

# Implementasi REST API Untuk Fitur Rencana Strategis Program Pada SIMPEDA

Muhammad Kevin Naufal Faza Affianto  
Program Studi Informatika – Program Sarjana  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
18523224@students.uii.ac.id

Andhik Budi Cahyono  
Program Studi Informatika – Program Sarjana  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
andhik.budi@uii.ac.id

**Abstract**— BAPEDDA DIY membutuhkan aplikasi website yang dapat meningkatkan pengelolaan serta pelayanan untuk melakukan pembangunan daerah. CV. Atsoft Teknologi memberikan solusi berupa penawaran aplikasi yang dapat memenuhi permintaan tersebut kepada BAPEDDA DIY. Aplikasi yang ditawarkan adalah Sistem Informasi Perencanaan Pembangunan Daerah (SIMPEDA). SIMPEDA adalah aplikasi berbasis *website* yang merupakan lanjutan dari SENGGUH (Sistem Evaluasi Pertanggungjawaban Pembangunan Daerah). Sistem Informasi SENGGUH sisi *client* tidak dapat mengambil data dari *server* secara langsung karena belum memiliki API. Dari permasalahan tersebut, dikembangkanlah Sistem Informasi SIMPEDA menggunakan API berarsitektur *Representational State Transfer* (REST). Aplikasi ini menggunakan REST API untuk mempercepat proses pertukaran data antara *client* dan *server*. Salah satu fitur yang terdapat pada SIMPEDA yang akan dibahas pada makalah ini adalah fitur rencana strategis program. Fitur ini, dapat memudahkan BAPEDDA DIY untuk melakukan pendataan program rencana strategis pembangunan daerah selama satu periode jabatan kepala daerah. Pembahasan dan hasil yang didapatkan yakni: Arsitektur REST API, Tahap implementasi REST API untuk fitur rencana strategis program pada SIMPEDA, Tampilan antarmuka SIMPEDA pada fitur rencana strategis program, Pengujian fitur rencana strategis program. Pada hasil menunjukkan bahwa REST API dapat mempercepat proses pertukaran data antara *client* dan *server* sehingga proses pendataan program perencanaan pembangunan daerah menjadi lebih efisien. Tujuan dari penelitian ini juga untuk meningkatkan pemahaman dan pengalaman developer dalam mengembangkan aplikasi *website* dengan teknologi REST API.

**Keywords**—REST API, SIMPEDA, Rencana Strategis Program, aplikasi website

## I. PENDAHULUAN

Perencanaan pembangunan daerah adalah suatu proses penataan kegiatan yang melibatkan berbagai *stakeholder* di dalamnya, memanfaatkan serta mengalokasikan sumber daya yang ada dalam rangka perbaikan sosial di lingkup daerah dalam durasi waktu tertentu. *Stakeholder* dapat diklasifikasikan ke dalam 3 domain, yaitu institusi pemerintah yang terdiri dari eksekutif dan legislatif daerah, sektor swasta yaitu pelaku bisnis baik individu maupun institusi, dan masyarakat baik individu maupun kelompok [1]. Pada saat melakukan perencanaan pembangunan, pemerintah daerah menggunakan metode konvensional, baik

distribusi maupun pengorganisasian data, sehingga pelaksanaan kegiatan yang diusulkan seringkali tidak sesuai dengan hasil. Selain itu, pihak kelurahan/desa serta Satuan Kerja Perangkat Daerah (SKPD) kerap mengalami kesulitan dalam memantau usulan karena belum adanya sistem yang memadai.

CV. Atsoft Teknologi berdiri sejak tahun 2006 dari beberapa lulusan Sarjana Teknik Informatika yang telah melakukan eksplorasi dan riset tentang sains dan teknologi dengan filosofi otomasi kerja dan integritas jaringan kinerja perusahaan. Perusahaan ini memberikan solusi otomasi proses bisnis dari manual menuju *computerized* serta mengedepankan efektifitas dan efisiensi kinerja sumber daya perusahaan yang fokus kepada konsumen sehingga *long term innovation* dapat terwujud. CV. Atsoft Teknologi bertarget untuk mengembangkan teknologi informasi pada berbagai bidang, seperti: pemerintahan, pendidikan, industri, dan kesehatan

