



## ***Analisis Throughput dan Skalabilitas Virtualized Network Function VyOS Pada Hypervisor VMware ESXi, XEN, dan KVM***

Revan Faredha Aswariza<sup>1</sup>, Doan Perdana<sup>2</sup>, Ridha Muldina Negara<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom  
<sup>1,2,3</sup>Jl. Telekomunikasi Terusan Buah Batu Bandung 40257 Indonesia  
Email korespondensi : [revan@datacommlab.id](mailto:revan@datacommlab.id)

Dikirim 24 Januari 2017, Direvisi 04 Februari 2017, Diterima 10 Februari 2017

Abstrak - Virtualisasi berjalan diatas suatu *hypervisor* yang merupakan suatu program untuk membuat dan menjalankan *virtual machine*. *Hypervisor* mengatur sejumlah *resources hardware* seperti RAM, CPU, dan *storage* yang dimiliki *hardware* aslinya untuk digunakan bersama-sama dengan *virtual environment*. Salah satu implementasi yang dapat dimanfaatkan dengan adanya virtualisasi adalah *Network Function Virtualization* atau yang biasa disebut NFW. Konsep ini memanfaatkan teknik virtualisasi untuk membuat suatu *Virtualized Network Function* (VNF) yang memiliki fungsi sama dengan *network device* aslinya. Salah satu VNF yang dapat digunakan secara bebas adalah VyOS. VyOS merupakan sistem operasi jaringan berbasis *open source* yang memiliki fungsi seperti *hardware router* tradisional, *firewall*, VPN, proxy dan fungsi jaringan lainnya. Pada penelitian ini dilakukan pengujian performansi VyOS pada *bare-metal hypervisor* (XEN, VMware ESXi) dan *hosted hypervisor* (*Kernel-based Virtual Machine* atau KVM). Penelitian ini bertujuan untuk mengetahui kinerja dari ketiga *hypervisor* tersebut dalam menjalankan VNF dengan parameter *throughput*, dan parameter skalabilitas. Dari hasil pengujian dan analisis dapat disimpulkan bahwa KVM memiliki performansi kecepatan tertinggi dengan besar *throughput* 19,29 Gbps. Sedangkan untuk parameter skalabilitas, VMware ESXi memiliki skalabilitas yang sangat baik yang ditunjukkan dengan tren penurunan *throughput* berkisar antara 0,034 – 0,1785 Gbps.

Kata Kunci - Virtualisasi, *Hypervisor*, *Network Function Virtualization*, VyOS.

Abstract - Virtualization is running on top of hypervisor which is a program for creating and running virtual machine. Hypervisor manages amount of hardware resources such as RAM, CPU, and storage owned by the original hardware to be used in conjunction with the virtual environment. One of the implementation that can be used with virtualization is the Network Function Virtualization or commonly called NFW. This concept utilizes virtualization technology to create a Virtualized Network Function (VNF), which has the same function with the original network device. One of VNF that can be used freely is VyOS. VyOS is an open source network operation system that has a same function as traditional router hardware, firewall, VPN, proxy, and other network functions. In this research, tests performed on VyOS to measure performance while it runs on a bare-metal hypervisor (XEN, VMware ESXi) and hosted hypervisor (*Kernel-based Virtual Machine*, or KVM). This research aims to determine the performance of these hypervisors in running VNF with the parameter of throughput, and scalability. From the test results and analysis, it can be concluded that the KVM had a great performance with a top speed of throughput 19.29 Gbps. As for the parameters of scalability, VMware ESXi has excellent scalability shown by its slightly performance degradation approximately 0.034 – 0.1785 Gbps.

Keywords - Virtualization, Hypervisor, Network Function Virtualization, VyOS.

