

Pemrograman Web Lanjut

Praktikum Jobsheet 8



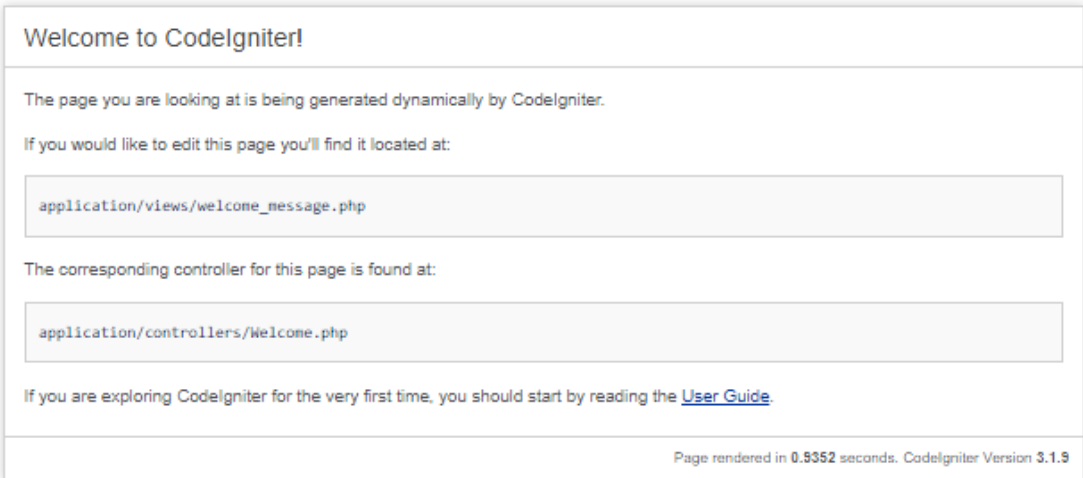
Osa Mahanani S.

TI-2B

1841720066

PROGRAM STUDI TEKNIK INFORMATIKA
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 :
3	<p>Extract codeigniter yang sudah di unduh dan letakkan pada folder httdocs. 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> <pre>37 */ 38 \$config['index_page'] = ''; 39 40 /*</pre> <p>Dan file autoload.php</p>

```

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

