

BAB I

PENDAHULUAN

A. Latar Belakang

Arsitektur jaringan pada masa sekarang sudah menjadi salah satu aspek yang sangat penting dalam dunia bisnis, perkantoran maupun dunia pendidikan. Terlebih dengan perkembangan jaringan komputer saat ini yang semakin meluas hingga hampir setiap kegiatan sehari-hari berhubungan dengan jaringan komputer. *Local Area Network (LAN)* atau jaringan lokal juga sangat cocok di implementasikan pada area tersebut. Oleh karena itu memiliki sistem dengan infrastruktur jaringan yang mudah dikelola namun memiliki kinerja yang optimal dan fleksibel menjadi salah satu faktor yang sangat penting. Menurut (Andri Kristanto, 2003) menyebutkan bahwa Jaringan komputer merupakan sekelompok komputer otonom yang saling berhubungan satu sama lain, dengan menggunakan satu protokol komunikasi sehingga seluruh komputer yang saling terhubung tersebut dapat berbagi informasi, program, sumber daya dan juga dapat saling menggunakan perangkat keras lainnya secara bersamaan, seperti *printer*, *harddisk* dan perangkat komputer lain-nya.

Namun di masa sekarang ini penggunaan jaringan tidak hanya mengenai saling berbagi *resource* seperti *hardware* dan *software* saja, selain itu dengan semakin tinggi kebutuhan jaringan yang semakin kompleks dan dinamis menjadikan jaringan komputer sukar untuk diatur. Dibutuhkan banyak peralatan untuk merancang dan mengatur jaringan komputer, seperti *router* dan *switch* peralatan *middlebox* seperti *firewall*, *translator* alamat jaringan, penyeimbang beban server (*load balance*), dan perangkat lainnya. Perangkat seperti *router* dan *switch* bekerja dengan sistem yang kompleks, kontrol perangkat lunak tersebut juga didistribusikan secara khusus, tertutup dan merupakan hak milik perusahaan yang memproduksi perangkat keras tersebut. Perangkat lunak yang digunakan juga mengimplementasikan *protocol* jaringan yang distandarisasi dan diuji coba beberapa tahun jauh kebelakang. *Administrator* jaringan secara khusus mengkonfigurasi perangkat jaringan menggunakan *interface* konfigurasi yang *vendor-based*, yang berarti setiap perangkat jaringan memiliki konfigurasi berbeda tiap *vendor*, bahkan banyak produk dengan satu *vendor* yang memiliki *interface* konfigurasi yang berbeda. Sistem jaringan seperti ini dioperasikan dengan *protocol*, mekanisme dan *interface* konfigurasi yang individual. Pengoperasian dengan model seperti ini dapat mengurangi inovasi pengembangan jaringan, meningkatkan kompleksitas dan menjadikan biaya operasional menjadi sangat tinggi dalam operasional jaringan. Selain pengoperasian yang kompleks dan membutuhkan biaya yang relatif tinggi dalam proses pengembangan jaringan metode ini akan menimbulkan

masalah yang rumit karena *administrator* sangat bergantung pada arsitektur *vendor-based* sehingga dalam prosesnya akan mengalami kesulitan.

Sebelumnya telah dilakukan beberapa penelitian mengenai SDN tentang Analisa Kinerja dan Karakteristik Arsitektur *Software Defined Network (SDN)* Berbasis Opendaylight Controller (M. H. Hidayat & Rosyid, 2017), perancangan Simulasi Jaringan Virtual Berbasis *Software-Define Networking* (Ummah, 2016) dan Analisis Perbandingan Performansi Kontroler Floodlight, Maestro, RYU, POX dan ONOS dalam Arsitektur *Software Defined Network (SDN)* (Putra et al., 2018). Penulis ingin membandingkan kinerja jaringan komputer dengan arsitektur konvensional dengan aritektur jaringan berbasis SDN. Berdasarkan fakta-fakta yang telah disebutkan, arsitektur jaringan konvensional seperti itu pada masa sekarang akan sulit untuk memenuhi kebutuhan tersebut sehingga arsitektur jaringan dengan arsitektur konvensional sudah tidak lagi cocok. Sehingga perlu dilakukan perubahan paradigma dalam mendesain dan mengatur jaringan menggunakan arsitektur SDN.

