BAB 1
USULAN GAGASAN

1.1 Deskripsi Umum Masalah

Infrastruktur jaringan telekomunikasi merupakan fondasi utama bagi hampir seluruh aktivitas operasional bisnis dan layanan digital di era modern. Seiring dengan pertumbuhan kebutuhan teknologi informasi dan komunikasi, lingkungan jaringan dalam organisasi skala menengah hingga besar cenderung berkembang menjadi ekosistem yang heterogen, terdiri atas berbagai perangkat jaringan seperti router, switch, dan firewall yang berasal dari beragam vendor, antara lain Cisco, MikroTik, dan Fortinet. Perbedaan sistem operasi, arsitektur perangkat lunak, serta sintaks perintah Command Line Interface (CLI) pada masing-masing perangkat menciptakan tantangan yang kompleks dalam aspek manajemen, konfigurasi, dan pemantauan jaringan [1].

Pada praktiknya, pengelolaan jaringan multivendor umumnya masih dilakukan secara manual oleh administrator jaringan. Aktivitas ini melibatkan proses koneksi ke setiap perangkat secara individual, umumnya melalui protokol SSH atau Telnet, diikuti dengan eksekusi perintah konfigurasi dan pemeliharaan secara berulang. Pendekatan konvensional semacam ini tidak hanya memerlukan waktu yang besar, tetapi juga sangat rentan terhadap kesalahan manusia (human error) yang dapat berdampak serius terhadap layanan dan keamanan [2]. Selain itu, metode ini tidak mampu mengakomodasi kebutuhan skalabilitas dan efisiensi operasional yang dituntut oleh sistem jaringan modern.

Ketiadaan sebuah platform terpusat yang mampu mengotomatisasi proses provisioning serta melakukan pemantauan terhadap perangkat jaringan multivendor secara simultan dan real-time menjadi hambatan utama. Hal ini menghalangi terwujudnya manajemen jaringan yang efisien, konsisten, dan andal. Permasalahan ini menjadi urgensi yang perlu ditangani melalui solusi terintegrasi yang adaptif terhadap kompleksitas jaringan masa kini [3].

1.2 Analisis Masalah

Kompleksitas manajemen jaringan multivendor menjadi tantangan utama dalam operasional infrastruktur modern. Keberagaman sistem operasi dan sintaks perintah menyebabkan tidak adanya standar tunggal dalam proses konfigurasi, sehingga menyulitkan administrator menjaga konsistensi. Hal ini berpotensi menimbulkan konfigurasi yang tidak selaras, kesalahan yang berdampak pada gangguan layanan, serta kesulitan dalam proses audit dan dokumentasi [2].

Pendekatan manajemen manual yang masih umum digunakan sangat repetitif, tidak efisien, dan sulit untuk diskalakan, terutama pada jaringan dengan jumlah perangkat yang besar. Tingginya ketergantungan pada kemampuan individu administrator juga meningkatkan risiko terjadinya kesalahan manusia, yang dapat memengaruhi stabilitas dan keamanan jaringan.

Di sisi monitoring, penggunaan perangkat lunak yang terpisah dan tidak terintegrasi sering kali menyebabkan pemantauan yang tidak menyeluruh. Administrator kesulitan memperoleh gambaran utuh mengenai performa jaringan secara real-time, seperti trafik, utilisasi CPU, dan status konektivitas. Ketiadaan sistem notifikasi terpusat juga memperlambat respons terhadap insiden jaringan, sebuah tantangan yang coba diatasi oleh platform observabilitas modern [4].

Berdasarkan permasalahan tersebut, dibutuhkan solusi berupa sistem terintegrasi yang mampu mengotomatisasi proses provisioning dan monitoring perangkat dari berbagai vendor dalam satu antarmuka terpusat. Sistem ini harus menstandarkan proses konfigurasi, menyediakan pemantauan performa secara real-time, serta memungkinkan skalabilitas guna meningkatkan efisiensi dan keandalan manajemen infrastruktur.

1.2.1 Aspek Ekonomi

Dari sisi ekonomi, manajemen jaringan manual memberikan dampak biaya yang signifikan. Pertama, biaya operasional (OPEX) meningkat karena kebutuhan jam kerja administrator yang lebih tinggi. Kedua, biaya yang timbul akibat gangguan jaringan (*downtime*) yang disebabkan oleh kesalahan konfigurasi manual bisa sangat besar, mencakup kerugian pendapatan dan penurunan produktivitas [5]. Ketiga, terdapat biaya tersembunyi untuk pelatihan tenaga ahli yang harus menguasai berbagai platform vendor yang berbeda.

