Optimalisasi *Devstack / Private Cloud Storage* Guna Meningkatkan

Performansi *Network Function Virtual*

Optimization of Devstack / Private Cloud Storage for Performance Improvement

of The Virtual Network Function

**Setiyo Budiyanto1, Kristiani N. Nahampun2, Freddy A. Silaban3, Lukman M. Silalahi4, Fajar R.5**

1,2,3,4Department of Electrical Engineering, Universitas Mercu Buana,

5Department of Electrical Engineering, Universitas Pembangunan Nasional “Veteran” Jakarta

1,2,3,4Jl Meruya Selatan No.1, Kec Kembangan, Jakarta Barat, Indonesia

5Jl. R.S Fatmawati No 1. Pondok Labu, Jakarta Selatan, Indonesia

sbudiyanto@mercubuana.ac.id1, kristianinatalia12@gmail.com2, freddy.artadima@mercubuana.ac.id3, lukman.medriavin@mercubuana.ac.id4, fajarrahayu@upnvj.ac.id5

***Abstrak –*** *Sistem Network Function Virtual (NFV) merupakan sebuah arsitektur jaringan yang menggunakan teknologi virtualisasi untuk seluruh fungsi platform jaringan dan fungsionalitas perangkat keras. Salah satu platform cloud computing yang dapat digunakan untuk membangun NFV adalah Devstack. Devstack merupakan sebuah tools untuk membangun layanan Private Cloud Storage; melalui sistem ini, layanan Private Cloud Storage dapat berjalan didalam sebuah jaringan virtual. Pada penelitian ini dilakukan implementasi jaringan virtual menggunakan devstack yang berbasis insfrastruktur NFV pada layanan Private Cloud Storage untuk mengukur kualitas jaringan Quality Of Service (QoS) : throughput, bit rate, jitter, dan delay. Melalui penelitian ini, dapat diketahui kemampuan suatu jaringan virtual pada suatu infrastruktur cloud computing.*

***Kata Kunci****:* Virtualisasi, Devstack, Network Function Virtualization, QoS, Cloud Storage

***Abstract*** *–  System Virtual Network Function (NFV) is a network architecture that uses virtualization technology for all network platform functions and hardware functionality. One of the cloud computing platforms that can be used to build NFV is Devstack. Devstack is a tool for building Private Cloud Storage services; through this system, the Private Cloud Storage service can run on a virtual network. In this study the implementation of virtual networks using devstack based on NFV infrastructure in the Private Cloud Storage service to measure network quality Quality Of Service (QoS) : throughput, bit rate, jitter, and delay. Through this research, it can be seen the ability of a virtual network in a cloud computing infrastructure.*

***Keywords****:* Virtualisasi, Devstack, Network Function Virtualization, QoS, Cloud Storage

1. **Pendahuluan**

Seiring perkembangan teknologi yang sangat cepat, sistem teknologi jaringan yang awalnya banyak menggunakan perangkat fisik bergeser menjadi perangkat virtual. Sistem virtualisasi merupakan sebuah solusi yang berguna untuk kemajuan teknologi dimana dengan sistem virtulisasi ini sebuah perangkat keras jaringan seperti router, switch dan lainnya dapat dibuat menjadi perangkat virtual. Virtualisasi adalah proses membuat suatu hal menggunakan software untuk merepresentasikan sesuatu, seperti aplikasi, server, storage, dan jaringan virtual. Virtualisasi merupakan salah satu cara yang cukup efektif untuk mengurangi biaya IT namun meningkatkan efisiensi dan fleksibilitas dalam melakukan pengembangan sistem jaringan pada suatu insfrastruktur [1, 2].

Pada penelitian ini dibahas Implementasi *Network Function Virtual* menggunakan *Devstack* pada *Private Cloud Storage.* Seperti yang kita ketahui dalam membangun sebuah insfrastruktur jaringan diperlukan sebuah perangkat keras , dimana setiap perangkat keras memiliki fungsi yang terbatas, sehingga dalam membangun sebuah infrastruktur jaringan diperlukan banyak perangkat keras dan konfigurasi jaringan yang kompleks pada setiap perangkat keras. Hal ini menyebabkan tidak efektif dan tidak efisien dalam membangun insfrastruktur jaringan. Oleh sebab itu dengan adanya sistem virtualisasi diharapkan dapat membantu dalam pembangunan insfrastruktur jaringan dimana sebuah perangkat keras dapat memiliki fungsi yang luas dalam pembangunan insfrastruktur jaringan, yaitu NFV [3].

