

PERTEMUAN KE 4

RESTFUL WEB SERVICE

Deskripsi

Pembahasan meliputi pengenalan *restful web service*, kebutuhan tools, implementasinya dengan mempersiapkan disain database, *setup restserver*, setting koneksi database, implementasi Restful Web Service dengan metode *GET*, *POST*, *PUT*, dan *DELETE*, dan pengujian melalui browser maupun Postman

Tujuan Pembelajaran

Mahasiswa mampu memahami restfull web service, kebutuhan tools, menerapkan layanan web service restfull akses database menggunakan metode *GET*, *POST*, *PUT*, dan *DELETE* dan menguji melalui browser

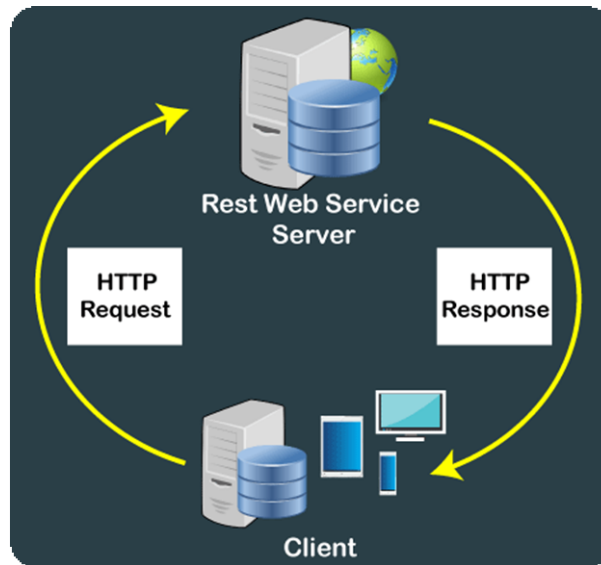
A. Pengenalan Restful Web Service

Salah satu web service yang biasa digunakan yaitu REST atau biasa disebut RESTful Web Service. REST sendiri merupakan singkatan dari *Representational State Transfer*, yaitu suatu gaya arsitektur perangkat lunak dalam menyediakan sumber daya /data (*resources*) pada REST server, serta dapat diakses dan ditampilkan *resources* tersebut untuk dapat digunakan pada REST Client.

Layanan Restful Web Service menggunakan metode HTTP dalam menerapkan konsep arsitektur REST, dimana setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifiers*) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, yang pada umumnya menggunakan JSON atau XML. Berikut beberapa metode HTTP yang umum digunakan dalam arsitektur Restful Web Service.

- **GET**, hanya menyediakan akses pembacaan pada *resource*
- **PUT**, dapat digunakan untuk mengperbaharui *resource* baru
- **DELETE**, dapat digunakan untuk menghapus *resource*
- **POST**, dapat digunakan untuk memperbarui *resource* yang ada atau membuat *resource* baru

Berikut gambaran ini cara kerja Restful Web Service



Sumber: <https://www.javatpoint.com/soap-and-rest-web-services>

Sebuah Client mengirimkan atau meminta sebuah data melalui HTTP Request ke Server, Server merespon melalui HTTP Response. Komponen dari HTTP Request bisa terdiri dari HTTP Method (GET, POST, DELETE, PUT, dan yang lainnya), *Uniform Resource Identifier* (URI) untuk mengidentifikasi lokasi *resource* pada server. Sementara komponen HTTP Response bisa terdiri dari *Status/Response Code* yang mengindikasikan status server terhadap *resource* yang direquest (misal : 404, artinya *resource* tidak ditemukan dan 200 response OK)

B. Kebutuhan Tools

Dalam implementasi RESTful Web Service pada modul pembelajaran ini dibutuhkan beberapa hal yang harus dipersiapkan pada PC/laptop, diantaranya yaitu:

1. Xampp sebagai web server, visual studio code/Sublime/atom sebagai editor (*pastikan dilaptop mahasiswa sudah terinstall tools tersebut, kelanjutan dari semester IV*)
2. Codeigniter dan library REST server yang diperlukan dapat diunduh di <https://github.com/chriskacerguis/codeigniter-restserver> untuk versi terbaru, versi yang digunakan disini adalah <https://github.com/ardisaurus/ci-restserver>.

C. Implementasi Restful Web Service

Dibawah ini akan dijabarkan contoh implementasi Restful Web Service menggunakan Framework Codeigniter. Dalam konsep MVC Codeigniter, *controller* adalah titik pusat logika dimana dipanggil ketika pengguna membuat permintaan dan kemudian berdasarkan

logika di *controller* itu mengambil data dan output dapat dilihat atau ditampilkan. Namun, dalam penerapan RESTful diperlukan library REST_Controller sebagai pengaturan logikanya.

1. Konfigurasi Database

- Buat database baru dengan nama **db_member**:
- Dalam database db_member, buat sebuah tabel dengan nama **member**.

```
CREATE DATABASE db_member;
```

N a m a <i>F i e l d</i>	T y p e <i>I n t</i>	S i z e 1 1	K e t <i>A u t o - I n c r e m e n t N o t N u l l , P r i m a r</i>
--	--	--------------------------------	---

			Key
nama_member	VARCHAR(40)	40	Not Null
Email	VARCHAR(35)	35	Not Null
no_telp	VARCHAR(16)	16	Not Null

