Nama : Sahlah Rizqiyyah ( 15220689 )

Kelas : 15.5A.05

1.Codingan Database.php

<?php

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

/\*

| -------------------------------------------------------------------

| DATABASE CONNECTIVITY SETTINGS

| -------------------------------------------------------------------

| This file will contain the settings needed to access your database.

|

| For complete instructions please consult the 'Database Connection'

| page of the User Guide.

|

| -------------------------------------------------------------------

| EXPLANATION OF VARIABLES

| -------------------------------------------------------------------

|

|   ['dsn']      The full DSN string describe a connection to the database.

|   ['hostname'] The hostname of your database server.

|   ['username'] The username used to connect to the database

|   ['password'] The password used to connect to the database

|   ['database'] The name of the database you want to connect to

|   ['dbdriver'] The database driver. e.g.: mysqli.

|           Currently supported:

|                cubrid, ibase, mssql, mysql, mysqli, oci8,

|                odbc, pdo, postgre, sqlite, sqlite3, sqlsrv

|   ['dbprefix'] You can add an optional prefix, which will be added

|                to the table name when using the  Query Builder class

|   ['pconnect'] TRUE/FALSE - Whether to use a persistent connection

|   ['db\_debug'] TRUE/FALSE - Whether database errors should be displayed.

|   ['cache\_on'] TRUE/FALSE - Enables/disables query caching

|   ['cachedir'] The path to the folder where cache files should be stored

|   ['char\_set'] The character set used in communicating with the database

|   ['dbcollat'] The character collation used in communicating with the database

|                NOTE: For MySQL and MySQLi databases, this setting is only used

|                as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7

|                (and in table creation queries made with DB Forge).

|                There is an incompatibility in PHP with mysql\_real\_escape\_string() which

|                can make your site vulnerable to SQL injection if you are using a

|                multi-byte character set and are running versions lower than these.

|                Sites using Latin-1 or UTF-8 database character set and collation are unaffected.

|   ['swap\_pre'] A default table prefix that should be swapped with the dbprefix

|   ['encrypt']  Whether or not to use an encrypted connection.

|

|           'mysql' (deprecated), 'sqlsrv' and 'pdo/sqlsrv' drivers accept TRUE/FALSE

|           'mysqli' and 'pdo/mysql' drivers accept an array with the following options:

|

|               'ssl\_key'    - Path to the private key file

|               'ssl\_cert'   - Path to the public key certificate file

|               'ssl\_ca'     - Path to the certificate authority file

|               'ssl\_capath' - Path to a directory containing trusted CA certificats in PEM format

|               'ssl\_cipher' - List of \*allowed\* ciphers to be used for the encryption, separated by colons (':')

|               'ssl\_verify' - TRUE/FALSE; Whether verify the server certificate or not ('mysqli' only)

|

|   ['compress'] Whether or not to use client compression (MySQL only)

|   ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections

|                           - good for ensuring strict SQL while developing

|   ['ssl\_options'] Used to set various SSL options that can be used when making SSL connections.

|   ['failover'] array - A array with 0 or more data for connections if the main should fail.

|   ['save\_queries'] TRUE/FALSE - Whether to "save" all executed queries.

|               NOTE: Disabling this will also effectively disable both

|               $this->db->last\_query() and profiling of DB queries.

|               When you run a query, with this setting set to TRUE (default),

|               CodeIgniter will store the SQL statement for debugging purposes.

|               However, this may cause high memory usage, especially if you run

|               a lot of SQL queries ... disable this to avoid that problem.

|

| The $active\_group variable lets you choose which connection group to

| make active.  By default there is only one group (the 'default' group).

|

| The $query\_builder variables lets you determine whether or not to load

| the query builder class.

\*/

$active\_group = 'default';

$query\_builder = TRUE;

$db['default'] = array(

    'dsn'   => '',

    'hostname' => 'localhost',

    'username' => 'root',

    'password' => '',

    'database' => 'db\_member',

    '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

);

2.Codingan Rest\_member.php

<?php

// Dokumentasi Pengerjaan Individu

// NIM  : 15220689

// Nama : Sahlah Rizqiyyah

// Kelas    : 15.5A.05

// Kampus   : UBSI Kaliabang

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

require APPPATH . '/libraries/REST\_Controller.php';

use Restserver\libraries\REST\_Controller;

class Rest\_member extends REST\_Controller {

    function \_\_construct($config = 'rest'){

        parent::\_\_construct($config);

        $this->load->database();

    }

    //awal fungsi operasi get

    function index\_get() {

        // Deskripsi Fungsi :

        // -------------------------------------------------------------------

        // Script dibawah ini merupakan implementasi dari metode GET

        // -------------------------------------------------------------------

        $id = $this->get('id');

        if ($id == '') {

            $member = $this->db->get('member')->result();

        } else {

            $this->db->where('id', $id);

            $member = $this->db->get('member')->result();

        }

        $this->response($member, 404);

    } //akhir fungsi operasi get

    //awal fungsi operasi post

    function index\_post() {

        $data = array(

                    'id'  => $this->post('id'),

                    'nama\_member'  => $this->post('nama\_member'),

                    'email'  => $this->post('email'),

                    'no\_telp'    => $this->post('no\_telp'));

        $insert = $this->db->insert('member', $data);

        if ($insert) {

            $this->response($data, 200);

        } else {

            $this->response(array('status' => 'fail', 502));

        }

    }// akhir fungsi post

    function index\_put() {

        $id = $this->put('id');

        $data = array(

                    'id'   => $this->put('id'),

                    'nama\_member'  => $this->put('nama\_member'),

                    'email'  => $this->put('email'),

                    'no\_telp'  => $this->put('no\_telp'));

        $this->db->where('id', $id);

        $update = $this->db->update('member', $data);

        if ($update) {

            $this->response($data, 200);

        } else {

            $this->response(array('status' => 'fail', 502));

        }

    }

    //awal funsi operasi delete

    function index\_delete() {

        $id = $this->delete('id');

        $this->db->where('id', $id);

        $delete = $this->db->delete('member');

        if ($delete) {

            $this->response(array('status'=>'sukses'), 200);

        } else {

            $this->response(array('status' => 'gagal', 502));