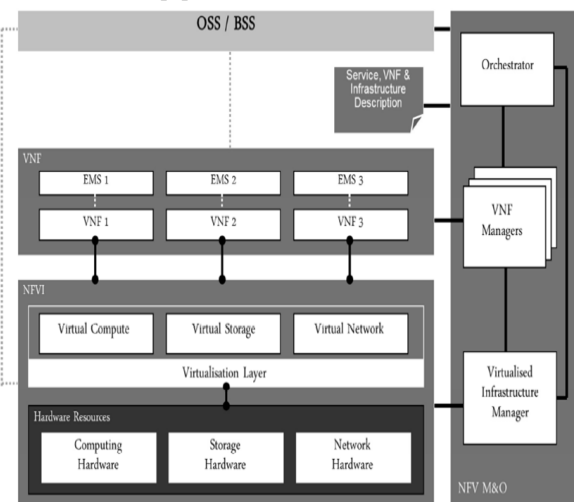
### I. PENDAHULUAN

*Software* virtualisasi mengabstraksi *hardware* dengan menciptakan sebuah *interface* untuk *virtual machine* (VMs), yang merupakan *virtualized*

*resources* seperti CPU, RAM, koneksi jaringan, dan blok perangkat lainnya. Kumpulan perangkat lunak termasuk OS dan aplikasi dieksekusi di atas VMs dan beberapa VMs dapat berjalan secara bersamaan pada

server fisik tunggal [1]. Virtualisasi berjalan diatas suatu *hypervisor* yang mengatur seluruh isolasi dan manajemen *hardware* yang digunakan untuk *virtual environment*. Beberapa *hypervisor* terkenal seperti XEN, VMware, dan KVM sering menjadi bahan perbincangan mengenai siapa yang terbaik dalam hal stabilitas dan fleksibilitas. Xen Project™ adalah *platform* virtualisasi *open source* yang mendukung beberapa *cloud* terbesar dalam produksi saat ini. Amazon Web Services, Aliyun, Rackspace Cloud Umum, Verizon Cloud dan banyak layanan *hosting* menggunakan *software* Xen[2]. VMware ESXi merupakan *platform* besar lainnya yang mendukung layanan *cloud* sekarang ini. KVM menjadi salah satu opsi yang paling banyak digunakan untuk skala *private* karena fleksibilitas yang tinggi dan dapat berjalan diatas mesin dengan prosesor standar dukungan Intel VT-x / AMD-V.

NFV (*Network Function Virtualization*) mengurangi biaya operasi, meningkatkan pengelolaan dan inovasi di ruang layanan fungsi jaringan. Teknologi NFV mengenalkan konsep baru dalam merancang, menyebarkan dan mengelola layanan jaringan dengan cara memisahkan fungsi jaringan dari peralatan *hardware proprietary* sehingga dapat berjalan pada suatu *hypervisor* atau *platform* virtualisasi [3]. IT Profesional, *developers*, dan pengguna virtualisasi lainnya sering mencari hasil kuantitatif untuk membandingkan *hypervisor-hypervisor* yang ada [4]. Namun dengan banyaknya pilihan *hypervisor* yang ada, membuat pengguna tidak dapat menentukan *hypervisor* yang tepat untuk mengimplementasikan NFV. NFV merupakan konsep yang membangun VNF diatas suatu infrastruktur bernama NFVi[5].



Gambar 1. Arsitektur NFV Berdasarkan ETSI

Berdasarkan penelitian yang dilakukan oleh badan standarisasi *European Telecommunications Standards Institute* (ETSI) bersama vendor-vendor telekomunikasi terbesar di dunia seperti AT&T, Deutsche Telekom, Telstra, Vodafone, dan lainnya dalam paper *Network Functions Virtualization White*

*Paper#3* pada *October 14-17, 2014 at the "SDN & OpenFlow World Congress", Dusseldorf-Germany* menjelaskan bahwa Infrastruktur NFV (NFVI) diperlukan untuk mendukung berbagai kasus penggunaan dan bidang aplikasi yang telah diidentifikasi oleh *NFV Industry Specification Group* (ISG) dengan menyediakan *platform* yang stabil untuk evolusi ekosistem VNF. Pada Gambar 1, NFVi berada pada layer elementer di arsitektur NFV yang dikeluarkan oleh ETSI. NFVi merupakan kumpulan dari komponen *hardware* dan *software* yang mengabstraksi *hardware* untuk lingkungan *virtual* [6]. NFVI menyediakan layanan infrastruktur multi-tenant dengan memanfaatkan teknologi virtualisasi yang dapat mendukung beberapa kasus penggunaan dan bidang aplikasi secara bersamaan [5][7][8][9][10].