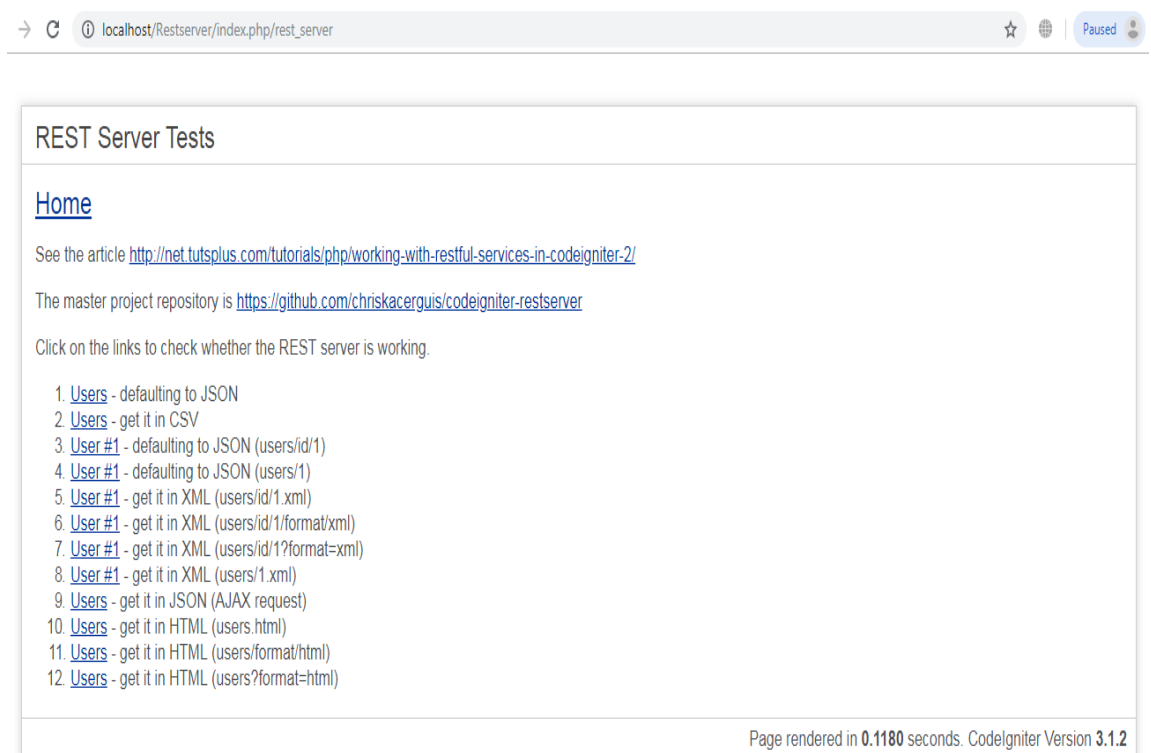
```
CREATE TABLE IF NOT EXISTS `member` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama_member` varchar(40) NOT NULL,
  `email` varchar(35) NOT NULL,
  `no_telp` varchar(16) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- Entry data dibawah ke dalam tabel member

```
INSERT INTO `member` (`nama_member`, `email`, `no_telp`) VALUES
('Tiwi Pertiwi','tiwi@gmail.com','111122223333'),
('Hanna Putri','hanna@gmail.com','555566667777'),
('Mus Dalifa','mus@gmail.com','888899990000');
```

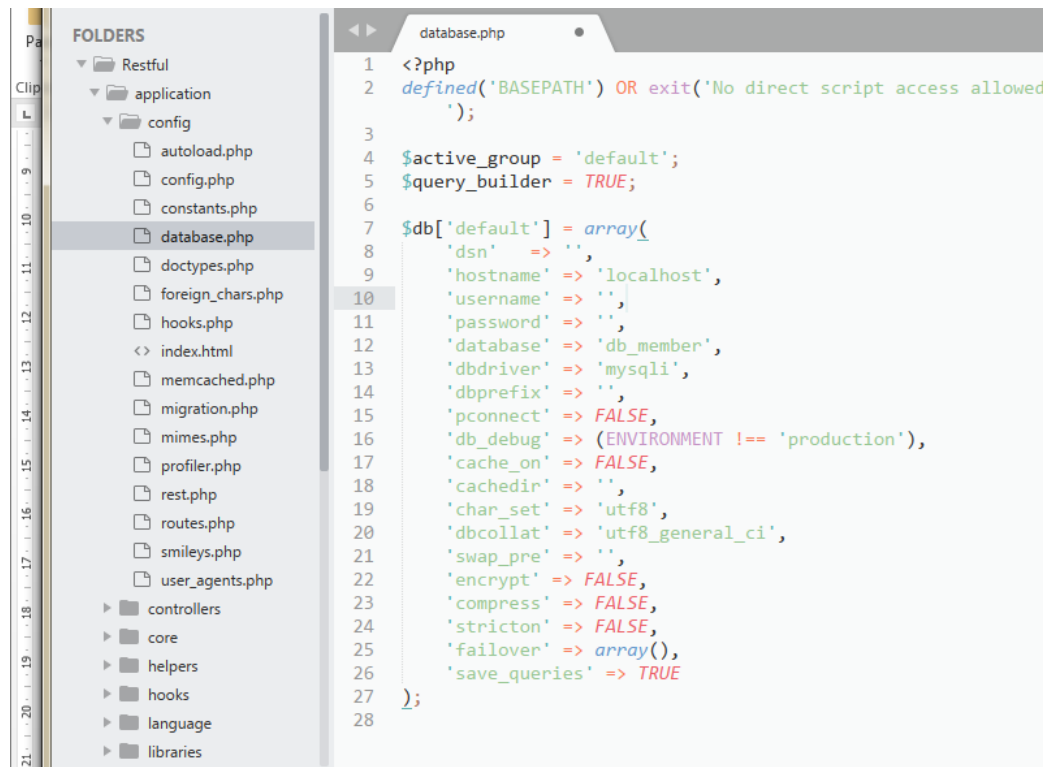
2. Persiapan Project RESTful Web Service

- Pastikan Folder master Codeigniter yang disertai library REST Server telah ada pada folder C:/xampp/htdocs dilaptop masing-masing. Dengan nama folder diubah dan diberi nama **Restserver**.
- Pastikan service Apache dan Mysql pada Xampp Control panel telah aktif.
- Jalankan Browser, lalu ketikan alamat : **localhost/Restserver/index.php/rest_server**
- Hingga muncul tampilan dibawah ini untuk memastikan project Restserver dan library REST Server berhasil running.



3. Setting File Koneksi Database

- Jalankan Editor Sublime Text, lalu buka Folder Project **Restful**.
- Buka File database.php pada folder application/config/database.php
- Setting **database** menjadi **db_member**



4. Implementasi Metode GET

Untuk melakukan implementasi Restful Web Service dengan metode GET, POST, PUT, dan DELETE, terlebih dahulu buat controller baru dengan nama “**Rest_member.php**” pada folder controller. Kemudian, untuk metode GET, ketikkan script dibawah ini:

