

Optimalisasi Private Cloud Storage Berbasis Devstack Guna Meningkatkan Performansi Network Function Virtual

The Optimization of Private Cloud Storage Based on Devstack to Improve The Performance of The Network Function Virtual

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

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

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

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

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

sbudiyanto@mercubuana.ac.id^{1*}, kristianinatalia12@gmail.com², freddy.artadima@mercubuana.ac.id³,

lukman.medriavin@mercubuana.ac.id⁴, fajarrahayu@upnvj.ac.id⁵

Abstrak – Sistem 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, dilanjutkan dengan pengukuran QoS diantaranya adalah throughput, bit rate, jitter, dan delay. Melalui penelitian ini, dapat diketahui kemampuan suatu jaringan virtual pada suatu insfrastruktur cloud computing. Berdasarkan standar ITU-T, nilai Jitter dan packet loss bernilai sangat baik.

Kata Kunci: Virtualisasi, Devstack, Network Function Virtualization, QoS, cloud storage.

Abstract – NFV system is a network architecture that used virtual 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. Based on this system, the Private Cloud Storage service can load on a virtual network. This research, implemented the virtual networks using Devstack based on NFV infrastructure in the private cloud storage service, then the system measured the QoS including throughput, bit rate, jitter, and delay. The ability of a virtual network in a cloud computing infrastructure can be seen through this research. Based on ITU-T standard, jitter and packet loss value had a very good performance.

Keywords: Virtualization, Devstack, Network Function Virtual, QoS, cloud storage.

1. Pendahuluan

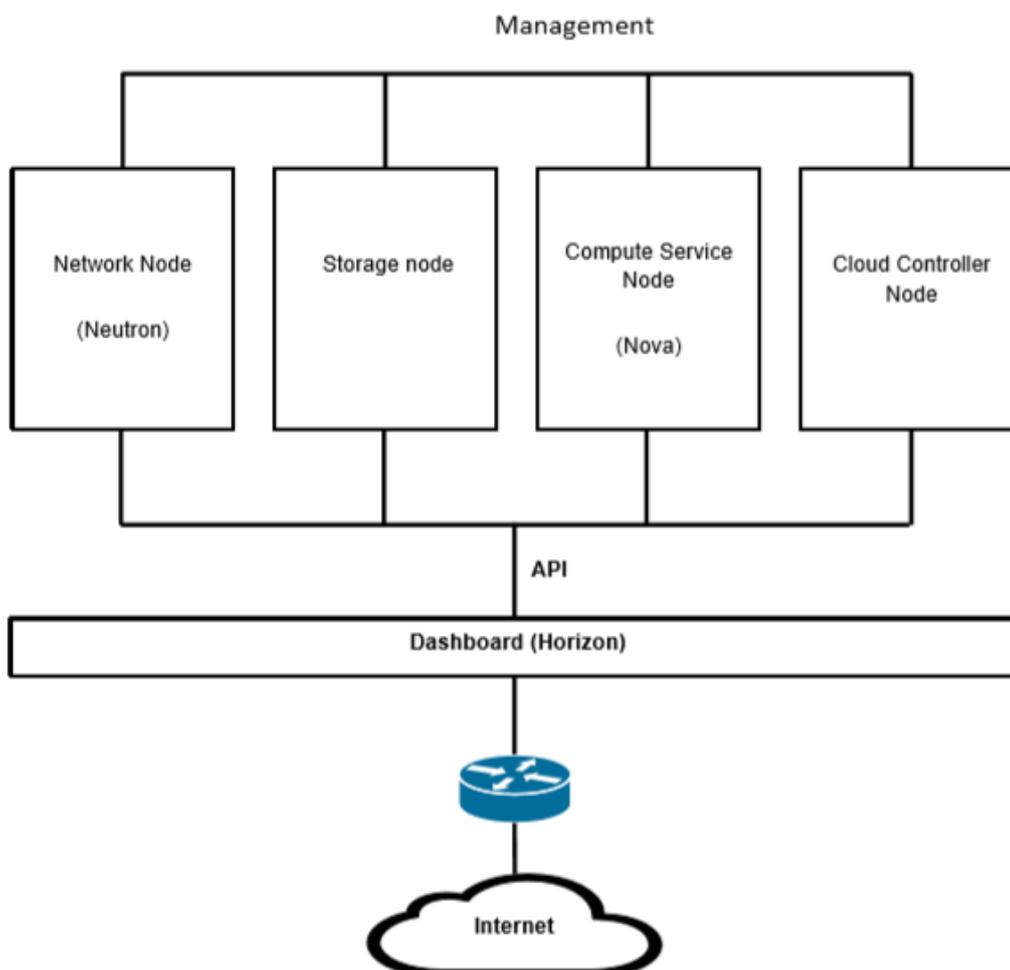
Seiring perkembangan teknologi yang sangat cepat, sistem teknologi jaringan yang awalnya banyak menggunakan perangkat fisik cenderung berubah menjadi perangkat virtual. Sistem virtualisasi merupakan sebuah solusi yang berguna untuk kemajuan teknologi dimana dengan sistem virtualisasi ini sebuah perangkat 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 bisa meningkatkan efisiensi dan fleksibilitas dalam melakukan pengembangan sistem jaringan pada suatu infrastruktur [1, 2].

