

Quality of Service (QoS) Server Konsolidasi dengan Menggunakan Hypervisor Proxmox VE

I Gusti Ngurah Wikranta Arsa
Program Studi Sistem Komputer
Institut Teknologi dan Bisnis STIKOM Bali
Denpasar, Indonesia
e-mail: arsa@stikom-bali.ac.id

Diajukan: 28 Januari 2022; Direvisi: 6 April 2022; Diterima: 25 April 2022

Abstrak

Dalam beberapa penelitian salah satu teknik yang dapat menunjang layanan infrastruktur adalah server konsolidasi, di mana server konsolidasi merupakan sebuah teknik yang menggunakan sebuah server fisik dengan sumber daya yang besar dan digunakan untuk menghasilkan beberapa server virtual. Dalam pengimplementasian server konsolidasi dengan teknik virtualisasi untuk layanan cloud memiliki beberapa masalah di antaranya adalah sejauh mana kualitas dari penyedia layanan infrastruktur dengan konsolidasi server, serta perlunya dilakukan beberapa pengujian untuk mendapatkan kualitas yang baik. Penelitian ini menggunakan Proxmox sebagai Hypervisor. Salah satu metode pengujian yang dapat digunakan adalah dengan pendekatan Quality of Service (QoS). Tujuan penelitian adalah melakukan pengukuran kualitas dan menentukan QoS dengan empat parameter untuk pelayanan pada server konsolidasi dengan virtualisasi sebagai penyedia layanan infrastruktur cloud yaitu Throughput, Packet Loss, Latency (delay), dan Jitter. Hasil penelitian ini berupa sebuah arsitektur konsolidasi server dengan empat lapisan yaitu layer mesin virtual, layer mesin fisik, layer jaringan dan layer pengguna, dihasilkan topologi jaringan dengan konsolidasi server, hasil rata-rata sepuluh mesin virtual untuk throughput adalah kurang, Jitter bagus, delay dan packet loss sangat bagus. Dengan indeks QoS masuk dalam kategori memuaskan dengan indeks 3.

Kata kunci: Server konsolidasi, QoS, Virtualisasi, Proxmox.

Abstract

In several studies, one of the techniques that can support infrastructure services is server consolidation, where server consolidation is a technique that uses a physical server with large resources and is used to produce several virtual servers. In implementing a consolidated server with virtualization techniques for cloud services, there are several problems including the extent to which the quality of the infrastructure service provider with server consolidation, as well as the need for some tests to get good quality. This study uses Proxmox as a Hypervisor. One of the testing methods that can be used is the Quality of Service (QoS) approach. The purpose of this study is to measure quality and determine QoS with four parameters for services on a consolidated server with virtualization as a cloud infrastructure service provider, namely Throughput, Packet Loss, Latency (delay), and Jitter. The results of this study are a server consolidation architecture with four layers, namely the virtual machine layer, physical machine layer, network layer and user layer, generated a network topology with server consolidation, the average result of ten virtual machines for throughput is less, good jitter, delay and packet loss is very good. The QoS index is in the satisfactory category with an index of 3.

Keywords: Server consolidations, QoS, Virtualizations, Proxmox.

1. Pendahuluan

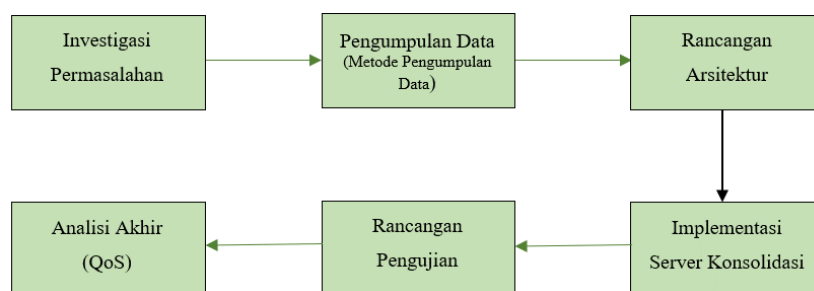
Kebutuhan akan komputasi menjadi semakin meningkat disebabkan oleh perkembangan digitalisasi dan internet sebagai sebuah kebutuhan dalam penunjang berbagai sektor. Permintaan komputasi yang meningkat mengakibatkan beberapa bisnis digital mulai mengembangkan layanan penyediaan komputasi, salah satunya berbasis *cloud computing*, di mana salah satu layanan yang direkomendasikan NIST [1] adalah layanan penyediaan infrastruktur atau dikenal dengan *Infrastructure As A Service (IAAS)*. Dalam beberapa penelitian salah satu teknik yang dapat menunjang layanan infrastruktur adalah server konsolidasi, di mana server konsolidasi merupakan sebuah teknik yang menggunakan sebuah server fisik

dengan sumber daya yang besar dan digunakan untuk menghasilkan beberapa *server* virtual atau dikenal dengan virtual mesin dan digabungkan menjadi sebuah servis [2]. Kelebihan dari konsolidasi *server* adalah merupakan pendekatan yang efektif dan banyak digunakan untuk mengurangi konsumsi energi total di pusat data [3]. Konsolidasi *server* dapat menyebabkan penurunan kinerja pekerjaan yang signifikan karena lokasi bersama dan migrasi mesin virtual [4].

