

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi merupakan bentuk perkembangan dan penerapan dari berbagai peralatan maupun sistem yang digunakan untuk menyelesaikan permasalahan dalam kehidupan sehari-hari. Seiring dengan kemajuan zaman, manusia terus menciptakan berbagai bentuk teknologi baru yang digunakan sebagai solusi atas berbagai persoalan (Efendi dkk., 2023). Salah satu bidang yang mengalami perkembangan pesat adalah teknologi informasi, yang telah membawa perubahan signifikan dalam berbagai aspek kehidupan. Di antara inovasi penting dalam bidang ini adalah penerapan Sistem Informasi Akademik (SIAK), yaitu sistem yang berfungsi untuk mendukung manajemen data akademik mahasiswa serta memfasilitasi proses pembelajaran dan administrasi kampus. Sistem ini tidak hanya membantu dalam pengelolaan data, tetapi juga mendorong inovasi digital yang dapat meningkatkan kreativitas dosen dan motivasi belajar mahasiswa (Febianti dkk., 2023).

Menurut (Saputri dkk., 2022), sistem informasi akademik merupakan salah satu sarana utama yang harus dimiliki oleh setiap lembaga pendidikan, baik sekolah maupun perguruan tinggi, yang dikembangkan sesuai dengan kebutuhan masing-masing institusi. Sistem ini dirancang untuk mengelola data akademik secara efisien mulai dari proses pendaftaran mahasiswa, pengelolaan nilai, jadwal kuliah, hingga manajemen data dosen dan mahasiswa. Namun, dalam implementasinya, Sistem Informasi Akademik Universitas Pendidikan Ganesha (SIAK Undiksha) menghadapi berbagai tantangan, terutama terkait kinerja dan ketersediaan layanan. Berdasarkan hasil wawancara dengan tim UPA TIK Undiksha, diketahui bahwa terdapat sekitar 51 server aktif yang menangani berbagai layanan digital universitas, termasuk sistem SIAK. Permasalahan utama yang sering muncul adalah penurunan performa dan downtime ketika terjadi lonjakan akses

pengguna secara bersamaan, terutama pada masa pengisian KRS atau penginputan nilai. Tingginya beban akses terhadap sistem SIAK sejalan dengan jumlah sivitas akademika yang dilayani. Berdasarkan data resmi PDDIKTI per tahun 2025, Universitas Pendidikan Ganesha memiliki 24.083 mahasiswa aktif dan 676 dosen, yang seluruh aktivitas akademiknya bergantung pada layanan Sistem Informasi Akademik. Data tersebut menunjukkan bahwa SIAK merupakan layanan kritis dengan tingkat akses yang sangat tinggi, sehingga menuntut ketersediaan dan keandalan sistem yang optimal. Masalah tersebut disebabkan oleh arsitektur sistem SIAK yang saat ini masih menggunakan pendekatan *vertical scaling*, yaitu peningkatan kapasitas dengan menambah sumber daya seperti CPU, RAM, atau *storage* pada satu server fisik. Pendekatan ini memang dapat meningkatkan performa dalam jangka pendek, namun memiliki keterbatasan karena tidak mampu menjamin ketersediaan layanan (*Availability*). Ketika satu server mengalami gangguan, seluruh layanan akan terhenti karena tidak ada node lain yang dapat mengambil alih proses. Hal ini menyebabkan sistem menjadi tidak resilien terhadap kegagalan perangkat keras maupun lonjakan beban permintaan pengguna. UPA TIK Undiksha telah melakukan beberapa upaya untuk mengatasi permasalahan tersebut, seperti meningkatkan kapasitas perangkat keras (*vertical scaling*), menggunakan *firewall* untuk memfilter permintaan berlebih, menambahkan CAPTCHA sebagai validasi pengguna, serta mencoba layanan Cloudflare untuk perlindungan trafik. Namun, solusi-solusi tersebut masih berfokus pada satu server utama tanpa adanya mekanisme replikasi dan distribusi beban yang terstruktur. Akibatnya, ketika terjadi lonjakan permintaan pengguna (*overload request*), server tunggal tersebut tetap menjadi titik kegagalan utama (*single point of failure*).