SDN adalah sebuah paradigma arsitektur baru dalam jaringan komputer yang memiliki karakteristik dinamis, *manageable*, *cost-effective*, dan *adaptable*, sehingga sangat ideal untuk kebutuhan aplikasi saat ini yang bersifat dinamis dan *high-bandwidth* (I. Hidayat & Perdana, 2019). Konsep pendekatan baru menggunakan arsitektur SDN ini memiliki cara kerja dengan memisahkan *network control (control plane)* dan fungsi *forwarding (data plane)* sehingga *control* pada jaringan berbasis SDN menjadi *directly programmable* (dapat diprogram secara langsung), dengan mengkonsolidasikan *control plane* menjadikan satu perangkat *control plane* dapat mengendalikan beberapa *control* program *data plane* melalui *Application Programming Interface (API)* yang terdefinisi. Untuk mengatur komunikasi antara control plane dan data plane dibuatlah standar komunikasi protocol pertama pada arsitektur SDN yaitu openflow. Openflow merupakan protocol yang dikembangkan oleh *Open Networking Foundation* untuk menangani proses komunikasi antara *control plane* dan *data plane*. Openflow memungkinkan *controller* untuk mengirimkan data pada switch dan openflow mendukung berbagai jenis vendor dalam sebuah jaringan.

Salah satu controller yang bisa digunakan dalam arsitektur SDN yaitu *Ryu controller*, *ryu* berfungsi untuk mengatur protokol openflow (*data plane*). Dalam *Ryu controller* terdapat *library* dan modul untuk mengatur aplikasi yang akan dibuat sesuai dengan yang diinginkan misalnya *switch*, *router load balancing* atau aplikasi lainnya yang sesuai dengan kebutuhan jaringan yang ada.

Berdasarkan pemaparan di atas, SDN dapat mengurangi kompleksitas dan memudahkan pengaturan jaringan komputer karena terpusat dalam suatu *controller*. Selain itu dengan *SDN* jaringan jadi lebih fleksibel dan dinamis karena bersifat

programmability network (Ummah, 2016). Pengimplementasian SDN pada jaringan komputer memberikan efisiensi yang cukup besar baik dari sisi penggunaan, pengembangan maupun biaya. Namun untuk mengetahui kualitas jaringan yang lebih baik, maka kedua arsitektur jaringan ini perlu dilakukan pengukuran kinerja dengan dilakukan pengujian pada jaringan tersebut kemudian dilakukan perbandingan. Pengukuran jaringan dilakukan menggunakan metode *Quality of Service (QOS)* sebagai parameter kualitas jaringan. Sehingga diajukan tugas akhir dengan judul “Optimasi Sistem Manajemen Jaringan Menggunakan Metode *Software Defined Network* pada *Ryu* dan *Openflow Protocol*”.

B. Permasalahan

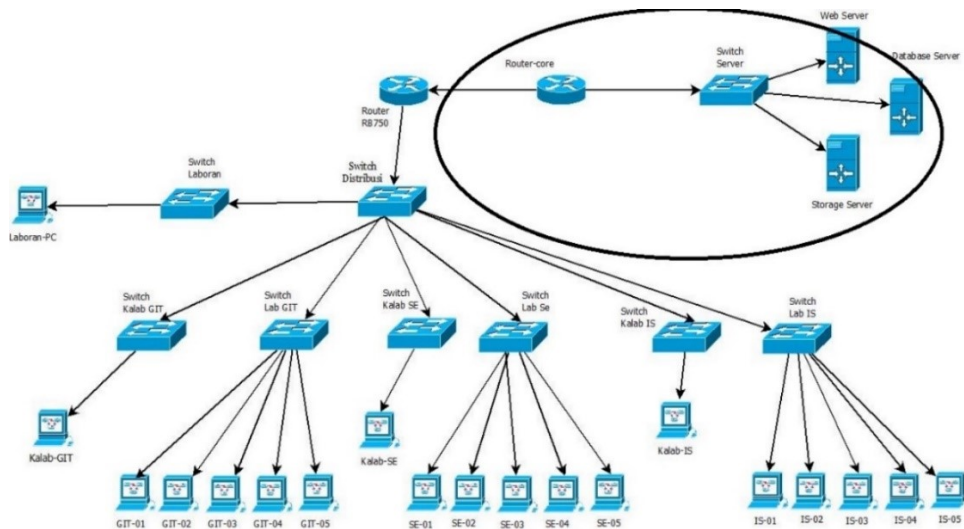
Fakultas Teknik dan Sains merupakan institusi tertua di kota bogor dalam bidangnya. Didirikan pada tahun 1974, fakultas teknik pada mulanya hanya mempunyai satu jurusan yaitu Teknik Sipil. Baru beberapa tahun kemudian jurusan Teknik Elektro didirikan (tepatnya pada tahun 1978) kemudian di susul dengan jurusan Teknik Mesin, dan kemudian membangun program studi teknik informatika. Dari masing-masing setiap program studi terdapat 4 laboratorium yang terbagi dalam 4 gedung yang berbeda. Dari masa ke masa fakultas teknik dan sains selalu dijadikan acuan universitas ibn khaldun sebagai fakultas yang memiliki teknologi yang lebih maju dibanding dengan fakultas yang lainnya hal ini ditunjukkan dari adaptasi sistem layanan akademis, pengelolaan sumber daya terutama pada bagian komputerisasi dan penggunaan jaringan komputer. pengelolaan jaringan di lingkungan Fakultas Teknik dan Sains pada awalnya masih sepenuhnya menggunakan perangkat dari beberapa vendor (*vendor-based*) yang mana dari masa ke masa mulai menjadi suatu ketergantungan akan kemudahan yang ditawarkan oleh vendor tersebut, pengelolaan jaringan dengan berbasis dari vendor seperti ini memang memiliki dampak yang positif bagi administrator jaringan dan fitur yang di tawarkan memiliki banyak pilihan bahkan seringkali melebihi dari apa yang dibutuhkan di dalam jaringan yang dikelola seperti user interface yang lebih mudah untuk di fahami, kemudahan konfigurasi, dukungan *technical support*, tersedianya buku manual atau dokumentasi mengenai perangkat yang dipakai dan datasheet dari perangkat yang digunakan karena setiap pabrikan pastinya memiliki catatan dokumentasi dari setiap perangkat yang diproduksi untuk kebutuhan *maintenance* seperti *update firmware* untuk memperbaiki masalah (*bug*) yang terjadi saat perangkat dijalankan.