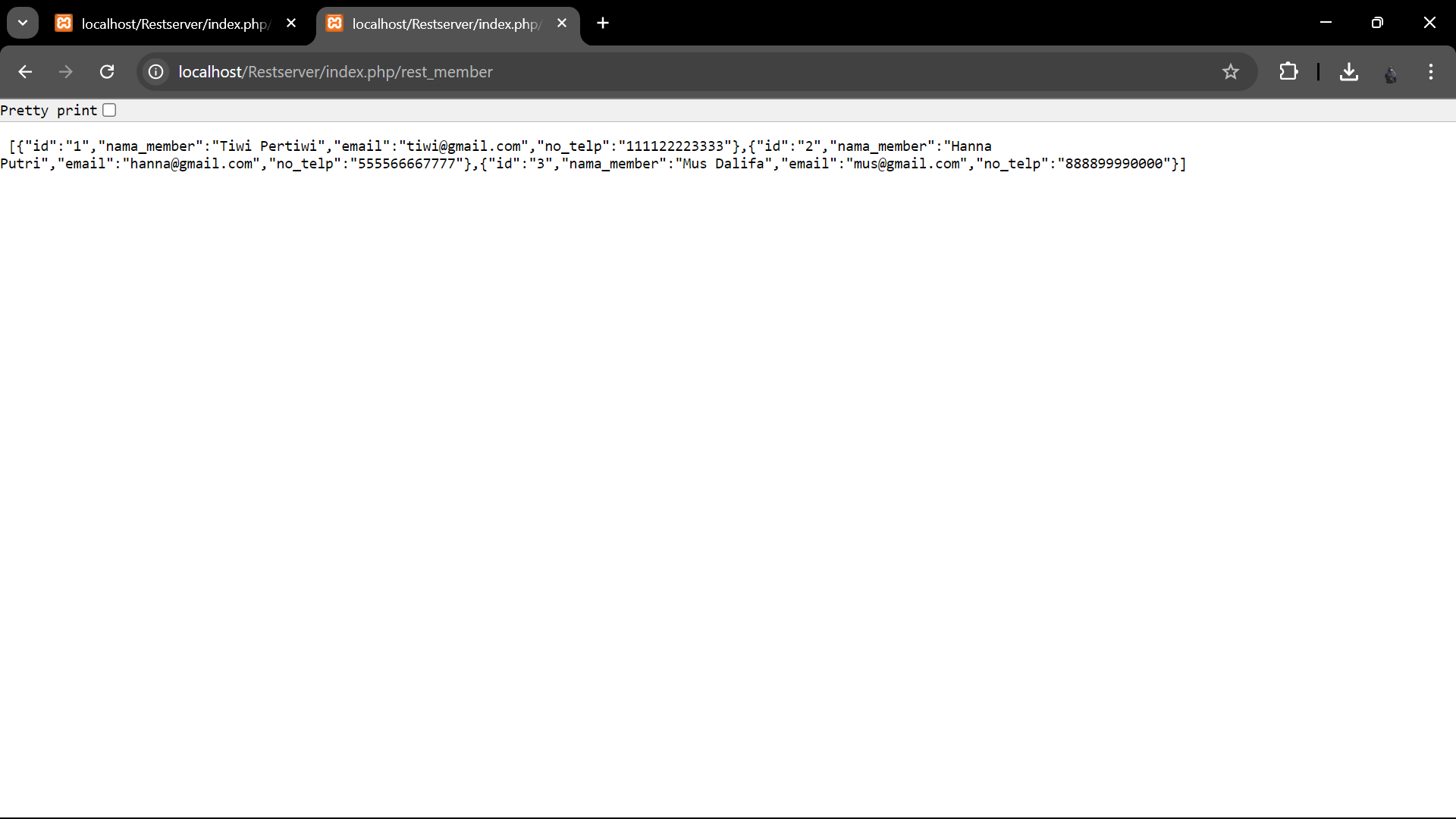
        }

    } // akhir fungsi delete

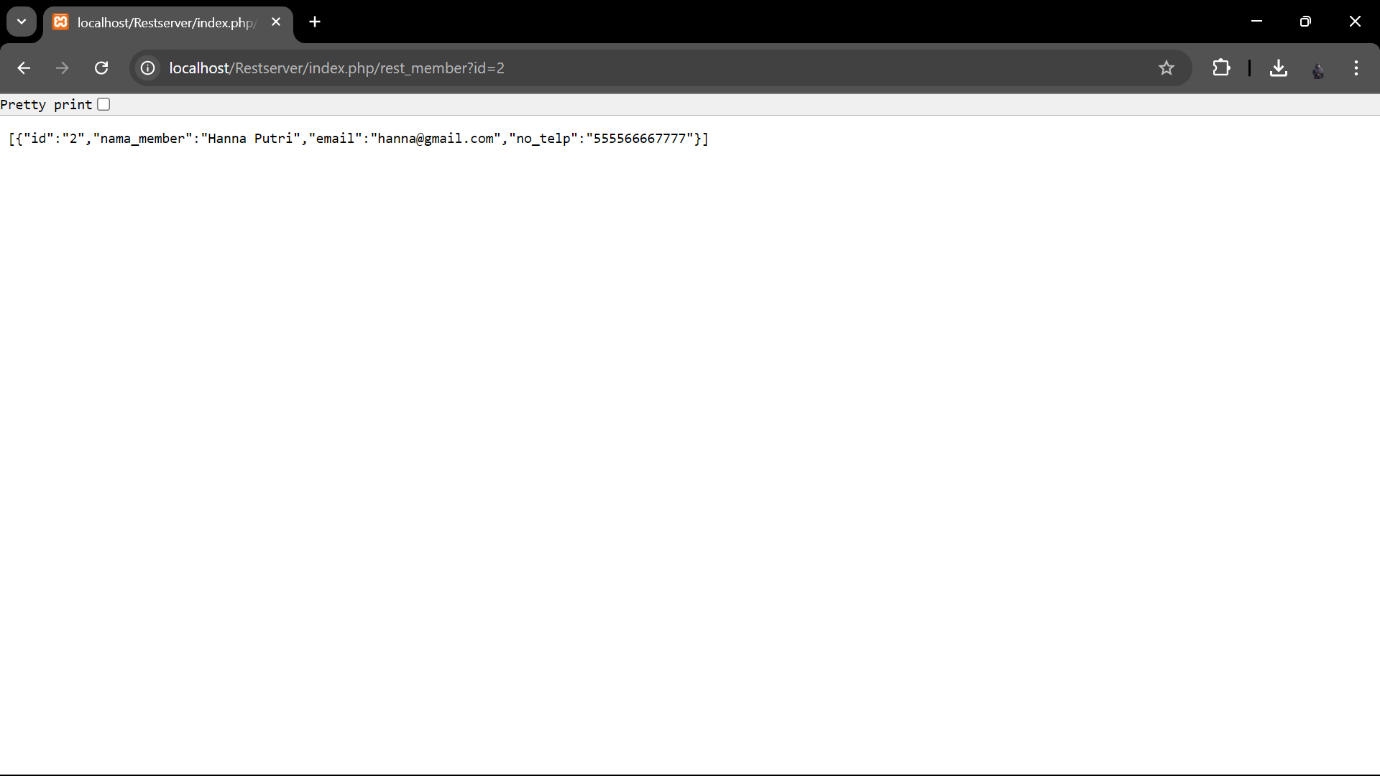
}

?>

3. Ketika tidak ada property id pada addrees bar localhost/Restserver/index.php/rest\_member



4. Ketika ada property id pada addrees bar : localhost/Restserver/index.php/rest\_member ?id=2

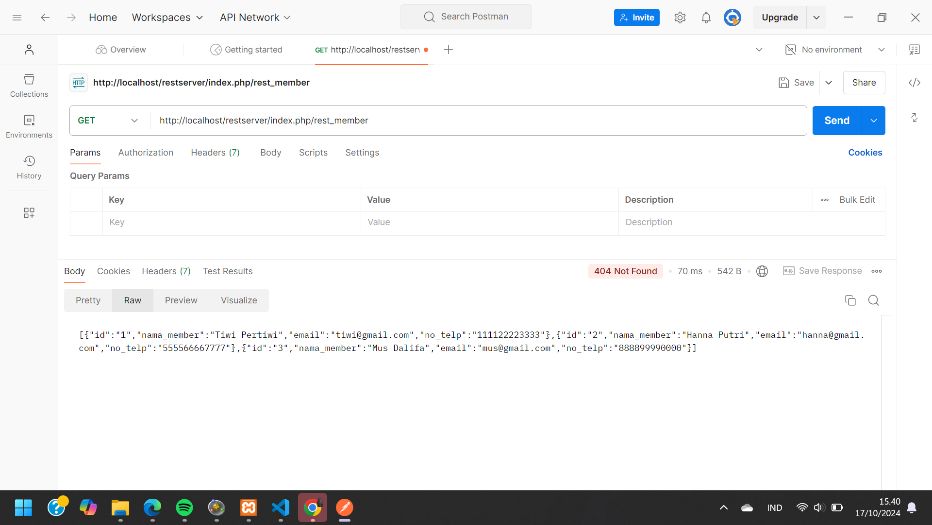


5.Pengujian Get

* Buka Aplikasi postman
* Paste url pada inputan pada postman seperti berikut :

<http://localhost/restserver/index.php/rest_member>

* Tampilannya



6. Pengujian Post

➢ Untuk pengujian POST, maka pilih jenis pengujian POST

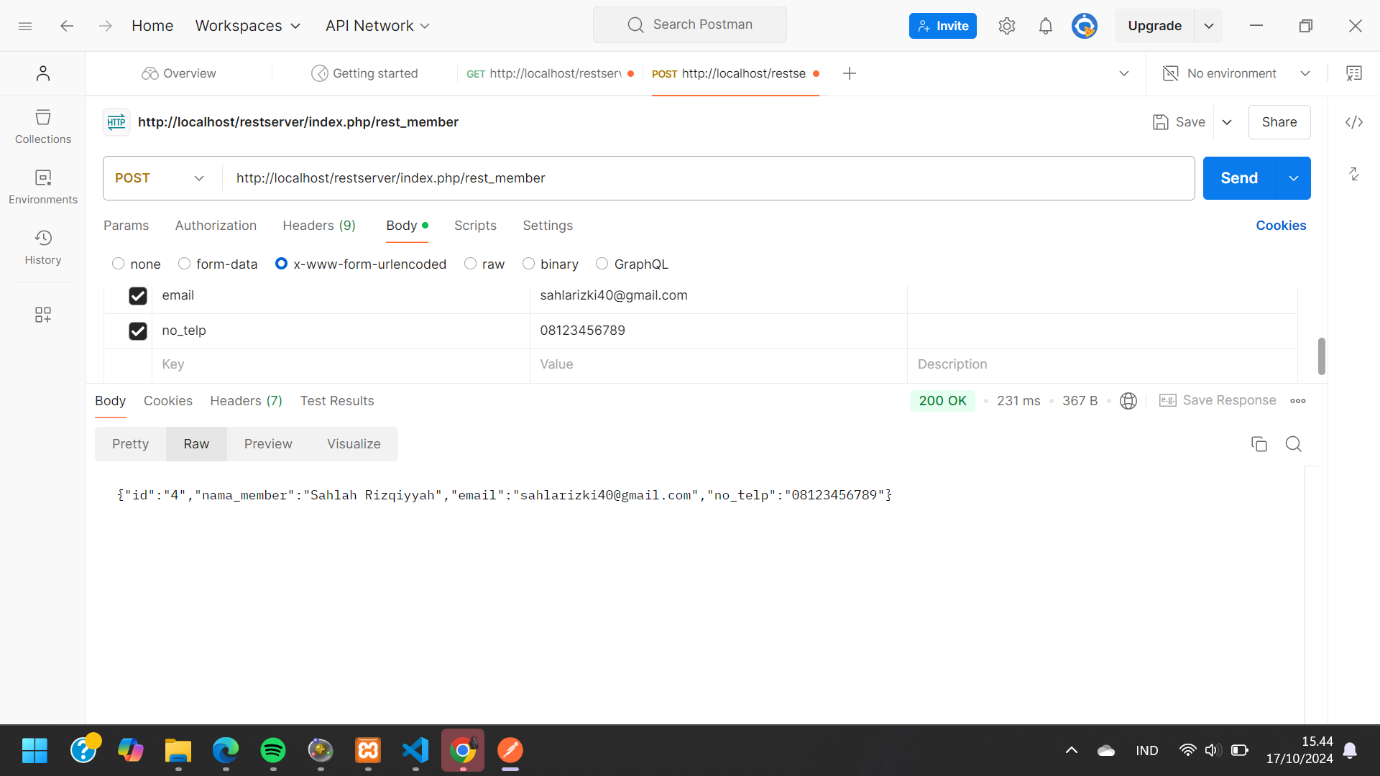
➢ Setelah pengujian dipilih, selanjutnya klik body yang ada dibagian bawah inputan url di atas, kemudian pilih x

www.form-urlendcode.

➢ Selanjutnya isi key sesuai dengan field-field yang ada pada tabel yang digunakan, sedangkan value diisi bebas.