Dalam pengimplementasian *server* konsolidasi dengan teknik virtualisasi untuk layanan *cloud* memiliki beberapa masalah di antaranya adalah sejauh mana kualitas dari penyedia layanan infrastruktur dengan konsolidasi *server*, serta perlunya dilakukan beberapa pengujian untuk mendapatkan kualitas yang baik. Salah satu yang dapat digunakan adalah dengan pendekatan *Quality of Service* (QoS). QoS adalah merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari satu servis dan Hasil analisis QoS, dapat dijadikan rekomendasi untuk implementasi fisik jaringan internet yang harapan ke depannya bisa menunjang penambahan layanan-layanan yang dapat menunjang kegiatan kantor [5]. Pengujian kinerja perlu dilakukan untuk melihat sejauh mana *server* dapat melayani permintaan dan melakukan beban kerja. Kebijakan konsolidasi *server* yang baik harus memastikan QoS yang andal bersama dengan pengurangan konsumsi energi untuk pelaksanaan beban kerja [6]. Tujuan QoS adalah untuk menyediakan kualitas layanan yang berbeda-beda untuk beragam kebutuhan akan layanan di dalam jaringan IP, sebagai contoh untuk menyediakan *bandwidth*, menurunkan hilangnya paket-paket, menurunkan waktu tunda dan variasi waktu tunda dan lainnya [7][8]. Adapun beberapa parameter yang dapat digunakan dalam pengukuran QoS adalah *Throughput*, *Packet Loss*, *Latency* (delay), *Jitter*, *MOS* (*Mean Opinion Score*), *Echo cancelation*, dan *Post Dial Delay*.

2. Metode Penelitian

Untuk lebih jelas tahapan-tahapan yang ditempuh dalam penelitian ini nantinya dapat dilihat pada Gambar 1. Di mana dari tahapan yang dilakukan tujuan akhirnya adalah menghasilkan QoS untuk penyediaan *server* infrastruktur *cloud* yang dilakukan pada penelitian sebelumnya.



Gambar 1. Tahapan Penelitian

Tahap pertama dalam penelitian ini adalah investigasi masalah. Yang menjadi masalah kenapa penelitian ini dilakukan adalah sejauh mana QoS dari sistem yang dirancang yaitu Bagaimana mengukur kualitas pelayanan pada *server* konsolidasi dengan virtualisasi sebagai penyedia layanan infrastruktur *cloud*. Selain itu bagaimana menentukan QoS dengan empat parameter untuk pelayanan pada *server* konsolidasi dengan virtualisasi sebagai penyedia layanan infrastruktur *cloud*.

Tahap selanjutnya adalah pengumpulan data, pada tahap ini data yang digunakan bersumber dari data kuantitatif yang diambil dari pengujian sistem. Pada tahap ini direncanakan jenis data nantinya diambil seperti apa. Kemudian pada tahapan ini juga dilakukan beberapa pengumpulan data untuk menunjang dalam pembentukan arsitektur dari konsolidasi *server* yang akan dibuat.

Tahap ketiga adalah perancangan arsitektur, pada tahap ini dirancang infrastruktur, instalasi bagaimana sebuah penyedia layanan *server*. Selanjutnya dilihat seperti apa pengaplikasian dan tentunya arsitektur yang akan dihasilkan dalam penelitian berdasarkan dari instalasi *server*. Tahap ini menghasilkan sebuah arsitektur *server* konsolidasi.

Tahap keempat adalah implementasi, tahap ini adalah pengimplementasian dari sistem penyedia layanan *cloud server* atau *cloud* infrastruktur berdasarkan dari arsitektur yang diusulkan. Implementasi di sini akan menggunakan Proxmox VE sebagai *hypervisor* yang dapat menghasilkan *server* virtual. Proxmox VE berbasis Linux Debian yang *open source*. Kemudian nantinya dibuatkan beberapa *server* virtual yang berjalan pada satu *server* fisik. Pada implementasi direncanakan menggunakan 2 *server* fisik dan 5-10 virtual mesin akan dibuat

Tahap kelima adalah rancangan pengujian, pada tahap ini di rancangan skenario pengujian dan pengambilan data. Dalam tahap ini dibuatkan beberapa skenario pengujian yang dibutuhkan pada tahap analisis akhir.

