

BAB I

PENDAHULUAN

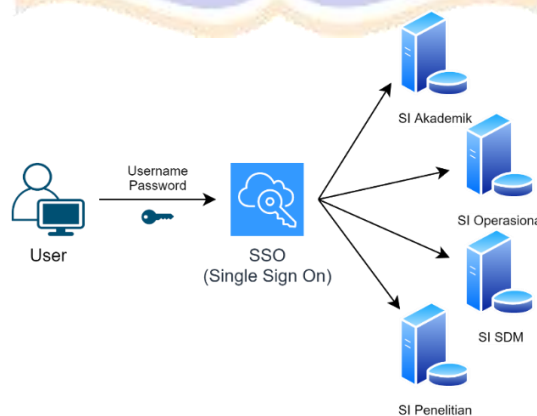
1.1 Latar Belakang

Kebutuhan pengguna mengharuskan data diperoleh dengan cepat dan pengolahan data memiliki akurasi yang tinggi serta integritas data yang baik, sehingga perkembangan sistem informasi dirancang sesuai dengan kebutuhan pengguna. Pertumbuhan data dari sistem informasi saat ini berbanding lurus dengan kebutuhan pengguna akan sistem informasi. Data yang tumbuh menjadi informasi yang dapat digunakan untuk mengambil keputusan, dan terus berkembang dari sekadar pencatatan data hingga penggalian beberapa sumber data (Oktavianti, 2019). Saat ini, pertumbuhan data yang terus meningkat tidak dapat lepas dari banyaknya sistem informasi yang digunakan, terlebih lagi dimana dalam organisasi atau instansi memiliki lebih dari satu sistem informasi yang belum terintegrasi satu dengan yang lainnya. Sistem yang banyak memerlukan sebuah integrasi data dari satu sistem ke sistem lainnya. Sistem informasi yang tidak terintegrasi membawa dampak negatif pada kualitas data, sehingga data induk sering ditemui tidak sama pada setiap sistem informasi.

Pada penelitian Dewi (2018), hasil dari integrasi data dapat meningkatkan rata-rata deviasi waktu proses mencapai 56,98 detik serta mempercepat proses pengisian data hingga 80,7%. Oleh sebab itu integrasi data merupakan salah satu konsep kunci keberhasilan dari manajemen sistem informasi (Blanchard, 1998). Data dan informasi pada sistem informasi sangat bermanfaat apabila saat diperlukan oleh sistem lainnya dapat diakses dengan cepat dan *real time*. Integrasi

data juga membuat *output* dalam suatu sistem dapat menjadi *input* bagi sistem lainnya. Capaian yang didapatkan dari proses integrasi suatu sistem yaitu kualitas data yang dihasilkan menjadi lebih baik dan akurat, selain itu membaiknya arus informasi dalam sebuah instansi maupun organisasi.

Universitas Udayana sebagai Perguruan Tinggi Badan Layanan Umum (PTN BLU), mewajibkan untuk melakukan segala bentuk layanan akademik, non akademik dan operasional menggunakan sistem informasi. Dalam pelaksanaannya, sejak tahun 2015 Universitas Udayana telah menggunakan sistem informasi berbasis *Single Sign On* (SSO) yang disebut IMISSU (*Integrated Management Information System, the Strategic of UNUD*) seperti ditunjukkan pada Gambar 1.1 (Krisnandari dkk., 2019). Hingga 2021, Universitas Udayana telah memiliki 52 sistem informasi dimana terdiri dari 35 sistem informasi inti (*core*) yang digunakan untuk menunjang operasional dan 17 sistem informasi pendukung (*support*) sebagai penunjang dari sistem inti. Selain itu Universitas Udayana juga memiliki 85 Website unit kerja, fakultas dan program studi. Seluruh sistem informasi tersebut digunakan untuk menjalankan kegiatan akademik, kemahasiswaan, operasional dan lain-lain di lingkungan Universitas Udayana.