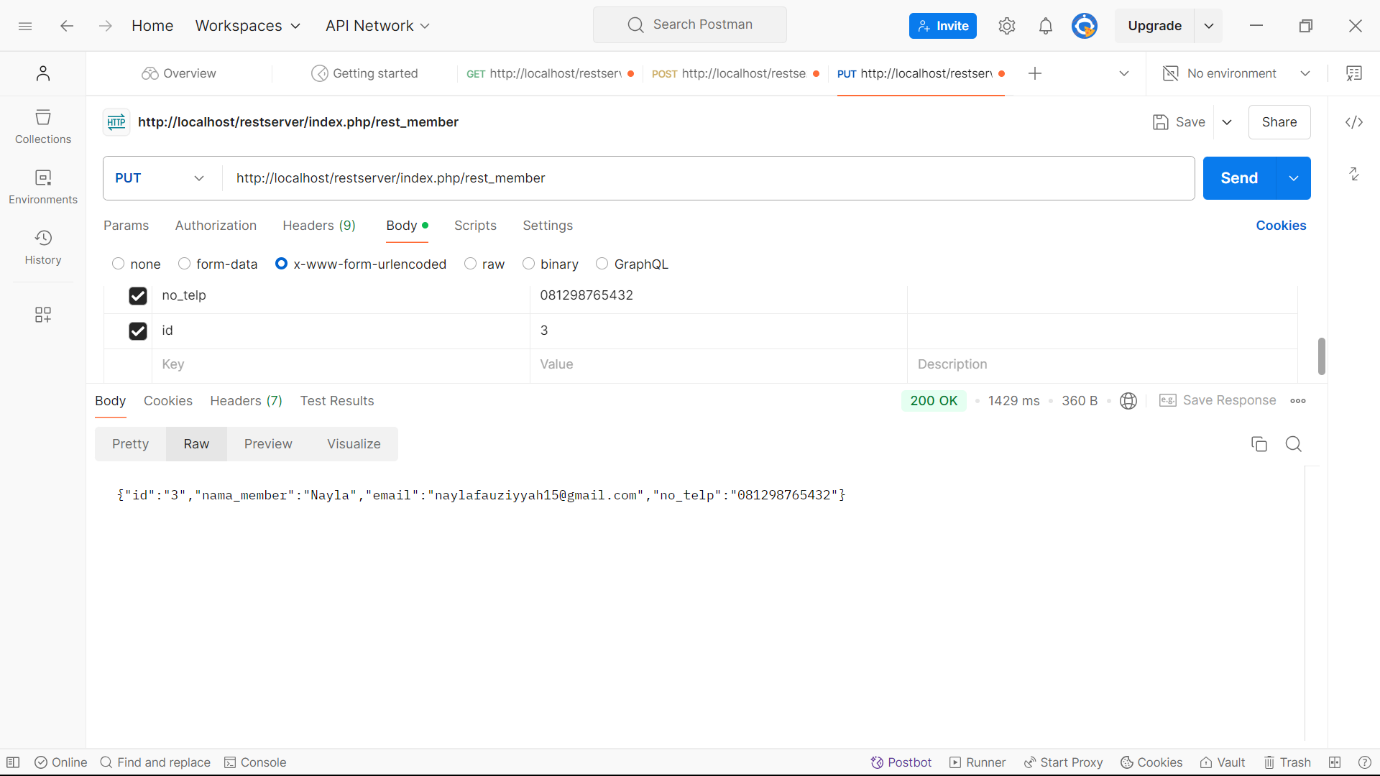
Lalu klik tombol send

➢ Hasilnya pengujian POST seperti berikut:



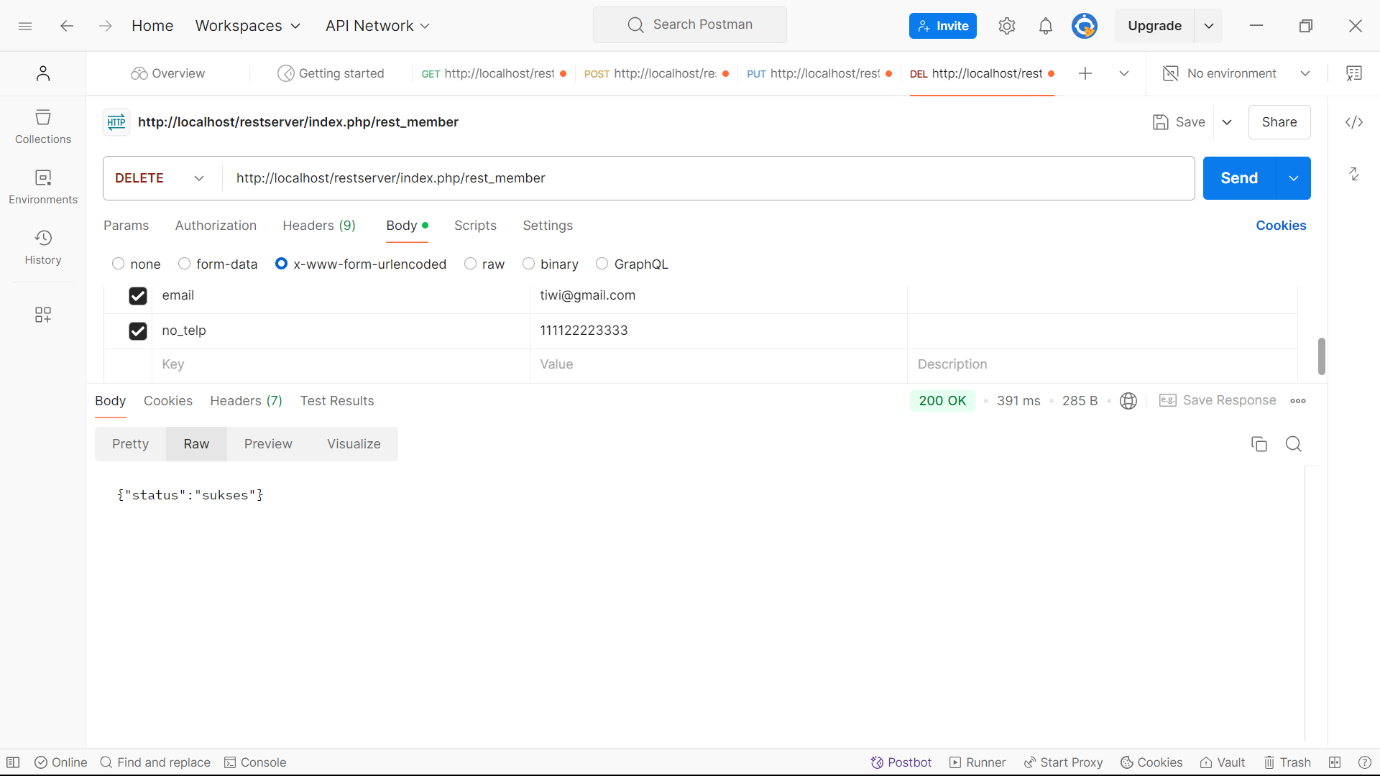
7.Pengujian PUT

➢ Pengujian PUT langkahnya sama seperti metode POST. Hasilnya seperti berikut:



8.Pengujian Delete

➢ Untuk pengujian delete, langkahnya kurang lebih sama dengan POST dan PUT.



9.Vigenesia

**Pengujian RESTFUL API VigeNesia**

Tabel Motivasi

Codingannya

<?php

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

require APPPATH . '/libraries/REST\_Controller.php';

use Restserver\Libraries\REST\_Controller;

class Rest\_member extends REST\_Controller {

    // Konstruktor

    function \_\_construct($config = 'rest') {

        parent::\_\_construct($config);

        $this->load->database(); // Load database

    }

    // Implementasi metode GET

    function index\_get() {

        $id = $this->get('id'

    ); // Mengambil parameter ID dari request GET

        if ($id == '') {

            // Jika ID tidak ada, ambil semua data dari tabel motivasi

            $motivasi = $this->db->get('motivasi')->result();

        } else {

            // Jika ID ada, ambil data sesuai dengan ID

            $this->db->where('id', $id);

            $motivasi = $this->db->get('motivasi')->result();

        }

        // Tampilkan response dalam format JSON

        if ($motivasi) {

            $this->response($motivasi, 200); // Status HTTP 200 OK

        } else {

            $this->response(array('status' => 'fail', 'message' => 'Data not found'), 404); // Status HTTP 404 Not Found

        }

    }

    // Implementasi metode POST

    function index\_post() {

        // Validasi input

        if (!$this->post('id') || !$this->post('isi\_motivasi') || !$this->post('iduser')) {

            $this->response(array('status' => 'fail', 'message' => 'ID, isi\_motivasi, or iduser is missing'), 400); // Status HTTP 400 Bad Request

            return;

        }

        // Data yang akan disimpan ke tabel motivasi

        $data = array(

            'id' => $this->post('id'),

            'isi\_motivasi' => $this->post('isi\_motivasi'),

            'iduser' => $this->post('iduser'),

            'tanggal\_input' => date('Y-m-d', strtotime($this->post('tanggal\_input'))), // Format tanggal

            'tanggal\_update' => ($this->post('tanggal\_update')) ? date('Y-m-d', strtotime($this->post('tanggal\_update'))) : null

        );

        // Insert data ke dalam tabel motivasi

        $insert = $this->db->insert('motivasi', $data);

        // Cek apakah insert berhasil

        if ($insert) {

            $this->response($data, 200); // Status HTTP 200 OK

        } else {

            $error = $this->db->error();

            $this->response(array('status' => 'fail', 'error' => $error), 502); // Status HTTP 502 Bad Gateway

        }

    }

    // Implementasi metode PUT

    function index\_put() {

        $id = $this->put('id'); // Mengambil parameter ID dari request PUT

        // Validasi input

        if (!$id || !$this->put('isi\_motivasi') || !$this->put('iduser')) {

            $this->response(array('status' => 'fail', 'message' => 'ID, isi\_motivasi, or iduser is missing'), 400); // Status HTTP 400 Bad Request

            return;

        }

        $data = array(

            'isi\_motivasi' => $this->put('isi\_motivasi'),

            'iduser' => $this->put('iduser'),

            'tanggal\_input' => date('Y-m-d', strtotime($this->put('tanggal\_input'))),

            'tanggal\_update' => date('Y-m-d', strtotime($this->put('tanggal\_update')))

        );

        // Update data motivasi berdasarkan ID

        $this->db->where('id', $id);

        $update = $this->db->update('motivasi', $data);

        // Cek apakah update berhasil

        if ($update) {

            $this->response($data, 200); // Status HTTP 200 OK

        } else {

            $error = $this->db->error();