```
<?php
// Dokumentasi Pengerjaan Individu
// NIM      : .....
// Nama     : .....
// Kelas    : .....
// Kampus   : .....

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();
    }

    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);
    }

}

?>
```

Penjelasan Code :

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

Script diatas menunjukkan pemanggilan Controller REST_Controller.php yang ada pada libraries untuk web service. *(setiap pembuatan controller baru, wajib disertakan deklarasi controller library REST web service)*

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

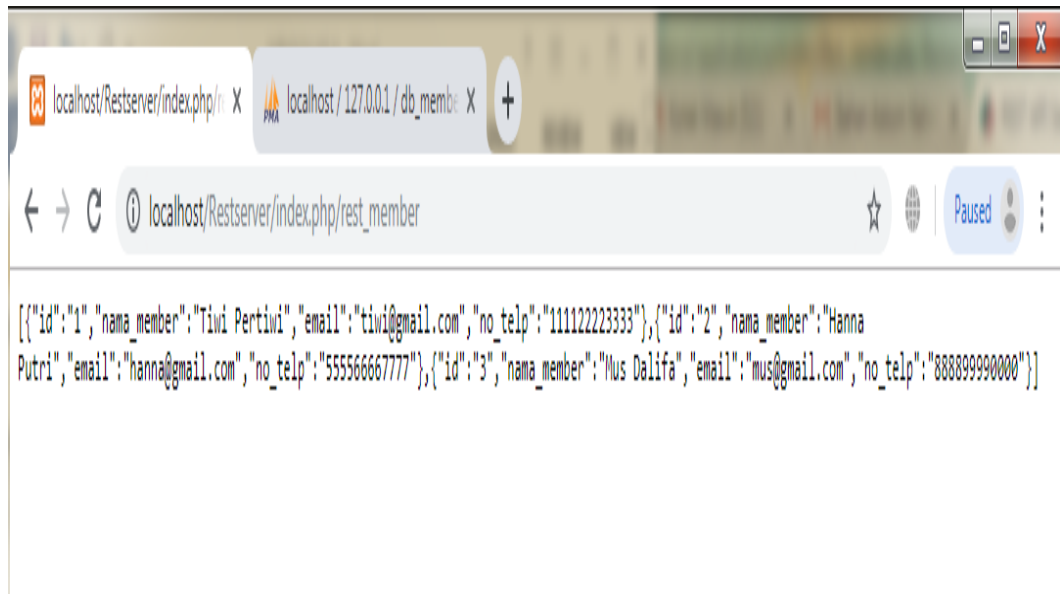
Script diatas menunjukkan function dengan parameter \$config yang diberi nilai 'rest'. Pemberian nilai rest merujuk pada function construct file REST_Controller(ada di folder libraries), jika pengaturasn REST web service ada, maka file database.php akan dijalankan.

```
// 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);
```

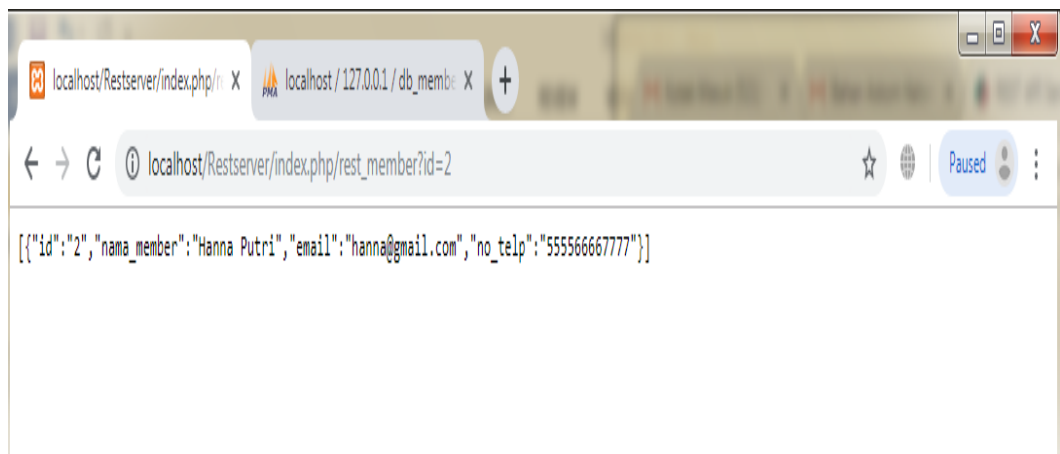
Script diatas menunjukkan pembuatan fungsi get, yang akan memeriksa apakah ada property “id” pada address bar, jika tidak ada maka data akan ditampilkan semua tanpa ada seleksi.

Tampilan uji coba pada browser :

- Ketika tidak ada property id pada addrees bar :
localhost/Restserver/index.php/rest_member



- Ketika ada property id pada address bar :
localhost/Restserver/index.php/rest_member?id=2



5. Implementasi Metode POST

Buka controller “**Rest_member.php**” pada folder controller yang telah berisi function `index_get`. Kemudian, untuk metode POST, tambahkan script dibawah ini


