

# **LAPORAN MODUL 13**

## **PRAKTIKUM Pemrograman Web Framework**



**Disusun oleh :**

Nama : Gilang Rahma Aji P

NIM : 223110001

Kelas : RPLA

**UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA**  
**YOGYAKARTA**

**2024**

## A. TUJUAN PRAKTIKUM

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

## B. DASAR TEORI

RESTful API (Representational State Transfer) adalah sebuah arsitektur berbasis web yang menggunakan protokol HTTP untuk pertukaran data antara client dan server. Arsitektur ini dikembangkan oleh Roy Fielding pada tahun 2000 dan didasarkan pada konsep sederhana menggunakan komponen-komponen berikut:

- **Client:** Komponen yang mengirimkan permintaan (request) ke server.
- **Server:** Komponen yang menerima permintaan dan memberikan respon (response) kepada client.
- **Resource:** Konsep yang digunakan untuk menggambarkan berbagai hal yang bisa diakses melalui API, seperti data atau layanan.

Cara kerjanya, RESTful API memungkinkan client untuk berinteraksi dengan server melalui metode HTTP seperti GET untuk membaca/mengambil resource, POST untuk mengirimkan resource baru ke server, PUT/PATCH untuk melakukan update terhadap resource, dan DELETE untuk menghapus sebuah resource. Respon dari server berupa status code yang mengindikasikan keberhasilan atau kegagalan permintaan, bersama dengan data yang umumnya disampaikan dalam format JSON atau XML.

## C. PEMBAHASAN LISTING PRAKTIK

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

```
PS D:\LARAVEL\7> php artisan make:controller KotaApiController
```

```
INFO Controller [D:\LARAVEL\7\app\Http\Controllers\KotaApiController.php] created successfully.
```

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

```
<?php
namespace App\Http\Controllers\Api;
use Illuminate\Http\Request;
use App\Models\Kota;
use App\Models\Propinsi;
use Illuminate\Support\Facades\DB;
use App\Http\Controllers\Controller;
//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.nama_propinsi as propinsi')
            ->orderBy('propinsi')
            ->get();

        return $kota;
    }

    /**
     * untuk pengisian dropDownList propinsi
     * @return objek
     */
    public function propinsi()
    {
        $propinsi = DB::table('propinsi')
            ->select('id','propinsi')
            ->orderBy('propinsi')
            ->get();

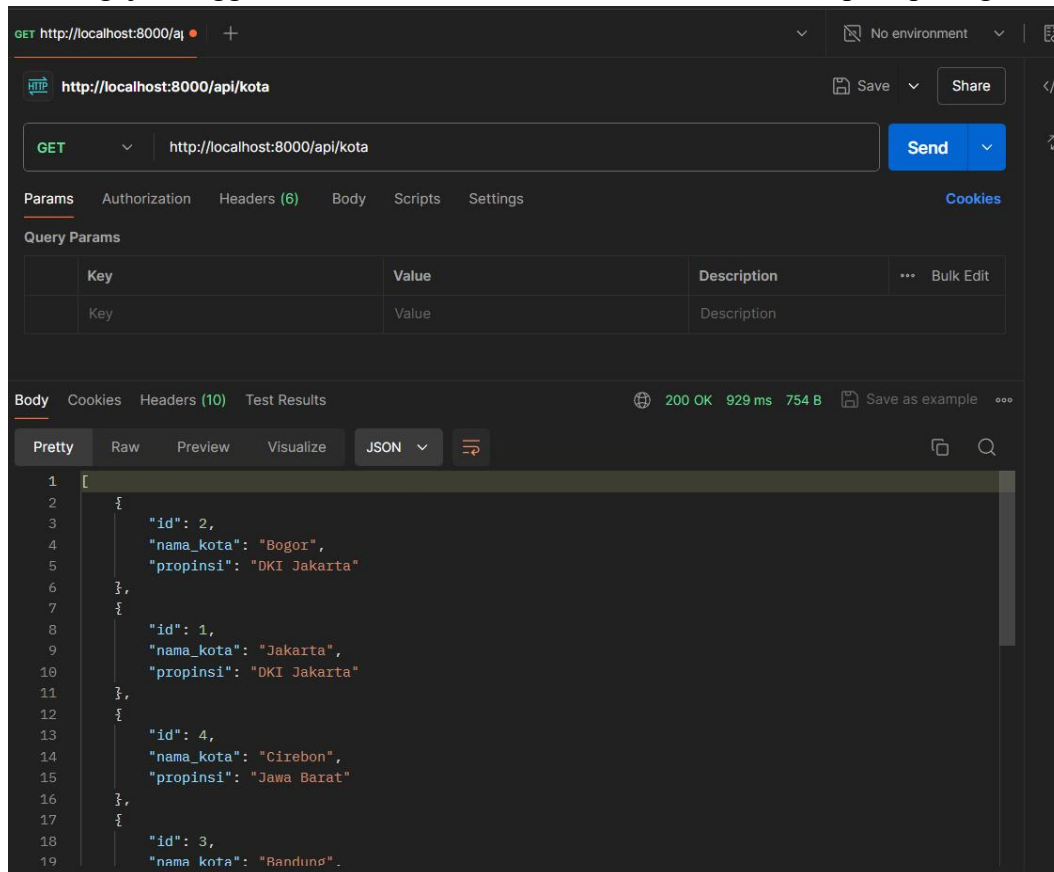
        return $propinsi;
    }

    /**
     * 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);
    }
}
```



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



6. Menambahkan rekaman menggunakan postman seperti pada gambar berikut:

7. Bagian server sebagai client menggunakan Fungsi cURL, buatlah Controller KotaClientController tambahkan seperti pada kode program berikut:

```
<?php
namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
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"
            ],
        ],
    );
    }
```

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

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

9. Ujikan dengan dua server secara lokal/virtual, gunakan command prompt Sebagai Server penyedia web service seperti berikut:

Sebagai Server yang mengakses web service seperti berikut:

10. Lakukan pengujian seperti hasil keluar pada gambar berikut:

## **D. PEMBAHASAN TUGAS**

Lampirkan hasil screenshoot/capture gambar baik source code maupun hasil output kemudian diberikan pembahasan. (Hasil capture tidak perlu full satu layar, namun tampak jelas).

## **E. KESIMPULAN**

RESTful API menyediakan antarmuka untuk berkomunikasi antara client dan server menggunakan protokol HTTP. Ini memungkinkan pengembang untuk mengimplementasikan operasi CRUD (Create, Read, Update, Delete) terhadap resource yang tersedia di server secara efektif. Dengan mematuhi prinsip-prinsip RESTful, pengembang dapat membangun aplikasi yang terstruktur, fleksibel, dan mudah diintegrasikan dengan sistem backend yang kompleks, sehingga meningkatkan skalabilitas dan kinerja aplikasi secara keseluruhan.