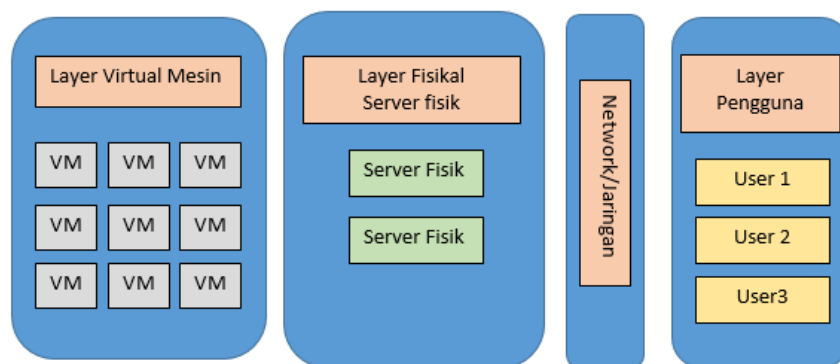
Tahap terakhir adalah analisis akhir, pada tahap ini dilakukan analisis dari hasil pengujian. Analisis akhir ini memberikan laporan berupa grafis hasil dari pengujian dan penjelasan kinerja yang dihasilkan dari *server* konsolidasi sesuai dengan parameter QoS.

3. Hasil dan Pembahasan

Terdapat beberapa hasil yang diperoleh dari penelitian ini. Dimulai dari hasil arsitektur, hasil rancangan sistem, sampai pembahasan.

3.1. Hasil Rancangan Arsitektur

Dalam rancangan arsitektur *server* konsolidasi dapat dilihat pada Gambar 5.1, di mana terlihat lapisan yang dirancang untuk diimplementasikan sebagai sebuah layanan *cloud* infrastruktur yang menggunakan teknik konsolidasi *server*. hal ini dapat dilihat dari layer fisik yang dapat terdiri dari 2 atau lebih *server* dan dapat dihasilkan beberapa *server* virtual (layer virtual mesin) yang nantinya akan menjadi layanan infrastruktur berupa *server* kepada pengguna (layer pengguna).



Gambar 2. Rancangan Arsitektur

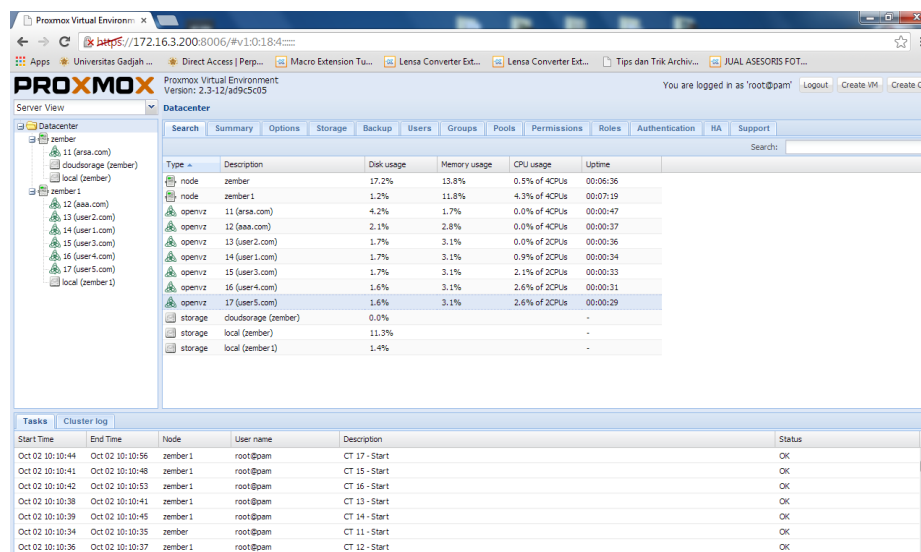
Dalam rancangan arsitektur pada Gambar 2 terlihat jelas layer per layer dari sistem yang akan menjadi layanan penyedia infrastruktur *cloud* dalam hal ini *server*. dari arsitektur tersebut dibuatlah topologi yang menggambarkan seperti apa gambaran detail sistem nantinya sebelum diimplementasikan dan dikonfigurasi agar sesuai dengan rancangan arsitektur sistem. Layer virtual mesin akan berada di atas layer fisik. Virtual mesin nantinya akan berada di dalam *server* fisik. *Server* fisik ini akan terhubung secara jaringan dan akan dikontrol oleh sebuah *server controller*. *Server controller* berfungsi sebagai *server* pengendali dari beberapa *server* fisik yang terhubung ke dalam sebuah jaringan (konsolidasi *server*). *Server* fisik ini nantinya akan dikonfigurasi dan diinstallkan sistem operasi dengan basis Linux yaitu Proxmox VE. Kemudian *server controller* akan menjadi *gateway* dari user-user yang memerlukan layanan infrastruktur berupa *server*. *User* akan merasakan memiliki *server* secara mandiri dan dapat dikelola secara independen. *Server* yang diterima oleh *user* ini sebenarnya adalah sebuah *server* virtual yang hidup atau menggunakan sumber daya dari *server* fisik. Penggunaan sumber daya fisik secara bersama-sama dalam bentuk *server* virtual merupakan konsep dari konsolidasi *server* atau dapat diistilahkan adalah *server* gotong royong.