```
<?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 {

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

    function index_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, 200);
    }

    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));
        }
    }
}
?>
```

Penjelasan Code :

```
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));  
    }  
}
```

Script diatas menunjukkan pembuatan fungsi POST, yang digunakan untuk mengirimkan data baru dari client ke REST server. Pada kasus diatas function `index_post` pada controller `rest_member.php` untuk menambahkan (*insert*) data pada tabel member dengan atribut `id`, `nama_member`, `email` dan `no_telp`.

6. Implementasi Metode PUT

Buka controller “**Rest_member.php**” pada folder controller yang telah berisi beberapa function. Kemudian, untuk metode PUT, tambahkan script dibawah ini

```
function index_put() {  
    $id = $this->put('id');  
    $data = array(  
        'id' => $this->post('id'),  
        'nama_member' => $this->post('nama_member'),  
        'email' => $this->post('email'),  
        'no_telp' => $this->post('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));  
    }  
}  
}  
?>
```

Penjelasan Code :

```
function index_put() {  
    $id = $this->put('id');  
    $data = array(  
        'id' => $this->post('id'),  
        'nama_member' => $this->post('nama_member'),  
        'email' => $this->post('email'),  
        'no_telp' => $this->post('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));  
        }  
    }  
}
```

Script diatas menunjukkan pembuatan fungsi PUT, yang digunakan untuk memperbaharui data (*update*) dari client ke REST server. Pada kasus diatas function `index_put` pada controller `rest_member.php` untuk memperbaharui (*update*) data pada atribut `nama_member`, `email` atau `no_telp` dengan `id` yang ditunjuk pada tabel `member`. Untuk pengujian metode PUT akan dibahas pada bahasan berikutnya mengenai tools postman.

7. Implementasi Metode DELETE

Buka controller “**Rest_member.php**” pada folder controller yang telah berisi beberapa function. Kemudian, untuk metode DELETE, tambahkan script dibawah ini

```
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));  
    }  
}  
}  
?>
```

Penjelasan Code :