Penelitian sebelumnya [11] menganalisis tiga solusi NFV (ClickOS, CoreOS, dan OSV) pada perspektif manajemen jaringan. Penelitian ini fokus pada metrik yang terkait dengan kinerja penyebaran dan *monitoring* VNFs seperti mengevaluasi waktu untuk menjalankan mesin *virtual* dan *container*.

Penelitian selanjutnya [12] menganalisis performansi NFV dengan membandingkan antara teknologi *docker* dan XEN dari perspektif performansi jaringan seperti performansi QoS seperti *packet delay*, *latency*, *round trip time* dan lainnya.

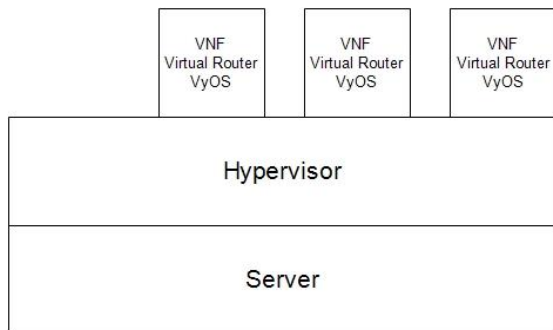
Penelitian berikutnya [13] menganalisis performansi NFV pada KVM, VMware, XEN, dan *container* menggunakan VNF ClickOS dengan *metric* performansi *latency*, *throughput*, dan *jitter*. Penelitian ini menghasilkan bahwa selain parameter *latency*, *throughput*, dan *jitter*, parameter skalabilitas sangatlah penting untuk diuji karena dapat mempengaruhi performansi VNF yang lainnya.

Berdasarkan latar belakang dan penelitian-penelitian sebelumnya, penulis terdorong untuk melakukan penelitian NFV berbasis teknologi virtualisasi *hypervisor* XEN, KVM, dan VMware ESXi dengan parameter *throughput* dan keterkaitan antara *throughput* dengan skalabilitas.

## II. METODE PENELITIAN

Pada bab ini akan dibahas perancangan dan implementasi *server* untuk menjalankan VNF berupa *virtual router* pada setiap *hypervisor*. XEN dan VMware ESXi merupakan *baremetal hypervisor* sedangkan KVM merupakan *hosted hypervisor*. Setiap *hypervisor* diimplementasikan menggunakan *harddisk* yang sama, artinya setelah selesai pengujian, seluruh data di *harddisk* akan dihapus dan instalasi dilanjutkan dengan *hypervisor* berikutnya. VNF yang dijalankan berupa fitur *virtual router* dari VyOS yang telah dimodifikasi agar seluruh aplikasi yang dibutuhkan selama pengujian telah tersedia pada VNF. VNF disimpan dalam bentuk *virtual harddisk* berekstensi *.ova* sehingga dapat dengan mudah dimigrasikan baik ke *hypervisor* XEN, VMware ESXi maupun KVM

