

## PENGEMBANGAN SERVIO MENGGUNAKAN *FULL REST API* UNTUK MENDUKUNG LAYANAN *MULTIPLATFORM*

Oleh:

Ade Ivan Priyatna<sup>1\*</sup>, Hamdan Arfandy<sup>2</sup>, Hendra Surasa<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, STMIK Kharisma Makassar

e-mail: <sup>1</sup>[adeivanpriyatna\\_18@kharisma.ac.id](mailto:adeivanpriyatna_18@kharisma.ac.id), <sup>2</sup>[hamdanarfandy@kharisma.ac.id](mailto:hamdanarfandy@kharisma.ac.id),  
<sup>3</sup>[hendras0366@gmail.com](mailto:hendras0366@gmail.com)

**Abstrak:** Penelitian ini bertujuan untuk menerapkan FullREST API pada SERVIO sehingga memberi kemudahan dalam pengembangan program ke platform lain. Penelitian ini menggunakan metode prototype sebagai permodelan untuk pembuatan web service yang akan terintegrasi dengan sistem SERVIO. Hasil dari pengujian menunjukkan API sudah layak digunakan berdasar pada penelitian dimana respon time dan data yang dihasilkan berbanding lurus semakin banyak data yang dikirim semakin tinggi respon time yang dibutuhkan, untuk jumlah request secara bersamaan didapatkan maksimal 30 request dalam satu waktu.

**Kata kunci:** FullREST API, Servio, Multiplatform, JSON, Web Server.

**Abstract:** The purpose of this research is to apply FullREST API to the SERVIO that make it easier developing progress to other platform. This research uses the prototype method as a model for making web services that will be integrated with the SERVIO system. The results of the test show that the API is feasible to use based on research where the response time and data generated are directly proportional the more data sent need higher of the response time, for the number of requests simultaneously obtained a maximum of 30 requests at one time.

**Keywords:** Full REST API, Servio, Multiplatform, JSON, Web Server.

### 1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi saat ini telah berkembang sangat pesat. Perkembangan ini telah memicu banyak perubahan secara digitalisasi baik dalam sektor industri maupun sektor lainnya. Dalam sektor industri, teknologi informasi memberikan kemudahan mulai dari proses pemasaran produk hingga transaksi jual beli dapat dilakukan dalam satu sistem. Sama halnya yang terdapat di SERVIO.

*Service Eletronica Online* (SERVIO) adalah sebuah aplikasi yang dapat mempermudah akses informasi untuk melakukan servis elektronik secara online. Servio juga membantu orang-orang yang ingin melakukan perbaikan ataupun perawatan terhadap barang elektroniknya tanpa harus mengunjungi tempat servis secara langsung.

Saat ini SERVIO masih berbasis pada *platform web* sehingga user perlu usaha lebih untuk mengakses atau membuka aplikasinya, berdasarkan observasi yang dilakukan melalui kuisisioner, 20 user yang mendaftar pada *platform web* memberikan komentar sama bahwa

\* Corresponding author : Ade Ivan Priyatna ([adeivanpriyatna\\_18@kharisma.ac.id](mailto:adeivanpriyatna_18@kharisma.ac.id))

akan lebih baik jika dibuat dalam versi android agar mudah di akses dan efisien untuk digunakan. Berdasarkan masukan tersebut maka penulis memikirkan sebuah masalah yaitu bagaimana menerapkan SERVIO ke *platform* yang lain tanpa melakukan pengkodean ulang fungsi-fungsi inti. Adapun fungsi inti yang dimaksud ialah fitur-fitur penting yang dibutuhkan untuk pemesanan produk.

Untuk mengatasi hal tersebut, maka perlu diterapkan FullREST API pada SERVIO sehingga memberi kemudahan dalam pengembangan program ke platform lain. REST (REpresentational State Transfer) merupakan sebuah arsitektur web service yang berbasiskan http verbs dalam mengolah data. Http verbs yang digunakan pada REST berupa GET, POST, PUT, DELETE, dan lainnya [1]. REST tidak memerlukan parsing XML dan tidak memerlukan sebuah header pesan ke dan dari penyedia layanan. Sedangkan API (*application programming interface*) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi ataupun bahasa pemrograman tertentu yang dapat digunakan oleh programmer saat membangun perangkat lunak [2].

Adapun RESTful web service atau juga dikenal dengan nama RESTful Web API merupakan sebuah web service yang di implementasikan dengan menggunakan http dengan menggunakan prinsip-prinsip REST. Service yang digunakan menggunakan method milik http antara lain GET, PUT, POST or DELETE [2]. Pada rancangan tersebut, *API Server* akan terkoneksi langsung ke database sebagai sumber data, lalu *API client* atau platform seperti android, IOS dan lainnya hanya perlu memanggil url dari API tersebut. Selain itu, framework yang digunakan dalam merancang sistem ini yaitu Codeigniter.

