BAB I. PENDAHULUAN

1.1 Latar Belakang

Dalam dekade terakhir, Kubernetes telah menjadi salah satu platform paling populer untuk orkestrasi container, memungkinkan pengelolaan aplikasi terdistribusi dengan skala besar. Sebelum adanya layanan cloud, Kubernetes diciptakan oleh Google pada tahun 2014 sebagai proyek open-source yang bertujuan untuk menyederhanakan penyebaran, skala, dan operasi aplikasi container (Johnston, S. J., dkk, 2020). Kubernetes berhasil menarik perhatian komunitas pengembang karena kemampuannya yang handal dalam mengelola berbagai container di berbagai lingkungan. Namun, pada awal kemunculannya, implementasi Kubernetes secara mandiri di server fisik atau virtual sering kali memerlukan sumber daya dan pengetahuan yang cukup besar dalam hal infrastruktur dan pemeliharaan.

Perkembangan teknologi cloud telah mengubah lanskap ini secara signifikan. Layanan cloud yang dikelola seperti *Azure Kubernetes Service* (AKS) dari Microsoft, *Amazon Elastic Kubernetes Service* (EKS), dan *Google Kubernetes Engine* (GKE) menyediakan fitur-fitur tambahan seperti auto-scaling, monitoring, dan integrasi yang mendalam dengan layanan cloud lainnya. Azure Kubernetes Service, khususnya, telah menjadi pilihan populer bagi banyak perusahaan karena kemudahan penggunaannya dan integrasinya yang kuat dengan ekosistem Azure.

Penelitian ini akan fokus pada komparasi performa ingress controller dalam cluster Kubernetes yang dikelola menggunakan AKS. Ingress controller adalah komponen penting dalam arsitektur Kubernetes yang bertanggung jawab untuk mengelola akses HTTP dan HTTPS ke layanan dalam cluster (Google, 2023). Ada berbagai jenis ingress controller yang tersedia, masing-masing dengan karakteristik dan performa yang berbeda-beda.

Ingress controller merupakan komponen krusial dalam arsitektur Kubernetes karena mengelola lalu lintas jaringan eksternal yang masuk ke dalam cluster, penerapan ingress controller dalam konteks layanan cloud sering kali lebih baik

dibandingkan dengan lingkungan on-premise. Salah satu alasan utamanya adalah kemudahan dalam pengimplementasian load balancer yang secara native didukung oleh penyedia layanan cloud. Di lingkungan server fisik, pengaturan load balancer memerlukan konfigurasi manual yang rumit, serta perangkat keras khusus yang bisa mahal dan memerlukan pemeliharaan intensif. Sebaliknya, di cloud, load balancer dapat diterapkan secara otomatis dengan beberapa klik atau baris perintah sederhana, berkat layanan terkelola yang disediakan. Selain itu, cloud load balancer dapat dengan mudah diskalakan untuk menangani lonjakan beban lalu lintas, menyediakan redundansi, dan meningkatkan ketersediaan aplikasi tanpa intervensi manual yang signifikan (Microsoft, 2023). Keunggulan ini membuat ingress controller di lingkungan cloud lebih mudah diimplementasikan dan dikelola.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan, maka rumusan masalah adalah:

1. Bagaimana performa Nginx Controller dalam mengatasi *latency*, *throughput,* serta *Delay/Packet Loss*?
2. Bagaimana kelebihan dan kekurangan dalam mengimplementasikan Nginx Controller?
3. Bagaimana Nginx Controller dapat mengelola dan mengurangi risiko *Single Point of Failure* (SPOF) dalam pengembangan perangkat lunak atau sistem?

1.3 Tujuan

Sesuai rumusan masalah yang telah dipaparkan, penelitian ini dibuat dengan tujuan untuk;

1. Mengukur performa yang meliputi, *latency*, *throughput,* serta *delay/packet loss* untuk memahami sejauh Nginx controller dapat mengatasi beban *traffic* yang tinggi.
2. Mengevaluasi performa nginx ingress dalam menangani *traffic* dengan menggunakan *virtual client*, yang terdiri dari 10, 50, 100, 200, dan 500 user.
3. Menjadi solusi untuk menghindari terjadinya Titik Kegagalan Tunggal (*Single Point of Failure).*