PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika
Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

Implementasi Restful Api Dalam Upaya Mensinkronisasi Data Pada Sistem Otomasi Perpustakaan Berbasis Web Dengan Menggunakan Metode Uji Coba Rmse Dan White Box

Yessy Fitriani[1]; M. Yoga Distra Sudirman[2]; Dine Tiara Kusuma[3]; Abiyyu Wahib Imantara[4]

[1,2,3,4] Institusi Teknologi PLN
[1] yessy.fitriani@itpln.ac.id

ABSTRACT

To solve the data redundancy problem that occurs in the STTPLN library automation system, namely the Senayan and Cendana systems, currently a web service has been designed to combine the two automation systems in the library so that data redundancy from the two systems will not occur. This web service application can migrate and merge data from the Senayan and Cendana databases to the Akasia database. The application of the REST architecture here is to test the ability of the REST architecture in handling data migration and merging with large amounts of data. There are two types of testing, namely speed testing and validation testing. The speed test is done by calculating the estimated time spent when transferring data with the amount of data ranging from 4000, 6000, 8000, 10000.The result is that using the REST architecture in migrating data can shorten the migration time.While the results of data validation testing using RMSE show a very small number, namely 0. From these results it can be stated that the acacia database from the results of the localhost migration already has sufficient data accuracy to be used as a new system database.

Keywords: *Library, API, RESTful,, Automation, Databases*

ABSTRAK

Untuk mengatasi permasalahan redundancy data yang terjadi pada sistem otomasi perpustakaan STTPLN yaitu sistem Senayan dan Cendana, saat ini telah dirancang webservice untuk menggabungkan dua sistem otomasi pada perpustakaan sehingga tidak akan terjadi data redundacy dari kedua sistem tersebut. Aplikasi web service ini dapat melakukan migrasi dan penggabungan data pada database Senayan dan Cendana ke database Akasia. Penerapan arsitektur REST disini adalah untuk menguji kemampuan arsitektur REST dalam menangani migrasi dan penggabungan data dengan jumlah data yang banyak. Terdapat dua macam pengujian yaitu pengujian kecepatan dan pengujian validasi. Pengujian kecepatan dilakukan dengan menghitung estimasi waktu yang dipakai saat transfer data dengan jumlah data berkisar antara 4000, 6000, 8000, 10000. Hasilnya adalah dapat mempersingkat waktu migrasi. Sedangkan hasil dari pengujian validasi data menggunakan RMSE menunjukan angka yang sangat kecil yaitu 0. Dari hasil tersebut dapat dinyatakan bahwa database akasia dari hasil migrasi localhost sudah memiliki akurasi data yang cukup untuk dapat digunakan sebagai database sistem yang baru.

Kata kunci: *Perpustakaan, API, RESTful, Otomasi, Basis data*

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

1. PENDAHULUAN

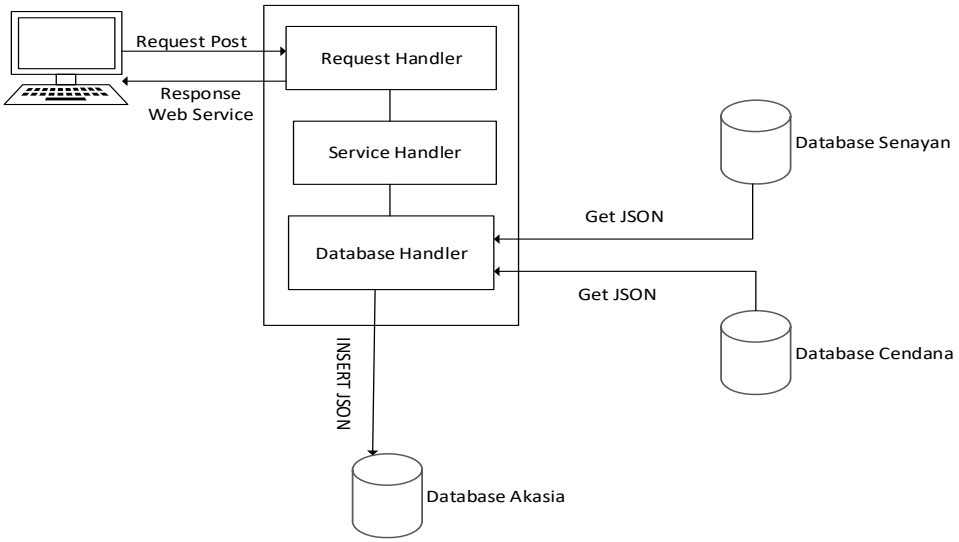
Sistem Otomasi yang terdapat di perpustakaan IT-PLN saat ini masih bersifat manual, data transaksi dari dua sistem otomasi yakni Senayan dan Cendana masih berjalan terpisah, tetapi saat ini telah dibuat webservice yang dapat menampung data dari dua sistem tersebut, sehingga tidak akan terjadi data *redundancy* ataupun data duplikat. Perkembangan sistem otomasi yang bersifat opensource, perangkat lunak sistem manajemen perpustakaan (*library management system*) sumber terbuka yang dilisensikan di bawah GPL v3. Aplikasi web yang dikembangkan oleh tim dari Pusat Informasi dan Humas Departemen Pendidikan Nasional Republik Indonesia ini dibangun dengan menggunakan PHP, basis data MySQL, dan kendali versi Git. Pada tahun 2009. yang dikembangkan oleh programmer indonesia untuk kebutuhan perpustakaan indonesia Saat ini perpustakaan ITPLN memiliki dua sistem otomasi perpustakaan yang berjalan secara bersamaan keduanya merupakan produk dari SLIMS(*Senayan Library Management System*) yang berbeda versi yaitu SLIMS versi 4 (Senayan) dan SLIMS versi 7 (Cendana) [1]. Pembaruan versi dari senayan dan cendana yang dilakukan oleh perpustakaan ITPLN menyebabkan redundancy data yang diakibatkan pada saat pengimplementasian sistem cendana dalam rangka pembaruan sistem, cendana diimplementasikan menggunakan data dari sistem Senayan namun setelah pembaruan tersebut kedua sistem tetap berjalan sampai saat ini tanpa adanya sinkronisasi data antar sistem, sehingga terjadi penggunaan fungsi yang berbeda yaitu sistem Senayan sebagai database buku dan transaksi, sementara Cendana digunakan sebagai database karya ilmiah dan untuk OPAC(*Online Public Access Catalog*). *Redundancy* data yang ada menyebabkan data yang dimiliki oleh perpustakaan ITPLN kurang akurat. Total data pada database yaitu pada database Senayan total 202.290 record data dan pada database cendana total 210.174 record data. Untuk mengatasi redundancy data, diperlukan sebuah sistem untuk melakukan migrasi dan penggabungan *database*. Dikarenakan dua sistem yang berjalan tersebut memiliki jumlah record yang banyak serta memiliki struktur database yang berbeda dan terdapat banyak data yang duplikasi menyebabkan tidak dimungkinkannya untuk melakukan pemasukan data ulang secara manual maupun secara SQL. Oleh karena itu dibutuhkan sebuah sistem untuk melakukan migrasi dan sinkronisasi data untuk mengatasi redundancy data pada database sistem. Dalam hal ini web service dapat digunakan untuk melakukan migrasi dan penggabungan data. *Web service* memiliki beberapa arsitektur yaitu SOAP (*Simple Object Access Protocol*) dan REST (*Representational State Transfer*) keduanya merupakan arsitektur pertukaran data yang dapat berjalan dengan protokol HTTP/HTTPS, Pemilihan REST sebagai dasar arsitektur adalah karena REST lebih sedikit memerlukan resource dan dapat diterapkan dalam bentuk JSON [2]. REST dapat diimplementasikan dalam bentuk web service yang disebut dengan RESTful *web service.*.