Namun setelah beberapa tahun dijalankan kemudahan yang ditawarkan oleh vendor memiliki konsekuensi yang cukup serius untuk perkembangan pengelolaan jaringan dalam jangka panjang hal ini mulai dirasakan ketika kebutuhan jaringan yang saat ini semakin kompleks serta tuntutan untuk membangun arsitektur jaringan yang

dinamis dengan biaya yang efisien. Jaringan yang dibangun dengan sepenuhnya menggunakan arsitektur *vendor-based* akan membutuhkan biaya yang sangat mahal ketika kita ingin mengembangkan spesifikasi dari perangkat yang akan kita gunakan sehingga akan menjadi beban yang tinggi karena proses pengembangan dilakukan dengan mengganti perangkat dengan yang lebih baru dan membuang perangkat lama karena dianggap sudah tidak mendukung teknologi pada perangkat baru yang akan digunakan. Hal ini disebabkan pada siklus produksi suatu jenis perangkat memiliki *mean time between failure* (MTBF) yang merupakan batasan dari penggunaan perangkat yang sudah memiliki banyak kegagalan dalam melaksanakan fungsinya selain itu perangkat dari vendor biasanya memberlakukan *discontinue product* dalam kurun waktu 1-3 tahun atau bahkan mencapai hingga 5 tahun bergantung pada perkembangan teknologi dari tahun ke tahun, Jadi secara tidak langsung akan menjadi proses membangun jaringan kembali dari nol yang akan membutuhkan waktu yang cukup lama. Konsekuensi lainnya akibat ketergantungan terhadap arsitektur *vendor-based* yaitu kemampuan atau kepiawayan administrator dalam mengelola jaringan tidak akan berkembang karena ketergantungan tersebut. Sehingga ketika akan melakukan pengembangan jaringan tugas tersebut akan ditangani kembali oleh *vendor* yang akan membuat biaya yang dikeluarkan kembali lebih besar dan kemampuan administrator dalam mengelola jaringan menjadi tidak berkembang.

