BAB I
PENDAHULUAN

1.1. Latar Belakang

Microservices Architecture (*MS*A) merupakan pendekatan yang populer dalam pengembangan perangkat lunak modern. *MS*A adalah sebuah teknik atau pendekatan dalam pengembangan aplikasi tunggal dengan menggunakan serangkaian layanan kecil yang berjalan pada proses terpisah dan berkomunikasi dengan mekanisme yang ringan, seperti API dengan sumber daya HTTP. Setiap layanan dibangun berdasarkan kemampuan bisnis dan dapat di-deploy secara independen melalui mekanisme deployment yang sepenuhnya otomatis. Manajemen terpusat dari layanan-layanan ini sangat minim, dan layanan-layanan tersebut dapat ditulis dalam bahasa pemrograman yang berbeda serta menggunakan teknologi penyimpanan data yang berbeda-beda (Martin Fowler & James Lewis, 2014). *MS*A mendefinisikan setiap layanan adalah sebuah layanan yang independen yang memiliki databasenya sendiri.

Pada prakteknya, *Microservices* terbagi dalam pola interaksi *centralization* dan *decentralization* yang disebut sebagai *Orchestration Pattern* dan *Choreography Pattern* secara berurutan. Dalam konsep tersebut menunjukan bagaimana sebuah *service* berkolaborasi dengan *service* lain, bagaimana business prosessnya dibangun, dan seperti apa urutan aktivitasnya (Cerny, Donahoo, and Pechanec 2017). *Microservices Orchestration* mengacu pada pola interaksi dimana business prosesnya terpusat pada sebuah *controller* (*orchestrator*) yang bertindak untuk mengkordinasikan atau mengarahkan tiap aktivitas yang berbeda pada *Microservices* dan mengkombinasikanya menjadi satu, sehingga menghasilkan sebuah fungsi tertentu. Dalam *Orchestration* pola interaksi antar layanannya dilakukan secara sinkronus yang berarti setiap aktivitas bisnis prosesnya dijalankan secara berurutan. Berbeda halnya dengan *Choreography* yang tidak memiliki element yang terpusat, sehingga logika bisnisnya tersebar pada masing-masing *Microservices*. Dalam *Choreography* layanan berkomunikasi dengan layanan lain

menggunakan protokol *messaging* secara asinkronus, Setiap layanan mampu menyelesaikan tugasnya tanpa harus menunggu layanan lainnya. Dengan kata lain, setiap layanan memiliki tanggung jawab yang jelas dan mampu bekerja secara independen.

Dalam konteks pengembangan perangkat lunak tidak ada satupun pendekatan yang cocok untuk menangani segala jenis skenario. Setiap pendekatan memiliki keunggulan dan kelemahannya masing-masing yang perlu dievaluasi untuk memastikan bahwa kita memilih pendekatan yang paling sesuai dengan kebutuhan spesifik aplikasi atau sistem yang sedang dikembangkan. Saat ini *Microservices* telah menjadi pendekatan yang populer dalam pengembangan aplikasi skala besar seperti *e-commerce*. Dalam skenario *e-commerce* terdapat banyak layanan yang saling berkolaborasi dan memiliki ketergantungan yang erat satu sama lain. Semakin banyaknya keterlibatan antar layanan maka semakin tinggi tingkat kompleksitas suatu sistem, sehingga dapat mempengaruhi kinerja sistem tersebut secara keseluruhan. Salah satu skenario yang melibatkan banyaknya kolaborasi antar layanan adalah skenario pemesanan dan transaksi pada *e-commerce*. Pada skenario tersebut, sistem harus mampu menangani proses pemesanan hingga transaksi yang berhasil maupun situasi dimana terjadi pembatalan transaksi. Proses ini melibatkan beberapa tindakan, seperti memperbarui status pesanan, membatalkan penagihan, hingga mengembalikan stok produk ke nilai semula. Dalam hal ini, terdapat proses kolaborasi antar layanan yang kompleks, dimana banyak layanan saling bergantung satu sama lain. Oleh karena itu penting untuk menentukan pola *Microservices* mana yang lebih baik digunakan pada skenario tersebut.

Pada penelitian ini bertujuan untuk mebandingkan pendekatan *Microservices Orchestration* dan *Choreography Pattern* pada skenario pemesanan dam transaksi pada kasus *e-commerce* mulai dari transaksi yang berhasil hingga transaksi yang dibatalkan. Dalam skenario ini terdapat beberapa *Microservices* yang terlibat yaitu *Order Service, Product Service, Billing Service, User Service, Payment Service*, dan *Shipping Service*. Dalam skenario tersebut, setiap layanan memiliki ketergantungan dengan layanan lain sehingga aktivitas harus diselesaikan secara berurutan. Untuk membandingkan kedua pendekatan ini, penulis akan melakukan

pengukuran kompleksitas dengan representasi *Business process Model and Notations* (BPMN) menggunakan *Controll-Flow Complexity Metrics* dan *Lines of Code Metrics*, serta pengujian terkait kinerja dan *Load Testing* sistem pada setiap layanan dalam skenario. Penulis akan mengevaluasi seberapa efisien kedua pendekatan tersebut dapat menangani skenario pemesanan dan transaksi, baik transaksi yang berhasil hingga transaksi yang dibatalkan.