Badan Perencanaan Pembangunan Daerah (BAPEDDA) DIY adalah pejabat teknis daerah yang bertanggung jawab kepada Walikota melalui Sekretaris Daerah pada bidang perencanaan pembangunan daerah [2]. BAPEDDA DIY telah diresmikan dan ditetapkan dengan Keputusan Kepala Daerah Tingkat I DIY Nomor 341 Tahun 1974 tentang Pembentukan Badan Perencanaan Pembangunan Daerah Tingkat I DIY [3]. BAPEDDA DIY membutuhkan aplikasi berbasis *website* yang dapat meningkatkan pengelolaan serta untuk mewujudkan perencanaan pembangunan daerah yang berkualitas. Sehingga CV. Atsoft Teknologi menawarkan solusi kepada BAPEDDA DIY berupa aplikasi berbasis *website* yang bernama SIMPEDA. BAPEDDA DIY mengharapkan SIMPEDA dapat bermanfaat untuk mempermudah penyusunan perencanaan pembangunan daerah serta merealisasikan sasaran rencana strategis program yang telah direncanakan secara akurat, *real-time*, dan fleksibel.

Aplikasi SIMPEDA dikembangkan dengan menggunakan pendekatan REST API. Penggunaan REST API pada aplikasi SIMPEDA bertujuan untuk memudahkan pertukaran data antara *client* dan *server* karena memiliki konfigurasi yang mudah dipahami dan bersifat *open source*. Pada REST API terdapat REST *server* yang memberikan jalur untuk mengakses *resource* atau data dan REST *client* yang dapat mengakses *resource* untuk menggunakan atau menampilkan data atau dapat dikatakan pertukaran antar *state*

dimana *state* tersebut merupakan halaman *web* yang akan dipanggil oleh *browser* melalui tautan HTTP [4]. Teknologi REST memiliki kelebihan yang dalam performa dan kecepatan transfer data dibandingkan dengan teknologi SOAP. Keluaran dari REST lebih flexibel yakni XML atau JSON sedangkan SOAP hanya XML [5].

Tujuan dari penulisan makalah ini adalah untuk menjelaskan bagaimana REST API dapat digunakan untuk mempercepat proses pertukaran data pada aplikasi *website* SIMPEDA. Manfaat yang dihasilkan dari penggunaan REST API yaitu dapat mempermudah *developer* dalam melakukan kerja sama antar sistem dan dapat mempercepat pembuatan fitur baru pada sistem yang dikembangkan [6]

## II. LANDASAN TEORI

### A. Fitur Rencana Strategis

Fitur rencana strategis adalah salah satu fitur yang ada pada aplikasi *website* SIMPEDA. Fitur ini berfungsi sebagai dasar untuk menentukan strategi dan memutuskan alokasi sumber daya selama satu sampai lima tahun sehubungan dengan tugas dan fungsi Satuan Kerja Perangkat Daerah SKPD [7]. Terdapat dua menu dalam fitur rencana Strategis (Renstra) yaitu renstra program dan renstra sasaran yang berkaitan dengan fungsi dan tugas Satuan Kerja Perangkat Daerah (SKPD) yang berdasar kepada Rencana Pembangunan Jangka Menengah Daerah (RPJMD) yang bersifat implisit [8].

### B. Application Programming Interface (API)

*Application Programming Interface* (API) adalah Bahasa dan format pesan yang digunakan aplikasi atau program untuk berkomunikasi dengan *Operational System* (OS) atau program kontrol lainnya seperti manajemen database. Implementasi API dengan cara menulis fungsi yang ada dalam program yang menyediakan tautan pada instruksi program yang akan dieksekusi. API sering disebut dengan *web service* karena API memberikan layanan *service* pada sisi *client*. Adapun fungsi pada API yaitu: JSON, XML, JavaScript, SOAP, REST [9].

*Application Programming Interface* (API) merupakan antarmuka yang digunakan untuk mengakses layanan atau aplikasi dari suatu program. API memungkinkan developer tidak perlu mengulang dari awal pengembangan sistem dengan memanfaatkan fungsi yang sudah ada dari aplikasi lain. Keuntungan menggunakan API yaitu: penggunaan API fleksibel yang artinya dapat digunakan di sistem operasi mana saja, dan bahasa yang digunakan mudah dimengerti seperti JSON [10].

### C. REST API

Teknologi Representational state transfer (REST) diciptakan oleh Roy Fielding dari University of California. Aplikasi REST sering digunakan untuk pengembangan layanan berbasis *web* atau *mobile* karena ringan dan sederhana apabila dibandingkan dengan SOAP. Prinsip utama desain dari REST API adalah kinerja, portabilitas dan kesederhanaan. [11].