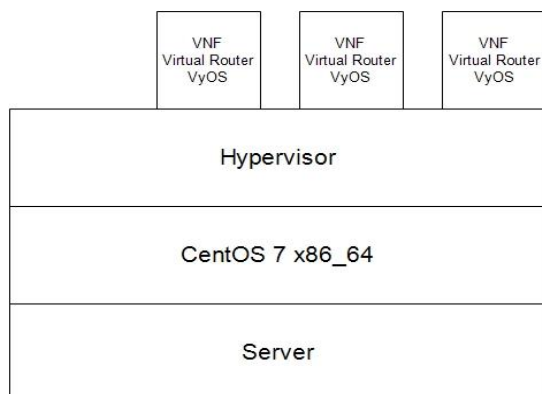
menggunakan aplikasi *manager* masing-masing *hypervisor*. Dalam melakukan migrasi, terdapat beberapa optimasi konfigurasi yang dilakukan sesuai dengan rekomendasi dari aplikasi *manager* yang digunakan. Setelah Vyatta OS dapat dijalankan, kemudian VNF akan diuji performansinya meliputi *throughput* dan skalabilitas. *Server* yang digunakan adalah HP Proliant DL180 Gen9 dengan prosesor 1,90 GHz Intel Xeon E5-2609 v3 (6 Cores), 24 GB RAM, 1000 GB hard drive, dan dua kartu Gigabit Ethernet.



Gambar 2. Arsitektur Bare-Metal Hypervisor

Pada Gambar 2, arsitektur *bare-metal hypervisor* tidak membutuhkan *host OS* pada saat diimplementasikan dan langsung berjalan di atas *hardware server*. VNF VyOS akan dijalankan di atas *hypervisor* dan diuji performansinya.

Pada Gambar 3, arsitektur *hosted hypervisor* diimplementasikan di atas suatu *host OS* dan pada penelitian ini digunakan CentOS 7 dengan alasan untuk mengimbangi kemampuan XEN dan VMware ESXi yang memiliki kelas *Enterprise*. Skenario uji performansi akan dibuat sama dengan uji pada *bare-metal hypervisor*.



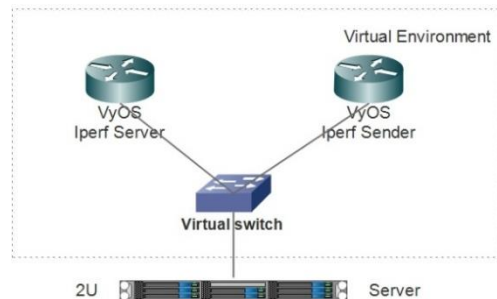
Gambar 3. Arsitektur Hosted Hypervisor

Pada penelitian ini dilakukan pengujian pada setiap *hypervisor* yang menjalankan VNF VyOS. Pengujian dilakukan dengan beberapa skenario yang bertujuan untuk mengetahui performansi *throughput* dan skalabilitas. Jumlah VNF yang dibuat sebanyak dua VNF untuk pengujian *throughput* sedangkan untuk pengujian skalabilitas dilakukan penambahan jumlah VNF hingga lima VNF pada masing-masing

*hypervisor*. Berikut skenario pengujian yang dilakukan.

#### A. Throughput

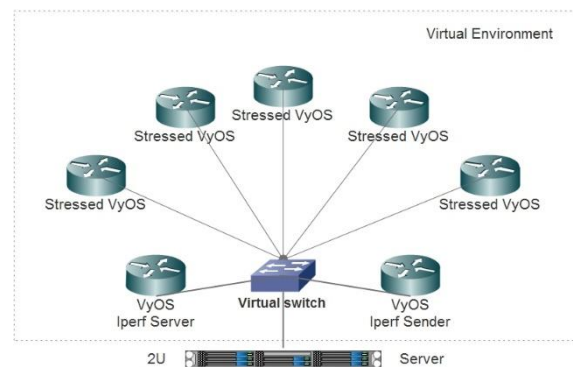
*Throughput* adalah jumlah data per satuan waktu yang dikirim untuk suatu terminal tertentu di dalam sebuah jaringan, dari suatu titik jaringan, atau dari suatu titik ke titik jaringan yang lain. Pengujian *throughput* dilakukan dengan topologi pada Gambar 4 dan menggunakan aplikasi *iperf* dengan protokol TCP [14].