1.2.2 Aspek Manufakturabilitas

Dalam konteks rekayasa perangkat lunak, "manufakturabilitas" dapat diartikan sebagai kemudahan dalam merancang, mengembangkan, dan memelihara sistem. Proyek ini dinilai memiliki tingkat manufakturabilitas yang tinggi karena dirancang dengan arsitektur microservice yang modular, menggunakan teknologi open-source yang populer seperti Django, Ansible, dan Docker [6]. Arsitektur ini mempermudah proses perancangan dan mengurangi kompleksitas pengembangan.

Proyek ini dinilai memiliki tingkat manufakturabilitas yang tinggi karena:

- Kemudahan Desain: Sistem dirancang dengan arsitektur *microservice* yang modular, menggunakan teknologi *open-source* yang populer dan memiliki dokumentasi sangat baik seperti Django, Ansible, dan Docker. Ini mempermudah proses perancangan dan mengurangi kompleksitas.

- Ketersediaan Bahan dan Peralatan: "Bahan baku" utama adalah kode, dan "peralatan" adalah lingkungan pengembangan. Semua teknologi inti yang digunakan bersifat *open-source*, sehingga tidak memerlukan biaya lisensi. Perkakas pengembangan seperti *code editor*, Git, dan Docker juga tersedia secara gratis.

- Keahlian Tenaga Kerja: Keterampilan yang dibutuhkan (Python, Ansible, Docker, Networking) merupakan standar dalam industri DevOps dan rekayasa perangkat lunak modern, sehingga ketersediaan sumber daya manusia yang kompeten relatif lebih mudah ditemukan dibandingkan ahli spesialis untuk setiap vendor perangkat.

1.2.3 Aspek Teknis dan Keberlanjutan

Secara teknis, proses manual menciptakan "utang teknis" (technical debt) dalam bentuk konfigurasi yang tidak standar. Sistem yang diusulkan memberlakukan standarisasi melalui playbook Ansible, yang juga berfungsi sebagai "dokumentasi hidup" dari konfigurasi jaringan [7]. Penggunaan arsitektur microservice di atas Docker memastikan keberlanjutan sistem, di mana setiap komponen dapat diperbarui secara independen, membuat pemeliharaan jangka panjang menjadi lebih efisien [6].

1.3 Analisis Solusi yang Ada

Dalam menjawab tantangan manajemen jaringan modern, khususnya pada aspek provisioning dan monitoring perangkat multivendor, telah tersedia berbagai solusi yang berkembang baik dalam bentuk open-source maupun komersial. Solusi-solusi ini memiliki keunggulan masing-masing, namun juga menunjukkan keterbatasan saat dihadapkan pada kebutuhan integrasi penuh dalam lingkungan berbasis microservice dan antarmuka yang ramah pengguna.

Solusi open-source seperti Prometheus dan Grafana telah menjadi standar dalam monitoring infrastruktur karena fleksibilitas, skalabilitas tinggi, serta dukungan komunitas yang luas. Prometheus bertindak sebagai sistem pengumpulan metrik utama dengan dukungan ekstensi seperti SNMP Exporter untuk mengakses data perangkat jaringan [4]. sedangkan Grafana digunakan untuk menampilkan data dalam bentuk dashboard visual yang informatif. Meski demikian, untuk mengakomodasi kebutuhan spesifik dalam konfigurasi perangkat jaringan, sistem ini tetap memerlukan integrasi dan kustomisasi tambahan agar dapat berfungsi secara optimal dalam konteks provisioning.

Di sisi lain, solusi monitoring seperti PRTG dan Cacti juga menawarkan fitur lengkap untuk pemantauan jaringan. Namun, sistem ini umumnya tidak dirancang untuk dijalankan dalam lingkungan microservice yang dinamis, serta memiliki kurva pembelajaran yang cukup tinggi, sehingga kurang sesuai bagi pengembangan sistem ringan dan fleksibel.

