

IMPLEMENTASI *WEB SERVICE* RESTFUL API UNTUK LAYANAN PERPUSTAKAAN SMAS DAYA UTAMA BEKASI

Briana Muham^{1*}, Siswanto², Hari Soetanto³, Mohammad Syafrullah⁴

^{1,2,3,4} Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, DKI Jakarta, Indonesia

Email: ^{1*}1811502044@student.budiluhur.ac.id, ²siswanto@budiluhur.ac.id, ³hari.soetanto@budiluhur.ac.id, ⁴mohammad.syafrullah@budiluhur.ac.id

(* : corresponding author)

Abstrak-Saat ini hampir semua bidang baik dunia pendidikan, pemerintahan serta perekonomian telah memanfaatkan perkembangan teknologi, perpustakaan juga merupakan salah satu bidang yang penerapan teknologi infomasinya berkembang pesat. Masalah sering terjadi dengan sistem saat ini di SMAS Daya Utama, seperti kesalahan pencatatan awal peminjaman buku, hingga pengembalian buku. Tujuan dari penelitian ini adalah membuat web service *RESTful API* untuk layanan perpustakaan SMAS Daya Utama Bekasi, sehingga bisa menjadi sarana agar mudah diakses dengan bentuk data JSON oleh aplikasi berbeda platform seperti melalui web ataupun aplikasi android. Dengan memanfaatkan web service, pencarian terhadap buku, pengecekan stok buku, pembacaan ebook, serta pengelolaan ketika siswa meminjam ataupun mengembalikan dapat dengan mudah dipublikasi melalui *client*, serta diakses oleh siapapun yang merupakan anggota serta admin perpustakaan. Aplikasi menyediakan informasi mengenai buku yang ada di perpustakaan, peminjaman buku, membaca eBook, sampai melakukan record kepada siapa saja anggota yang berkunjung ke perpustakaan. Berdasarkan pengujian *UAT* didapatkan hasil dari kepuasan user untuk setiap pertanyaan dalam bentuk presentase dimana Q1 mendapat 68,5%, Q2 mendapat 66,75%, Q3 mendapat 76,75%, Q4 mendapat 80,5%, Q5 mendapat 67,5%, Q6 mendapat 73,25%, Q7 mendapat 81,5%, Q8 mendapat 71,25%, Q9 mendapat 78,75%. Kemudian dibuat kesimpulan bahwa aplikasi mempunyai tampilan yang cukup menarik, menu menu yang dapat dengan mudah dikenali, aplikasi mudah dan nyaman digunakan, mudah dioperasikan, kemampuan dan fungsi sesuai harapan. Namun aplikasi masih belum terdapat fitur notifikasi, serta proses loading gambar yang diambil dari *API* masih bisa ditingkatkan lagi.

Kata Kunci: *RESTful API*, Android, *Web Service*, Layanan Perpustakaan

IMPLEMENTATION OF RESTFUL API WEB SERVICE FOR LIBRARY SERVICES SMAS DAYA UTAMA BEKASI

Abstract-At the moment almost all sectors, either in the education sector, government and the economy have taken advantage of technological developments. The library is also one of the sectors where the application of information technology is growing rapidly. Problems that often occur with the current system at SMAS Daya Utama, such as errors when recording borrowing of books, or returning books. The purpose of this research is to create a *RESTful API* web service for the SMAS Daya Utama Bekasi library service, so that it can be a means for easy access in the form of JSON data by applications on different platforms such as via the web or Android applications. By utilizing web services, searching for books, checking book stock, reading ebooks, and managing when students borrow or return them can be easily published via the client, and accessed by anyone who is a member and administrator of the library. The application provides information about books in the library, borrowing books, reading eBooks, and making records to all members who visit the library. Based on the *UAT* test, the results obtained from user satisfaction for each question in the form of a percentage where Q1 gets 68.5%, Q2 gets 66.75%, Q3 gets 76.75%, Q4 gets 80.5%, Q5 gets 67.5%, Q6 gets 73.25%, Q7 gets 81.5%, Q8 gets 71.25%, Q9 gets 78.75%. Then it was concluded that the application has a fairly attractive appearance, menus that can be easily recognized, the application is easy and comfortable to use, easy to operate, the capabilities and functions are as expected. However, the application still does not have a notification feature, and the process of loading images taken from the *API* can still be improved.