Permasalahan tersebut menunjukkan bahwa arsitektur *vertical scaling* yang digunakan saat ini tidak cukup untuk mendukung sistem akademik dengan tingkat aktivitas tinggi. Untuk meningkatkan

ketersediaan layanan, diperlukan pendekatan *horizontal scaling*, yaitu penambahan beberapa node server yang bekerja secara paralel. Dengan model ini, beban kerja dapat didistribusikan ke beberapa node sehingga apabila salah satu node mengalami gangguan, node lainnya tetap dapat melanjutkan layanan tanpa menyebabkan *downtime*. Penerapan *horizontal scaling* memerlukan sistem *clustering* yang mampu melakukan replikasi layanan, pembagian beban (*load balancing*), serta pemulihan otomatis (*failover*) antar node. Salah satu teknologi yang dapat memenuhi kebutuhan tersebut adalah Kubernetes.

Kubernetes merupakan platform *open-source* untuk orkestrasi kontainer yang dirancang untuk mengelola pengelompokan aplikasi dalam bentuk *pod* secara otomatis (Poniszewska-Marańda & Czechowska, 2021). Kubernetes memiliki kemampuan dalam penjadwalan, penskalaan, pemantauan, serta *self-healing*, yang memungkinkan sistem tetap beroperasi meskipun terjadi kegagalan pada salah satu node. Jika dibandingkan dengan platform lain seperti Docker Swarm atau Proxmox, Kubernetes memiliki keunggulan dalam hal skalabilitas, otomatisasi, serta dukungan komunitas yang luas. Docker Swarm memang lebih sederhana dalam hal konfigurasi, tetapi memiliki keterbatasan dalam mekanisme *failover*, sedangkan Proxmox lebih berfokus pada virtualisasi berbasis *virtual machine* (Sturley dkk., 2024). Dengan memanfaatkan Kubernetes, Sistem Informasi Akademik Undiksha dapat diimplementasikan dalam lingkungan *cluster container* yang mendukung mekanisme redundansi pada level aplikasi dan distribusi beban layanan secara terintegrasi melalui replikasi *pod* dan konfigurasi *load balancing*. Melalui mekanisme replikasi *pod* dan konfigurasi *load balancing* menggunakan NGINX, sistem dapat menangani lonjakan beban pengguna secara efisien (Kristell dkk., 2025).

Meskipun Kubernetes pada umumnya diterapkan pada infrastruktur berspesifikasi tinggi, dalam praktiknya banyak institusi pendidikan dan organisasi skala menengah masih memiliki

keterbatasan sumber daya perangkat keras. Oleh karena itu diperlukan kajian yang menganalisis efektivitas penerapan Kubernetes pada lingkungan dengan spesifikasi hardware terbatas atau masih dalam kriteria minimum spesifikasi dalam menjalankan fungsi Kubernetes secara fungsional. Penelitian ini memanfaatkan test bed berbasis perangkat dengan spesifikasi rendah sebagai representasi lingkungan non-production untuk mengevaluasi apakah konsep evaluasi redundansi level aplikasi dan load balancing Kubernetes tetap dapat berjalan secara fungsional dan stabil dalam kondisi keterbatasan sumber daya.

Oleh karena itu, penelitian ini berjudul "Implementasi Kubernetes Cluster untuk Evaluasi Redundansi Level Aplikasi dan Load Balancing pada Sistem Informasi Akademik Universitas Pendidikan Ganesha". Kontribusi utama penelitian ini terletak pada penyajian studi kasus empiris penerapan Kubernetes Cluster pada Sistem Informasi Akademik perguruan tinggi di Indonesia, dengan analisis perbandingan langsung terhadap arsitektur single-server berbasis Docker tanpa orkestrasi. Penelitian ini bersifat Proof of Concept (PoC) untuk memvalidasi manfaat Kubernetes dalam meningkatkan ketersediaan layanan dan kemampuan skalabilitas sistem. Hasil penelitian diharapkan dapat menjadi dasar pertimbangan bagi UPA TIK dalam merencanakan migrasi arsitektur yang lebih komprehensif di lingkungan production.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, masalah utama yang dihadapi adalah Arsitektur Sistem Informasi Akademik Undiksha yang bersifat monolitik dan single-server menyebabkan kerentanan terhadap downtime dan ketidakmampuan dalam menangani lonjakan beban pengguna, khususnya pada periode akademik kritis.

Berdasarkan uraian tersebut dapat dirumuskan beberapa permasalahan yang dihadapi sebagai berikut:

1. Bagaimana efektivitas implementasi Kubernetes Cluster dalam mempertahankan redundansi level aplikasi Sistem Informasi Akademik Undiksha pada skenario kegagalan node di lingkungan dengan spesifikasi perangkat keras terbatas?
2. Bagaimana Kubernetes Cluster mampu meningkatkan kapasitas penanganan beban (load balancing) Sistem Informasi Akademik Undiksha dibandingkan arsitektur monolitik existing?
3. Konfigurasi Kubernetes Cluster seperti apa yang paling sesuai dan optimal untuk lingkungan test bed dengan spesifikasi perangkat keras terbatas dalam mendukung layanan Sistem Informasi Akademik Undiksha?