Dalam penelitian ini dilakukan implementasi jaringan virtual menggunakan devstack yang berbasis pada infrastruktur NFV pada layanan *Private Cloud Storage* untuk mengukur kualitas jaringan seperti throughput, bitrate, jitter, dan delay. Sehingga dapat mengetahui kehandalan suatu jaringan virtual pada suatu infrastruktur *cloud computing*. Framework NFV adalah sebuah desain infrastruktur yang dikeluarkan oleh *(European Telecommunications Standards Institute)* ETSI*.* Framework NFV ini digunakan untuk menstandarisasikan desain arsitektur dari NFV. Melalui *framework* tersebut diimplementasikan NFV, hal tersebut bisa dijadikan sebagai solusi rancangan sistem yang akan dibuat. Tujuan adanya *framework* tersebut pada dasarnya akan membuat fungsi-fungsi yang ada di NFV menjadi modular. Pada *framework* tersebut terdapat beberapa bagian atau layer diantaranya ada *Network Function Virtualization Infrastruktur* (NFVI), *Virtualised Network Funtion* (VNF), *Element Management System* (EMS), dan *Operations and Bussiness Support System* (OSS/BSS). VNF, EMS, dan OSS/BSS itu semua berjalan di layer NFVI. NFVI dapat dibagi menjadi dua infrastruktur, yang pertama adalah *Physical infrastructur*e yaitu komputasi perangkat keras, penyimpanan perangkat keras , dan jaringan perangkat keras. Kemudian yang kedua adalah infrastruktur perangkat lunak, infrastruktur ini berfungsi untuk menjalankan lapisan virtualisasi menggunakan teknologi yang disebut dengan *hypervisor. Hypervisor* ini untuk menjalankan komputasi virtual (ESXi, KVM, QEMU), penyimpanan virtual (LVM, NFS), dan jaringan virtual (MPLS VPN, VRF).

1. **Metode Penelitian**

Pada bagian ini di bahas metode penelitian yang dilakukan pada setiap tahap dari penelitian yang dilakukan. Sebelum dijelaskan pada tahapan penelitian, dilakukan penjabaran terlebih dahulu terkait perangkat yang digunakan pada penelitian ini, Berikut perangkat keras juga perangkat lunak yang dimaksud:

a. *Nextcloud* merupakan perangkat lunak cloud yang memberi kendali penuh atas data yang dimiliki. Nextcloud dirancang untuk individu dan organisasi dengan banyak pengguna. Secara fungsional hampir sama dengan dropbox yang banyak digunakan, dengan perbedaan nextcloud bersifat bebas (*opensource*) dengan demikian memungkinkan seseorang untuk memasang dan mengoprasikannya tanpa biaya pada server pribadi. Nextcloud adalah aplikasi yang dikelola secara aktif dari ownCloud. Nextcloud merupakan perangkat lunak (software) opensource enterprise file sync dan share berbasis cloud [xxxxx]. Nextcloud mulai oleh inventaris sendiri oleh Frank Karlitschek, 12 wirausahawan dan insinyur opensource berpengalaman untuk memberdayakan pengguna agar dapat mengambil alih kendala atas data dan komunikasi mereka. Perusahaan ini diluncurkan pada tahun 2016 sebagai *spin-off* dari struktur AG, yaitu sebuah perusahaan konferensi web dan perangkat lunak perencanaan keuangan terkemuka sejak tahun 1995 yang melayani pelanggan seperti Deutsche Bank, Vodavone, BNP Paribas dan masih banyak lainnya. Nextcloud memberi control yang baik kepada organisasi terhadap akses data, memberkan fasilitas sinkronisasi file dan berbagi di seluruh perangkat, sehingga sangat memungkinkan kolaborasi di dalam dan di seluruh batasan organisasi [4].

b. *Cloud Storage* memiliki banyak keuntungan dibandingkan dengan penyimpanan data secara tradisional. Data yang disimpan pada *Cloud Storage* akan dapat diakses dimana saja serta kapanpun. *Cloud Storage* akan diintegrasikan ke berbagai perangkat untuk mendapatkan kemudahan pengaksesan seperti ke perangkat bergerak (*smartphone*), tablet serta personal komputer. Fitur unggulan lainnya yaitu tersedianya *file sharing* yang memudahkan untuk berbagi file dengan pihak lain. Hal ini akan sangat memudahkan misalkan saja terlibat dalam sebuah proyek kolaboratif untuk bertukar salinan file baik secara *online* maupun *offline.* Selain itu tingkat keamanan pada *Cloud Storage* sangat baik, hal tersebut dikarenakan *Cloud Storage* menggunakan sistem *Private Cloud. Private Cloud* merupakan sebuah mekanisme penyediaan resource IT secara menyeluruh, namun akan dikembangkan secara terpisah dari *public cloud* dan hanya bisa diakses melalui jaringan private, sehingga memiliki tingkat keamanan yang lebih baik. Model ini cocok untuk perusahaan dengan skala enterprise yang sangat memperhatikan privasi dan keamanan data [5].

