

LAPORAN PRAKTIKUM  
PEMROGRAMAN WEB LANJUT

Jobsheet 8  
Implementasi Restful API Pemrograman Web Lanjut



NAMA : SILVI INDAH NOVITA SARI  
NIM : 1841720006  
KELAS : 2B – TI

D-IV TEKNIK INFORMATIKA  
TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2020



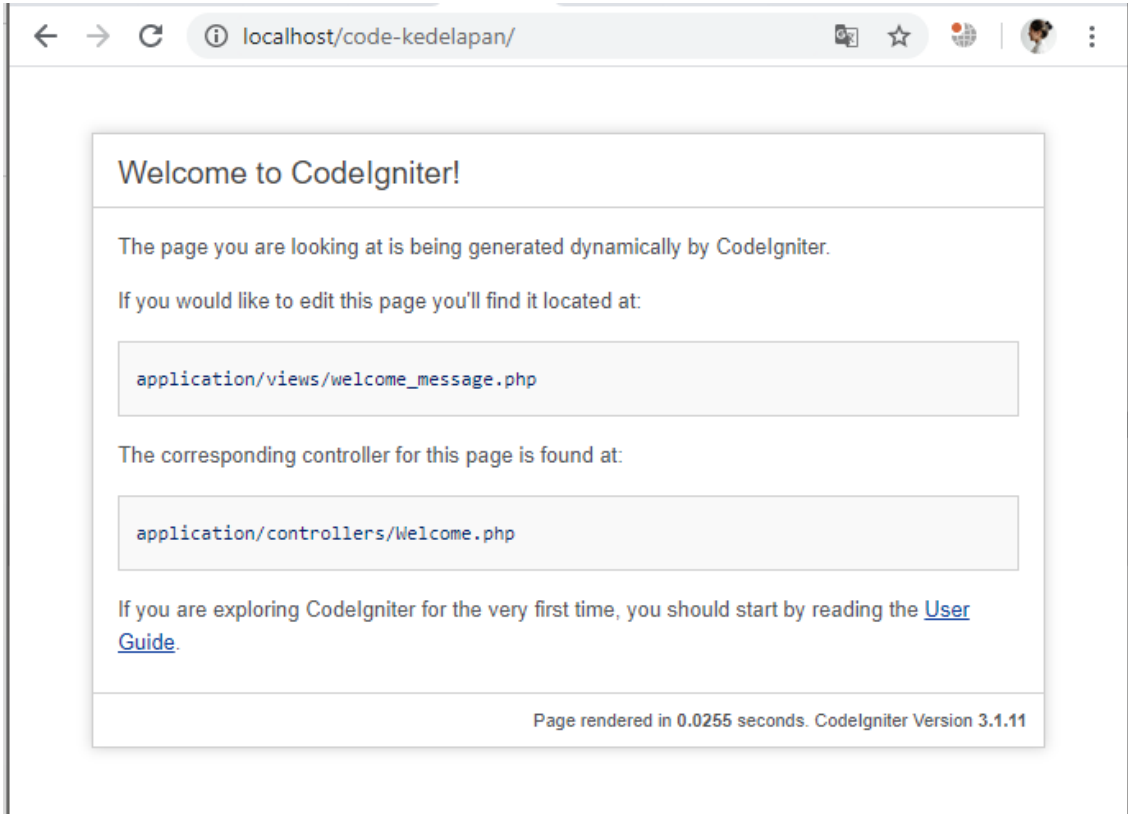
Jurusan Teknologi Informasi Politeknik Negeri Malang  
**Implementasi Restful API Pemrograman Web Lanjut**

**Mata Kuliah Pemrograman WEB Lanjut**

Pengampu: Tim Ajar Pemrograman Web Lanjut

23 Maret 2020

**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 :
3	<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> 

4

Selanjutnya lakukan konfigurasi dasar pada **config.php** seperti berikut:

```
24 |
25 */
26 $config['base_url'] = '';
27
28 /*
```

```
37 */
38 $config['index_page'] = 'index.php';
39
40 /*
41 |-----
```

