BAB I
PENDAHULUAN

1.1 Latar Belakang

Pada dewasa ini teknologi *cloud computing* dan internet berkembang sangat cepat. Ditandai dengan penggunaan layanan internet menjadi sangat banyak, sehingga tidak cukup hanya menggunakan server fisik sebagai penyedia layanan. Untuk mengatasi hal tersebut maka dikembangkan teknologi baru seperti virtualisasi dan *container*[1].

Virtualisasi merupakan teknologi yang memungkinkan satu perangkat komputer dapat menjalankan lebih dari satu *operating system* secara bersamaan[2]. Tujuan dari virtualisasi agar dapat memaksimalkan sumber daya pada komputer. Perkembangan dari teknologi virtualisasi adalah teknologi *container*. *Container* secara garis besar mirip seperti virtualisasi. Namun pada *container* menggunakan *kernel host* dibawahnya atau yang disebut *shared kernel*[3]. Ini menyebabkan *container* menjadi lebih ringan dalam penggunaan sumber daya dibandingkan virtualisasi. Karena penggunaan sumber daya pada *container* lebih kecil dibandingkan virtualisasi, maka jumlah *container* akan lebih banyak dibandingkan virtualisasi[3]. *Container* yang berjumlah banyak membutuhkan aplikasi tambahan untuk mengatur *container* atau biasa disebut *container orchestration*. Terdapat beberapa aplikasi *container orchestration* yaitu Docker Swarm, Apache Mesos, dan yang terkenal adalah Kubernetes[4].

Pada kubernetes setidaknya membutuhkan dua *node server* yang digunakan sebagai *Master Node* dan *Worker Node*. Kubernetes tidak menyediakan komponen jaringan didalamnya, namun hanya menyediakan model jaringannya saja. Sehingga konfigurasi jaringan pada Kubernetes diatur oleh aplikasi tambahan yang disebut *Container Network Interface* (CNI). Beberapa CNI yang direkomendasikan oleh *Cloud Native Computing Foundation* (CNCF) adalah flannel dan calico [5]. Namun pada flannel dan calico belum dapat memantau trafik paket secara jelas. Serta per-

forma yang dihasilkan masih belum bisa maksimal, karena pemilihan protokol *tunneling* yang kurang tepat[1], dan menggunakan *interface* yang sama untuk komunikasi *internal* dan *external* [4]. Sehingga diharapkan dengan menggunakan konsep SDN pada CNI dapat memberikan performa yang maksimal pada jaringan. Karena terdapat *controller* yang mengatur keseluruhan jaringan. Serta dengan dipisahnya *interface* yang digunakan untuk akses dari luar diharapkan juga akan meningkatkan performa jaringan pada kubernetes.

Dalam tugas akhir ini akan menggunakan *SDN Controller ONOS* dengan proyek *SONA-CNI* Kubernetes. Karena pada *SONA-CNI* terdapat pemisahan pemrosesan jaringan secara *logical*, menggunakan konsep SDN pada pengaturan jaringannya, dan memisahkan *network interface* yang digunakan untuk manajemen dan pengaksesan dari luar. Serta memiliki beberapa kelebihan yaitu protokol *tunneling* yang bervariasi serta dapat diubah, dan dapat memantau trafik paket pada jaringan.

1.2 Rumusan Masalah

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

1. Bagaimana integrasi SONA-CNI pada *container orchestration* kubernetes.
2. Parameter apa saja yang digunakan dalam mengukur kinerja CNI di jaringan kubernetes.
3. Bagaimana cara memonitoring trafik paket pada jaringan *kubernetes* yang menggunakan SONA-CNI.
4. Protokol *tunneling* apa saja yang dapat digunakan pada CNI di kubernetes.

1.3 Tujuan dan Manfaat

Tujuan dari penelitian ini adalah agar dapat mengetahui performansi ONOS SONA-CNI pada jaringan kubernetes. Dengan cara membandingkan SONA-CNI dari segi protokol *tunneling* dan *interface network* yang digunakan, dengan CNI yang direkomendasikan oleh CNCF yaitu Flannel dan Calico.

Manfaat dari penelitian ini agar dapat memilih CNI terbaik yang dapat digunakan pada jaringan Kubernetes. Dengan menentukan protokol tunneling tepat untuk komunikasi antar *pod* yang berbeda *host*.

1.4 Batasan Masalah

Adapun batasan masalah yang dibuat dalam tugas akhir ini adalah

1. Trafik yang diamati hanya pada *pod* dan *host* Kubernetes.
2. Protokol *tunneling* yang digunakan pada ONOS-CNI hanya *VxLAN, GRE* dan *GENEVE*.
3. Membandingkan CNI Flannel dan Calico dengan *ONOS SONA-CNI*.
4. Tidak membahas tentang keamanan jaringan pada jaringan Kubernetes.
5. Hanya Menggunakan *Maximum Transmission Unit* (MTU) bawaan *host* dan bawaan CNI.
6. Hanya membahas layanan jaringan pada Kubernetes yaitu CNI.

1.5 Metode Penelitian

Berikut merupakan poin-poin metodologi penelitian:

1. Studi Literatur
   Studi Literatur bertujuan untuk mempelajari teori-teori mengenai konsep CNI pada jaringan kubernetes.
2. Analisa Masalah
   Melakukan analisa mengenai masalah yang dihadapi dan berdiskusi dengan dosen pembimbing.
3. Perancangan
   Melakukan perancangan topologi jaringan yang akan dibuat dan skenario pengujian yang akan digunakan.

4. Simulasi dan Analisis

   Melakukan simulasi terhadap topologi dan skenario pengujian yang telah dibuat serta melakukan analisis terhadap performansi jaringan.

5. Penyusunan Laporan

   Mencatat dan menyusun laporan hasil penelitian untuk dijadikan sebagai laporan tugas akhir.