Untuk solusi komersial, platform seperti Datadog dan Dynatrace memberikan fungsionalitas monitoring dan observabilitas yang sangat canggih, termasuk deteksi otomatis, pemetaan layanan, serta integrasi dengan cloud. Walau begitu, biaya lisensi yang tinggi serta ketergantungan pada vendor menjadi kendala utama, khususnya untuk implementasi di lingkungan pendidikan atau organisasi berskala menengah ke bawah yang membutuhkan solusi mandiri dan terbuka.

Dalam hal otomasi konfigurasi, tools seperti Ansible digunakan secara luas karena mampu mengotomatisasi proses provisioning melalui pendekatan deklaratif dan agentless [7]. Namun, fokus utamanya adalah pada otomasi, bukan monitoring, sehingga perlu digabungkan dengan sistem lain untuk mencapai solusi yang lengkap.

Berdasarkan evaluasi, dapat disimpulkan bahwa dibutuhkan sebuah platform terintegrasi yang menggabungkan otomasi berbasis Ansible, pemantauan berbasis Prometheus-Grafana, serta antarmuka pengguna yang terpusat, yang didukung oleh arsitektur microservice [8].

1.4 Tujuan Tugas Akhir

Tujuan dari tugas akhir ini adalah untuk merancang dan mengimplementasikan sistem provisioning dan monitoring perangkat jaringan multivendor berbasis web, dengan pendekatan arsitektur microservice dan menggunakan protokol Secure Shell (SSH) sebagai media komunikasi utama. Sistem ini diharapkan mampu mengotomatisasi proses konfigurasi (provisioning) perangkat dari berbagai vendor secara terpusat dan efisien, serta menyediakan pemantauan performa jaringan secara real-time melalui antarmuka visual yang informatif.

Secara lebih spesifik, tujuan yang ingin dicapai dalam tugas akhir ini adalah sebagai berikut:

1. Mengembangkan sebuah sistem berbasis web yang mampu melakukan provisioning (konfigurasi) secara otomatis ke berbagai perangkat jaringan multivendor (Cisco, MikroTik, Fortinet) menggunakan Ansible.
2. Membangun antarmuka pengguna grafis (GUI) yang terpusat untuk mengelola dan mengeksekusi playbook Ansible, sehingga menyederhanakan proses manajemen jaringan bagi administrator.
3. Mengintegrasikan sistem monitoring performa jaringan secara real-time menggunakan Prometheus dan Grafana untuk menampilkan metrik-metrik penting dari perangkat yang dikelola dalam satu dashboard.
4. Menerapkan arsitektur microservice menggunakan Docker untuk mengisolasi setiap komponen fungsional sistem (aplikasi web, otomasi, monitoring) sehingga meningkatkan modularitas dan kemudahan pemeliharaan.
5. Mengimplementasikan sistem notifikasi otomatis (alerting) yang akan mengirimkan peringatan ke platform Telegram ketika terdeteksi adanya anomali atau gangguan pada perangkat jaringan.

1.5 Batasan Tugas Akhir

Agar pengerjaan Tugas Akhir ini lebih terarah dan sesuai dengan lingkup yang telah ditetapkan, maka ditetapkan batasan-batasan masalah sebagai berikut:

1. Sistem dikembangkan dan diuji dalam lingkungan simulasi menggunakan PNETLab. Perangkat yang didukung terbatas pada Cisco IOSv, MikroTik CHR, dan FortiGate VM.
2. Fokus provisioning adalah pada konfigurasi dasar yang umum dilakukan, seperti perubahan hostname, konfigurasi interface (alamat IP), pembuatan VLAN, dan konfigurasi banner login. Tugas Akhir ini tidak mencakup otomasi konfigurasi protokol routing dinamis yang kompleks (contoh: BGP, OSPF).
3. Metrik monitoring yang ditampilkan terbatas pada data yang dapat diekstraksi melalui SNMP Exporter (utilisasi CPU, memori, trafik antarmuka) dan Blackbox Exporter (status ketersediaan via ICMP dan latency).
4. Sistem tidak melakukan proses hardening keamanan tingkat lanjut. Fokus keamanan hanya pada enkripsi data kredensial perangkat menggunakan fitur Ansible Vault.
5. Manajemen pengguna pada aplikasi web hanya mencakup satu level akses (administrator) dan tidak mengimplementasikan Role-Based Access Control (RBAC) yang kompleks.