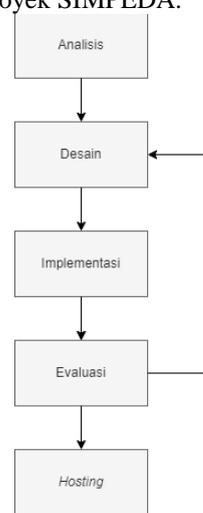
REST API mendukung beragam sistem untuk berinteraksi serta menerima/mengirim data dengan mudah. Setiap penggunaan REST API, didukung oleh URL dan HTTP. Data dalam *database* pada suatu aplikasi dipetakan dengan *endpoint* API pada REST API [12].

REST API menggunakan HTTP untuk menentukan *request* dan *response*. Agar informasi dapat lebih sederhana dibaca dan di analisa pada sisi *client*, keluaran yang dihasilkan oleh API *server* berupa JSON [13]. Beberapa *method* dan tipe *request* yang tersedia untuk melakukan komunikasi data, antara lain:

1. GET, adalah *method* yang digunakan untuk mendapatkan data dari *server*.
2. POST, adalah *method* yang digunakan untuk melakukan *create* data dan data yang telah dibuat akan disimpan kedalam *database*
3. DELETE, adalah *method* yang diguakan untuk menghapus data dalam *database*.

## III. METODOLOGI

Metodologi yang digunakan untuk mengembangkan proyek SIMPEDA dengan REST API Tahap metodologi dimulai dari analisis hingga *hosting*. **Gambar 1** merupakan Tahap metodologi proyek SIMPEDA.



**Gambar 1.** Tahap Perancangan Aplikasi

Metode ini merupakan metode yang terprogram secara sistematis sehingga setiap tahap dari metode tersebut mudah diimplementasikan pada aplikasi yang dikembangkan. Pada tahap evaluasi, Tim developer dapat kembali ke tahap implementasi apabila sistem memiliki kesalahan atau kekurangan pada tahap akhir. Setelah seluruh tahapan telah disetujui, developer dapat melanjutkan ke tahap *Hosting*. Tim developer terdiri dari *Project Manager* (PM), *UI/UX Designer*, *Frontend developer* dan *Backend developer*.

### A. Analisis

Analisis adalah tahap pengembangan yang dilakukan oleh tim developer yaitu *Project Manager* (PM). Pada tahap ini PM berdiskusi dengan klien untuk menganalisa seluruh kebutuhan klien. Langkah-langkah pada proses analisis sistem yaitu: mengidentifikasi masalah, menganalisa kebutuhan sistem serta mencari solusi untuk permasalahan tersebut. Apabila proses analisa telah selesai, PM menyerahkan hasil analisa kepada bagian *UI/UX Designer* untuk melakukan pengembangan pada tahap *Design*. Tujuan dari proses ini adalah untuk mengetahui kelemahan dan kendala sistem

### B. Desain

Desain adalah tahap yang dilaksanakan oleh tim developer yang berposisi sebagai *UI/UX Designer*. Pada

tahap ini *UI/UX designer* mengembangkan desain berupa *mockup* yang akan diserahkan kepada tim developer yang berposisi sebagai *Frontend Developer*. Alat yang digunakan oleh *UI/UX* dalam mengembangkan *mockup* adalah Figma dan Adobe XD.

### C. Implementasi

Implementasi adalah tahap pengembangan yang dilakukan oleh *Frontend developer* dan *Backend developer*. Pada tahap ini, *Backend developer* mengimplementasikan REST API dan melakukan pengujian aplikasi SIMPEDA. Pada tahap ini juga, *Frontend developer* menambahkan tampilan user dengan pemrograman PHP dan JavaScript menggunakan *framework* Phalcon.

### D. Evaluasi

Evaluasi adalah tahap pengembangan yang dilakukan oleh tim developer untuk melakukan perbaikan pada proses pengembangan sistem. Berikut adalah proses yang dilakukan untuk melakukan evaluasi:

1. *Project Manager* bersama tim developer berdiskusi dengan klien. Apabila kedua belah pihak sudah setuju dan dinilai cukup, tim developer melanjutkan ke tahap *Deploy*.
2. Tim developer dapat mengulang kembali tahap Desain apabila perlu mengalami perbaikan atau belum disetujui.
3. Tim developer yang berposisi sebagai *Backend developer* mengimplementasikan REST API pada aplikasi *website* SIMPEDA dan melakukan pengujian.

Apabila pihak klien dan tim developer puas dengan hasil evaluasi, tim developer dapat melanjutkan ke tahap *Deploy*.