2. METODE/PERANCANGAN PENELITIAN

2.1. Perancangan Arsitektur REST

Pada bagian perancangan arsitektur REST menggambarkan bagaimana alur kerja arsitektur REST yang akan diimplementasikan pada sistem migrasi dan penggabungan data [3,4,5].

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228



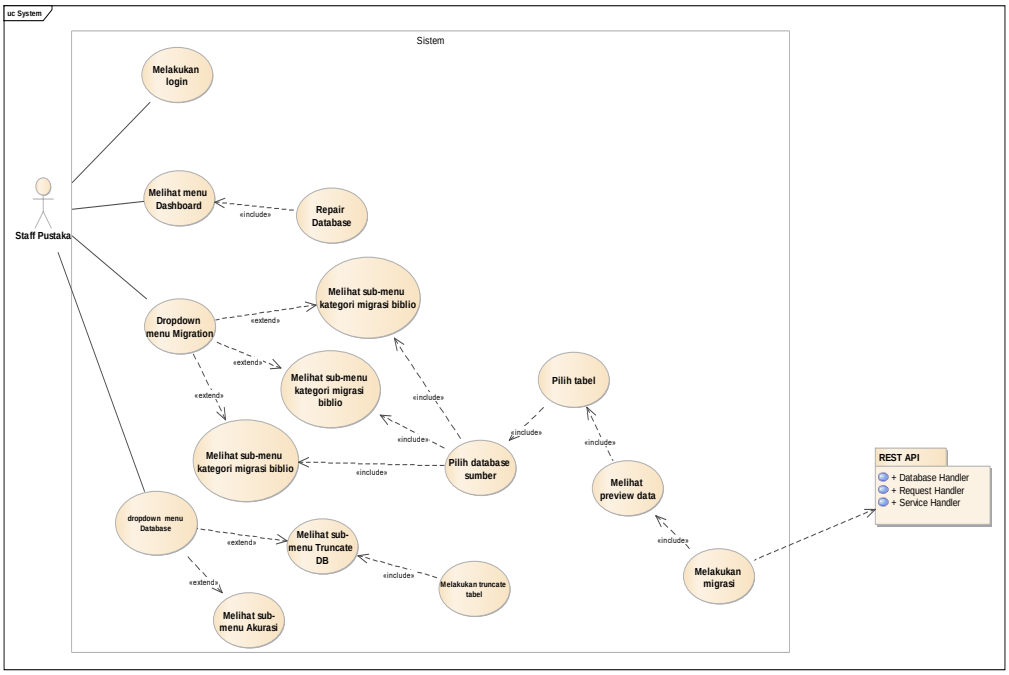
Gambar 1. Arsitektur REST Migrasi dan Penggabungan Data

2.2. Perancangan Alur Sistem

Pada bagian perancangan alur sistem penulis melakukan perancangan alur sistem berdasarkan analisis kebutuhan yang telah dilakukan, perancangan ini dilakukan menggunakan *UML* yang menggambarkan proses-proses yang terjadi pada sistem. berikut merupakan perancangan alur sistem[5]:

Use case diagram

Pada bagian ini penulis mencoba menggambarkan perilaku sistem dengan menggunakan *use case diagram*. Berikut adalah *use case diagram* pada sistem:



Gambar 2. Use case diagram System

2.3. Teknik Analisis

Teknik analisis yang dilakukan penulis pada penelitia ini adalah untuk melihat kemampuan arsitektur REST dalam menangani transfer data dalam jumlah yang banyak[6]. Teknik analisis akan

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

dilakukan dengan melakukan berbagai percobaan diantaranya adalah pengujian kecepatan, dan pengujian validitas data yang ditransfer, kemudian melakukan pengujian *whitebox* terhadap desain *activity* sistem[7]. Pengujian kecepatan dan validitas data akan dilakukan dalam dua macam skenario yaitu skenario *localhoST.*, skenario LAN, dan skenario LAN server produksi[8].

