

# Implementation Of Web Services On University Library Information Systems

Implementasi *Web Service* Pada Sistem Informasi Perpustakaan Perguruan Tinggi

Febyana Wijaya, Agustinus Jacobus, Alwin Sambul

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia

e-mails : [17021106047@student.unsrat.ac.id](mailto:17021106047@student.unsrat.ac.id), [a.jacobus@unsrat.ac.id](mailto:a.jacobus@unsrat.ac.id), [a.sambul@unsrat.ac.id](mailto:a.sambul@unsrat.ac.id)

Received: 9 June 2021; revised:23 July 2021; accepted:23 July 2021

**Abstract** — *Libraries are important in the world of education. This shows that there is a need for a system that supports library services. The existence of a library information system makes it easier for libraries to share information and data. In order for the system to share information or data, it requires Web Service technology, to enable the exchange of such information or data. Web Service is a form of development of current technology that can be used for system integration. Web Service is able to be a bridge between various existing systems. In this research, the Web Service technology used is RESTful Web Service. RESTful is one of the architectural designs contained in the API, which is able to work well because it has high and stable performance. The purpose of this research is to produce a library information system that is capable of exchanging data with other systems by implementing a RESTful Web Service using the Django REST Framework. Django REST Framework is a tool that can be used to create Web APIs, making it easier for us to create RESTful Web APIs. So this research produces a RESTful Web Service which is implemented in a university library information system with the GET, POST, PUT, DELETE methods and from the test results obtained the amount of data from the response time on these methods.*

**Key words**— *Web Service; RESTful; Django REST Framework; Information Systems; Library.*

**Abstrak** — *Perpustakaan merupakan hal penting di dunia pendidikan. Hal ini menunjukkan bahwa perlu adanya suatu sistem yang mendukung pelayanan perpustakaan. Dengan adanya sistem informasi perpustakaan menjadikan perpustakaan lebih mudah dalam berbagi informasi dan data. Agar sistem tersebut dapat berbagi informasi atau data maka dibutuhkan teknologi Web Service, untuk memungkinkan terjadinya pertukaran informasi atau data tersebut. Web Service merupakan salah satu bentuk perkembangan dari teknologi saat ini yang dapat digunakan untuk integrasi sistem. Web Service mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada. Pada penelitian ini, teknologi Web Service yang digunakan adalah RESTful Web Service. RESTful merupakan salah satu dari desain arsitektur yang terdapat di dalam API, yang mampu bekerja dengan baik karena memiliki kinerja yang tinggi dan stabil. Tujuan dari penelitian ini adalah untuk menghasilkan sistem informasi perpustakaan yang mampu bertukar data dengan sistem yang lain dengan cara mengimplementasikan RESTful Web Service menggunakan Django REST Framework. Django REST Framework merupakan tools yang dapat digunakan untuk membuat Web API, sehingga memudahkan kita untuk membuat Web API yang RESTful. Maka penelitian ini menghasilkan RESTful Web Service yang*

*diimplementasikan pada sistem informasi perpustakaan perguruan tinggi dengan methods GET, POST, PUT, DELETE dan dari hasil pengujian didapatkan jumlah data dari response time pada methods tersebut.*

**Kata kunci** — *Web Service; RESTful; Django REST Framework; Sistem Informasi; Perpustakaan.*

## I. PENDAHULUAN

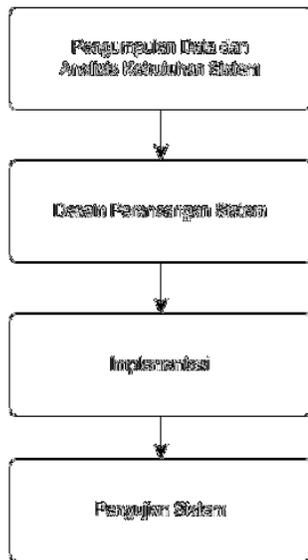
Pemanfaatan teknologi saat ini memberi pengaruh yang besar dalam kehidupan manusia. Teknologi selalu dikembangkan dengan tujuan untuk memenuhi kebutuhan penggunaannya. Pertukaran data dengan sangat cepat dan akurat merupakan salah satu peran teknologi. Salah satu bentuk penerapan teknologi berada pada sistem informasi[1]. Perpustakaan merupakan hal penting di dunia pendidikan. Hal ini menunjukkan bahwa perlu adanya suatu sistem yang mendukung pelayanan perpustakaan. Dengan adanya Sistem Informasi Perpustakaan menjadikan perpustakaan lebih mudah dalam berbagi informasi dan data sehingga dapat meningkatkan minat membaca yang mampu dilaksanakan secara efisien dan efektif[2].