### E. Hosting

Hosting adalah tahap yang dilakukan untuk melakukan publikasi aplikasi ke sebuah *website*. Aplikasi *website* SIMPEDA pada tanggal 10 Juni 2022 Masih belum dipublikasikan karena masih dalam proses pengembangan. aplikasi berbasis *website* SIMPEDA akan dipublikasikan setelah tim developer menyelesaikan seluruh proses pembuatan sistemnya. Setelah aplikasi berhasil dipublikasikan, pengguna dapat melihat dan menggunakan SIMPEDA melalui *website* resmi dengan *url* yang telah ditentukan

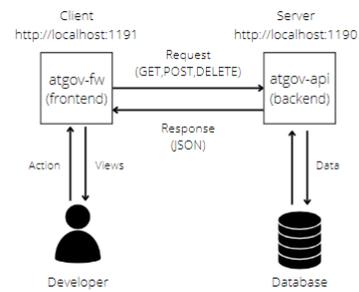
## IV. HASIL DAN PEMBAHASAN.

### A. Arsitektur REST API

Bahasa pemrograman yang digunakan dalam merancang REST API adalah PHP dan *framework* Phalcon. Setelah rancangan REST API telah dibuat, dilanjutkan oleh *frontend developer* untuk membuat tampilan antarmuka. REST API juga berguna untuk pertukaran data antara *client* dan *server*. REST API mampu memisahkan dan menyederhanakan interaksi antara *client* dan *server*. Perubahan yang terjadi pada *server* tidak mempengaruhi *client* sehingga kinerja *client* dan *server* menjadi lebih efisien karena kedua sisi dapat mengembangkan modul tanpa mempengaruhi proses pertukaran data. Berikut method yang digunakan untuk proses pertukaran data *client* dan *server* antara lain:

1. GET

2. POST
3. DELETE



**Gambar 2.** Arsitektur Sistem

Berdasarkan arsitektur pada **Gambar 2**, atgov-fw berfungsi sebagai *frontend* yang menggunakan pemrograman JavaScript, atgov-api berfungsi sebagai *backend* yang menggunakan pemrograman PHP dengan *framework* Phalcon. HTTP *Request* dilakukan pada sisi *client* dengan memanfaatkan *method* GET, POST, DELETE pada *server* yang menyimpan *resource* melalui *endpoint*. Selanjutnya, REST *server* memberikan *feedback* berupa HTTP *response* seperti permintaan dari sisi *client*. *Server* membalas *response* kepada sisi *client* dengan format JSON. *Output* yang berformat JSON akan diganti sehingga sisi *client* dapat membaca format dengan jelas.

### B. Tahap Implementasi REST API untuk Fitur Rencana strategis program pada SIMPEDA

Pada tahapan implementasi REST API, atgov-api adalah sisi *server* (REST *server*) dan atgov-fw adalah sisi *client* (REST *client*) sesuai dengan yang telah dijelaskan pada **Gambar 2**. SIMPEDA beroperasi menggunakan CRUD (*Create, Read, Update, Delete*) dengan fitur tambahan lainnya seperti *limit show*, pencarian, dan *pagination*. Dengan menggunakan REST API pertukaran data antar *client* dan *server* dapat menjadi lebih mudah.

#### 1) REST Server

Pada arsitektur REST API terdapat beragam *method* yang digunakan pada protokol HTTP, namun pada proyek ini hanya memerlukan tiga *method* yaitu GET, POST, dan DELETE. Pada **Tabel I** merupakan daftar *endpoint* REST API yang sedang dikembangkan

Tabel I *Endpoint* REST API

Endpoint	Method
/renstra_program/get	GET
/renstra_program/get_list	GET
/renstra_program/insert	POST
/renstra_program/update	POST
/renstra_program/delete	DELETE

Selanjutnya, perlu membuat *Controller* untuk menentukan proses pengiriman data agar data dapat diakses oleh pengguna. *Controller* terdiri dari seluruh *endpoint* REST API, sehingga pertukaran data menjadi lebih efektif. Pada **Gambar 3** adalah salah satu contoh implementasi dari

endpoint `/renstra_program/getlist` pada *controller* menggunakan method GET.

```
public function getListAction()
{
    $p = $this->request->getJsonRawBody();

    //njikuk data
    $hs = new RenstraProgram();
    $hs->setParams($p);
    //filter data
    $hs->go_unor_id_opd = $p->go_unor_id_opd;
    $hs->renstra_periode_id = $p->renstra_periode_id;
    $data = $hs->getData();

    //ngetung jumlah record terfilter
    $filtered_record = $hs->getDataCount();

    //ngetung jumlah record total
    $hstot = new RenstraProgram();
    $total_record = $hstot->getDataCount();

    $other_data = new stdClass();
    $other_data->total_record = $total_record;
    $other_data->filtered_record = $filtered_record;

    return $this->setJsonRespon(null, null, $data, $other_data);
}
```