Gambar 4. Topologi Pengujian Throughput

#### B. Skalabilitas

Pengujian dilakukan untuk mengukur kemampuan *hypervisor* dalam melindungi atau mengisolasi performansi VNF dari penggunaan *resource* yang tinggi dari VNF lainnya. VNF akan terus ditambah hingga lima VNF yang dijalankan secara bersamaan. Pada pengujian ini dilakukan pengujian *throughput* pada VNF *Iperf server* dan VNF *Iperf sender* lalu dilakukan penambahan *stressed VNF* baru dengan *workload* hingga mengkonsumsi CPU sebesar 99%.



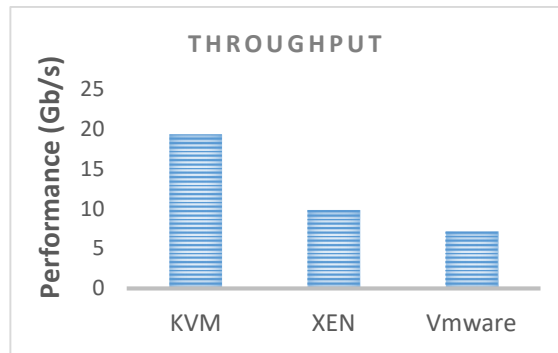
Gambar 5. Topologi Pengujian Skalabilitas

### III. HASIL DAN PEMBAHASAN

Pada bab ini dibahas analisis dari hasil pengujian dengan skenario dan parameter yang telah ditentukan, dengan mengimplementasikan NFV pada *hypervisor* XEN, KVM, dan VMware ESXi. Sebagai pembandingan untuk melihat performansi asli dari *server* yang kemudian disebut *Native*, diimplementasikan sistem operasi CentOS 7 dengan alasan untuk mengimbangi kemampuan stabilitas dan performansi kelas *Enterprise* dari XEN dan VMware ESXi.

### A. Throughput

Berdasarkan grafik hasil pengujian pada Gambar 6, terlihat bahwa KVM memiliki *throughput* yang sangat tinggi dengan nilai 19,29 Gbps, sedangkan XEN memiliki nilai 9,83 Gbps dan VMware ESXi memiliki nilai 7,18 Gbps. Dengan adanya perbedaan *throughput* tersebut maka terjadi pula perbedaan pada konsumsi CPU saat pengujian *throughput* yang ditunjukkan pada Tabel 1.



Gambar 6. Grafik Hasil Pengujian *Throughput*

Tabel 1. Hasil Pengujian *Throughput*

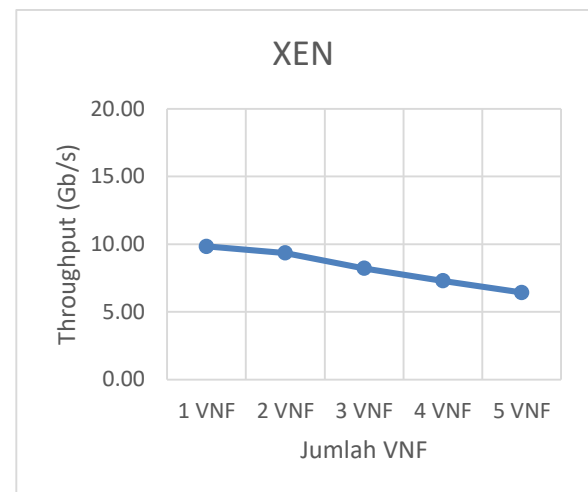
	<i>Throughput</i>		
<i>Hypervisor</i>	KVM	XEN	Vmware
<i>Performance (Gb/s)</i>	19,29	9,83	7,18
<i>CPU Load Average (%)</i>	83,51	65,03	60,38

Pada pengujian *throughput*, paket yang dilewatkan dibuat tidak terbatas sehingga akan menyesuaikan dengan besarnya *bandwidth* yang dapat dilewati. Akan tetapi setiap *hypervisor* memiliki hasil yang berbeda dari nilai *throughput* dan besarnya penggunaan CPU saat pengujian *throughput* berlangsung. Perbedaan penggunaan CPU ini memerlukan analisis lebih lanjut mengenai limitasi *bandwidth* yang diterapkan pada masing-masing *hypervisor*. Namun mengacu pada hasil pengujian ini, KVM memiliki limitasi *bandwidth* yang lebih besar dibanding dengan XEN dan VMware ESXi walaupun berdampak pada penggunaan CPU yang lebih besar pula. Besarnya limitasi *bandwidth* ini menjadi sebuah *trade off* antara performansi dengan beban kerja CPU.