1. **Skenario localhost**
   Skenario localhost akan dilakukan dengan unit 01 sebagai server untuk sistem *web service* dan sebagai *database* server sebagai ketiga *database* yaitu *database* senayan, cendana dan akasia.
2. **Skenario LAN simulasi**
   Skenario LAN akan dilakukan dengan unit 01 sebagai server untuk sistem *web service* serta *database* server untuk *database* akasia, sementara unit 02 hanya digunakan sebagai *database* server untuk *database* senayan dan cendana. Kedua unit tersebut terhubung dalam jaringan LAN. Masing -masing unit memiliki alamat ip statis yang terhubung menggunakan media transmisi kabel UTP yang terhubung ke switch dengan kecepatan transfer data 100MB/detik.
3. **Skenario LAN server produksi**
   Pada skenario LAN server produksi adalah dengan menginstall sistem pada server produksi untuk sistem, yang kemudian sistem akan mencoba berkomunikasi dengan server cendana dan server senayan via LAN dalam jaringan dalam gedung ITPLN. Skenario pengujian yang dilakukan akan sesuai dengan sistem usulan.

2.4. Pengujian Kecepatan

Pada pengujian kecepatan penulis akan melakukan percobaan dengan dengan dua cara yaitu dengan cara localhost dan dengan cara LAN. Pengujian dilakukan dengan cara mengakses API menggunakan postman untuk mengukur kecepatan proses API tersebut dalam merespon sebuah *request.* Pengujian dilakukan dengan mengirimkan jumlah data yang bervariasi yaitu 4000,6000,8000, dan 10000 record yang diambil dari *database* cendana pada tabel biblio. Pengujian akan dilakukan pada ketiga skenario yang telah ditentukan yaitu skenario *localhoST.*, LAN simulasi, dan LAN *server* produksi.

2.5. Pengujian Validasi Data

Pengujian akan dilakukan dengan mengukur tingkat error pada jumlah data yang diinputkan dan jumlah data yang berhasil masuk pada *database*. Tingkat error akan diukur menggunakan metode RMSE(*Root Mean Square Error*)[9].RMSE merupakan suatu ukuran kesalahan yang didasarkan pada selisih antara dua buah nilai yang bersesuaian, yang didefinisikan sebagai berikut:

$$RMSE = \sqrt{\frac{E_{i=1}^{n}(Yi - \hat{Y}i)^2}{n}} \quad (1)$$

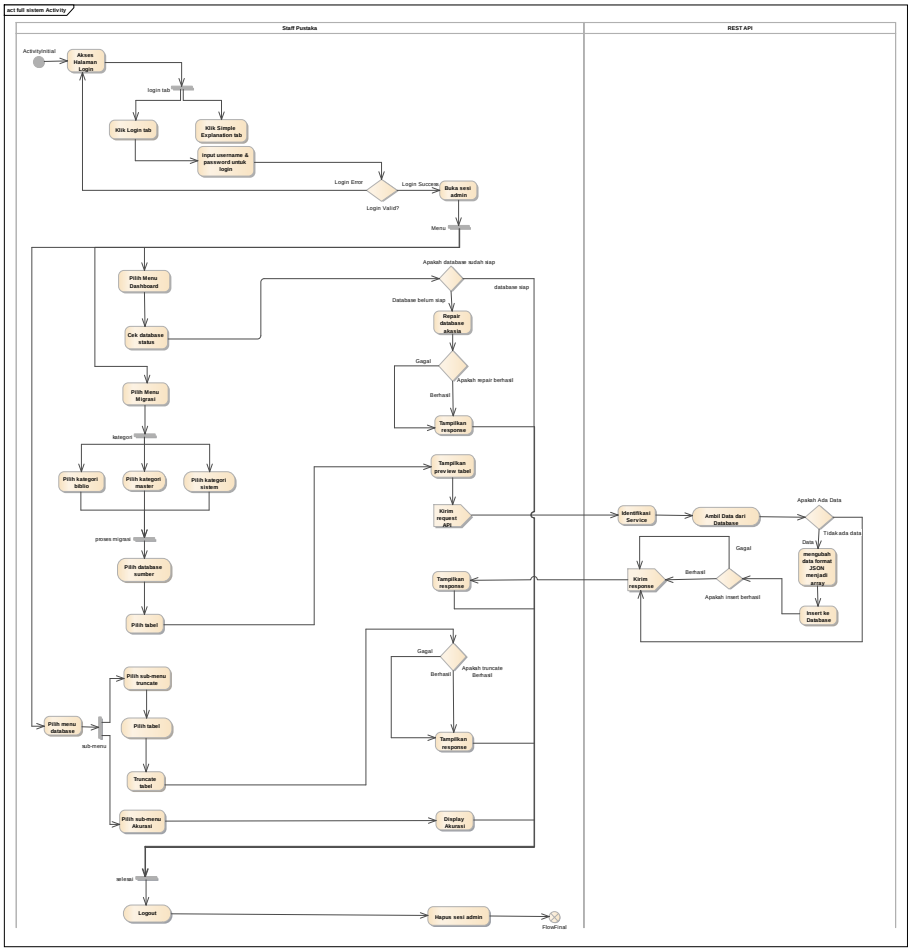
Keterangan: Yi = data awal, $\hat{Y}i$ = data akhir, n= jumlah data

