

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang Masalah

Lahirnya jaringan komputer bermula dari sebuah konsep pada tahun 1940an di Amerika yang dikembangkan oleh proyek pengembangan komputer model 1 di laboratorium Bell dengan riset Universitas Harvard yang di pimpin profesor Howard Aiken. Proyek tersebut hanya ingin memanfaatkan sebuah perangkat komputer yang di pakai bersama untuk mengerjakan beberapa proses tanpa banyak membuang waktu kosong dibuat proses beruntun (*Batch Processing*), sehingga beberapa program bisa di jalankan dalam sebuah komputer dengan gambaran antrian [6].

Jaringan komputer sendiri dapat meminta dan memberikan layanan, pihak yang meminta/menerima layanan disebut klien (*client*) dan yang memberikan/mengirim layanan disebut peladen (*server*), desain ini disebut sistem *client-server* digunakan pada hampir seluruh aplikasi jaringan komputer [4].

*ONF* mendefinisikan *Software-defined Network (SDN)* arsitektur jaringan yang muncul di mana kontrol jaringan dipisahkan dari *forwarding* dan langsung *diprogram*, migrasi kontrol sebelumnya terikat erat dalam individu perangkat jaringan, kedalam perangkat komputasi diakses memungkinkan mendasari infrastruktur dicabut untuk aplikasi dan layanan jaringan, yang dapat mengobati jaringan sebagai entitas logis atau virtual [5].

Tujuan dari *Software-defined Network (SDN)* sendiri penyederhanaan jaringan dan memunculkan aplikasi-aplikasi jaringan yang baru, dalam *Software-*

*defined Network (SDN)* terdapat sebuah *kontroller* yang dapat memberikan perintah semua perangkat yang terhubung secara langsung. *Software-defined Network (SDN)* memiliki arsitektur yang berbeda dari arsitektur jaringan komputer sebelumnya. *Software-defined Network (SDN)* erat kaitannya dengan *OpenFlow* sehingga banyak yang beranggapan bahwa *Software-defined Network (SDN)* adalah *OpenFlow* itu sendiri. [4]

*OpenFlow* sendiri sebuah elemen dalam arsitektur *Software-defined Network (SDN)*. *OpenFlow* merupakan open standar komunikasi protokol yang mampu melakukan pemisahan antara *control plane* dan *data plane* dari sebuah perangkat jaringan, serta mampu menciptakan komunikasi yang sangat baik antara *control plane* dan *data plane* [10].

*OpenFlow* bukanlah satu-satunya protokol yang dapat digunakan untuk mengembangkan *Software-defined Network (SDN)* masih banyak protokol lainnya baik yang *open source* maupun yang *proprietary*. *Software-defined Network (SDN)* memiliki banyak *tools* yang dapat digunakan salah satu nya adalah *OpenFlow* yang memanipulasi *FIB/TCAM* [5].

Semakin banyaknya perangkat yang terhubung ke dalam jaringan yang menyebabkan sulitnya melakukan kontrol pada jaringan yang ada pada saat ini, yang membuat *administrator* jaringan cukup kesulitan dalam mengatur sebuah jaringan, menganalisa lalu lintas jaringan, baik pada jaringan *internal* maupun *external* maka digunakanlah *Software-defined Network (SDN)* untuk mengatasi masalah yang ada pada jaringan saat ini dengan menggunakan *tools OpenFlow* yang terdapat pada *Software-defined Network (SDN)*. Terdapat sebuah aplikasi

yang dinamakan *mininet* yang digunakan untuk membuat simulasi jaringan *Software-defined Network (SDN) OpenFlow*, *mininet* selanjutnya yang biasa kita kenal *cisco packet tracer*. Namun pada *mininet* tidak mengenal *GUI*, *mininet* sendiri lebih dominan menggunakan *command terminal* layaknya tampilan *console*.

*OpenDayLight* adalah sebuah proyek *Open Source software* dalam naungan *Linux Foundation* untuk jaringan *programmability* mengaktifkan *SDN* dan *NFV* untuk jaringan di setiap ukuran dan skala. Tujuannya melanjutkan penerapan dan inovasi *Software-Defined Networking (SDN)* melalui penciptaan *framework* yang umum dikalangan industri [4].

Pada konfigurasi *mininet* bahasa pemrograman yang digunakan adalah bahasa pemrograman yang telah biasa digunakan sebagian orang yaitu *Python*. *Mininet* juga bisa terhubung langsung ke controller jaringan yang dipakai dengan begitu memudahkan untuk melihat bentuk dan mengkonfigurasi jaringan itu sendiri. controller pada *Software-defined Network (SDN)* merupakan tempat untuk mengatur atau mengontrol jaringan. Terdapat beberapa controller yang dapat dipakai pada saat ini seperti *Beacon* dan *OpenDayLight*.

Kedua controller akan ditesting dengan alat uji performa yang sering digunakan dalam penelitian yaitu *Collective Benchmark Tool* atau *cbench*. *Cbench* didefinisikan sebagai sebuah *benchmarking tool* yang merupakan bagian dari *Oflaps*. Dengan menggunakan *cbench* kita bisa melihat nilai *Throughput* dan *Latency* dari hasil pengujian masing-masing *controller*, yang memungkinkan untuk mengetahui nilai performa dari kinerja *controller* tersebut. Untuk itu

peneliti melakukan penelitian yang berjudul uji performa *controller Beacon* dan *open daylight* pada *software-defined network OpenFlow* menggunakan emulator *mininet*.