**Gambar 3** Contoh penggunaan method GET pada *controller*

Pada **Gambar 3**, penggunaan method GET berfungsi untuk menampilkan semua data rencana strategis program dengan cara mengakses alamat sesuai *endpoint* yang dibuat. Pada contoh ini, Alamat yang digunakan adalah [http://localhost:1190/renstra\\_program/get\\_list](http://localhost:1190/renstra_program/get_list). Bagian pertama merupakan nama domain. Proses implementasi masih menggunakan database local sehingga dituliskan "localhost". Penomoran *port* dari *server* dituliskan dengan Angka "1190". Selanjutnya merupakan direktori sebagai penanda bahwa layanan API telah dapat diakses. Pada bagian akhir, merupakan data yang terdapat dalam *controller*. Pada **Gambar 4**, merupakan hasil eksekusi *url* diatas menggunakan Postman. *Output* yang didapatkan berupa format JSON.

**Gambar 4** Output Data Dengan Format JSON pada Postman

## 2) REST Client

Pada sisi *client* diimplementasikan dengan atgov-fw menggunakan Bahasa pemrograman PHP dan JavaScript dengan *framework* Phalcon sebagai salah satu *framework* berbasis *website*. Agar *client* dapat terhubung dengan API, maka diperlukan fungsi `getFormatDatatables`. Fungsi tersebut berguna untuk memanggil *data table* yang ada pada sisi *server*. **Gambar 5** menampilkan *source code* saat *client* melakukan HTTP *request* untuk memanggil data rencana

strategis program yang telah disimpan di dalam *database* menggunakan *method* GET.

```
35
36
37 /*
38  * get list bentuk datatable server side
39  */
40 function getFormatDatatables($order_arr, $api_controller, $param = null)
41 {
42     $param['offset'] = $this->request->get('start');
43     $param['limit'] = $this->request->get('length');
44     $param['search_str'] = $this->request->get('search')['value'];
45     $order_column = $this->request->get('order')['0']['column'];
46     $order_dir = $this->request->get('order')['0']['dir'];
47     $param['order_by'] = $order_arr[$order_column] . " " . $order_dir;
48     //print_r($param);
49     //exit;
50     $result = $this->req($api_controller . '/get_list', $param);
51     //print_r($result->debug);
52     //exit;
53     $data = $result->data;
54     if ($data == null) {
55         $data = [];
56     }
57 }
```

**Gambar 5** Kode *request* dengan method GET

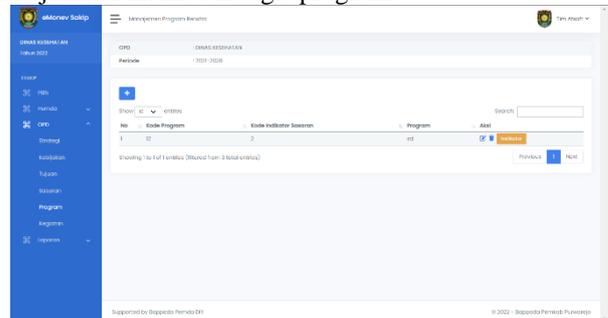
Pada fungsi `getFormatDatatables`, terdapat *state result* yang akan menarik data dari API. Kemudian data akan disimpan kedalam *state* data yang bertipe array (`[]`) dengan format JSON.

## C. Tampilan antarmuka SIMPEDA pada fitur Rencana strategis program

Berikut adalah tampilan antarmuka Rencana Strategis (Renstra) Program:

### a) Halaman manajemen Rencana strategis (Renstra) Program

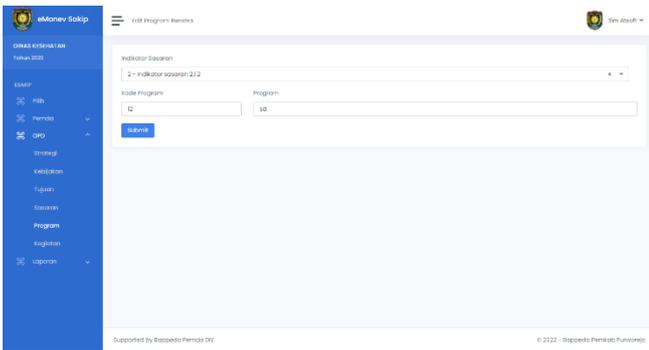
Halaman manajemen rencana strategis program berfungsi untuk memperlihatkan Organisasi Perangkat Daerah (OPD) dan Periode Satuan Kerja Perangkat Daerah (SKPD) yang login kedalam sistem serta mengelola seluruh data rencana strategis program melalui fitur *CRUD* yang tersedia seperti: menambahkan program baru, edit dan update data rencana strategis program, serta menghapus data program. Pada halaman ini, juga terdapat fitur *pagination*, pencarian, dan *limit show*. **Gambar 6**, merupakan tampilan halaman manajemen rencana strategis program.



