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 850ms Size: 383 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": true,
3   "data": [
4     {
5       "id": "2",
6       "nrrp": "023040123",
7       "nama": "Erik",
8       "email": "erik@gmail.com",
9       "jurusan": "Teknik Industri"
10    }
11  ]
12 }

```

Bootcamp Build Browse

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                 //ok
49                 $this->response([
50                     'status' => true,
51                     'id' => $id,
52                     'message' => 'deleted.'
53                 ], REST_Controller::HTTP_OK);
54             }else{
55                 //id not found
56                 $this->response([
57                     'status' =>false,
58                     'message' => 'id not found!'
59                 ], REST_Controller::HTTP_BAD_REQUEST);
60             }
61         }
62     }
63
64     public function index_post(){
65         $data = [
66             'nrp' => $this->post('nrp'),
67             'nama' => $this->post('nama'),
68             'email' => $this->post('email'),
69             'jurusan' => $this->post('jurusan'),

```

```

70     ];
71
72     if($this->mahasiswa->createMahasiswa($data) > 0){
73         $this->response([
74             'status' => true,
75             'message' => 'new mahasiswa has been created'
76         ], REST_Controller::HTTP_CREATED);
77     }else{
78         //id not found
79         $this->response([
80             'status' => false,
81             'message' => 'failed to create new data!'
82         ], REST_Controller::HTTP_BAD_REQUEST);
83     }
84 }
85
86 public function index_put(){
87     $id = $this->put('id');
88     $data = [
89         'nrp' => $this->post('nrp'),
90         'nama' => $this->post('nama'),
91         'email' => $this->post('email'),
92         'jurusan' => $this->post('jurusan'),
93     ];
94
95     if($this->mahasiswa->updateMahasiswa($data, $id) > 0){
96         $this->response([
97             'status' => true,
98             'message' => 'new mahasiswa has been created'
99         ], REST_Controller::HTTP_OK);
100     }else{
101         //id not found
102         $this->response([
103             'status' => false,
104             'message' => 'failed to update data!'
105         ], REST_Controller::HTTP_BAD_REQUEST);
106     }
107
108 }
109
110 }
111
112 /* End of file Mahasiswa.php */
113
114 ?>

```

5

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

```

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

```

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:

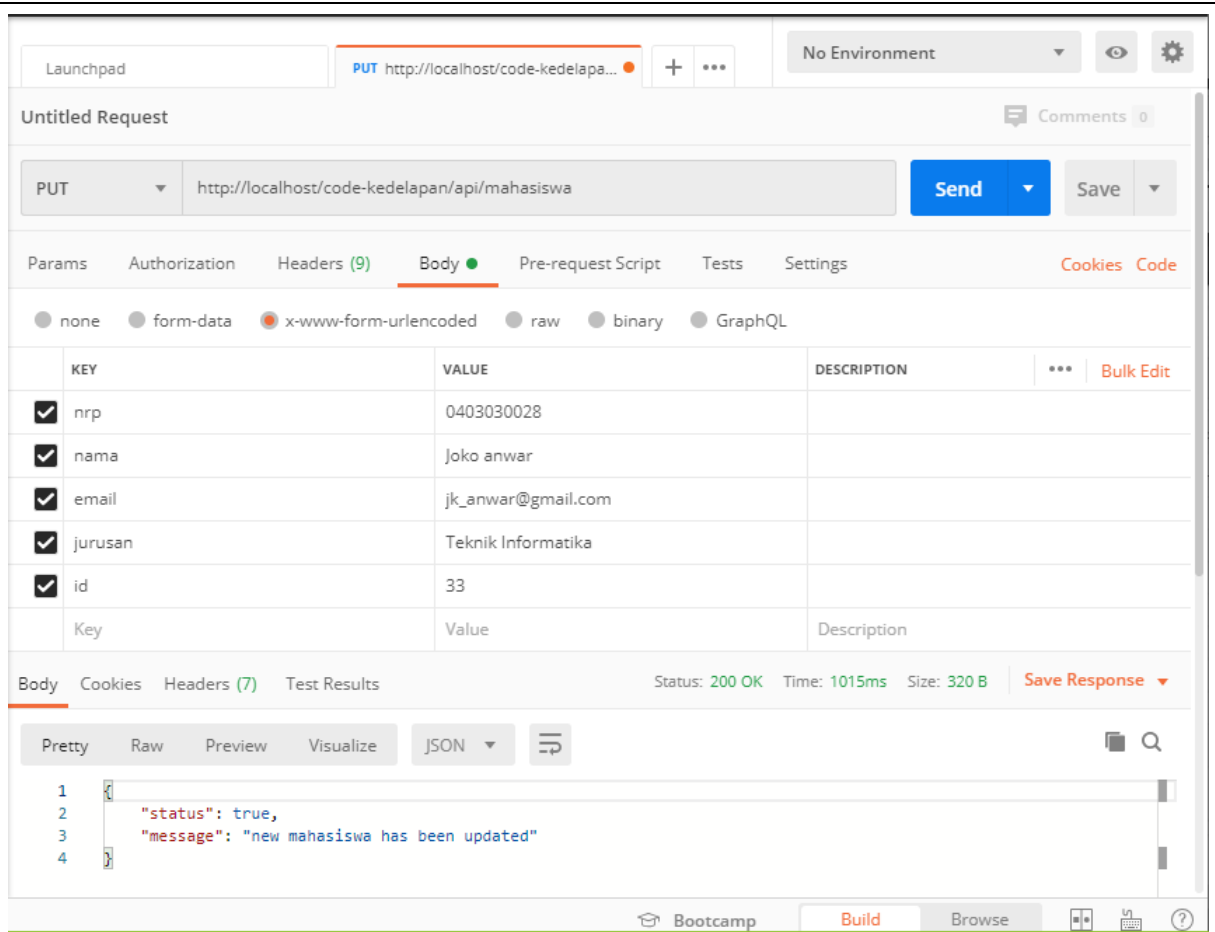
The screenshot shows the Postman interface with a POST request configured. The URL is `http://localhost/code-kedelapan/api/mahasiswa`. The request body is set to `x-www-form-urlencoded` and contains the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nrp	0403030028	
<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