Setiap perguruan tinggi memiliki perpustakaan sebagai salah satu sarana yang dapat dimanfaatkan civitas akademik perguruan tinggi tersebut[3]. Begitu pula dengan Universitas Sam Ratulangi. Namun sistem yang ada saat ini, untuk berbagi informasi dengan sistem yang lain atau belum bisa untuk dilakukan pengintegrasian data dengan sistem lain. Oleh karena itu, agar sistem tersebut dapat berbagi informasi dengan baik atau data maka dibutuhkan teknologi *Web Service*, untuk memungkinkan terjadinya pertukaran informasi atau data tersebut dari sistem yang satu kesistem yang satu ke sistem yang lainnya.

*Web Service* juga merupakan salah satu bentuk perkembangan dari teknologi saat ini yang dapat digunakan untuk integrasi sistem yang satu dengan sistem yang lainnya. *Web Service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada[4]. *Web service* memiliki *interface* yang dideskripsikan dalam format yang dapat dibaca oleh mesin. Arsitektur pada *Web Service* dapat dibangun dengan menggunakan *Simple Object Access Protocol*

TABEL I  
DAFTAR REST API

No	Layanan Web Service	HTTP Methods	Path	Deskripsi	HTTP Status
1	Buku	GET	api/buku/all	Read seluruh daftar buku	201 OK
2		POST	api/buku/search?id=..	Read data buku berdasarkan 'id'	201 Created
3		PUT	api/buku/all	Create data buku	200 OK
4		DELETE	api/buku/id=..	Update data buku	204 No Content
5	Transaksi	DELETE	api/buku/id=..	Delete data buku berdasarkan id	204 No Content
6		GET	api/trans/all	Read seluruh daftar transaksi	200 OK
7		POST	api/trans/search?id=..	Read data transaksi berdasarkan nama peminjam	201 Created
8		PUT	api/trans/id=..	Create data transaksi	200 OK
		DELETE	api/trans/id=..	Delete data transaksi berdasarkan id	204 No Content



Gambar 1 Kerangka Berpikir

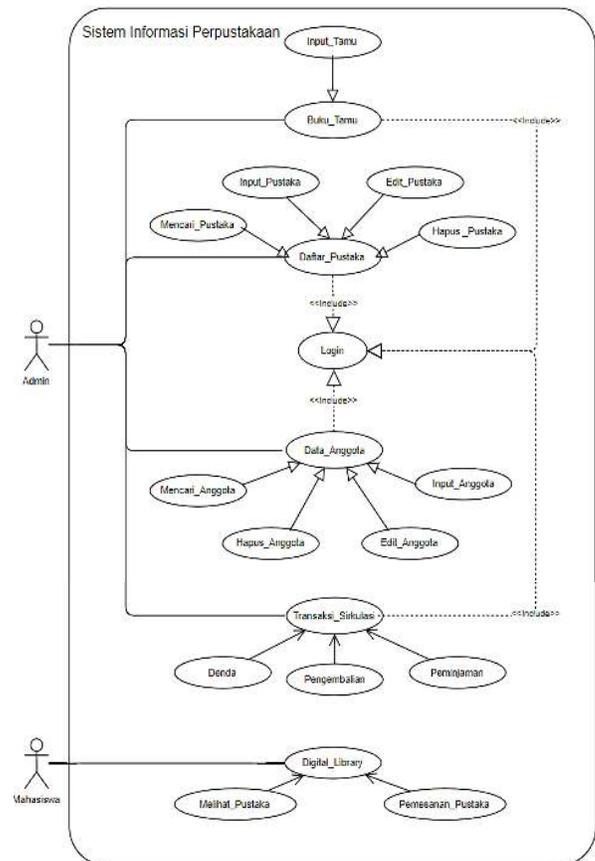
(SOAP) atau *Representational State Transfer (REST)*.

Pada penelitian ini, teknologi *Web Service* yang digunakan adalah *RESTful Web Service*. *RESTful Web Service* bekerja dengan baik karena memiliki kinerja yang tinggi dan stabil. *RESTful* merupakan salah satu dari desain arsitektur yang terdapat di dalam *API*. Kemudian, data yang diberikan oleh *REST server* adalah berupa format *text JSON*.

Oleh karena itu, penulis membuat Sistem Informasi Perpustakaan yang mampu untuk diintegrasikan atau bertukar data dengan sistem yang lain dengan cara mengimplementasikan *RESTful Web Service*.