Pembangunan sebuah infrastruktur 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 infrastruktur jaringan. Oleh sebab itu dengan adanya sistem virtualisasi diharapkan bisa membantu dalam pembangunan infrastruktur jaringan dimana sebuah perangkat keras dapat memiliki fungsi yang luas dalam pembangunan infrastruktur jaringan, yaitu *Network Function Virtual* (NFV) [3].

Dalam penelitian ini dilakukan implementasi jaringan virtual menggunakan devstack yang berbasis pada parameter kinerja yang diukur pada infrastruktur NFV pada layanan *private cloud storage* yang sudah diimplementasikan diantaranya adalah *throughput*, *bitrate*, *jitter*, dan *delay*. Hasil penelitian ini dapat diketahui 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, sehingga 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* terdapat beberapa bagian atau *layer* diantaranya ada *Network Function Virtualization Infrastruktur* (NFVI), *Virtualised Network Function* (VNF), *Element Management System* (EMS), dan *Operations and Business 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 infrastructure* 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* berfungsi untuk menjalankan komputasi virtual (ESXi, KVM, QEMU), penyimpanan virtual (LVM, NFS), dan jaringan virtual (MPLS VPN, VRF).

2. Metode Penelitian

Penelitian yang dilakukan menggunakan pendekatan *engineering*. Arsitektur sistem mengacu pada platform Devstack. Berdasarkan pada arsitekturnya, rancangan penelitian yang dirancang dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur Devstack.

Dengan mengacu pada skema di atas, sistem dikembangkan dengan *Nextcloud*, yaitu 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 (*open-source*) 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 *opensource* enterprise *file sync* dan *share* berbasis *cloud* [4]. Pengembangan sistem mengacu pada konsep komputasi *cloud*, termasuk penyimpanan dan pemrosesannya. *Cloud storage* memiliki banyak keuntungan dibandingkan dengan penyimpanan data secara tradisional. Dalam penelitian ini digunakan sistem *private cloud* karena model ini cocok untuk perusahaan dengan skala enterprise yang sangat memperhatikan privasi dan keamanan data [5]. *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 memori [6], [7]. Sistem diuji dengan melakukan proses pengambilan data dari devstrack ke *client*. Selama proses ini akan diukur parameter QoS-nya. QOS merupakan suatu parameter uji kelayakan suatu layanan agar dapat dikatakan baik [6]. Beberapa parameter uji untuk 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 [8], [9].

Tabel 1. Rekomendasi ITU-T G 114 untuk nilai *delay*.

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

Nilai *delay* dihitung dengan persamaan

$$\text{Rerata Delay} = \frac{\text{Total delay}}{\text{Paket yang diterima}} \quad (1)$$

2. Jitter

Jitter merupakan variasi *delay* yang terjadi karena adanya selisih waktu kedatangan paket pada jaringan IP [1]. Nilai jitter dihitung dengan persamaan

$$\text{Jitter} = \frac{\text{Total variasi delay}}{\text{Pkt diterima}-1} \quad (2)$$

Nilai parameter untuk *jitter* dapat menggunakan rekomendasi ITU T G 114, ditunjukkan pada Tabel 2 [1], [11].

