

MODUL 13

MEMBANGUN WEB SERVICE RESTful API LARAVEL (1)



CAPAIAN PEMBELAJARAN

1. Mampu membuat aplikasi Web Service menggunakan RESTful API menggunakan Laravel
2. Mampu menguji pertukaran dengan aplikasi Postman
3. Mampu menguji pertukaran data antar server dengan format JSON menggunakan curl PHP



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Sistem Operasi Linux/Window 10
2. PHP 5.x, diutamakan versi 7.x keatas
3. Database MySQL
4. Web server Apache
5. Composer
6. Text Editor Notepad++ atau Visual Studio Code
7. Postman
8. Jaringan lokal
9. 2 unit web server



DASAR TEORI

1. RESTful API ?

REST (*Representational State Transfer*) merupakan salah satu arsitektur web menggunakan protokol HTTP untuk pertukaran data. Sedangkan *API (Application Programming Interface)* sebuah antarmuka dikemas dalam bentuk kelas, method. Sehingga REST API merupakan antarmuka program yang menyediakan pertukaran data. Konsep REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000.

Arsitektur REST dirancang menggunakan komponen-komponen berikut:

- client: bagian mengirim permintaan.
- Server: Proses server dan menanggapi permintaan.

- *Resource*: adalah konsep untuk mendeskripsikan berbagai hal berdasarkan penggunaan.

Cara kerjanya, REST Server menyediakan jalur untuk akses data, sedangkan REST client melakukan akses *resource* ke REST server, kemudian untuk ditampilkan. *Resource* yang dihasilkan sebenarnya berupa teks, namun formatnya bisa bermacam-macam, sesuai dengan keinginan, umumnya menggunakan format JSON dan XML, seperti ditunjukkan pada gambar berikut:



Beberapa method yang digunakan untuk akses ke Server, paling sederhana sebagai berikut:

- GET /resources: digunakan untuk membaca/mengambil resource.
- GET /resources/{id}: membaca resource dengan kunci id
- POST /resources: Untuk mengirim resource baru ke server.
- PUT/PATCH /resources/{id}: Untuk melakukan update terhadap resource
- DELETE /resources/{id}: Untuk menghapus sebuah resource

2. Kerja RESTful

Pada saat client mengirimkan request melalui HTTP Request, server merespon melalui HTTP Response.

1. Komponen dari http request:

- Verb, HTTP method yang digunakan misalnya GET, POST, DELETE, PUT
- *URI, Uniform Resource Identifier (URI)* untuk mengidentifikasi lokasi resource pada server.
- HTTP Version, menunjukkan versi yang digunakan
- Request Header, berisi metadata untuk HTTP Request.
- Request Body, adalah isi dari data.

2. Komponen dari http response:

- Status/Response Code, mengindikasikan status server terhadap resource yang *direquest*.
misal jika nilai status: 404, artinya resource tidak ditemukan dan 200 response OK.
- HTTP Version, menunjukkan versi dari HTTP yang digunakan.
- Response Header, berisi metadata untuk HTTP Response.
- Request Body, adalah isi dari data.



PRAKTIK

1. Buatlah Controller baru menggunakan nama KotaApiController, dengan artisan berikut;

```
php artisan make:controller KotaApiController
```

2. Buka KotaApiController.php kemudian tambahkan seperti pada kode program berikut:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\models\Kota;
use App\models\Propinsi;
use Illuminate\Support\Facades\DB;

//menyediakan api
class KotaApiController extends Controller
{
    /*****
     * Method memberi layanan pembacaan semua rekaman
     * @return objek
     *****/
    public function index()
    {
        $kota = DB::table('kota')
            ->join('propinsi', 'kota.propinsi_id', '=',
                'propinsi.id')
            ->select('kota.id', 'kota.nama_kota', 'propinsi.propinsi')
            ->orderBy('propinsi.propinsi')
            ->get();
        return $kota;
    }
}
```