Menjadi seperti berikut :

```
24 |
25 */
26 $config['base_url'] = 'http://localhost/rest_server/';
27
28 /*
29 |-----
```

```
36 |
37 */
38 $config['index_page'] = '';
39
40 /*
41 |-----
```

Dan file **autoload.php**

```
59 | $autoload['libraries'] = array('user_agent' => 'ua');
60 */
61 $autoload['libraries'] = array();
62
63 /*
64 |-----
```

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

Menjadi

```
59 | $autoload['libraries'] = array('user_agent' => 'ua');
60 */
61 $autoload['libraries'] = array('database');
62
63 /*
```

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

	<p>Dan database.php seperti berikut:</p> <pre> 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', </pre>
5	<p>Selanjutnya buatlah database <b>rest_server</b> pada phpmyadmin dan masukkan data dummy. Untuk data dummy dapat dibuat sebagai berikut:</p> <pre> 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; </pre>
6	<p>Selanjutnya extract file Library Rest Server, buka folder <b>application</b> dan duplikasi file dan folder tersebut kedalam folder <b>application</b> pada project <b>Code-kedelapan</b>. Klik <b>Replace</b> files jika ada notifikasi duplikat.</p>

<< ci-restserver-... > application

Search application

Name	Date modified
config	03/23/2020 8:31 AM
controllers	03/23/2020 8:31 AM
core	03/23/2020 8:31 AM
helpers	03/23/2020 8:31 AM
hooks	03/23/2020 8:31 AM
language	03/23/2020 8:31 AM
libraries	03/23/2020 8:31 AM
logs	03/23/2020 8:31 AM
models	03/23/2020 8:31 AM
tests	03/23/2020 8:31 AM
third_party	03/23/2020 8:31 AM
views	03/23/2020 8:31 AM
index	03/23/2020 8:31 AM

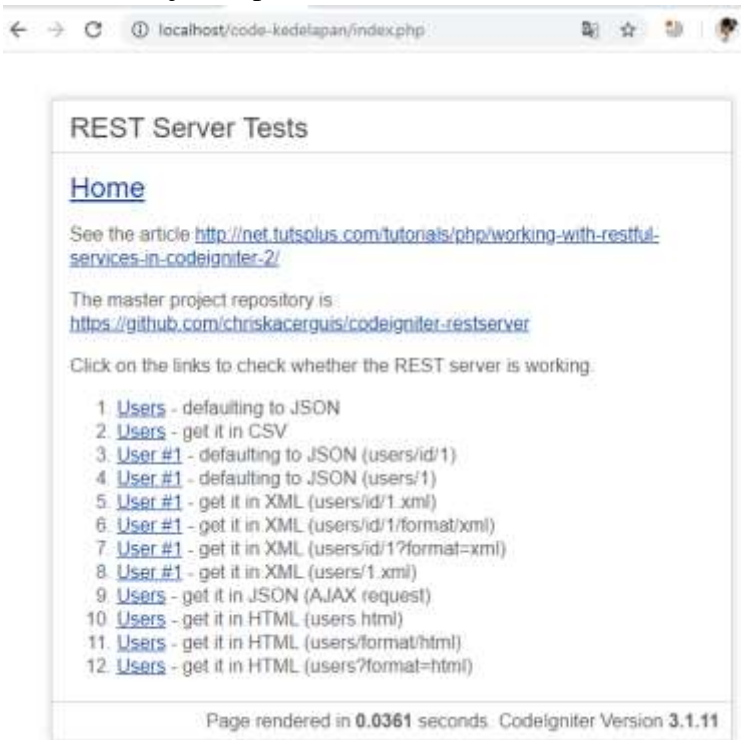
<< code-kedalap... > application

Search application

Name	Date modified
cache	03/24/2020 11:16 AM
config	03/24/2020 11:41 AM
controllers	03/24/2020 11:41 AM
core	03/24/2020 11:16 AM
helpers	03/24/2020 11:16 AM
hooks	03/24/2020 11:16 AM
language	03/24/2020 11:41 AM
libraries	03/24/2020 11:41 AM
logs	03/24/2020 11:17 AM
models	03/24/2020 11:16 AM
third_party	03/24/2020 11:16 AM
views	03/24/2020 11:41 AM
.htaccess	09/19/2019 5:08 AM