Keywords: *RESTful API*, Android, *Web Service*, Library Services

1. PENDAHULUAN

Komputer adalah alat bantu manusia untuk mengolah data yang menghasilkan informasi [1]. Web API ini memakai arsitektur jaringan REST[2]. Teknologi *Web Service APIs*, terdiri dari *REST* atau *RESTful*. *REST* adalah *system request* yang dapat mengakses dan memanipulasi teks yang ditampilkan dari sebuah *Web Service*. *Web service* dipakai untuk pertukaran data antar aplikasi atau sistem. Kemudian, aplikasi ini dapat ditulis dengan bahasa pemrograman yang berbeda serta dapat berjalan di sistem operasi yang berbeda [3]. Arsitektur *web service* yang digunakan *Simple Object Access Protocol (SOAP)* dan *Representational State Transfer (REST)* [4].

Perpustakaan adalah tempat menyimpan buku-buku dan diorganisasikan sebagai media belajar siswa [5]. Masalah yang ada di Perpustakaan di SMAS Daya Utama, seperti kesalahan pencatatan awal peminjaman buku, pengembalian buku, pencarian buku hingga menghabiskan banyak waktu. Diharapkan dengan adanya *web service*

ini, dapat membantu sirkulasi perpustakaan, serta dapat membantu petugas menjadi lebih praktis dan mudah dalam melakukan pencatatan kegiatan yang berhubungan dengan peminjaman buku di perpustakaan. Tujuan dari penelitian ini adalah membuat *web service RESTful API* untuk layanan perpustakaan SMAS Daya Utama Bekasi, sehingga bisa menjadi sarana agar mudah diakses dengan bentuk data JSON oleh aplikasi berbeda *platform* seperti melalui web ataupun aplikasi *android*.

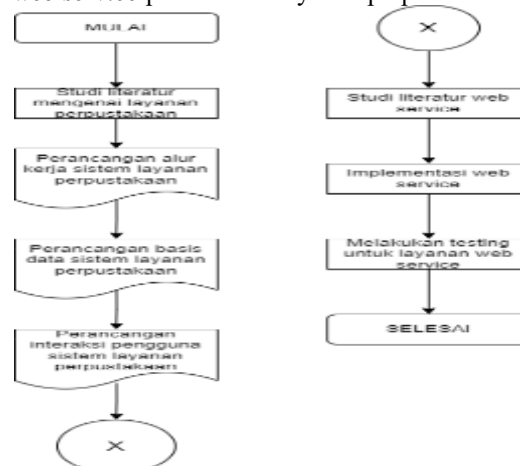
JWT adalah format representasi yang sederhana yang kemudian ditujukan untuk ruang yang bisa digunakan seperti *HTTP Header Authorization* dan parameter *URI request* [6]. Cara kerja JWT seperti *password*, *token* dipakai untuk mengakses halaman web, pengguna akan mengirim balik *token* tersebut sebagai bukti bahwa pengguna telah susah login [7]. Penamaan dan struktur *endpoint* akan menghasilkan *API* yang baik dan mudah dipahami developer [7]. Dalam mengakses data, *RESTful API* menggunakan perintah *HTTP request* untuk mendapatkan *resource* dan format *JSON* sering digunakan dalam merepresentasikan *resource* di *REST* [8]. *UAT* digunakan untuk pengujian final dari perangkat lunak [9]. Gap analisis yang terjadi diaplikasi yang dibuat ini dengan penelitian sebelumnya adalah dilengkapi dengan menggunakan *CR Code create* dan *scanning* buat buku-buku fisik dan buku *digital* serta kartu anggota perpustakaan di *smartphone mobile & web service*, sehingga kesalahan dalam sirkulasi layanan perpustakaan dapat diminimalizer.