```

/*****
    untuk pengisian dropdownlist propinsi
    @return objek
    *****/

public function propinsi()
{
    $propinsi = DB::table('propinsi')
        ->select('id','propinsi')
        ->orderBy('propinsi')
        ->get();

    return $propinsi;
}

/*****
    Menambahkan rekaman baru
    @return String
    *****/

public function create(Request $request)
{
    $kota = new Kota;
    $kota->propinsi_id = $request->propinsi_id;
    $kota->nama_kota    = $request->nama_kota;
    $kota->user_id      = $request->user_id;

    try {
        $kota->save();
    } catch (\Illuminate\Database\QueryException $e) {
        return "Ada kesalahan = ".$e->getMessage();
    }
    return "Data sudah tersimpan";
}

/*****
    Menampilkan 1 rekaman
    @return objek
    *****/

public function show($id)
{
    $kota = DB::table('kota')
        ->join('propinsi', 'kota.propinsi_id', '=',
            'propinsi.id')
        ->select('kota.id','kota.nama_kota','propinsi.propinsi')
        ->where('kota.id',$id)
        ->first();
    return json_encode($kota);
}

/*****
    untuk mengubah rekaman

```

```

@param Request
@param $id integer
@return String
*****/

public function update(Request $request, $id)
{
    // simpan
    $kota = Kota::find($id);
    $kota->propinsi_id = $request->propinsi_id;
    $kota->nama_kota = $request->nama_kota;
    try {
        $kota->save();
    } catch (\Illuminate\Database\QueryException $e) {
        return "Ada kesalahan = ".$e->getMessage();
    }
    return "Kota sudah diubah";
}

/*****
untuk menghapus rekaman
@param $id integer
@return String
*****/

public function delete($id)
{
    $kota = Kota::find($id);
    $kota->delete();

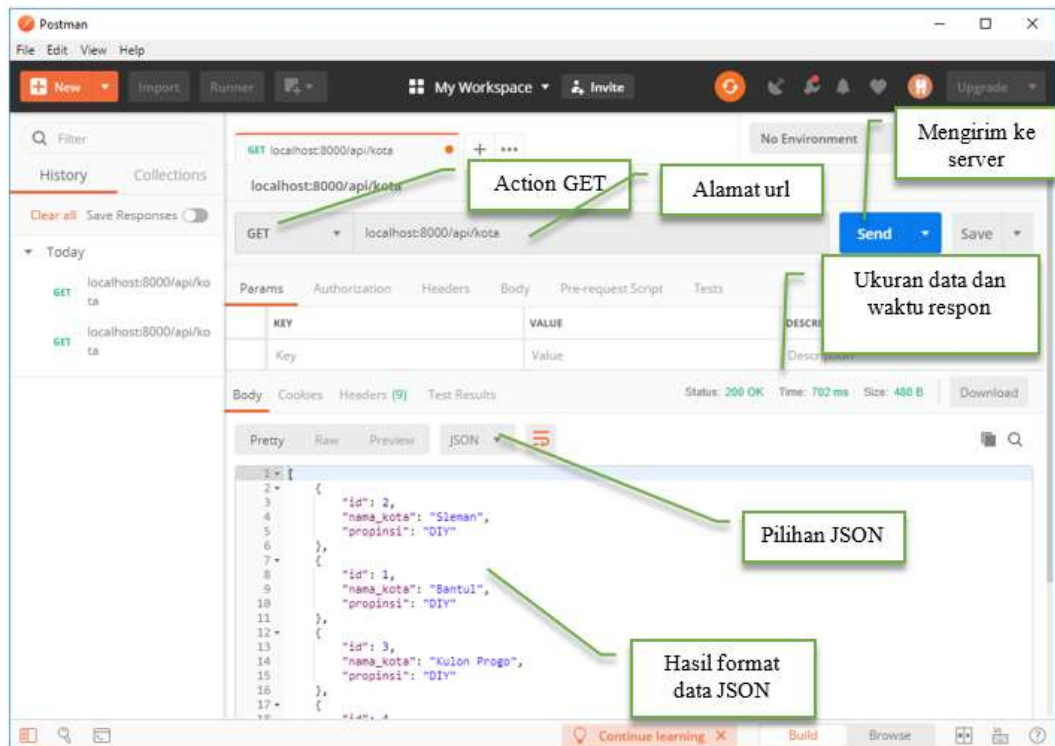
    return "Kota telah dihapus";
}
}

```

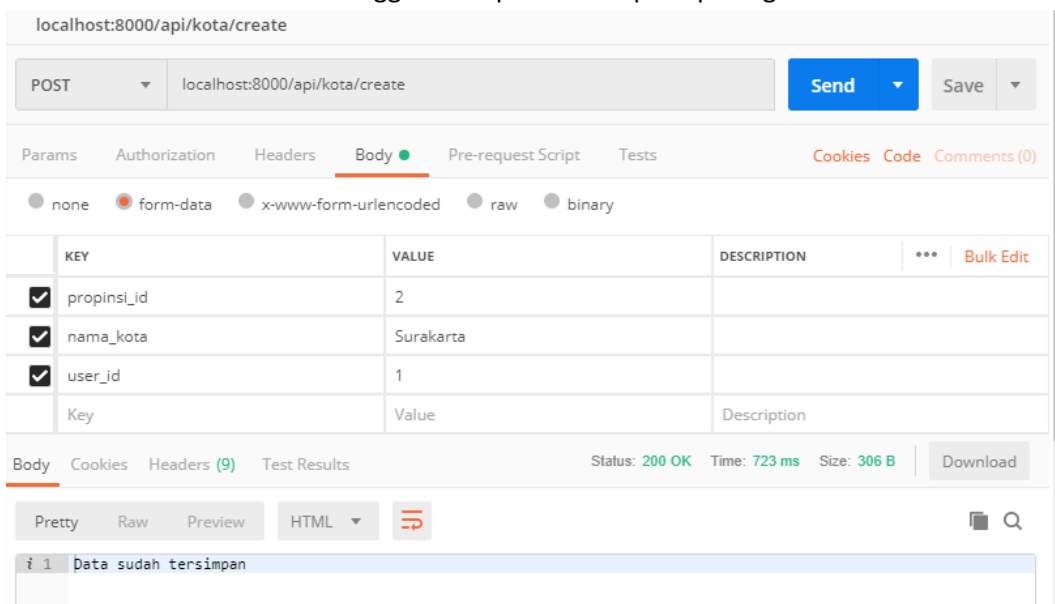
3. Menguji dengan menggunakan browser, seperti pada gambar berikut:



4. Menguji menggunakan Postman, membaca semua rekaman, seperti pada gambar berikut:



5. Menambahkan rekaman menggunakan postman seperti pada gambar berikut:



6. Menggunakan Fungsi cURL, buatlah Controller KotaClientController tambahkan seperti pada kode program berikut:

```
<?php
```

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class KotaClientController extends Controller
{
    public function index()
    {
        $curl = curl_init();

        curl_setopt_array($curl,[
            CURLOPT_PORT => "8000",
            CURLOPT_URL => "http://localhost:8000/api/kota/",
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_ENCODING => "",
            CURLOPT_MAXREDIRS => 10,
            CURLOPT_TIMEOUT => 30,
            CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
            CURLOPT_CUSTOMREQUEST => "GET",
            CURLOPT_HTTPHEADER =>[
                "accept: */*",
                "Content-type: application/json",
                "Postman-Token:
                    340cbc02-867b-4db9-85d5-2b6d6d64b090"
            ],
        ]);

        $response = curl_exec($curl);
        $err = curl_error($curl);

        curl_close($curl);

        if ($err) {
            echo "cURL Error #:" . $err;
        } else {
            $data=json_decode($response);
        }
        echo "<table
            border=\"1\"><tr><td>ID</td><td>KOTA</td>
            <td>PROPINSI</td></tr>";
        foreach($data as $d)
        {
            echo "<tr><td>$d->id</td>
                <td>$d->nama_kota</td>
                <td>$d->propinsi</td></tr>";
        }
        echo "</table>";
    }
}

```

```
}
```

7. Tambahkan file routes/web.php, seperti berikut:

```
Route::get('kotaClient/', 'KotaClientController@index')  
    ->name('kotaClient.index');
```