c. *Cloud Computing* biasanya tersedia sebagai layanan kepada siapa saja di internet. Akan tetapi, varian yang disebut *Private Cloud* semakin popular untuk infrastruktur pribadi yang mempunyai *atribut Cloud* di atas. *Cloud Computing* berbeda dengan *Grid Computing* atau *Paralel Computing*, dimana *Grid Computing* dan *Paralel Computing* adalah lebih merupakan sebuah bagian dari prasarana fisik bagi penyediaan konsep Cloud Computing. Server konvensional akan di batasi oleh jumlah core processor, harddisk dan memory [6, 7]. Memperhatikan keterbatasan fisik yang ada maka kita tidak mungkin membebani sebuah server konvensional dengan beban maksimal. Jika sumber daya habis, maka biasanya kita harus melakukan install ulang seluruh aplikasi dan data di server yang kapasitasnya lebih besar dan memigrasi semua aplikasi yang ada ke server yang baru. Konsep NFV berawal dari para operator/telco yang mencari jalan untuk mempercepat implementasi layanan baru jaringan untuk mendukung strategi bisnis dan pertumbuhan pendapatan perusahaan. Salah-satu hambatan signifikan yang perusahaan-perusahaan rasakan adalah ketergantungan terhadap perangkat keras -*based appliance*. Pada konsep jaringan tradisional, setiap fungsi jaringan spesifik biasanya dilakukan oleh *appliance* dengan perangkat keras. Perangkat lunak dan perangkat keras dalam *applianc*e sengaja dibuat tidak bisa dipisahkan dan tergantung satu sama lain. *Quality of Service (QOS)* [2].

d. *Quality of Service* (QoS) atau Kualitas layanan adalah metode pengukuran yang digunakan untuk menentukan kemampuan sebuah jaringan seperti; aplikasi jaringan, host atau router dengan tujuan memberikan *network service* yang lebih baik dan terencana sehingga dapat memenuhi kebutuhan suatu layanan. *Quality of service* (QOS) merupakan suatu parameter uji kelayakan suatu layanan agar dapat dikatakan baik [6]. Beberapa parameter uji untuk Q*uality of Service* (QOS) antara lain:

1. Delay

Delay merupakan waktu yang dibutuhkan oleh suatu paket data untuk menempuh jarak saat dikirim hingga sampai ke tujuan dan melewati antrian pengiriman yang padat [1]. Nilai parameter untuk delay dapat menggunakan rekomendasi ITU T G 114 . Seperti yang ditunjukkan pada tabel 1

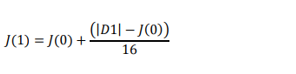
Tabel 1.Rekomendasi ITU T G 114 untuk Nilai Delay [8, 9]

|  |  |
| --- | --- |
| Nilai Delay | Kualitas |
| 0 | Sangat Baik |
| 1 - 150 ms | Baik |
| 151 - 400 ms | Cukup |
| > 401 ms | Buruk |

1. Jitter

Jitter merupakan variasi delay yang terjadi karena adanya selisih waktu kedatangan paket pada jaringan IP. Besar nilai jitter dapat dipengaruhi oleh beban trafik yang semakin besar maka nilai jitter akan semakin besar pula. Jika nilai jitter semakin besar maka kualitas jaringan akan semakin turun [1].

(0,1)=(𝑅1−𝑅0)−(𝑆1−𝑆0)



Keterangan:

S1, S0 = Timestamp dari paket 1 dan 0

R1, RO = Waktu Kedatangan Paket 1 dan 0

Nilai parameter untuk jitter dapat menggunakan rekomendasi ITU T G 114, ditunjukkan pada tabel 2:

Tabel 2. Rekomendasi ITU T G 114 untuk Nilai Jitter [10, 11]

|  |  |
| --- | --- |
| Nilai Jitter | Kualitas |
| 0 | Sangat Baik |
| 1 - 20 ms | Baik |
| 21 - 50 ms | Cukup |
| > 51 ms | Buruk |

1. Packet Loss

Packet loss merupakan parameter yang menunjukan besarnya nilai paket yang hilang akibat adanya penurunan sinyal pada jaringan, paket yang rusak serta kesalahan yang terjadi pada perangkat keras . Saat terjadi packet loss maka penerima akan meminta retransmisi sehingga dapat mengurangi nilai efisiensi pada jaringan [9, 10]. Parameter untuk nilai packet loss dapat menggunakan rekomendasi ITU T G 114 seperti yang ditunjukkan pada tabel 3.



Tabel 3..Rekomendasi ITU T G 114 Nilai Packet Loss [10, 12]

|  |  |
| --- | --- |
| Nilai Packet Loss | Kualitas |
| 0 % | Sangat Baik |
| 1 - 3 % | Baik |
| 4 - 15 % | Cukup |
| 16 - 25 % | Buruk |

1. Troughput

Throughput mengamati paket yang sukses terkirim ke destinasi selama selang waktu tertentu. Kecepatan transfer data yang diukur dalam satuan bit per detik (bps) Throughput dapat dicari menggunakan persamaan berikut [1]. transfer data yang diukur dalam satuan bit per detik (bps) Throughput dapat dicari menggunakan persamaan berikut [1

Setelah throughput didapatkan, maka dapat melihat nilai indeks dari throughput tersebut dalam Tabel 4.

