

# Analisis Pengaruh Jumlah Konten Dan Node Dengan Cache Replacement Lru Pada Virtual Node NDN

## *Analysis The Number Of Contents And Nodes With Lru Cache Replacement On NDN Virtual Nodes*

1<sup>st</sup> Muhammad Nandaliyan Rais  
Al Azizi  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
nandaliyanrais@student.telkomuni-  
versity.ac.id

2<sup>nd</sup> Ridha Muldina Negara  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
ridhanegara@tass.telkomuniversit-  
y.ac.id

3<sup>rd</sup> Leanna Vidya Yovita  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
leanna@tass.telkomuniversity.ac.i  
d

### Abstrak

*Named Data Network (NDN)* adalah arsitektur jaringan baru yang mampu menggantikan model komunikasi yang bersifat *host-centric* menjadi *data-centric*. NDN memiliki sebuah penyimpanan sementara yang mampu menyimpan salinan data yaitu *cache*. Untuk mendukung proses penyimpanan *cache* terdapat beberapa teknik *caching*, salah satunya adalah *cache replacement* yang merupakan teknik penggantian konten apabila penyimpanan *cache* telah penuh. Dalam NDN terdapat dua jenis paket yang mewakili proses pengiriman konten, yaitu *interest packet* dan *data packet*. Pada Tugas Akhir ini dilakukan penelitian tentang pengaruh dari perubahan konten dan *node* dengan *cache replacement LRU* pada *virtual node NDN*. Penelitian dilakukan dengan melakukan perubahan jumlah konten dan *node* secara berkala yang selanjutnya dilakukan analisis berdasarkan parameter *Cache Hit Ratio (CHR)* dan *Round Trip Time (RTT)*. Hasil analisis berdasarkan beberapa skenario perubahan jumlah konten dan *node* secara berkala terbukti dapat mempengaruhi nilai dari CHR dan RTT yang dihasilkan. Jumlah konten yang semakin banyak memberi dampak penurunan nilai CHR dan peningkatan nilai RTT. Sedangkan jumlah *node* yang semakin banyak memberi dampak peningkatan nilai RTT tetapi tidak ada dampak yang terlalu signifikan terhadap nilai CHR yang dihasilkan.

**Kata kunci :** NDN, konten, *node*, LRU, *Cache Hit Ratio*, *Round Trip Time*.

### Abstract

*Named Data Network (NDN)* is a new network architecture that is able to replace the communication model that is *host-centric* to *data-*

*centric*. NDN has a temporary storage capable of storing copies of data called *cache*. To support the *cache storage process*, there are several *caching techniques*, one of them is *cache replacement*, which is a *content replacement technique* when the *cache storage is full*. In this final project, research is conducted on the effect of changing content and nodes with LRU cache replacement on NDN virtual nodes. The research was conducted by periodically changing the number of content and nodes which were then analyzed based on the *Cache Hit Ratio (CHR)* and *Round Trip Time (RTT)* parameters. The results of the analysis based on several scenarios of changes in the number of content and nodes periodically proved to be able to affect the value of the resulting CHR and RTT. The increasing amount of content has the effect of decreasing the CHR value and increasing the RTT value. Meanwhile, the increasing number of nodes has an impact on increasing the RTT value but there is no significant impact on the resulting CHR value.

**Keywords:** NDN, contents, nodes, LRU, *Cache Hit Ratio*, *Round Trip Time*.

## I. PENDAHULUAN

. 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 Networking* (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 penggunaanya dapat merancang dan memodifikasi metode NDN.

## II. KAJIAN TEORI

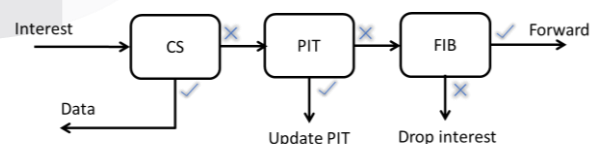
### 2.1 Named Data Network (NDN)

*Named Data Network* (NDN) merupakan konsep arsitektur jaringan baru yang dirancang untuk mengatasi permasalahan yang ada pada arsitektur jaringan internet saat ini. NDN mengganti arsitektur jaringan *host-centric* (IP) menjadi *data-centric* sehingga paket yang dikirimkan bukan lagi berupa alamat IP sumber dan tujuan, melainkan

permintaan data atau konten. Terdapat dua jenis paket dalam NDN, yaitu *interest packet* atau paket permintaan dan *data packet* atau paket konten.