#### A. Penelitian Terkait

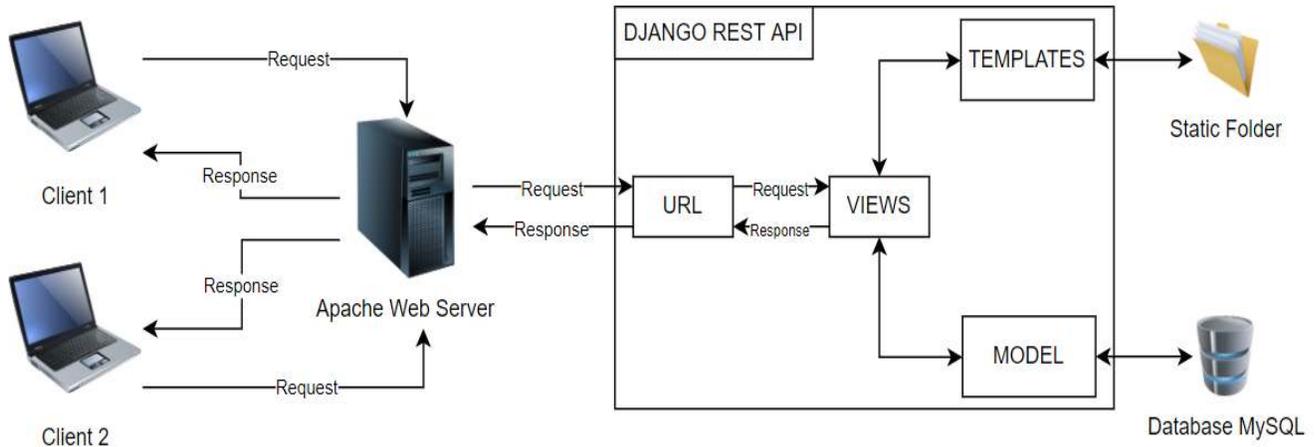
- 1) Penelitian yang dilakukan oleh Somya, dan Nathanael[5] mengenai Pengembangan Sistem Informasi Pelatihan Berbasis Web menggunakan Teknologi *Web Service* dan *Framework Laravel*. Penelitian ini mengembangkan sistem informasi pelatihan dengan menerapkan teknologi *web service* untuk integrasi data sehingga data disimpan terpusat. Berdasarkan penelitian tersebut disimpulkan bahwa Sistem Informasi Pelatihan dengan menggunakan teknologi *Web Service* berbasis website dapat berjalan



Gambar 2 Usecase Diagram

dengan baik dengan hasil pengujian dari sistem dinyatakan layak dan secara fungsional sistem menghasilkan *output* yang diharapkan.

- 2) Penelitian yang dilakukan oleh Saputra, dan Aji[6] mengenai Analisis Perbandingan Performa *Web Service Rest* menggunakan *Framework Laravel*, *Django*, dan *Ruby on Rails* untuk Akses Data dengan Aplikasi *Mobile*. Penelitian ini membahas tentang *web framework* yang tepat dengan metode *rest* yang dibuat dengan tiga *framework* berbeda untuk integrasi data. Berdasarkan penelitian tersebut disimpulkan bahwa dari tiga *framework* tersebut, *Web Service* menggunakan *framework Django* diprediksi memiliki waktu respon yang cepat, penggunaan *CPU* dan memori yang paling



Gambar 3 Arsitektur Sistem

```

1  {
2    "ISBN": "kode ISBN buku",
3    "BhsId": "nomor id bahasa buku",
4    "NoKlasifikasi": "nomor klasifikasi buku",
5    "GolId": "nomor id golongan buku",
6    "Judul": "judul buku lengkap",
7    "EntriUtama": "penulis utama buku",
8    "EntriTambahan": "penulis buku lainnya",
9    "Edisi": "edisi buku",
10   "Kota": "kota buku diterbitkan",
11   "Tahun": "tahun terbit buku",
12   "Penerbit": "penerbit buku",
13   "Deskripsi": "berisi deskripsi buku",
14   "JnsId": "berisi jenis buku",
15   "Subyek": "berisi subyek keterangan buku",
16   "CatatanUmum": "catatan tambahan penting buku",
17   "Bibliografi": "bibliografi buku",
18   "Abstraksi": "abstraksi buku",
19   "PathCover": "cover buku",
20   "Jilid": "keterangan jilid buku",
21   "MataKuliah": "keterangan buku mata kuliah"
22 }
    
```

Gambar 4 Skema JSON POST Buku

```

1  {
2    "id": "id transaksi",
3    "tanggal_buat": "tanggal",
4    "tanggal_pinjam": "tanggal",
5    "tanggal_kembali": "tanggal",
6    "peminjam": "nama peminjam",
7    "buku": "judul buku",
8    "status": "status transaksi"
9  }
    
```

sedikit sehingga penulis merekomendasikan untuk menggunakan *framework Django*  
 Gambar 5 Skema JSON POST Transaksi

