REST API Server Sederhana dengan Codeigniter 3

REST, singkatan bahasa Inggris dari *Representational State Transfer*, adalah suatu gaya arsitektur perangkat lunak untuk untuk pendistibusian sistem hipermedia seperti www. Istilah ini diperkenalkan pertama kali pada tahun 2000 pada disertasi doktoral Roy Fielding. Pada arsitektur REST, REST server menyediakan *resources* (sumber daya/data) dan REST client mengakses dan menampilkan *resource* tersebut untuk penggunaan selanjutnya.

CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. Dalam penerapan REST pada Codeigniter diperlukan beberapa library tambahan yang tidak disediakan secara *default* pada Codeigniter, salah satu library yang dapat digunakan adalah library dari **Chris Kacerguis**.

Sebagai contoh penerapan REST pada Codeigniter di bawah ini akan dijelaskan langkah-langkah pembuatan REST API server sederhana tentang CRUD kontak nomor telepon. REST API server tersebut selanjutnya akan diuji menggunakan Postman yang merupakan aplikasi ekstensi/tambahan dari Google Chrome.

Persiapan

Dalam pembuatan Rest API server ini diperlukan :

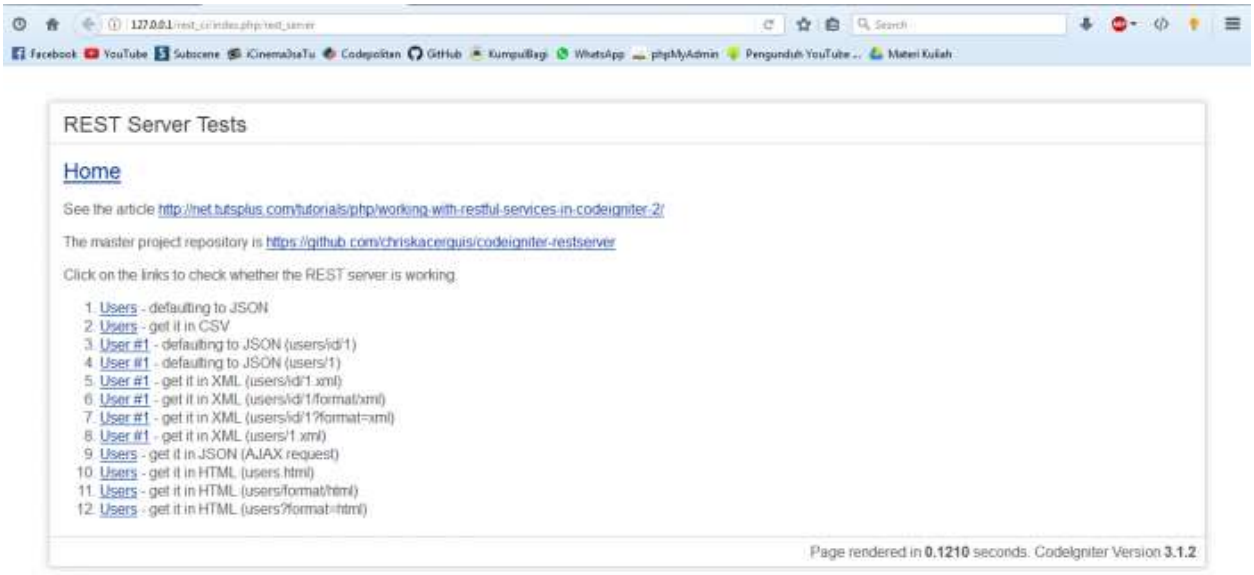
1. Webserver seperti Xampp, Wampp, atau lainnya.

2. Codeigniter dan library REST server yang diperlukan dapat diunduh

   di **https://github.com/ardisaurus/ci-restserver**.

   Setelah semua yang diperlukan telah siap, extract Codeigniter dan library REST server yang telah didownload dan pindah ke htdocs pada direktori xampp lalu rename folder Codeigniter dan library REST server menjadi rest_ci.

Masukan **http://127.0.0.1/rest_ci/index.php/rest_server** pada address bar browser anda jika muncul gambar seperti dibawah maka instalasi telah berhasil.