Ketika konsumen mengirimkan *interest packet*, router NDN akan mengingat *interface* asal permintaan datang. Kemudian meneruskan *interest packet* dengan mencari data *node* produsen pada *Forwarding Information Base* (FIB) yang berisi protokol *routing* berbasis nama. Setelah *interest packet* mencapai *node* produsen, selanjutnya produsen akan mengirimkan *data packet* berisi konten yang diminta kepada konsumen. Router kemudian menyimpan informasi *interest* pada *Pending Interest Table* (PIT), dimana terdapat entri yang berisi nama dan *interface* sesuai dengan *interest packet* yang telah diterima. Ketika *data packet* tiba, router akan mencari entri yang sesuai dan meneruskan data ke semua *interface* yang telah terdaftar dalam entri tersebut. Router kemudian menghapus entri PIT yang telah digunakan dan menyimpan salinan *data packet* berupa *cache* ke dalam *Content Store* (CS) [2].

Proses *forward* data pada NDN dimulai ketika konsumen meminta konten berupa *interest packet* ke produsen. Router NDN yang menerima paket akan memeriksa apakah konten yang diminta tersedia didalam CS. Jika tersedia, maka router akan mengirimkan *data packet* berisi konten yang diminta langsung menuju konsumen. Dan jika tidak tersedia, router akan memeriksa pada PIT apakah konten tersebut pernah diminta dan belum dibalas dengan *data packet* yang diinginkan. Jika didalam PIT terdapat informasi tersebut, maka PIT akan menambahkan informasi tersebut kedalam entri nya dan akan membuat jalur yang akan mengirimkan *data packet* menuju konsumen. Dan Jika didalam PIT tidak ada konten yang diminta, maka selanjutnya router akan memeriksa pada FIB. *Interest packet* akan diteruskan ke *node* penyedia konten sesuai dengan informasi yang ada didalam FIB. Jika FIB tidak menyimpan informasi *node* penyedia konten, *interest packet* akan ditolak dan dibuang oleh router NDN [6]. Untuk lebih jelas dapat dilihat pada Gambar 1.



Gambar 1. Proses Forwarding Data pada NDN [6].

Apabila terdapat *interest packet* yang sama, maka data akan diambil dari *cache* terdekat dengan konsumen. Data yang di *cache* dekat dengan konsumen akan meningkatkan kinerja pengiriman paket dan mengurangi ketergantungan pada sumber data tertentu yang mungkin gagal karena kesalahan atau serangan. Sehingga suatu *node* NDN dapat berfungsi sebagai produsen maupun konsumen.

## 2.2 Caching

*Caching* merupakan suatu metode untuk menyimpan informasi di beberapa *end point* dan diharapkan dapat digunakan kembali kapanpun jika diperlukan. *Caching* dapat meningkatkan kecepatan transmisi data yang biasanya bersifat berulang. Selain itu *caching* dapat mengurangi waktu dan biaya selama proses transmisi data berlangsung [7]. *Caching* pada NDN memiliki *privacy* yang berbeda dengan IP. Pada IP, seseorang dapat memeriksa *header packet* dan *payload packet* dari paket yang dikirimkan sehingga menyebabkan mudah diketahui asal dan sumber dari data tersebut. Sedangkan NDN memfasilitasi pengamatan permintaan paket tetapi tanpa disertai alamat sumber dan tujuannya, sehingga NDN memiliki perlindungan *privacy* yang lebih baik dibandingkan dengan IP [3]. Didalam *caching* terdapat beberapa skenario yang akan membantu dalam penyimpanan *data packet*, antara lain *cache placement* dan *cache replacement*.

## 2.3 Cache Hit

*Cache hit* secara umum adalah suatu kondisi dimana suatu data dapat ditemukan dalam suatu *cache* atau penyimpanan [8]. Sehingga *cache hit* dalam NDN dapat diartikan sebagai suatu kondisi dimana sebuah *interest packet* mendapatkan balasan *data packet* yang diminta secara langsung melalui *node* terdekat.

## 2.4 Mini-NDN

MiniNDN adalah sebuah perangkat lunak gratis yang dapat digunakan untuk emulasi jaringan NDN [9]. Perangkat lunak ini memungkinkan penggunaannya dapat melakukan pengujian, eksperimen, dan penelitian terkait *Named Data Network* (NDN). Pada dasarnya MiniNDN merupakan perkembangan dari perangkat lunak Mininet.