Iriani[7] mengenai *RESTful Web Service* untuk Sistem Pencatatan Transaksi. Penelitian ini membahas mengenai arsitektur *RESTful Web Service* yang memanfaatkan keamanan dengan menerapkan *JWT* untuk menunjang distribusi data pada sistem yang berbeda. Penelitian ini menghasilkan sebuah arsitektur *RESTful Web Service* yang aman bagi PT. XYZ, yang dibangun menggunakan *JSON Web Token (JWT)* dalam mengamankan komunikasi yang terjadi. Dan disimpulkan sangat memungkinkan bagi PT. XYZ dalam mengintegrasikan

```

1  {
2    "Judul": "Matematika Terapan Bagian 2"
3  }
    
```

Gambar 6 Skema JSON PUT Buku

sumberdaya dari sistem menggunakan aplikasi yang berbeda dan dapat diintegrasikan dengan jalur komunikasi yang aman dengan memanfaatkan *RESTful Web Service*.

- Penelitian yang dilakukan Ridwan, dan Maisura[8] mengenai *Design And Implementation Of Web Services For The Integration Systems For The Distribution Of Covid-19 Cases*. Penelitian ini merancang dan membangun suatu Sistem Informasi Penyebaran yang dapat digunakan secara bersamaan antara pihak posko desa, kantor kecamatan, puskesmas, dan rumah sakit. Penelitian ini berhasil menghasilkan *prototype* Sistem Informasi Penyebaran kasus Covid-19 yang dapat digunakan secara bersamaan antara pihak posko desa, kantor kecamatan, puskesmas, dan rumah sakit umum, *prototype* aplikasi pada pemerintah kabupaten/kota dan *API Web Service* berbasis *REST* menggunakan format

```

Body  Cookies  Headers (7)  Test Results  Status: 200 OK  Time: 2.51 s  Size: 8.35 MB
Pretty  Raw  Preview  Visualize  JSON
Body  Cookies  Headers (7)  Test Results  Status: 201 Created  Time: 520 ms  Size: 695 B
Pretty  Raw  Preview  Visualize  JSON
    
```

```

1  {
2    "id": "26163",
3    "ISBN": "abc123",
4    "BhsId": "2",
5    "NoKlasifikasi": "129875",
6    "GolId": "1",
7    "Judul": "Ilmu Psikologi",
8    "EntriUtama": "Sabrina. N.",
9    "EntriTambahan": "Kumaat. Ever.",
10   "Edisi": "eed",
11   "Kota": "Manado",
12   "Tahun": "2021",
13   "Penerbit": "erlangga",
14   "Deskripsi": "iii. 28 hl.. 29 cm",
15   "JnsId": "2",
16   "Subyek": "Pengembangan android dengan android studio",
17   "CatatanUmum": "afsf",
18   "Bibliografi": "sdgd",
19   "Abstraksi": "4910",
20   "PathCover": "fsdf",
21   "Jilid": "sdfdsf",
22   "Tanggal": "2021-07-19",
    
```

Gambar 8 Output POST

TABEL II  
PENGUJIAN SISTEM MENGGUNAKAN *POSTMAN*

No	Skenario Pengujian	Kasus Pengujian	Hasil yang diharapkan	Hasil Pengujian
1	<i>GET</i>	Masukkan <i>path</i> <code>http://127.0.0.1:8000/api/buku/all</code> , lalu klik tombol “Send”	Mendapatkan respon kode status 200 OK, dengan ditampilkan seluruh seluruh data	Sesuai
2	<i>GET</i>	Masukkan <i>path</i> <code>http://127.0.0.1:8000/api/buku/search?id='d'</code> , dengan ‘id’ yang ingin di request, klik tombol “Send”	Mendapatkan respon kode status 200 OK, dengan ditampilkan data yang di request	Sesuai
3	<i>POST</i>	Masukkan <i>path</i> <code>http://127.0.0.1:8000/api/buku/all</code> , masukkan key dan value yang ingin ditambahkan sesuai dengan form pada bagian Body, klik tombol “Send”	Berhasil menambahkan data, dengan respon kode status 201 <i>Created</i>	Sesuai
4	<i>PUT</i>	Masukkan <i>path</i> <code>http://127.0.0.1:8000/api/buku/id='id'</code> , dengan ‘id’ yang ingin diubah masukkan key dan value yang ingin diubah pada bagian Body, klik tombol “Send”	Berhasil mengubah data, dengan respon kode status 200 OK	Sesuai
5	<i>DELETE</i>	Masukkan <i>path</i> <code>http://127.0.0.1:8000/api/buku/id='id'</code> , dengan ‘id’ yang ingin dihapus, klik tombol “Send”	Berhasil menghapus data, dengan respon kode status 204 <i>No Content</i>	Sesuai

```

12  "Tahun": 2021,
13  "Penerbit": "2296",
14  "Deskripsi": "iii. 28 hlm.. 29 cm",
15  "JnsId": "2",
16  "Subyek": "Matematika Terapan",
17  "CatatanUmum": "afsf",
18  "Bibliografi": "sdgd",
19  "Abstraksi": "4910",
20  "PathCover": "fsdf",
21  "Jilid": "sdfdsf",
22  "Tanggal": "2021-07-12",
23  "MataKuliah": "dsfsdgsdf"
24

```

waktu (waktu), 500 server error (kesalahan pada server atau resource)[10].