	<div> <div>code-kedelap... &gt; application</div> <div>Search application</div> <table> <tr> <th>Name</th><th>Date modified</th></tr> <tr> <td>cache</td><td>03/24/2020 11:16 AM</td></tr> <tr> <td>config</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>controllers</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>core</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>helpers</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>hooks</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>language</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>libraries</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>logs</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>models</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>tests</td><td>03/24/2020 11:42 AM</td></tr> <tr> <td>third_party</td><td>03/24/2020 11:43 AM</td></tr> <tr> <td>views</td><td>03/24/2020 11:43 AM</td></tr> <tr> <td>.htaccess</td><td>09/19/2019 5:08 AM</td></tr> <tr> <td>index</td><td>03/23/2020 8:31 AM</td></tr> </table> </div>	Name	Date modified	cache	03/24/2020 11:16 AM	config	03/24/2020 11:42 AM	controllers	03/24/2020 11:42 AM	core	03/24/2020 11:42 AM	helpers	03/24/2020 11:42 AM	hooks	03/24/2020 11:42 AM	language	03/24/2020 11:42 AM	libraries	03/24/2020 11:42 AM	logs	03/24/2020 11:42 AM	models	03/24/2020 11:42 AM	tests	03/24/2020 11:42 AM	third_party	03/24/2020 11:43 AM	views	03/24/2020 11:43 AM	.htaccess	09/19/2019 5:08 AM	index	03/23/2020 8:31 AM
Name	Date modified																																
cache	03/24/2020 11:16 AM																																
config	03/24/2020 11:42 AM																																
controllers	03/24/2020 11:42 AM																																
core	03/24/2020 11:42 AM																																
helpers	03/24/2020 11:42 AM																																
hooks	03/24/2020 11:42 AM																																
language	03/24/2020 11:42 AM																																
libraries	03/24/2020 11:42 AM																																
logs	03/24/2020 11:42 AM																																
models	03/24/2020 11:42 AM																																
tests	03/24/2020 11:42 AM																																
third_party	03/24/2020 11:43 AM																																
views	03/24/2020 11:43 AM																																
.htaccess	09/19/2019 5:08 AM																																
index	03/23/2020 8:31 AM																																

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	<p>Buatlah file <b>Mahasiswa.php</b> pada <b>controller/api</b>. Selanjutnya tuliskan kode berikut:</p> <pre>application &gt; controllers &gt; api &gt; Mahasiswa.php &gt; PHP Intelephense &gt; Mahasiswa  1  &lt;?php 2 3  use Restserver\Libraries\REST_Controller; 4 5  defined('BASEPATH') or exit('No direct script access allowed'); 6 7  require APPPATH . 'libraries/REST_Controller.php'; 8  require APPPATH . 'libraries/Format.php'; 9 10 class Mahasiswa extends REST_Controller 11 { 12 13     public function __construct() 14     { 15         parent::__construct(); 16         \$this-&gt;load-&gt;model('Mahasiswa_model', 'mahasiswa'); 17     } 18     public function index_get() 19     { 20         \$id = \$this-&gt;get('id'); 21         if (\$id === null) { 22             \$mahasiswa = \$this-&gt;mahasiswa-&gt;getMahasiswa(); 23         } else { 24             \$mahasiswa = \$this-&gt;mahasiswa-&gt;getMahasiswa(\$id); 25         } 26         if (\$mahasiswa) { 27             \$this-&gt;response([ 28                 'status' =&gt; true, 29                 'data' =&gt; \$mahasiswa 30             ], REST_Controller::HTTP_OK); 31         } else { 32             \$this-&gt;response([ 33                 'status' =&gt; false, 34                 'message' =&gt; 'id not found' 35             ], REST_Controller::HTTP_NOT_FOUND); 36         } 37     } 38 } 39</pre>
2	<p>Selanjutnya Buatlah file <b>Mahasiswa_model.php</b> pada model. Selanjutnya tuliskan kode berikut:</p>