## 2.5 Cache Hit Ratio

CHR adalah persentase jumlah permintaan *interest packet* yang berhasil dilayani oleh router atau *node* NDN tanpa perlu meminta *data packet* dari produsen [10]. Berikut persamaan untuk menghitung CHR :

$$\text{CHR} = \frac{\text{interest sent client} - \text{interest received server}}{\text{interest sent client}} \times 100 \quad (1)$$

## 2.6 Round Trip Time

RTT adalah waktu yang diperlukan suatu permintaan data yang dikirimkan oleh konsumen hingga mendapatkan balasan dari produsen data tersebut [11]. Biasanya menunjukkan lama waktu pengiriman data dari satu *node* ke *node* lainnya.

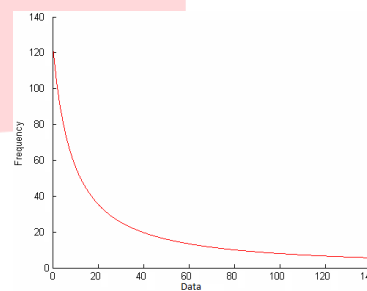
## 2.7 Zipf Mandelbrot

Dalam teori probabilitas dan statistik, *Zipf Mandelbrot* adalah sebuah probabilitas diskrit yang biasa digunakan untuk penentuan peringkat data

[12]. Berikut persamaan untuk menghitung nilai *Zipf Mandelbrot* :

$$f(k; N, q, s) = \frac{1/(k+q)^s}{\sum_{i=1}^N \frac{1}{(i+q)^s}} \quad (2)$$

Dimana  $k$  adalah peringkat suatu data, dan  $q$  dan  $s$  adalah parameter distribusinya. Dalam NDN, *Zipf Mandelbrot* dapat digunakan sebagai frekuensi suatu pola permintaan konten yang diminta oleh konsumen. Dimana pada konten pertama akan memiliki frekuensi terbanyak dan konten selanjutnya akan memiliki frekuensi yang semakin menurun. Gambar 2.2 merupakan bentuk pola distribusi *Zipf Mandelbrot* jika digambarkan pada sebuah kurva.

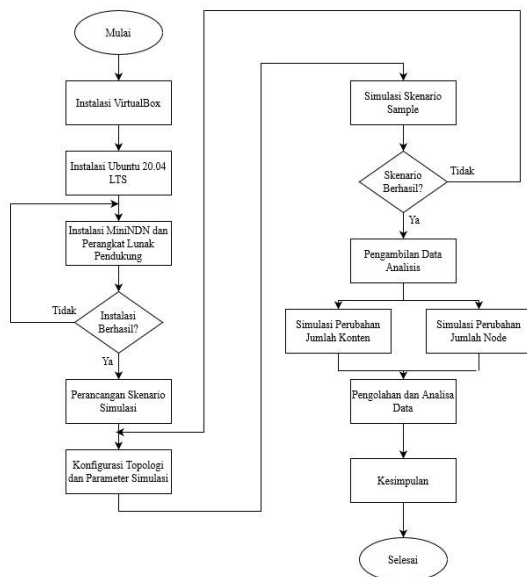


Gambar 2. Kurva Distribusi *Zipf Mandelbrot*

## III. METODE

### 3.1 Gagasan Pendekatan

Sistem penelitian yang akan dibangun dalam Tugas Akhir ini adalah pemodelan *virtual node* NDN untuk memperoleh serta mengetahui pengaruh dari perubahan konten dan *node* pada *cache replacement* LRU. Simulasi model sistem akan dilakukan menggunakan perangkat lunak MiniNDN. Dari model tersebut akan didapatkan manakah konfigurasi model terbaik dalam memaksimalkan kemampuan NDN terutama dalam *cache hit ratio* (CHR) dan *round trip time* (RTT) pada *packetnya*. Simulasi akan dilakukan dalam dua tahap penelitian, antara lain analisis perubahan *node* dan analisis perubahan konten.



Gambar 3. Flowchart Proses Perancangan Sistem

### 3.2 Perangkat Simulasi

Dalam melakukan simulasi sistem, penulis menggunakan beberapa komponen, baik itu perangkat keras maupun perangkat lunak. Penelitian ini dilakukan menggunakan laptop dengan sistem operasi Windows 10 64 bit. Tabel 1. menunjukkan spesifikasi lengkap laptop yang digunakan dalam penelitian ini. Simulasi NDN dijalankan pada sistem operasi Linux, sehingga diperlukan sebuah *virtual machine* untuk menjalankan sistem operasi tersebut. VM yang digunakan pada penelitian ini adalah VirtualBox dengan sistem operasi Linux Ubuntu 20.04 LTS. Berikut spesifikasi komponen simulasi yang digunakan.

