# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang Masalah

Perkembangan Teknologi yang sangat cepat di era digital saat ini tentu akan diperlukan infrastruktur dilayanan Teknologi Informasi. Dengan adanya pergeseran ke arah arsitektur mikroservis dan kontainerisasi, penggunaan kubernetes sebagai orkestrator kontainer telah berkembang pesat. Kubernetes memungkinkan pengelolaan dan penyebaran aplikasi yang kompleks dengan cepat dan efisien. Namun, pertumbuhan ini juga diikuti oleh meningkatnya kompleksitas log dari berbagai kontainer yang berjalan dalam cluster kubernetes [1]. Kubernetes merupakan sebuah platform open-source yang mampu melakukan management clustering container dalam jumlah besar atau bisa disebut sebagai cluster *orchestration* yang memiliki tugas melakukan penjadwalan, *scaling*, *recovery* dan monitoring container [2]. Klaster kubernetes memungkinkan organisasi untuk mudah mengelola, mendistribusikan, dan memperluas penyebaran aplikasi sesuai dengan kebutuhan. Dalam lingkungan klaster Kubernetes, *load balancing* memiliki peran penting dalam mendistribusikan lalu lintas pengguna secara merata dan mengoptimalkan sumber daya.

Dalam suatu ekosistem Kubernetes, setiap kontainer yang berjalan menghasilkan log yang signifikan, mencakup aktivitas aplikasi, performa sistem, dan pesan debugging. Log ini merupakan sumber informasi penting yang mencerminkan kesehatan sistem, status operasional aplikasi, serta deteksi dini terhadap potensi masalah atau anomali. Dengan pertumbuhan skala dan kompleksitas lingkungan Kubernetes, manajemen log menjadi semakin krusial, terutama untuk memahami kinerja aplikasi, mendeteksi masalah yang muncul, dan memenuhi kebutuhan keamanan seperti audit, pelacakan aktivitas, dan kepatuhan terhadap standar tertentu. Namun, dengan pertumbuhan infrastruktur yang semakin besar dan dinamis, tantangan dalam mengelola log menjadi semakin kompleks. Setiap kontainer, pod, dan node di dalam kluster Kubernetes menghasilkan volume log yang besar dan terus bertambah seiring dengan peningkatan jumlah layanan atau aplikasi yang dikelola. Data log ini tidak hanya berasal dari aplikasi, tetapi juga mencakup aktivitas sistem, seperti komunikasi antar-pod, aktivitas jaringan, serta interaksi dengan penyimpanan atau sumber daya lainnya. Selain itu, log tersebut tersebar di berbagai lokasi dalam kluster, yang membuat pengelolaannya menjadi sulit jika dilakukan secara manual. Kendala

lain yang muncul adalah bagaimana mengelola log secara efisien di tengah skala besar. Volume log yang besar memerlukan sistem penyimpanan yang andal serta mekanisme untuk memproses, mengindeks, dan menganalisis data secara real-time. Dalam situasi ini, pendekatan tradisional yang mengandalkan pengumpulan log manual atau pencatatan lokal tidak lagi memadai. Setiap perubahan kecil dalam aplikasi dapat menghasilkan ribuan baris log baru, sehingga dibutuhkan solusi terpusat untuk mengintegrasikan dan mengelola log secara efisien. Dengan pertumbuhan skala dan kompleksitas lingkungan Kubernetes, manajemen log menjadi semakin krusial untuk memahami kinerja aplikasi, mendeteksi masalah, dan memenuhi kebutuhan keamanan. Namun, dengan pertumbuhan infrastruktur yang semakin besar dan kompleks, mengelola log (catatan aktivitas sistem dan aplikasi) di dalam lingkungan kubernetes menjadi semakin menantang. Setiap kontainer, pod, dan node dalam cluster kubernetes menghasilkan volume log yang besar, yang mencakup informasi penting tentang kinerja aplikasi, aktivitas pengguna, dan masalah sistem yang perlu diatasi [3].