            $this->response(array('status' => 'fail', 'error' => $error), 502); // Status HTTP 502 Bad Gateway

        }

    }

    // Implementasi metode DELETE

    function index\_delete() {

        $id = $this->delete('id'); // Mengambil parameter ID dari request DELETE

        if (!$id) {

            $this->response(array('status' => 'fail', 'message' => 'ID is required'), 400); // Status HTTP 400 Bad Request

            return;

        }

        $this->db->where('id', $id); // Menentukan kondisi untuk menghapus

        $delete = $this->db->delete('motivasi'); // Menghapus data

        // Cek apakah delete berhasil

        if ($delete) {

            $this->response(array('status' => 'success', 'message' => 'Data deleted'), 200); // Status HTTP 200 OK

        } else {

            $error = $this->db->error();

            $this->response(array('status' => 'fail', 'error' => $error), 502); // Status HTTP 502 Bad Gateway

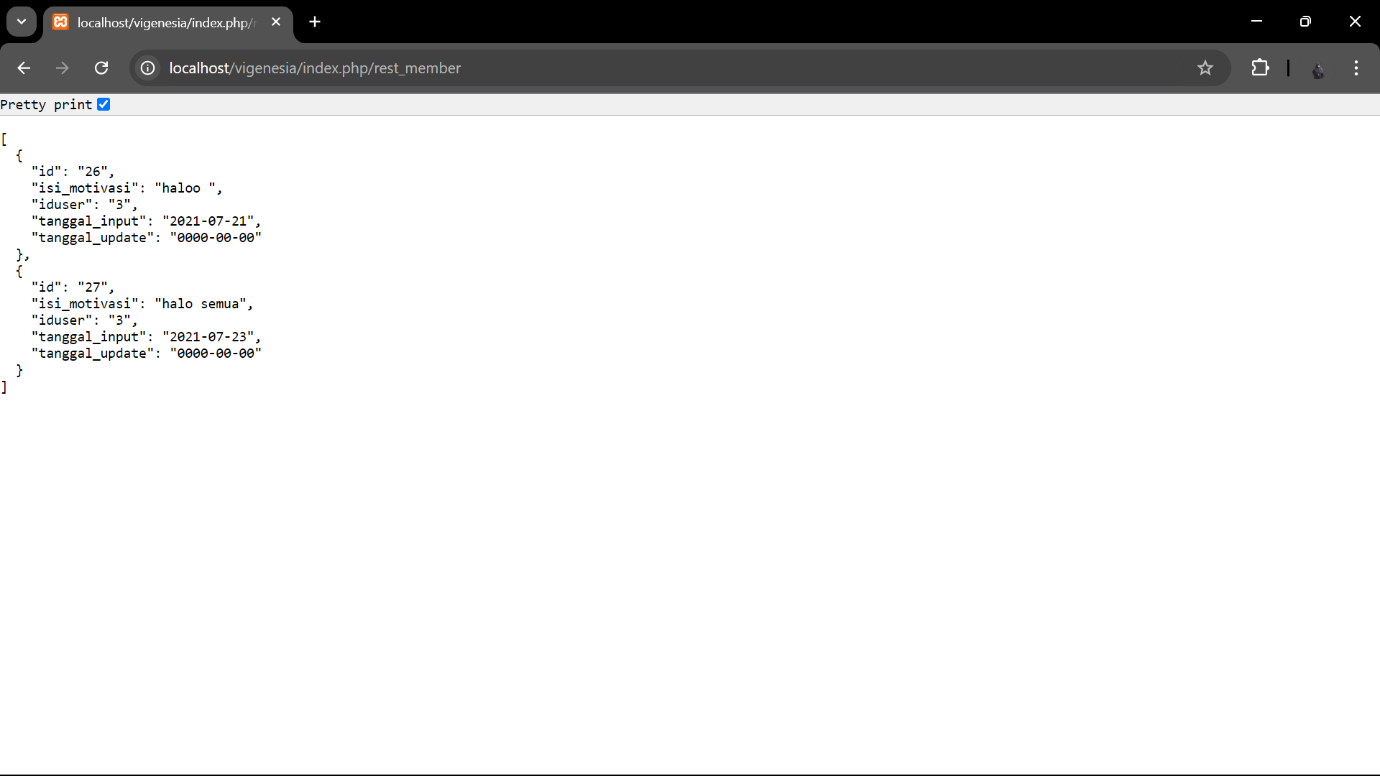
        }

    }

}