TABEL III  
 SPESIFIKASI API

No	Method Type	Parameter	Message Type	Return Value	Access Token	Request URL
1	GET	Pathparam	string	JSON	required	http://127.0.0.1:8000/api/buku/
2	POST	Pathparam	string	JSON	required	http://127.0.0.1:8000/api/buku/all
3	PUT	Pathparam	string	JSON	required	http://127.0.0.1:8000/api/buku/id
4	DELETE	Pathparam	string	JSON	required	http://127.0.0.1:8000/api/buku/id

### E. Access Token

Access token merupakan sebuah *string unique* dan acak yang berisi angka dan huruf. Access token adalah sebuah kunci yang dibutuhkan oleh pengirim layanan yang harus digunakan untuk mengidentifikasi dan memverifikasi apakah pengirim memiliki hak akses dalam *web service*. Masing-masing pemilik akun pada layanan *web service* memiliki sebuah *access token* yang bersifat unik. Access token ini berfungsi untuk menambah tingkat keamanan pada *web service*, sehingga yang dapat mengaksesnya yang memiliki hak akses saja.

### F. Django

Django adalah *web framework* Python yang didesain untuk membuat aplikasi web yang dinamis, kaya fitur, dan aman. Django membantu membuat website dengan berbagai fiturnya untuk mencegah para *developer* menulis kode python yang sama berulang kali. Django juga fokus untuk membuat *frameworknya* cepat, aman dan mudah dikembangkan (*scalable*). Django memiliki sistem *ORM (Object relational Model)* untuk mengolah data, sistem *Views* mengatur logika dan sistem *templating* di bagian tampilannya.

### G. JSON (JavaScript Object Notation)

JSON (*Javascript Object Notation*) adalah sebuah format data yang digunakan untuk pertukaran dan penyimpanan data antara *server* dengan *client*. Filenya hanya memuatkan teks dan berekstensi *json*, format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan bahasa yang umum digunakan oleh programmer. Oleh karena itu, JSON ideal sebagai bahasa pertukaran data.

### H. JSON Web Token (JWT)

JSON Web Token adalah keamaan berbasis *JSON token encoding* yang memungkinkan identitas dan keamanan informasi untuk dibagikan di seluruh domain keamanan. Hal ini memungkinkan *client* untuk mendapatkan *token* dengan memberikan *username* dan *password* mereka. Kelebihan dari JWT adalah bersifat *stateless*.

### I. Postman

Postman adalah sebuah aplikasi yang berfungsi sebagai *REST CLIENT* untuk uji coba *REST API* dengan berbagai fungsi di dalamnya seperti *GET, POST, PUT, DELETE*. Postman biasa digunakan oleh *developer* pembuat *API* sebagai *tools* untuk menguji *API* yang telah dibuat.

## II. METODE

### A. Kerangka Berpikir

Pada Gambar 1 menjelaskan yaitu, sebagai berikut :

- 1) Tahap pertama yang dilakukan adalah melakukan identifikasi dan mengumpulkan data yang dibutuhkan. Tahap pengumpulan data ini dilakukan dengan cara studi pustaka dengan mencari informasi-informasi yang relevan dengan objek yang akan diteliti melalui buku dan jurnal *online* di internet. Kemudian studi literatur dari hasil penelitian karya ilmiah maupun jurnal tentang objek terkait. Serta observasi terhadap sistem yang sudah ada agar bisa menganalisa dan menentukan rancangan sistem yang baru.
- 2) Tahap kedua yaitu desain perancangan sistem. Tahap ini dilakukan desain perancangan sistem berdasarkan hasil analisis menggunakan *UML (Unified Modelling Language)* yang terdiri dari *use case diagram, sequence diagram, dan activity diagram*. *UML* adalah sebuah standar yang bertujuan untuk merancang, memvisualisasikan dan mendokumentasikan suatu sistem perangkat lunak. Tambahan desain lainnya yaitu perancangan tabel *database*, daftar *rest api web service* yang dibuat, skema *json*, dan perancangan *interface*.
- 3) Tahap ketiga yaitu tahap implementasi rancangan sistem. Tahap ini dilakukan implementasi rancangan sistem menggunakan *framework Django* baik *frontend* maupun *backend*, *bootstrap* sebagai desain untuk tampilan.
- 4) Tahap keempat dilakukan pengujian sistem dan pengujian untuk *web service* yang dibuat. Untuk pengujian *restful api* digunakan aplikasi *Postman* sebagai media uji coba.

