

BAB I PENDAHULUAN

1.1 Latar Belakang

Seiring dengan pesatnya perkembangan teknologi informasi, aktivitas jual beli secara daring mengalami peningkatan yang sangat signifikan dalam beberapa tahun terakhir [1]. Masyarakat kini semakin terbiasa memanfaatkan platform digital untuk memenuhi berbagai kebutuhan, mulai dari produk rumah tangga hingga barang elektronik, karena kemudahan, kenyamanan, serta kecepatan proses transaksi yang ditawarkan. Kemajuan ini didorong pula oleh semakin meluasnya akses internet, meningkatnya penetrasi perangkat *mobile*, serta kemunculan berbagai platform *e-commerce* yang bersaing menyediakan layanan terbaik bagi konsumen. Fenomena ini telah mendorong transformasi besar dalam ekosistem perdagangan, di mana efisiensi dan kecepatan dalam pemenuhan pesanan menjadi faktor penentu kepuasan pelanggan.

Dalam konteks tersebut, industri logistik—khususnya layanan ekspedisi—berperan sebagai tulang punggung dari proses pemenuhan pesanan daring. Permintaan terhadap layanan pengiriman barang dalam jumlah besar, dengan cakupan wilayah yang luas dan waktu pengiriman yang semakin singkat, mengalami lonjakan tajam. Hal ini menuntut perusahaan ekspedisi untuk mampu menyediakan sistem operasional yang tidak hanya efisien dan andal, tetapi juga memiliki kemampuan adaptif terhadap dinamika kebutuhan pasar. Semakin kompleksnya alur bisnis pengiriman, seperti pengelolaan rute, pelacakan *real-time*, integrasi dengan sistem *e-commerce*, hingga manajemen gudang dan kurir, menuntut adanya sistem teknologi yang modular, dan mudah untuk dikembangkan secara berkelanjutan.

Namun, pendekatan arsitektur terpusat atau monolitik yang umum digunakan dalam tahap awal pengembangan sistem sering kali menghadapi berbagai tantangan ketika sistem mulai berkembang. Ketika kompleksitas sistem meningkat dan kebutuhan fitur bertambah, struktur monolitik dapat menjadi hambatan dalam hal fleksibilitas. Perubahan atau penambahan fungsi pada satu bagian sistem berisiko memengaruhi bagian lain yang tidak terkait secara langsung, sehingga menyulitkan proses pengembangan, pengujian, dan pemeliharaan. Selain

itu, kebutuhan untuk merespons perubahan bisnis secara cepat menjadi lebih sulit ketika seluruh komponen saling bergantung secara erat dan tidak terpisah secara modular.

Sebagai solusi atas tantangan tersebut, pendekatan *microservice* muncul sebagai strategi arsitektur yang menjanjikan dalam membangun sistem ekspedisi modern. Dengan membagi sistem ke dalam layanan-layanan kecil yang berdiri sendiri dan berfokus pada domain bisnis tertentu, *microservice* memberikan fleksibilitas tinggi dalam pengembangan dan pengelolaan sistem. Setiap layanan dapat dikembangkan, diuji, dan *dideploy* secara independen, memungkinkan tim pengembang untuk merespons kebutuhan bisnis dengan lebih cepat dan efisien. Selain itu, layanan-layanan ini dapat diskalakan secara horizontal sesuai dengan beban masing-masing, tanpa harus memengaruhi layanan lain.

Untuk memastikan pembagian layanan sesuai dengan struktur dan kebutuhan bisnis, diterapkan pendekatan *Domain-Driven Design (DDD)*. *DDD* membantu dalam memetakan kompleksitas domain logistik ke dalam *bounded context* yang jelas dan bermakna, sehingga setiap tim dapat fokus pada pengembangan layanan dalam ruang lingkup yang spesifik dan terdefinisi dengan baik. Dengan demikian, komunikasi antar tim dan pemahaman terhadap model bisnis menjadi lebih efektif dan konsisten.

Di samping itu, guna menjaga kualitas arsitektur sistem yang dibangun, digunakan pula prinsip *Clean Architecture*. Prinsip ini memisahkan tanggung jawab dalam sistem ke dalam beberapa lapisan (seperti *domain*, *use case*, dan *interface*), sehingga perubahan pada satu lapisan tidak berdampak langsung ke lapisan lain. Pemisahan ini juga mempermudah proses pengujian dan pemeliharaan kode dalam jangka panjang, serta mendukung keberlangsungan pengembangan sistem oleh tim yang berbeda.

Dalam praktiknya, komunikasi antar layanan *microservice* dilakukan secara *synchronous* menggunakan kombinasi *REST API* dan *gRPC*. *REST* digunakan untuk kebutuhan interoperabilitas dan kemudahan integrasi, sementara *gRPC* memberikan efisiensi komunikasi pada layanan yang membutuhkan performa tinggi. Untuk menghubungkan *client* dengan seluruh layanan *microservice* yang tersebar, digunakan *API Gateway* berbasis *GraphQL*. Gateway ini tidak hanya

berfungsi sebagai pintu masuk utama yang menangani proses autentikasi, tetapi juga memungkinkan pengambilan data secara efisien berdasarkan kebutuhan *query* dari *client*, sehingga mengurangi *over-fetching* dan *under-fetching* data.

