

# **BAB 1. PENDAHULUAN**

## **1.1 Latar Belakang**

Di era digital saat ini, aplikasi berbasis cloud semakin banyak digunakan karena menawarkan kemudahan, fleksibilitas, dan skalabilitas yang tinggi. Cloud computing memungkinkan perusahaan untuk mengakses sumber daya komputasi secara dinamis dan sesuai kebutuhan tanpa perlu investasi besar dalam infrastruktur fisik (Peter Mell & Grance, 2011). Namun, seiring dengan perkembangan teknologi, tantangan dalam pengelolaan aplikasi juga semakin besar. Salah satu tantangan utama adalah bagaimana cara mengembangkan dan memperbarui aplikasi dengan cepat tanpa mengganggu pengguna.

Continuous Integration dan Continuous Deployment (CI/CD) merupakan pendekatan modern dalam pengembangan perangkat lunak yang memungkinkan pengembang untuk mengotomatisasi proses build, testing, dan deployment aplikasi secara efisien. Dengan CI/CD, perubahan kode dapat diuji dan dirilis secara cepat dan berulang, sehingga mempercepat siklus pengembangan serta meningkatkan kualitas dan reliabilitas perangkat lunak (Fowler, 2018). Dalam konteks infrastruktur berbasis cloud, Kubernetes menjadi platform orkestrasi container yang sangat populer karena kemampuannya dalam mengelola container secara otomatis, meningkatkan skalabilitas, serta memastikan ketersediaan dan kontinuitas layanan aplikasi.

Dengan mengintegrasikan CI/CD pipeline menggunakan Jenkins, proses deployment aplikasi ke dalam lingkungan Kubernetes dapat dilakukan secara otomatis dan konsisten. Jenkins, sebagai salah satu tools open-source yang paling banyak digunakan dalam CI/CD, menyediakan berbagai plugin dan integrasi yang memudahkan orkestrasi pipeline secara fleksibel dan dapat disesuaikan dengan kebutuhan proyek. Hal ini dapat mengurangi risiko kesalahan manusia (human error) dalam proses deployment dan mempercepat waktu rilis aplikasi.

Meskipun banyak perusahaan telah mengadopsi teknologi ini, masih terdapat tantangan dalam implementasi yang optimal, seperti kompleksitas konfigurasi pipeline, jaminan keamanan dalam proses deployment, serta efisiensi dalam penggunaan sumber daya cloud. Oleh karena itu, kegiatan ini bertujuan untuk mengimplementasikan CI/CD pipeline menggunakan Jenkins dan Kubernetes serta menganalisis dampaknya terhadap kinerja aplikasi berbasis cloud. Penelitian ini diharapkan dapat memberikan solusi yang efektif dalam meningkatkan efisiensi pengembangan perangkat lunak dan memastikan aplikasi berjalan dengan performa yang optimal.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, dapat dirumuskan beberapa masalah yang menjadi fokus pada latar belakang diatas:

1. Bagaimana mengimplementasikan pipeline CI/CD Pipeline menggunakan Jenkins dan Kubernetes dalam aplikasi berbasis cloud?
2. Bagaimana Alur CI/CD Pipeline menggunakan Jenkins dan Kubernetes pada aplikasi berbasis cloud?
3. Bagaimana performa CI/Cd Pipeline menggunakan Jenkins dan Kubernetes pada Aplikasi Berbasis Cloud?

## **1.3 Tujuan**

Kegiatan ini bertujuan untuk menerapkan CI/CD pipeline menggunakan Jenkins dan Kubernetes dalam pengembangan aplikasi berbasis cloud. Secara lebih spesifik, tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan pipeline CI/CD menggunakan Jenkins dan Kubernetes untuk mengotomatisasi proses build, dan deployment aplikasi berbasis cloud.
2. Membangun alur kerja otomatisasi dari komit kode (GitHub) hingga deployment ke Kubernetes cluster menggunakan Jenkins
3. Menganalisis dan mendesain Sistem CI/CD yang Efisien untuk mengurangi terjadinya kesalahan pada manusia.

#### **1.4 Manfaat**

Adapun manfaat yang diharapkan dari pembuatan laporan ini adalah sebagai berikut :

1. Membantu pengembang dalam mengotomatisasi proses build, testing, dan deployment menggunakan Jenkins dan Kubernetes.
2. Memudahkan tim pengembang dalam mengelola dan meningkatkan kinerja aplikasi berbasis cloud.
3. Mengurangi risiko kesalahan manual dalam deployment serta mempercepat waktu rilis aplikasi.