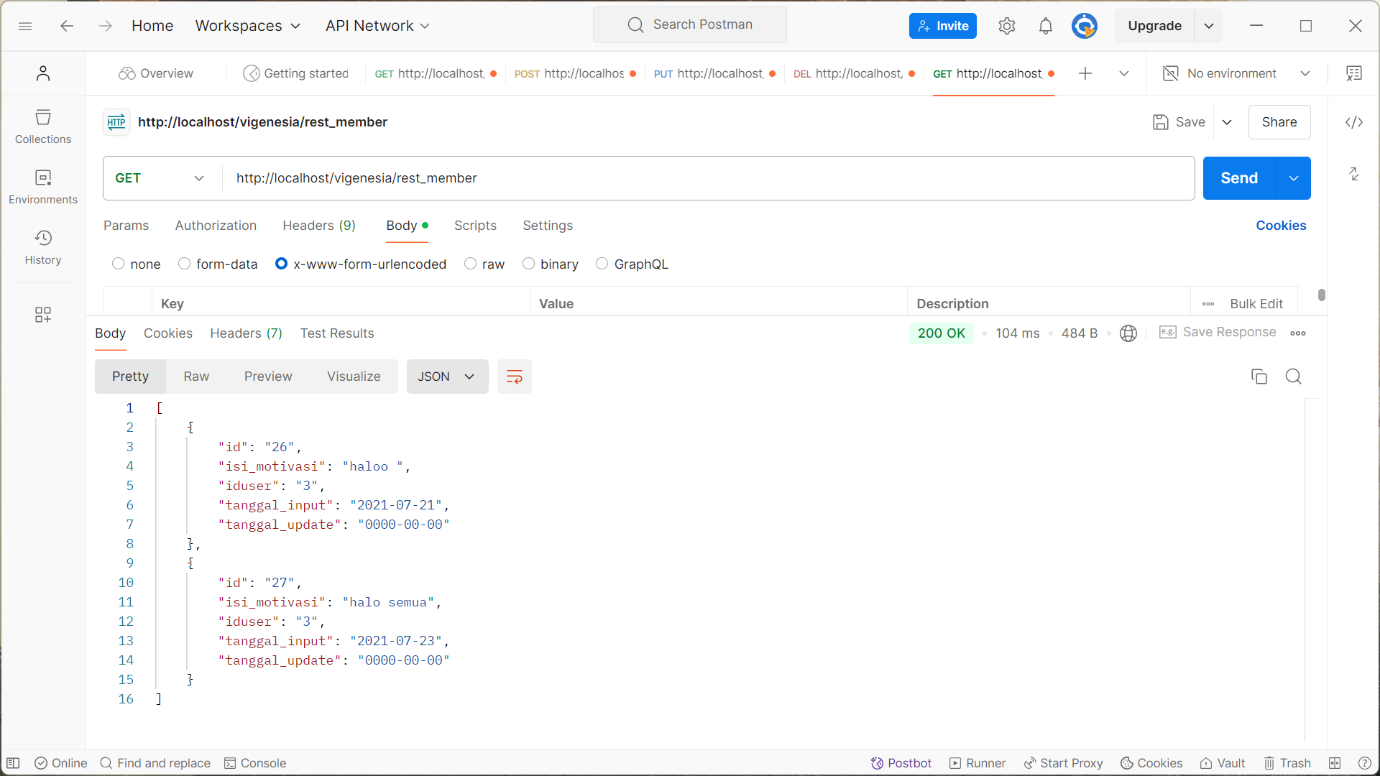
?>

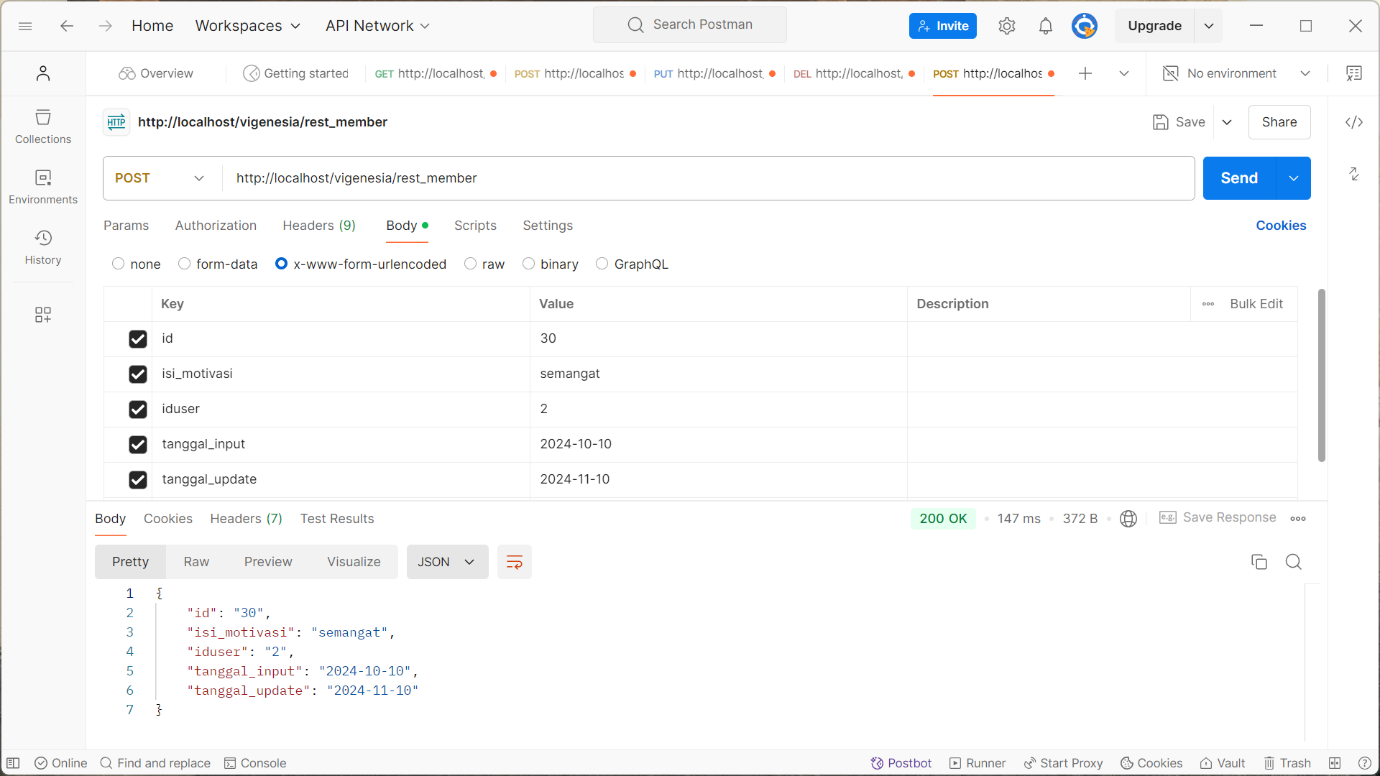
Tampilan Browser



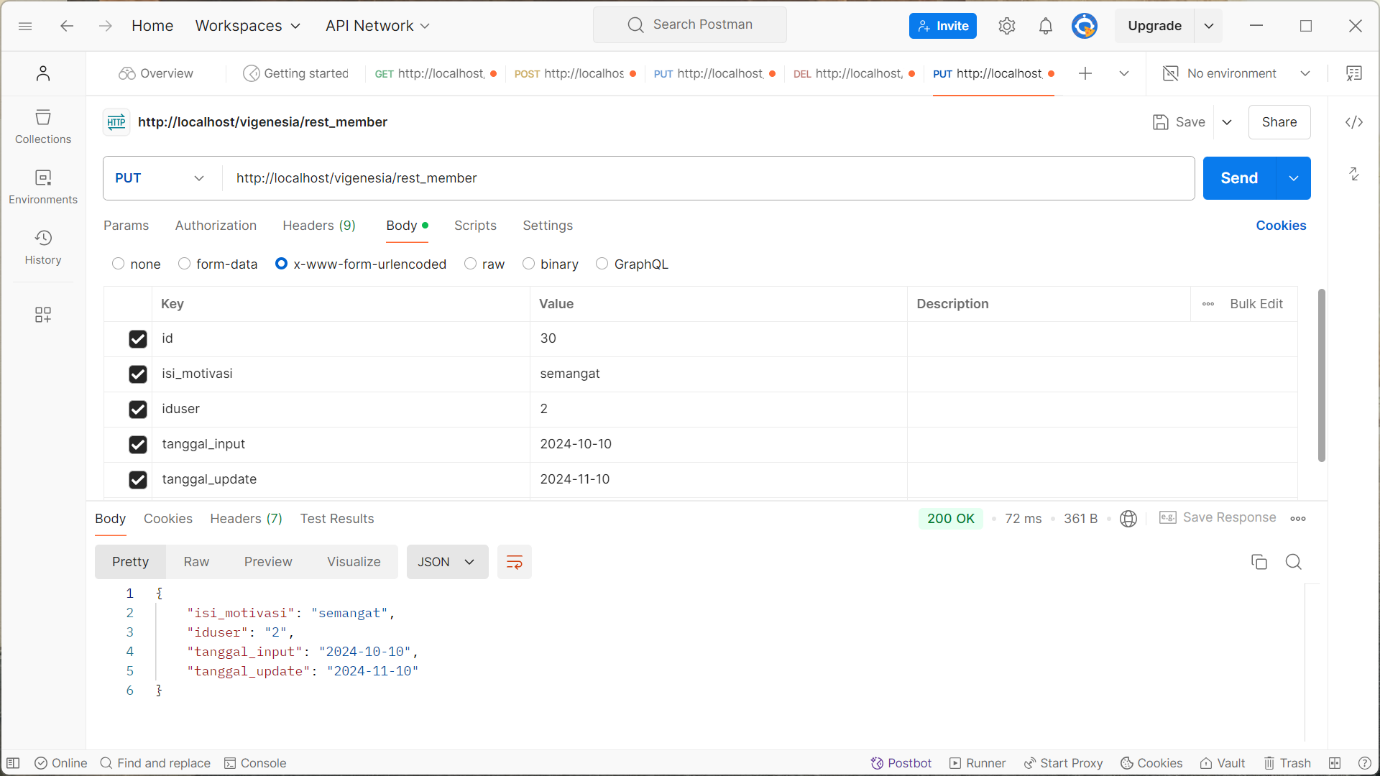
Tampilan Postman

Get

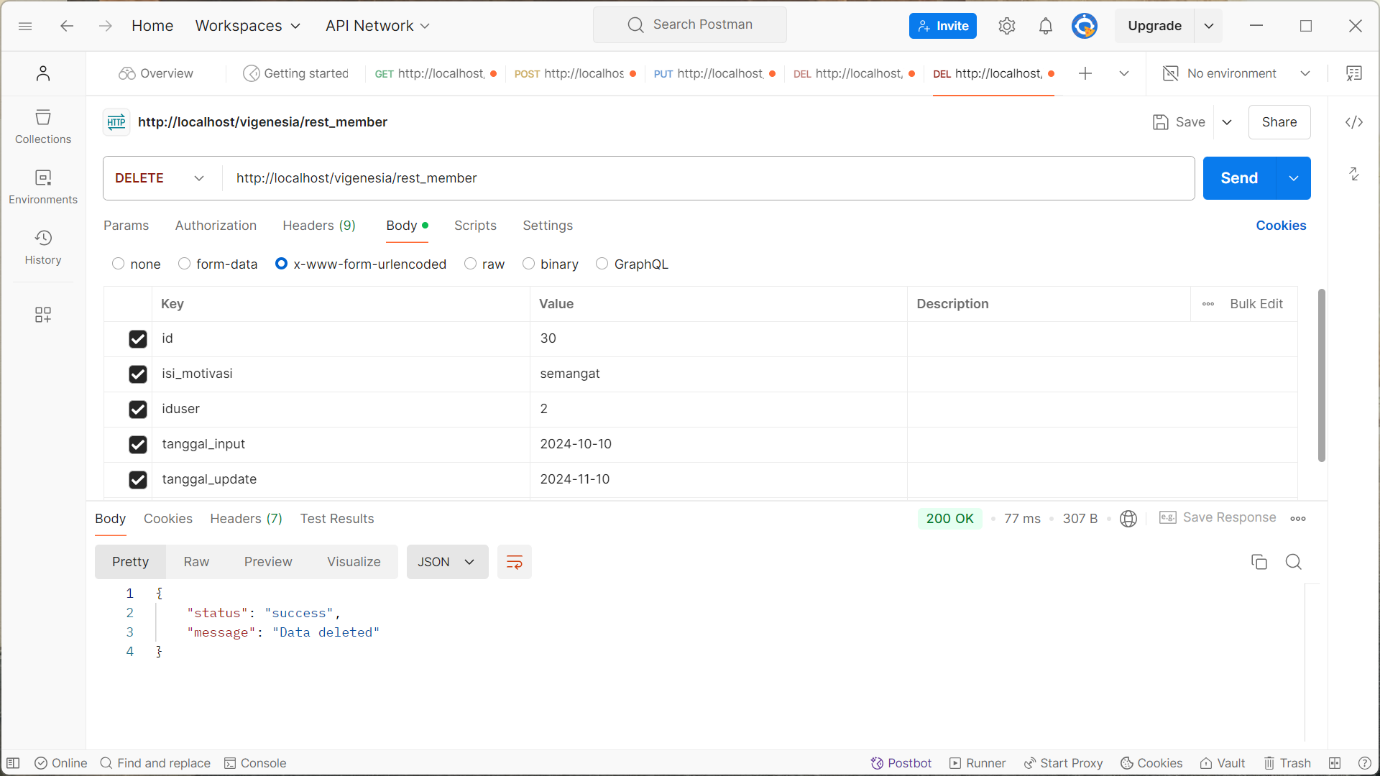
Post



Put

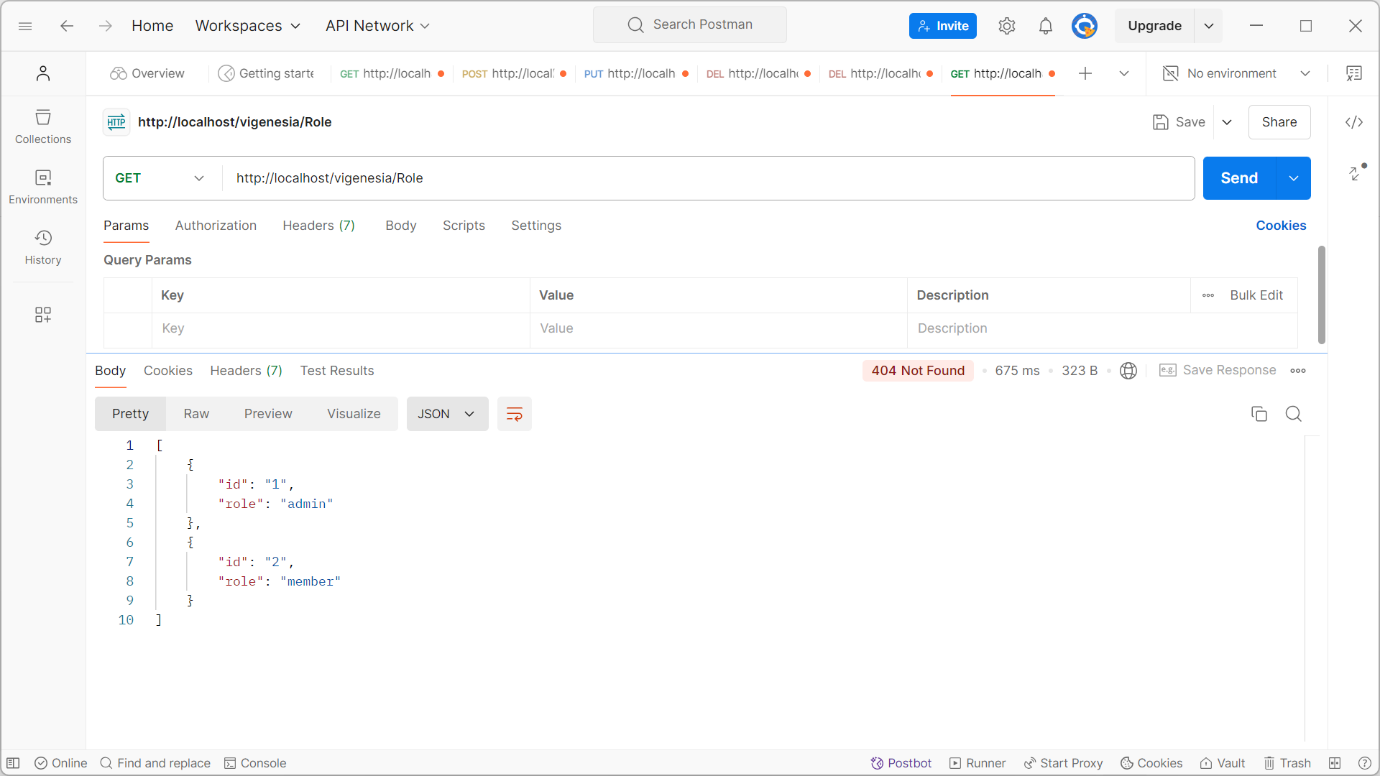


Delete

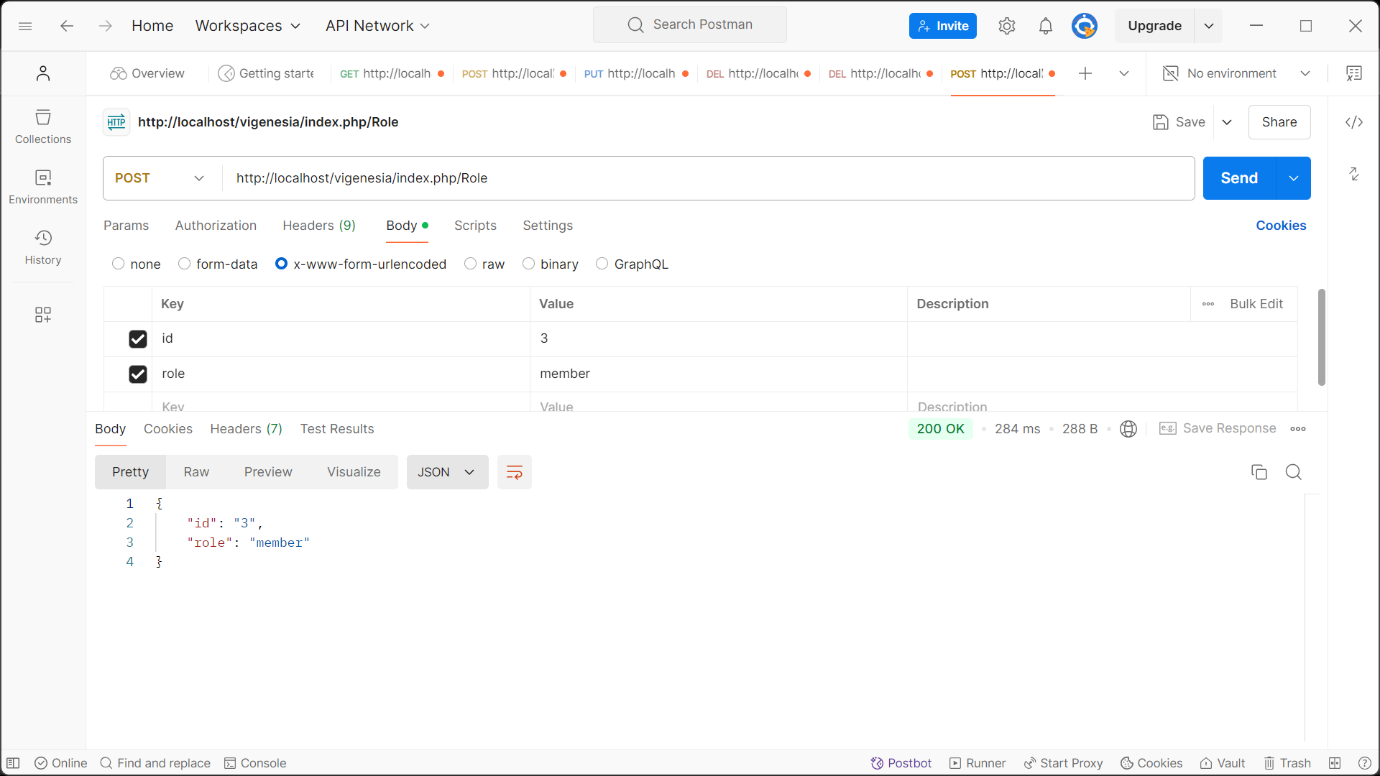


Tabel Role

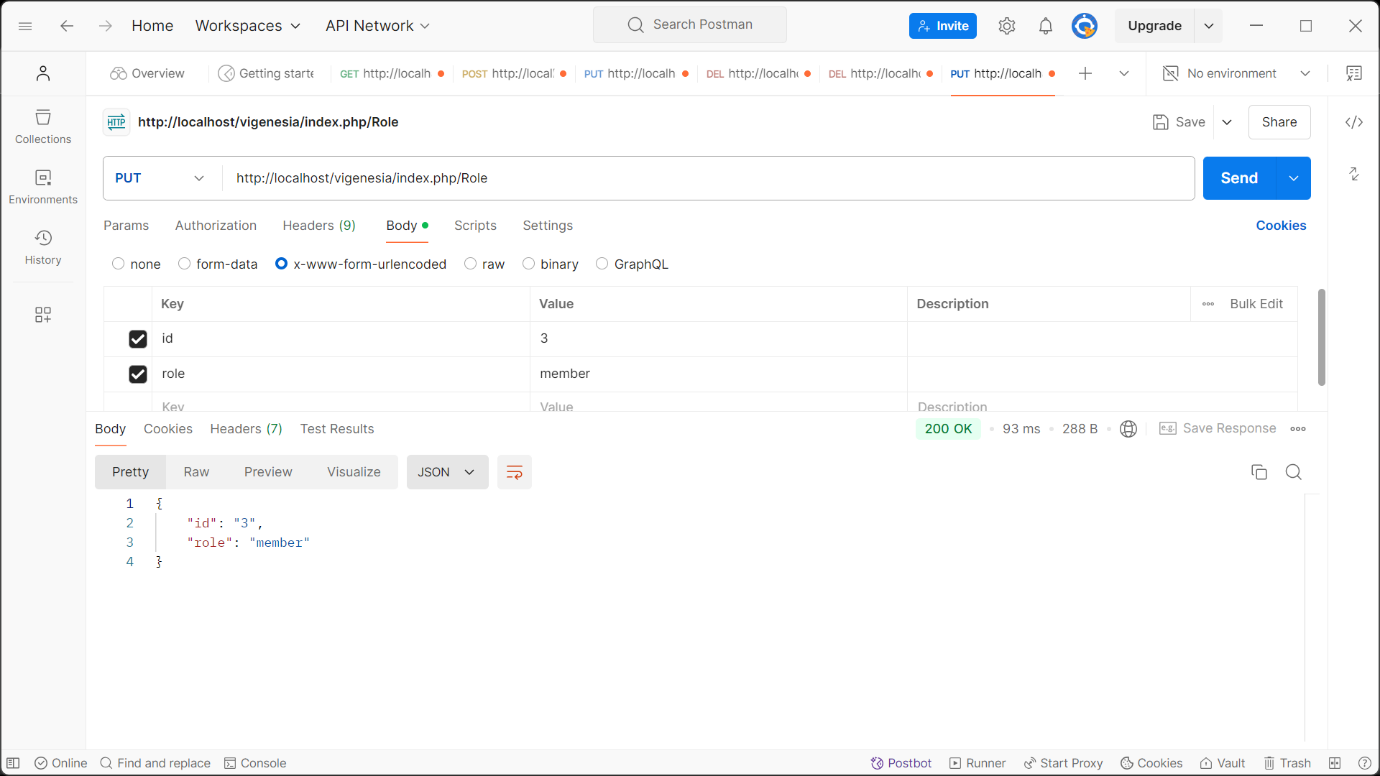
Get



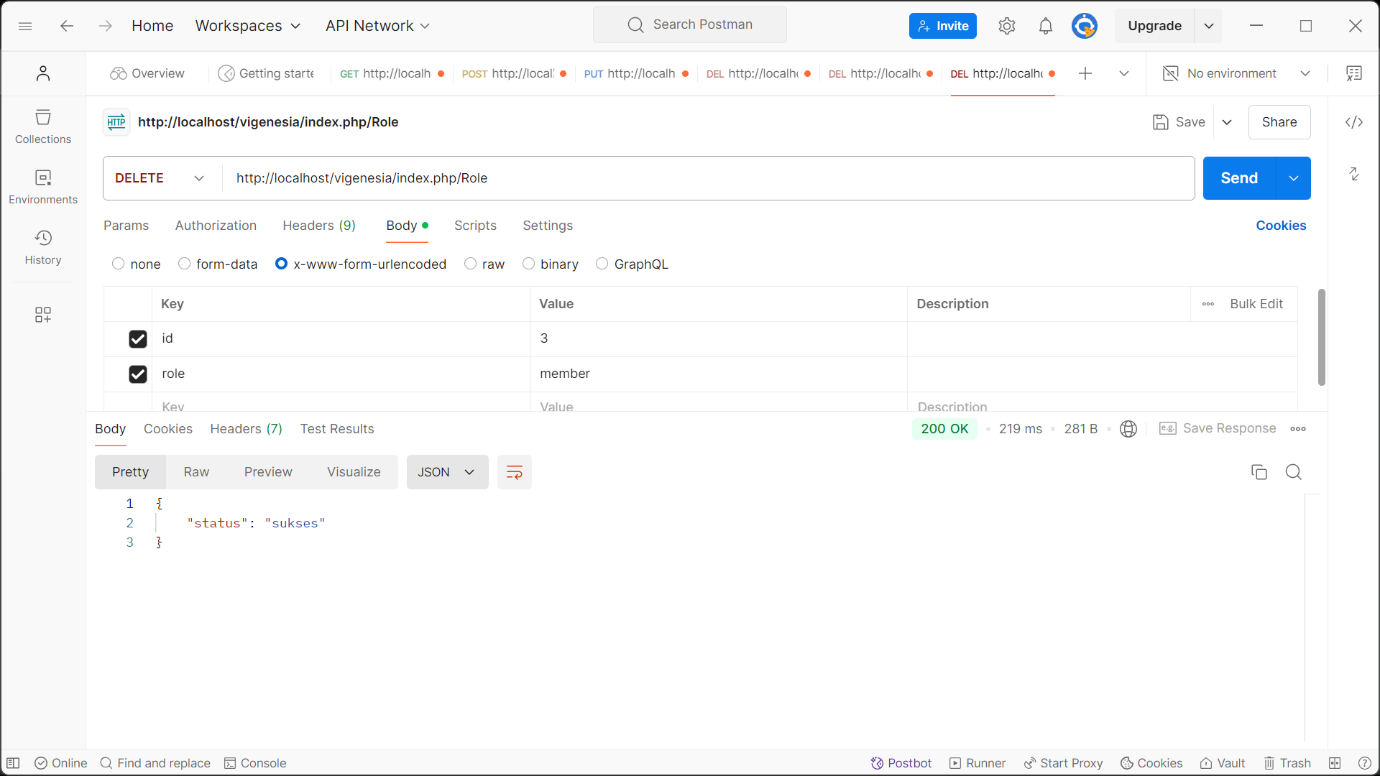
Post



Put



Delete



Codingan Role

<?php

// Dokumentasi Pengerjaan Individu

// NIM : 15220689

// Nama : Sahlah Rizqiyyah

// Kelas : 15.5A.05

// Kampus : UBSI Kaliabang

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

require APPPATH . '/libraries/REST\_Controller.php';

use Restserver\libraries\REST\_Controller;

class Role extends REST\_Controller

{

    function \_\_construct($config = 'rest'){

        parent::\_\_construct($config);

        $this->load->database();

    }

//awal fungsi operasi get

    function index\_get()

    {

// Deskripsi Fungsi :

// ------------------------------------------------------------------

// Script dibawah ini merupakan implementasi dari metode GET

// -------------------------------------------------------------------

    $id = $this->get('id');

    if ($id == '') {

    $member = $this->db->get('role')->result();

    } else {

    $this->db->where('id', $id);

    $member = $this->db->get('role')->result();

    }

    $this->response($member, 404);

}

    function index\_post() {

        $data = array(

            'id' => $this->post('id'),

            'role' => $this->post('role'),

        );

        $insert = $this->db->insert('role', $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'),

            'role' => $this->put('role'),

        );

        $this->db->where('id', $id);

        $update = $this->db->update('role', $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('role');

        if ($delete) {

            $this->response(array('status'=>'sukses'), 200);

        } else {

            $this->response(array('status' => 'gagal', 502));

        }

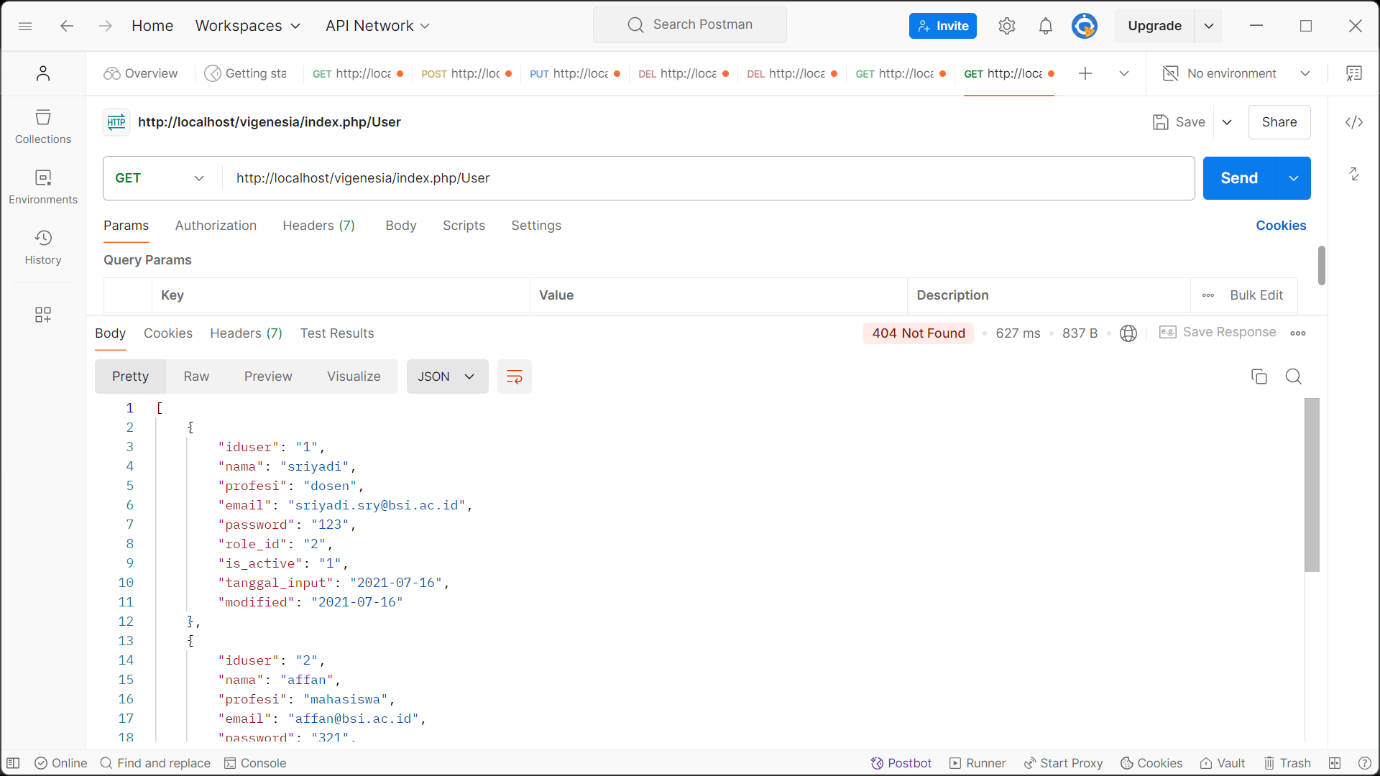
        }

}

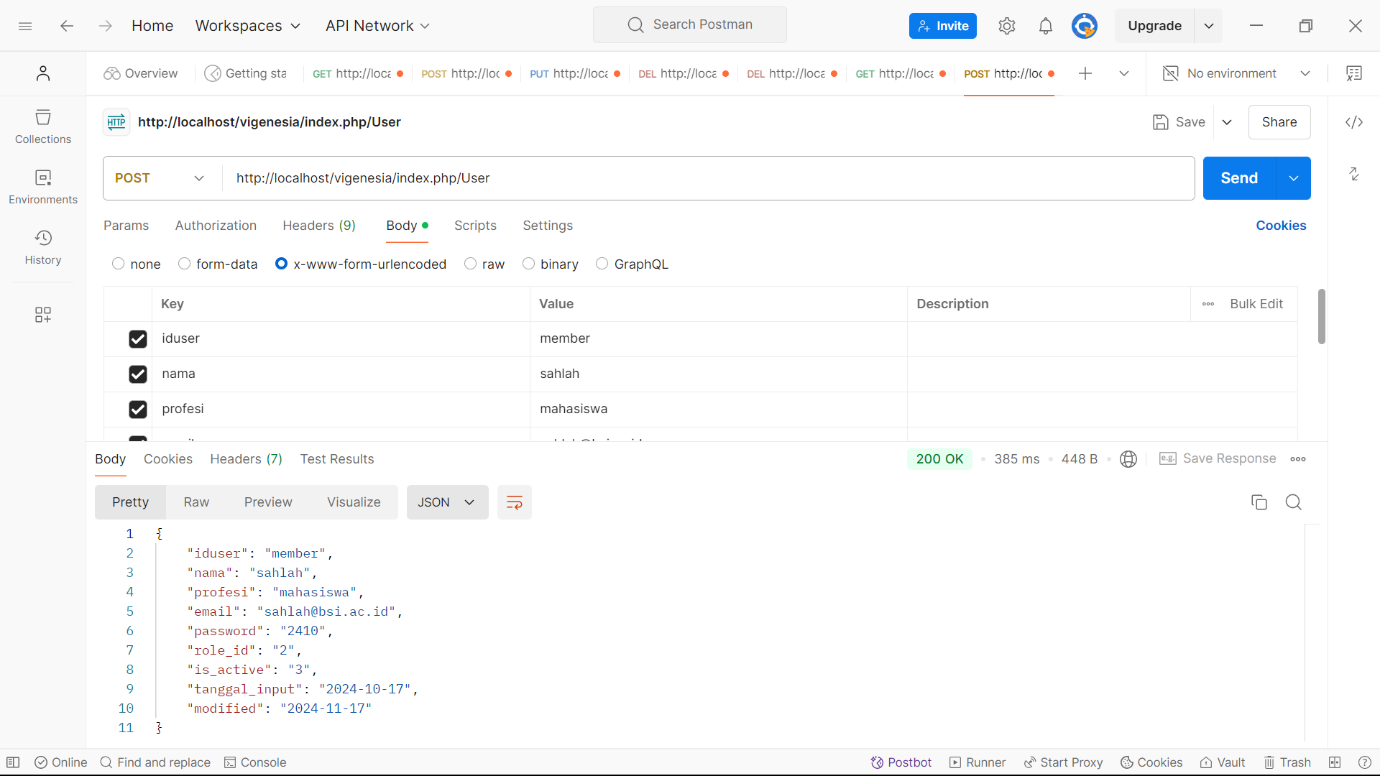
?>

Tabel User

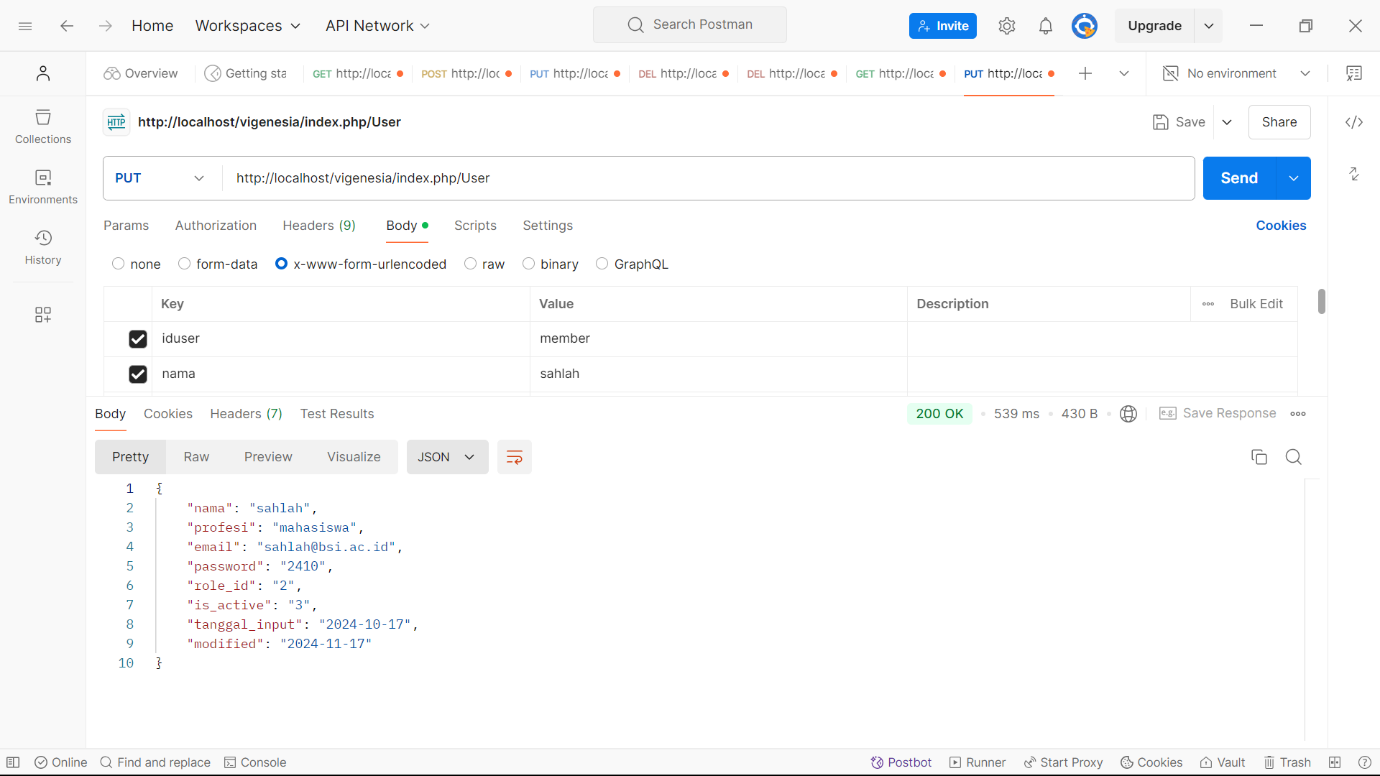
Get



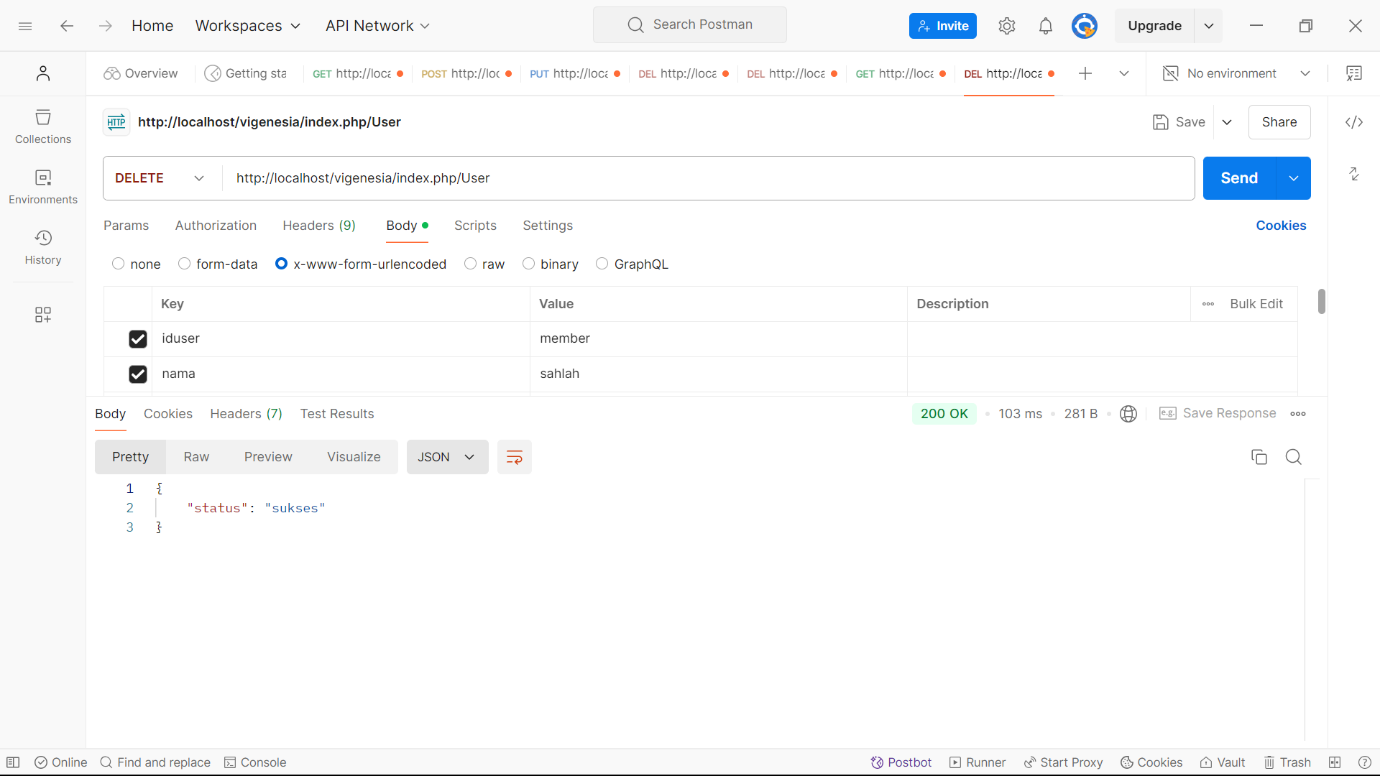
Post



Put



Delete



Codingan User

<?php

// Dokumentasi Pengerjaan Individu

// NIM : 15220689

// Nama : Sahlah Rizqiyyah

// Kelas : 15.5A.05

// Kampus : UBSI Kaliabang

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

require APPPATH . '/libraries/REST\_Controller.php';

use Restserver\libraries\REST\_Controller;

class User extends REST\_Controller

{

    function \_\_construct($config = 'rest') {

        parent::\_\_construct($config);

        $this->load->database();

    }

    // Fungsi untuk operasi GET

    function index\_get() {

        $id = $this->get('iduser');

        if ($id == '') {

            $member = $this->db->get('user')->result();

        } else {

            $this->db->where('iduser', $id);

            $member = $this->db->get('user')->result();

        }

        $this->response($member, 200);

    }

    // Fungsi untuk operasi POST

    function index\_post() {

        $data = array(

            'iduser' => $this->post('iduser'),

            'nama' => $this->post('nama'),

            'profesi' => $this->post('profesi'),

            'email' => $this->post('email'),

            'password' => $this->post('password'),

            'role\_id' => $this->post('role\_id'),

            'is\_active' => $this->post('is\_active'),