### B. Arsitektur Sistem

Pada gambar 3 menjelaskan arsitektur sistem sebagai hal yang mendefinisikan dan menggambarkan komponen yang lebih spesifik dan terstruktur dengan tujuan agar kebutuhan yang dirancang sesuai dengan kebutuhan. Gambar diatas terbagi menjadi tiga bagian utama yaitu *database, rest api, dan client*. *Client* melakukan *request* kepada *rest api*, kemudian *rest api* menyediakan fungsi-fungsi yang bisa diakses *client*, dan *database* adalah tempat penyimpanan data.

### C. Desain Sistem

Gambar 2 yaitu *use case* yang menjelaskan bagaimana sistem ini berjalan. Sistem ini memiliki tiga aktor, yaitu admin, petugas, dan mahasiswa. Admin merupakan orang yang memiliki tanggung jawab penuh pada keseluruhan kerja sistem, dimana admin memiliki hak akses pada setiap fitur yang ada. Petugas adalah orang yang telah terdaftar. Petugas

TABEL IV  
PENGUJIAN RESPONSE TIME PADA BUKU

PERCOBAAN	GET	POST	PUT	DELETE
1	1834 ms	1399 ms	1897 ms	192 ms
2	1841 ms	1964 ms	511 ms	1919 ms
3	1825 ms	1990 ms	199 ms	1923 ms
4	2100 ms	429 ms	197 ms	1921 ms
5	1824 ms	1880 ms	1910 ms	428 ms
6	2290 ms	1867 ms	14 ms	399 ms
7	1872 ms	1895 ms	1953 ms	492 ms
8	1855 ms	1865 ms	1911 ms	492 ms
9	2100 ms	1833 ms	142 ms	1911 ms
10	1958 ms	422 ms	141 ms	1886 ms
RATA- RATA	1949.9 ms	1554.4 ms	887.5 ms	1156.3 ms

berperan menjalankan sistem ini dan dapat mengakses daftar pustaka, buku tamu, transaksi sirkulasi pustaka, dan usulan pengadaan pustaka. Mahasiswa merupakan orang yang dapat melihat daftar pustaka, melakukan pemesanan pustaka, serta melakukan *upload* mandiri pustaka baru. Sebelum masuk ke dalam sistem user harus melakukan *login* terlebih dahulu. Saat melakukan *login* sistem akan mengotentikasi dan menentukan halaman mana yang akan dituju tiap user. Admin dan petugas akan diarahkan ke tampilan *dashboard* sistem, sedangkan mahasiswa akan diarah ke tampilan digital *library*.

#### D. Daftar REST API

Daftar *REST API* dapat dilihat pada Tabel 1, yang menjelaskan tentang deskripsi dari Layanan *Web Service* apa saja yang tersedia, kemudian *methods* apa saja yang dan *path* yang digunakan, serta status *HTTP* yang ditampilkan.

#### E. Desain Skema JSON

##### 1) Skema JSON POST Buku

Gambar 4 ini format *JSON* yang digunakan untuk dikirim ketika melakukan *method POST* untuk *input* data baru. Diletakkan pada bagian *body* kemudian *raw*, kemudian masukkan format seperti pada gambar 4.

##### 2) Skema JSON POST Transaksi

Gambar 5 ini format *JSON* yang digunakan untuk dikirim ketika melakukan *method POST* untuk *input* data transaksi baru. Diletakkan pada bagian *body* kemudian *raw*, kemudian masukkan format seperti pada Gambar 5.

##### 3) Skema JSON PUT Buku

*Method PUT* digunakan untuk mengubah data yang sudah ada. Untuk menggunakannya *request* yang dikirim adalah berupa *key*/atribut dan *value*. Pada Gambar 6 merupakan contoh format *JSON* untuk mengubah data, dengan cara menuliskan atribut (*key*) apa yang akan diganti dan *value* baru.

##### 4) Skema JSON PUT Transaksi

Untuk mengubah data pada transaksi terlebih dahulu mengisi *path* transaksi berdasarkan *id* data yang ingin diganti, kemudian menuliskan atribut (*key*) apa yang akan diganti dan mengisi *value* baru.