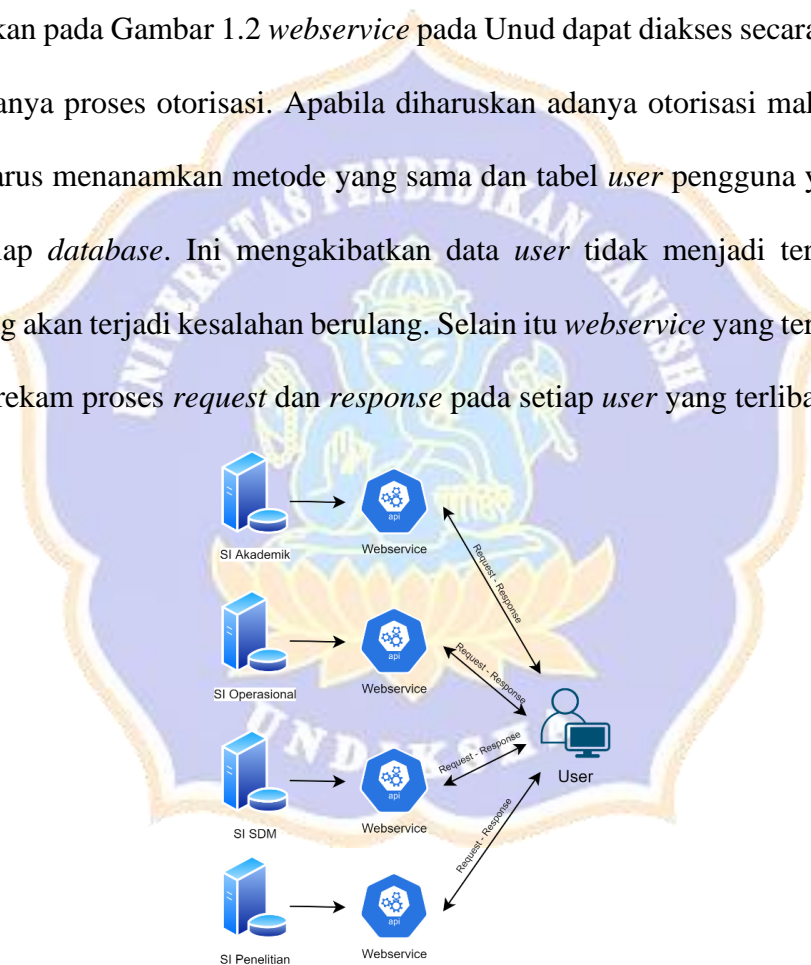
Gambar 1.1 SSO pada IMISSU Universitas Udayana

Saat ini seluruh sistem informasi yang dikembangkan Universitas Udayana masih menggunakan model Arsitektur Monolitik (*Monolithic Architecture*). Arsitektur monolitik merupakan model pengembangan sistem informasi tradisional, dimana seluruh fungsionalitas baik dari *webservice*, *database* dan proses bisnis dienkapsulasi ke dalam satu aplikasi tunggal, sehingga setiap modul tidak dapat dieksekusi secara independen (Ponce dkk., 2019). Permasalahan yang timbul dalam jangka panjang yaitu saat sistem telah memiliki jumlah data yang banyak dan pertumbuhan data yang cepat, sehingga informasi yang dihasilkan dituntut harus dapat mendukung sistem informasi lainnya. Saat ini proses integrasi data antar sistem informasi di Universitas Udayana masih menggunakan metode tradisional dengan *join* antar tabel antar *database* dan membuat banyak koneksi ke *database* secara langsung pada program sehingga menurunkan performa dari sistem informasi karena proses pemecahan virtualisasi *server* berdasarkan beban *server* menjadi sulit dilakukan. Dengan penurunan performa tersebut, maka penambahan kapasitas *server* menjadi solusi yang harus dipilih untuk meningkatkan performa dari sistem informasi. Selain itu, ada beberapa *client* yang sudah membuat *webservice* sederhana, dimana *webservice* tersebut tidak mengikuti aturan penulisan *Restful API*, sehingga variasi dalam penerapannya membuat kesulitan terhadap implementasi, hal ini disebabkan juga karena tidak adanya dokumentasi dalam penerapan *webservice* sehingga setiap melakukan integrasi sistem selalu terkendala dalam pemahaman atau mengakitbatkan *miss parameter* dan tidak sesuai dalam arsitektur *database* yang ada.