Tabel 4. Kategori penilaian throughput [10, 12]

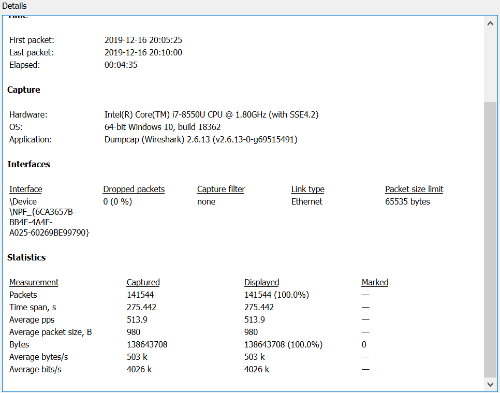
|  |  |
| --- | --- |
| Throughput | Kategori |
| 76 - 100 | Sangat Baik |
| 51 - 75 | Baik |
| 26 - 50 | Cukup |
| 0 - 25 | Buruk |

1. **Hasil dan Pembahasan**

a. Troughput

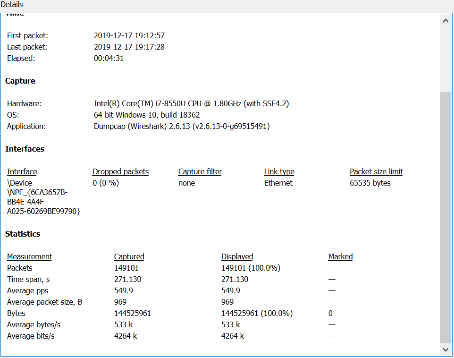
Pengujian throughput mempunyai tujuan untuk mengetahui kecepatan (rate) transfer data efektif, yang diukur dalam bps dimana jumlah total kedatangan paket yang sukses yang diamati pada destination selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut. Dilakukan pengujian sebanyak 3 kali sebagai berikut:

1. Pengujian 1

 Pngujian ke-1 yang dilakukan didapatkan data hasil dimana proses pengambilan data dari devstack ke client berlangsung selama 04 menit 35 detik seperti gambar 1. dimana nilai throughput pada platform devstack berdasarkan pada gambar 1.

Gambar 1. Data Pengujian 1 *Throughput* Devstack

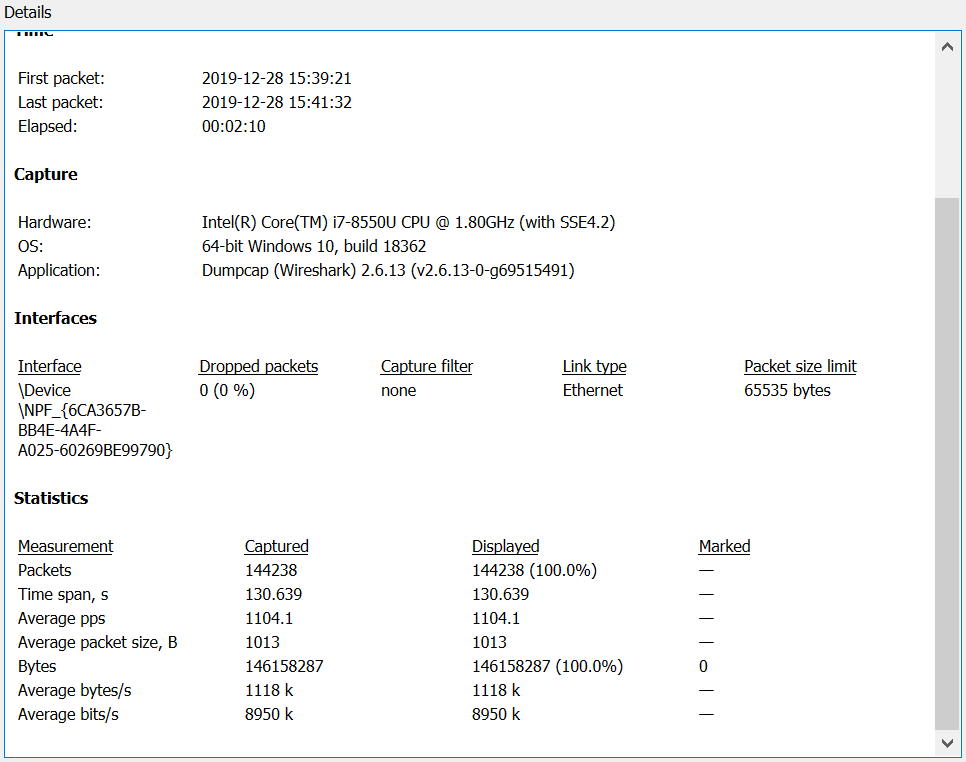
1. Pengujian 2

Pengujian ke-2 yang telah dilakukan berdasarkan gambar 2. didapatkan hasil dimana pengujian throughput berlangsung selama 04 menit 31 detik dimana nilai throughput pada pengujian ke-2 s didapatkan nilai sebagaimana dapat di lihat pada gambar 2:

Gambar 2. Data Pengujian 2 *Throughput* Devstack

1. Pengujian 3

Pengujian ke-3 yang telah dilakukan didapatkan waktu pengiriman throughput berlangsung selama 02 menit 10 detik dimana nilai throughput pada pengujian ke-3.