```
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));  
    }  
}
```

}
Script diatas menunjukkan pembuatan fungsi DELETE, yang digunakan untuk menghapus data (*delete*) dari client ke REST server. Pada kasus diatas function `index_delete` pada controller `rest_member.php` untuk menghapus (*delete*) data pada atribut `nama_member`, `email` atau `no_telp` dengan id yang ditunjuk pada tabel `member`.

D. Pengujian Restful Web Service Menggunakan Aplikasi Postman

1. Sejarah Singkat Postman

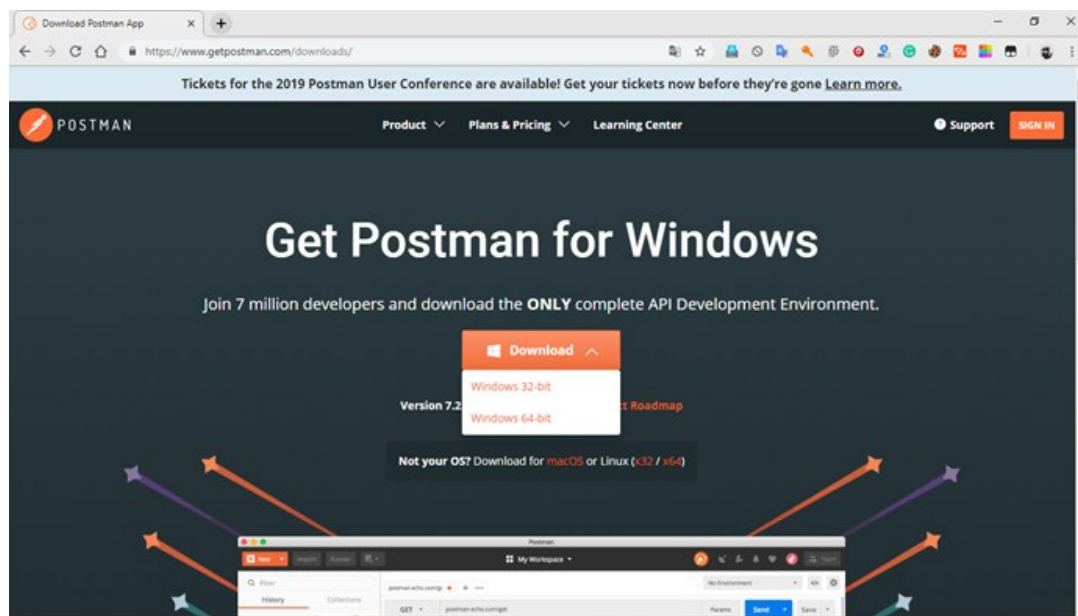
Postman pertama kali dibuat sebagai proyek sampingan yang dikerjakan oleh Abhinav Asthana untuk mengatasi tantangan dalam pengujian API, kemudian postman mulai populer pada tahun 2012, yaitu ketika Abhinav selaku CEO dan co-founder Postman dengan ditemani Ankit Soboti dan Abhijit Kane. mengunggah perojek yang dia kerjakan ke Chrome Web Store artinya aplikasi postman masih dalam bentuk plug-in dari web browser Chrome. Namun akhirnya postman berhasil release dalam bentuk aplikasi yang berdiri sendiri dengan user interface yang mirip dengan tampilan web browser. Saat ini postman sudah digunakan oleh 7 juta pengembang dan 300 perusahaan.

2. Pengertian Postman

Postman adalah sebuah aplikasi fungsinya adalah sebagai REST Client atau istilahnya adalah aplikasi yang digunakan untuk melakukan uji coba REST API yang telah kita buat. Postman ini merupakan tools wajib bagi para developer yang bergerak pada pembuatan API, fungsi utama postman ini adalah sebagai GUI API Caller Pemanggil. namun sekarang postman juga menyediakan fitur lain yaitu Sharing Collection API for Documentation (free), Testing API (free), Realtime Collaboration Team (paid), Monitoring API (paid), Integration (paid) detailnya silahkan dicek disini. Dulu awal pertama kali postman muncul sebagai add on dari Chrome namun sekarang sudah menjadi aplikasi sendiri. Jika kalian sedang membuat API sangat direkomendasikan untuk menggunakan Postman untuk testing API yang kalian buat.

3. Instalasi Postman