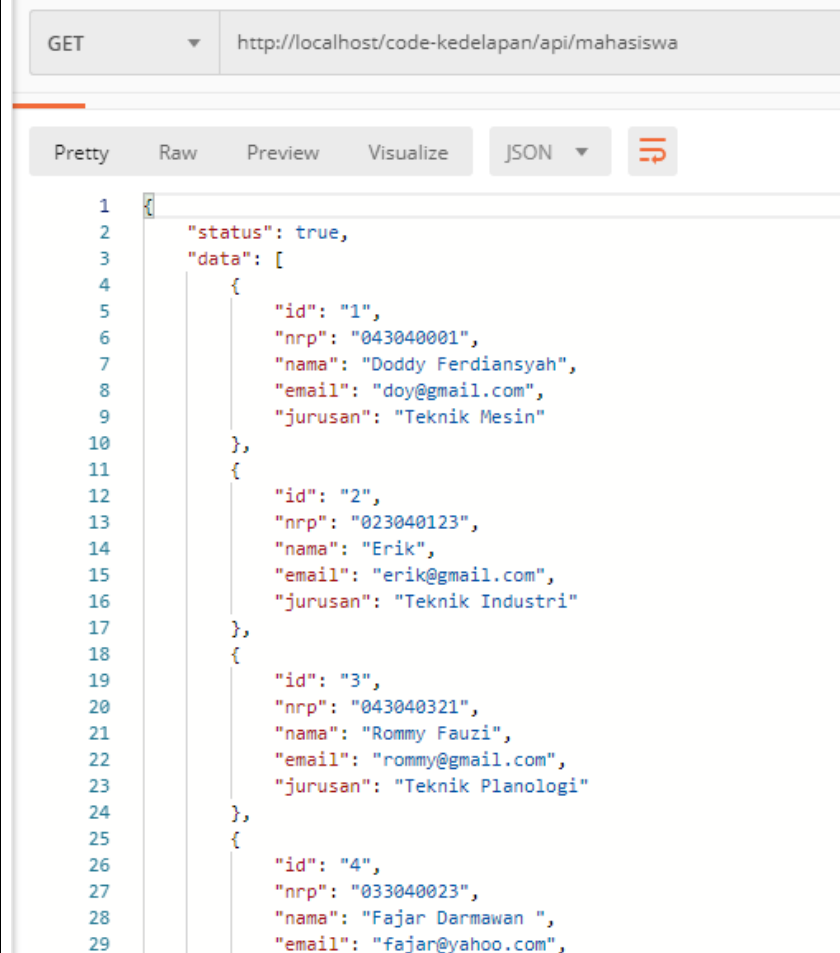
```

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

```

3

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



The screenshot shows a Postman interface with a GET request to `http://localhost/code-kedelapan/api/mahasiswa`. The response is in JSON format, displaying a list of four students with their details.