Salah satu metode integrasi data yaitu *Service Oriented Architecture* (SOA). SOA merupakan metode pendekatan arsitektur dengan pengembangan yang cepat dan berbiaya yang rendah dan memiliki kemampuan dalam mengatur aplikasi terdistribusi dalam lingkungan yang heterogen (Hantana, 2013). Dalam perkembangannya, integrasi data dalam sebuah sistem informasi berdampak pada muara akhir dimana data akan mulai diperlukan oleh pihak eksternal maupun dari internal itu sendiri. Untuk dapat memberikan informasi data tanpa harus masuk ke dalam *database* internal maka diperlukan *microservices*. Menurut Larrucea dkk. (2018), *microservices* merupakan aplikasi kecil dengan satu tanggung jawab yang dapat diterapkan, diskalakan, dan diuji secara independen.

SOA memiliki keunggulan dalam memungkinkan aplikasi untuk berkomunikasi dengan aplikasi lain yang berbeda *platform* dan teknologi dengan mudah serta dapat mempercepat pengembangan aplikasi karena layanan yang sudah dibangun dapat digunakan kembali pada aplikasi yang berbeda. Selain itu, SOA juga memudahkan dalam penyesuaian aplikasi dengan kebutuhan bisnis yang berubah-ubah dan penambahan atau pengurangan kapasitas aplikasi dengan mudah. SOA dan *Webservices* adalah dua teknologi yang digunakan untuk membangun aplikasi modern. SOA memiliki keunggulan karena aplikasi dapat dibangun dengan modul-modul terpisah yang mudah dikembangkan dan dikelola secara terpisah. *Webservices* lebih terfokus pada protokol-protokol komunikasi antar aplikasi seperti SOAP dan REST. Perbandingan tersebut menyimpulkan bahwa SOA memiliki keunggulan lebih fleksibel dan memiliki interoperabilitas yang lebih baik dibandingkan *Webservices*.

Metode SOA merupakan metode yang paling sesuai digunakan pada kondisi *webservice* di Universitas Udayana. Saat ini seluruh *endpoint webservice* pada Unud dapat diakses secara langsung tanpa melalui proses otentikasi keamanan. Hal ini memungkinkan *webservice* rentan terhadap permasalahan keamanan seperti data yang dapat diakses oleh pihak ketiga secara langsung ataupun pihak-pihak yang mengetahui *endpoint webservice* tersebut (Saputra & Setianto, 2020). Seperti ditunjukkan pada Gambar 1.2 *webservice* pada Unud dapat diakses secara langsung tanpa adanya proses otorisasi. Apabila diharuskan adanya otorisasi maka seluruh sistem harus menanamkan metode yang sama dan tabel *user* pengguna yang sama pada setiap *database*. Ini mengakibatkan data *user* tidak menjadi terpusat dan cenderung akan terjadi kesalahan berulang. Selain itu *webservice* yang tersedia juga tidak merekam proses *request* dan *response* pada setiap *user* yang terlibat.



Gambar 1.2 Arsitektur *Webservice* pada Universitas Udayana

Dengan mengubah arsitektur *webservice* Unud ke dalam bentuk arsitektur SOA maka akan mengurangi permasalahan seperti tidak adanya proses perekaman *user* terhadap proses *request* dan *response* data pada *webservice*. Selain itu

webservice yang terpusat dengan SOA dapat dengan mudah memblokir lalu lintas *user* tanpa identitas, melakukan blokir pada IP tertentu, mengidentifikasi pola penggunaan API, dan memantau aktifitas lalu lintas API.

Berdasarkan uraian masalah diatas dilakukan implementasi perubahan proses integrasi data dengan membuat *service* kecil pada setiap sistem informasi menggunakan metode SOA. *Service* yang telah dibuat kemudian disimpan pada *management microservices* untuk dikonversi kedalam bentuk penulisan REST API sehingga menjadi layanan *webservice* satu pintu. Kelemahan-kelemahan yang ada pada *monolithic architecture* dapat ditangani dengan penerapan *webservice architecture* (Sasidharan & Kumar N, 2018). Pada penelitian (Munawar & Hodijah, 2018), keberhasilan proses migrasi dari sistem informasi berbasis monolitik ke bentuk arsitektur *microservices* membuat sistem lebih adaptif (*adaptability*) terhadap perubahan kebutuhan (*requirement changes*). Pengembangan manajemen *microservices* berbasis SOA diharapkan mampu menyelesaikan permasalahan layanan integrasi data di lingkungan Universitas Udayana.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah diatas, berikut ini adalah identifikasi masalah yang akan dijadikan bahan penelitian sebagai berikut.