2.6. Pengujian White Box

Pengujian kompleksitas sistem mungkin dapat dikaitkan dengan fakta bahwa itu melibatkan pengujian sistem yang sepenuhnya terintegrasi yang mungkin besar dan kompleks. Tidak mengherankan, pengujian sistem pada sistem tipikal yang seringkali melebihi upaya desain uji manual. Oleh karena itu, dengan ukuran sistem yang terus meningkat, masalah desain otomatis kasus uji sistem dianggap sangat penting . Rangkaian pengujian yang dihasilkan dengan benar mungkin tidak hanya menemukan kesalahan dalam sistem perangkat lunak, tetapi juga membantu mengurangi biaya tinggi yang terkait dengan pengujian perangkat lunak [10,11]. *Unified Modeling Language* (UML) adalah standar *de-facto* untuk analisis pemodelan dan perancangan desain[12].

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

Pengujian kompleksitas sistem akan menguji diagram alir sistem keseluruhan untuk mengetahui risiko error yang mungkin dihasilkan dari desain sistem secara keseluruhan, berikut adalah diagram alir sistem keseluruhan:



Gambar 3. Diagram alir sistem keseluruhan

Pengujian kompleksitas sistem akan menguji diagram alir secara keseluruhan dengan mengubah menjadi *flowgraph* yang kemudian diuji menggunakan metode *Cyclomatic Complexity* setelah itu membandingkan hasil dengan tabel hubungan *Cyclomatic Complexity and Risk* menurut McCabe, yaitu:

Tabel 1. Tabel hubungan Cyclomatic Complexity dan Risiko

Cyclomatic Complexity	*Risk Evaluation*	*Probability of Bad fix*
1-10	*Low Risk, testable code*	5%
11-20	*Moderate Risk*	10%
21-50	*High Risk*	30%
>50	*Very High Risk, untestable code*	40%

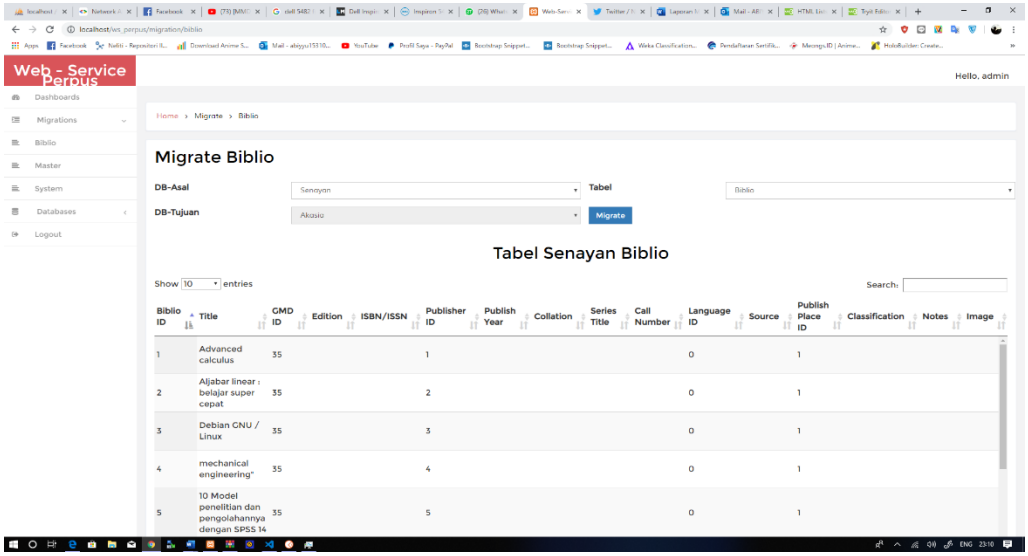
3. HASIL DAN PEMBAHASAN

3.1. Hasil Perancangan Antar Muka

Sistem migrasi dan penggabungan data dibangun Menggunakan bahasa pemograman *Hypertext Preprocessor (PHP)* dalam *framework CodeIgniter*. Berikut adalah hasil perancangan antarmuka sistem:

PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika
Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

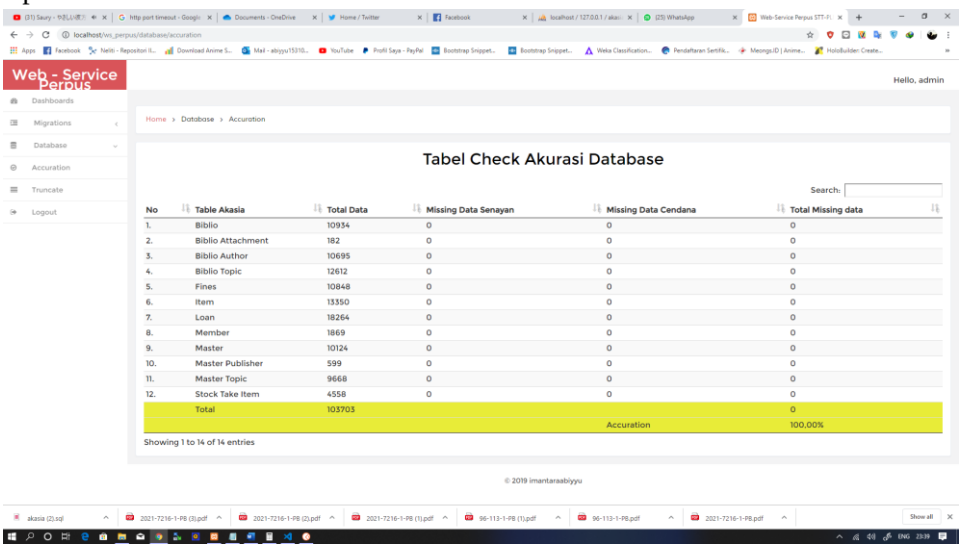
1. Tampilan Menu Migrasi dan Penggabungan Data