**Gambar 6.** Halaman manajemen Rencana strategis (Renstra)

### b) Halaman edit Rencana strategis (Renstra) Program

**Gambar 7** merupakan halaman edit rencana strategis program yang sudah ada pada *database* SIMPEDA. Untuk melakukan proses edit, pengguna wajib mengisi indikator sasaran, kode program dan program yang diberikan oleh kepala daerah aktif pada form halaman tersebut. Untuk menuju halaman ini, dengan cara memilih ikon pada bagian Aksi pada halaman utama rencana strategis program.



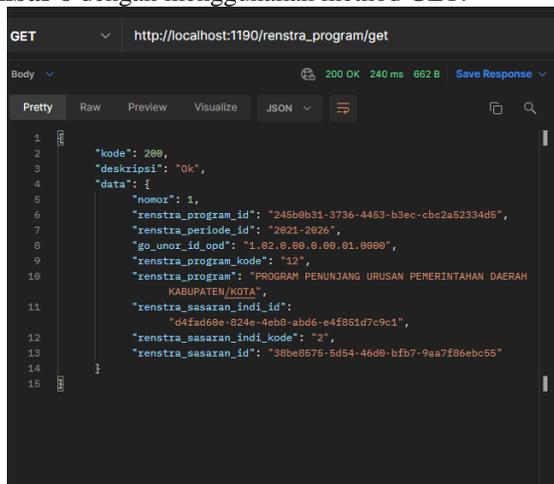
**Gambar 7.** Halaman edit Rencana Strategis (Renstra) Program

#### D. Pengujian fitur Rencana Strategis Program

Tahap terakhir yang dilakukan pada pengembangan proyek SIMPEDA adalah pengujian. Pengujian dilakukan menggunakan Postman untuk memastikan kelayakan REST API pada fitur rencana strategis program dengan menyusun *method* yang akan diuji serta memasukkan *endpoint* sebagai berikut:

##### a) *Url/renstra\_program/get (GET)*

Hasil uji coba url/renstra\_program/get diperlihatkan pada **Gambar 8** dengan menggunakan *method* GET.

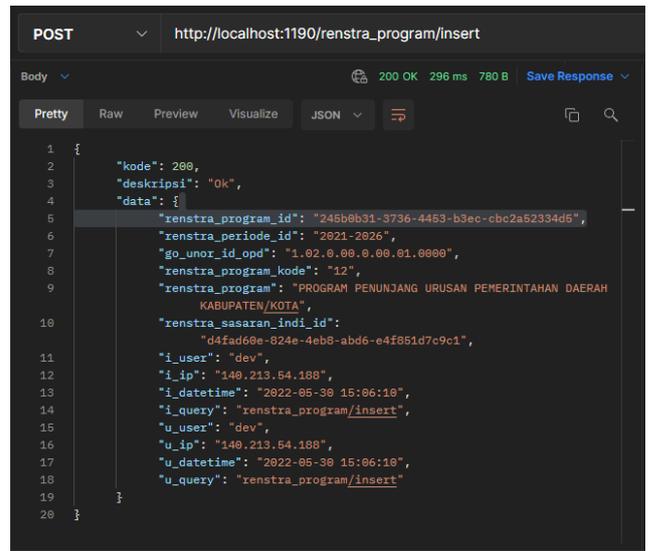


**Gambar 8.** Url/renstra\_program/get (GET)

Pada *url* ini, membutuhkan *form-data* pada *body* untuk mengisi *id* program yang dicari. *Output* yang dihasilkan adalah respon satu *id* pada data JSON rencana strategis program.

##### b) *Url/renstra\_program/insert (POST)*

Hasil uji coba pada url/renstra\_program/insert diperlihatkan pada **Gambar 9** menggunakan *method* POST.

