

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perangkat lunak berbasis layanan adalah salah satu arsitektur yang populer saat ini, di mana *Microservices Architecture* menjadi pendekatan yang paling banyak digunakan. Arsitektur ini cocok untuk menciptakan perangkat lunak yang independen, memudahkan pengembangan dalam skala besar, dan mengurangi ketergantungan antar layanan (*service*) [1]. Dengan *microservices*, tim pengembang dapat fokus pada pengembangan, penerapan, dan pemeliharaan setiap layanan tanpa harus berhadapan dengan kompleksitas keseluruhan sistem. Salah satu pola desain yang sering diterapkan dalam arsitektur ini adalah *Event-Driven Architecture* (EDA), yang mengatur komunikasi antar layanan secara asinkron melalui mekanisme pengiriman dan penerimaan *event* [2].

Aplikasi analisis media sosial merupakan sebuah sistem *big data* yang menganalisis percakapan di media sosial X (Twitter). Pada tahap awal pembangunannya menggunakan jenis arsitektur *monolithic*. Arsitektur ini menerapkan aspek fungsional yang menggunakan *code base* dan teknologi yang seragam dalam penerapannya. Hal ini terjadi dikarenakan tuntutan aplikasi untuk dapat dibangun secara cepat dan keterbatasan jumlah tim yang tersedia. Berdasarkan wawancara yang telah dilakukan kepada *developer* aplikasi analisis sosial media, penggunaan arsitektur *monolithic* saat ini membuat *developer* kesulitan dalam melakukan proses pengembangan aplikasi [Lampiran B]. Selain itu, muncul masalah lain terkait muncul *web service* masih belum optimal, hal ini dibuktikan ketika fungsional *traffic* yang terus meningkat. Hal ini dibuktikan dengan melakukan pengujian performa pada seluruh *web services* menggunakan tool *blazemeter* pada aplikasi analisis sosial. Sehingga saat fungsional mengalami *error* maka akan berdampak pada keseluruhan *web services*.

Microservices adalah desain arsitektur aplikasi yang memecah aplikasi menjadi *service-service* kecil yang terpisah sesuai dengan fungsinya. pemecahan ini bertujuan agar setiap *service* tidak memiliki ketergantungan yang tinggi dan

memungkinkan setiap bagian *service* ini memiliki bahasa pemrograman dan teknologi yang berbeda. Dalam penerapan *microservices* untuk memecahkan aplikasi *monolith* menjadi *service-service* nya sendiri menggunakan metode *Domain Driven Design (DDD)* [3]. Penerapan *microservices* sebagai arsitektur perangkat lunak dengan menggunakan bertujuan untuk membuat *web services* yang *scalable*, *resilience*, dan *high-availbilty*. Penerapan *microservices* diharapkan dapat menanganani masalah yang terjadi pada aplikasi analisis sosial media.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, dapat dirumuskan suatu permasalahan yaitu apakah dengan menerapkan arsitektur *microservices* dapat meningkatkan performa dan skalabilitas pada *web services* aplikasi analisis sosial media.

1.3 Maksud dan Tujuan

Penelitian ini bermaksud untuk menerapkan arsitektur *microservices* berdasarkan *services* yang sudah dibuat oleh aplikasi analisis sosial media, dengan tujuan meningkatkan skalabilitas dan performa pada aplikasi analisis sosial media.

1.4 Batasan Masalah

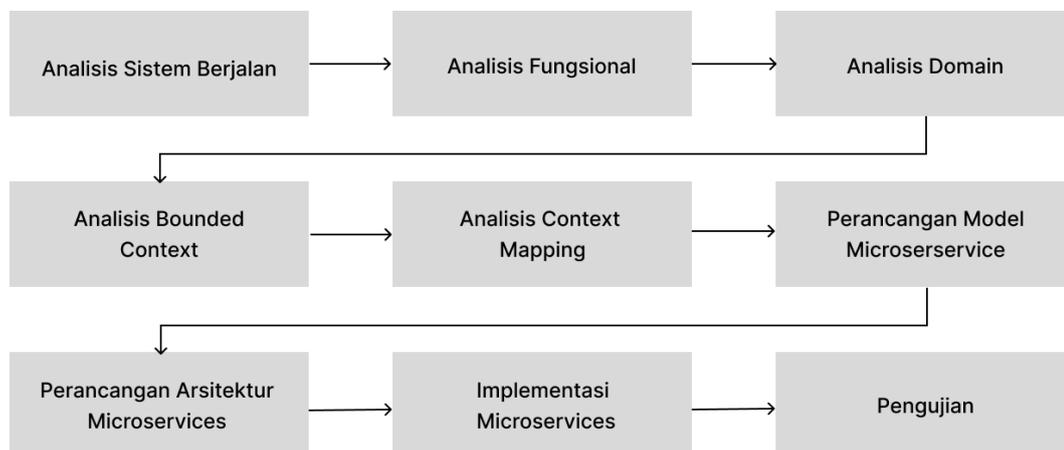
Agar penelitian tidak menyimpang dari tujuan, telah ditentukan batasan masalah sebagai berikut :

1. Tidak membahas faktor-faktor lain yang memungkinkan mempengaruhi performa aplikasi di luar arsitektur, seperti infrastruktur jaringan atau sumber daya perangkat keras
2. Pada proses pertukaran informasi antar *client* dengan *server* menggunakan format *JSON* dengan mekanisme *REST*
3. Menggunakan Pola Arsitektur *Event Drivent Architecture* disetiap komunikasi antar *service*.
4. Dalam penelitian ini analisis yang dilakukan fokus kepada performa yang tercatat dalam *Quality Of Services (QOS) Web services*.