Gambar 4. Halaman Migrasi dan Penggabungan Data

Pada halaman tersebut user dapat melakukan migrasi dan penggabungan data dari *database* Senayan dan Cendana ke dalam *database* Akasia. Migrasi dan penggabungan data dilakukan per tabel dengan terdapat tampilan tampilan table yang berisi data yang belum tersedia pada *database* akasia.

2. Tampilan Sub-menu Akurasi



Gambar 5. Halaman Sub-menu Akurasi

Pada halaman ini user dapat melihat detail dari akurasi *database* akasia apakah *database* tersebut sudah memiliki ketersediaan data dari *database* senayan dan cendana atau belum.

3.2. Hasil Analisis Basis Data

Seperti yang telah dibahas sebelumnya, bahwa terdapat *redundancy data* yang ada pada *database* senayan dan cendana. Penulis melakukan analisis struktur yang bisa dilihat pada lampiran

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

tiga dan jumlah data pada kedua *database* tersebut. Berikut adalah jumlah data asli (data tanpa duplikasi) yang ada pada *database* senayan dan cendana:

Tabel 2. Tabel jumlah data nonduplikat

No.	Nama Tabel	Senayan	Senayan nonduplikat	Cendana	Cendana nonduplikat	Total data nonduplikat
1.	biblio	4537	4537	10759	6397	10934
2.	biblio_attachment	0	0	182	182	182
3.	biblio_author	4503	4503	10724	6192	10695
4.	biblio_topic	6330	6330	13102	6282	12612
5.	fines	10848	10848	6268	0	10848
6.	item	7861	7861	13202	5541	13350
7.	loan	18264	18264	11808	0	18264
8.	member	1028	1028	981	841	1869
9.	master_author	3560	3560	10130	6570	10130
10.	master_publisher	595	595	599	4	599
11.	master_topic	2869	2869	9668	6799	9668
12.	stock_take_item	4558	4558	4558	0	4558

Dari hasil analisis tersebut dapat dijadikan sebagai acuan untuk pengujian validasi data dari *database* akasia yang telah melalui proses migrasi dan penggabungan *database.*

Analisis perbedaan struktur pada database senayan,cendana dan akasia

Tabel 3. Analisis perbedaan tabel database senayan dan akasia

No.	Perbedaan	Senayan	Cendana	Akasia	Keterangan
1.	Penggunaan *Primary Key*	Ya	Ya	Ya	Semua tabel pada ketiga *database* memiliki *primary key* dari sebuah kolom atau berupa *composite key*.
2.	Penggunaan *Foreign Key*	Tidak	Tidak	Tidak	Semua tabel pada *database* tidak menggunakan *foreign key.*
3.	Penggunaan *Constraint Check*	Tidak	Tidak	Tidak	Semua tabel pada ketiga *database* tidak menggunakan *Constraint Check.*
4.	Penggunaan *Index*	Ya	Ya	Ya	Semua tabel dapa ketiga *database* menggunakan index.
5.	Penggunaan *Auto Increment*	Ya	Ya	Ya	Untuk semua *primary key* yang berupa sebuah kolom.
6.	Jumlah Tabel	37	41	51	Terdapat penambahan tabel pada cendana yaitu tabel *department*, yang kemudian akan ditambahkan juga pada tabel akasia.
7.	Jumlah Kolom	287	305	393	Terdapat penambahan kolom pada cendana yaitu pada tabel biblio yang berupa kolom nim,*department_id*,pembimbing1 dan pembimbing2 yang kemudian akan ditambahkan juga pada tabel akasia.

3.3. Hasil Pengujian Kecepatan

Uji coba dilakukan dengan mengakses API yang telah dibuat untuk melihat kecepatan transfer yang dapat dilakukan. Uji coba dilakukan dengan menggunakan aplikasi postman untuk mengakses API, postman merupakan aplikasi untuk melakukan pengujian terhadap *web service* atau API.

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

Pengujian dilakukan empat kali dengan besar data yaitu 4000, 6000, 8000, dan 10000 record pada tabel biblio.