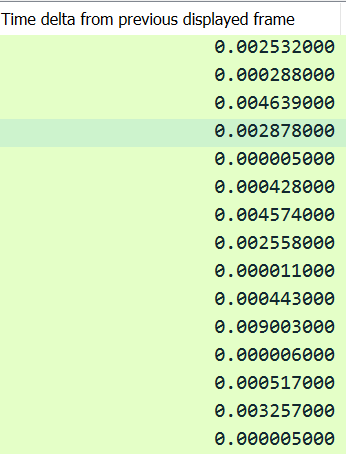
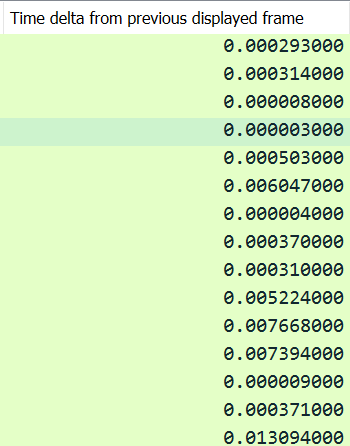
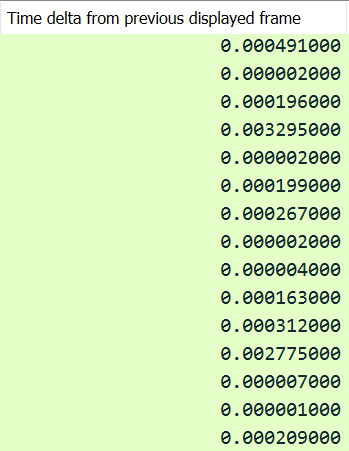


Gambar 3. Data Pengujian 3 *Throughput* Devstack

Dari pengujian yang telah dilakukan, didapatkan nilai throughput pada pengujian ke-1 mendapatkan nilai sebesar 40.20% dan berdasarkan ITU-T termasuk kedalam kategori cukup, sedangkan pada pengujian ke-2 mendapatkan nilai throughput sebesar 42.60% yang berdasarkan pada ketentuan ITU-T termasuk kedalam kategori cukup dan pada pengujian ke-3 pada platform devstack didapatkan nilai throughput sebesar 89,5% berdasarkan pada ketentuan ITU-T nilai tersebut termasuk kedalam karegori sangat baik, Hal ini menandakan nilai throughput pada platform devstack pada pengujian ke-3 lebih baik dari pada pengujian ke-1 dan ke-2. Dimana pada pengujian ke-3 nilai throughput pada cloud computing Devstack mampu melayani permintaan transfer rate dengan baik sehingga nilai throughput yang didapatkan termasuk dalam kategori sangat baik sedangkan pada pengujian ke-1 dan ke-2 termasuk kedalam kegori cukup dimana hal ini disebabkan pada saat proses pengiriman berlangsung tidak bisa melayani permintaan transfer rate dengan baik sehingga terjadi penumpukan paket data yang mempengaruhi nilai throughput.

1. Delay

Delay adalah waktu tunda saat paket yang diakibatkan oleh proses transmisi dari satu titik lain yang menjadi tujuannya. Delay diperoleh dari selisih waktu kirim antara satu paket TCP dengan paket lainnya. Mekanisme yang dilakukan pada pengujian parameter delay yang telah dilakukan sama seperti mekanisme yang dilakukan pada pengukuran throughput

.

Gambar 4. Data Pengujian 1,2,3 *Delay* Devstack

1. Pengujian 1

Sehingga pada pengujian ke-1 didapatkan nilai total delay sebesar 275.442 detik dengan total paket yang diterima sebesar 141554 byte, sehingga untuk nilai

Delay = Total delay / Paket yang diterima

= 275.442 s / 141554 s

= 0.00195 s

= 1.95 ms

1. Pengujian 2

Sedangakan pada pengujian ke-2 untuk parameter delay yang telah dilakukan menggunakan wireshark pada platform devstack. Didapatkan hasil capture pengiriman paket data sebesar 149101 byte dan pada pengujian ke-2 nilai dengan total delay sebesar 271.130 second, jadi untuk nilai rata –rata delay didapatkan hasil sebagai berikut;

