**Membuat RESTFul API Menggunakan Framework CodeIgniter 3 PART 2**



Haii….  
Pada kesempatan sebelumnya saya telah menjelaskan cara instalasi sampai *testing rest library* pada *codeigniter* 3. Bagi yang belum baca bisa lihat di [sini](https://medium.com/@erthru/membuat-restful-api-menggunakan-framework-codeigniter-3-f350755b1274).

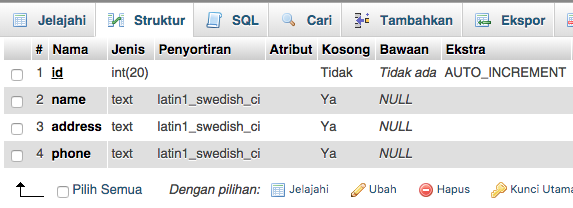
Pada tutorial kali ini saya akan mencoba menggunakan *database MySQL* sebagai data yang akan dibuat *API*-nya.

**1.Buat *Database***  
Buatlah database baru dengan nama db\_restci

create database db\_restci

**2.Tambahkan Tabel**  
Tambahkan tabel baru dengan nama tb\_person dengan 4 field yaitu *id*, *name*, *address*, dan *phone*

create table tb\_person (id INT(20) PRIMARY KEY NOT NULL AUTO\_INCREMENT, name TEXT, address TEXT, phone TEXT)

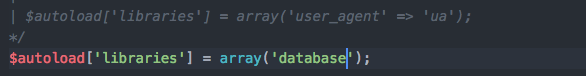


struktur tabel person

**3.Setting database di codeigniter**  
Masuk ke file database.php (**restCi/application/config/database.php**), kemudian masih di *directory* yang sama buka file **autoload.php**



sesuaikan dengan username dan password di host kalian



ubah atuload[‘libaries’] menjadi seperti digambar

**4.Buat File Model Baru**  
Masuk kembali ke directory restCI dan masuk ke directory Model (**restCi/application/models/**) kemudian buat satu file dengan nama **PersonM.php** , buat script yang ada pada **PersonM.php** menjadi seperti berikut.

<?php// extends class Model  
class PersonM extends CI\_Model{// response jika field ada yang kosong  
 public function empty\_response(){  
 $response['status']=502;  
 $response['error']=true;  
 $response['message']='Field tidak boleh kosong';  
 return $response;  
 }// function untuk insert data ke tabel tb\_person  
 public function add\_person($name,$address,$phone){if(empty($name) || empty($address) || empty($phone)){  
 return $this->empty\_response();  
 }else{  
 $data = array(  
 "name"=>$name,  
 "address"=>$address,  
 "phone"=>$phone  
 );$insert = $this->db->insert("tb\_person", $data);if($insert){  
 $response['status']=200;  
 $response['error']=false;  
 $response['message']='Data person ditambahkan.';  
 return $response;  
 }else{  
 $response['status']=502;  
 $response['error']=true;  
 $response['message']='Data person gagal ditambahkan.';  
 return $response;  
 }  
 }}// mengambil semua data person  
 public function all\_person(){$all = $this->db->get("tb\_person")->result();  
 $response['status']=200;  
 $response['error']=false;  
 $response['person']=$all;  
 return $response;}// hapus data person  
 public function delete\_person($id){if($id == ''){  
 return $this->empty\_response();  
 }else{  
 $where = array(  
 "id"=>$id  
 );$this->db->where($where);  
 $delete = $this->db->delete("tb\_person");  
 if($delete){  
 $response['status']=200;  
 $response['error']=false;  
 $response['message']='Data person dihapus.';  
 return $response;  
 }else{  
 $response['status']=502;  
 $response['error']=true;  
 $response['message']='Data person gagal dihapus.';  
 return $response;  
 }  
 }}// update person  
 public function update\_person($id,$name,$address,$phone){if($id == '' || empty($name) || empty($address) || empty($phone)){  
 return $this->empty\_response();  
 }else{  
 $where = array(  
 "id"=>$id  
 );$set = array(  
 "name"=>$name,  
 "address"=>$address,  
 "phone"=>$phone  
 );$this->db->where($where);  
 $update = $this->db->update("tb\_person",$set);  
 if($update){  
 $response['status']=200;  
 $response['error']=false;  
 $response['message']='Data person diubah.';  
 return $response;  
 }else{  
 $response['status']=502;  
 $response['error']=true;  
 $response['message']='Data person gagal diubah.';  
 return $response;  
 }  
 }}}?>

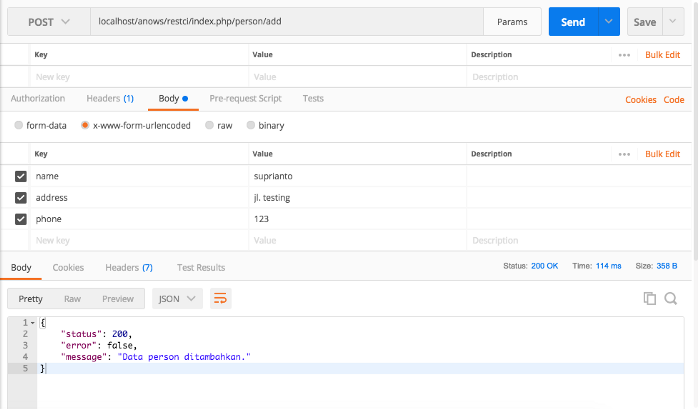
\*penjelasan ada pada script.

**5.Buat *Controller* baru**  
Buatlah *controller* baru dengan nama **Person.php** (**restCi/application/controllers/**), kemudian buat script dari file **Person.php** menjadi seperti berikut.