A. Uji Coba *Localhost*

Uji coba dilakukan dengan unit 01 dalam DBMS yang sama yaitu MySQL. Berikut hasil percobaan kecepatan migrasi data:

Tabel 4. Hasil uji coba kecepatan pada localhost

No.	Besar Data (Record)	Besar Data	Estimasi Waktu	Besar Data Sukses
1.	4000	405 Byte	477 ms	4000
2.	6000	405 Byte	817 ms	6000
3.	8000	405 Byte	1087 ms	8000
4.	10000	405 Byte	1415 ms	10000

B. Uji Coba LAN

Dari hasil uji coba melalui media LAN, berikut hasil percobaan kecepatan migrasi data:

Tabel 5. Hasil uji coba kecepatan dalam jaringan LAN simulasi

No.	Besar Data (Record)	Besar Data	Estimasi Waktu	Besar Data Sukses
1.	4000	436 B	2035 ms	4000
2.	6000	436 B	6967 ms	6000
3.	8000	420 B	7637 ms	8000
4.	10000	436 B	17183 ms	10000

C. Uji Coba LAN server produksi

Dari hasil uji coba melalui media LAN pada server produksi, berikut hasil percobaan kecepatan migrasi data:

Tabel 6. Hasil uji coba kecepatan dalam jaringan LAN server produksi

No.	Besar Data (Record)	Besar Data	Estimasi Waktu	Besar Data Sukses
1.	4000	356 B	1944 ms	4000
2.	6000	484 B	3949 ms	6000
3.	8000	340 B	5527 ms	8000
4.	10000	356 B	8704 ms	10000

3.4. Hasil Pengujian Validasi Data

Seperti yang telah dibahas sebelumnya untuk pengujian validasi data akan menggunakan metode RMSE untuk mengetahui tingkat error dari hasil migrasi database. Berikut detail hasil uji validasi :

Tabel 7. Tabel hasil pengujian validasi data localhost

No	Tabel Akasia	Yi	$\hat{Y}i$	$Yi - \hat{Y}i$	$(Yi - \hat{Y}i)^2$	RMSE
1.	*Biblio*	10934	10934	0	0	0
2.	*Biblio_Attachment*	182	182	0	0	0
3.	*Biblio Author*	10695	10695	0	0	0
4.	*Biblio Topic*	12612	12612	0	0	0
5.	*Fines*	10848	10848	0	0	0
6.	*Item*	13350	13350	0	0	0

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

No	Tabel Akasia	Yi	$\hat{Y}i$	$Yi - \hat{Y}i$	$(Yi - \hat{Y}i)^2$	RMSE
7.	*Loan*	18264	18264	0	0	0
8.	*Member*	1869	1869	0	0	0
9.	*Master Author*	10130	10130	0	0	0
10.	*Master Publisher*	599	599	0	0	0
11.	*Master Topic*	9668	9668	0	0	0
12.	*Stock Take Item*	4558	4558	0	0	0

Tabel 8. Tabel hasil pengujian validasi data LAN simulasi

No	Tabel Akasia	Yi	$\hat{Y}i$	$Yi - \hat{Y}i$	$(Yi - \hat{Y}i)^2$	RMSE
1.	*Biblio*	10934	10934	0	0	0
2.	*Biblio_Attachment*	182	182	0	0	0
3.	*Biblio Author*	10695	10695	0	0	0
4.	*Biblio Topic*	12612	12612	0	0	0
5.	*Fines*	10848	10848	0	0	0
6.	*Item*	13350	13350	0	0	0
7.	*Loan*	18264	18264	0	0	0
8.	*Member*	1869	1869	0	0	0
9.	*Master Author*	10130	10130	0	0	0
10.	*Master Publisher*	599	599	0	0	0
11.	*Master Topic*	9668	9668	0	0	0
12.	*Stock Take Item*	4558	4558	0	0	0

Tabel 1. Hasil pengujian validasi data LAN server produksi

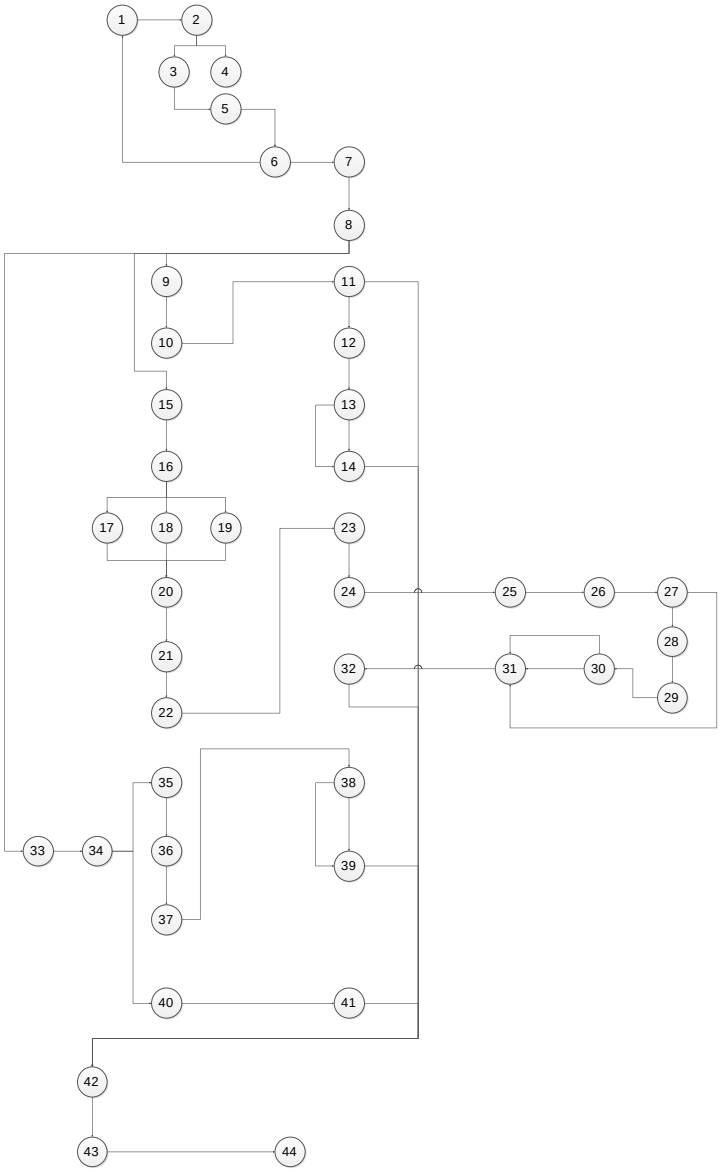
No	Tabel Akasia	Yi	$\hat{Y}i$	$Yi - \hat{Y}i$	$(Yi - \hat{Y}i)^2$	RMSE
1.	*Biblio*	10934	10934	0	0	0
2.	*Biblio_Attachment*	182	182	0	0	0
3.	*Biblio Author*	10695	10695	0	0	0
4.	*Biblio Topic*	12612	12612	0	0	0
5.	*Fines*	10848	10848	0	0	0
6.	*Item*	13350	13350	0	0	0
7.	*Loan*	18264	18264	0	0	0
8.	*Member*	1869	1869	0	0	0
9.	*Master Author*	10130	10130	0	0	0
10.	*Master Publisher*	599	599	0	0	0
11.	*Master Topic*	9668	9668	0	0	0
12.	*Stock Take Item*	4558	4558	0	0	0