3.2. Implementasi Sistem

Sistem ini dijalankan dengan menggunakan jaringan lokal yang sudah ada. Sehingga untuk IP *address* cukup menyesuaikan dengan IP *address* yang sudah ada. Sedangkan untuk perangkat lunak, sistem ini menggunakan:

1. Proxmox VE 2.3 untuk *server cloud* baik *node* ataupun *controller*.
2. Ubuntu 12.04 LTS untuk *server* pembanding.
3. Putty untuk *remote* SSH di Windows.
4. Browser Mozilla Firefox yang terinstal Java *plugin*.
5. Beberapa *template* OpenVZ di Proxmox seperti *debian-6.0-eyeos*, dll.
6. Httperf sebagai *tool* untuk melakukan pengujian.

Sistem ini akan diimplementasikan dengan mengonfigurasi satu *server* sebagai *cloud controller* dan satu *server* menjadi *cloud node*. Kedua *server* ini diisi dengan sistem operasi Proxmox VE 2.3. Kemudian kedua *server* dikonfigurasi menjadi *cluster*, sehingga *cloud node* dapat dikelola dalam satu buah *server* yaitu *cloud controller*. Gambar 3 adalah tampilan web manajemen *cloud* Proxmox.



Gambar 3. Tampilan Proxmox.

Server Controller juga dapat diakses menggunakan terminal dengan memanfaatkan aplikasi seperti Putty. Proxmox pada dasarnya adalah sistem operasi *cloud* yang berbasis Debian. Jadi perintah dasar terminalnya menggunakan perintah dasar dari Debian.

3.3. Skenario Pengujian

Setelah implementasi sistem dilakukan, tahapan selanjutnya adalah merancang skenario pengujian untuk mendapatkan data kuantitatif yang nantinya digunakan untuk melihat sejauh mana kualitas dari layanan yang telah dibuat. Dalam skenario pengujian ini ada beberapa yang rancang pengujian untuk mendapatkan hasil dari empat parameter *Quality of Service* (QoS) yang digunakan yaitu *Throughput*, *Packet Loss*, *Latency* (delay), dan *Jitter*. Adapun skenario pengujian sebagai berikut:

1. Mengimplementasikan arsitektur dan topologi sesuai dengan rancangan serta melakukan konfigurasi *server*
2. Membuat 10 *server* virtual dengan menggunakan Proxmox VE pada *server controller*.
3. Melakukan konfigurasi mesin virtual. Dalam skenario ini mesin virtual yang diterima akan dikonfigurasi menjadi sebuah *server* web. Pemilihan sebagai *server* web dimaksudkan untuk mempermudah dalam proses pengukuran (*Benchmarking*) dalam penelitian ini.
4. Permintaan halaman (*request*). Skenario ini dipilih karena salah satu peran yang paling sering ditemukan pada komputer *server* adalah permintaan suatu halaman web. Pengujian akan dilakukan dengan mengakses aplikasi dengan melakukan HTTP *request* halaman web dalam hal ini *default* web *server* yang sudah diinstal pada *server native* (*server* konvensional) dan *server virtual cloud*. Proses *request* menggunakan *tool* Httperf. Kemudian akan dicatat waktu yang digunakan sistem untuk merespons *request* yang dibebankan.. Selanjutnya akan di catat waktu responsnya.
5. Selanjutnya dilakukan pencatatan paket *loss* dari jumlah paket yang dikirimkan.