Delay = Total delay / Paket yang diterima

= 271.130 / 149101

= 0.00181 s

= 1.81 ms

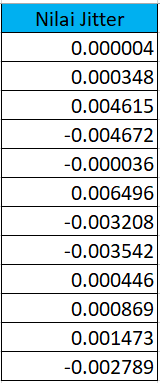
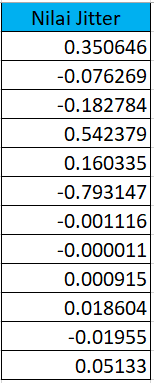
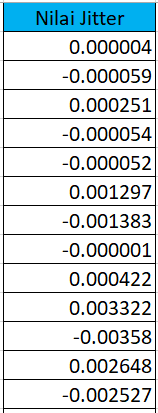
1. Pengujian 3

Pada pengujian ke-3 untuk parameter delay yang telah dilakukan menggunakan mekanisme yang sama dengan pengujian thoughput yaitu pengiriman paket data dari server cloud storage ke client dan pada proses transmisi tersebut dicapture menggunakna wireshark, sehingga didapatkan hasil capture pengiriman paket data sebesar 144238 byte dan pada pengujian ke-3 nilai dengan total delay sebesar 130.638 second, jadi untuk nilai rata –rata delay didapatkan hasil sebagai berikut;

Delay = Total delay / Paket yang diterima = 129.47 s / 130506 s = 0.00099 s = 0.99 ms

Dari pengujian yang telah dilakukan, didapatkan hasil nilai delay pada platform devstack berbeda-beda dengan hasil nilai delay terbesar terdapat pada pengujian ke-1 yaitu sebesar 1.95 ms dan hasil nilai delay terkecil adalah sebesar 0,99 ms dan nilai delay. Pada pengujian ke-1 hasil nilai delay sebesar 1.95 ms berdasarkan ITU-T termasuk kedalam kategori baik dan pada pengujian ke-2 didapatkan nilai sebesar 1.81 ms berdasarkan pada kategori ITU-T termasuk kedalam kategori baik, sedangkan pada pengujian ke-3 yaitu mendapatkan nilai sebesar 0.99 ms dan berdasarkan pada ketetapan ITU-T termasuk kedalam kategori sangat baik. Dari pengujian yang telah dilakukan nilai pada parameter delay pada pengujian ke-3 termasuk dalam kategori sangat baik dan pada pengujian ke-1 dan ke-2 digolongkan kedalam kategori baik hal ini menunjukan bahwa teknologi system jaringan pada platform devstack mampu memberikan performance yang baik dalam sebuah insfrastruktur jaringan komunikasi hal ini ditandai dengan minimnya nilai delay pada pengujian ini.

1. Jitter

Pada pengujian jitter bertujuan untuk mengetahui kualitas kecepatan jaringan dengan mengumpulkan semua delay yang terjadi selama proses data dikirimkan sampai dengan data diterima, Mekanisme pengujian yang dilakukan yaitu dengan proses pengiriman data dari server ke client

.

Gambar 5. Data Pengujian 1,2,3 *Jitter* Devstack

1. Pengujian 1

Dari pengujian ke-1 yang telah dilakukan didapatkan nilai total variasi delay sebesar 0.002326 dan paket yang diterima sebesar 141544 byte. Sehingga nilai total rata-rata jitter adalah sebagai berikut.

Jitter = 0.002326/141544 -1

= 0.0000000164 s

= 0.0000164 ms

1. Pengujian 2

Pada pengujian ke-2 didapatkan hasil dimana untuk total variasi *delay* mendapatkan nilai sebesar 0.70553 second dan paket yang diterima sebesar 149101 bytes. sehingga nilai total pada parameter *jitter* pada pengujian ke-2 ini adalah sebagai berikut:

*Jitter* = 0.70553 /   
 149101-1

= 0.00000473 s

= 0.0047 ms

1. Pengujian 3

Pada pengujian ke-3 sehingga didapatkan hasil dimana untuk total variasi delay mendapatkan nilai sebesar 0.041607 second dan paket yang diterima sebesar 144238 bytes. sehingga nilai total pada parameter jitter pada pengujian ke-3 ini adalah sebagai berikut:

*Jitter*=0.041607   
 144238-1

= 0.000000288463 s

= 0.00028 ms

Dari pengujian yang telah dilakukan didapatkan hasil dengan nilai rata-rata jitter pada pengujian ke-1 sebesar 0.0000164 ms, nilai rata-rata jitter pada pengujian ke-2 didapatkan nilai sebesar 0.004700 ms dan pengujian ke-3 sebesar 0.00028 ms. Jika memacu terhadap nilai standarisasi pada ITU-T nilai pada ketiga pengujian tersebut termasuk dalam kategori sangat baik sehingga system jaringan pada cloud computing mampu memberikan layanan kecepatan komunikasi dengan baik.