Hasil dari pengujian validasi data menggunakan RMSE menunjukan angka yang sangat kecil yaitu 0 pada pengujian localhost , LAN simulasi, dan LAN pada server produksi. Dari hasil tersebut dapat dinyatakan bahwa *database* akasia dari hasil migrasi *localhost* sudah memiliki akurasi data yang cukup untuk dapat digunakan sebagai *database* sistem yang baru.

3.5. Hasil Pengujian White Box

Pada pengujian *whitebox* yang dilakukan pada sistem berdasarkan perancangan sistem yang digunakan. Pengujian akan dilakukan dengan menganalisis diagram alir sistem keseluruhan yang kemudian diubah kedalam *flowgraph* untuk menjadi *test case.* Berikut hasil pengujian white box:

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228



Gambar 6. Flowgraph sistem web service perpustakaan

Dari *test case* yang telah dibuat diatas penulis dapat melakukan pengujian *whitebox* dengan teknik pengujian *Cyclomatic Complexity*.

Hasil Pengujian Cyclomatic Complexity

Dari gambar 5.6. dapat diketahui bahwa:

Node(N) = 44, Edge(E) = 54, Region (R) = 12

Cyclomatic Complexity digunakan untuk mencari jumlah path dalam satu *flowgraph*. *Cyclomatic Complecity* V(G) untuk grafik alir dihitung dengan rumus:

V(G) = E – N + 2 Dimana: E = jumlah edge pada grafik alir N = jumlah node pada grafik alir

Maka V (G) = E – N + 2 –> V (G) = 54 – 44 + 2 = 12

Setelah didapatkan hasil V(G), setelah itu dilihat pada tabel hubungan *Cyclomatic Complexity* dan Risiko menurut McCabe, yaitu: Jadi *Cyclomatic Complexity* untuk flowgraph evaluasi adalah 12. Berdasarkan tabel hubungan antara *Cyclomatic Complexity* dan Risiko menurut McCabe, menunjukkan bahwa nilai CC 11 – 20 masuk dalam kategori *Moderate Risk* dimana memiliki strukturnya masih terbilang baik dan prosedur stabil, namun cukup berisiko.

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

3.6. Pembahasan

Sistem *web service* ini dibuat untuk megatasi permasalahan *redundancy data* yang terjadi pada sistem otomasi perpustakaan ITPLN yaitu sistem Senayan dan Cendana. Sistem ini dibuat dalam rangka untuk mempersiapkan *database* untuk pengimplimentasian sistem otomasi baru untuk perpustakaan ITPLN yaitu Sistem Akasia. Aplikasi *web service* ini dapat melakukan migrasi dan penggabungan data pada *database* Senayan dan Cendana ke *database* Akasia. Sistem ini juga mendukung pembuatan field dan table tambahan untuk field dan table yang belum tersedia pada struktur awal *database* akasia.

Penerapan arsitektur REST disini adalah untuk menguji kemampuan arsitektur REST dalam menangani migrasi dan penggabungan data dengan jumlah data yang banyak. Pengujian dilakukan secara *localhost* dan melalui jaringan LAN yang ada dalam gedung ITPLN. Terdapat dua macam pengujian yaitu pengujian kecepatan dan pengujian validasi. Pengujian kecepatan dilakukan dengan menghitung estimasi waktu yang dipakai saat transfer data dengan jumlah data berkisar antara 4000, 6000, 8000, 10000. Hasilnya adalah, dengan pemanfaatan arsitektur REST dalam melakukan migrasi data dapat mempersingkat waktu migrasi karena data yang ditransfer dari *database* sumber berformat JSON yang memiliki ukuran yang lebih kecil dibanding data dalam format SQL. Data dalam format JSON tersebut kemudian diubah kedalam bentuk *array* oleh RESTful API untuk kemudian dimasukkan ke dalam *database* akasia. Sehingga proses migrasi memiliki waktu yang singkat karena proses transfer data dari *server database* sumber sampai ke *server* RESTful API dalam format JSON[15]. Sedangkan hasil dari pengujian validasi data menggunakan RMSE menunjukan angka yang sangat kecil yaitu 0. Dari hasil tersebut dapat dinyatakan bahwa *database* akasia dari hasil migrasi *localhost* sudah memiliki akurasi data yang cukup untuk dapat digunakan sebagai *database* sistem yang baru. Setelah melakukan berbagai uji coba tersebut, maka dapat disimpulkan bahwa migrasi yang akan digunakan adalah hasil migrasi localhost dimana migrasi hanya dilakukan satu kali saja dari hasil percobaan tersebut atas permintaan Manager perpustakaan.