3.4. Hasil Pengujian

Hasil Pengujian Dengan menggunakan Httperf dapat dilihat pada Gambar 4 untuk satu VM yang aktif pada *server* fisik. Gambar 5 menunjukkan hasil sampling dari 2 virtual mesin dari 10 virtual mesin yang aktif dan dibebankan masing-masing 100 *connections* dengan masing-masing 10 *request* per detik. Untuk kebutuhan pengujian pada penelitian ini *server* virtual dikonfigurasi menjadi *server* web.

```

root@administrator-To-be-filled-by-O-E-M:/home/arsa
root@administrator-To-be-filled-by-O-E-M:/home/arsa#
root@administrator-To-be-filled-by-O-E-M:/home/arsa# httpperf --server 172.16.3.201 --port 10004 --num-conns 100 --num-calls 10 --uri /index.html
httpperf --client=0/1 --server=172.16.3.201 --port=10004 --uri=/index.html --send-buffer=4096 --recv-buffer=16384 --num-conns=100 --num-calls=10
Maximum connect burst length: 1

Total: connections 100 requests 1000 replies 1000 test-duration 5.535 s

Connection rate: 18.1 conn/s (55.4 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 28.2 avg 55.4 max 90.1 median 53.5 stdev 6.2
Connection time [ms]: connect 5.2
Connection length [replies/conn]: 10.000

Request rate: 180.7 req/s (5.5 ms/req)
Request size [B]: 75.0

Reply rate [replies/s]: min 180.2 avg 180.2 max 180.2 stdev 0.0 (1 samples)
Reply time [ms]: response 5.0 transfer 0.0
Reply size [B]: header 284.0 content 177.0 footer 0.0 (total 461.0)
Reply status: 1xx=0 2xx=1000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 0.92 system 4.60 (user 16.5% system 83.1% total 99.7%)
Net I/O: 94.6 KB/s (0.8*10^6 bps)

Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
root@administrator-To-be-filled-by-O-E-M:/home/arsa#
    
```

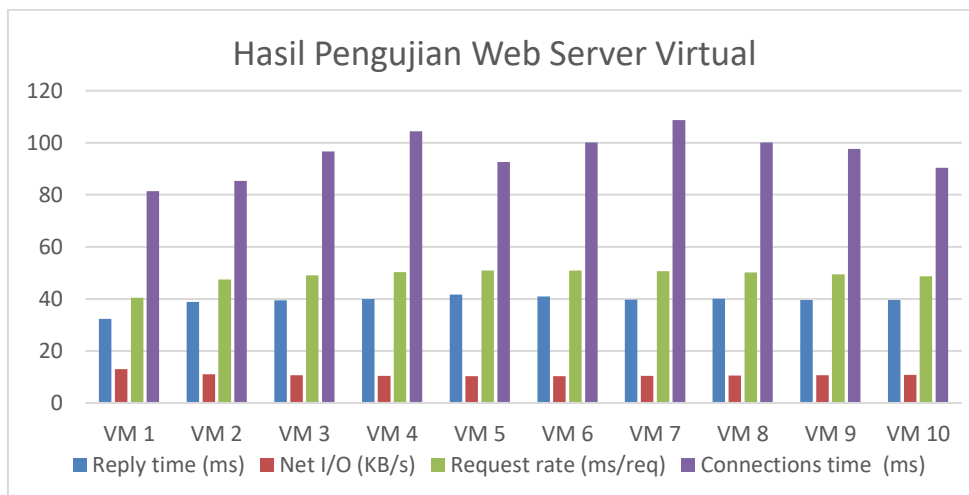
Gambar 4. Hasil Pengujian Satu VM Aktif

Gambar 5. Sample Gambar hasil pengujian dari 10 VM aktif

Setelah server Proxmox dibuat selanjutnya dibangun 10 server virtual. Selanjutnya sepuluh virtual server dikonfigurasi menjadi server web dan dilakukan uji web server dengan menggunakan tool Httpperf. Di mana dilakukan 100 connections dengan masing-masing 10 request per detik. Dapat dilihat rangkuman hasilnya pada Tabel 1 dan hasil grafik dapat dilihat pada Gambar 6.

