

BAB I

PENDAHULUAN

1.1 Latar Belakang

Arsitektur *microservice* mendapat perhatian dari industri karena kemampuan yang ditawarkannya dalam optimalisasi arsitektur sistem. Hal tersebut didukung fakta menurut *International Data Corporation* (IDC) pada akhir tahun 2021, 80% aplikasi *cloud-based* akan dikembangkan menggunakan arsitektur *microservice* (Waseem, Liang, and Shahin 2020). Gaya arsitektur *microservice* dapat menggantikan arsitektur *monolithic* karena fleksibilitas untuk beradaptasi dengan perubahan teknologi dan membatu penataan tim pengembang yang lebih baik (Balalaie, Heydarnoori, and Jamshidi Dermani 2016). Dalam arsitektur *microservice* layanan-layanan dapat berdiri secara independen, sehingga pengembangan yang dilakukan oleh tim pada suatu layanan tidak akan berpengaruh pada layanan yang lain. Banyak perusahaan besar telah mengembangkan aplikasinya ke arah arsitektur *microservice* seperti Amazon, Netflix, LinkedIn, Spotify, SoundCloud dan perusahaan lainnya (Di Francesco, Lago, and Malavolta 2019).

Arsitektur *microservice* dengan berbagai kelebihan yang ditawarkannya masih terdapat beberapa masalah dalam implementasinya diantaranya: komunikasi yang sering terjadi antara layanan dalam *microservice* melalui HTTP *synchronous* bisa menurunkan kinerja sistem (Waseem, Liang, and Shahin 2020). Komunikasi HTTP *synchronous* merupakan komunikasi yang biasa dilakukan *microservice*

dengan arsitektur berbasis API (*API-Driven Architecture*) (Kannedy 2018).

Beberapa solusi untuk menangani masalah diatas pada arsitektur *microservice* telah dicoba dalam penelitian sebelumnya diantaranya: penggunaan *container* yang menyediakan cara mudah untuk menskalakan operasi dengan membuat lebih banyak salinan layanan, dapat membantu menangani masalah skalabilitas (Trihinas and Pallis 2018), dan elastisitas (Di Francesco, Lago, and Malavolta 2019). Docker merupakan suatu teknologi representatif yang menerapkan teknik *containerization*, memiliki karakteristik yang ringan, dapat membantu dan menjalankan banyak *microservice* yang berkontribusi pada pemanfaatan sumber daya yang lebih tinggi, sehingga dapat meningkatkan kinerja *microservice* (Kang, Le, and Tao 2016). Selain dari itu, percobaan perbandingan analisis kinerja RESTful API dan RabbitMQ pada aplikasi web *microservice* menunjukkan bahwa, ketika sejumlah besar pengguna mengirim *request* ke aplikasi web pada saat yang sama RabbitMQ lebih stabil dibanding metode komunikasi REST API (Hong, Sik Yang, and Kim 2018). *Event-Driven* dapat diterapkan sebagai cara berkomunikasi antar *microservice*, ketika ada *volume* data besar yang perlu diproses dan ketika tidak ada tanggapan (*response*) yang diharapkan (Akbulut and Perros 2019).

Pada penelitian ini arsitektur komunikasi berbasis kejadian (*Event-Driven Architecture*) akan diimplementasikan untuk menangani komunikasi internal, sehingga dapat meningkatkan kinerja pada *microservice*.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, adapun rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Apakah dengan menerapkan arsitektur komunikasi berbasis kejadian pada antar layanan *microservice* dapat mengatasi penurunan kinerja sistem?
2. Seberapa besar peningkatan kinerja sistem pada antar layanan *microservice* apabila diterapkan arsitektur komunikasi berbasis kejadian?

1.3 Batasan Penelitian

Adapun batasan penelitian yang dilakukan adalah sebagai berikut:

1. Proses komunikasi *internal* dalam *microservice* yang dirancang dan dikembangkan menggunakan arsitektur komunikasi berbasis kejadian.
2. Pengukuran berfokus pada kinerja sistem.

1.4 Tujuan Penelitian

Berdasarkan latar belakang masalah dan rumusan masalah, maka tujuan dari penelitian ini adalah:

1. Mengetahui pengaruh penerapan arsitektur komunikasi berbasis kejadian pada kinerja sistem antar layanan *microservice*
2. Mengukur kinerja sistem (*response time, error rate, CPU usage*) terhadap hasil penerapan arsitektur komunikasi berbasis kejadian pada *microservice*.

1.5 Manfaat Penelitian

Manfaat penelitian yang dimaksud adalah sebagai berikut:

1. Dapat melakukan penerapan arsitektur komunikasi berbasis kejadian pada *microservice*.
2. Dapat memberikan solusi mengenai meningkatkan kinerja sistem pada *microservice* dengan menerapkan arsitektur komunikasi berbasis kejadian.
3. Dapat mengetahui hasil dari penerapan arsitektur komunikasi berbasis kejadian pada *microservice*.
4. Manfaat yang didapatkan dari penelitian ini adalah dapat mengimplementasikan ilmu yang didapat selama perkuliahan mengenai *Web Service* dalam bidang *service oriented architecture* (SOA) dan Sistem Terdistribusi.