Penelitian mengenai implementasi REST API telah dilakukan oleh [4] Dari penelitian tersebut berhasil menyelesaikan permasalahan menggunakan arsitektur REST, yang kemudian proses input serta verifikasi data hanya dilakukan satu kali, hal tersebut juga mengatasi terjadinya duplikasi data serta mengurangi perkerjaan input data. Pada penelitian yang dilakukan oleh [10] Dari penelitian tersebut menunjukkan bahwa penggunaan JSON Web Token Authentication mampu membentuk sistem yang sangat skalabel, aman, serta mampu berinteraksi multi-platform dan dapat diandalkan. Pada penelitian yang dilakukan oleh [2] Dari penelitian tersebut dihasilkan sebuah website yang bersifat terpusat dengan cara memanfaatkan *web service* metode REST API dalam proses pengambilan datanya dan bertujuan untuk menggabungkan data-data tersebut meskipun berasal dari database yang berbeda-beda.

2. METODE PENELITIAN

2.1 Alur Penelitian

Penelitian ini diawali dengan melakukan studi literatur tentang layanan perpustakaan berbasis android. Selanjutnya perancangan alur kerja sistem layanan perpustakaan. Dilanjutkan dengan perancangan basis data. Lalu, dilakukan perancangan interaksi pengguna sistem layanan perpustakaan dengan *use case diagram*. Gambar 1 menunjukkan alur pengembangan *web service* pada sistem layanan perpustakaan SMAS Daya Utama.



Gambar 1. Alur Penelitian

Data penelitian yang digunakan dalam penelitian ini yaitu melalui wawancara serta observasi. Data penelitian primer yang digunakan dalam penelitian ini yaitu data yang diperoleh melalui wawancara secara langsung di perpustakaan SMAS Daya Utama. Data penelitian primer merupakan data yang diambil oleh peneliti kepada sumbernya tanpa ada perantara dengan cara meneliti sumber asli secara langsung melalui responden. Sumber data penelitian primer dalam penelitian ini adalah penanggung jawab perpustakaan SMAS Daya Utama. Hasil data yang didapatkan dari wawancara digunakan sebagai data buku pada aplikasi layanan perpustakaan SMAS Daya Utama.

2.2 Alur Kerja Sistem

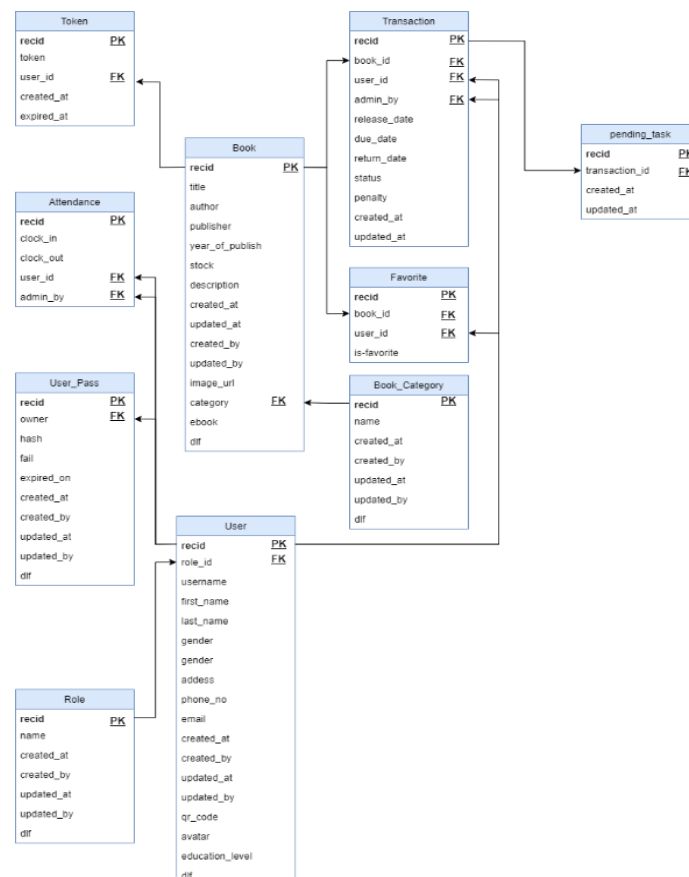
Rancangan layanan *web service* sebagai penyedia layanan untuk aplikasi lain, serta spesifikasi basis data yang akan dipakai untuk mengakses data yang digunakan. Gambar 2 menunjukkan alur kerja dari sistem yang akan dibuat. Setiap kali *client* berkomunikasi dengan *server*, lalu *server* meminta *token* dan akan memvalidasi token tersebut. Hal tersebut memungkinkan *client* untuk bisa mendapat *token* dengan memberikan *username* dan *password* mereka. Sebuah *token* pada umumnya diterbitkan oleh penyedia layanan dan dikonsumsi oleh pihak yang mengandalkan konten untuk bisa mengidentifikasi subjek *token* dengan tujuan yang berhubungan dengan keamanan.