## 1.2 Rumusan Masalah

Adapun rumusan masalah yang diperoleh dari latar belakang permasalahan adalah bagaimana perbandingan performa *controller Beacon* dan *controller OpenDayLight* pada jaringan *Software-defined Network (SDN) OpenFlow*?

## 1.3 Batasan Masalah

Dengan adanya batasan masalah maka dapat lebih disederhanakan lagi dan lebih terarah agar penelitian tidak menyimpang dari apa yang di teliti. Adapun batasan masalah adalah sebagai berikut :

1. *Controller* yang digunakan adalah *Beacon* dan *OpenDayLight*.  
*Software* yang digunakan untuk menguji performa *controller* adalah *Cbench*.
2. Penelitian ini tidak membahas masalah keamanan pada arsitektur SDN/OpenFlow.
3. Penelitian lebih difokuskan pada pengujian performa dari masing masing *controler*.
4. Point yang di uji dari tiap *controller* adalah *throughput* dan *latency*.
5. *System operasi* yang digunakan *Ubuntu 14.04 LTS*.

## 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Untuk merancang dan menerapkan jaringan *Software-defined Network (SDN) OpenFlow* menggunakan *emulator mininet*.
2. Mengetahui performa terbaik dari *controller Beacon* dan *OpenDayLight*.

### 1.5 Manfaat Penelitian

Penelitian ini dilakukan dan memiliki beberapa manfaat bagi beberapa pihak yang antara lain adalah sebagai berikut :

1. Bagi penulis
  - a. Menambah wawasan tentang penerapan jaringan *Software-defined Network (SDN) OpenFlow* menggunakan *emulator mininet*.
  - b. Mengetahui perbedaan performa dari *controller Beacon* dan *controller OpenDayLight*.
2. Bagi instansi
  - a. Menjadi referensi bagi pihak kampus untuk melakukan penelitian-penelitian selanjutnya.
  - b. Menjadi dokumentasi bagi pihak kampus.
3. Bagi pembaca
  - a. Mengetahui konsep jaringan *Software-defined Network (SDN) OpenFlow*.
  - b. Mengetahui cara membangun jaringan *Software-defined Network (SDN) OpenFlow*.

- c. Mengetahui *controller* mana yang lebih baik dalam penggunaan jaringan *OpenFlow*.
- d. Menjadi referensi untuk penelitian-penelitian selanjutnya.

## 1.6 Metodologi Penelitian

Metode penelitian yang dilakukan penulis guna lancanya penelitian yang berlangsung adalah sebagai berikut :

### 1. Alat dan Bahan Penelitian

Dalam perancangan jaringan dilakukan konsep permodelan *system* seperti perancangan *topology*, gambaran umum *system*.

### 2. Alur Penelitian

Proses yang terdapat pada alur penelitian adalah sebagai berikut :

- a. Instalasi *controller* Beacon dan *OpenDay-Light* ke dalam sistem operasi Ubuntu 14.04 LTS.
- b. Uji kesiapan *controller Beacon* dan *OpenDay-Light*, menguji keberhasilan langkah pertama dengan *emulator mininet* yang sebelumnya telah *diinstall*, dengan skenario *topologi standart*. Bila *controller* tidak dapat mengenal *switch* pada *mininet*, maka proses *installasi* akan di ulang.
- c. Melakukan pengujian *latency controller Beacon* dan *controller OpenDayLight* dengan jumlah *switch* mulai 20 hingga 450 *switch* dan jumlah *host* setiap *switch* adalah 5 *host*.

- d. Melakukan pengujian *throughput Beacon* dan *OpenDayLight* dengan jumlah *switch* optimal dari pengujian *latency* dan melakukan variasi jumlah *host* hingga 450 *host*.
- e. Dokumentasi hasil
- f. Menarik analisis dan kesimpulan

### 1.7 Sistematika Penulisan

Adapun sistematika penulisan agar dapat membantu dan mempermudah penulis dalam melakukan penulisan laporan agar tidak menyimpang dari batasan masalah yang dijadikan sebagai kerangka penulisan laporan Skripsi maka penulis menyusun laporan penelitian ini menjadi 5 bab yaitu :

#### **BAB I : PENDAHULUAN**

Bab ini merupakan pendahuluan yang menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

#### **BAB II : LANDASAN TEORI**

Bab ini membahas mengenai dasar – dasar teori yang digunakan untuk menerapkan serta membandingkan performa *controller Beacon* dan *OpenDayLight SDN OpenFlow*.

#### **BAB III : METODOLOGI PENELITIAN**

Bab ini membahas tentang analisa masalah, analisa kinerja dan bagaimana cara membangun sistem serta bagaimana cara membandingkan performa *controller*.

#### **BAB IV : HASIL DAN PEMBAHASAN**

Bab ini membahas tentang hasil pengujian dari masing-masing *controller* dimana nantinya di bab ini akan di bandingkannya hasil dari *controller Beacon* dan *controller OpenDaylight* sehingga di dapatkan *controller* mana yang lebih baik.

#### **BAB V : PENUTUP**

Bab ini merupakan bagian akhir dari penulisan skripsi yang berisi kesimpulan dan saran dari perbandingan peforma *controller SDN OpenFlow*.