### III. HASIL DAN PEMBAHASAN

#### A. REST Authentication

Setiap melakukan permintaan atau *request* membutuhkan header autentikasi. Jadi sebelum melakukan permintaan harus

TABEL V  
PENGUJIAN RESPONSE TIME PADA TRANSAKSI

PERCOBAAN	GET	POST	PUT	DELETE
1	17 ms	1981 ms	522 ms	1892 ms
2	11 ms	1895 ms	423 ms	439 ms
3	10 ms	522 ms	172 ms	178 ms
4	19 ms	454 ms	225 ms	207 ms
5	12 ms	1945 ms	510 ms	147 ms
6	10 ms	428 ms	183 ms	242 ms
7	19 ms	162 ms	415 ms	171 ms
8	12 ms	429 ms	162 ms	150 ms
9	12 ms	431 ms	411 ms	171 ms
10	69 ms	159 ms	433 ms	157 ms
RATA- RATA	19.1 ms	840.6 ms	345.6 ms	375.4 ms

memasukkan *token* akses yang baru. bila tidak memasukan autentikasi maka akan menampilkan pesan “*authentication credentials were not provided.*”.

Terdapat dua akses *token*, yaitu pertama *refresh token* yang digunakan untuk me-*refresh token* untuk mendapatkan *token* akses yang baru ketika *token* akses kita sudah tidak valid. Kedua kita mendapatkan *access token* yang digunakan saat ingin mengakses data perlu memasukkan *token* akses ini.

Bila akses *token* yang digunakan sudah tidak valid sehingga kode status yang didapatkan yaitu 401 *Unauthorized*. Untuk dapat mengakses kembali data, kita harus mendapatkan *token* yang baru dengan melakukan *refresh token* dengan menggunakan *refresh token* yang didapat.

#### B. Spesifikasi API

*RESTful Web Service* yang dibuat pada penelitian ini memiliki 4 *methods*, yaitu *GET*, *POST*, *PUT*, dan *DELETE*. Fungsi *Method GET* adalah untuk menampilkan data. *Method POST* dilakukan untuk menambahkan data buku atau data transaksi. *Method PUT* dilakukan untuk mengubah data buku atau data transaksi. *Method DELETE* dilakukan untuk menghapus data buku atau data transaksi.

##### 1) GET

Pada Tabel 3 berisikan deskripsi mengenai *input GET*, untuk melakukan permintaan mengambil seluruh data dan mencari data berdasarkan *id*.

Pada Gambar 7 merupakan contoh penggalan *response* yang diberikan ketika melakukan *request GET* seluruh data ditampilkan dalam bentuk format *JSON* dengan status kode 200 OK dengan memasukan *input HTTP request* <http://127.0.0.1:8000/api/buku/all>.

##### 2) POST

Pada bagian *method type POST* dari tabel 3 berisikan deskripsi untuk setiap atribut pada saat ingin menambahkan data dengan menggunakan *method POST*.

Ketika kita melakukan *POST* yang sesuai dengan tipe data tiap *field* maka akan menghasilkan *output* seperti pada gambar 8 yang merupakan *response* setelah berhasil menambah data baru, dengan status kode 201 *Created*.

##### 3) PUT

Gambar 9 merupakan tampilan untuk mengubah data yang sudah dengan memasukkan *key* yang ingin diganti dan *value* baru.

Gambar 10 merupakan *response* yang diterima setelah mengirim *request* untuk mengubah data. Kode status yang

ditampilkan adalah 200 OK, berarti data permintaan berhasil, dan data yang ingin diubah telah berhasil diubah dengan *value* baru.

#### 4) DELETE

Ketika ingin menghapus data, terlebih dahulu memasukan path berdasarkan id data yang akan dihapus, ketika berhasil dihapus akan muncul respon http status kode 204 *No Content*, berarti permintaan untuk menghapus data telah berhasil dilakukan.

#### C. Response Time

Pengujian *response time* terbagi atas 4 yaitu, pada metode *GET*, *POST*, *PUT*, *DELETE* lalu dilakukan pada buku dan transaksi dapat dilihat pada tabel 4. Untuk setiap *method* dilakukan sebanyak 10 kali kemudian diambil nilai rata-ratanya. Pengujian yang dilakukan pada fitur buku memperoleh waktu rata-rata untuk melakukan permintaan adalah untuk *method GET* adalah 1949.9 ms, *method POST* adalah 1554.4 ms, *method PUT* adalah 887.5 ms dan untuk *method DELETE* adalah 1156.3 ms.

Dan untuk pengujian yang dilakukan pada fitur transaksi dapat dilihat pada tabel 5 memperoleh waktu rata-rata untuk melakukan permintaan adalah untuk *method GET* adalah 19.1 ms, *method POST* adalah 840.6 ms, *method PUT* adalah 345.6 ms dan untuk *method DELETE* adalah 375.3 ms. Pengujian transaksi memiliki jumlah *response time* yang lebih singkat karena jumlah datanya yang sedikit.