Gambar 2. Alur Kerja Sistem

2.3 Desain Basis Data

Basis data yang dibuat terdiri dari sepuluh tabel, yaitu tabel *book*, *user*, *token*, *attendance*, *user pass*, *role*, *transaction*, *favorite*, *book category*, dan *pending_task* seperti ditunjukkan pada Gambar 3.



Gambar 3. Desain Basis Data

Tabel *attendance* berfungsi untuk menyimpan data kunjungan yang dilakukan oleh seluruh *member*, tabel *book* berfungsi untuk menyimpan seluruh data buku, tabel *book category* berfungsi untuk menyimpan data kategori buku yang nantinya akan menjadi rujukan kategori dari setiap buku, tabel *favorite* berfungsi untuk menyimpan data buku yang difavoritkan oleh setiap *user*, tabel *pending task* akan diisi oleh data peminjaman yang masih tertunda atau belum disetujui oleh admin dan transaksi akan berstatus *pending*, tabel *role* berfungsi untuk

menyimpan jenis *role* yang akan digunakan oleh user, tabel *token* berfungsi untuk menyimpan data token seluruh user yang nantinya akan digunakan untuk mengakses *API*, tabel *transaction* berfungsi untuk menyimpan data peminjaman baik yang berstatus *pending* hingga yang berstatus *returned*, tabel *user* berfungsi untuk menyimpan data *user*, dan tabel *user password* berfungsi untuk menyimpan data *password* dari setiap *user*.

2.4 Use Case Diagram

Gambar 4 adalah *use case diagram* dari *REST API* sistem layanan perpustakaan SMAS Daya Utama. Pengguna dari *API* sistem layanan perpustakaan dibagi menjadi dua, yaitu admin sebagai *user* yang mampu melakukan manajemen pada buku, data user maupun peminjaman. Dan *member* berperan sebagai *user* yang mampu melakukan aktivitas yang berhubungan transaksi peminjaman buku.



Gambar 4. Use Case Diagram REST API sistem layanan perpustakaan SMAS Daya Utama

Berdasarkan *use case diagram* tersebut, terdapat:

- Dua jenis aktor, yaitu admin yang mempunyai hak akses terhadap CRUD buku, CRUD user, serta menyetujui atau menolak peminjaman. Dan *member* yang dapat mengajukan peminjaman buku serta membaca *eBook*.
- Total ada 9 buah *use case* yang bisa dilakukan oleh dan aktor tersebut.

2.5 API Endpoint

API endpoint merepresentasikan sebuah alamat tujuan dari setiap fungsi pada sistem layanan perpustakaan yang dikembangkan, di mana *endpoint* bisa diakses oleh *client* untuk memperoleh, menambah, ataupun mengubah *resource* yang ada.