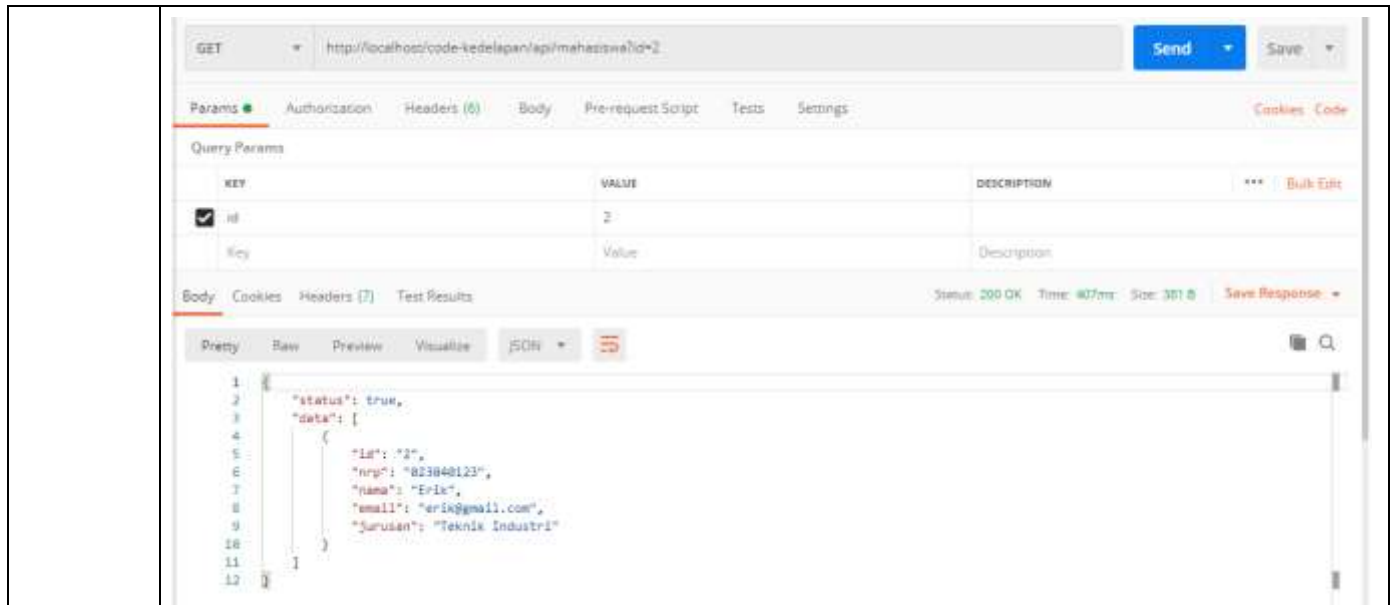
```

1  {
2      "status": true,
3      "data": [
4          {
5              "id": "1",
6              "nrp": "043040001",
7              "nama": "Doddy Ferdiansyah",
8              "email": "doy@gmail.com",
9              "jurusan": "Teknik Mesin"
10         },
11         {
12             "id": "2",
13             "nrp": "023040123",
14             "nama": "Erik",
15             "email": "erik@gmail.com",
16             "jurusan": "Teknik Industri"
17         },
18         {
19             "id": "3",
20             "nrp": "043040321",
21             "nama": "Rommy Fauzi",
22             "email": "rommy@gmail.com",
23             "jurusan": "Teknik Planologi"
24         },
25         {
26             "id": "4",
27             "nrp": "033040023",
28             "nama": "Fajar Darmawan ",
29             "email": "fajar@yahoo.com",

```

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**:

```

application > controllers > api > Mahasiswa.php > PHP IntelliSense > Mahasiswa >
35     }
36   }
37
38   // DELETE
39   function index_delete() {
40       $id = $this->delete('id');
41       if ($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   }
61

```

application > controllers > api > Mahasiswa.php > PHP IntelliSense > Mahasiswa >

```
62 // POST
63 public function index_post(){
64     $data = [
65         'nrp' => $this->post('nrp'),
66         'nama' => $this->post('nama'),
67         'email' => $this->post('email'),
68         'jurusan' => $this->post('jurusan'),
69     ];
70
71     if ($this->mahasiswa->createMahasiswa($data) > 0) {
72         $this->response([
73             'status' => true,
74             'message' => 'new mahasiswa has been created'
75         ], REST_Controller::HTTP_CREATED);
76     } else {
77         $this->response([
78             'status' => false,
79             'message' => 'failed to create new data'
80         ], REST_Controller::HTTP_BAD_REQUEST);
81     }
82 }
83
```

application > controllers > api > Mahasiswa.php > PHP IntelliSense > Mahasiswa >

```
84 // PUT
85 public function index_put(){
86     $id = $this->put('id');
87     $data = [
88         'nrp' => $this->put('nrp'),
89         'nama' => $this->put('nama'),
90         'email' => $this->put('email'),
91         'jurusan' => $this->put('jurusan'),
92     ];
93     if ($this->mahasiswa->updateMahasiswa($data, $id) > 0) {
94         $this->response([
95             'status' => true,
96             'message' => 'new mahasiswa has been updated'
97         ], REST_Controller::HTTP_OK);
98     } else {
99         $this->response([
100             'status' => false,
101             'message' => 'failed to update data'
102         ], REST_Controller::HTTP_BAD_REQUEST);
103     }
104 }
105 }
106 >>
107
```