```

```

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

```

Menjadi

```

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

```

```

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

```

Dan database.php seperti berikut:

```

76  $db['default'] = array(
77      'dsn' => '',
78      'hostname' => 'localhost',
79      'username' => 'root',
80      'password' => 'mysql',
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');

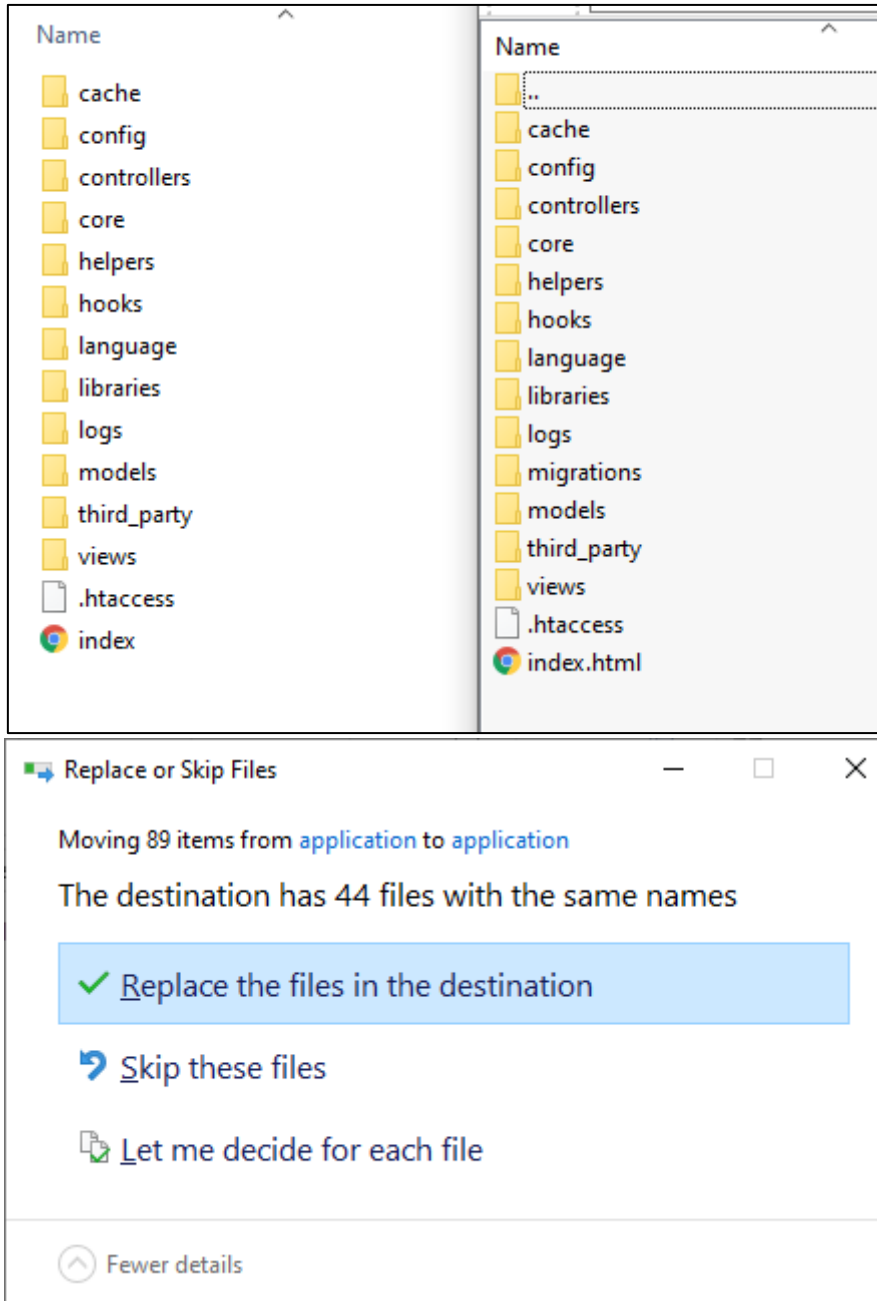
ALTER TABLE `mahasiswa`
  ADD PRIMARY KEY (`id`);

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

```

6

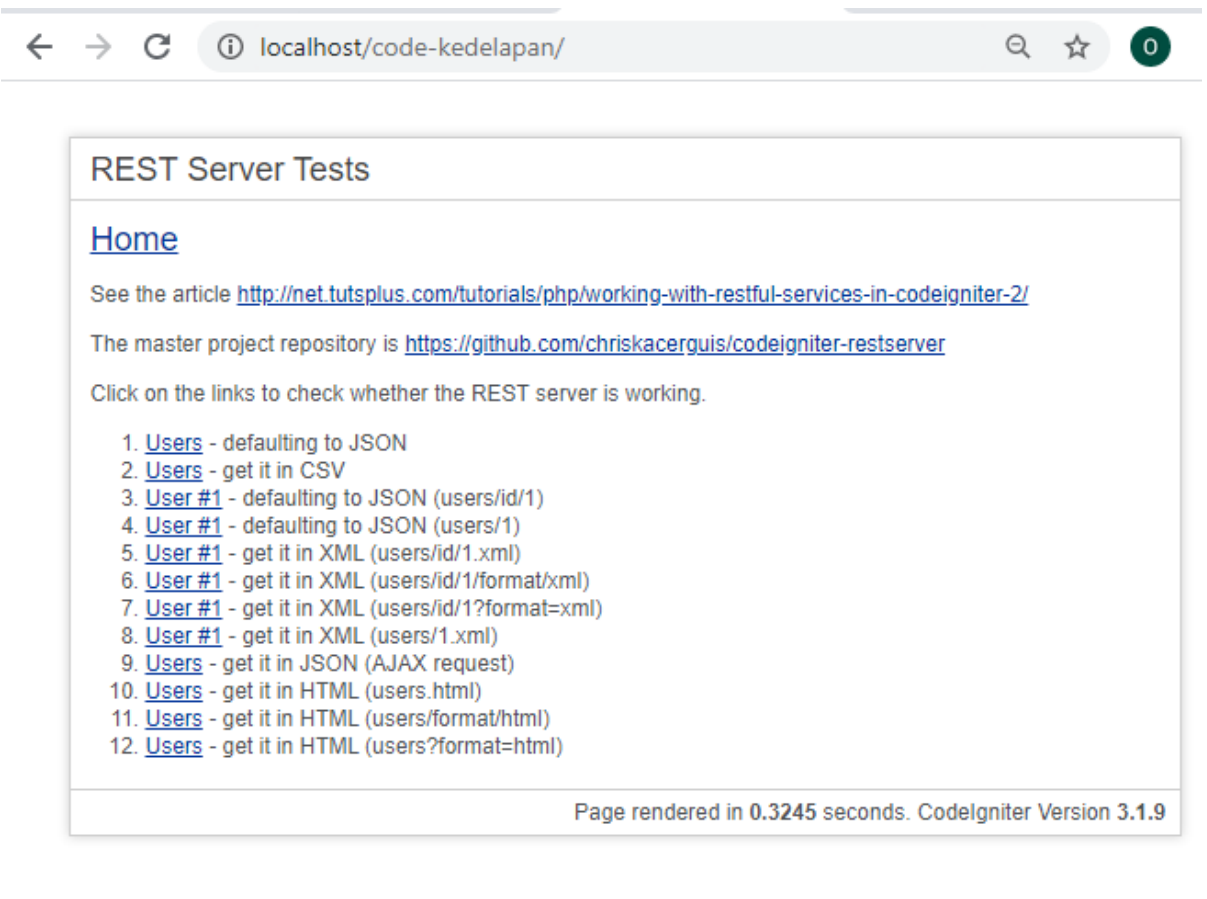
Selanjutnya extract file Library Rest Server, buka folder **application** dan duplikasi file dan folder tersebut kedalam folder **application** pada project **Code-kedelapan**. Klik **Replace** files jika ada notifikasi duplikat.