Postman tersedia sebagai aplikasi asli untuk sistem operasi macOS, Windows (32-bit dan 64-bit), dan Linux (32-bit dan 64-bit). Untuk mendapatkan aplikasi Postman, dapat diunduh pada website resminya yaitu getpostman.com



atau bisa di download melalui link berikut:

- Linux

https://drive.google.com/file/d/1I-U-7sqAowZhL5dosM3_5B0XS9QMLMMa/view?usp=sharing

- Mac



https://drive.google.com/file/d/1nTXd-9cPPXuq4JLx_OBUr9jBWld61bML/view?usp=sharing

- Windows

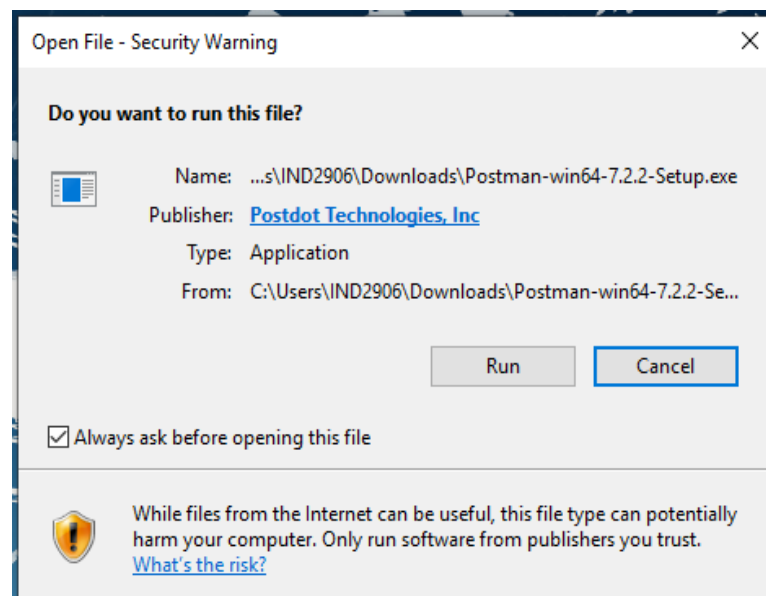
[Versi 32 Bit](#)

[Versi 64 Bit](#)

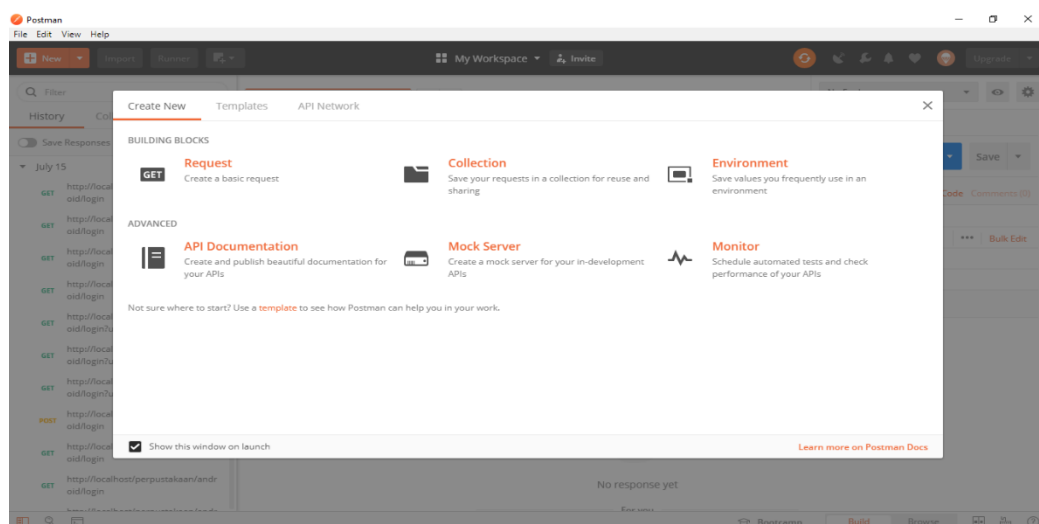
Setelah postman selesai didownload, selanjutnya jalankan paket instalasi postman dengan cara double klik.

 Postman-win32-4.10.7-Setup.exe	15/05/2017 6:34	Application	57.553 KB
 Postman-win64-4.10.7-Setup.exe	15/05/2017 6:35	Application	67.509 KB

Pilih Run jika muncul popup seperti berikut:



Kemudian tunggu sampai Postman nya terbuka Seperti ini:



Selesai, postman sudah terinstal pada perangkat anda.

4. Implementasi Pengujian menggunakan Postman

Pengujian Restful Web Service menggunakan postman ini perlu ada beberapa persiapan terlebih dahulu. Persiapan yang diperlukan adalah sebagai berikut:

1. Webserver seperti Xampp, Wampp, atau lainnya. Jalankan
2. Konfigurasi Database

Database yang akan digunakan adalah database yang sudah dibuat pada pertemuan sebelumnya yaitu menggunakan database **db_member**

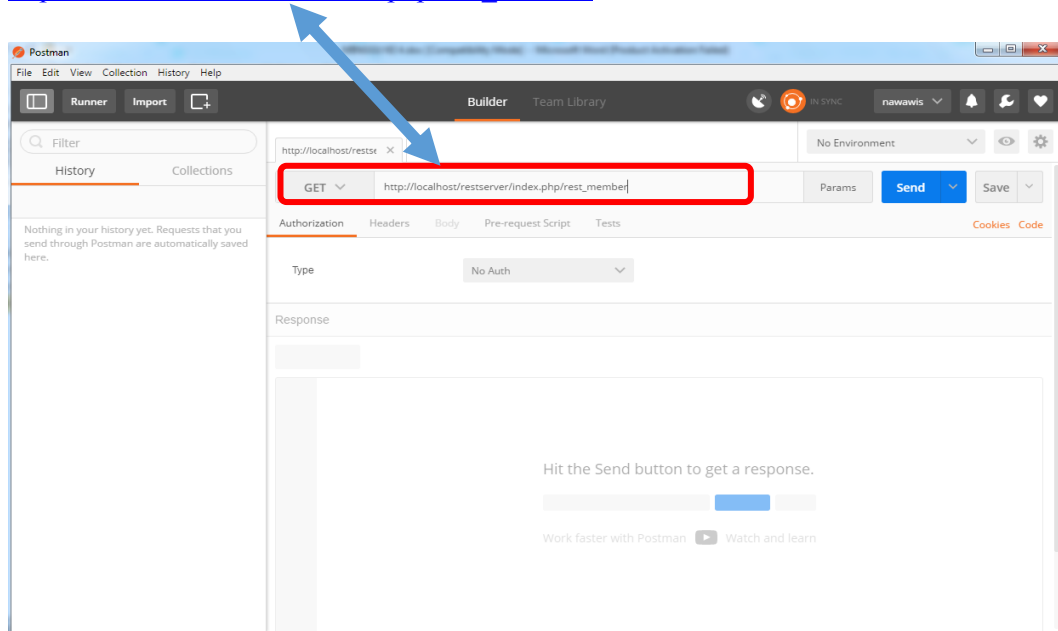
3. Proyek program yang ingin diuji

Proyek yang akan diuji dalam hal ini adalah proyek yang sudah dibuat pada pertemuan minggu sebelumnya juga yaitu tentang restful pendaftaran member.

Setelah persiapan selesai barulah kita akan menguji dengan postman. Langkahnya sebagai berikut:

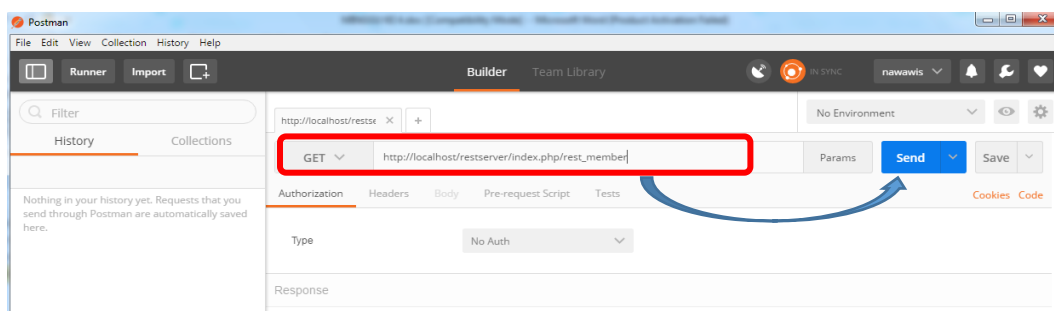
1. Buka aplikasi postman yang sudah terinstal
2. Paste url pada inputan pada postman. Seperti gambar berikut:

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



a. Pengujian GET

1. pilih GET terlebih dahulu, lalu klik tombol Send



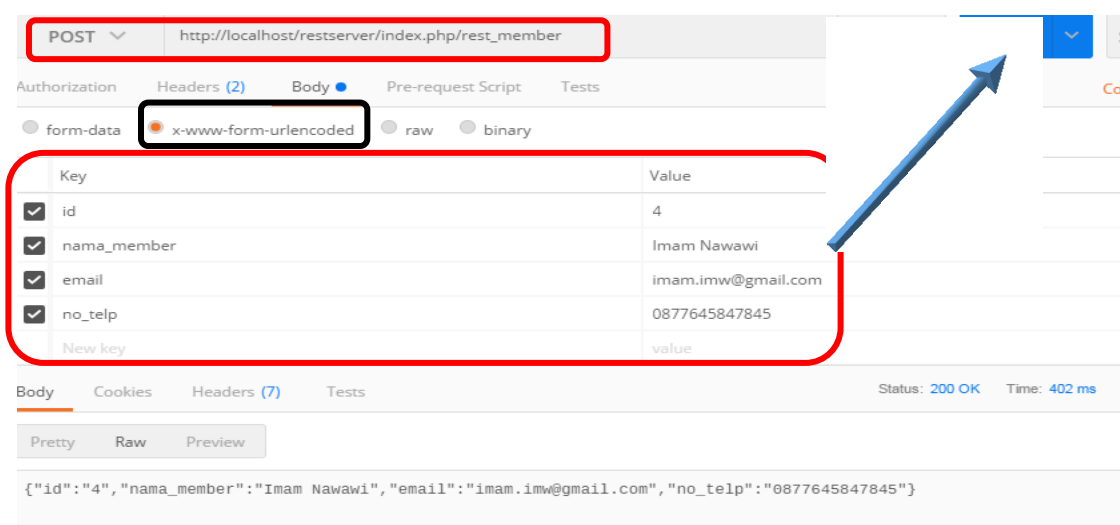
2. Hasilnya seperti berikut:

```
Pretty Raw Preview
```

```
[{"id": "1", "nama_member": "Tiwi Pertiwi", "email": "tiwi@gmail.com", "no_telp": "11112223333"}, {"id": "2", "nama_member": "Hanna Putri", "email": "hanna@gmail.com", "no_telp": "55556666777"}, {"id": "3", "nama_member": "Mus Dalifa", "email": "mus@gmail.com", "no_telp": "88889999000"}]
```

b. Pengujian POST

1. Untuk pengujian POST, maka pilih jenis pengujian POST
2. Setelah pengujian dipilih, selanjutnya klik body yang ada dibagian bawah inputan url di atas, kemudian pilih x-www.form-urlencoded.
3. Selanjutnya isi key sesuai dengan field-field yang ada pada tabel yang digunakan, sedangkan value diisi bebas. Lalu klik tombol send
4. Hasilnya pengujian POST seperti berikut:



c. Pengujian PUT

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

d.

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

DELETE http://localhost/restserver/index.php/rest_member Params Send

Authorization Headers (2) Body Pre-request Script Tests

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value
<input checked="" type="checkbox"/> id	1
<input checked="" type="checkbox"/> nama_member	Tiwi Pertiwi
<input checked="" type="checkbox"/> email	tiwi@gmail.com
<input checked="" type="checkbox"/> no_telp	111122223333
New key	value

Body Cookies Headers (7) Tests Status: 200 OK Time: 1604 ms

Pretty Raw Preview

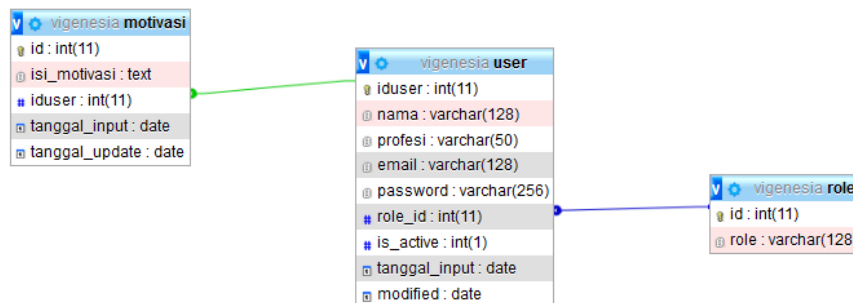
```
{"status": "berhasil"}
```

E. Persiapan Aplikasi ViGeNesia

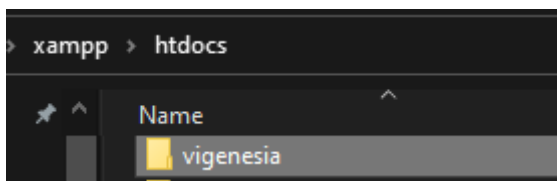
ViGeNesia adalah hasil pemikiran dari diskusi dengan mahasiswa Universitas Bina Sarana Informatika untuk membangun startup, satu ide produknya adalah **ViGeNesia** (Visi Generasi Indonesia). Mimpi besar **ViGeNesia** adalah aplikasi ini bisa menyatukan generasi muda Indonesia dari sabang sampai merauke dalam merajut nasionalisme, persahabatan, saling memotivasi, mengedukasi, berbagi cerita dan cita-cita. **ViGeNesia** akan dikembangkan secara bertahap mulai menjadi *case study* di mata kuliah Teknologi Web Service, dikembangkan oleh mahasiswa sebagai proyek (*projct base*), dan dalam jangka panjang akan

menjadi icon nasional melalui google playstore dan mendapatkan support dari Kemendikbud maupun masyarakat luas.

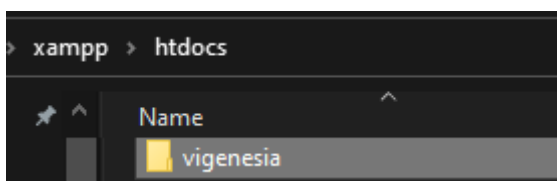
Sebagai tahap awal, **ViGeNesia** disusun berdasarkan proses bisnis yang sangat sederhana, generasi muda Indonesia dapat mempublikasikan atau memberikan sebuah semangat/motivasi melalui aplikasi ViGeNesia. Pesan motivasi tersebut akan menjadi motivasi bagi generasi yang lain. Berdasarkan ide kecil tersebut diperoleh struktur data yang dibutuhkan untuk mensupport manajemen data **ViGeNesia**. Berikut struktur data yang diwakilkan menggunakan normalisasi



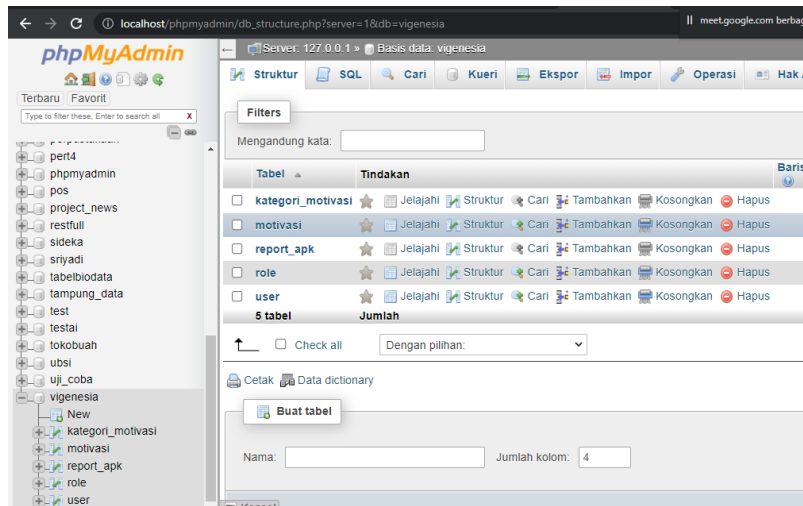
1. Kebutuhan database dan resful API **ViGeNesia** mahasiswa unduh melalui link <https://tinyurl.com/DB-API-vigenesia>
2. Extrak file zip tersebut dan letakan pada httdoc



3. Aktifkan phpmyadmin, create database baru : **vigenesia**
4. Lakukan import **vigenesia.sql** (Temukan file **vigenesia.sql** pada folder DB pada folder **vigenesia**)