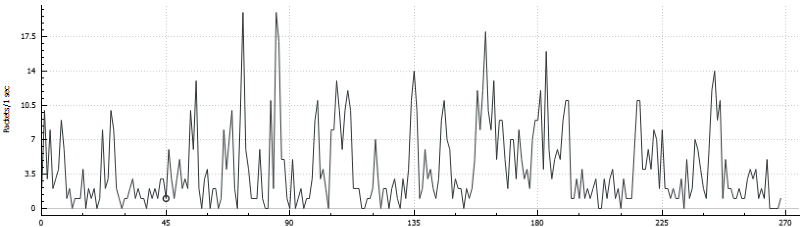
d. Packet Loss

Pada pengujian *Packet loss* dilakukan untuk mengatahui gagalnya paket data yang ditransmisikan kepada alamat tujuan sehingga menyebabkan hilangnya beberapa paket data pada saat proses pengiriman berjalan

* 1. Pengujian 1

Paket loss yang terjadi berlangsung sampai paket data selesai dikirimkan, dimana total packet yang terkirim pada platform devstack sebesar 141544 bytes dan paket data yang hilang sebesar 1101 bytes.

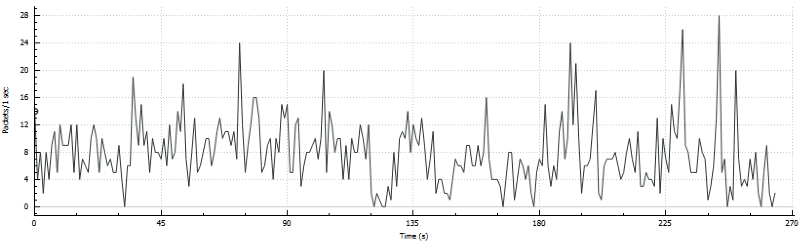
Sehingga nilai rata-rata pada parameter *Packet loss* adalah sebagai berikut:

Packet Loss = 1101 x 100% / 141544 = 0.77 %

Gambar 6. Data Pengujian 1 *Packet loss* Devstack

* 1. Pengujian ke 2

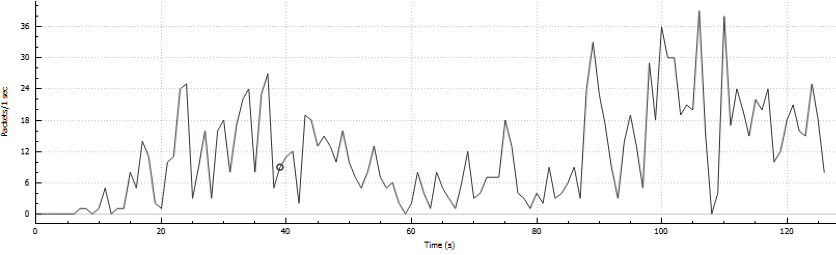
Pengujian ke-2 untuk Packet loss terjadi dengan total packet data yang terkirim sebesar 149101 bytes dan paket data yang hilang sebesar 2061 bytes Sehingga didapatkan nilai rata-rata Packet loss yang terjadi adalah sebagai berikut:

Packet Loss = 2061 x 100% / 149101 = 1.38 %

Gambar 7. Data Pengujian 2 *Packet loss* Devstack

* 1. Pengujian ke 3

Pengujian ke-3 dengan Packet loss terjadi dengan total packet data yang terkirim sebesar 144238 bytes dan paket data yang hilang sebesar 1422 bytes Sehingga didapatkan nilai rata-rata Packet loss yang terjadi adalah sebagai berikut:

*Packet loss* = 1422 x 100% / 144238  = 0.98 %

Gambar 8. Data Pengujian 3 *Packet loss* Devstack

Dari pengujian yang telah dilakukan terhadapa parameter Packet loss didapatkan bahwa nilai Packet loss pada pengujian ke-1 mendapatkan nilai sebesar 0.77% berdasarkan pada ITU-T nilai Packet loss pada pengujian ke-1 termasuk kedalam kategori sangat baik, sedangkan pada pengujian ke-2 mendapatkan nilai sebesar 1.38% sehingga berdasarkan ITU-T termasuk kedalam kategori baik dan nilai Packet loss pada pengujian ke-3 sebesar 0.98% dan berdasarkan pada ketetapan ITU-T termasuk kedalam kategori sangat baik, Dari data hasil pengujian tersebut dimana nilai Packet loss pada pengujian ke-1 dan ke-3 lebih baik dari pada pengujian ke-2 dengan nilai Packet loss tertinggi sebesar 1.38% .

1. **Kesimpulan**

Dari analisa yang dilakukan pada hasil pengujian layanan NFV (Network Function Virtual) pada Private Cloud Storage dapat disimpulkan antara lain:

1. Dalam melakukan pengukuran Quality Of Service pada layanan Network Function Virtual parameter-parameter yang digunakan yaitu delay, jitter, packet loss, dan throughput dengan menggunakan wireshark sebagai tools pengukurannya.

2. Waktu pengujian yang dibutuhkan sebuah paket terhitung sejak pengiriman terjadi oleh server dan sampai diterima oleh client dengan nilai throughput pada pengujian ke-3 sebesar 895 kbps (sangat baik) dan pada pengujian ke-1 sebesar 40.2% dengan nilai kecepatan throughput 402 kbps dan ke-2 sebesar 42.6% tercatat sebesar 426 kbps (baik).