Tabel 1. Spesifikasi Perangkat Keras

<i>System Model</i>	Acer Aspire E1-470
<i>Processor</i>	Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz
<i>Storage</i>	SSD 250 GB
<i>RAM</i>	8 GB
<i>Operating System</i>	Windows 10 64 bit

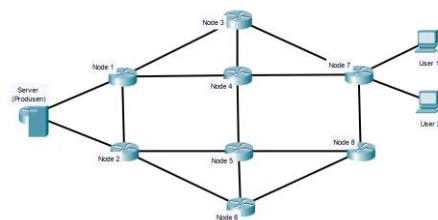
### 3.3 Desain Sistem

Penelitian ini dilakukan untuk mendapatkan analisis terkait pengaruh dari perubahan konten dan *node* untuk memaksimalkan kemampuan NDN. Sistem penelitian dibangun dengan menggunakan perangkat lunak MiniNDN yang selanjutnya hasil keluaran akan dianalisis berdasarkan parameter *cache hit ratio* (CHR) dan *round trip time* (RTT). Penelitian Tugas Akhir ini dibagi menjadi dua tahap, yaitu analisis perubahan konten dan analisis perubahan *node*. Simulasi menggunakan dua user sebagai konsumen yang akan mengirimkan *interest packet* kepada produsen. Penggunaan dua buah user bertujuan untuk mengetahui perbedaan performansi

yang dihasilkan. User 1 menggunakan pola permintaan *interest packet* yang diatur secara manual dimana setiap konten memiliki frekuensi permintaan yang sama. Sedangkan User 2 memanfaatkan *Zipf Mandelbrot* sebagai pola distribusi *interest packemya*.

#### 3.3.1 Simulasi Perubahan Konten

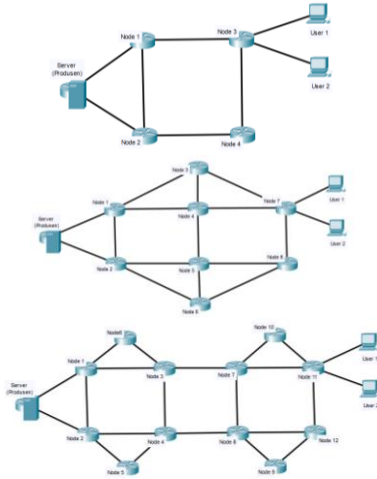
Simulasi perubahan konten dibagi menjadi tiga skenario, yaitu perubahan jumlah konten yang tersedia pada produsen, perubahan jumlah *interest packet* yang dikirimkan oleh user, dan perubahan ukuran *content store* pada *node*. Topologi yang digunakan merupakan topologi custom dimana terdapat 1 produsen, 8 *node* NDN, dan 2 konsumen. Skenario perubahan jumlah konten dilakukan dengan cara merubah jumlah konten yang tersedia pada produsen secara berkala, jumlah konten yang digunakan adalah 10, 30, dan 50 konten. Skenario perubahan jumlah *interest* dilakukan dengan cara merubah jumlah *interest packet* yang dikirimkan oleh user, jumlah *interest packet* yang digunakan adalah 100, 200, dan 300 interest. Dan scenario perubahan ukuran content store dilakukan dengan cara merubah ukuran *content store* pada setiap *node* NDN, ukuran CS yang digunakan adalah 10, 20, dan 30 konten. Berikut gambar topologi yang digunakan dalam simulasi perubahan konten.



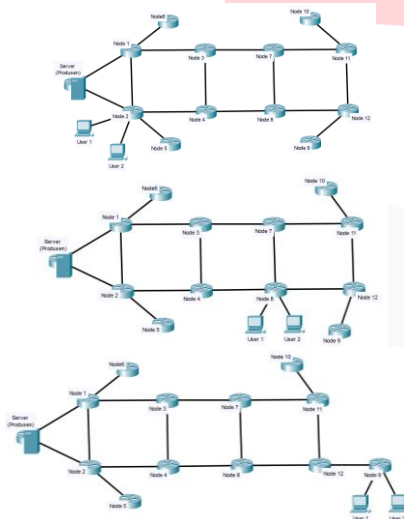
Gambar 4. Topologi Simulasi Perubahan Konten

