

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini *cloud computing* sangat berperan penting dalam penerapan layanan baru dan telah menjadi infrastruktur bagi masyarakat digital [1]. Seperti layanan *infrastructure as a service (IaaS)*, *Platform as a Service (PaaS)*, *Software as a Service (SaaS)*, dan lain-lain [2]. Sehingga mendorong perusahaan untuk membangun data centernya sendiri dan mendorong dari segi ekonomi telah menyebabkan munculnya berbagai penyedia layanan data center, seperti *Amazon Web Services*, *Google Cloud Platform*, dan *Microsoft Azure*.

Membangun dan mengelola infrastruktur *data center* semakin menantang karena harus memenuhi kebutuhan *high performance*, *high availability*, *visibility*, *dynamic*, dan *programmable* [6]. Untuk memenuhi persyaratan tersebut langkah pertama adalah memahami kondisi yang sedang terjadi pada infrastruktur jaringan [7]. Sejak tahun 1990, SNMP (*Simple Network Management Protocol*) telah digunakan untuk manajemen jaringan dan didukung secara luas oleh perangkat jaringan dan platform monitoring [8]. Mekanisme *polling* dan "*pull the data*" digunakan pada SNMP, proses *request response* terjadi pada metode *polling*, dimana sistem manajemen perlu membuat dan mengirim permintaan SNMP *request* ke setiap perangkat, kemudian perangkat atau *agent* akan mengirimkan informasi terkait kondisi jaringan melalui SNMP *response*. Proses tersebut diulang terus menerus setiap kali ingin mendapatkan informasi terkait kondisi infrastruktur jaringan [9]. sehingga terdapat waktu pemrosesan dan *size* paket informasi yang semakin luas karena adanya *header request & response* sehingga berpengaruh pada *storage* [8]. Masalah lain dengan SNMP adalah umumnya pemrosesan *request response* yang terletak pada infrastruktur jaringan seperti *switch*, *router*, dll berjalan pada CPU yang akan memperlambat kinerja perangkat [10].

Streaming network telemetry adalah pendekatan baru untuk memantau dan mengelola jaringan, *streaming telemetry* menggunakan mekanisme berbasis *pushing* "*push the data*" [9], yang memungkinkan data informasi jaringan ditransmisikan secara otomatis dan terus menerus secara *real time* dari infrastruktur jaringan ke satu atau lebih platform terpusat sebagai penyimpanan dan visualisasi [10]. Teknologi *software defined networking (SDN)* digunakan untuk mengontrol dan *monitoring*

streaming telemetry infrastruktur jaringan secara terpusat dan terprogram, saat ini kebanyakan SDN menggunakan pendekatan *bottom-up programming* untuk mendukung *streaming telemetry*, Tools yang biasa digunakan adalah *NetFlow* atau *sFlow*. Untuk memahami keadaan jaringan menggunakan pendekatan *bottom-up programming*, setiap perangkat atau *data plane* pada infrastruktur jaringan berfungsi sebagai *agent* yang secara *passive* melakukan *streaming data*, Kemudian manajemen atau *control plane* menggabungkan informasi tersebut ke dalam *storage* terpusat, dan kemudian hanya menampilkan informasi yang dibutuhkan, namun informasi yang tidak dibutuhkan berdampak pada penyimpanan, *bandwidth* dan *over-head* pemrosesan yang tinggi [7]. Hal tersebut karena metode *bottom-up programming* memiliki kekurangan untuk mendefinisikan bagaimana *data plane* bekerja, karena tidak bisa membuat atau menghapus *match action pipelines* dan terbatas oleh jumlah *fixed table*, *table size*, *match fields* dan *match actions* [11]. Sehingga *streaming telemetry* hanya terbatas pada fitur yang disediakan, sedangkan kebutuhan untuk monitoring saat ini sangat kompleks, seperti informasi status jaringan *hop-by-hop* untuk mendukung *end-to-end visualization* [12].