5

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

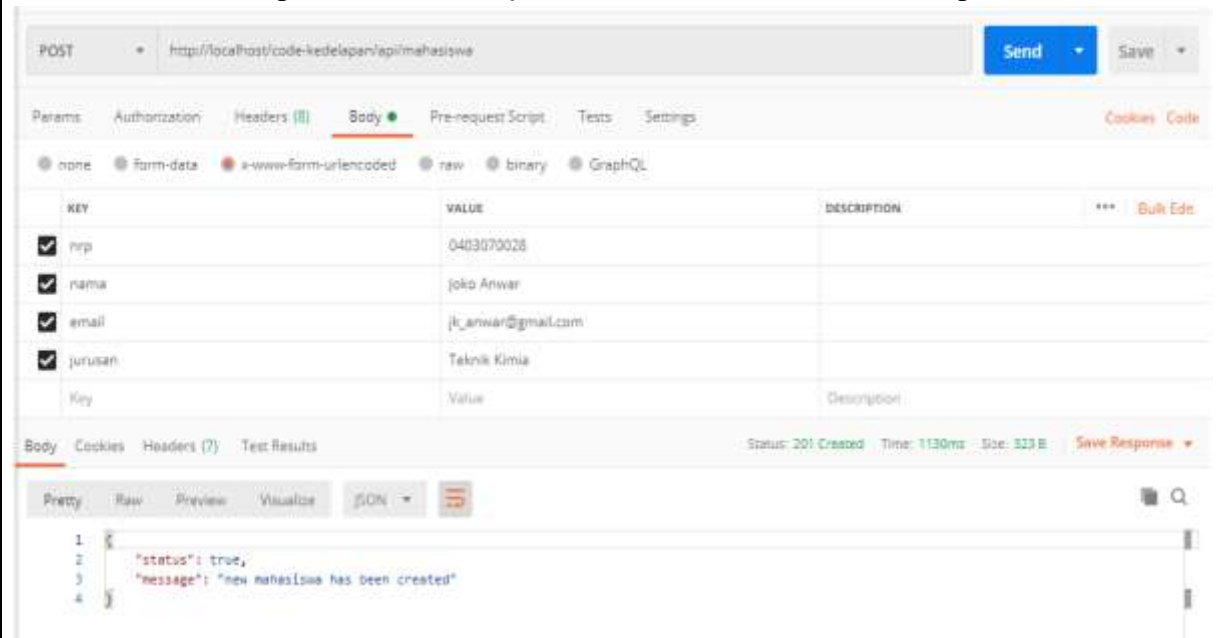
```

application > models > Mahasiswa_model.php > PHP IntelliSense > Mah
13
14
15 public function deleteMahasiswa($id){
16     $this->db->delete('mahasiswa', ['id' => $id]);
17     return $this->db->affected_rows();
18
19 }
20
21 public function createMahasiswa($data){
22     $this->db->insert('mahasiswa', $data);
23     return $this->db->affected_rows();
24
25 }
26
27 public function updateMahasiswa($data, $id){
28     $this->db->update('mahasiswa', $data, ['id' => $id]);
29     return $this->db->affected_rows();
30
31 }
32
33
34
35 /* End of file Mahasiswa_model.php */
36
37 ?>

```

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:

The image shows a Postman interface for a PUT request. The URL is `http://localhost/code-kedelapan/api/mahasiswa`. The request body is configured as `x-www-form-urlencoded`. The body contains the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> mp	0403070038	
<input checked="" type="checkbox"/> nama	Jaka Anwar	
<input checked="" type="checkbox"/> email	jk_anwar@gmail.com	
<input checked="" type="checkbox"/> jurusan	Teknik Kimia	
<input checked="" type="checkbox"/> id	39	

At the bottom, the status is `400 Bad Request`, Time is `120ms`, and Size is `282 B`. There is a `Save Response` button.

8

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

The image shows a Postman interface for a DELETE request. The URL is `http://localhost/code-kedelapan/api/mahasiswa`. The request body is configured as `x-www-form-urlencoded`. The body contains the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	39	

At the bottom, the status is `400 Bad Request`, Time is `120ms`, and Size is `274 B`. There is a `Save Response` button.