Dengan kombinasi pendekatan *microservice*, *Domain-Driven Design*, dan *Clean Architecture*, sistem ekspedisi yang dibangun menjadi lebih modular, fleksibel, dan tahan terhadap perubahan. Arsitektur ini memungkinkan sistem untuk terus berkembang seiring kebutuhan bisnis tanpa mengorbankan stabilitas dan kualitas. Hal ini menjadikannya sebagai solusi strategis dalam menjawab tantangan pengembangan sistem logistik modern yang kompleks, dinamis, dan terus berkembang dari waktu ke waktu.

1.2 Perumusan masalah

Berdasarkan latar belakang yang telah dipaparkan, terdapat beberapa permasalahan utama yang menjadi fokus dalam penelitian ini, yaitu:

1. Bagaimana merancang dan mengimplementasikan sistem ekspedisi berbasis *microservice* secara modular menggunakan pendekatan *Clean Architecture* dan *Domain-Driven Design (DDD)*?
2. Bagaimana pendekatan *Domain-Driven Design (DDD)* dapat digunakan untuk mengidentifikasi batasan eksplisit dan memetakan domain logistik ke dalam struktur layanan *microservice*?
3. Bagaimana penerapan *Clean Architecture* membantu dalam memisahkan tanggung jawab antar lapisan sistem untuk mendukung *maintainability* sistem?
4. Bagaimana merancang pola komunikasi antar layanan *microservice* serta peran *API Gateway* berbasis *GraphQL* dalam menyederhanakan integrasi *client* terhadap layanan-layanan tersebut?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk mengeksplorasi dan mendemonstrasikan penerapan pendekatan arsitektural dalam membangun sistem ekspedisi berbasis *microservice*. Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menunjukkan bagaimana sistem ekspedisi dapat dirancang secara modular dengan memanfaatkan arsitektur *microservice* untuk mendukung

pengelolaan proses logistik yang kompleks dan dinamis.

2. Menerapkan pendekatan *Domain-Driven Design (DDD)* untuk memetakan domain sistem ekspedisi ke dalam *bounded context* yang selaras dengan struktur layanan *microservice*.
3. Menerapkan prinsip *Clean Architecture* di setiap layanan untuk memisahkan tanggung jawab antar lapisan sistem guna meningkatkan *maintainability*.
4. Menjelaskan rancangan komunikasi antar layanan *microservice* yang efisien dan interoperabel, serta mengeksplorasi penggunaan *API Gateway* berbasis *GraphQL* sebagai penghubung antara client dan layanan-layanan tersebut.

1.4 Batasan Masalah

Agar penelitian ini lebih terfokus dan terarah, maka ditetapkan beberapa batasan masalah sebagai berikut:

1. Penelitian ini berfokus pada studi implementasi sistem ekspedisi menggunakan pendekatan *Domain-Driven Design (DDD)* dan *Clean Architecture* dalam konteks arsitektur *microservice*, tanpa bertujuan untuk membandingkan keunggulan sistem ini dengan sistem logistik lain yang sudah ada.
2. Seluruh *service* dikembangkan menggunakan satu bahasa pemrograman, yaitu *Golang*, guna mempermudah proses pengembangan dan mendukung potensi *code sharing* antar layanan, meskipun *code sharing* antar *service* tidak menjadi fokus utama.
3. Implementasi arsitektur *microservice* difokuskan pada desain dan pengembangan layanan utama, tanpa mencakup otomatisasi deployment secara menyeluruh seperti penerapan *CI/CD pipeline*, *service mesh*, atau orkestrasi menggunakan *Kubernetes*.
4. *API Gateway* yang digunakan hanya menangani proses autentikasi, sedangkan fitur otorisasi belum diimplementasikan secara lengkap.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi terhadap

pengembangan sistem berbasis *microservice* dengan pendekatan *Domain-Driven Design* dan *Clean Architecture*, khususnya dalam konteks sistem ekspedisi. Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Memberikan contoh implementasi nyata penerapan *Clean Architecture* dan *Domain-Driven Design* dalam sistem ekspedisi berbasis *microservice*, yang dapat menjadi referensi bagi pengembang dan peneliti dalam membangun sistem serupa.
2. Meningkatkan pemahaman teknis mengenai pemisahan tanggung jawab antar lapisan sistem dan pemetaan domain ke dalam *bounded context*, sehingga dapat mendukung pengembangan perangkat lunak yang lebih modular dan *maintainable*.
3. Mendukung praktik arsitektur modern dalam pengembangan sistem logistik yang fleksibel, selaras dengan kebutuhan industri akan transformasi digital berbasis layanan terpisah (*service-oriented*).