#### 3.3.2 Simulasi Perubahan Node

Simulasi perubahan *node* dibagi menjadi dua skenario, yaitu perubahan jumlah *node* yang ada pada jaringan dan perubahan letak user atau jumlah *hop* dari user ke produsen. Dimana *node* tersebut terhubung pada satu produsen dan memiliki dua konsumen yang akan meminta *interest packet*. Skenario perubahan jumlah *node* dilakukan dengan cara merubah jumlah *node* pada topologi yang digunakan, jumlah *node* yang digunakan adalah 4, 8, dan 12 *node*. Sedangkan skenario perubahan jumlah *hop* dilakukan dengan cara merubah letak user atau jumlah *hop* dari user menuju produsen pada *node* dalam topologi yang digunakan, jumlah *hop* yang digunakan adalah 1, 3, dan 5 hop. Berikut beberapa bentuk konfigurasi topologi yang digunakan pada simulasi.



Gambar 5. Bentuk Topologi Skenario Perubahan Jumlah Node



Gambar 6. Bentuk Topologi Skenario Perubahan Jumlah Hop

IV. HASIL DAN PEMBAHASAN

Pengukuran berdasarkan CHR bertujuan untuk mengetahui kemampuan *node-node* pada jaringan NDN dalam memberikan jawaban dari *interest packet* yang diminta oleh konsumen. Sedangkan RTT digunakan untuk mengetahui waktu rata-rata yang diperlukan suatu data/konten selama proses pengiriman data dari *node* ke *node*. Performansi terbaik dapat dilihat dari nilai CHR yang tinggi dan nilai RTT yang rendah. CHR yang tinggi membuktikan bahwa *node* NDN dapat berfungsi dengan baik karena dapat memberikan konten yang diminta tanpa melakukan permintaan kepada produsen. Dan nilai RTT yang rendah membuktikan bahwa NDN berjalan dengan baik karena hanya

membutuhkan waktu yang singkat dalam pengiriman paketnya.

4.1 Hasil Simulasi

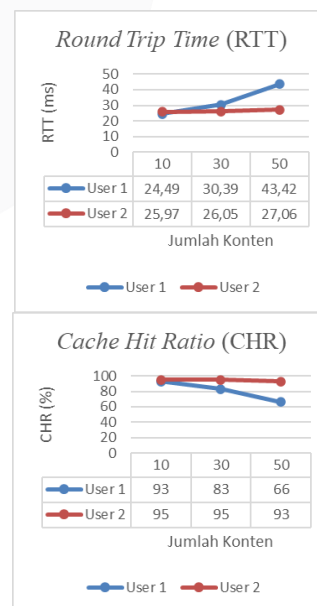
Hasil dari simulasi yang telah dilakukan berdasarkan beberapa skenario menunjukkan bahwa perubahan konten dan *node* dapat mempengaruhi nilai dari parameter CHR dan RTT yang dihasilkan. Simulasi dilakukan untuk mendapatkan analisis konfigurasi terbaik yang diharapkan mampu memaksimalkan nilai dari parameter CHR dan RTT yang akan dihasilkan selanjutnya.

4.1.1 Hasil Simulasi Perubahan Konten

Simulasi dilakukan menggunakan topologi jaringan yang sama yaitu terdapat 1 produsen, 8 *node* NDN, dan 2 konsumen. Konfigurasi yang tidak tercantum akan disesuaikan dengan nilai *default* pada MiniNDN. Dari hasil beberapa skenario perubahan konten, terbukti bahwa perubahan konten dapat mempengaruhi parameter CHR dan RTT yang akan dihasilkan.

4.1.1.1 Skenario Perubahan Jumlah Konten

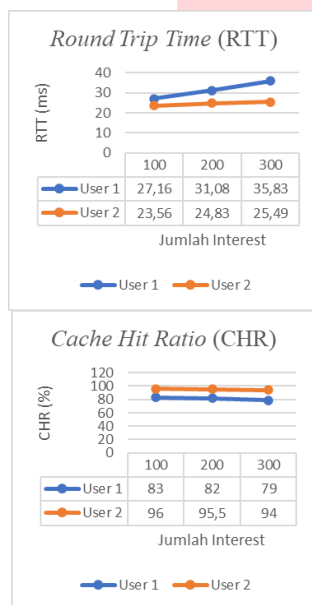
Berdasarkan simulasi yang telah dilakukan, perubahan jumlah konten pada produsen dapat mempengaruhi nilai CHR dan RTT yang akan dihasilkan. Jumlah konten yang besar akan menyebabkan CHR yang dihasilkan semakin rendah, sedangkan RTT yang dihasilkan semakin tinggi. Nilai CHR yang semakin rendah dapat terjadi karena jumlah konten yang banyak mengakibatkan konten yang tersimpan pada CS *node* NDN melebihi batas ukuran, sehingga ada beberapa konten yang tidak tersimpan dan harus meminta konten tersebut kepada produsen. Sedangkan nilai RTT yang semakin tinggi dapat terjadi karena jumlah konten yang banyak mengakibatkan NDN membutuhkan waktu yang lebih lama untuk mencari konten yang diminta oleh konsumen.



