

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kemajuan teknologi virtualisasi, *cloud computing* merupakan salah satu teknologi utama di dalam internet, terutama dalam tingkat ketersediaan infrastruktur, yang tujuan utama dari teknologi virtualisasi adalah *user* dapat menjalankan layanan pada mesin fisik yang sama dan terisolasi satu sama lain [1].

Teknologi *container* merupakan teknologi baru yang dalam segi jaringan belum terlalu matang. Banyak solusi jaringan yang di usulkan untuk memecahkan masalah dalam jaringan pada *container*. Salah satu masalah terdapat pada *container* adalah komunikasi antara *container* dalam permasalahan isolasi trafik *container*. Solusi yang tepat untuk permasalahan ini sangat penting untuk efisiensi komunikasi antara *container* dan juga dalam meningkatkan kinerja *cloud data center*. Solusi untuk mengatasi permasalahan pada jaringan *container* adalah *flanel*[4], *Docker swarm overlay*[5][6], *calico*[7], SDN dan *protocol tunneling* dalam lingkup *cloud data center*[2].

Pada teknologi *overlay* untuk menghubungkan *container* melalui jaringan dapat menggunakan teknologi SDN atau *protocol tunneling*. Untuk memaksimalkan kinerja *container* dapat menjalankan *container* dengan layanan jaringan yang berbeda pada host yang sama, atau dapat menggunakan satu *container* per host dengan banyak layanan yang berbeda[3]. Untuk solusi yang bias di gunakan dengan menggunakan teknologi SDN dan *protocol tunneling* pada *overlay virtual network*.

Dalam teknologi virtual mempunyai *protocol tunneling* yang dapat digunakan dalam yaitu *Virtual Extensible Local Area Network (VXLAN)*, *Network Virtualization Generic Routing Encapsulation (NVGRE)*, *Stateless Transport Tunneling (STT)* dan GENEVE [8] yang disebut sebagai *Overlay Virtual network*. Penggunaan *protocol tunneling* dapat mengisolasi trafik hingga 16 juta *tunnel ID* dan memiliki fitur *multitenancy* yang menambah kemampuan fleksibilitas dari pengguna. *Protocol tunneling* memiliki kekurangan dalam penggunaan CPU dan efektifitas

penggunaan *bandwidth*, hal ini dikarenakan paket data yang dikirimkan akan mendapat tambahan proses enkapsul terlebih dahulu sehingga CPU perlu bekerja lebih ekstra dalam memproses paket data dan seiring dengan hal itu, ukuran paket akan bertambah besar [16].

Dalam tugas akhir ini akan melakukan analisis perbandingan kinerja 4 protokol tunneling, yaitu NVGRE, VXLAN, STT dan GENEVE yang bertujuan untuk melihat performansi protokol mana yang berjalan paling optimal untuk komunikasi antara *container*, dengan menggunakan parameter *throughput*, *delay*, *paket loss* dan jitter.

1.2 Tujuan Penelitian

Pada penelitian tugas akhir ini untuk menganalisis metode *tunnel* yang bekerja paling optimal pada komunikasi antara *container* docker dalam *overlay virtual network*, akan dilakukan pengujian dan analisa data. Baik dari QoS berupa *delay*, *jitter*, *throughput* dan *packet loss*.

1.3 Penelitian Terkait

Beberapa penelitian yang terkait pada Tugas Akhir ini antara lain adalah:

1. Analisa kinerja *protocol tunneling* pada *overlay virtual network*: Pada penelitian ini mengusulkan mengunakan *container* docker.
2. Analisa kinerja *protocol tunneling* pada *overlay virtual network*: pada penelitian ini meneliti lebih jauh tentang teknik traffic isolation khususnya pada sisi optimasi agar kinerja dari protokol *tunneling* ini dapat berkerja lebih baik dan efisien khususnya di sisi penggunaan QOS dan vCPU.

1.4 Rumusan Masalah

1. Apakah metode *tunneling* dapat mengisolasi trafik didalam lingkungan virtualisasi berbasis *container* docker?
2. Bagaimana kinerja dari protokol *tunneling* VXLAN, NVGRE, STT, GENEVE pada komunikasi antara *container*?
3. Protokol *tunneling* manakah yang performasinya paling optimal?
4. Apakah kinerja *protocol tunneling* lebih baik dari pada *non-tunneling(native)*

5. Apakah performansi *Container* lebih baik dari *Virtual Machine* untuk penggunaan protokol tunneling

1.5 Batasan Masalah

Untuk memperjelas ruang lingkup pembahasan yang akan dikerjakan pada tugas akhir ini maka diterapkan batasan masalah, antara lain:

1. Menggunakan perangkat keras laptop sebagai infrastruktur jaringan.
2. Menggunakan sistem operasi Ubuntu 16.04 LTS.
3. Menggunakan *linux ovs tree* pada *software Open vSwitch 2.7.1* untuk komunikasi *tunneling* pada *container*.
4. Pengambilan dan analisa paket menggunakan *tolls Iperf3,D-ITG,Netperf* dan *Wireshark*.
5. Menggunakan aplikasi *platform Docker 17.01* sebagai virtualisasi *container*.

1.6 Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah :

1. Studi Literatur
Melakukan pendalaman materi terkait topik yang di ambil sebagai judul tugas akhir
2. Perancangan
Melakukan perancangan topologi sistem dan konfigurasi.
3. Simulasi
Mensimulasikan program dan mengoptimalisasi agar hasil yang didapat sesuai dengan spesifikasi yang diinginkan.
4. Pengujian dan Analisis
Melakukan pengujian terhadap parameter-parameter othentikasi tunneling dan Qos. Pegujian Qos seperti: throughput, packet loss, dan delay *jitter* sesuai skenario yang ditentukan.
5. Pembuatan Laporan
Segala proses yang dilakukan pada pengerjaan tugas akhir ini akan dibukukan.

1.7 Sistematika Penulisan

Sistematika pada penulisan laporan Tugas Akhir ini, antaran lain:

✓ BAB 1 PENDAHULUAN

Pada bab ini Berisi latar belakang, rumusan masalah, tujuan, metode penelitian, dan sistematika penulisan.

✓ BAB 2 DASAR TEORI

Pada bab ini Berisi tentang teori yang berkaitan dengan penelitian yang dilakukan , infrastruktur/hardware yang digunakan, dan software/aplikasi yang digunakan.

✓ BAB 3 PEMODELAN SISTEM DAN SIMULASI

Pada bab ini menjelaskan alur kerja, alur perancangan program dan menjelaskan konfigurasi untuk merancang tugas akhir serta sekenario pengujian yang dilakukan.

✓ BAB 4 SIMULASI, PENGUJIAN DAN ANALISIS

Pada bab ini menjelaska langkah yang dilakukan untuk melakukan simulasi , pengujian, hasil pengujian yang didapat dari pengujian, dan analisis dari hasil pengujian yang didapat.

✓ BAB 5 KESIMPULAN DAN SARAN

Berisi kesimpulan dari tugas akhir ini dan saran untuk penelitian selanjutnya