8. Ujikan dengan dua server secara lokal/virtual, gunakan *command prompt*

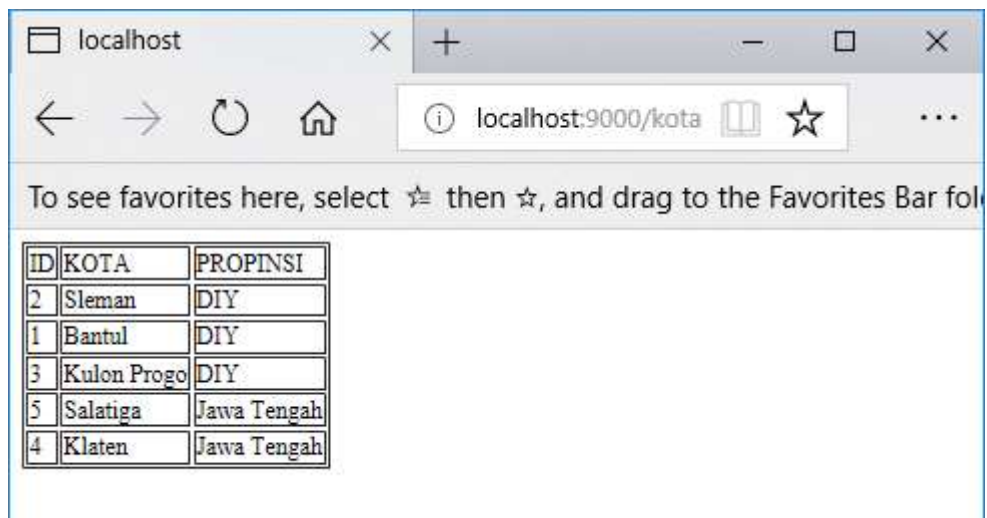
Sebagai Server penyedia web service seperti berikut:

```
C:\xampp\htdocs\sipusta>php -S localhost:8000 -t public  
PHP 7.1.11 Development Server started at Wed Aug 14 09:56:01 2019  
Listening on http://localhost:8000
```

Sebagai Server mengakses web service seperti berikut:

```
C:\xampp\htdocs\proyek1>php -S localhost:9000 -t public  
PHP 7.1.11 Development Server started at Wed Aug 14 15:17:23 2019  
Listening on http://localhost:9000
```

9. Lakukan pengujian seperti hasil keluar pada gambar berikut:





LATIHAN

1. Ujian dengan postman dengan perintah POST, DELETE
2. Gunakan dua server berpasangan dengan rekan Anda, periksa alamat IP masing-masing komputer yang Anda gunakan.
3. Misalkan komputer Server API ip: 172.18.100.25, jalankan menggunakan perintah:

```
php -S 172.18.100.25:8000 -t public
```

4. Dan komputer Server yang mengakses dengan ip: 172.18.100.26, jalankan menggunakan perintah:

```
php -S 172.18.100.26:7000 -t public
```

5. Jalankan <http://172.18.100.26:7000/kotaClient/>, amati hasilnya, jika database di server Server API ip: 172.18.100.25 ditambahkan, refresh ulang di <http://172.18.100.26:7000/kotaClient/>, apa hasilnya?, buatlah kesimpulan.
6. Buatlah Operasi CRUD kota, menggunakan 2 sever.



TUGAS

Sebuah perusahaan pengiriman barang menyediakan webservice untuk bisa diakses tarif jasa pengiriman dari beberapa kota di Indonesia. Agar supaya pengguna jasa pengiriman sistemnya bisa menggunakan tarif tersebut maka:

1. Buat daftar tarif, jarak antar kota, harga per 1 kg
2. Buat web service API untuk mengakses tarif
3. Tunjukkan hasil pengujian dan cara menggunakan API



REFERENSI

1. Andres Castelo, Laravel API Tutorial: How to Build and Test a RESTful API, <https://www.toptal.com/laravel/restful-laravel-api-tutorial>
2. cURL Functions, <https://www.php.net/manual/en/function.curl-init.php>