Tabel 1. API Endpoint REST API Sistem Layanan Perpustakaan SMAS Daya Utama

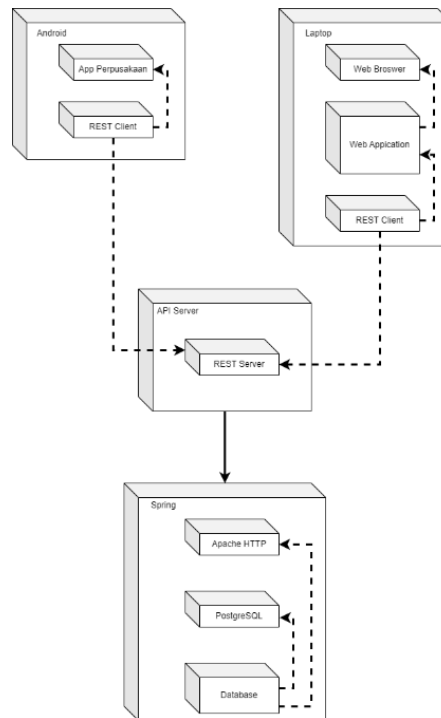
HTTP Method	Path	Fungsi
POST	/transaction/create	Membuat pengajuan peminjaman buku
POST	/transaction/return	Mengembalikan buku
POST	/pendingtask/approve	Menyetujui peminjaman
POST	/pendingtask/reject	Menolak peminjaman
POST	/book/create	Membuat buku baru
POST	/book/list	Mendapatkan daftar buku
DELETE	/book/delete	Menghapus buku
POST	/user/create	Membuat user baru
POST	/user/list	Mendapatkan daftar user
DELETE	/user/delete	Menghapus user
GET	/media/book/show/{imageName}	Mendapatkan gambar (poster) buku

Tabel 1 adalah tabel yang berisi *API endpoint* yang dikembangkan pada *REST API* sistem layanan perpustakaan SMAS Daya Utama dikembangkan berjumlah 11 *API endpoint* berupa URL *API RESTful* diakses menggunakan protokol *HTTP*.

3. HASIL DAN PEMBAHASAN

3.1 Lingkungan Percobaan

Adapun *deploy diagram* dapat dilihat pada gambar 5.



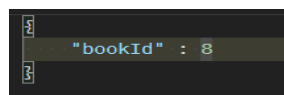
Gambar 5. Deploy Diagram

3.2 REST API Method

Penelitian ini, *web service* yang digunakan adalah Spring. *Controller* mempunyai peran untuk menentukan informasi yang dikirim ketika *request* dibuat. Dengan adanya arsitektur *REST*, maka *method-method* yang digunakan dalam penelitian ini antara lain:

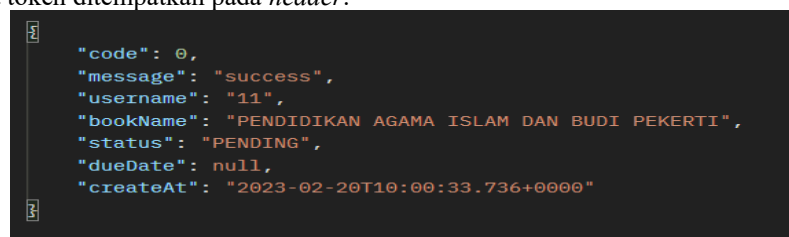
- GET, digunakan untuk mengakses ataupun membaca data yang terdapat pada *resource*.
- POST, digunakan untuk memperbaharui ataupun menambah data dan jika diperlukan mampu membaca data pada *resource*.
- DELETE, digunakan untuk menghapus *resource*.

3.3 Pengujian Terhadap Program



Gambar 6. Request Body Yang Digunakan Untuk API Testing dari API Endpoint berfungsi untuk membuat transaksi peminjaman

Gambar 6 adalah *request body* untuk mengajukan peminjaman buku. Informasi yang dicantumkan adalah id dari buku. Sementara token ditempatkan pada *header*.



Gambar 7. Response dari API testing dari API endpoint yang berfungsi untuk membuat transaksi peminjaman dengan kondisi pengajuan berhasil dilakukan

Gambar 7 adalah response dari berhasilnya *request* yang dilakukan saat pengajuan peminjaman buku. Informasi yang terlihat adalah *response code* 0 (OK), lalu data yang ada di response body. Nilai *response code* selain dari 0 adalah nilai *error*.

```

{
  "code": -1,
  "message": "Check Your Waiting List Or Transaction List"
}

```

Gambar 8. Response dari API testing dari API endpoint yang berfungsi untuk membuat transaksi peminjaman dengan kondisi pengajuan gagal dilakukan

Gambar 8 adalah response dari gagalnya dilakukan pengajuan peminjaman buku karena buku sedang dipinjam dan belum dikembalikan atau buku sudah dalam tahap pengajuan dan menunggu respon *admin*. Informasi yang terlihat adalah response code -2 dimana pesan gagalnya adalah "Check Your Waiting List Or Transaction List".