Tabel 1. Rangkuman hasil pengujian

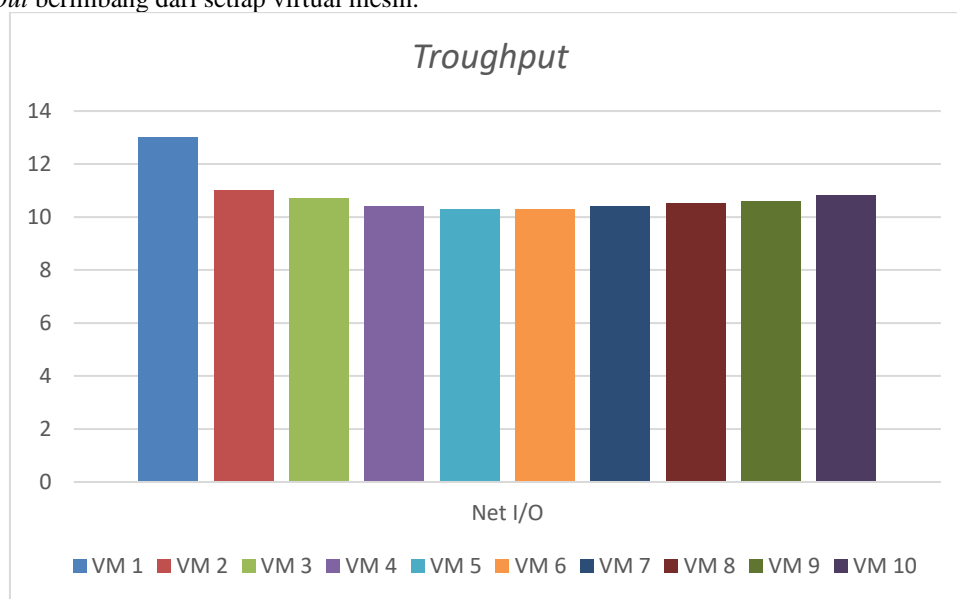
Virtual Server	Reply time (ms)	Net I/O (KB/s)	Request rate (ms/req)	Connections time (ms)	Test Duration (s)
VM 1	32.3	13	40.4	81.4	40.379
VM 2	38.8	11	47.4	85.3	47.373
VM 3	39.4	10.7	49.1	96.7	49.05
VM 4	39.9	10.4	50.3	104.4	50.35
VM 5	41.7	10.3	50.9	92.6	50.934
VM 6	40.9	10.3	50.9	100.1	50.919
VM 7	39.7	10.4	50.6	108.7	50.563
VM 8	40.1	10.5	50.1	100.1	50.072
VM 9	39.6	10.6	49.4	97.6	49.405
VM 10	39.6	10.8	48.7	90.4	48.668
Rerata	39.2	10.8	48.7	95.73	48.7713



Gambar 6. Grafik hasil pengukuran

3.4.1. *Troughput*

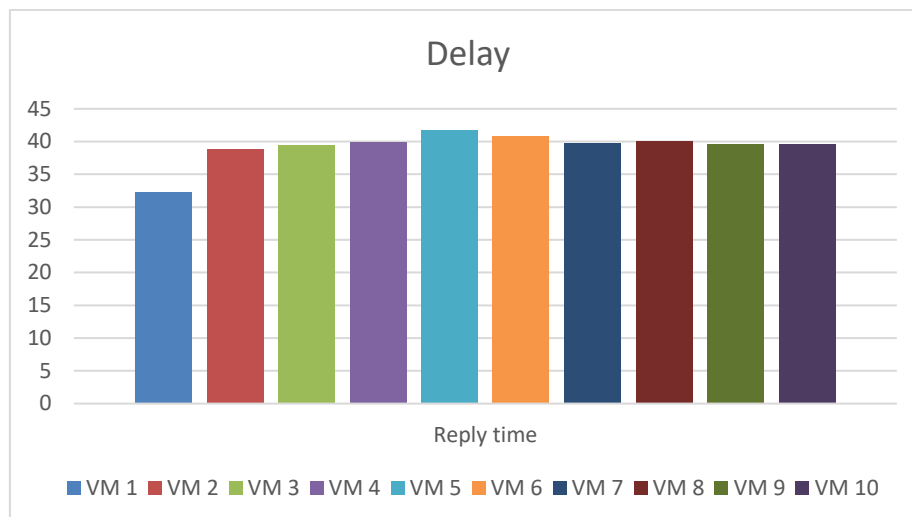
Data *troughput* diambil dari hasil pengukuran Net/IO pada Tabel 1 di mana rata-rata *Troughput* untuk 10 VM adalah 10,8 KB/s. Dapat dilihat grafik *Troughput* pada Gambar 7. Pada gambar terlihat nilai *troughput* berimbang dari setiap virtual mesin.



Gambar 7. Grafik hasil pengukuran *Troughput*

3.4.2. *Latency (Delay)*

Latency atau *delay* diambil dari *Replay Time* pada hasil pengukuran Tabel 1. Di mana rata-rata *delay* dari 10 mesin virtual adalah 39.2 ms. Hasil grafik pengukuran *delay* dapat dilihat pada Gambar 8. *Delay* pada grafik terlihat berimbang antar virtual mesin yang diujikan.



