

LAPORAN PEMROGRAMAN WEB LANJUT

JOBSHEET 8 IMPLEMENTASI RESTFUL API

Oleh:

KINANTI PERMATA PUTRI NIM. 1841720022

TI-2A



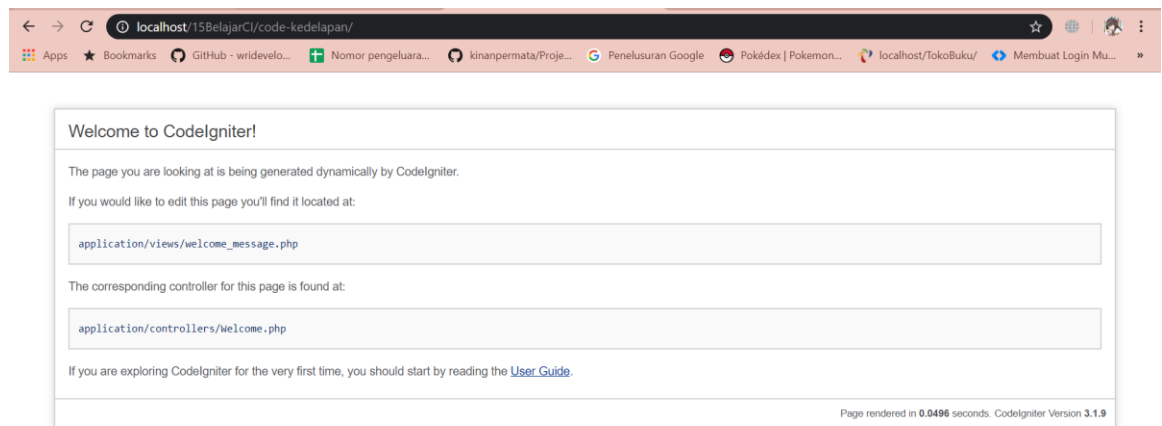
**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
MARET 2020**

1. Praktikum – Bagian 1: Instalasi library restful codeigniter

- 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) 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:



-
- 4) Selanjutnya lakukan konfigurasi dasar pada config.php seperti berikut:

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

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

Menjadi seperti berikut:

```
25  */
26  $config['base_url'] = 'http://localhost/15BelajarCI/rest_server/';
27
```

```
37  */
38  $config['index_page'] = '';
39
```

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;
```

Server: localhost » Database: rest_server

Structure SQL Search Query Export Import Operations Privileges Routines More

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0557 seconds.)

```
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
```

[Edit inline] [Edit] [Create PHP code]

Server: localhost » Database: rest_server

Structure SQL Search Query Export Import Operations Privileges Routines More

Show query box

✓ 5 rows inserted. (Query took 0.0196 seconds.)