1. Sistem informasi Universitas Udayana masih menggunakan pola integrasi tradisional dengan melibatkan *resource database* secara langsung dan berakibat pada meningkatnya beban *server*.

2. *Webservice* Universitas Udayana memiliki karakteristik dan pola yang berbeda sehingga timbul masalah pada keseragaman aturan *request*, *response* dan pesan *error*.
3. Universitas Udayana belum memiliki sistem untuk pengelolaan *webservice* dan dokumentasi seluruh API untuk memonitoring segala kegiatan yang berhubungan dengan *webservice*.

1.3 Batasan Masalah

Pengembangan integrasi data dengan *management microservices* berbasis SOA sangat luas cakupannya. Untuk itu dalam penelitian ini batasan masalah dibatasi sebagai berikut.

1. Layanan *Management microservices* digunakan untuk menempatkan *service-service* kecil berupa *webservice* dari obyek penelitian sehingga menjadi layanan *service* satu pintu terintegrasi.
2. Standarisasi *webservice* pengembangan *management microservice* yang digunakan pada penelitian ini menggunakan *Restful API*.
3. Metode yang digunakan dalam proses integrasi data menggunakan *Service Oriented Architecture (SOA)*.

1.4 Rumusan Masalah

Berdasarkan uraian latar belakang masalah di atas, maka dapat dirumuskan permasalahan sebagai berikut.

1. Bagaimana rancang bangun *management microservices* dengan menerapkan metode SOA di Universitas Udayana?

2. Bagaimana implementasi setiap *webservice* ke *management microservices* dengan menerapkan SOA?
3. Bagaimana hasil pengujian setiap *webservice* yang terlibat terhadap *client service* dengan menerapkan SO?
4. Bagaimana hasil analisis kompatibilitas *management microservices* pada setiap *webservice* yang terlibat?
5. Bagaimana hasil analisis kinerja sistem untuk meningkatkan proses kinerja dengan menerapkan SOA pada *management microservices*?

1.5 Tujuan Penelitian

Adapun tujuan yang diharapkan dapat dicapai dari penelitian ini adalah sebagai berikut.

1. Menghasilkan rancangan dan aplikasi *management microservices* dengan menerapkan metode SOA di Universitas Udayana.
2. Mengetahui hasil implementasi setiap *webservice* ke *management microservices* dengan menerapkan SOA.
3. Mengetahui hasil pengujian *webservices* yang terlibat terhadap *client services* dengan menerapkan SOA.
4. Menganalisis kompatibilitas *management microservices* pada setiap *webservices* yang terlibat.
5. Menganalisis kinerja sistem untuk meningkatkan proses kinerja dengan menerapkan SOA pada *management microservices*.

1.6 Manfaat Penelitian

Adapun manfaat penelitian ini dapat dikelompokkan menjadi dua, yaitu manfaat teoretik dan manfaat praktis.

1.6.1 Manfaat Teoretik

Adapun manfaat teoretik yang mungkin timbul dari penelitian ini adalah sebagai berikut.

1. Pengembangan *management microservices* dengan menerapkan SOA dapat memberikan manfaat untuk setiap proses pengembangan sistem informasi selanjutnya bagi Universitas Udayana karena setiap *service* memiliki dokumentasi yang akurat sehingga data induk dipastikan sama pada setiap sistem informasi yang dikembangkan.
2. Penerapan SOA memungkinkan organisasi untuk mencapai fleksibilitas, interoperabilitas, dan *reusable components* yang lebih tinggi dalam pengembangan perangkat lunak.

1.6.2 Manfaat Praktis

Adapun manfaat praktis yang mungkin timbul dari penelitian ini adalah sebagai berikut.

1. Mampu memberikan tawaran solusi pada Universitas Udayana akan pembaharuan mekanisme proses integrasi yang baik dengan memanfaatkan SOA. Selain itu dari manfaat dari proses integrasi yang baik membuat kualitas data yang dihasilkan menjadi lebih baik dan *real time* sehingga dapat digunakan untuk proses pengambilan keputusan dan kebijakan pada level manajemen tertinggi.

2. Hasil rancangan arsitektur pengembangan *microservices management* dapat digunakan sebagai dasar pengembangan *microservices* skala besar di penelitian selanjutnya seperti pengembangan *load balance* dan *failover microservices*.