3. Hasil dari parameter delay mengetahui perbedaan selang waktu kedatangan antar paket pada client yaitu didapatkan hasil pada pengujian ke-3 sebesar 0.99 ms yang termasuk kategori sangat baik dan untuk pengujian ke-1 sebesar 1.95 ms dan untuk pengujian ke-2 sebesar 1.81 ms dengan kategori baik.

4. Jumlah bit yang diterima dengan baik melalui media komunikasi (kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data) yaitu jitter didapatkan nilai pada pengujian ke-1

5. sebesar 0.0000164 ms, untuk pengujian ke-2 sebesar 0.0047 dan pengujian ke-3 0.00028 ms berdasarkan dengan parameter ITU-T termasuk kategori sangat baik.

6. Untuk banyaknya paket yang hilang selama proses tranmisi ke client atau untuk nilai packet loss yang didapatkan pada pengujian ke-1 sebesar 0.77% berdasarkan dengan ITU-T termasuk kedalam kategori sangat baik dan pada pengujian ke-2 didapatkan nilai 1.38% berdasarkan ITU-T termasuk dalam kategori baik dan untuk pengujian ke-3 nilai untuk packet loss sebesar 0.98% sehingga termasuk kedalam kategori sangat baik.

7. *Network Function Virtual* berdasarkan pengujian yang telah dilakukan mampu memberikan kemampuan layanan jaringan dengan sangat baik dimana hal ini ditunjukan dari hasil pada parameter-parameter QOS yang telah dilakukan pengujian.

**Ucapan Terima Kasih**

Terima kasih kami ucapkan kepada semua pihak yang telah membantu menyelesaikan penelitian ini. Serta terima kasih kepada Universitas Mercu Buana atas dukungannya untuk publikasi penelitian ini.

**Referensi**

1. Oktavianus, Y. L. (2013). Membangun Sistem Cloud Computing dengan Implementasi Load Balancing dan Pengujian Algoritma Penjadwalan Linux Virtual Server pada FTP Server. Jurnal Nasional Teknik Elektro, 2(1), 25-30.
2. Ge, X., Liu, Y., Du, D. H., Zhang, L., Guan, H., Chen, J., & Hu, X. (2014). OpenANFV: Accelerating network function virtualization with a consolidated framework in openstack. ACM SIGCOMM Computer Communication Review, 44(4), 353-354.
3. Nurohman, M., Hidayat, A. S., & Riana, E. (2018). Perancangan Private Cloud Computing Berbasis Nextcloud Pada Kementerian Perindustrian Jakarta. Jurnal Teknik Komputer, 4(1), 48-55.
4. Santiko, I., Rosidi, R., & Wibawa, S. A. Private Cloud Storage As a Saving Media of Elearning Data on Educational Institutions.
5. Sumarto, S., Thohari, A. H., & Apriyani, M. E. (2014). Rancang Bangun Lab Komputer Virtual Berbasis Cloud Computing Menggunakan Ovirt Pada Jaringan Terpusat. Jurnal Integrasi, 6(1), 72-76.
6. Wati, A., Suroso, S., & Sarjana, S. (2018). Desain Penggunaan Qos (Quality Of Service) Pada Layanan Video Conference Point To Point Dan Multipoint Dengan Metode Kompresi Codec H. 264 Pada Jaringan 4G. Prosiding SENIATI, 4(1), 37-42.
7. Andi Adriansyah, Nanda Ferdana, Setiyo Budiyanto and Julpri Andika. (2019),"Design of Telemedicine Robot using Behavior-based Control Architecture with Two-Step Fuzzy Logic Optimization", Journal of Computer Science, Volume 15, Issue 11, Pages 1617-1626
8. Recommendation, G. (2003). 114-One-way transmission time ITU.
9. Pratiwi, P. E., Isnawati, A. F., & Hikmaturokhman, A. (2013). Analisis QoS Pada Jaringan Multi Protocol Label Switching (MPLS) Studi Kasus di Pelabuhan Indonesia III Cabang Tanjung Intan Cilacap. Purwokerto: Akatel Sandhy Putra Purwokerto.
10. Adilah, A. (2016). Virtualisasi Jaringan Dengan IP Multimedia Subsystem (IMS).
11. Oki Teguh Karya, S S Saesaria and Setiyo Budiyanto, (2018), "RTP analysis for the video transmission process on WhatsApp and Skype against signal strength variations in 802.11 network environments", IOP Conference Series: Materials Science and Engineering, vol: 453, issue : 1
12. Wulandari, A., Putri, A. R., & Fadilah, R. (2019). Rancang Bangun Internet Protocol Television (IPTV) pada Jaringan Wireless LAN di Laboratorium Teknik Telekomunikasi “Protokol Signaling dan Performansi”. In Seminar Nasional Teknik Elektro (Vol. 4, No. 3, pp. 393-403).