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:

	
8	Tahap instalasi Library Rest Server selesai

Praktikum – Bagian 2: Membuat Aplikasi CRUD Restful Codeigniter

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

```

application > controllers > api > 🐞 Mahasiswa.php
1  <?
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 }
39
40 /* End of file Mahasiswa.php */
41 ?>

```

2

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

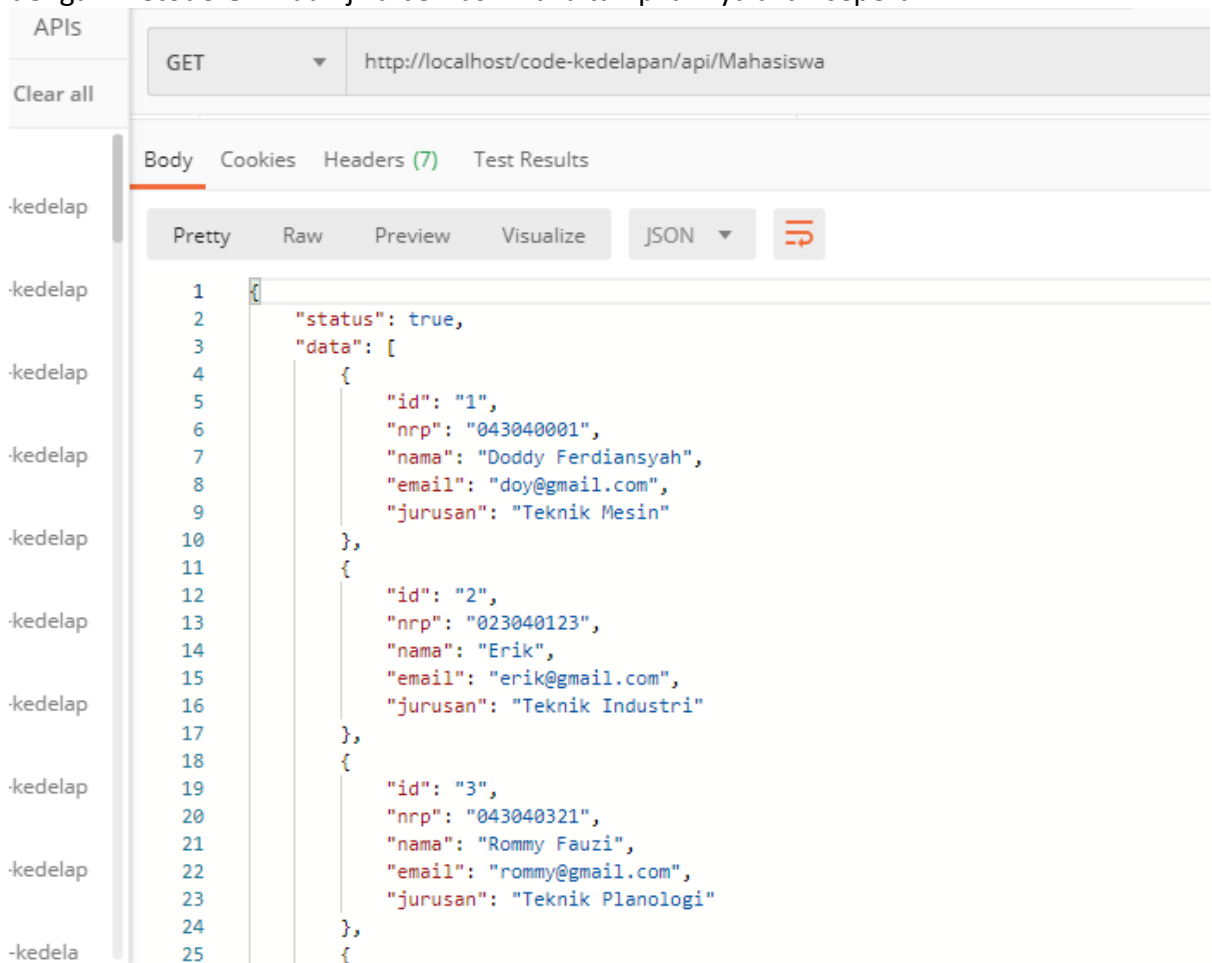
```

application > models > Mahasiswa_model.php
1  <?
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 }
16 ?>

```

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 the Postman interface with the following details:

- Method:** GET
- URL:** http://localhost/code-kedelapan/api/Mahasiswa
- Response Format:** JSON
- Response Body:**

```

{
  "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"
    }
  ]
}

```

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

KEY	VALUE
<input checked="" type="checkbox"/> id	2
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```

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

```

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

```

41 // DELETE
42 function index_delete()
43 {
44     $id = $this->delete('id');
45     if ($id === null) {
46         $this->response([
47             'status' => false,
48             'message' => 'provide an id!'
49         ], REST_Controller::HTTP_BAD_REQUEST);
50     } else {
51         if ($this->mahasiswa->deleteMahasiswa($id) > 0) {
52             //ok
53             $this->response([
54                 'status' => true,
55                 'id' => $id,
56                 'message' => 'deleted.'
57             ], REST_Controller::HTTP_OK);
58         } else {
59             //id not found
60             $this->response([
61                 'status' => false,
62                 'message' => 'provide an id!'
63             ], REST_Controller::HTTP_BAD_REQUEST);
64         }
65     }
66 }

```



```

67 // POST
68 public function index_post()
69 {
70     $data = [
71         'nrp' => $this->post('nrp'),
72         'nama' => $this->post('nama'),
73         'email' => $this->post('email'),
74         'jurusan' => $this->post('jurusan'),
75     ];
76
77     if ($this->mahasiswa->createMahasiswa($data) > 0) {
78         $this->response([
79             'status' => true,
80             'message' => 'new mahasiswa has been created'
81         ], REST_Controller::HTTP_CREATED);
82     } else {
83         //id not found
84         $this->response([
85             'status' => false,
86             'message' => 'failed to create new data!'
87         ], REST_Controller::HTTP_BAD_REQUEST);
88     }
89 }
90
91 // PUT
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' => $this->put('jurusan'),
100     ];
101
102     if ($this->mahasiswa->updateMahasiswa($data, $id) > 0) {
103         $this->response([
104             'status' => true,
105             'message' => 'data mahasiswa has been updated'
106         ], REST_Controller::HTTP_OK);
107     } else {
108         //id not found
109         $this->response([
110             'status' => false,
111             'message' => 'failed to update data!'
112         ], REST_Controller::HTTP_BAD_REQUEST);
113     }
114 }
115
116 }
117
118

```

5

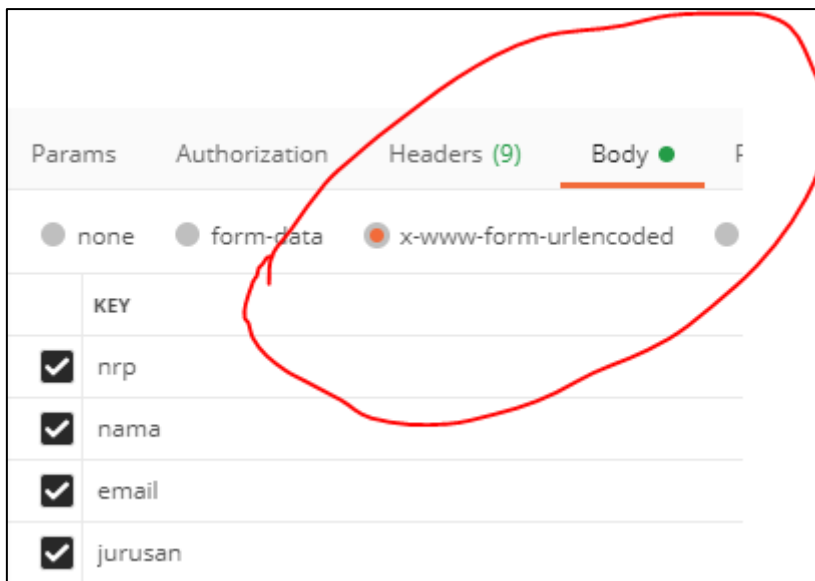
Dan lanjutkan kode pada file **Mahasiswa_model.php**:

```

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

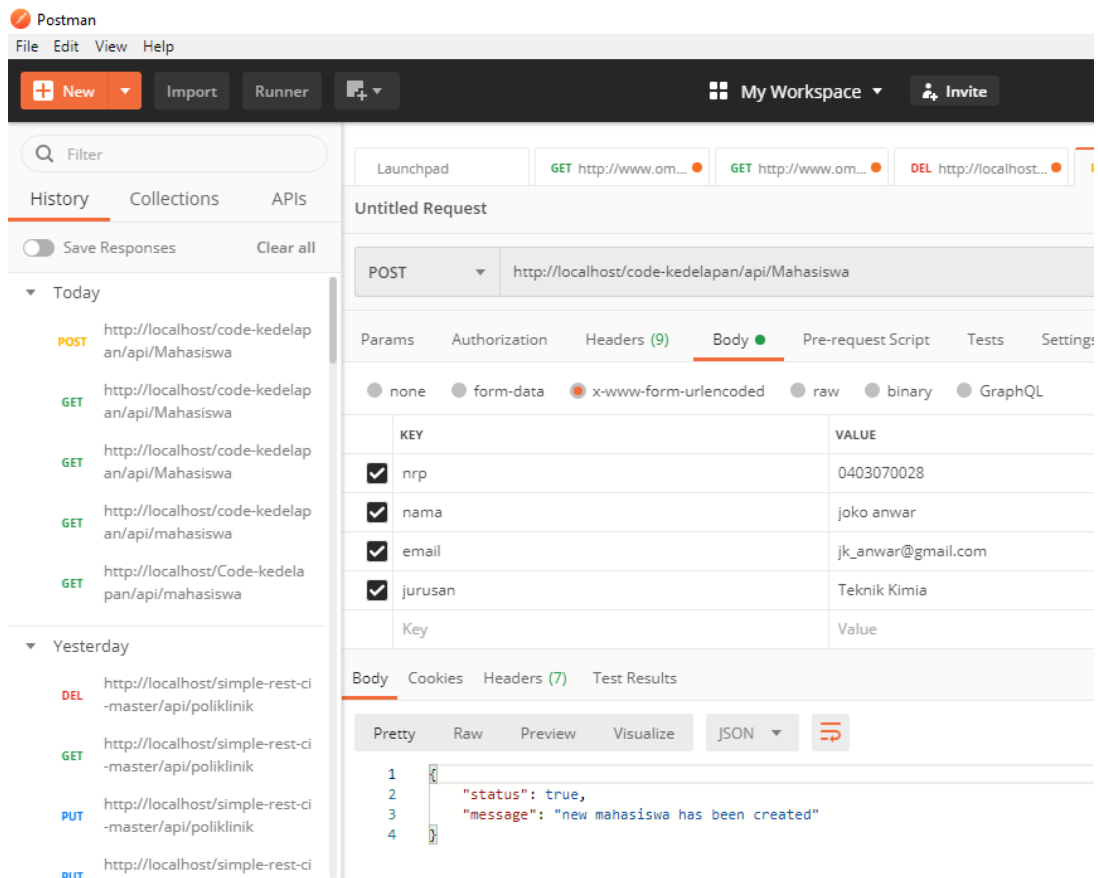
```

- 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:



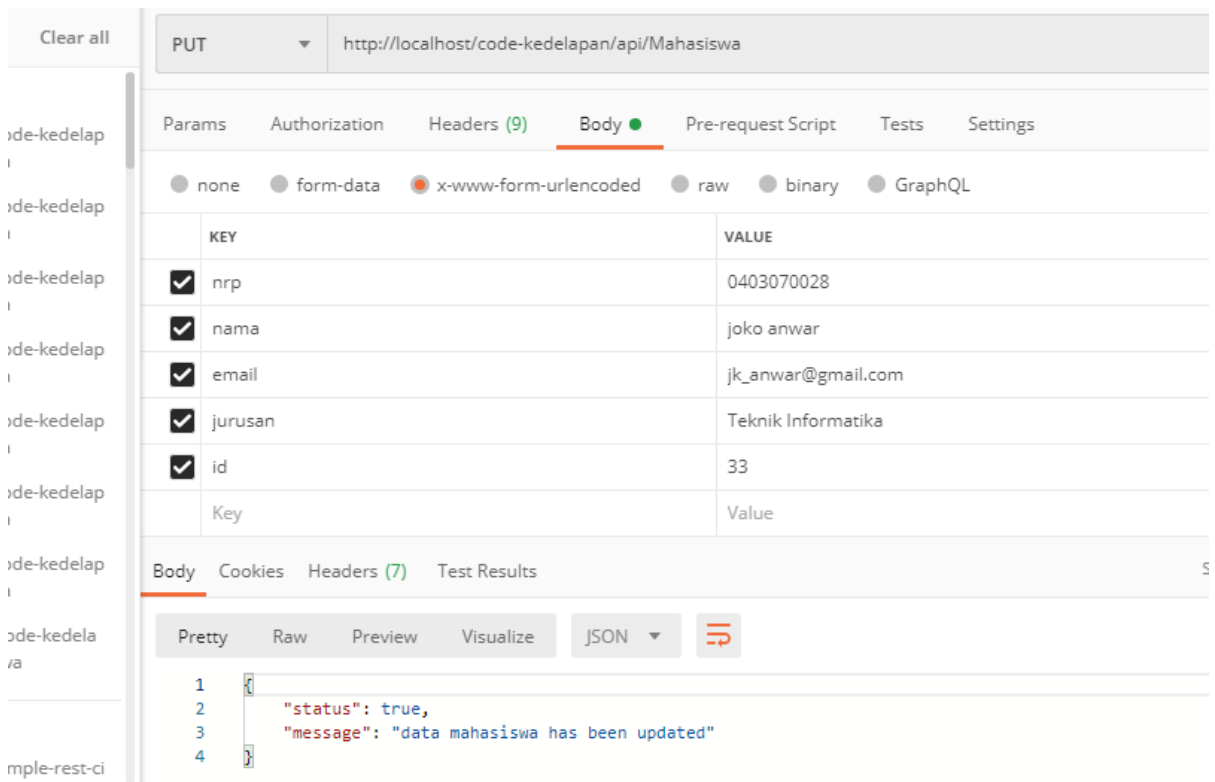
The screenshot shows the Postman interface with the 'Body' tab selected. The encoding is set to 'x-www-form-urlencoded'. A red circle highlights the encoding selection area. Below the encoding selection, there is a table with the following data:

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



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:

ions

APIs

Clear all

10st/code-kedela
asiswa

10st/code-kedela
asiswa

10st/code-kedela
asiswa

10st/code-kedela
asiswa

10st/code-kedela
asiswa

10st/code-kedela
asiswa

10st/code-kedela
asiswa

10st/code-kedela
asiswa

10st/Code-kedela
asiswa

DELETE

http://localhost/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
<input checked="" type="checkbox"/>	id	33
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
	Key	Value

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

{

"status": true,

"id": "33",

"message": "deleted."

}