Gambar 7. Grafik Hasil Skenario Perubahan Jumlah Konten

#### 4.1.1.2 Skenario Perubahan Jumlah Interest

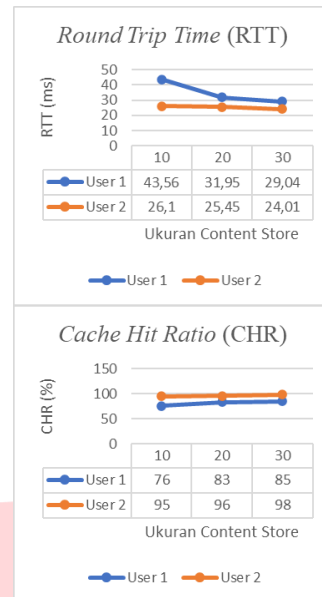
Seperti pada skenario sebelumnya, perubahan jumlah *interest packet* yang diminta oleh user juga mampu mempengaruhi nilai RTT dan CHR yang dihasilkan. Semakin besar jumlah *interest packet* yang diminta oleh user maka semakin tinggi RTT yang dihasilkan, hal ini dapat terjadi karena jumlah *interest* yang banyak mengakibatkan beban *node* semakin besar sehingga membutuhkan waktu yang lebih lama untuk melayani semua *interest* yang diminta. Sedangkan nilai CHR yang dihasilkan akan semakin menurun, hal ini dapat terjadi karena pada simulasi *cache replacement* yang digunakan adalah LRU, sehingga ada kemungkinan bahwa konten yang diminta telah dihapus pada CS *node* NDN yang terdekat dan perlu dikirimkan *interest* konten lagi kepada produsen.



Gambar 8. Grafik Hasil Skenario Perubahan Jumlah *Interest*

#### 4.1.1.3 Skenario Perubahan Ukuran Content Store

Perubahan ukuran *content store* yang ada pada setiap *node* jaringan NDN memiliki dampak yang positif jika dilihat dari nilai RTT dan CHR yang dihasilkan. Ukuran CS yang semakin besar akan meningkatkan nilai CHR dan menurunkan nilai RTT yang dihasilkan. Hal ini dapat terjadi karena ukuran CS yang besar dapat menyimpan lebih banyak konten pada *node* NDN, sehingga *interest* yang menuju pada produsen semakin berkurang dan waktu pengiriman *packet* semakin singkat.



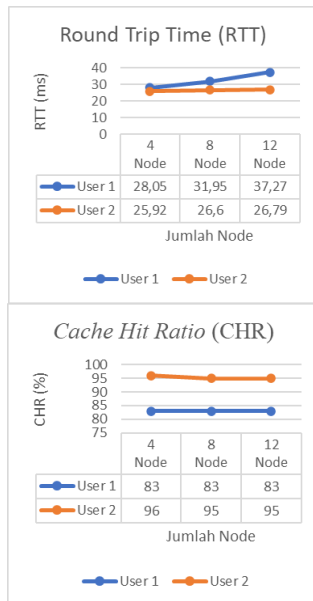
Gambar 9. Grafik Hasil Skenario Perubahan Ukuran *Content Store*

#### 4.1.2 Hasil Simulasi Perubahan Node

Simulasi dilakukan dengan cara merubah jumlah *node* dan jumlah *hop* secara berkala. Berdasarkan kedua skenario yang telah dilakukan, terbukti bahwa perubahan *node* mampu mempengaruhi nilai RTT tetapi tidak mempengaruhi nilai CHR yang akan dihasilkan. Nilai CHR cenderung tidak berubah karena *node* NDN dapat menyimpan konten yang pernah dikirimkan, sehingga selama *node* terdekat memiliki konten yang diminta oleh konsumen maka jumlah *node* yang semakin banyak tidak akan berpengaruh pada nilai CHR yang dihasilkan.