```
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')
```

[Edit inline] [Edit] [Create PHP code]

Server: localhost » Database: rest_server

Structure SQL Search Query Export Import Operations Privileges Routines Events More

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1831 seconds.)

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

[Edit inline] [Edit] [Create PHP code]

Server: localhost » Database: rest_server

Structure SQL Search Query Export Import Operations Privileges Routines More

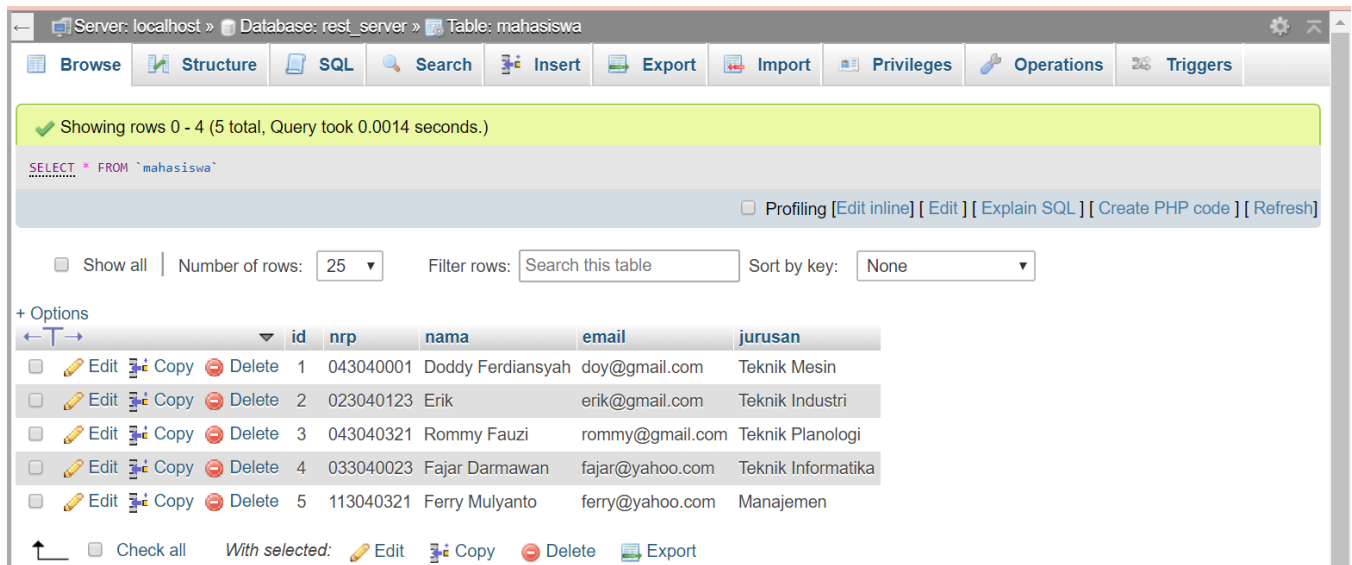
Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0987 seconds.)

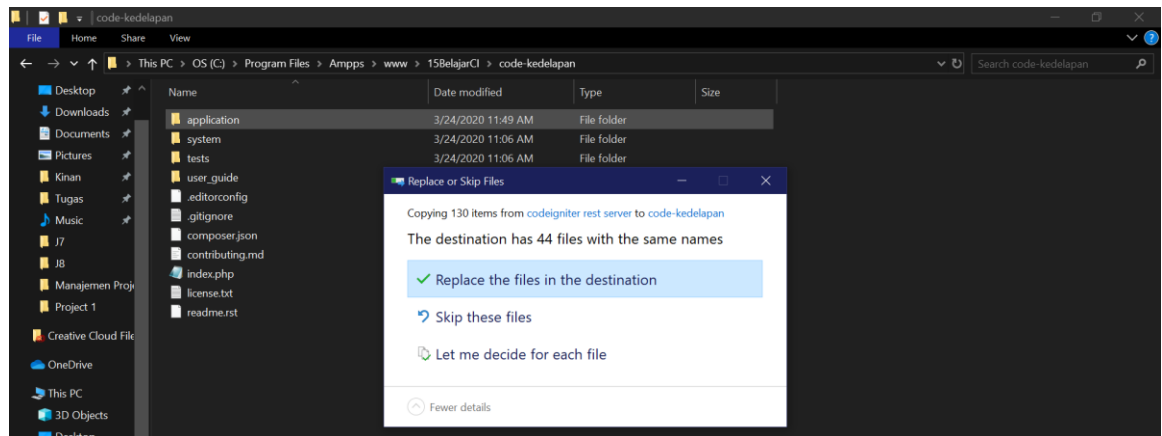
```
ALTER TABLE `mahasiswa` MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=33
```

[Edit inline] [Edit] [Create PHP code]

⚠ Warning: #1681 Integer display width is deprecated and will be removed in a future release.

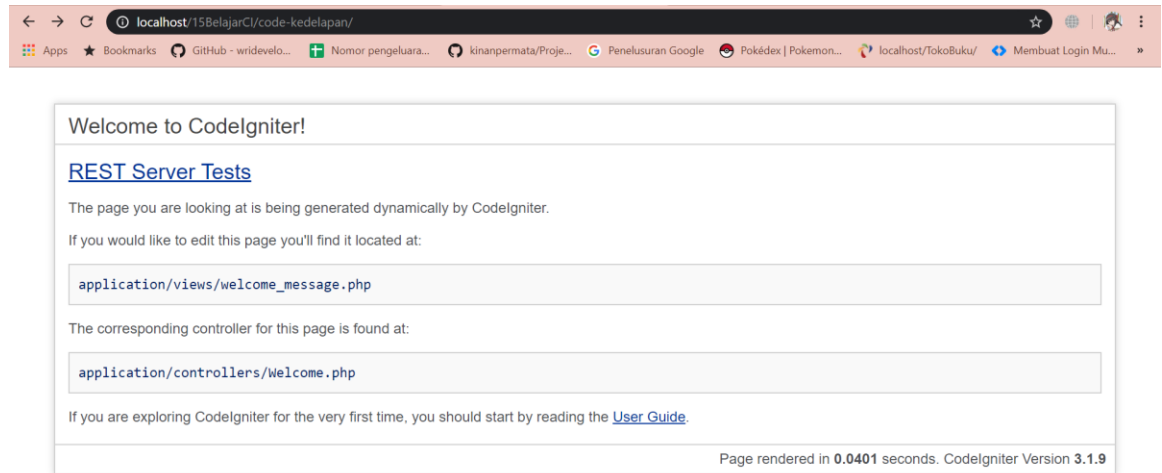


- 6) Selanjutnya extract file Library Rest Server, buka folder application dan duplikasi file dan folder tersebut kedalam folder application pada project Code-kedelapan. Klik Replace files jika ada notifikasi duplikat.



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

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

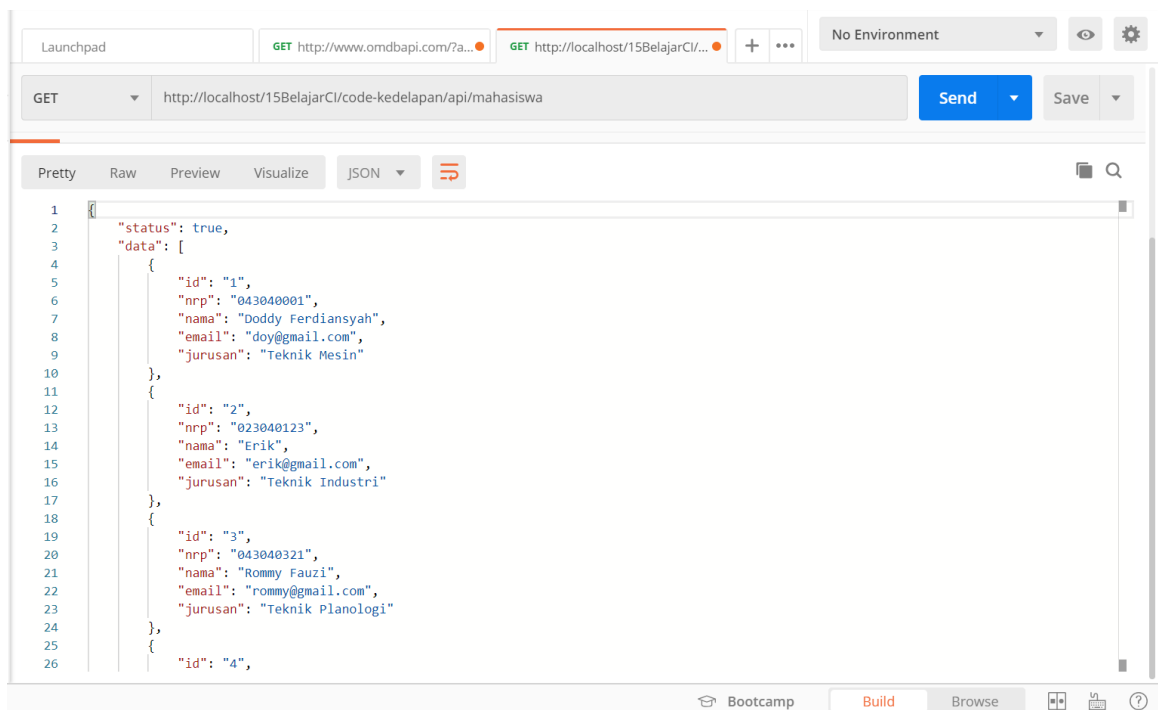
1) Buatlah file Mahasiswa.php pada controller/api. Selanjutnya tuliskan kode berikut:

```
application > controllers > api > Mahasiswa.php > PHP Intelephense > Mahasiswa
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
11     public function __construct()
12     {
13         parent::__construct();
14         $this->load->model('Mahasiswa_model', 'mahasiswa');
15     }
16
17
18     public function index_get()
19     {
20         $id = $this->get('id');
21         if ($id === null) {
22             $mahasiswa = $this->mahasiswa->getMahasiswa();
23         } else {
24             $mahasiswa = $this->mahasiswa->getMahasiswa($id);
25         }
26
27         if($mahasiswa) {
28             $this->response([
29                 'status' => true,
30                 'data' => $mahasiswa
31             ], REST_Controller::HTTP_OK);
32         } else {
33             $this->response([
34                 'status' => false,
35                 'message' => 'id not found'
36             ], REST_Controller::HTTP_NOT_FOUND);
37         }
38     }
39 }
40
41
42 /* End of file Controllername.php */
43
44
45 >>
```

- 2) Selanjutnya Buatlah file Mahasiswa_model.php pada model. Selanjutnya tuliskan kode berikut:

```
application > models > 🐞 Mahasiswa_model.php > ...
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class Mahasiswa_model extends CI_Model {
6
7      public function getMahasiswa($id = null)
8      {
9          if ($id === null){
10
11              return $this->db->get('mahasiswa')->result_array();
12          } else {
13              return $this->db->get_where('mahasiswa', ['id'=>$id])->result_array();
14          }
15      }
16
17  }
18
19  /* End of file ModelName.php */
20
21
22  <?>
```

- 3) Buka aplikasi postman, masukkan alamat <http://localhost/15BelajarCI/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:

The screenshot shows a REST client interface with the following details:

- Request:** GET `http://localhost/15BelajarCI/code-kedelapan/api/mahasiswa?id=2`
- Params:** A table with one entry:

KEY	VALUE	DESCRIPTION
id	2	
- Response:** Status: 200 OK, Time: 96ms, Size: 383 B. The response body is a JSON object:

```
{  "status": true,  "data": [    {      "id": "2",      "nnp": "023040123",      "nama": "Erik",      "email": "erik@gmail.com",      "jurusan": "Teknik Industri"    }  ]}
```

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

```
41 public function index_delete()
42 {
43     $id = $this->delete('id');
44
45     if ($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' => 'id not found'
63             ], REST_Controller::HTTP_BAD_REQUEST);
64         }
65     }
66 }
```

```

68 public function index_post(){
69     $data = [
70         'nrp' => $this->post('nrp'),
71         'nama' => $this->post('nama'),
72         'email' => $this->post('email'),
73         'jurusan' => $this->post('jurusan'),
74     ];
75
76     if ($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' => 'failed to create new data'
86         ], REST_Controller::HTTP_BAD_REQUEST);
87     }
88 }

```

```

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

```

5) Dan lanjutkan kode pada file Mahasiswa_model.php:

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

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:

Create

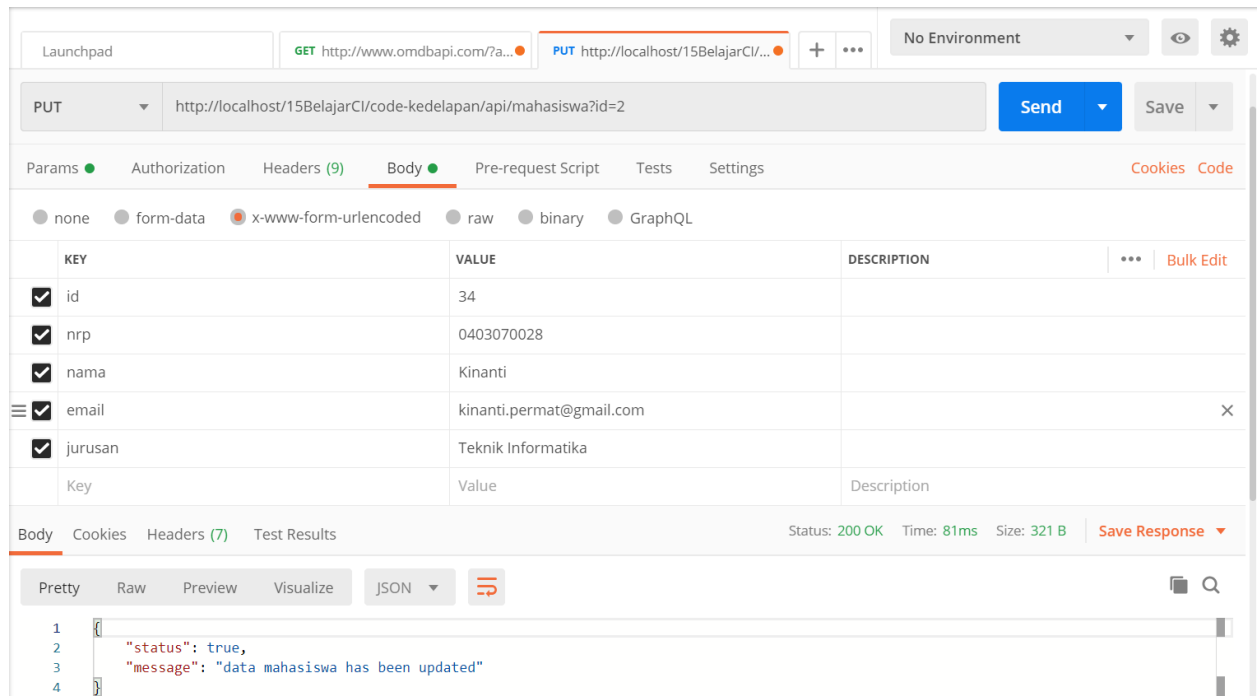
The screenshot shows the Postman interface for a POST request. The URL is `http://localhost/15BelajarCI/code-kedelapan/api/mahasiswa?id=2`. The request body is set to `x-www-form-urlencoded` and contains the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nrp	0403070028	
<input checked="" type="checkbox"/> nama	Kinanti Permata	
<input checked="" type="checkbox"/> email	kinantipermata@gmail.com	
<input checked="" type="checkbox"/> jurusan	Teknik Informatika	
Key	Value	Description

The response status is `201 Created` with a time of `80ms` and size of `325 B`. The response body is JSON:

```
{
  "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:

