

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT**

Jobsheet - 8

Implementasi Restful API



Oleh:

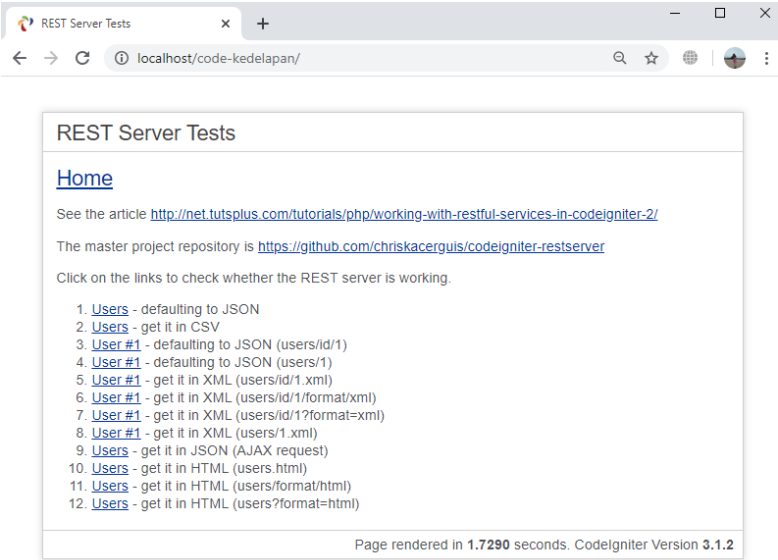
Farradila Ayu Damayanti

1841720074

TI-2B

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020**

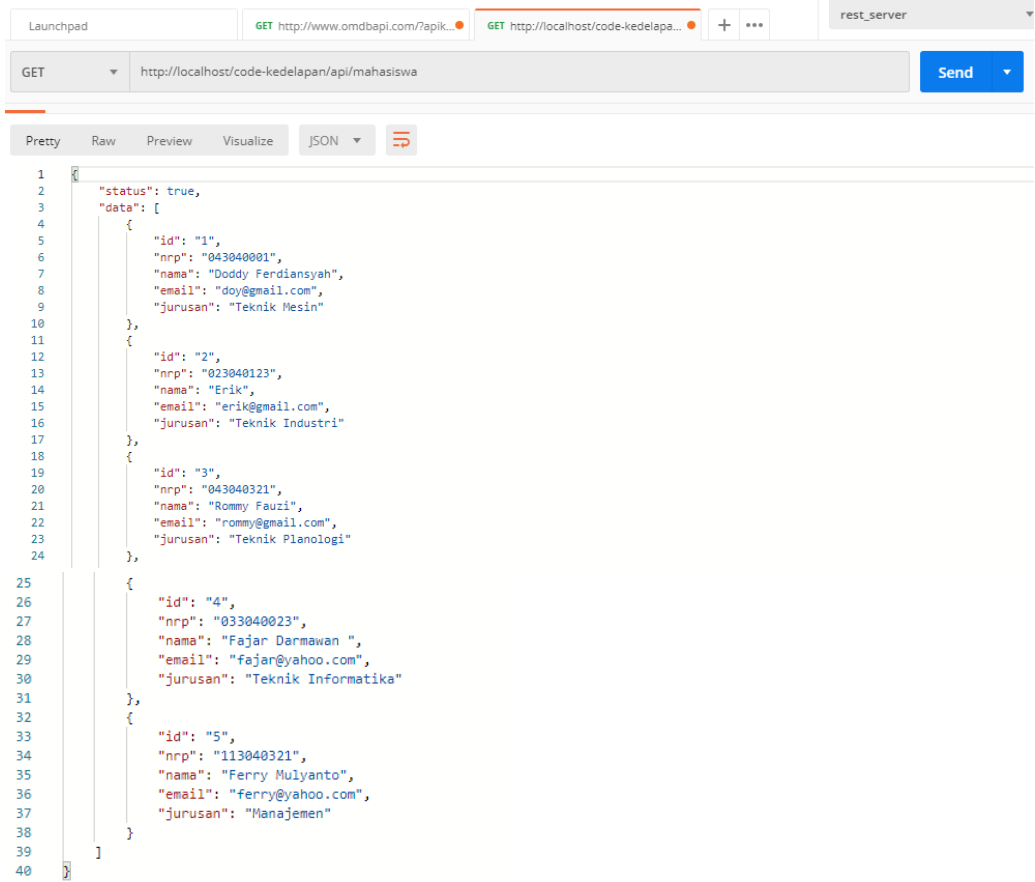
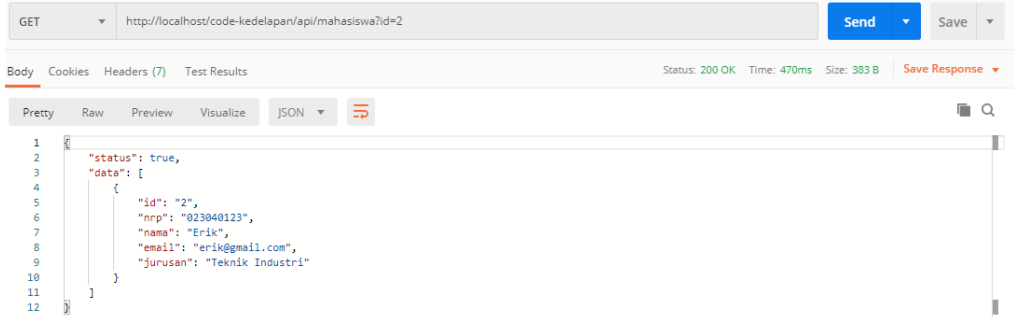
Praktikum – Bagian 1: Instalasi library restful codeigniter

Langkah	Keterangan
1	<p>Extract codeigniter yang sudah di unduh dan letakkan pada folder htdocs. Beri nama Code-kedelapan dan coba akses di http://localhost/Code-kedelapan/.</p> 
2	<p>Selanjutnya lakukan konfigurasi dasar pada config.php seperti berikut:</p> <pre data-bbox="342 1041 1060 1486"> 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 34 renamed it to 35 something else. If you are using mod_rewrite to remove the 36 page set this 37 variable so that it is blank. 38 */ 39 \$config['index_page'] = ''; 40 */ </pre> <p>Dan file autoload.php</p> <pre data-bbox="342 1570 1156 1898"> 59 \$autoload['libraries'] = array('user_agent' => 'ua'); 60 */ 61 \$autoload['libraries'] = array('database'); 62 63 /* 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 \$autoload['helper'] = array('url', 'file'); 91 */ 92 \$autoload['helper'] = array('url'); 93 94 /* </pre>

	<p>Dan database.php seperti berikut:</p> <pre> 76 ▼ \$db['default'] = array(77 'dsn' => '', 78 'hostname' => 'localhost', 79 'username' => 'root', 80 'password' => 'mysql', 81 'database' => 'rest_server', 82 'dbdriver' => 'mysqli', 83 'dbprefix' => '', </pre>
3	<p>Selanjutnya buatlah database rest_server pada phpmyadmin dan masukkan data dummy.</p> <p>Untuk data dummy dapat dibuat sebagai berikut:</p>  <pre> 1 CREATE TABLE `mahasiswa` (2 `id` int(11) NOT NULL, 3 `nrp` char(9) NOT NULL, 4 `nama` varchar(250) NOT NULL, 5 `email` varchar(250) DEFAULT NULL, 6 `jurusan` varchar(64) NOT NULL 7) ENGINE=InnoDB DEFAULT CHARSET=latin1; 8 9 INSERT INTO `mahasiswa` (`id`, `nrp`, `nama`, `email`, `jurusan`) VALUES 10 (1, '043040001', 'Doddy Ferdiansyah', 'doy@gmail.com', 'Teknik Mesin'), 11 (2, '023040123', 'Erik', 'erik@gmail.com', 'Teknik Industri'), 12 (3, '043040321', 'Rommy Fauzi', 'rommy@gmail.com', 'Teknik Planologi'), 13 (4, '033040023', 'Fajar Darmawan', 'fajar@yahoo.com', 'Teknik Informatika'), 14 (5, '113040321', 'Ferry Mulyanto', 'ferry@yahoo.com', 'Manajemen'); 15 </pre>
4	<p>Selanjutnya buka project http://localhost/Code-kedelapan/. Jika berhasil, tampilan akan berubah menjadi seperti ini:</p>  <p>REST Server Tests</p> <p>Home</p> <p>See the article http://net.tutsplus.com/tutorials/php/working-with-restful-services-in-codeigniter-2/</p> <p>The master project repository is https://github.com/chriskacerguis/codeigniter-restserver</p> <p>Click on the links to check whether the REST server is working.</p> <ol style="list-style-type: none"> 1. Users - defaulting to JSON 2. Users - get it in CSV 3. User #1 - defaulting to JSON (users/id/1) 4. User #1 - defaulting to JSON (users/1) 5. User #1 - get it in XML (users/id/1.xml) 6. User #1 - get it in XML (users/id/1/format/xml) 7. User #1 - get it in XML (users/id/1?format=xml) 8. User #1 - get it in XML (users/1.xml) 9. Users - get it in JSON (AJAX request) 10. Users - get it in HTML (users.html) 11. Users - get it in HTML (users/format/html) 12. Users - get it in HTML (users?format=html) <p>Page rendered in 0.0608 seconds. CodeIgniter Version 3.1.2</p>
5	<p>Tahap instalasi Library Rest Server selesai</p>

Praktikum – Bagian 2: Membuat Aplikasi CRUD Restful Codeigniter

Langkah	Keterangan
1	<p>Buatlah file Mahasiswa.php pada controller/api. Selanjutnya tuliskan kode berikut:</p> <pre> 1 <?php 2 defined('BASEPATH') OR exit('No direct script access allowed'); 3 4 require APPPATH . 'libraries/REST_Controller.php'; 5 require APPPATH . 'libraries/Format.php'; 6 7 class Mahasiswa extends CI_Controller { 8 public function __construct(){ 9 parent::__construct(); 10 \$this->load->model('Mahasiswa_model','mahasiswa'); 11 } 12 13 public function index_get() 14 { 15 \$id = \$this->get('id'); 16 if (\$id === null){ 17 \$mahasiswa = \$this->mahasiswa->getMahasiswa(); 18 }else{ 19 \$mahasiswa = \$this->mahasiswa->getMahasiswa(\$id); 20 } 21 22 if(\$mahasiswa){ 23 \$this->response([24 'status' => true, 25 'data' => \$mahasiswa 26], REST_Controller::HTTP_OK); 27 }else{ 28 \$this->response([29 'status' => false, 30 'message' => 'id not found' 31], REST_Controller:: HTTP_NOT_FOUND); 32 } 33 } 34 } 35 36 /* End of file Mahasiswa.php */ 37 /* Location: ./application/controllers/api/Mahasiswa.php */ 38 ?> </pre>
2	<p>Selanjutnya Buatlah file Mahasiswa_model.php pada model. Selanjutnya tuliskan kode berikut:</p> <pre> 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 if (\$id === null) { 9 return \$this->db->get('mahasiswa')->result_array(); 10 } else { 11 return \$this->db->get_where('mahasiswa', ['id' => \$id])-> 12 result_array(); 13 } 14 } 15 } 16 17 /* End of file Mahasiswa_model.php */ 18 19 ?> </pre>
3	<p>Buka aplikasi postman, masukkan alamat http://localhost/Code-kedelapan/api/mahasiswa dengan metode GET. dan jika berhasil maka tampilannya akan seperti ini:</p>

	 <p>Kita juga dapat melakukan pencarian berdasarkan id, hasilnya akan seperti ini:</p> 
4	<p>Untuk fungsi lain Create, Update dan Delete. Lanjutkan kode berikut pada file mahasiswa.php:</p> <ul style="list-style-type: none"> Delete

```

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

```

- Post

```

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 }

```

- Put

```

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 ?>

```

5

Dan lanjutkan kode pada file **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($id=null){
8          if ($id == null) {
9              return $this->db->get('mahasiswa')->result_array();
10         } else {
11             return $this->db->get_where('mahasiswa', ['id' => $id])->
12                 result_array();
13         }
14     }
15
16     public function deleteMahasiswa($id){
17         $this->db->delete('mahasiswa', ['id' => $id]);
18         return $this->db->affected_rows();
19     }
20
21     public function createMahasiswa($data){
22         $this->db->insert('mahasiswa', $data);
23         return $this->db->affected_rows();
24     }
25
26     public function updateMahasiswa($data, $id){
27         $this->db->update('mahasiswa', $data, ['id' => $id]);
28         return $this->db->affected_rows();
29     }
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:

The screenshot shows the Postman interface for a POST request to `http://localhost/code-kedelapan/api/mahasiswa`. The request is configured with the following body data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nrp	0403070028	
<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 JSON format:

```

1  {
2    "status": true,
3    "message": "new mahasiswa has been created"
4  }

```

Response details: Status: 201 Created, Time: 5.34s, Size: 325 B.

7

Lakukan juga operasi **Update** data pada aplikasi postman, masukkan parameter id seperti berikut:

PUT http://localhost/code-kedelapan/api/mahasiswa

Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nrp	0403070028	
<input checked="" type="checkbox"/> nama	Joko Anwar	
<input checked="" type="checkbox"/> email	jk_anwar@gmail.com	
<input checked="" type="checkbox"/> jurusan	Teknik Informatika	
<input checked="" type="checkbox"/> id	33	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 459ms Size: 320 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": true,
3   "message": "new mahasiswa has been updated"
4 }
```

8

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

DELETE http://localhost/code-kedelapan/api/mahasiswa

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	33	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 1716ms Size: 307 B Save

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": true,
3   "id": "33",
4   "message": "deleted"
5 }
```