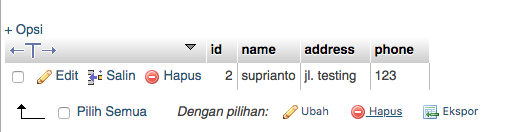
<?phprequire APPPATH . 'libraries/REST\_Controller.php';class Person extends REST\_Controller{// construct  
 public function \_\_construct(){  
 parent::\_\_construct();  
 $this->load->model('PersonM');  
 }// method index untuk menampilkan semua data person menggunakan method get  
 public function index\_get(){  
 $response = $this->PersonM->all\_person();  
 $this->response($response);  
 }// untuk menambah person menaggunakan method post  
 public function add\_post(){  
 $response = $this->PersonM->add\_person(  
 $this->post('name'),  
 $this->post('address'),  
 $this->post('phone')  
 );  
 $this->response($response);  
 }// update data person menggunakan method put  
 public function update\_put(){  
 $response = $this->PersonM->update\_person(  
 $this->put('id'),  
 $this->put('name'),  
 $this->put('address'),  
 $this->put('phone')  
 );  
 $this->response($response);  
 }// hapus data person menggunakan method delete  
 public function delete\_delete(){  
 $response = $this->PersonM->delete\_person(  
 $this->delete('id')  
 );  
 $this->response($response);  
 }}?>

\*Penjelasan ada pada   
Pada controller saya mencoba beberapa method yang sering digunakan pada rest api yaitu get, post, put, delete.

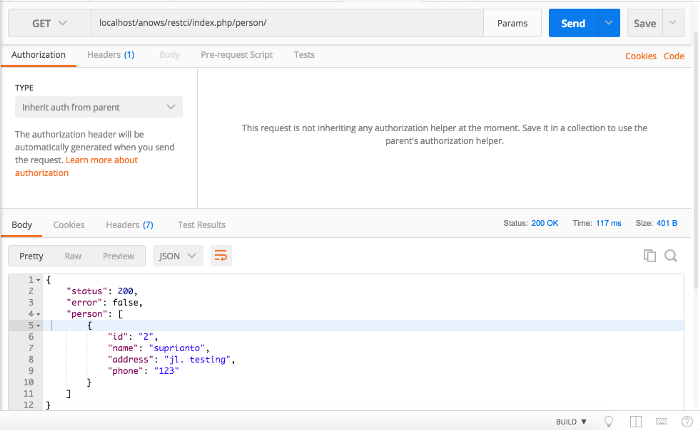
**6.*Testing***Setelah langkah langkah di atas maka saya akan mencoba rest api pada aplikasi postman.



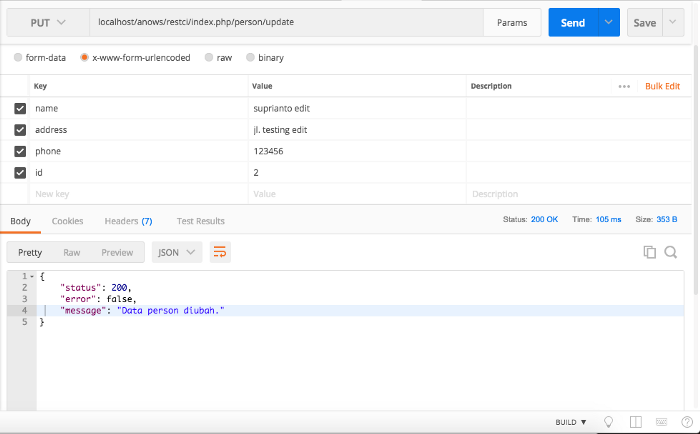
tambah data person



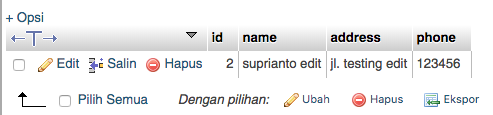
berhasil ditambahkan ke database



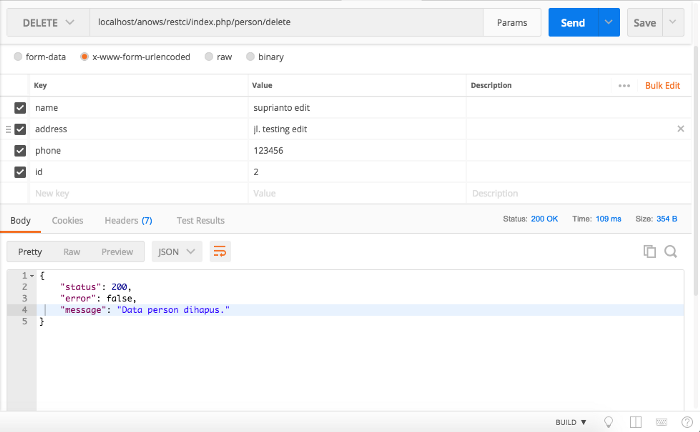
menampilkan semua data person



update data person



data person di database berhasil diubah



hapus data person



data yang hanya 1 tadi di database berhasil dihapus

Oke.., sampai disini saya telah membuat operasi *Create, Read, Update, Delete* pada *RESTFul API* menggunakan *codeigniter* 3.  
Sampai jumpa di tutorial berikutnya.

[[download project](https://github.com/erthru/restCI)]

Yang ingin Fullstack CRUD tutorial, Laravel dan VueJS bisa lihat di sini : <https://medium.com/@erthru/laravel-dan-vue-part-1-laravel-sebagai-backend-defafdd7ed0c>

<https://erthru.medium.com/membuat-restful-api-menggunakan-framework-codeigniter-3-part-2-163cadb32e3a>