#### IV. KESIMPULAN DAN SARAN

Penelitian ini berhasil diimplementasikan *web service* pada sistem informasi perpustakaan perguruan tinggi, yang dirancang dan dibangun menggunakan *framework Django*, Penelitian ini menghasilkan *Restful Web Service* yang diimplementasikan ke dalam sistem dapat bekerja dengan baik dengan *http method GET*, *POST*, *PUT*, dan *DELETE*. *RESTful Web Service* ini dibangun menggunakan *JSON Web Token (JWT)* untuk mengamankan komunikasi data. Prosedur melakukan *request*, *client* harus menyertakan *token* yang didapat dari hasil *registration* atau *login* dengan user yang telah ada untuk di validasi sebelum *request* di proses.

Saran dari peneliti dengan adanya *web service* dapat digunakan untuk mengintegrasikan sistem ini dengan sistem informasi yang berbeda, dan dengan adanya *web service* ini dapat dikembangkan pada platform yang berbeda.

#### V. KUTIPAN

- [1] Musrifah, "Implementasi Teknologi Informasi Menggunakan Human Organization Technology (HOT) FIT Model di Perpustakaan Perguruan Tinggi," *J. Ilmu Perpust. dan Inf.*, vol. 2, no. 2, pp. 222–242, 2017, [Online]. Available: <http://jurnal.uinsu.ac.id/index.php/jipi/article/download/1258/1015>.
- [2] Syahril and Y. Sumarni, "MANAJEMEN PERPUSTAKAAN DALAM MENINGKATKAN MINAT BACA MAHASISWA PERPUSTAKAAN IAIN BENGKULU."
- [3] F. F. Theo, V. Tulenan, and A. Sambul, "Rancang Bangun Aplikasi Digital *Library* Universitas Sam Ratulangi," *J. Tek. Inform.*, vol. 15, no. 4, 2020.
- [4] S. Sibagariang, "Penerapan *Web Service* Pada Perpustakaan Berbasis Android," *JurnalMaharjana Inf.*, vol. 1, no. 2, pp. 8–11, 2016, [Online]. Available: <http://e-journal.sari-mutiara.ac.id/index.php/7/article/view/172>.

- [5] 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, pp. 51–58, 2019, doi: 10.33480/techno.v16i1.164.
- [6] D. Saputra, "Analisis Perbandingan Performa *Web Service Rest* Menggunakan *Framework Laravel*, *Django* Dan *Ruby On Rails* Untuk Akses Data Dengan," *J. Bangkit Indones.*, vol. 7, no. 2, p. 17, 2018, doi: 10.52771/bangkitindonesia.v7i2.90.
- [7] P. F. Tanaem, D. Manongga, and A. Iriani, "RESTful *Web Service* Untuk Sistem Pencatatan Transaksi Studi Kasus PT . XYZ," vol. 2, no. April, 2016.
- [8] R. Ridwan and M. Maisura, "Design and Implementation of *Web Services* for the Integration of Information Systems for the Distribution of Covid-19 Cases," *Cybersp. J. Pendidik. Teknol. Inf.*, vol. 4, no. 2, p. 104, 2020, doi: 10.22373/cj.v4i2.7936.
- [9] R. Rizal and A. Rahmatulloh, "Restful *Web Service* Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan," *J. Ilm. Inform.*, vol. 7, no. 01, p. 54, 2019, doi: 10.33884/jif.v7i01.1004.
- [10] A. Jin, Brenda; Sahni, Saurabh; Shevat, *Designing Web APIs: Building APIs That Developers Love*. 2018.



**Febyana Wijaya** Lahir di Jayapura pada tanggal 01 Februari 1999 dari Teddy Wijaya (Ayah) dan Mariana Wuwung (Ibu). Penulis adalah anak ketiga dari tiga bersaudara, yaitu Christiansen Wijaya (Kakak 1) dan Melinda Wijaya (Kakak 2). Sekarang bertempat tinggal di Desa Koha Selatan, Jaga V, Kecamatan Mandolang, Kabupaten Minahasa. Penulis mulai menempuh pendidikan pertama di TK Kartika Abepura, kemudian melanjutkan pendidikan di SD YPPK Gembala Baik Abepura, kemudian melanjutkan pendidikan di SMP YPPK Santo Paulus Abepura, dan kemudian melanjutkan pendidikan di SMA Katolik Santo Ignatius Manado. Pada tahun 2017 penulis melanjutkan pendidikan di salah satu perguruan tinggi di Manado, yaitu Universitas Sam Ratulangi Manado, Fakultas Teknik, Jurusan Elektro, Program Studi Informatika. Selama perkuliahan penulis tergabung dalam organisasi mahasiswa, yaitu Unit Pelayanan Kerohanian Keluarga Mahasiswa Katolik Fakultas Teknik Unsrat (UPK-KMK FT Unsrat), dan tergabung sebagai anggota Himpunan Mahasiswa Elektro (HME).