Konfigurasi database

Buat database baru dengan nama "kontak" :

```
CREATE DATABASE kontak;
```

Buat tabel baru dengan nama "telepon" dengan field id (int 11 AUTO_INCREMENT), nama (varchar 30), nomor (varchar 11):

```
USE kontak;
CREATE TABLE IF NOT EXISTS `telepon` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama` varchar(50) NOT NULL,
  `nomor` varchar(13) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```

Masukan beberapa data contoh :

```
USE kontak;
INSERT INTO `telepon` (`id`, `nama`, `nomor`) VALUES
(1, 'Orion', '08576666762'),
(2, 'Mars', '08576666770'),
(7, 'Alpha', '08576666765');
```

Buka database.php pada rest_ci/application/config ubah menjadi

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn'   => '',
    'hostname' => 'localhost',
    'username' => 'root',
```

```
    'password' => '',
    'database' => 'kontak',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

GET

Metode GET menyediakan akses baca pada sumber daya yang disediakan oleh REST API. Sebagai contohnya digunakan untuk membaca data dari tabel telepon pada database kontak. Untuk membaca data dari database dapat dilakukan dengan active record yang telah disediakan Codeigniter. Sebelum membaca data dari database, fungsi GET yang akan dibuat terlebih dahulu memeriksa apakah terdapat property id pada address bar sehingga data yang ditampilkan dapat di seleksi berdasarkan id atau ditampilkan semua.

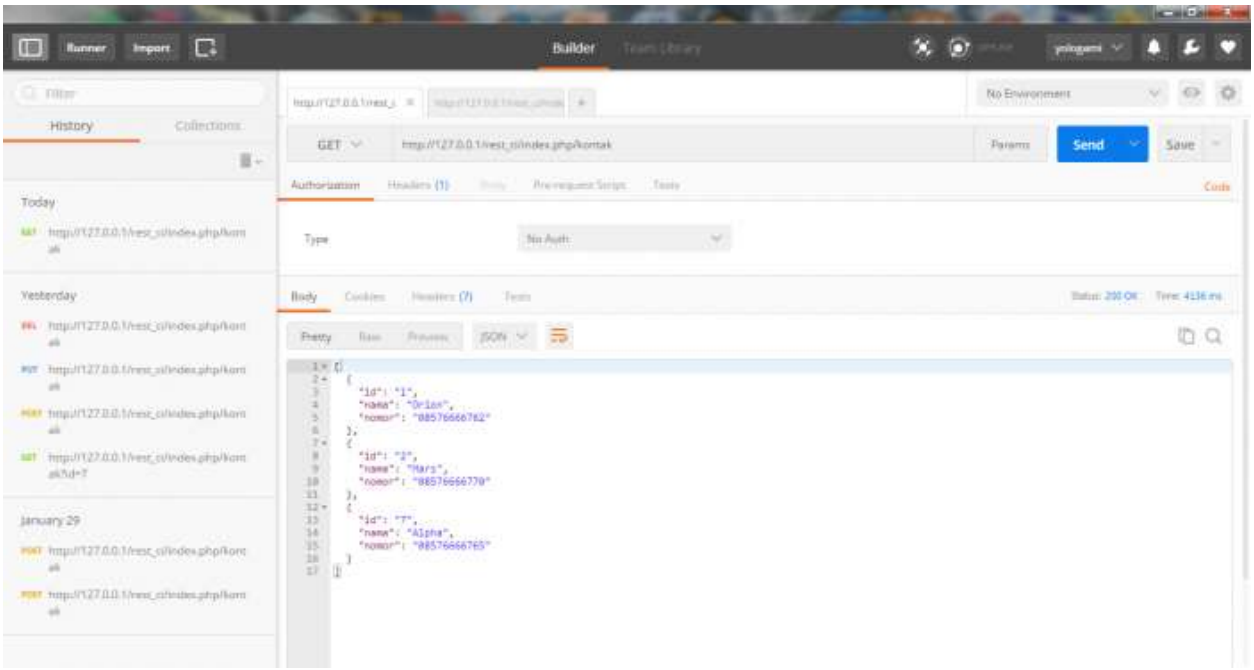
Buat file php baru di di rest_ci/application/controller dengan nama kontak.php.

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');
```

```
require APPPATH . '/libraries/REST_Controller.php';
use Restserver\Libraries\REST_Controller;

class Kontak extends REST_Controller {

    function __construct($config = 'rest') {
        parent::__construct($config);
        $this->load->database();
    }

    //Menampilkan data kontak
    function index_get() {
        $id = $this->get('id');
        if ($id == '') {
            $kontak = $this->db->get('telepon')->result();
        } else {
            $this->db->where('id', $id);
            $kontak = $this->db->get('telepon')->result();
        }
        $this->response($kontak, 200);
    }

    //Masukan function selanjutnya disini
}
?>
```

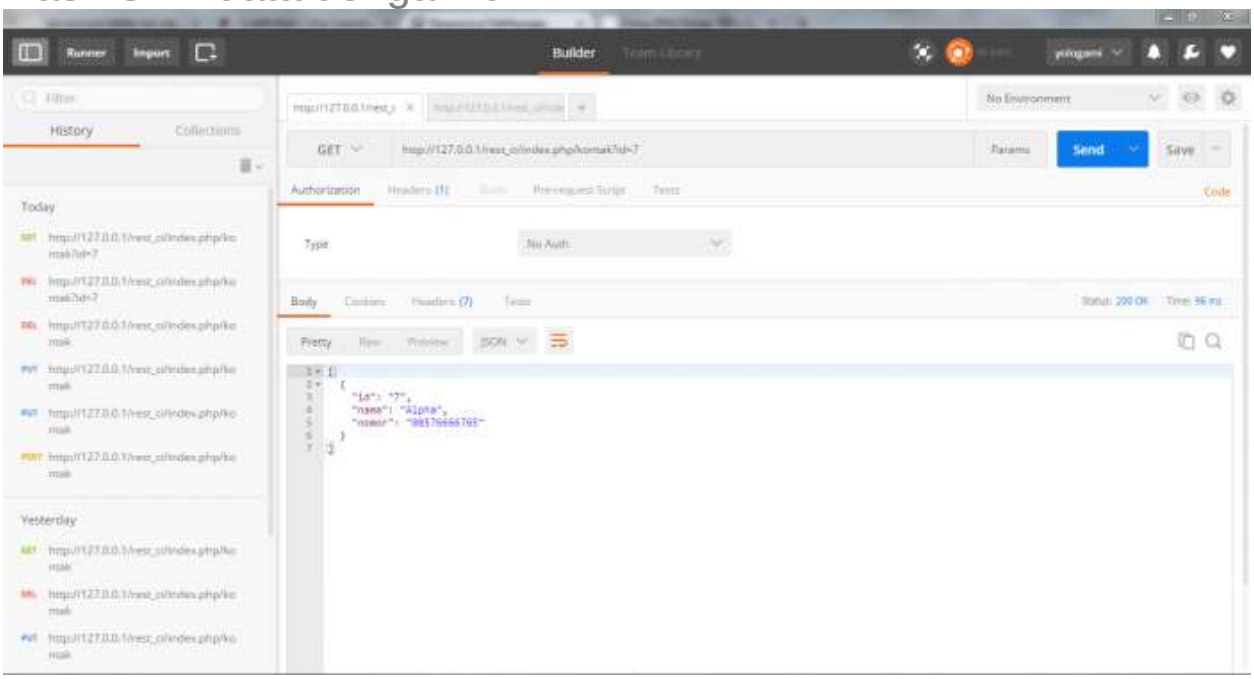
Untuk menguji kode yang telah dibuat buka Postman, Pilih metode GET,
masukan **http://127.0.0.1/rest_ci/index.php/kontak** pada address bar lalu klik "Send".

Hasil GET semua data :



Ubah address pada address bar menjadi **http://127.0.0.1/rest_ci/index.php/kontak?id=7** lalu klik "Send".

Hasil GET data dengan id :



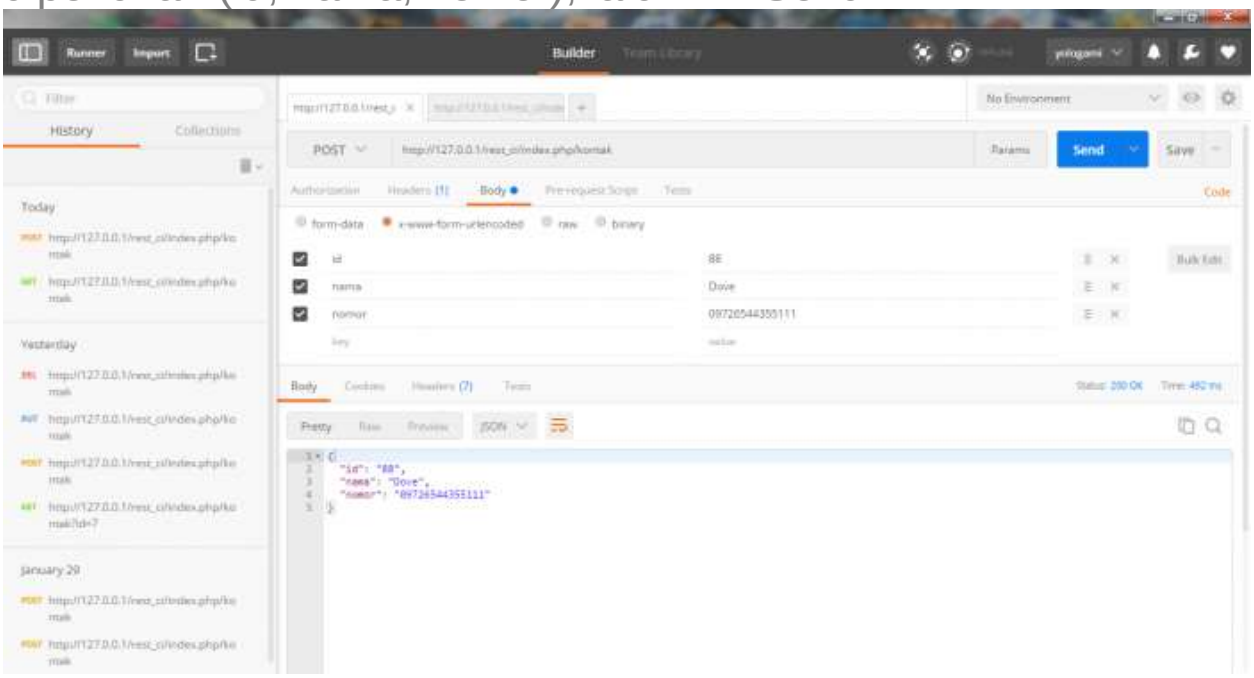
POST

Metode POST digunakan untuk mengirimkan data baru dari client ke server REST API. Sebagai contohnya digunakan untuk menambahkan kontak baru yang terdiri dari id, nama, dan nomor.

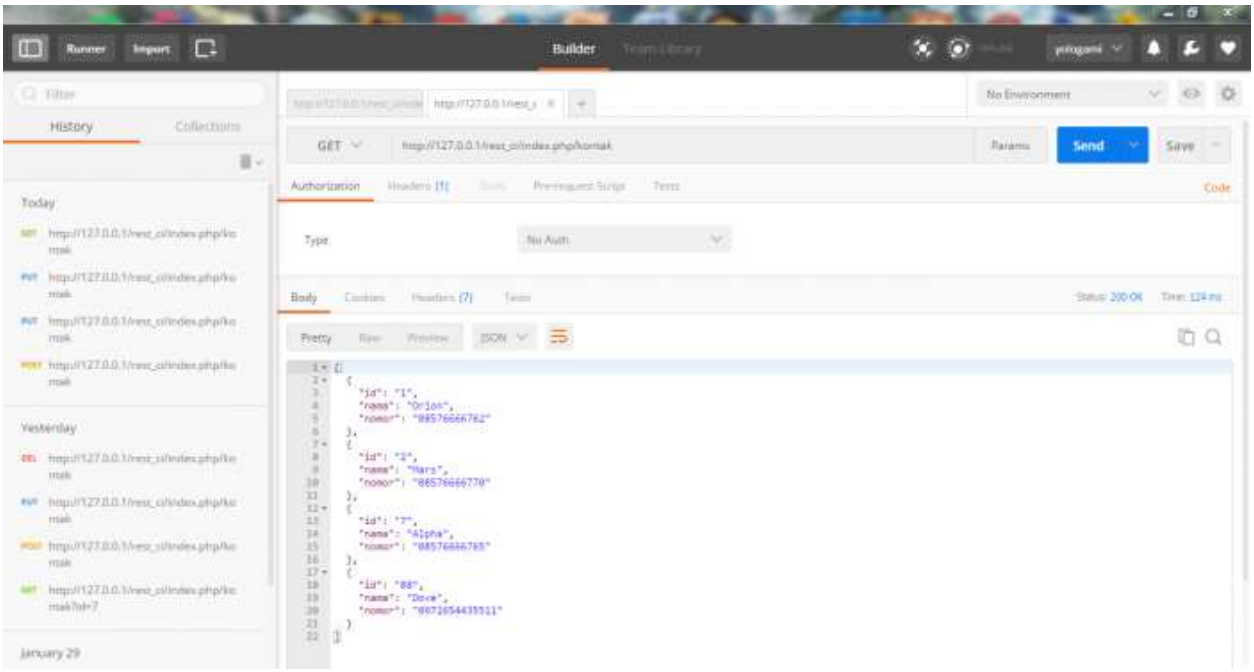
```
//Mengirim atau menambah data kontak baru
function index_post() {
    $data = array(
                'id'           => $this->post('id'),
                'nama'          => $this->post('nama'),
                'nomor'    => $this->post('nomor'));
    $insert = $this->db->insert('telepon', $data);
    if ($insert) {
        $this->response($data, 200);
    } else {
        $this->response(array('status' => 'fail', 502));
    }
}

//Masukan function selanjutnya disini
```

Untuk mengujinya buka Postman, pilih metode POST, masukan **http://127.0.0.1/rest_ci/index.php/kontak** pada address bar, klik "Body" pada menu dibawah address bar, pilih x-www-form-urlencoded, masukan key dan value yang

diperlukan (id, nama, nomor), lalu klik "Send".



Lakukan metode GET untuk melihat data terbaru.



PUT

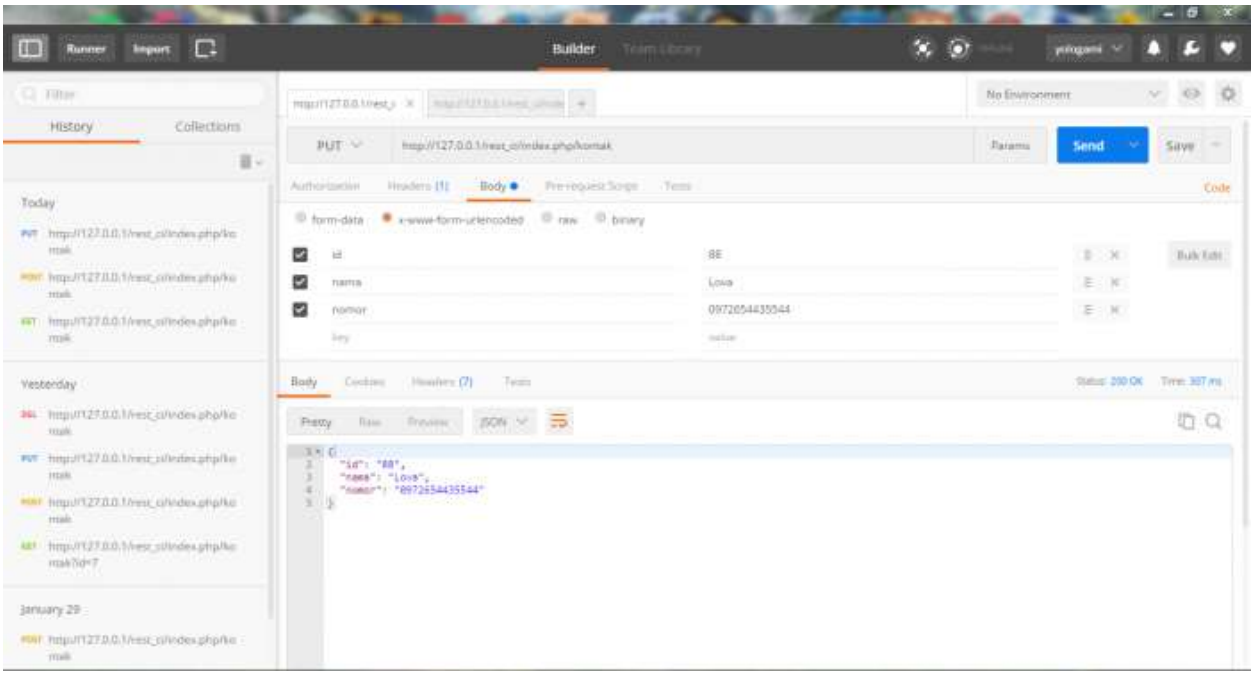
Metode PUT digunakan untuk memperbarui data yang telah ada di server REST API. Sebagai contohnya digunakan untuk

memperbarui data dengan id 88 pada tabel telepon database kontak.

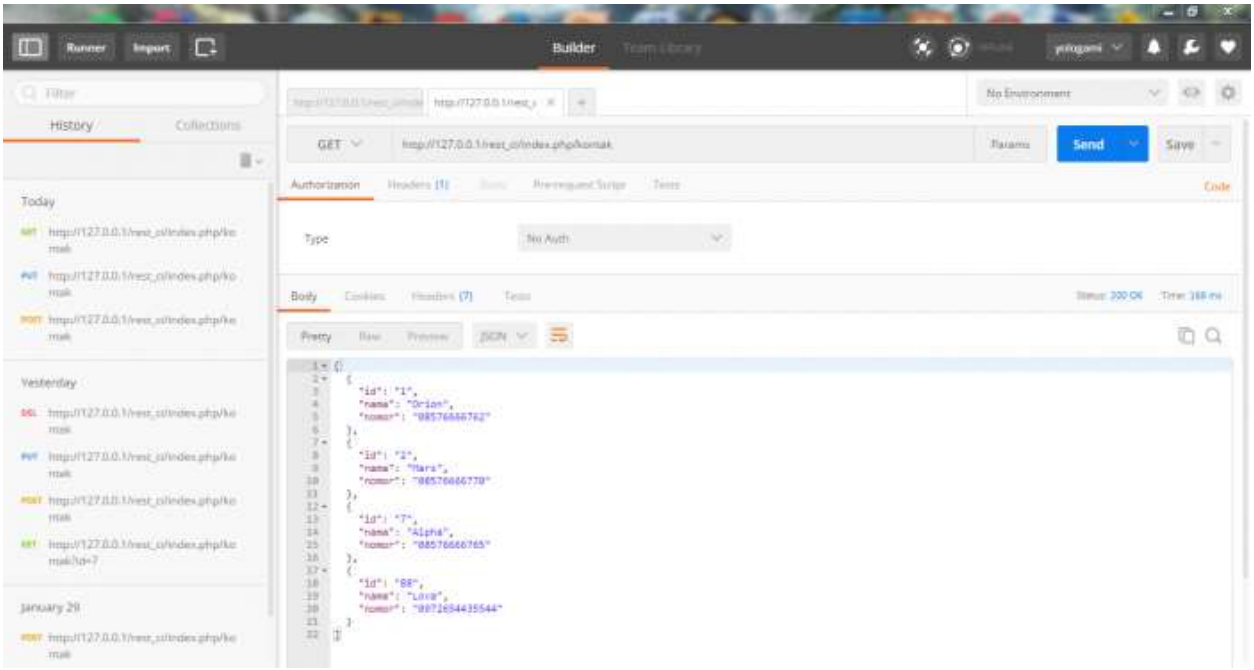
```
    //Memperbarui data kontak yang telah ada
    function index_put() {
        $id = $this->put('id');
        $data = array(
                    'id'       => $this->put('id'),
                    'nama'          => $this->put('nama'),
                    'nomor'    => $this->put('nomor'));
        $this->db->where('id', $id);
        $update = $this->db->update('telepon', $data);
        if ($update) {
            $this->response($data, 200);
        } else {
            $this->response(array('status' => 'fail', 502));
        }
    }

    //Masukan function selanjutnya disini
```

Untuk mengujinya buka Postman, pilih metode PUT, masukan **http://127.0.0.1/rest_ci/index.php/kontak** pada address bar, klik "Body" pada menu dibawah address bar, pilih x-www-form-urlencoded, masukan key id dan value id yang akan diubah (88) diikuti key dan value selanjutnya, lalu klik

"Send".



Lakukan metode GET untuk melihat data terbaru.



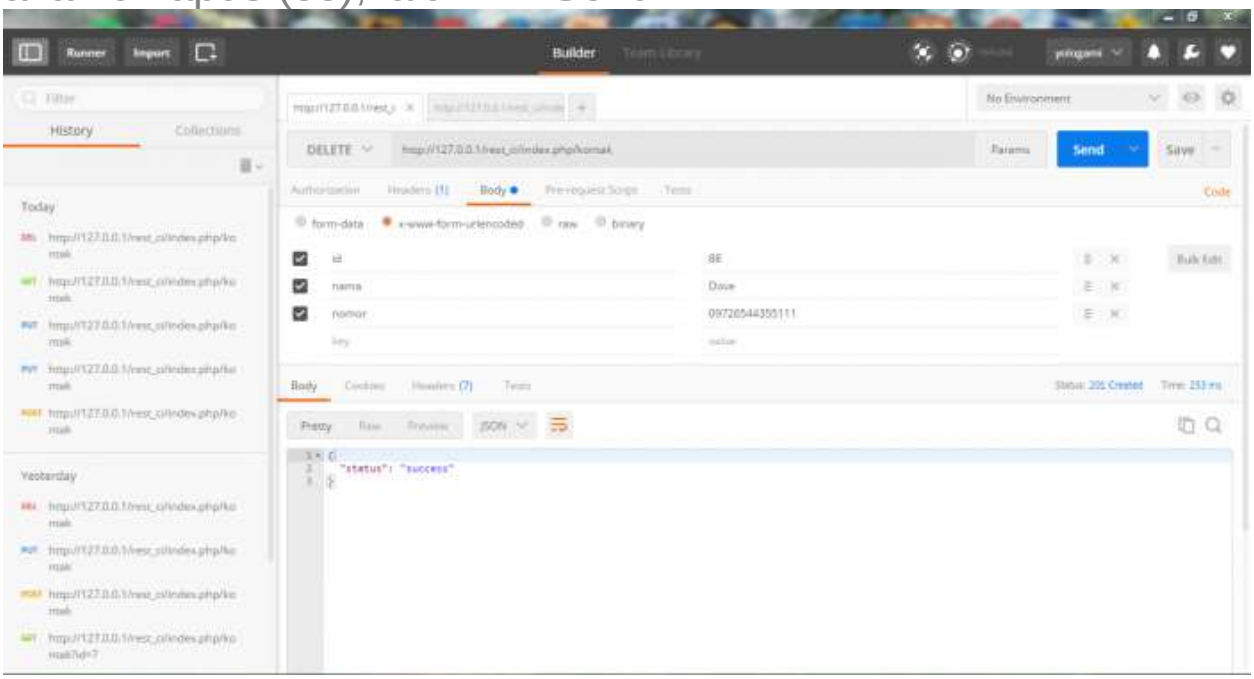
DELETE

Metode DELETE digunakan untuk menghapus data yang telah ada di server REST API. Sebagai contohnya digunakan untuk

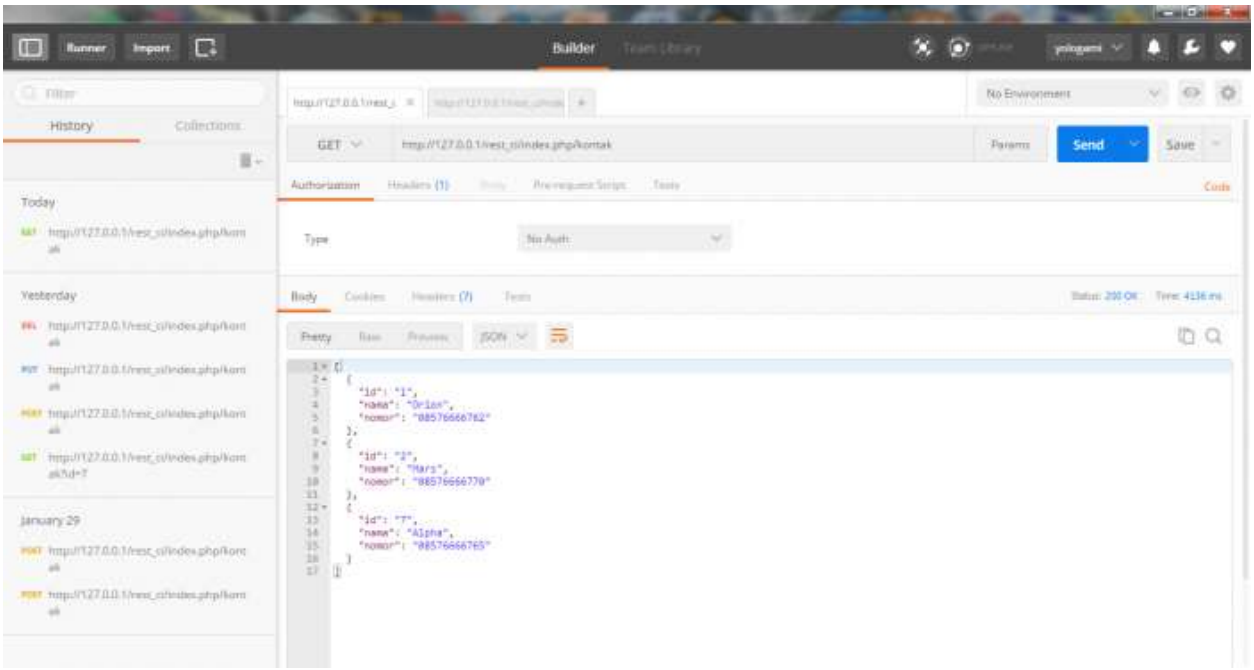
menghapus data dengan id 88 pada tabel telepon database kontak.

```
//Menghapus salah satu data kontak
function index_delete() {
    $id = $this->delete('id');
    $this->db->where('id', $id);
    $delete = $this->db->delete('telepon');
    if ($delete) {
        $this->response(array('status' => 'success'), 201);
    } else {
        $this->response(array('status' => 'fail', 502));
    }
}
```

Untuk mengujinya buka Postman, pilih metode DELETE, masukan **http://127.0.0.1/rest_ci/index.php/kontak** pada address bar, klik "Body" pada menu dibawah address bar, pilih x-www-form-urlencoded, masukan key id dan value id yang akan dihapus (88), lalu klik "Send".



Lakukan metode GET untuk melihat data terbaru.



Keseluruhan isi dari kontak.php pada rest_ci pada rest_ci/application/controller akan tampak seperti dibawah ini.

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';
use Restserver\Libraries\REST_Controller;

class Kontak extends REST_Controller {

    function __construct($config = 'rest') {
        parent::__construct($config);
        $this->load->database();
    }

    function index_get() {
        $id = $this->get('id');
        if ($id == '') {
            $kontak = $this->db->get('telepon')->result();
        } else {
            $this->db->where('id', $id);
```

```
            $kontak = $this->db->get('telepon')->result();
        }
        $this->response($kontak, 200);
    }

    function index_post() {
        $data = array(
                    'id'           => $this->post('id'),
                    'nama'          => $this->post('nama'),
                    'nomor'    => $this->post('nomor'));
        $insert = $this->db->insert('telepon', $data);
        if ($insert) {
            $this->response($data, 200);
        } else {
            $this->response(array('status' => 'fail', 502));
        }
    }

    function index_put() {
        $id = $this->put('id');
        $data = array(
                    'id'       => $this->put('id'),
                    'nama'          => $this->put('nama'),
                    'nomor'    => $this->put('nomor'));
        $this->db->where('id', $id);
        $update = $this->db->update('telepon', $data);
        if ($update) {
            $this->response($data, 200);
        } else {
            $this->response(array('status' => 'fail', 502));
        }
    }

    function index_delete() {
        $id = $this->delete('id');
        $this->db->where('id', $id);
        $delete = $this->db->delete('telepon');
        if ($delete) {
            $this->response(array('status' => 'success'), 201);
        } else {
            $this->response(array('status' => 'fail', 502));
        }
    }
```

```
}
?>
```

Selanjutnya sumber daya dari REST API tersebut dapat dimanfaatkan dengan aplikasi web, desktop, atau mobile yang menjadi client dari REST API tersebut.