BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kemajuan teknologi informasi menyebabkan meningkatnya kebutuhan manusia akan permintaan komunikasi jaringan internet. Sebagian besar pengguna internet saat ini hanya peduli dengan konten yang diinginkan saja tanpa mengetahui dari mana konten tersebut berasal, oleh karena itu diperlukan adanya layanan sistem komunikasi yang lebih efisien dan handal [1]. Arsitektur internet yang banyak digunakan saat ini merupakan model komunikasi *host-to-host* dimana proses pengiriman paket berdasarkan pada alamat sumber dan tujuan, sehingga menyebabkan banyak lalu lintas data yang terjadi. Untuk mengatasi permasalahan ini, maka dibuatlah suatu konsep baru yang mampu menggantikan komunikasi yang berpusat pada alamat sumber dan tujuan menjadi berpusat pada pengidentifikasian konten. Konsep ini dinamakan *Named Data Network* (NDN) [2].

Pada NDN terdapat *cache* yang berisi data atau konten yang disimpan pada *node* NDN dan memungkinkan untuk dapat digunakan kembali pada saat konsumen membutuhkannya [3]. Dalam proses *caching* terdapat beberapa teknik dalam melakukan optimasi suatu *node* NDN, salah satunya adalah *Replacement Algorithm*. Teknik ini akan memutuskan konten mana yang harus dikeluarkan atau dihapus pada saat penyimpanan *cache* penuh. Ada beberapa metode berdasarkan *replacement algorithm*, yaitu *Least Recently Used* (LRU), *Least Frequently Used* (LFU), dan *First In First Out* (FIFO) [4]. Dalam NDN, konsumen akan meminta data bukan lagi berdasarkan alamat sumber melainkan berupa konten langsung pada jaringan NDN. Apabila suatu *node* NDN memiliki konten yang diminta, maka *node* tersebut akan mengirimkan konten, sehingga tidak perlu meminta konten lagi kepada produsen. Oleh karena itu NDN memiliki dua komponen penting yaitu *node* atau router NDN dan konten yang saling berkaitan satu sama lain.

Pada penelitian sebelumnya yang dilakukan oleh Christos Natsis, Christos-Alexandros Sarros, dan Vassilis Tsaoussidis, melakukan penelitian terhadap perubahan konfigurasi NDN dengan cara membandingkan *total volume* dan ukuran *chunk* data pada NDN dengan beberapa *metric* penelitian [5]. Semakin sedikit jumlah *chunk* pada suatu *node* maka semakin sedikit jumlah interest yang dikirimkan oleh konsumen. Jumlah *chunk* yang lebih banyak akan menyebabkan lebih banyak pula entri pada PIT. Tetapi pada penelitian tersebut tidak membahas parameter yang berfungsi untuk mengetahui kemampuan dari kedua komponen penting pada NDN tersebut. Oleh karena itu, pada Tugas Akhir ini, penulis melakukan penelitian mengenai pengaruh perubahan jumlah konten dan *node* NDN pada *cache replacement* LRU berdasarkan parameter *Cache Hit Ratio* (CHR) dan *Round Trip Time* (RTT). Penelitian dilakukan menggunakan perangkat lunak MiniNDN yang dapat membantu selama proses simulasi dan memungkinkan untuk setiap penggunanya dapat merancang dan memodifikasi metode NDN.

1.2 Rumusan Masalah

Berdasarkan latar belakang, dapat dirumuskan beberapa detail permasalahan, yaitu:

1. Bagaimana cara mengimplementasikan berbagai jumlah konten dan *node* NDN pada perangkat lunak MiniNDN?
2. Bagaimana cara mengolah keluaran simulasi agar mendapatkan hasil analisis sesuai dengan sistem yang dibuat?
3. Bagaimana pengaruh dari perubahan jumlah konten dan *node* pada *virtual node* NDN?
4. Bagaimana cara memaksimalkan nilai CHR dan RTT yang akan dihasilkan?

1.3 Tujuan dan Manfaat

Tujuan dan Manfaat dari penulisan tugas akhir ini, yaitu :

1. Mengetahui implementasi berbagai jumlah konten dan *node* NDN pada perangkat lunak MiniNDN.

2. Mengetahui hasil dan cara mengolah keluaran simulasi sesuai dengan sistem yang dibuat.
3. Mengetahui pengaruh dari perubahan jumlah konten dan *node* pada *virtual node* NDN.
4. Mengetahui cara memaksimalkan nilai CHR dan RTT yang akan dihasilkan.

1.4 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah sebagai berikut :

1. Menggunakan perangkat lunak *VirtualBox* sebagai mesin virtual untuk sistem operasi Linux.
2. Simulasi perubahan jumlah konten dan *node* dilakukan dengan perangkat lunak MiniNDN.
3. *Cache replacement* yang digunakan adalah *Least Recently Used* (LRU).
4. Parameter analisis hanya berdasarkan *Cache Hit Ratio* (CHR) dan *Round Trip Time* (RTT).
5. Hanya menggunakan *dataset* hasil keluaran dari MiniNDN.

1.5 Metode Penelitian

Metode penelitian yang digunakan dalam penyusunan tugas akhir ini, yaitu :

1. Studi Literatur

Tahap ini meliputi pengumpulan informasi yang diperlukan dalam penyelesaian tugas akhir ini mengenai skenario pada *dataset* NDN dan teknik perubahan konfigurasi serta penerapannya dalam berbagai algoritma *caching* NDN. Informasi didapatkan dari beberapa jurnal ilmiah, artikel, dan *e-book* yang berkaitan dengan penelitian.

2. Analisis Kebutuhan

Tahap ini meliputi pengumpulan informasi terkait kebutuhan yang diperlukan dalam penelitian dan simulasi berdasarkan penerapan dari perubahan jumlah konten pada *dataset* NDN sesuai dengan studi literatur.

3. Perancangan Sistem

Tahap ini meliputi perancangan skenario yang akan digunakan selama penelitian sesuai dengan hasil tahap analisis kebutuhan. Perancangan yang dilakukan berupa pembuatan skema penelitian sesuai dengan proses simulasi pada perangkat lunak MiniNDN.

4. Implementasi Sistem

Tahap ini meliputi pembuatan sistem yang akan digunakan pada tahap simulasi berdasarkan pada tahap perancangan sistem. *Tools* yang digunakan untuk penelitian yaitu perangkat lunak MiniNDN.

5. Simulasi Sistem

Tahap ini meliputi pelaksanaan simulasi berdasarkan pada tahap yang sudah dilakukan sebelumnya untuk memperoleh keluaran sesuai dengan parameter-parameter yang diinginkan.

6. Analisis Sistem

Tahap ini meliputi pengumpulan hasil simulasi sistem yang kemudian akan dilakukan analisis terharap semua parameter yang telah ditentukan sebelumnya. Tahap ini juga dilakukan pengambilan kesimpulan dari semua tahap mengacu pada hasil penelitian yang didapatkan.

7. Penyusunan Laporan

Tahap ini meliputi proses penyusunan laporan serta dokumentasi selama pelaksanaan penelitian terkait pengaruh perubahan jumlah konten dan *node* pada *cache replacement* LRU.