Menurut [3] *Codeigniter* adalah sebuah *framework* yang menggunakan bahasa pemrograman PHP sebagai basenya dan bersifat *opensource*. Codeigniter dibangun dengan berbagai fitur dengan tujuan agar developer tidak perlu untuk melakukan pengkodean dari awal.

*Web service* yang merupakan bentuk tingkatan ketiga dari ASP (*Application Service Provider*). Pada tingkat ini komponen perangkat lunak akan diintegrasikan dengan *software* lain. [4]. Pada umumnya *Web service* identic dengan bahasa pemrograman PHP. Dalam sebuah *Web service* juga mempunyai berbagai format data untuk pertukaran seperti XML, JSON, SOAP dan lain-lain [5]. Adapun format data yang digunakan ialah JSON.

JSON dikenal sebagai standar pertukaran data, yang secara subtekstual menyiratkan bahwa ia dapat digunakan sebagai format data di mana pun terjadi pertukaran [6]. Pertukaran data dapat terjadi antara browser dan server dan bahkan server ke server, dalam hal ini. Tentu saja, ini bukan satu-satunya cara yang mungkin untuk menukar JSON, dan membiarkannya di kedua cara itu agak membatasi. Keunggulan JSON dibanding XML diantaranya ukuran karakter yang dibutuhkan JSON lebih kecil, proses browser parsing JSON lebih sederhana, dan pertukaran data JSON lebih unggul dibandingkan XML [7].

**2. METODOLOGI PENELITIAN**

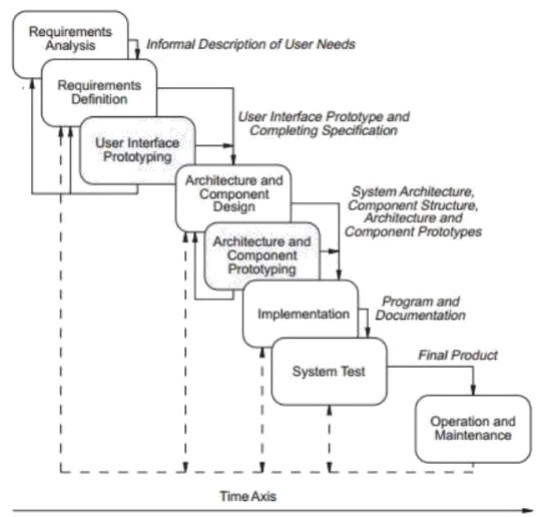
**A. Jenis Data dan Sumber Data**

Jenis data yang digunakan dalam penelitian ini ialah data kualitatif dimana data dan informasi diperoleh bukan dalam bentuk angka serta menggunakan dua sumber data yaitu sumber data primer dan sekunder.

1. Untuk data primer penulis mendapatkan data dari pengguna web SERVIO yang telah dihosting sebelumnya, beberapa pengguna memberikan beberapa masukan dan pendapat terkait dengan penelitian yang dilakukan.
2. Untuk sumber data sekunder penulis melakukan studi literatur baik itu melalui jurnal, buku ataupun website yang dapat dipercaya.

**B. Tahapan Penelitian**

Penelitian ini menggunakan metode *prototype* sebagai permodelan untuk pembuatan web service yang akan terintegrasi dengan sistem SERVIO adapun alur dari prototype dapat dilihat pada Gambar 1[8].



Gambar 1. Alur Proses Metode Prototype [Sumber: Afghan A., 2015]