3.4 Implementasi REST API

3.4.1 Tampilan Layar Android

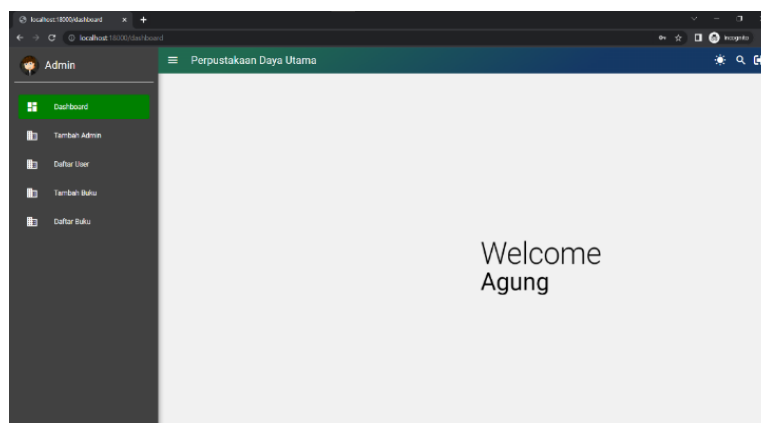
Halaman *dashboard* merupakan halaman di mana ketika *user* dengan *role* admin berhasil login, maka akan muncul halaman *dashboard* admin seperti pada gambar 9.



Gambar 9. Halaman Dashboard Dari Sisi Admin

3.4.2 Tampilan Layar Web

Halaman *dashboard* merupakan halaman di mana ketika admin berhasil *login*, maka akan muncul halaman *dashboard website* yang dapat di lihat pada gambar 10.



Gambar 10. Halaman Dashboard Dari Sisi Admin

3.5 Pengujian UAT (User Acceptance Test) Aplikasi Android

Pengujian UAT telah dilakukan dengan memberikan 9 pertanyaan kepada 27 responden, di mana jawaban dari pertanyaan tersebut terdiri dari pilihan jawaban yang dapat dipilih seperti pada tabel 2. Kemudian akan ada penjelasan mengenai bobot nilai pada Tabel 3.

Tabel 2. Pilihan Jawaban

A	Sangat: baik/mudah/bagus/seusai/jelas
B	baik/mudah/bagus/seusai/jelas
C	Cukup: baik/mudah/bagus/seusai/jelas
D	Sangat: sulit/jelek/tidak sesuai/tidak jelas

Tabel 3. Bobot Nilai *UAT*

Jawaban	Bobot
A. Sangat: baik/mudah/bagus/seusai/jelas	4
B. baik/mudah/bagus/seusai/jelas	3
C. Cukup: baik/mudah/bagus/seusai/jelas	2
D. Sangat: sulit/jelek/tidak sesuai/tidak jelas	1

Tabel 4. Pertanyaan dan Data Jawaban Kuesioner

No	Pertanyaan	Hasil			
		A	B	C	D
1	Apakah fungsi <i>login</i> pada aplikasi berjalan dengan baik?	4	12	11	0
2	Apakah tampilan aplikasi ini menarik?	3	12	12	0
3	Apakah aplikasi ini bermanfaat?	7	15	5	0
4	Apakah semua menu berfungsi dengan baik?	10	13	4	0
5	Apakah informasi yang anda cari mudah ditemukan?	3	16	5	3
6	Secara keseluruhan apakah aplikasi ini nyaman digunakan?	4	17	6	0
7	Apakah tampilan menu dalam aplikasi mudah untuk dikenali?	12	10	5	0
8	Apakah aplikasi ini mempermudah anda dalam peminjaman buku di perpustakaan?	5	15	5	2
9	Apakah tampilan layar aplikasi tampil baik sesuai dengan ukuran <i>smartphone</i> anda?	11	11	4	0

Dari tabel 4 lalu dibuat tabel 5 yang merupakan jawaban pertanyaan kuesioner dikalikan bobot.

