BAB I
PENDAHULUAN

1.1 Latar Belakang

Jaringan komputer tersusun dari banyak perangkat seperti *switch,firewall* dan *router* serta umumnya dikelola oleh penyedia layanan internet. Pada suatu organisasi,jaringan komputer dikelola oleh divisi yang dibentuk khusus untuk mengelola,menangani dan memanajemen jaringan komputer yang ada. Dengan pentingnya peran jaringan komputer di era modern saat ini maka infrastruktur jaringan harus memiliki skalabilitas,fleksibilitas,*availability* dan optimasi yang baik. Kebutuhan akan jaringan komputer,secara tidak langsung akan membuat jaringan komputer menjadi sangat kompleks dengan melibatkan ribuan perangkat jaringan. Ribuan perangkat jaringan ini harus dikelola dengan baik oleh administrator jaringan. Didalam konsep jaringan tradisional atau jaringan statis yang *control plane* dan *data plane* masih dalam perangkat yang sama maka manajemen jaringan yang besar ini akan sulit dilakukan. Pada dasarnya protokol jaringan yang ada tidak banyak berubah tetapi setiap vendor membatasi dan memiliki konfigurasi protokol yang berbeda-beda. Sehingga saat sebuah kebijakan telah diterapkan atau diimplementasikan pada sistem jaringan maka ketika kebijakan tersebut akan diubah karena suatu sebab atau akan dimuat ulang karena ada kesalahan maka salah satu cara yang bisa dilakukan adalah dengan melakukan konfigurasi ulang keseluruhan perangkat yang terdampak.[1] Tentu saja melakukan konfigurasi ulang keseluruhan perangkat akan menyita banyak waktu,memiliki resiko kesalahan konfigurasi dan belum lagi dengan sulitnya melakukan integrasi teknologi baru yang disebabkan oleh perbedaan platform perangkat. Dengan demikian jaringan tradisional/statis memiliki beberapa kekurangan seperti tidak memiliki fleksibilitas yang baik,perangkat jaringan yang digunakan tertutup dan bukan hak organisasi sehingga administrator jaringan harus melakukan konfigurasi berdasarkan *vendor-based*.[1] Oleh Karena itu diperlukan suatu pendekatan untuk memudahkan konfigurasi,memiliki fleksibilitas dan skalabilitas yang baik.

Pendekatan yang dapat digunakan dalam menjawab masalah diatas yaitu *Software Defined Network (SDN)*. *Software Defined Network (SDN)* adalah sebuah paradigma arsitektur baru dalam bidang jaringan komputer yang memiliki karakteristik dinamis, *manageable*, *cost effective*,dan *adaptable* sehingga sangat ideal untuk kebutuhan aplikasi yang bersifat dinamis.[2] *Software-Define Network (SDN)* merupakan sebuah konsep pendekatan baru untuk mendesain, membangun dan mengelola jaringan komputer dengan memisahkan *control plane* dan *data plane*. *Control plane* pada jaringan bertanggungjawab untuk melakukan pemilihan keputusan dalam meneruskan paket dan *data plane* bertanggungjawab melakukan packet *switch*ing. Konsep utama dari *software define network* adalah sentralisasi kendali dengan semua pengaturan kebijakan terletak di *control plane*. Konsep ini akan memudahkan administrator jaringan dalam mengelola infrastruktur jaringan yang besar. *Software-Define Network* mampu menjawab permasalahan jaringan yang diuraikan diatas dengan menawarkan kemudahan mengintegrasikan teknologi baru,tidak bergantung pada vendor dan dapat melakukan perubahan kebijakan jaringan dengan cepat. *Data plane* pada konsep *software define network* tetap berada dalam perangkat jaringan sedangkan *control plane* terdapat pada entity terpisah yang dinamakan sebagai *controller*. *Controller* merupakan *software* yang bersifat fleksibel untuk dikonfigurasi sehingga jaringan dapat dikontrol dengan mudah. *Data plane* yang terletak pada perangkat jaringan menjadikan perangkat jaringan hanya bisa melakukan *forwarding* atau meneruskan data dari perangkat ke perangkat sampai data tersebut sampai ke tujuan. Di Dalam *switch* terdapat *table-flow* yang berisi aturan yang digunakan sebagai acuan untuk meneruskan data supaya sampai ke tujuan. *Controller* akan bertanggungjawab dalam dalam bagaimana menangani paket dan mengelola *table-flow* dengan menambahkan atau menghapus isi *table-flow* melalui *secure channel*. Dalam konsep *software define network* peran *controller* sangat penting maka performa *controller* perlu diuji sehingga dapat mengetahui kemampuan *controller* yang digunakan.[3]