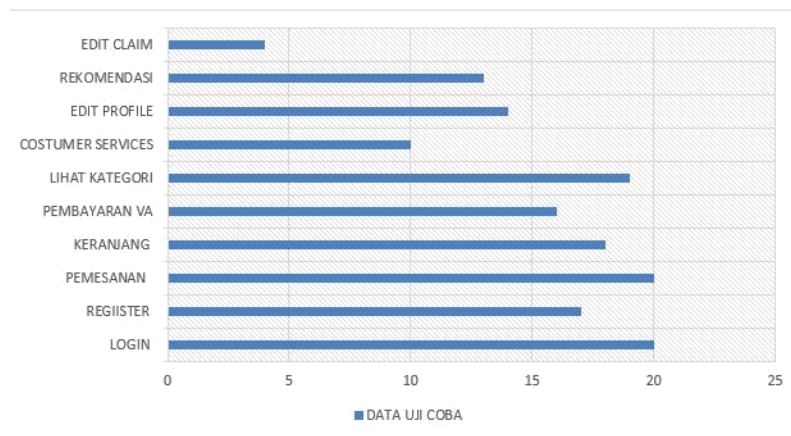
Dalam tulisan [9] metode ini akan sering melibatkan *developer* dan *tester* (pengguna) sehingga sangat cocok mengingat kebutuhan yang dibutuhkan tester belum diketahui pasti seperti model data json, jumlah array per json dan lainnya. Adapun tahapan dalam prototype sebagai berikut. Langkah pertama, dilakukan analisis kebutuhan. Kebutuhan yang dimaksudkan adalah fungsi yang akan dibuat kedalam REST API. Untuk tahap kedua dilakukan perancangan yang mudah dimengerti oleh pengguna setelah perancangan selesai dilanjutkan dengan pengkodean, pada tahapan ini bentuk prototype akan diterjemahkan kedalam bahasa pemrograman yang diinginkan (Program contoh atau sudah pada tahap diatas 50 %) selanjutnya tahapan Evaluasi, program yang telah dihasilkan akan diuji sesuai kebutuhan dasar. jika pada tahapan evaluasi terdapat masukan atau

kekurangan maka akan dilengkapi sesuai dengan masukan yang diajukan tester pada tahap perbaikan prototype sebelum akhirnya dinyatakan siap digunakan.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Analisis Kebutuhan

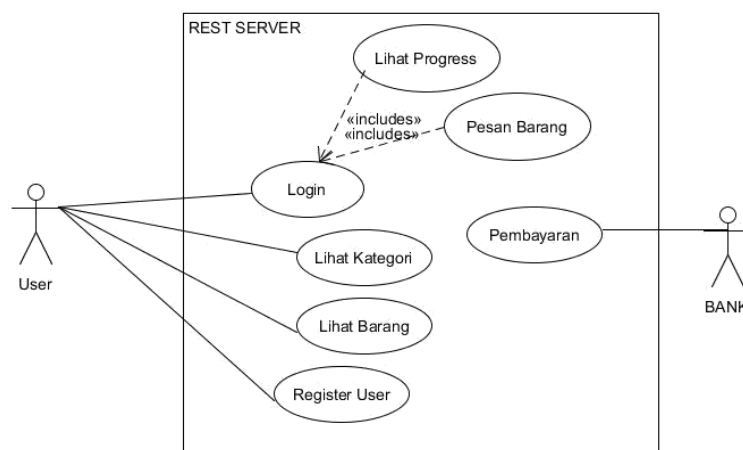
Untuk menentukan fungsi yang akan dibuat pada REST API penulis mewawancarai 20 user yang telah mencoba servio di platform web, data tersebut kemudian diolah menggunakan Ms Excel untuk melihat angka pemilihan secara lebih rinci dan diperolehlah fungsi-fungsi inti yang akan dihadirkan ke platform mobile, pemilihan tersebut dapat dilihat pada Gambar 3.



Gambar 3. Diagram Pemilihan Fitur

#### 3.2. Perancangan Prototype

Tahapan ini memuat rancangan yang nantinya akan dilihat oleh tester (pengguna) dan penyajian data yang telah dianalisis diutakan kedalam bentuk yang mudah dimengerti.



Gambar 4. Rancangan Usecase Rest Server

Pada Gambar 4 diatas terdapat dua aktor yang akan menggunakan layanan API tersebut, user selaku platform selain web akan membutuhkan fungsi-fungsi dasar sebuah aplikasi

sedangkan untuk fitur tambahan yang diinginkan pengguna akan digunakan oleh bank guna melakukan request data pembayaran.

### 3.3. Implementasi Sistem

Dalam pembuatan API penulis menggunakan bahasa pemrograman PHP dengan framework Codeigniter memanfaatkan *library fullRESTAPI*, pemilihan bahasa diatas berdasar pada tulisan [10] dia menyebutkan framework ini mudah digunakan dan sudah banyak diimplementasikan oleh programmer diluar sana serta kemudahan dalam penanganan masalah (*Error Handling*) karena forum dan dokumentasi yang melimpah.



```
$.ajax({  
  url:'http://servio.suksessejuk.id/servio_api/kategori/?servio=servio123',  
  type:"post",  
  data:new FormData(this),  
  processData:false,  
  contentType:false,  
  cache:false,  
  async:false,  
  success: function(data){  
    $("#loaddata").html(data)  
  }  
});
```

Gambar 5. Hasil Pemanggilan REST API

Gambar 5 menampilkan hasil dari pemanggilan *REST API* yang telah digunakan menggunakan jquery sebagai *API CLIENT* server sudah mampu mengembalikan data json sesuai dengan kondisi yang seharusnya. Setelah tahapan implementasi diperoleh titik akses dan parameter yang seharusnya digunakan, berikut url dan parameter yang dibutuhkan dalam menggunakan *API SERVIO*.