Tabel 2. Rekomendasi ITU-T G 114 untuk nilai *jitter*.

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

3. Packet Loss

Packet loss merupakan parameter yang menunjukkan 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 [10, 12]. *Packet loss* dihitung dengan persamaan

$$\text{Packet Loss} = \frac{Y}{A} \times 100\% \quad (3)$$

dengan Y paket yang hilang, dan A adalah total paket yang dikirim.

Tabel 3. Rekomendasi ITU-T G 114 untuk nilai *packet loss*.

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

4. Throughput

Pengujian dilakukan dengan mengamati paket yang sukses terkirim ketujuan selama selang waktu tertentu. Kecepatan transfer data yang diukur dalam satuan bit per detik (bps). *Throughput* dapat dicari menggunakan persamaan [1]. Transfer data yang diukur dalam satuan bit per detik (bps). 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

Hasil dari pengujian kualitas jaringan akan dibandingkan dengan layanan *platform Network Function Virtual AWS*.

3. Hasil dan Pembahasan

Pengujian dan analisis hasil implementasi NFV menggunakan *Devstack* pada *cloud storage* yang telah dibuat ini bertujuan untuk mengetahui performansi kualitas jaringan NFV yang dibuat menggunakan *Devstack* pada layanan *cloud storage*. Hasil dari pengujian kualitas jaringan akan dibandingkan dengan layanan *platform* NFV lainnya seperti AWS yang berguna mengetahui platform NFV yang lebih baik berdasarkan parameter-parameter standarisasi ITU-T pada *quality of service*. Pengujian pada penelitian ini dilakukan dengan mengambil hasil yang didapat dari perangkat lunak *Wireshark*. Adapun skenario yang dilakukan pada pengujian ini dapat dilihat pada Tabel 5.

Tabel 5. Skenario pengujian.

Keterangan	Alat
Alat Pengujian	Laptop
Jaringan ISP	Indihome
<i>Software Monitoring</i>	<i>Wireshark</i>
<i>Software Server / OS</i>	<i>Nextcloud (Devstack dan AWS)</i>
<i>Software Client / OS</i>	<i>Google Chrome (Windows 10)</i>

3.1. 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 ke-1
Pengujian ke-1 yang telah dilakukan dengan mengambil data dari *Devstack* ke *client* didapatkan selama 4 menit 35 detik.
2. Pengujian ke-2
Pengujian ke-2 yang telah dilakukan dengan mengambil data dari *Devstack* ke *client* didapatkan selama 4 menit 31 detik.
3. Pengujian ke-3
Pengujian ke-3 yang telah dilakukan dengan mengambil data dari *Devstack* ke *client* didapatkan selama 2 menit 10 detik.

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. 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, 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*.

3.2. Delay

Delay adalah waktu tunda yang diakibatkan oleh proses transmisi paket dari satu pengiriman ke tujuan. *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*.

1. Pengujian ke-1

Pada pengujian ke-1 didapatkan nilai total *delay* sebesar 275.442 detik dengan total paket yang diterima sebesar 141554byte. Dengan menggunakan (1), diperoleh rata-rata *delay* pengujian ke-1

$$\text{Rerata Delay} = \frac{\text{Total delay}}{\text{Paket diterima}} = \frac{275.442}{141554} \approx 1.95 \text{ milidetik.}$$

2. Pengujian ke-2

Pada pengujian ke-2 didapatkan nilai total *delay* sebesar 271.130 detik dengan total paket yang diterima sebesar 149101byte. Dengan menggunakan (1), diperoleh rata-rata *delay* pengujian ke-2

$$\text{Rerata Delay} = \frac{\text{Total delay}}{\text{Paket diterima}} = \frac{271.130}{149101} \approx 1.81 \text{ milidetik.}$$

3. Pengujian ke-3

Pada pengujian ke-3 didapatkan nilai total *delay* sebesar 130.638 detik dengan total paket yang diterima sebesar 144238byte. Dengan menggunakan (1), diperoleh rata-rata *delay* pengujian ke-3

$$\text{Rerata Delay} = \frac{\text{Total delay}}{\text{Paket diterima}} = \frac{129.47}{130506} \approx 0.99 \text{ milidetik.}$$