5. Pendekatan yang digunakan untuk memecah *microservices* aplikasi menggunakan pendekatan *Domain Driven Design* [2]

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah dengan pendekatan kuantitatif yang menggunakan jenis penelitian komparatif. Penelitian ini bertujuan membandingkan antara dua objek dengan variabel tertentu. Dalam kasus ini objek yang dibandingkan merupakan tingkat performa *web services* pada Aplikasi Analisis Media Sosial yang menggunakan arsitektur sebelumnya (*monolithic*) dengan arsitektur yang akan diterapkan yaitu *Microservices*. Berikut adalah tahapan-tahapan yang akan dilakukan dalam penelitian ini mengacu pada buku [3] [4] dapat dilihat pada Gambar 1.1



Gambar 1.1. Metodologi Penelitian

1. Analisis Sistem Berjalan

Pada tahap ini, dilakukan analisis terhadap arsitektur sistem yang sudah ada. Tujuannya adalah untuk memahami fungsionalitas, komponen, dan interaksi dalam sistem yang sedang berjalan. Hal ini membantu dalam mengidentifikasi masalah yang terjadi.

2. Analisis Fungsional

Tahap ini berfokus pada identifikasi kebutuhan fungsional sistem. Kebutuhan ini mencakup semua fitur dan fungsi yang harus disediakan oleh sistem untuk memenuhi kebutuhan pengguna.

3. Analisis *Domain*

Pada tahapan Analisis *Domain* dalam metodologi penelitian arsitektur *microservices*, terdapat tiga langkah utama: *Ubiquitous Language*, *Event Storming*, dan Identifikasi *Domain*. *Ubiquitous Language* memastikan tidak ada ambiguitas dalam komunikasi, sehingga semua anggota tim memiliki pemahaman yang sama tentang proses bisnis. Setelah memiliki pemahaman yang sama, proses selanjutnya adalah *Event Storming*, yang digunakan untuk memetakan alur proses bisnis secara *visual* serta membantu dalam pembuatan *Bounded Context*. Tahap terakhir adalah Identifikasi *Domain*, yaitu memetakan *Domain* apa saja yang ada dalam sistem berdasarkan hasil dari tahap sebelumnya.

4. Identifikasi *Bounded Context*

Pada langkah ini, dilakukan identifikasi *Bounded Context*, yaitu batas-batas yang jelas untuk setiap konteks yang telah diidentifikasi dalam tahapan analisis *Domain*. *Bounded Context* memastikan bahwa setiap bagian dari sistem memiliki tanggung jawab yang jelas dan terisolasi dari bagian lain. Ini penting dalam arsitektur *microservices* untuk menghindari ketergantungan yang kompleks.

5. Identifikasi *Context Mapping*

Tahap ini melibatkan pemetaan hubungan antar *Bounded Context*. *Context mapping* membantu memahami bagaimana berbagai bagian sistem saling berinteraksi dan berkomunikasi. Ini penting untuk merancang komunikasi antar layanan dalam arsitektur *microservices*.

6. Perancangan *Domain Model Microservices*

Pada tahap ini, dilakukan perancangan *Model Domain* untuk setiap *microservice*. *Model Domain* mencakup entitas, nilai objek, dan agregat yang relevan dalam setiap *Bounded Context*.

7. Perancangan *Microservices*

Tahap ini berfokus pada perancangan arsitektur keseluruhan *microservices*. Ini mencakup definisi layanan, pola komunikasi antar layanan, perancangan basis data, dan mekanisme orkestrasi atau *choreography*.

8. Implementasi *Microservices*

Pada tahap ini hal yang dilakukan yaitu menerapkan *microservices* yang telah dirancang sebelumnya dalam bentuk kode dan *web services*.

9. Pengujian

Pada tahap ini dilakukan pengujian aplikasi, pengujian dilakukan untuk menguji apakah aplikasi sudah berjalan sesuai dengan *Domain* yang telah di rancang, kemudian dilanjutkan dengan pengujian performa *API* yang telah dibuat. Pengujian performa *web services* pada penelitian mengguna dua skema pengujian yaitu *Unit Testing* dan *Performance Testing*

1.6 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini dibagi menjadi 5 bab yang tersusun secara sistematis untuk memberikan gambaran umum mengenai penelitian yang dikerjakan.

BAB 1 Pendahuluan

Pada bab ini akan menjelaskan mengenai latar belakang masalah, perumusan masalah, maksud dan tujuan, batasan masalah, metodologi penelitian dan juga sistematika penulisan.

BAB 2 Landasan Teori

Pada bagian ini akan menjelaskan teori – teori dasar yang akan digunakan pada penelitian ini. Teori – teori disini adalah teori yang berkaitan dan dalam lingkup tema penelitian ini seperti *reengineering*, *microservices*, pembangunan perangkat lunak dan lain sebagainya.

BAB 3 Analisis dan Perancangan Sistem

Pada bab ini akan menjelaskan tentang analisis permasalahan dan perancangan perangkat lunak yang akan dibangun kembali.

BAB 4 Implementasi dan Pengujian Sistem

Bab ini akan membahas tentang pengimplementasian aplikasi yang baru dan pengujian sistem yang baru untuk memastikan sesuai dengan ketentuan yang telah ditentukan diawal.

BAB 5 Kesimpulan dan Saran

Pada bab terakhir ini akan menjelaskan kesimpulan yang diperoleh dari hasil keseluruhan penelitian ini dan juga saran untuk penelitian ini kedepannya.