Pengelolaan log di lingkungan kubernetes memerlukan solusi yang mampu mengumpulkan, menyimpan, dan menganalisis log dengan efisien dan efektif. Salah satu pendekatan yang telah banyak digunakan dalam mengatasi tantangan ini adalah dengan menggabungkan alat-alat khusus seperti Fluentd dan Elasticsearch [4]. Fluentd adalah agen pengumpul log open-source yang dirancang untuk beroperasi dalam lingkungan terdistribusi. Dengan kemampuannya untuk mengumpulkan log dari berbagai sumber yang tersebar di dalam cluster Kubernetes, Fluentd menjadi salah satu komponen kunci dalam infrastruktur manajemen log terdistribusi. Sementara itu, Elasticsearch adalah mesin pencarian dan analisis log yang kuat. Dengan kemampuannya untuk menyimpan, mengindeks, dan memungkinkan pencarian cepat terhadap volume data log yang besar, Elasticsearch menjadi solusi yang ideal untuk menyediakan wawasan yang mendalam tentang kinerja aplikasi dan infrastruktur di dalam lingkungan kubernetes [5]. Dalam konteks inilah penelitian tentang "Manajemen Log Terdistribusi di Cluster Kubernetes Menggunakan Fluentd dan Elasticsearch" menjadi relevan. Penelitian ini bertujuan untuk menginvestigasi dan mengembangkan solusi yang efektif untuk pengelolaan log di lingkungan kubernetes, dengan memanfaatkan potensi dari Fluentd dan Elasticsearch.

Pada penelitian sebelumnya [6], telah dibuat monitoring pada kluster kubernetes menggunakan Prometheus dan Grafana, dimana Prometheus digunakan sebagai agen

untuk mengambil data dari kluster Kubernetes. Prometheus digunakan sebagai agen untuk mengambil data dari kluster Kubernetes dan menyimpan metrik tersebut ke dalam basis data time-series yang mendukung visualisasi oleh Grafana. Kombinasi Prometheus dan Grafana menunjukkan hasil yang cukup efektif dalam memonitor performa aplikasi di lingkungan Kubernetes, terutama untuk memantau metrik seperti penggunaan CPU, memori, dan status pod. Namun, solusi ini terbatas pada monitoring metrik dan tidak secara khusus berfokus pada pengelolaan log aplikasi atau sistem, yang merupakan aspek penting untuk mendeteksi anomali, debugging, dan audit keamanan.

Pada penelitian [7] monitoring log menggunakan logtash sebagai engine pengumpul data dimana pada logtash ini memiliki kekurangan dari sisi support plugin yang tidak terlalu luas dan penggunaan resources pada server yang tinggi sehingga akan membebani server. Logstash mampu memproses data log dengan pipeline yang dapat dikustomisasi, tetapi memiliki beberapa kekurangan signifikan. Salah satunya adalah dukungan plugin yang tidak terlalu luas, sehingga membatasi fleksibilitas dalam mengintegrasikan berbagai jenis sumber log. Selain itu, penggunaan sumber daya server oleh Logstash relatif tinggi, yang dapat menyebabkan overhead besar pada sistem, terutama di lingkungan dengan jumlah log yang masif. Hal ini dapat menghambat performa sistem utama, membuatnya kurang efisien untuk digunakan di lingkungan cloud yang membutuhkan efisiensi tinggi. Perbedaan penelitian ini dengan penelitian sebelumnya adalah pada penelitian ini agen untuk mengumpulkan log atau data dari server adalah fluentd dimana dengan fluentd yang mendukung plugin lebih luas dan resources server yang digunakan tidak terlalu tinggi. Dalam penelitian ini dilakukan dengan menguji implementasi Fluentd sebagai agen pengumpul log dan Elasticsearch sebagai sistem penyimpanan dan analisis log. Pengujian dilakukan dengan membandingkan pencarian log secara manual menggunakan perintah Kubernetes (kubectl logs) dan pencarian log menggunakan Fluentd yang diteruskan ke Elasticsearch. Pengukuran dilakukan untuk mengevaluasi latensi pengiriman log, akurasi waktu pengumpulan log, kelengkapan log, penggunaan sumber daya. Diharapkan penelitian ini dapat memberikan kontribusi signifikan dalam meningkatkan efisiensi, ketersediaan, dan keamanan infrastruktur cloud dalam menghadapi tuntutan yang semakin kompleks dan dinamis.