Beberapa *controller* yang telah berkembang diantaranya *OpenDayLight (ODL)*, *POX*, *NOX*, *Beacon*, *Ryu*, *ONOS*,*Floodlight*, *Maestro* dan masih banyak lainnya yang masih terus dikembangkan. Dengan banyaknya *controller* maka ada 10 hal

yang harus dipertimbangkan dalam memilih *controller*,salah satunya adalah performa yang dimiliki oleh *controller*.[4]

Dengan banyaknya *controller* yang tersedia maka diperlukan pengujian untuk mengetahui sejauh mana performa *controller* tersebut serta melakukan perbandingan dengan *controller* lain untuk mengetahui perbedaannya.

Dalam penelitian ini diambil dua *controller* yaitu *OpenDayLight* yang merupakan *controller* berbasis *java* yang dapat dijalankan di semua perangkat keras yang mendukung *java* dan *ONOS (Open Network Operating System)* merupakan *software define network controller* yang *open source* dan didesain sebagai solusi dalam *software define network*.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan diatas maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana perbandingan kinerja *controller ONOS* dan *OpenDayLight*?
2. Bagaimana performa *controller ONOS* dan *OpenDayLight* dalam topologi *tree,mesh,ring,star* serta *switch* yang digunakan yaitu 24 *switch* 48 *host* dan 32 *switch* 64 *host*?
3. Seberapa besar perbedaan nilai parameter *IPD,jitter,throughput ,packet loss* antara *controller ONOS* dan *OpenDayLight* dalam pengujian skenario yang sama?

1.3 Batasan Masalah

Adapun batasan masalah agar penelitian ini tidak menyimpang dari identifikasi masalah, diantaranya :

1. Kontroler yang digunakan adalah *ONOS* 2.7.0 *(code-named X-Wing)* dan *OpenDayLight (8.0.4 (Oxygen))*
2. Penelitian ini sepenuhnya dilakukan dengan simulasi menggunakan *Mininet*.
3. Menggunakan protokol *OpenFlow* versi 1.3
4. *Software* yang digunakan untuk pengujian adalah *ping* dan *iperf*.

5. Penelitian ini tidak membahas masalah keamanan pada arsitektur *SDN/OpenFlow*.
6. Penelitian lebih difokuskan pada pengujian performa dari masing masing kontroler.
7. Spesifikasi perangkat yang digunakan memiliki,RAM 16GB,CPU AMD Ryzen 9 4900H.
8. Menggunakan sistem operasi Linux Debian 11.

1.4 Tujuan Penelitian

Tujuan yang akan dicapai oleh peneliti dalam penelitiannya adalah menghasilkan informasi tentang sejauh mana performa dari *controller OpenDayLight* dan *ONOS*.

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan kontribusi yang bermanfaat baik teoritis maupun secara praktis.

1. Mengetahui sejauh mana performa *controller ONOS* dan *OpenDayLight* dalam beberapa skenario pengujian.
2. Dengan mengetahui performa *controller ONOS* dan *OpenDayLight* dapat memilih penggunaan *controller* yang tepat.
3. Mengetahui perbandingan nilai parameter *IPD,jitter,throughput ,packet loss* antara *controller ONOS* dan *OpenDayLight*
4. Diharapkan menjadi panduan atau referensi mengenai perbandingan kinerja *Controller ONOS* dan *OpenDayLight*.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN Bab pendahuluan mendeskripsikan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA Berisi tentang teori-teori yang digunakan dalam penelitian, perancangan dan pembuatan sistem.

BAB III METODE PENELITIAN Dalam bab ini penulis mengemukakan metode penelitian yang dilakukan dalam perancangan dan implementasi.

BAB IV HASIL DAN PEMBAHASAN Memaparkan dari hasil-hasil tahapan penelitian, mulai dari analisis, desain, hasil testing dan implementasinya.

BAB V PENUTUP Berisi kesimpulan dan saran dari seluruh penelitian yang telah dilakukan.