Menyadari permasalahan tersebut dalam melakukan optimasi jaringan administrator jaringan memutuskan untuk mengganti manajemen jaringan dengan metode lain yang dapat memenuhi kebutuhan akan fleksibilitas jaringan karena dalam jaringan yang berbasis *vendor* tidak memiliki *interface* konfigurasi lintas *vendor* (*cross platform*) sehingga banyak keterbatasan dalam mengelola jaringan yang menjadikannya tidak fleksibel, faktor lain yang mendorong *administrator* untuk mengganti *management* jaringan adalah faktor biaya yang dikeluarkan dapat menjadi lebih efisien. Dalam proses perubahan manajemen jaringan saat ini masih mencakup pada tahap lingkup server dengan menggunakan metode *Infrastruktur as a Service* (IaaS) dimana penerapan *Network Function Virtualization* (NFV) ini hanya mencakup pada area komputer *server* yang digunakan untuk layanan aplikasi yang berjalan berbasis web dan sebagai server penyimpanan, belum mencakup pada perangkat jaringan yang lain seperti *switch* dan perangkat jaringan lain. Area yang menggunakan metode IaaS dapat terlihat pada gambar yang terdapat lingkaran pada gambar 1.1 topologi jaringan dan metode pengelolaan jaringan yang ada, sedangkan pada bagian luar lingkaran masih bergantung pada *vendor*, proses pengelolaannya masih menggunakan cara konvensional, oleh sebab itu pengelolaan jaringan harus dioptimalkan agar semua perangkat jaringan dapat dikelola dengan maksimal. Untuk

mengoptimalkan pengelolaan jaringan tersebut harus menggunakan metode yang tepat dimana dapat mencakup pada keseluruhan jaringan tidak bergantung pada vendor serta fleksibilitas dalam pengembangannya dapat dilakukan tanpa kendala terhadap merek tertentu. Diperlukan sebuah arsitektur yang dinamis, dapat memudahkan pengelolaan, hemat biaya, dan mudah beradaptasi, sehingga dapat menghilangkan ketergantungan terhadap vendor.



Gambar 1.1 Topologi Jaringan dan Metode Pengelolaan Jaringan yang ada

1. Identifikasi Masalah

Faktor – faktor permasalahan yaitu:

- a. Penggunaan jaringan komputer dengan metode IaaS belum mencakup pada keseluruhan jaringan dimana hanya terkonsentrasi pada NFV karena hanya berada pada cakupan server.
- b. Jaringan sepenuhnya bergantung pada perangkat vendor sehingga mengelola dan mengembangkannya menjadi hal yang kompleks bagi administrator jaringan.
- c. Kontrol pada jaringan belum terpusat sehingga kinerja jaringan dilakukan seluruhnya secara individual oleh masing-masing perangkat.

2. Pernyataan Masalah / Problem Statement

Berdasarkan identifikasi masalah maka dapat disimpulkan pokok masalah yaitu metode pengelolaan jaringan belum sepenuhnya efektif karena seluruhnya bergantung pada vendor sehingga untuk melakukan pengembangan sangatlah kompleks dan membutuhkan biaya yang mahal.

3. Pertanyaan Penelitian / *Research Question*

Pertanyaan penelitian yang dapat di ajukan adalah sebagai berikut:

- a. Bagaimana mengelola jaringan menggunakan metode SDN?
- b. Apakah kualitas jaringan menjadi lebih baik menggunakan metode SDN dibandingkan dengan jaringan dengan metode konvensional?
- c. Bagaimana menghilangkan ketergantungan *administrator* terhadap suatu *vendor* dalam mengelola infrastruktur jaringan?

C. Maksud Dan Tujuan Pengembangan

1. Maksud

Maksud dari penelitian ini adalah menguji kinerja dua arsitektur jaringan yaitu jaringan konvensional dan SDN untuk mengetahui kinerja kedua jaringan tersebut serta mengoptimalkan pengelolaan jaringan komputer melalui pendekatan paradigma baru SDN menggunakan *Ryu controller* dan *Openflow protocol* untuk mengatasi ketergantungan *administrator* pada suatu *vendor* (*vendor-base*) serta membangun jaringan terprogram (*Programmable Network*).

2. Tujuan

- a. Mengimplementasikan metode jaringan SDN menggunakan *Ryu Controller* dan *Openflow protocol* untuk menghilangkan ketergantungan *administrator* terhadap *vendor*.
- b. Mempermudah proses *maintenance* dan manajemen jaringan bagi *administrator*
- c. Mendapatkan nilai QoS dengan parameter *throughput*, *delay* dan *packet loss* pada jaringan, untuk menentukan pengelolaan dan pengembangan jaringan dikemudian hari.

D. Spesifikasi Produk Yang Diharapkan

Melalui penelitian ini diharapkan Implementasi SDN dapat mengoptimalkan penggunaan jaringan komputer dilingkungan Fakultas Teknik dan Sains Universitas Ibn Khaldun dengan memudahkan proses *maintenance* dan manajemen jaringan serta dapat menghilangkan ketergantungan *administrator* terhadap suatu *vendor* untuk mengembangkan konsep jaringan modern seperti SDN, selain itu mendapatkan nilai QoS dengan parameter *throughput*, *delay* dan *packet loss* pada jaringan yang di ukur untuk kemudian data tersebut dapat dijadikan sebagai acuan data pengembangan jaringan untuk kedepannya.