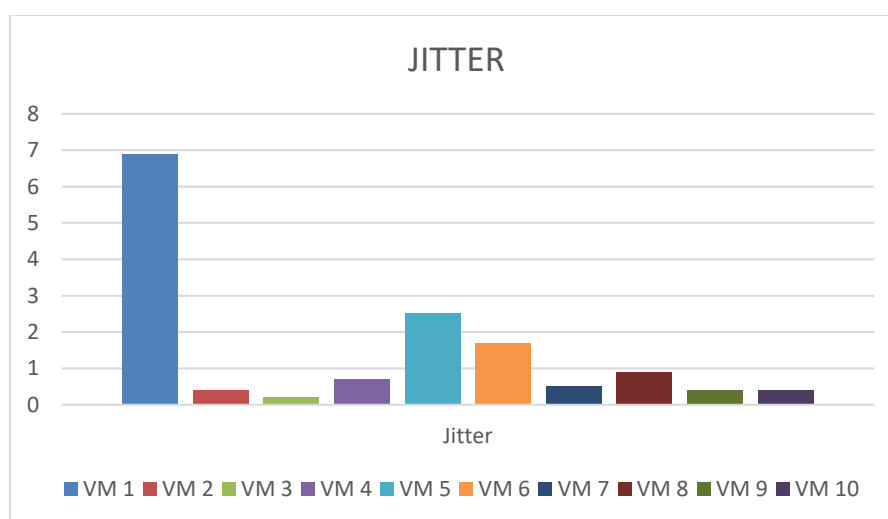
Gambar 8. Hasil pengukuran *Delay*

3.4.3. Jitter

Jitter adalah variasi *delay*, yang disebabkan oleh variasi-variasi panjang antrian dalam waktu mengolah data. Dari hasil pengujian diambil dari Total variasi *delay*, di mana total variasi *delay* didapat dari hasil absolut *delay* dikurangi dengan rata-rata *delay* (tanda – dihilangkan) maka hasilnya dapat dilihat pada Tabel 2. Rata-rata *Jitter* dari sepuluh VM adalah 1,46 ms. Grafik hasil pengujian dapat dilihat Gambar 9.

Tabel 2. Hasil Perhitungan *Jitter*

Virtual Server	Delay (ms)	Jitter (ms)
VM 1	32.3	6.9
VM 2	38.8	0.4
VM 3	39.4	0.2
VM 4	39.9	0.7
VM 5	41.7	2.5
VM 6	40.9	1.7
VM 7	39.7	0.5
VM 8	40.1	0.9
VM 9	39.6	0.4
VM 10	39.6	0.4
Rata-rata	39.2	1.46



Gambar 9. Hasil pengukuran *Jitter*

Variasi *delay* atau *jitter* pada Gambar 9 terlihat beberapa perbedaan yang signifikan seperti VM 1, 5, 6, dan 8 memiliki nilai *jitter* yang cukup tinggi.

3.4.4. Packet Loss

Berdasarkan hasil pengujian dengan Httperf dengan 100 *request* dan setiap *request* terdapat 10 permintaan pada satu virtual mesin (1000 *request*) didapati sesuai dengan hasil pengukuran pada Gambar 7. Jumlah *request* 1000 dengan *replay* 1000 dan total *request size* adalah 75 Byte, didapati *packet loss* adalah 0. Ini juga dibuktikan dengan jumlah *error* yang terjadi adalah 0. Hal ini bisa saja didapat karena pengujian dilakukan pada jaringan lokal sehingga hambatan yang ada tidak terlalu mengganggu proses komunikasi paket-paket data. Hal ini tentunya akan berbeda jika jaringan yang digunakan adalah internet.

3.5. Analisis QoS

Dalam menentukan standar *Quality of Service* s (QoS) salah satu yang digunakan adalah THIPON (*Telecommunications and Internet Protocol Harmonization Over Network*) TR.101329.V2.1.1.1999-06 yang dikeluarkan oleh ETSI (*European Telecommunications Standards Institute*) [9] nilai *Quality of Service* (QoS) dapat dilihat pada Gambar 10 [10] . Berdasarkan Gambar 10 maka dibuatlah standar pengategorian QoS sesuai pada Tabel 3 dan 4.

TIPHON QoS Class		Network Delay Requirements	
4 Best	<150 ms	<150 ms	<150 ms
3 High	<250 ms	<250 ms	<250 ms
2 Medium	<350 ms	<350 ms	<350 ms
1 Low	<450 ms	<450 ms	<450 ms
NOTE: Delay in both IWFs is included in the network delay requirements.			

Degradation Category	Packet loss (note 2)	Peak jitter(note 3)
Perfect	zero	0 ms
Good	3 %	75 ms
Medium	15 %	125 ms
Poor	25 %	225 ms

Gambar 10. Standar Nilai QoS menurut THIPON

Tabel 3. Indeks Parameter QoS (sumber: THIPON)

Nilai	Indeks
3,8 – 4	Sangat Memuaskan
3 – 3,79	Memuaskan
2 – 2,99	Kurang Memuaskan
1 – 1,99	Buruk

Tabel 4. Standar QoS menurut THIPON (sumber: THIPON)

Kategori	Throughput (%)	Jitter	Delay	Packet Loss	Indeks
Sangat Bagus	100	0 ms	<150 ms	0	4
Bagus	75	0-75 ms	<250 ms	3%	3
Sedang	50	76-122 ms	<350 ms	15%	2
Kurang	<25	123-225 ms	<450 ms	25%	1