            'tanggal\_input' => $this->post('tanggal\_input'),

            'modified' => $this->post('modified'),

        );

        $insert = $this->db->insert('user', $data);

        if ($insert) {

            $this->response($data, 200);

        } else {

            $this->response(array('status' => 'fail'), 502);

        }

    }

    // Fungsi untuk operasi PUT

    function index\_put() {

        $id = $this->put('iduser');

        $data = array(

            'nama' => $this->put('nama'),

            'profesi' => $this->put('profesi'),

            'email' => $this->put('email'),

            'password' => $this->put('password'),

            'role\_id' => $this->put('role\_id'),

            'is\_active' => $this->put('is\_active'),

            'tanggal\_input' => $this->put('tanggal\_input'),

            'modified' => $this->put('modified'),

        );

        $this->db->where('iduser', $id);

        $update = $this->db->update('user', $data);

        if ($update) {

            $this->response($data, 200);

        } else {

            $this->response(array('status' => 'fail'), 502);

        }

    }

    // Fungsi untuk operasi DELETE

    function index\_delete() {

        $id = $this->delete('iduser');

        $this->db->where('iduser', $id);

        $delete = $this->db->delete('user');

        if ($delete) {

            $this->response(array('status' => 'sukses'), 200);

        } else {

            $this->response(array('status' => 'gagal'), 502);

        }

    }

}

?>