Dari pengujian yang telah dilakukan, didapatkan hasil nilai *delay* pada *platform Devstack* berbeda-beda dengan rata-rata *delay* terdapat pada pengujian ke-1 yaitu sebesar 1.95 milidetik dan hasil nilai *delay* terkecil adalah sebesar 0,99 milidetik diperoleh pada pengujian ke-3. Pada pengujian ke-1 hasil nilai *delay* sebesar 1.95 milidetik, berdasarkan ITU-T, termasuk kedalam kategori baik. Pada pengujian ke-2, didapatkan nilai sebesar 1.81 milidetik, dan berdasarkan pada kategori ITU-T, termasuk ke dalam kategori baik. Sedangkan pada pengujian ke-3, dengan nilai sebesar 0.99 milidetik, berdasarkan pada ketentuan ITU-T, termasuk kedalam kategori sangat baik. Dari pengujian menunjukkan bahwa teknologi sistem jaringan pada *platform Devstack* mampu memberikan performansi yang baik dalam sebuah infrastruktur jaringan komunikasi hal ini ditandai dengan kecilnya nilai *delay* pada pengujian ini.

3.3. Jitter

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*.

1. Pengujian ke-1

Dari pengujian ke-1 yang telah dilakukan dengan mengambil data dari *devstack* ke *client* didapatkan nilai total variasi *delay* sebesar 0,002326 detik dan paket yang diterima sebesar 141544 byte. Dengan menggunakan (2), diperoleh rata-rata nilai *jitter* pengujian ke-1 sebesar

$$\text{Jitter} = \frac{\text{Total variasi delay}}{\text{Paket diterima} - 1} = \frac{0,002326}{141544 - 1} \approx 0,0000164 \text{ milidetik.}$$

2. Pengujian ke-2

Dari pengujian ke-2 yang telah dilakukan dengan mengambil data dari *devstack* ke *client* didapatkan nilai total variasi *delay* sebesar 0,70553s dan paket yang diterima sebesar 149101 byte. Dengan menggunakan (2), diperoleh rata-rata nilai *jitter* pengujian ke-2 sebesar

$$Jitter = \frac{\text{Total variasi delay}}{\text{Paket diterima} - 1} = \frac{0,70553}{149101 - 1} \approx 0,0047 \text{ milidetik.}$$

3. Pengujian ke-3

Dari pengujian ke-3 yang telah dilakukan dengan mengambil data dari *devstack* ke *client* didapatkan nilai total *jitter* sebesar 0,041607s dan paket yang diterima sebesar 144238 byte. Dengan menggunakan (2), diperoleh rata-rata nilai *jitter* pengujian ke-3 sebesar

$$Jitter = \frac{\text{Total variasi delay}}{\text{Paket diterima} - 1} = \frac{0,041607}{144238} \approx 0,00028 \text{ milidetik.}$$

Dari pengujian yang telah dilakukan didapatkan hasil dengan nilai rata-rata *jitter* pada pengujian ke-1 sebesar 0,0000164 milidetik, nilai rata-rata *jitter* pada pengujian ke-2 didapatkan nilai sebesar 0,004700 milidetik dan pengujian ke-3 sebesar 0,00028 milidetik. Jika mengacu terhadap nilai standarisasi pada ITU-T, nilai pada ketiga pengujian tersebut termasuk dalam kategori sangat baik sehingga sistem jaringan pada *cloud computing* mampu memberikan layanan kecepatan komunikasi dengan baik.

3.4. Packet Loss

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

1. Pengujian ke-1

Packet loss yang terjadi berlangsung sampai paket data selesai dikirimkan, dimana total paket yang terkirim pada *platform Devstack* sebesar 141544 byte dan paket data yang hilang sebesar 1101 byte. Menggunakan (3), nilai rata-rata pada parameter *packet loss* pengujian ke-1 adalah

$$\text{Packet Loss} = \frac{\text{Paket hilang}}{\text{Total paket}} \times 100\% = \frac{1101}{141544} \times 100\% = 0,77\%.$$