- Fungsi pembayaran

http://servio.suksessejuk.id/servio\_api/?servio= &date=&va= dengan menggunakan method GET. Dan action seperti inquiry, reversal dan payment akan diseleksi oleh system berdasarkan parameter tambahan.

- Fungsi login

http://servio.suksessejuk.id/servio\_api/login/?servio= &email= &pass= dengan menggunakan method GET.

- Fungsi registrasi user

http://servio.suksessejuk.id/servio\_api/register/?servio=&email=&pass=&alamat= &nohp= dengan menggunakan method POST.

- Fungsi pemesanan barang

http://servio.suksessejuk.id/servio\_api/pesan/?servio=&email= &id\_barang=29&kerusakan=

- Fungsi lihat barang

http://servio.suksessejuk.id/servio\_api/barang/?servio=&id\_kat=24&from=0&page=5

- Fungsi kategori

http://servio.suksessejuk.id/servio\_api/kategori/?servio=

- Fungsi Progress

http://servio.suksessejuk.id/servio\_api/vprogress/?servio=&email= per\_page=5&from=0

### 3.4 Pengujian Sistem

Untuk pengujian *system* sendiri dibutuhkan dua perangkat lunak untuk menguji *response time* guna menentukan apakah API ini layak atau tidak, perangkat lunak tersebut ialah POSTMAN dan APACHE Jmeter, POSTMAN digunakan untuk menentukan atau memvalidasi response data dari setiap fungsi yang diuji sedangkan APACHE Jmeter digunakan untuk mengetahui seberapa banyak request yang dapat dilayani dengan memperhatikan jumlah byte data dan sample time yang dihasilkan.

Tabel 1: Pengujian Request Fungsi

Sample	Function	Time
1	Get va running pertama	300 ms
2	Get va running ke 2	329 ms
3	Login running pertama	686 ms
4	Login running kedua	354 ms
5	Login running ketiga data tidak valid	498 ms
6	Get kategori	176 ms

Untuk pengujian pertama dapat dilihat di Tabel 1 kita menggunakan POSTMAN untuk beberapa fungsi response time rata-rata diangka 390 ms , pada penelitian (M.I Perkasa,dkk: 2018) respon time dengan data berupa *attribute* masyarakat seperti NIK, tempat tanggal lahir dan semacamnya membutuhkan waktu sekitar 168 ms untuk *single request* dan 1032 ms untuk 100 *request* secara bersamaan, jika dibandingkan dengan hasil pengujian pada penelitian ini angka tersebut sudah masuk dalam kategori baik.

Tabel 2: Pengujian Akses Perdetik

NO OF REQ/ S	REQUEST	SAMPLE TIME	STATUS	BYTE					
1	HTTP Request	132	Success	724	26	HTTP Request	1763	Success	724
2	HTTP Request	197	Success	724	27	HTTP Request	2349	Success	724
3	HTTP Request	412	Success	724	28	HTTP Request	2569	Success	724
4	HTTP Request	565	Success	724	29	HTTP Request	2592	Success	724
5	HTTP Request	598	Success	724	30	HTTP Request	2499	Success	724
6	HTTP Request	670	Success	724	31	HTTP Request	2114	Success	724
7	HTTP Request	857	Success	724	32	HTTP Request	1908	Success	724
8	HTTP Request	827	Success	724	33	HTTP Request	2196	Success	724
9	HTTP Request	921	Success	724	34	HTTP Request	2540	Success	724
10	HTTP Request	1031	Success	724	35	HTTP Request	2404	Success	724
11	HTTP Request	1070	Success	724	36	HTTP Request	2748	Success	724
12	HTTP Request	1199	Success	724	37	HTTP Request	2604	Success	724
13	HTTP Request	2088	Success	724	38	HTTP Request	2169	Success	724
14	HTTP Request	2306	Success	724	39	HTTP Request	2601	Success	724
15	HTTP Request	2157	Success	724	40	HTTP Request	2984	Success	724
16	HTTP Request	1954	Success	724	41	HTTP Request	3127	Success	724
17	HTTP Request	2298	Success	724	42	HTTP Request	3061	Success	724
18	HTTP Request	2260	Success	724	43	HTTP Request	3306	Success	724
19	HTTP Request	2328	Success	724	44	HTTP Request	3399	Success	724
20	HTTP Request	2394	Success	724	45	HTTP Request	3713	Success	724
21	HTTP Request	2221	Success	724	46	HTTP Request	3717	Success	724
22	HTTP Request	2346	Success	724	47	HTTP Request	4351	Success	724
23	HTTP Request	2342	Success	724	48	HTTP Request	4323	Success	724
24	HTTP Request	2488	Success	724	49	HTTP Request	4482	Success	724
25	HTTP Request	2447	Success	724	50	HTTP Request	4115	Success	724