Untuk mengatasi masalah tersebut adalah dengan menggunakan *In-band Network Telemetry* (INT), INT adalah *framework* abstraksi baru dalam teknologi monitoring yang dapat memberikan informasi yang sangat rinci tentang perilaku jaringan dengan cara menyisipkan sejumlah kecil informasi atau INT *header* langsung ke dalam paket yang melewati perangkat jaringan yang memiliki fungsionalitas INT [13]. operator dapat menambahkan INT *header* berdasarkan *flows*, *packet*, *protocols*, sampai *high-level names* [14]. Dengan begitu *in-band network telemetry* memungkinkan *end-to-end visualization* seperti melihat *end-to-end flows latency*, karakteristik *flow*, *queue/buffer occupancy*, dan lain-lain [13]. Untuk mendukung INT digunakan pendekatan *top-down programming* [7]. P4 adalah bahasa pemrograman untuk *top-down programming* yang dapat menentukan bagaimana *pipelines switch* bekerja dan bagaimana paket-paket diproses [15]. Dengan begitu kemampuan untuk mengumpulkan data bisa diproses pada tingkat yang lebih rendah, seperti ASICs, *fabric switching*, *chip switching*, dan sebagainya. Sehingga tidak berpengaruh pada kinerja *processing* karena berada di luar CPU [10].

Oleh karena itu pada Tugas Akhir ini penulis akan melakukan simulasi dan analisis terhadap *in-band network telemetry* pada *programmable network infrastructure* yang berbasis *P4 language*. Penelitian ini akan dilakukan dengan menggunakan *emulator mininet* dan *P4 BMv2 switch* untuk membuat simulasi infrastruktur jaringan, *open network operating system* (ONOS) sebagai *controller* dan platform lain yang dijelaskan pada bab terpisah. Analisis yang dilakukan adalah pengujian fungsional-

alitas program INT pada P4 dan performansi dengan melakukan pengujian *storage overhead*, *protocol overhead*, *protocol overhead* dan *hop latency*.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah dari tugas akhir ini adalah sebagai berikut:

1. Bagaimana pengaruh *header In-band Network Telemetry* terhadap jaringan ?
2. Bagaimana performansi *protocol overhead* pada *In-band Network Telemetry* ?
3. Bagaimana *hop latency* pada *In-band Network Telemetry* ?
4. Bagaimana performansi *storage overhead* pada *In-band Network Telemetry* ?

1.3 Tujuan dan Manfaat

Tujuan dari penelitian Tugas Akhir ini adalah melakukan simulasi dan analisis terhadap *in-band network telemetry* pada *programmable network infrastructure* yang berbasis *P4 language*. Adapun manfaat yang dapat diperoleh melalui penelitian Tugas Akhir ini, diantaranya adalah:

1. Mengetahui dampak *header In-band Network Telemetry* terhadap jaringan
2. Menganalisis *protocol overhead* pada *In-band Network Telemetry*
3. Menganalisis *hop latency* pada *In-band Network Telemetry*
4. Menganalisis *storage overhead* pada *In-band Network Telemetry*

1.4 Batasan Masalah

Tugas Akhir ini menerapkan beberapa batasan masalah sebagai berikut:

1. Tugas Akhir ini dilakukan dalam skala Laboratorium *virtual*.
2. Menggunakan *resource cloud server* untuk infrastruktur pengerjaan tugas akhir.
3. *SDN controller* yang digunakan adalah *Open Networking Operation System* (ONOS).

4. Simulasi menggunakan *emulator mininet*.
5. Simulasi menggunakan *Virtual BMv2 Switch* yang bukan digunakan untuk *production-grade software switch*.

1.5 Metode Penelitian

Metode penelitian yang digunakan pada Tugas Akhir ini adalah:

1. Studi Literatur
Mencari dan mempeleajari literatur tentang *SNMP streaming telemetry*, *software defined networking*, *in-band network telemetry*, *P4 Language* dan lain-lain dari jurnal, paper, artikel, dan video.
2. Analisis Masalah
Melakukan analisis kemungkinan masalah yang akan terjadi. Selanjutnya melakukan diskusi dengan dosen pembimbing terkait permasalahan yang akan dibahas pada Tugas Akhir.
3. Perancangan Sistem
Melakukan perancangan sistem yang akan dilakukan kemudian melakukan instalasi software yang digunakan.
4. Pengujian dan Analsis
Melakukan pengujian terhadap sistem yang telah dibuat kemudian dilakukan analisis dari hasil pengujian parameter yang telah dipakai pada sistem.
5. Penyusunan Laporan
Mencatat dan menyusun hasil dari penelitian untuk dilakukan pembukuan.