2. Pengujian 2

Packet loss yang terjadi berlangsung sampai paket data selesai dikirimkan, dimana total paket yang terkirim pada *platform Devstack* sebesar 149101 byte dan paket data yang hilang sebesar 2061 byte. Menggunakan (3), nilai rata-rata pada parameter *packet loss* pengujian ke-2 adalah

$$\text{Packet Loss} = \frac{\text{Paket hilang}}{\text{Total paket}} \times 100\% = \frac{2061}{149101} \times 100\% = 1,38\%.$$

3. Pengujian 3

Packet loss yang terjadi berlangsung sampai paket data selesai dikirimkan, dimana total paket yang terkirim pada *platform devstack* sebesar 144238 byte dan paket data yang hilang sebesar 1422 byte. Menggunakan (3), nilai rata-rata pada parameter *packet loss* pengujian ke-3 adalah

$$\text{Packet Loss} = \frac{\text{Paket hilang}}{\text{Total paket}} \times 100\% = \frac{1422}{144238} \times 100\% = 0,98\%.$$

Dari pengujian yang telah dilakukan terhadap 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. Pada pengujian ke-2 didapatkan nilai sebesar 1,38%, sehingga berdasarkan ITU-T termasuk kedalam kategori baik. Nilai *packet loss* pada pengujian ke-3 sebesar 0,98% dan berdasarkan pada ketentuan ITU-T termasuk kedalam kategori sangat baik. Dari data hasil pengujian tersebut 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%.

4. Kesimpulan

Dari analisis yang dilakukan terhadap hasil pengujian layanan NFV pada *Private Cloud Storage* dapat disimpulkan bahwa *Network Function Virtual* berdasarkan pengujian yang telah dilakukan mampu memberikan kemampuan layanan jaringan dengan sangat baik, hal ini ditunjukkan 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. “Membangun Sistem Cloud Computing dengan Implementasi Load Balancing dan Pengujian Algoritma Penjadwalan Linux Virtual Server pada FTP Server”, *Jurnal Nasional Teknik Elektro*, vol. 2 No.1, p.25-30,2013.
- [2] Ge, X., Liu, Y., Du, D. H., Zhang, L., Guan, H., Chen, J., & Hu, X, “OpenANFV: Accelerating network function virtualization with a consolidated framework in openstack”, *ACM SIGCOMM Computer Communication Review*, Vol. 44 No. 4)p. 353-354,2014.
- [3] Nurohman, M., Hidayat, A. S., & Riana, E, “Perancangan Private Cloud Computing Berbasis Nextcloud Pada Kementerian Perindustrian Jakarta”, *Jurnal Teknik Komputer*, Vol 4 No.1,p. 48-55, 2018
- [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, “Rancang Bangun Lab Komputer Virtual Berbasis Cloud Computing Menggunakan Ovirt Pada Jaringan Terpusat”, *Jurnal Integrasi*, Vol. 6 No.1,p. 72-76,2014
- [6] Wati, A., Suroso, S., & Sarjana, S, “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*, Vol.4 No. 1, 37-42,2018
- [7] Andi Adriansyah, Nanda Ferdana, Setiyo Budiyanto and Julpri Andika, “Design of Telemedicine Robot using Behavior-based Control Architecture with Two-Step Fuzzy Logic Optimization”, *Journal of Computer Science*, Vol. 15, Issue 11, Pages 1617-1626, 2019.
- [8] Recommendation, G, 114-One-way transmission time ITU, 2003
- [9] Pratiwi, P. E., Isnawati, A. F., & Hikmaturokhman, A, “Analisis QoS Pada Jaringan Multi Protocol Label Switching (MPLS) Studi Kasus di Pelabuhan Indonesia III Cabang Tanjung Intan Cilacap”. Purwokerto: *Akatel Sandhy Putra Purwokerto*, 2013.
- [10] Adilah, A, *Virtualisasi Jaringan Dengan IP Multimedia Subsystem (IMS)*,2016
- [11] Oki Teguh Karya, S S Saesaria and Setiyo Budiyanto, “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, No. 1, 2018

- [12] Wulandari, A., Putri, A. R., & Fadilah, R, Rancang Bangun Internet Protocol Television (IPTV) pada Jaringan Wireless LAN di Laboratorium Teknik Telekomunikasi “Protokol Signaling dan Performansi”, *Prosiding Seminar Nasional Teknik Elektro* Vol. 4, No. 3, pp. 393-403, 2019.