Untuk testing menggunakan APACHE Jmeter pada Tabel 2 server dari servio dapat melayani setidaknya 50 request secara bersamaan dengan response time sekitar 4000 ms, tentunya ini angka yang cukup besar jika merujuk pada penelitian yang dilakukan [11] maka dari itu diperlukan pembatasan pada pengaksesan secara bersamaan dan angka yang paling aman ialah di 1-30 request atau dilakukan upgrade pada sisi server yang tentunya akan memakan biaya yang lebih besar.

## KESIMPULAN

Dari hasil penelitian dan pengembangan sistem yang dilakukan terdapat beberapa point kesimpulan diantaranya :

1. Pembuatan *fullREST API* menggunakan framework codeigniter dengan pertimbangan dokumentasi dan forum yang melimpah serta telah banyak digunakan.
2. Berdasarkan analisis menggunakan prototype ditetapkan beberapa fungsi inti yang akan dibuat seperti login, register, pemesanan, keranjang, pembayaran VA, kategori dan barang hasil dari wawancara 20 responden.
3. Hasil dari pengujian menunjukkan API sudah layak digunakan berdasar pada penelitian dimana *respon time* dan data yang dihasilkan berbanding lurus semakin banyak data yang dikirim semakin tinggi *respon time* yang dibutuhkan, untuk jumlah request secara bersamaan didapatkan maksimal 30 request dalam satu waktu dimana *average respon time* diangka 2000 ms dengan 724 BYTE data untuk mengatasi hal tersebut dilakukan pembatasan pada webserver.

## DAFTAR PUSTAKA

- [1] R. Y. MF, *Membangun REST API Service dengan Deno dan PostgreSQL*, Indonesia: Native Enterprise, 2020.
- [2] M. Putra and P. M.I.A, "Analisis Perbandingan Metode SOAP dan REST yang digunakan pada Framework Flask Untuk Membangun Web Service," *SCAN-Jurnal Teknologi Informasi dan Komunikasi*, vol. XIV No. 2, pp. 1-7, 2019.
- [3] M. Destiningrum and Q. J. Adrian, "Sistem Informasi Penjadwalan Dokter Berbassis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)," *J. Teknoinfo*, vol. 11, no. 2, pp. 30–37, 2017
- [4] T. K. Priyambodo, "IMPLEMENTASI WEB-SERVICE UNTUK PENGEMBANGAN SISTEM LAYANAN PARIWISATA TERPADU," *Semin. Nas. Apl. Teknol. Inf.*, pp. 11–19, 2005.
- [5] B. Santosa, "Analisa Dan Perancangan Web Services Untuk Sistem Informasi Universitas," *Proc. Konf. Nas. Sist. Inf.*, pp. 267–271, 2008.
- [6] B. Smith, *Beginning JSON*, Melbourne, 2015
- [7] R. Somya and T. M. E. Nathanael, "Pengembangan Sistem Informasi Pelatihan Berbasis Web Menggunakan Teknologi Web Service Dan Framework Laravel," *J.Techno Nusa Mandiri*, vol. 16, no. 1, 2019, doi: 10.33480/techno.v16i1.164.
- [8] A. A. dk. Pradipta, "Pengembangan Web E-Commerce Bojana Sari Menggunakan Metode Prototype," *J. Tugas Akhir | Fak. Rekayasa Ind.*, vol. 3, no. 1, pp. 341–347, 2019.
- [9] F. Susanto, "Sistem informasi pengolahan data pasien pada puskesmas abung pekurun menggunakan metode prototype," *J. Mikrotik*, vol. 8, no. 1, pp. 65–73, 2018.
- [10] B. W. Putra, A. Saputra, R. Sanjaya, and D. Kurniawan, "Implementasi Framework Codeigniter dan Restful API pada Sistem Informasi Manajemen Tugas Akhir," *Pros. Annu. Res. Semin.* 2019, vol. 5, no. 1, pp. 978–979, 2019.
- [11] M. I. Perkasa and E. B. Setiawan, "Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Akses Token," *Ultima Computing*, vol. X. No.1, pp. 19-26, 2018.