Dengan membandingkan kedua pendekatan tersebut, penulis berharap dapat memberikan rekomendasi yang jelas dan berdasarkan data untuk memilih teknik yang lebih tepat dalam pengembangan aplikasi kedepan. Hasil dari penelitian ini akan menjadi kontribusi penting untuk memahami kelebihan dan kekurangan masing-masing pendekatan dalam situasi yang diberikan, sehingga dapat membantu dalam pengambilan keputusan untuk mencapai tujuan bisnis dan teknis yang lebih baik.

1.2. Rumusan Masalah

Berikut ini adalah rumusan masalah yang dapat digunakan untuk mengeksplorasi perbandingan antara *Microservices Orchestration* dan *Choreography Pattern* pada skenario pemesanan dan transaksi di *e-commerce*.

1. Bagaimana kompleksitas pada *Microservices Orchestration* dan *Choreography Pattern* dalam skenario pemesanan dan transaksi di *e-commerce*?
2. Bagaimana kinerja *Microservices Orchestration* dan *Choreography Pattern* dalam menangani seknario pemesanan dan transaksi di *e-commerce*.
3. Bagaimana pengaruh *Microservices Orchestration* dan *Choreography* terhadap waktu respons, *throughput*, dan tingkat kesalahan sistem pada skenario pemesanan dan transaksi di *e-commerce*?
4. Apa kelebihan dan kekurangan pada masing-masing pendekatan?

1.3. Batasan Masalah

Berikut ini adalah beberapa batasan masalah dalam konteks perbandingan kompleksitas dan kinerja *Microservices Orchestration* dan *Choreography Pattern* pada skenario pemesanan dan transaksi pada *e-commerce.*

1. Penelitian ini melakukan perbandingan kompleksitas dengan pendekatan *Business process Modeling and Notations* (BPMN) menggunakan metrik *Controll-Flow Complexity Metrics* (CFC) *Microservices Orchestration* dan *Choreography Pattern* pada skenario pemesanan dan transaksi pada *e-commerce*.
2. Penelitian ini tidak akan membahas aspek teknologi lainnya yang tidak terkait dengan arsitektur sistem, seperti penggunaan platform cloud, perbandingan teknologi *framework*, dan hal semacamnya.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk memberikan wawasan tentang bagaimana cara memilih dan menerapkan pendekatan yang paling sesuai untuk kebutuhan bisnis yang berbeda. Berikut ini adalah tujuan-tujuan khusus dari penelitian ini :

1. Menganalisis kompleksitas pada pola *Orchestration* dan *Choreography* dalam implementasi *Microservices* pada skenario pemesanan dan transaksi di *e-commerce* untuk memahami perbedaan struktur, interaksi, dan koordinasi antara kedua pendekatan tersebut.
2. Mengukur dan membandingkan kinerja pola *Microservices Orchestration* dan *Choreography* dalam menangani skenario pemesanan dan transaksi di *E-commerce* berdasarkan parameter waktu respons, througput, dan tingkat kesalahan sistem.
3. Menganalisis pengaruh pola *Orchestration* dan *Choreography* terhadap latensi, *throughput*, dan tingkat kesalahan sistem dalam konteks skenario pemesanan dan transaksi di *e-commerce* untuk memahami keunggulan dan kelemahan masing-masing pendekatan.

4. Mengidentifikasi kelebihan dan kekurangan pada masing-masing pendekatan (orkestrasi dan koreografi) dalam hal *skalabilitas*, *fleksibilitas*, dan *performa* system.

1.5. Manfaat Penelitian

Berikut ini adalah beberapa manfaat yang diperoleh pada penelitian ini terkait perbandingan kompleksitas dan kinerja *Microservices Orchestration* dan *Choreography* pada skenario pemesanan dan transaksi di *E-commerce* :

1. Memberikan pemahaman yang lebih baik tentang *Microservices Orchestration* dan *Choreography Pattern*, dan perbedaan antara kedua pendekatan. Hal ini dapat membantu pengembang dan arsitektur perangkat lunak untuk memilih pendekatan yang paling sesuai untuk kebutuhan bisnis.
2. Memberikan rekomendasi terkait cara memilih dan menerapkan pendekatan yang paling sesuai untuk kebutuhan bisnis yang berbeda. Hal ini dapat membantu pengembang dan arsitek perangkat lunak untuk menghindari masalah yang mungkin terjadi dalam penerapan pendekatan.
3. Memberikan kontribusi pada pengembangan dan peningkatan metode dan teknologi terkait dengan pengembangan layanan mikro, terutama pada pemilihan dan penerapan pendekatan yang paling sesuai untuk kebutuhan bisnis.