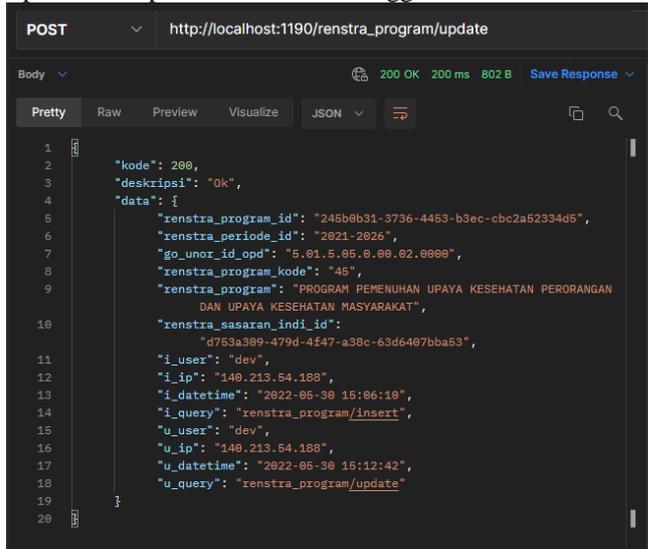


**Gambar 9.** Url/renstra\_program/insert (POST)

Pada *url* ini, memerlukan beberapa data pada *body* berupa *field* yang akan disimpan pada *database* dan menghasilkan respon JSON berupa rencana strategis program yang telah selesai diposting.

##### c) *Url/renstra\_program/update (POST)*

Hasil pengujian pada url/renstra\_program/update diperlihatkan pada **Gambar 10** menggunakan *method* POST.

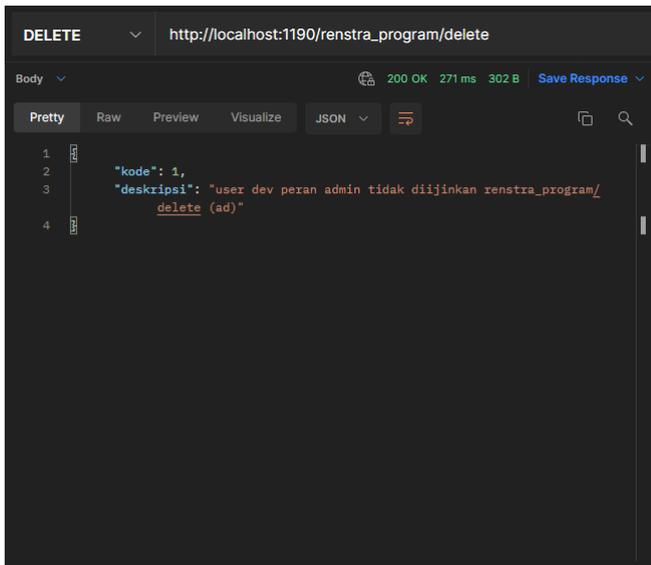


**Gambar 10.** Url/renstra\_program/update (POST)

Pada *url* ini, membutuhkan beberapa data program yang ingin diubah. Apabila data selesai diubah, *text* pada *body* akan disimpan pada *database* dan akan menghasilkan respon JSON berupa data rencana strategis program yang telah diperbarui.

##### d) *Url/renstra\_program/delete (DELETE)*

Hasil uji coba pada url/renstra\_program/delete diperlihatkan pada **Gambar 11** dengan menggunakan *method* DELETE.



**Gambar 11.** Url/renstra\_program/delete (DELETE)

Pada *url* ini, digunakan untuk menghapus data rencana strategis program pada *database* dengan mengisi *id* pada *body*. Namun, pada *action* tersebut tidak sembarang pengguna dapat menghapus data rencana strategis program.

e) *Tabel pengujian REST API*

Berdasarkan pengujian yang dilakukan dengan metode *equivalence partitioning*, penggunaan REST API untuk fitur rencana strategis program pada SIMPEDA mampu membuat proses pertukaran data menjadi efisien. Pada **Tabel II** menunjukkan daftar pengujian REST API beserta hasilnya