Tabel 5. Jawaban Pertanyaan Kuisisioner x Bobot Nilai *UAT*

No	Pertanyaan	Hasil				Total
		A	B	C	D	
1	Q1	16	36	22	0	74
2	Q2	12	36	24	0	72
3	Q3	28	45	10	0	83
4	Q4	40	39	8	0	87
5	Q5	12	48	10	3	73
6	Q6	16	51	12	0	79
7	Q7	48	30	10	0	88
8	Q8	20	45	10	2	77
9	Q9	44	33	8	0	85

Hasil dari kuesioner dibuat kesimpulan dalam bentuk presentase. Di mana 1 mengacu pada tabel 4 Q1 adalah pertanyaan, Q2 pertanyaan 2, dan seterusnya. Dan A, B, C, D adalah tingkat kepuasan user.

a. Analisa pertanyaan Q1:

Dari tabel 5 dapat dilihat bahwa jumlah nilai dari 27 responden untuk pertanyaan pertama adalah 74. Nilai rata-ratanya adalah $74/27 = 2,74$. Presentase nilainya adalah $2,74/4 \times 100\% = 68,5\%$.

b. Analisa pertanyaan Q2:

Dari tabel 5 dapat dilihat bahwa jumlah nilai dari 27 responden untuk pertanyaan kedua adalah 72. Nilai rata-ratanya adalah $72/27 = 2,67$. Presentase nilainya adalah $2,67/4 \times 100\% = 66,75\%$.

c. Analisa pertanyaan Q3:

Dari tabel 5 dapat dilihat bahwa jumlah nilai dari 27 responden untuk pertanyaan ketiga adalah 83. Nilai rata-ratanya adalah $83/27 = 3,07$. Presentase nilainya adalah $3,07/4 \times 100\% = 76,75\%$.

- d. Analisa pertanyaan Q4:
Dari tabel 5 dapat dilihat bahwa jumlah nilai dari 27 responden untuk pertanyaan keempat adalah 87. Nilai rata-ratanya adalah $84/27 = 3,22$. Presentase nilainya adalah $2,22/4 \times 100 \% = 80,5 \%$.
- e. Analisa pertanyaan Q5:
Dari tabel 5 dapat dilihat bahwa jumlah nilai dari 27 responden untuk pertanyaan kelima adalah 73. Nilai rata-ratanya adalah $73/27 = 2,7$. Presentase nilainya adalah $2,7/4 \times 100 \% = 67,5 \%$.
- f. Analisa pertanyaan Q6:
Dari tabel 5 dapat dilihat bahwa jumlah nilai dari 27 responden untuk pertanyaan keenam adalah 79. Nilai rata-ratanya adalah $79/27 = 2,93$. Presentase nilainya adalah $2,93/4 \times 100 \% = 73,25 \%$.
- g. Analisa pertanyaan Q7:
Dari tabel 5 dapat dilihat bahwa jumlah nilai dari 27 responden untuk pertanyaan ketujuh adalah 88. Nilai rata-ratanya adalah $88/27 = 3,26$. Presentase nilainya adalah $3,26/4 \times 100 \% = 81,5 \%$.
- h. Analisa pertanyaan Q8:
Dari tabel 5 dapat dilihat bahwa jumlah nilai dari 27 responden untuk pertanyaan kedelapan adalah 77. Nilai rata-ratanya adalah $77/27 = 2,85$. Presentase nilainya adalah $2,85/4 \times 100 \% = 71,25 \%$.
- i. Analisa pertanyaan Q9:
Dari tabel 5 dapat dilihat bahwa jumlah nilai dari 27 responden untuk pertanyaan kesembilan adalah 85. Nilai rata-ratanya adalah $85/27 = 3,15$. Presentase nilainya adalah $3,15/4 \times 100 \% = 78,75 \%$.

4. KESIMPULAN

Kesimpulan dari penelitian ini adalah dengan sistem yang ada, proses peminjaman serta pengembalian buku dapat dilakukan secara lebih cepat, akurat, tepat, serta efisien, dengan Metode REST API anggota perpustakaan bisa mendapatkan serta mengubah data pada *database* yang sama, tanpa perlu memikirkan dari *platform* mana sistem diakses, metode REST API dapat diterapkan pada platform android dan web dengan baik, aplikasi web terbatas dan hanya dapat diakses oleh admin, sedangkan aplikasi android bisa diakses oleh member maupun admin. Pada aplikasi *android*, admin juga mampu melakukan operasi CRUD (*Create, Read, Update, Delete*), dengan sistem yang ada, penyimpanan data dalam jumlah yang besar sudah bisa teratasi. Dan bisa meminimalisir terjadinya kehilangan atau kerusakan pada data, berdasarkan pengujian UAT didapatkan hasil dari kepuasan pengguna untuk setiap pertanyaan dalam bentuk presentase dimana Q1 mendapat 68,5%, Q2 mendapat 66,75%, Q3 mendapat 76,75%, Q4 mendapat 80,5%, Q5 mendapat 67,5%, Q6 mendapat 73,25%, Q7 mendapat 81,5%, Q8 mendapat 71,25%, Q9 mendapat 78,75%. Kemudian dibuat kesimpulan bahwa aplikasi mempunyai tampilan yang cukup menarik, menu menu yang dapat dengan mudah dikenali, aplikasi mudah dan nyaman digunakan, mudah dioperasikan, kemampuan dan fungsi sesuai harapan.

DAFTAR PUSTAKA

- [1] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 18, no. 2, pp. 284–293, May 2019, doi: 10.30812/matrik.v18i2.407.
- [2] H. Wakkang and B. Irianto, "Implementasi Web Service Dengan Metode REST API Untuk Integrasi Data Covid 19 Di Sulawesi Selatan," vol. 2, no. 1, pp. 12–22, 2022, doi: 10.31850/jsilog.v2i1.
- [3] F. Surahman, S. H. al Ikhsan, F. Satrya, and F. Kusumah, "Rancang Bangun Web Service Untuk Transaksi Data Pada Aplikasi Sahabat Jasa Dengan Metode REST," *SEMANTI*, vol. 1, pp. 256–264, 2018.
- [4] R. Rizal and A. Rahmatulloh, "RESTful Web Service untuk Integrasi Sistem Akademik dan Perpustakaan Universitas Perjuangan," *Jurnal Ilmiah Informatika*, vol. 7, no. 1, pp. 55–59, 2019.
- [5] A. Eskha, "Peran Perpustakaan Sebagai Sumber Belajar," *Jurnal Imam Bonjol: Kajian Ilmu Informasi dan Perpustakaan*, vol. 2, no. 1, pp. 12–18, 2018.
- [6] A. Lendrasidi, "Implementasi Web Service RESTful Dengan Autentikasi JSON Web Token Berbasis Web Dan Android Pada PT Kahaptex," Universitas Budi Luhur, Jakarta, 2018.
- [7] H. Darmadi, "Merancang Web API yang Mudah Dieksplorasi," 2018. <https://binus.ac.id/knowledge/2018/05/merancang-web-api-yang-mudah-dieksplorasi/> (accessed Jul. 24, 2022).
- [8] I. A. Kaniya *et al.*, "Perancangan Dan Implementasi RESTful API Pada Sistem Informasi Manajemen Dosen Universitas Udayana," *Jurnal SPEKTRUM*, vol. 9, no. 3, pp. 15–23, 2022.
- [9] M. A. Chamida, A. Susanto, and A. Latubessy, "Analisa User Acceptance Testing Terhadap Sistem Informasi Pengelolaan Bedah Rumah Di Dinas Perumahan Rakyat Dan Kawasan Permukiman Kabupaten Jepara," *Indonesian Journal of Technology, Informatics and Science (IJTIS)*, vol. 3, no. 1, pp. 36–41, Dec. 2021, doi: 10.24176/ijtis.v3i1.7531.
- [10] R. Gunawan and A. Rahmatulloh, "JSON Web Token(JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service," *Jurnal Edukasi dan Penelitian Informatika*, vol. 5, no. 1, pp. 74–79, 2019.