E. Signifikasi Penelitian

Dilaksanakannya penelitian ini dalam rangka menguji kinerja dua arsitektur jaringan yaitu jaringan konvensional dan jaringan berbasis SDN untuk mengetahui kualitas kedua jaringan tersebut dengan di uji menggunakan metode QoS dan menghilangkan ketergantungan *administrator* terhadap suatu *vendor* serta memusatkan *control* jaringan pada *Ryu controller*. Adapun manfaat dari penelitian ini yaitu:

1. Berkontribusi dalam pengembangan ilmu pengetahuan khususnya mengenai *SDN*.
2. Manfaat praktis dari penelitian ini adalah memudahkan *administrator* dalam pengelolaan jaringan dan menghilangkan ketergantungan terhadap *vendor*.
3. Kontrol jaringan menjadi lebih efektif karena terpusat pada *Ryu controller* sehingga dapat memangkas waktu *maintenance*.
4. Manajemen jaringan yang lebih baik dengan terciptanya *programmability network* yaitu jaringan yang dapat diprogram sesuai dengan kebutuhan.
5. Manfaat kebijakan penelitian ini yaitu dapat dijadikan acuan dalam mengoptimalkan jaringan komputer menggunakan metode SDN.

F. Asumsi Dan Keterbatasan Penelitian

Asumsi dalam penelitian yang akan dikembangkan yaitu:

1. Terbentuknya jaringan berbasis SDN di lingkungan laboratorium Teknik Informatika.
2. Sistem yang dibuat akan membantu institusi agar tidak bergantung pada *vendor*.
3. Terciptanya *programmable network* pada jaringan di lingkungan laboratorium Teknik Informatika.

Dalam penelitian ini, terdapat beberapa keterbatasan dalam pengembangan jaringan berbasis *Software Defined Network (SDN)* yaitu:

1. Menggunakan *Ryu controller* sebagai *control plane*, *Mininet* sebagai *Data Plane*.
2. Pada penelitian ini berfokus pada perbandingan kinerja jaringan komputer dengan arsitektur konvensional dan jaringan dengan arsitektur *Software Defined Network (SDN)*.
3. *Controller* yang digunakan hanya berjalan pada sistem operasi berbasis linux ubuntu, dengan proses simulasi menggunakan *Mininet* sebagai *data plane*

G. Definisi Istilah

1. *Switch*

Switch adalah suatu komponen jaringan komputer yang berfungsi untuk menghubungkan beberapa perangkat komputer agar dapat melakukan pertukaran paket, baik menerima, memproses, dan meneruskan data ke perangkat yang dituju.

2. *Administrator*

Orang yang bertugas untuk mengatur sebuah jaringan komputer baik itu dalam skala kecil maupun skala besar

3. Virtualisasi

Virtualisasi merupakan upaya untuk menghasilkan suatu bentuk virtual dari sesuatu yang bersifat fisik. Hal ini bisa berupa perangkat penyimpanan data maupun sistem operasi, termasuk juga pembuatan sumber daya tunggal seperti *server*. Jenis virtualisasi pun ada banyak, seperti perangkat keras, perangkat lunak, jaringan, data, dan memori.

4. *Server*

Server adalah sebuah sistem komputer yang menyediakan jenis layanan (*service*) tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat *scalable* dan *RAM* yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan (*network operating system*). *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau alat pencetak (*printer*), dan memberikan akses kepada *workstation* anggota jaringan.

5. *Emulator*

Emulator atau lebih tepatnya peranti lunak emulator memungkinkan suatu program atau peranti lunak yang dibuat pada awalnya oleh suatu sistem komputer (arsitektur dan sistem operasi) dan untuk dijalankan dalam sistem itu (atau dijalankan dalam suatu sistem yang didedikasikan), dapat dijalankan dalam sistem komputer yang sama sekali berbeda.

6. *Router*

Router merupakan perangkat keras jaringan komputer yang dapat digunakan untuk menghubungkan beberapa jaringan yang sama atau berbeda. *Router* adalah sebuah alat untuk mengirimkan paket data melalui jaringan atau internet untuk dapat menuju tujuannya, proses tersebut dinamakan *routing*.