##### 4.1.2.1 Skenario Perubahan Jumlah Node

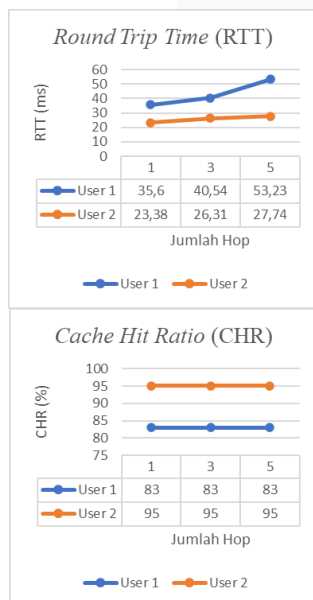
Jumlah *node* yang digunakan pada suatu topologi jaringan NDN, tidak terlalu mempengaruhi CHR yang dihasilkan. Perubahan jumlah *node* menyebabkan nilai RTT berubah, semakin banyak jumlah *node* pada suatu jaringan NDN, maka akan meningkatkan nilai RTT yang dihasilkan. Hal ini dapat terjadi karena jumlah *node* yang banyak menyebabkan konten tersimpan tersebar pada *node* yang banyak, sehingga membutuhkan waktu lebih lama untuk mencari konten yang diminta.



Gambar 10. Grafik Hasil Skenario Perubahan Jumlah Node

4.1.2.2 Skenario Perubahan Jumlah Hop

Perubahan letak user atau jumlah hop tidak terlalu banyak merubah CHR dan menyebabkan perubahan pada sisi RTT. Semakin banyak jumlah hop yang dilalui paket, maka semakin tinggi juga nilai RTT yang dihasilkan. Hal ini dapat terjadi karena jumlah hop yang semakin kecil berarti semakin dekat pula dengan produsen sehingga tidak membutuhkan banyak waktu untuk mendapatkan konten yang diminta.



Gambar 11. Grafik Hasil Skenario Perubahan Jumlah Hop

4.2 Analisis Simulasi

Dari hasil simulasi yang telah dilakukan, terbukti bahwa perubahan konten dan node dapat mempengaruhi nilai dari CHR dan RTT yang

dihasilkan. Jumlah konten yang semakin banyak memberi dampak penurunan nilai CHR dan peningkatan nilai RTT. Sedangkan jumlah node yang semakin banyak memberi dampak peningkatan nilai RTT tetapi tidak ada dampak yang terlalu signifikan terhadap nilai CHR yang dihasilkan. Penggunaan dua user/konsumen yang berbeda pola distribusi juga menunjukkan hasil yang berbeda pula.

Dapat dilihat bahwa secara keseluruhan User 2 dimana menggunakan pola distribusi Zipf Mandelbrot mendapatkan nilai RTT dan CHR yang lebih baik dan stabil jika dibandingkan dengan User 1, hal ini dapat terjadi karena pada penggunaan distribusi Zipf Mandelbrot, hanya ada beberapa konten saja yang memiliki frekuensi permintaan konten tinggi yang dapat menyebabkan hanya ada beberapa konten saja yang akan diminta oleh konsumen. Sehingga nilai CHR yang dihasilkan pada User 2 mendapatkan nilai yang tinggi. Dan nilai RTT yang dihasilkan User 2 juga mendapatkan nilai yang lebih baik. Hal ini dapat terjadi karena penggunaan frekuensi permintaan interest yang diatur sama rata pada User 1 menyebabkan banyak terjadi perubahan konten yang tersimpan pada node NDN ketika kapasitas penyimpanan CS telah penuh, sehingga harus meminta kembali konten kepada server yang dapat mengakibatkan waktu pengiriman konten semakin lama.

Tabel 2. Hasil Simulasi Peningkatan Nilai Skenario

Skenario Parameter	Jumlah Konten	Jumlah Interest	Ukuran CS	Jumlah Node	Letak User
CHR	Turun	Turun	Naik	-	-
RTT	Naik	Naik	Turun	Naik	Naik

Dari Tabel 2. dapat dilihat bahwa peningkatan jumlah konten dan interest mengakibatkan beberapa dampak terhadap nilai CHR dan RTT yang dihasilkan. Penggunaan cache replacement LRU menyebabkan konten-konten dengan frekuensi permintaan yang kecil akan dihapus pada node NDN ketika kapasitas CS node tersebut telah penuh. Hal ini menyebabkan sering terjadi pergantian konten yang tersimpan pada node NDN apabila memiliki jumlah permintaan interest packet yang banyak. Berbeda dengan peningkatan ukuran CS dimana mampu menghasilkan nilai CHR dan RTT yang lebih baik. Untuk memaksimalkan performansi NDN terutama dalam hal CHR dan RTT dapat dilihat dengan skenario yang memiliki nilai CHR yang tinggi dan nilai RTT yang rendah. Sehingga diperoleh cara meningkatkan dan memaksimalkan performansi tersebut dapat dilakukan dengan cara meningkatkan ukuran CS, mengurangi konten yang

tersedia, dan mengurangi jumlah *interest* yang dikirimkan oleh konsumen.