Tabel II. Daftar Pengujian REST API

Endpoint	Fungsionalitas yang diuji	Hasil yang diharapkan	Hasil Pengujian
/renstra_program/get	Mengambil data renstra program berdasarkan id	Menampilkan data berdasarkan id	Sukses
/renstra_program/get_list	Menampilkan seluruh data renstra program	Menampilkan seluruh data renstra program	Sukses
/renstra_program/insert	Menambahkan data renstra program	Menampilkan “Data renstra program berhasil ditambahkan” pada halaman tambah data renstra program	Sukses
/renstra_program/update	Mengubah data renstra program	Menampilkan “Data renstra program berhasil diubah” pada halaman manajemen data renstra program	Sukses
/renstra_program/delete	Menghapus data renstra program	Menampilkan “Data renstra program berhasil	Sukses

		dihapus” pada halaman manajemen renstra program	
--	--	---	--

Hasil dan pembahasan REST API sesuai dengan yang telah dijelaskan. Layanan REST API mampu memisahkan dan menyederhanakan interaksi antara *client* dan *server* Perubahan yang terjadi pada *server* tidak mempengaruhi *client* sehingga kinerja *client* dan *server* menjadi lebih efisien karena kedua sisi dapat mengembangkan modul tanpa mempengaruhi proses pertukaran data.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil pengembangan aplikasi berbasis *website* SIMPEDA untuk fitur rencana strategis program dengan mengimplementasikan REST API, dapat disimpulkan bahwa:

1. Penggunaan metodologi pada pengembangan proyek SIMPEDA dapat meningkatkan kualitas dan nilai proyek karena dalam proses pengembangan aplikasi terdapat interaksi terhadap klien sehingga permintaan dan kebutuhan aplikasi klien dapat terpenuhi.
2. Penggunaan format JSON sebagai *output* pada pengujian REST API memudahkan sisi *client* untuk melakukan transkripsi data karena hasil teks yang dikeluarkan sederhana dan mudah dipahami.
3. Implementasi REST API untuk fitur rencana strategis pada aplikasi SIMPEDA dapat memudahkan *client* untuk mengambil data dari *server* sehingga proses pertukaran data menjadi lebih efisien.

Saran untuk mendukung peningkatan perkembangan proyek selanjutnya, yakni dibutuhkan analisis lebih mendalam untuk implementasi teknologi REST API pada pengembangan aplikasi *mobile based* seperti *iOS* dan *Android*.

DAFTAR PUSTAKA

- [1] Yogyakarta, D. I, "RKPD DI Yogyakarta 2018", Bappeda Provinsi Daerah Istimewa Yogyakarta, June 2018.
- [2] Yogyakarta, D. I, "Gambaran Umum". Bappeda Provinsi Daerah Istimewa Yogyakarta. [Online]. Available: <https://bappeda.jogjakota.go.id/page/index/gambaran-umum>. [Accessed 3 June 2022]
- [3] Yogyakarta, D. I, "Sejarah", Bappeda. [Online]. Available: <http://bappeda.jogjaprov.go.id/page/sejarah-bappeda>. [Accessed 3 June 2022].
- [4] Umami, Z., & Ningrum, N. K, "Pengujian Implementasi REST API pada Website Sistem Pencarian Informasi Produk Fashion di Shopee Penulis Korespondensi". *Jurnal Sistem Informasi dan Teknologi*, vol. 03, December 2020.
- [5] Putra, M, & Putera, M, “Analisis Perbandingan metode SOAP dan REST yang digunakan pada Framework Flask untuk membangun Web Service”, *SCAN-Jurnal Teknologi Informasi dan Komunikasi*, vol. 18, June 2019.
- [6] Safitri, R. K., & Putro, H. P, "Implementasi REST API untuk Komunikasi Antara ReactJS dan NodeJS (Studi Kasus: Modul Manajemen User Solusi247)", *Automata*, vol. 02, January 2021.
- [7] Bulelengkab, B, "Rencana Strategis", Badan Perencanaan Pembangunan Daerah, 20 September 2017. [Online]. Available: <https://bappeda.bulelengkab.go.id/informasi/detail/artikel/pengertian-renstra-68>. [Accessed 13 June 2022].
- [8] Kearsipan, D. P, "Rencana Strategis", Dispersip, Januari 2017.

- [9] Pahlevy, B. A., "Rancang Bangun Application Programming Interface (API) Ongkos Kirim Dan Tracking Ekspedisi Indonesia", *Universitas Muhammadiyah Malang*, December 2017.
- [10] Hasanuddin, Asgar, H., & Hartono, B., "Rancang Bangun REST API Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan", *Jurnal Informatika, Teknologi Dan Sains*, vol. 4(1), February 2022
- [11] Soni, A., & Ranga, V., "API features individualizing of web services: REST and SOAP". *International Journal of Innovative Technology and Exploring Engineering*, vol. 08, July 2019.
- [12] Rizki, K., & Adil, A., "Implementasi Google Maps API Berbasis Android untuk Lokasi Fasilitas Umum di Kabupaten Sumbawa". *Jurnal MATRIK*, vol. 17, May 2018.
- [13] Choirudin, R., & Adil, A., "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa". *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, vol 18, May 2019.