Hasil yang diperoleh adalah pemanfaatan arsitektur REST terbukti sangatlah tepat untuk melakukan migrasi dan penggabungan data. Dari hasil pengujian kecepatan RESTful API hanya membutuhkan waktu dalam hitungan detik untuk melakukan migrasi dan pengggabungan *database* dengan validasi data yang memiliki tingkat error 0.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil dari pengujian arsitektur REST pada sistem web service untuk migrasi dan penggabungan data pada sistem otomasi perpustakaan ITPLN dapat diambil kesimpulan sebagai berikut:

1. *Redundancy* data yang terjadi pada data sistem otomasi perpustakaan ITPLN dapat diatasi dengan cara melakukan pencocokan terhadap key-key yang mewakili satu record tersebut.
2. Dengan pengimplementasian arsitektur REST dalam sistem web service untuk migrasi dan penggabungan data pada database sistem otomasi perpustakaan ini menunjukan tingkat kompeksitas sistem secara keseluruhan yang dihitung menggunakan metode *Cyclomatic Complexity* berada pada angka dua belas, yang berarti kompleksitas sistem adalah termasuk kedalam kategori *Moderate Risk* jika dibandingkan dalam tabel *Cyclomatic Complexity* and Risk menurut McCabe. Dimana berarti sistem secara keseluruhan memiliki algoritma yang cukup stabil namun berisiko.
3. Proses migrasi dan penggabungan database yang menggunakan arsitektur *RESTful API* ini terbukti berhasil diterapkan pada sistem dengan hasil validasi data yang dihitung berdasarkan tingkat error metode RMSE menunjukan angka 0, dan penggunaan arsitektur RESTful API

Vol. 15, No. 1, Maret 2022, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v15i1.1228

dalam proses transfer terbukti lebih efisien karena berdasarkan uji coba dari tiga skenario yang dilakukan yaitu skenario localhoST., LAN simulasi, dan LAN server produksi menunjukan waktu yang sangat singkat yaitu dibawah satu menit untuk melakukan migrasi sebuah tabel dengan variasi jumlah data bervariasi antara 4000, 6000, 8000, dan 10000 data.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Institut Teknologi PLN dan Fakultas Telematika Energi serta Lembaga Penelitian Pengabdian Masyarakat yang telah memberi dukungan yang membantu pelaksanaan penelitian dan atau penulisan artikel.

DAFTAR PUSTAKA

[1] Ridho, M. R. Senayan Library Management System for dummies. Jurnal Maju Mundur, 2(3), 1–10. Retrieved from http://slims.web.id. 2018

[2] Tihomirovs, J., & Grabis, J.Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics. Information Technology and Management Science, 19(1), 92–97. https://doi.org/10.1515/itms-2016-0017. 2017

[3] Connolly, T., & Begg, C.Database Systems. Pearson Education. 2005. Ed.4th

[4] Flanders, J. RESTful .NET. Retrieved from https://books.google.com/books?id=9T6N8NPHqLEC&pgis=1. 2008. Vol 28

[5] Gootschalk, K. Introduction to Web Services Architecture. IBM SYSTEMS JOURNAL, 41(2). https://doi.org/10.1147/sj.412.0170. 2015

[6] Nandang Mulyadi, I. R. Implementasi Rest Web Service Dan Saran Peminjaman. 2015

[7] Nidhra, S. Black Box and White Box Testing Techniques - A Literature Review. International Journal of Embedded Systems and Applications, 2(2), 29–50. https://doi.org/10.5121/ijesa.2012.2204. 2012

[8] Bansel, A., Gonzalez-Velez, H., & Chis, A. E. Cloud-Based NoSQL Data Migration. Proceedings - 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2016, 224–231. https://doi.org/10.1109/PDP.2016.111E. Kamel and A. M. Memari, "State-of-the-Art Review of Energy Smart Homes," J. Archit. Eng., vol. 25, no. 1, p. 03118001, 2019, doi: 10.1061/(asce)ae.1943-5568.0000337. 2014

[9] Febriana, C. E., & Sudibyo, U. Penerapan Data Mining Untuk Estimasi Nilai Matematika Dengan Menggunakan Algoritma Regresi Linier Pada SMA Kesatrian 1 Semarang, 2016

[10] Hartono, N., Utami, E., & Amborowati, A. Migrasi dan Optimalisasi Database Sistem Informasi Manajemen Universitas Cokroaminoto Palopo. Jurnal Buana Informatika, 7(4), 255–264. https://doi.org/10.24002/jbi.v7i4.766. 2017

[11]Pressman, R. S. SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH (7th Editio). New York: McGraw-Hill. 2010

[12] Swain, R. K., Panthi, V., & Behera, P. K.Generation of test cases using activity diagram. 2013

[13] Rianto, R. PENGEMBANGAN LAYANAN WEB SERVICE DATA AKADEMIK. JURNAL TEMATIKA, VOL. 5. 2017

[14]Prettyman, S. Learn PHP 7 Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL. https://doi.org/10.1007/978-1-4842-1730-6. 2015

[15] Smith, B. Beginning JSON. Beginning JSON. Apress. https://doi.org/10.1007/978-1-4842-0202-9. 2015