## 1.2 Rumusan Masalah

Adapun rumusan masalah dalam skripsi ini sebagai berikut

1. Bagaimana mengimplementasikan manajemen log terdistribusi di dalam cluster Kubernetes menggunakan Fluentd dan Elasticsearch ?
2. Bagaimana efektivitas solusi yang diusulkan dalam mengumpulkan, menyimpan, dan menganalisis log di lingkungan Kubernetes?
3. Bagaimana dampak penggunaan manajemen log terdistribusi pada Kubernetes?

## 1.3 Batasan Masalah

Adapun batasan masalah dalam buku skripsi ini adalah berikut

1. Penelitian ini akan membatasi lingkup pada penggunaan Fluentd sebagai agen pengumpul log dan Elasticsearch sebagai mesin penyimpanan dan pencarian log di dalam cluster Kubernetes.
2. Fokus utama penelitian ini adalah pada implementasi dan evaluasi solusi manajemen log menggunakan Fluentd dan Elasticsearch, tanpa memperhitungkan aspek keamanan secara mendalam.
3. Parameter analisis yang diuji adalah konsistensi log yaitu mengukur kesesuaian log yang dikumpulkan manual dan log yang dikumpulkan oleh Fluentd dalam jumlah pod 20 *replica* dan parameter yang diukur adalah latensi pengiriman log.

## 1.4 Tujuan

Berdasarkan dari latar belakang maka skripsi ini memiliki tujuan sebagai berikut

1. Mengimplementasikan sistem manajemen log terdistribusi di dalam cluster Kubernetes menggunakan Fluentd dan Elasticsearch.
2. Mengevaluasi efektivitas solusi yang diusulkan dalam mengatasi masalah pengumpulan dan analisis log di lingkungan Kubernetes.
3. Menganalisis dan mengidentifikasi tantangan yang dihadapi dalam pengelolaan log di lingkungan Kubernetes menggunakan Fluentd dan Elasticsearh.

## 1.5 Manfaat

Manfaat yang diharapkan dari manajemen log di Kluster Kubernetes menggunakan Fluentd dan Elasticsearch

1. Memberikan kontribusi pada pengembagan infrastruktur Teknologi Informasi serta dapat memberikan kontribusi yang signifikan dalam pengelolaan log terdistribusi di cluster Kubernetes.
2. Membantu para praktisi Kubernetes dalam mengelola log terdistribusi secara efektif dan efisien.

## 1.6 Sistematika Penulisan

Dalam penulisan skripsi terdiri dari lima bab utama, dengan keterangan sebagai berikut

**BAB I PENDAHULUAN**

Pada bab ini akan dibahas mengenai latar belakang, rumusan masalah, tujuan skripsi, manfaat, batasan masalah dalam skripsi, metodologi penelitian, serta sistematika penulisan dalam buku skripsi ini.

**BAB II DASAR TEORI**

Pada bab berikut akan dibahas mengenai kajian pustaka dan teori-teori pendukung yang digunakan dalam pengerjaan skripsi, seperti penjelasan mengenai manajemen log terdistribusi pada kluster kubernetes menggunakan Fluentd.

**BAB III METODE PENELITIAN**

Pada bab tiga ini akan membahas tentang deskripsi skripsi, serta alur pengerjaan skripsi.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab empat berikut ini akan membahas tentang hasil perancangan system log terdistribusi pada kluster Kubernetes menggunakan fluentd.

**BAB V PENUTUP**

Pada bab penutup ini akan dilakukan pembahasan mengenai kesimpulan yang didapatkan dari pengerjaan skripsi dan saran bagi pembaca yang akan melakukan pengujian menggunakan topik yang sama.