1.3 Hipotesis Penelitian

Berdasarkan permasalahan dan kebutuhan tersebut, hipotesis penelitian ini adalah: "Jika Sistem Informasi Akademik Universitas Pendidikan Ganesha diimplementasikan menggunakan Kubernetes cluster pada lingkungan dengan spesifikasi perangkat keras terbatas, maka sistem diperkirakan akan memiliki redundansi level aplikasi yang lebih baik serta kemampuan penanganan beban yang lebih stabil dibandingkan arsitektur single-server berbasis *vertical scaling*"

1.4 Tujuan Penelitian

Adapun tujuan penelitian yang ingin dicapai dari pelaksanaan pengembangan penelitian ini adalah:

1. Merancang dan mengimplementasikan arsitektur Kubernetes Cluster sebagai solusi redundansi level aplikasi dan *load balancing* pada layanan Sistem Informasi Akademik Undiksha di lingkungan dengan spesifikasi perangkat keras terbatas.
2. Menganalisis perilaku dan kemampuan sistem dalam mempertahankan redundansi level aplikasi serta mekanisme *load balancing* pada implementasi Kubernetes Cluster di lingkungan dengan spesifikasi perangkat keras terbatas.

1.5 Batasan Penelitian

Agar pembahasan dari penelitian ini tetap fokus dan lebih terarah, Adapun batasan-batasan masalah yang berikan penulis sebagai berikut:

1. Penelitian ini berfokus pada peningkatan ketersediaan layanan pada level aplikasi (pod). Arsitektur yang dibangun menggunakan satu master node, sehingga aspek redundansi level aplikasi pada level control plane (master node) tidak menjadi cakupan penelitian ini. Pengujian redundansi level aplikasi dilakukan dengan mensimulasikan kegagalan pada worker node untuk mengukur efektivitas mekanisme self-healing Kubernetes dalam mempertahankan kontinuitas layanan aplikasi.
2. Penelitian ini bersifat Proof of Concept dengan spesifikasi hardware terbatas (i7-3770, 6GB RAM) yang tidak merepresentasikan environment production UPA TIK. Fokus penelitian adalah validasi konsep redundansi level aplikasi dan scalability, bukan pengukuran performa absolut.
3. Keterbatasan sumber daya menyebabkan load balancer ditempatkan di master node, yang tidak direkomendasikan dalam environment production.
4. Pengujian dilakukan pada aplikasi web representatif yang menyerupai karakteristik dasar layanan SIAK, terutama pada aspek akses halaman (hit page).
5. Case logika pengujian hanya sampai pada hit page pada endpoint webserver sesuai dengan studi kasus yang terjadi.
6. Monitoring server atau node yang terhubung ke cluster Kubernetes dilakukan hanya menggunakan Prometheus dan Grafana.
7. Pengujian hanya menggunakan synthetic load (JMeter) tanpa simulasi pola traffic nyata.
8. Aspek keamanan Kubernetes belum dievaluasi secara komprehensif.

9. Penelitian ini hanya mengevaluasi redundansi layanan pada level aplikasi (pod). Aspek high availability pada lapisan basis data dan control plane Kubernetes tidak menjadi cakupan penelitian ini.

1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat secara teoritis maupun praktis sebagai berikut:

1. Manfaat Teoritis

Secara teoritis, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan kajian mengenai penerapan arsitektur Kubernetes Cluster pada lingkungan dengan keterbatasan sumber daya perangkat keras, khususnya dalam konteks redundansi level aplikasi dan load balancing layanan sistem informasi.

2. Manfaat Praktis

a. Bagi Teknisi IT

Dapat menjadi referensi alternatif bagi teknisi IT dalam merancang solusi redundansi level aplikasi dan load balancing berbasis Kubernetes pada lingkungan dengan keterbatasan sumber daya infrastruktur.

b. Bagi UPA TIK Undiksha

Dapat membantu dalam meningkatkan kualitas dari layanan di UPA TIK Undiksha dengan adanya penerapan *kubernetes cluster*. Mengetahui konfigurasi *kubernetes cluster* yang optimal untuk sistem Sistem Informasi Akademik Universitas Pendidikan Ganesha.

c. Bagi Peneliti

Dapat mengimplementasikan teori-teori yang telah dipelajari di bangku perkuliahan dalam bentuk kasus di dunia nyata melalui penerapan *kubernetes cluster*.