Berdasarkan tabel standar QoS yang telah dibuat maka diketahui nilai *Throughput* yang diperoleh dari rumus $Throughput = \frac{\text{Paket data diterima}}{\text{lama pengamatan}}$ Diperoleh hasil berdasarkan Tabel 5.1 rata-rata *throughput* dan *test durations* adalah :

$$Throughput = \frac{\text{Paket data diterima}}{\text{lama pengamatan}} = \frac{10,8}{48,7713} = 0,221442 = 22\%$$

Dari hasil 22% diperoleh standar QoS THIPON masuk kategori kurang dengan indeks 1. Selanjutnya berdasarkan *Jitter* diperoleh rata-rata *Jitter* adalah 1,46 ms berada pada standar bagus dengan indeks 3. Selanjutnya untuk *delay* rata-rata 10 vm adalah 39,2 ms <150 ms sehingga masuk dalam kategori

sangat bagus dengan indeks 4. Sedangkan untuk *packet loss* sesuai dengan penjelasan sebelumnya yaitu 0 % dengan kategori sangat bagus dengan indeks 4. Simpulan hasil analisis dapat dilihat pada Tabel 5.

Tabel 5. Hasil QoS rata-rata 10 Virtual Mesin

Parameter Uji	Hasil Uji	Indeks	Standar TIPHON
Troughput	22%	1	Kurang
Jitter	1,46 ms	3	Bagus
Delay	39,2 ms	4	Sangat Bagus
Packet Loss	0%	4	Sangat Bagus
Rata-Rata		3	Memuaskan

4. Kesimpulan

Adapun kesimpulan yang diperoleh dari penelitian yang dilakukan adalah :

1. Dihasilkan sebuah arsitektur konsolidasi *server* dengan empat lapisan yaitu layer mesin virtual, layer mesin fisik, layer jaringan dan layer pengguna.
2. Dihasilkan topologi jaringan dengan konsolidasi *server*.
3. Dari hasil pengujian dan analisis QoS dengan standar THIPON diperoleh hasil rata-rata sepuluh mesin virtual untuk *throughput* adalah kurang, *Jitter* bagus, *delay* dan *packet loss* sangat bagus. Dengan indeks QoS masuk dalam kategori memuaskan dengan indeks 3.

Daftar Pustaka

- [1] P. Mell and T. Grance, "The NIST Definition of *Cloud* Computing Recommendations of the National Institute of Standards and Technology," NIST SP 800-145, 2011.
- [2] L. Tsai and W. Liao, *Virtualized Cloud Data Center Networks: Issues in Resource Management*. Cham: Springer International Publishing, 2016.
- [3] A. Varasteh and M. Goudarzi, "Server Consolidation Techniques in Virtualized Data Centers: A Survey," *IEEE Syst. J.*, vol. 11, no. 2, pp. 772–783, Jun. 2017, doi: 10.1109/JSYST.2015.2458273.
- [4] K. Ye *et al.*, "Profiling-Based Workload Consolidation and Migration in Virtualized Data Centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 3, pp. 878–890, Mar. 2015, doi: 10.1109/TPDS.2014.2313335.
- [5] R. Wulandari, "ANALISIS QoS (*QUALITY OF SERVICE*) PADA JARINGAN INTERNET (STUDI KASUS : UPT LOKA UJI TEKNIK PENAMBANGAN JAMPANG KULON – LIPI)," *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 162–172, 2016, doi: 10.28932/jutisi.v2i2.454.
- [6] H. U.-R. Qaiser, G. Shu, and A. W. Malik, "Utilization Driven Model for *Server* Consolidation in *Cloud* Data Centers," *IEEE Access*, vol. 8, pp. 1998–2007, 2020, doi: 10.1109/ACCESS.2019.2962272.
- [7] M. F. W. Simanjuntak, O. D. Nurhayati, and E. D. Widiyanto, "Analisis *Quality of Service* (QoS) Jaringan Telekomunikasi High-Speed Downlink Packet Access (HSDPA) pada Teknologi 3.5G," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 1, p. 67, Jan. 2016, doi: 10.14710/jtsiskom.4.1.2016.67-76.
- [8] S. W. P. Kusri and E. Pramono, "Analisis *Quality of Service* (QoS) Pada Jaringan Hotspot SMA Negeri XYZ," *e-Jurnal JUSITI (Jurnal Sist. Inf. dan Teknol. Informasi)*, vol. 7–2, pp. 142–152, Oct. 2018, doi: 10.36774/jusiti.v7i2.249.
- [9] Anonim, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of *Quality of Service* (QoS)," 1999.
- [10] S. W. Pamungkas and E. Pramono, "Analisis *Quality of Service* (QoS) Pada Jaringan Hotspot SMA Negeri XYZ," *J. Sist. Inf. dan Teknol. Inf.*, vol. 7, no. 2, pp. 142–152, 2018.