5. Setelah sukses import, pastikan stuktur database **vigenesia** seperti pada gambar dibawah



6. Mahasiswa dapat melakukan explore struktur dari masing-masing tabel role, user, dan motivasi.

Role :

#	Nama	Jenis	Penyortiran	Atribut	Tak Terilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	role	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

☐ Pilih Semua Dengan pilihan: Jelajahi Ubah Hapus Utama Unik Indeks Teks penuh

[Remove from central columns](#)

[Cetak](#) [Usulkan struktur tabel](#) [Lacak tabel](#) [Move columns](#) [Normalisasi](#)

User :

#	Nama	Jenis	Penyortiran	Atribut	Tak Terilai	Bawaan	Komentar	Ekstra	Tindakan
1	iduser	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	profesi	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	email	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	password	varchar(256)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
6	role_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
7	is_active	int(1)			Tidak	Tidak ada			Ubah Hapus Lainnya
8	tanggal_input	date			Tidak	Tidak ada			Ubah Hapus Lainnya
9	modified	date			Tidak	Tidak ada			Ubah Hapus Lainnya

☐ Pilih Semua Dengan pilihan: Jelajahi Ubah Hapus Utama Unik Indeks Teks penuh Add to cent

Motivasi :



The screenshot shows a database management interface with a table structure view. The table has five columns: #, Nama, Jenis, Penyortiran, Atribut, Tak Ternilai, Bawaan, Komentar, Ekstra, and Tindakan. The rows are as follows:

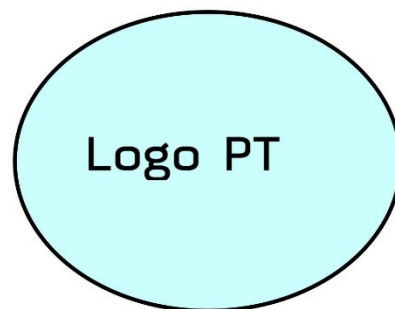
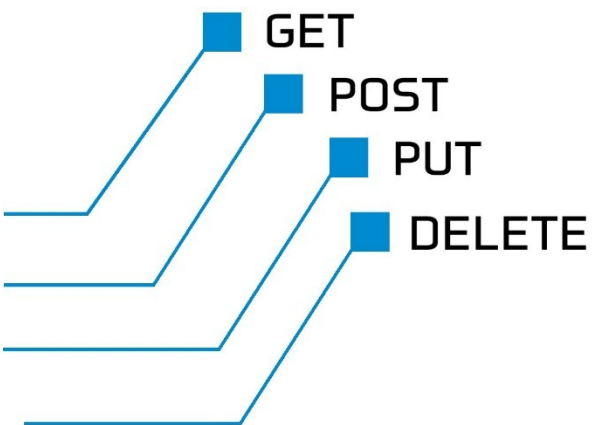
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	isi_motivasi	text	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	iduser	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
4	tanggal_input	date			Tidak	Tidak ada			Ubah Hapus Lainnya
5	tanggal_update	date			Tidak	Tidak ada			Ubah Hapus Lainnya

Below the table, there are options to 'Pilih Semua' (Select All) and 'Remove from central columns'. At the bottom, there are buttons for 'Cetak' (Print), 'Usulkan struktur tabel' (Propose table structure), 'Lacak tabel' (Track table), 'Move columns', and 'Normalisasi' (Normalize).

7. Tugas mandiri mahasiswa. Melakukan dokumentasi pengujian API **ViGeNesia** menggunakan *Postman*. Bentuk/media unjuk kerja pengujian diperbolehkan memilih salah satu diantaranya:

1. Video. Upload video tersebut pada chanel youtube. Catatan video wajib ada identitas kampus (logo), identitas prodi, identitas mahasiswa dan dibuat dengan kreatifitas mahasiswa.
2. Dalam bentuk laporan dengan format pdf. (format pada halaman berikutnya)

PENGUJIAN RESTFUL API ViGeNesia



Nama TIM:

12192706 - Fauko Misalam
nim - nama
nim-nama
dst

DOKUMENTASI PENGUJIAN VIGENESIA TOOLS; POSTMAN

A. GET

Uraikan tahapan dan tampilkan gambar jika diperlukan

1. ?
2. ?
3. ?

B. POST

Uraikan tahapan dan tampilkan tabel atau gambar jika diperlukan

1. ?
2. ?
3. ?

C. PUT

Uraikan tahapan dan tampilkan tabel/gambar jika diperlukan

1. ?
2. ?
3. ?

D. DELETE

Uraikan tahapan dan tampilkan tabel/gambar jika diperlukan

1. ?
2. ?

3. ?