#### V. KESIMPULAN

Penggunaan pola distribusi *Zipf Mandelbrot* memiliki hasil RTT dan CHR yang lebih stabil jika dibandingkan dengan pola distribusi yang diatur secara manual. Teknik yang dapat dilakukan untuk meningkatkan nilai CHR, antara lain meningkatkan ukuran CS, mengurangi jumlah konten yang tersedia, dan mengurangi jumlah *interest*. Sedangkan teknik yang dapat dilakukan untuk mengurangi nilai RTT, antara lain mengurangi jumlah *node* pada jaringan, meningkatkan ukuran CS, mengurangi jumlah *interest*, dan mengurangi jumlah konten yang tersedia. Hal yang dapat dilakukan untuk memaksimalkan kemampuan NDN terutama dalam sisi CHR dan RTT adalah meningkatkan ukuran CS, mengurangi jumlah konten dan *interest*.

#### REFERENSI

- [1] T. Koponen *et al.*, "A data-oriented (and Beyond) network architecture," *Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, 2007, doi: 10.1145/1282427.1282402.
- [2] G. Bartolomeo and T. Kovacicova, "Named Data Networking Project," *Identif. Manag. Distrib. Data*, pp. 125–134, 2013, doi: 10.1201/b14966-17.
- [3] L. Zhang *et al.*, "Named data networking," *Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014, doi: 10.1145/2656877.2656887.
- [4] H. Khelifi, S. Luo, B. Nour, and H. Mounjla, "A QoS-aware cache replacement policy for vehicular named data networks," *2019 IEEE Glob. Commun. Conf. GLOBECOM 2019 - Proc.*, 2019, doi: 10.1109/GLOBECOM38437.2019.9013461.
- [5] C. Nasis, C. A. Sarros, and V. Tsaoussidis, "The Impact of Chunk Size on Named Data Networking Performance," *2020 3rd Int. Conf. Hot Information-Centric Networking, HotICN 2020*, no. February 2021, pp. 108–113, 2020, doi: 10.1109/HotICN50779.2020.9350754.
- [6] L. V. Yovita and N. R. Syambas, "Caching on Named Data Network: a Survey and Future Research," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 6, p. 4456, 2018, doi: 10.11591/ijece.v8i6.pp4456-4466.
- [7] M. C. C., "a Comparative Survey on Different Caching Mechanisms in Named Data Networking ( Ndn ) Architecture," vol. 6, no. April, pp. 264–271, 2017.
- [8] N. Beckmann, H. Chen, and A. Cidon, "LHD: Improving cache hit rate by maximizing hit density," *Proc. 15th USENIX Symp. Networked Syst. Des. Implementation, NSDI 2018*, pp. 389–403, 2018.
- [9] L. Wang, V. Lehman, A. K. M. Mahmudul Hoque, B. Zhang, Y. Yu, and L. Zhang, "A Secure Link State Routing Protocol for NDN," *IEEE Access*, vol. 6, pp. 10470–10482, 2018, doi: 10.1109/ACCESS.2017.2789330.
- [10] A. Basuki and E. S. Pramukantoro, "Desain dan Implementasi In-Network Caching Pada Content Centric Networking Menggunakan CCN-Lite Dengan Simulator OMNeT ++," vol. 2, no. 9, pp. 2770–2776, 2018.
- [11] S. Astuti, T. A. Wibowo, R. Mayasari, I. Asror, and G. P. Satriawan, "KLASIFIKASI DATA DELAY DENGAN LFDI STRATEGI FORWARDING MENGGUNAKAN MACHINE LEARNING UNTUK MEMAKSIMALKAN KINERJA JARINGAN NDN ( NAMED DATA NETWORK )," vol. 14, no. 2, pp. 115–122, 2020.
- [12] M. Tunnicliffe and G. Hunter, "Random Sampling of the Zipf-Mandelbrot Distribution as a Representation of Vocabulary Growth."