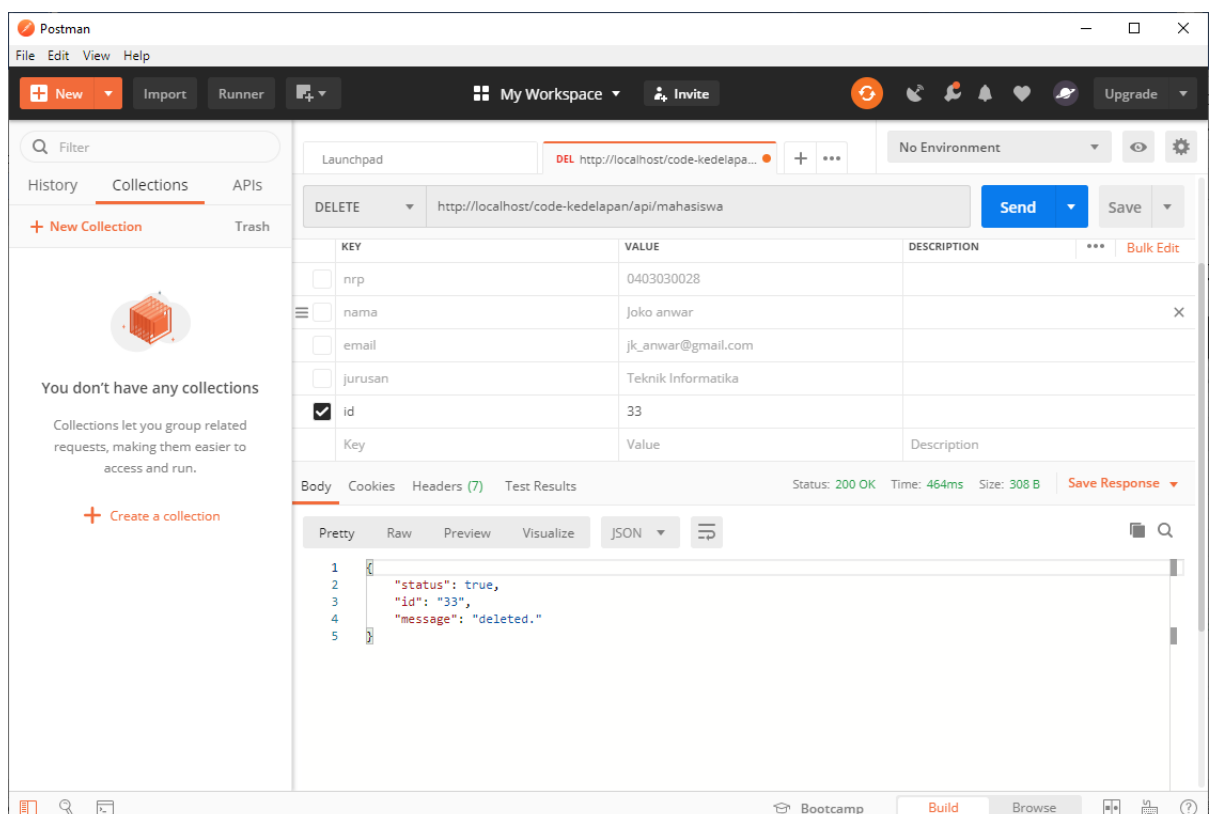
The response is shown in the bottom panel, indicating a successful status 201 Created with the message: `"status": true, "message": "new mahasiswa has been created"`.

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:



--	--

***-- Selamat Mengerjakan --***