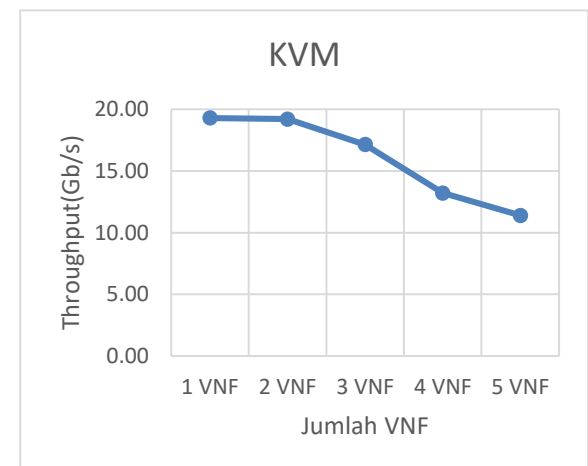
### B. Skalabilitas

Berdasarkan grafik hasil pengujian pada Gambar 10, terlihat bahwa *hypervisor* yang memiliki skalabilitas terbaik adalah VMware ESXi, lalu XEN, dan yang terakhir adalah KVM. Penurunan atau degradasi performansi yang terjadi diakibatkan oleh VNF baru yang diberikan *workload* sangat tinggi hingga mempengaruhi *throughput* pada VNF yang diuji *throughputnya* [11]. KVM memiliki degradasi performansi yang sangat tinggi atau penurunan performansi berkisar antara 0,085 – 3,932 Gbps. Hal ini disebabkan oleh pembagian *resource* dengan *operating system* non virtualisasi yang digunakan pada

saat menjalankan KVM. Solusi agar KVM memiliki performansi yang lebih baik adalah *tuning* atau optimasi terhadap perangkat *virtual* pada KVM. Sedangkan XEN memiliki degradasi performansi yang tidak terlalu tinggi yaitu berkisar antara 0,485 – 1,138 Gbps hal ini disebabkan oleh kemampuan *kernel* XEN yang terbatas dalam alokasi *resource* antar VNF. VMware ESXi memiliki degradasi performansi yang paling kecil yaitu penurunan *throughput* yang berkisar antara 0,034 – 0,1785 Gbps dan menunjukkan bahwa *kernel* VMware ESXi memiliki fitur alokasi *resource* dan isolasi performansi yang lebih baik dibandingkan dengan XEN dan KVM.



Gambar 7. Grafik Hasil Pengujian Skalabilitas Pada *Hypervisor* XEN

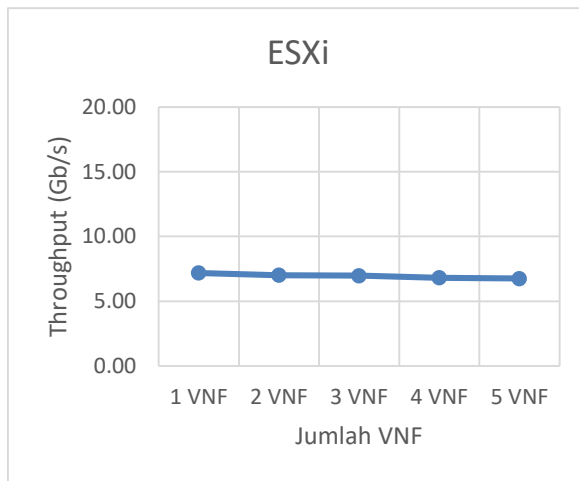


Gambar 8. Grafik Hasil Pengujian Skalabilitas Pada *Hypervisor* KVM

Tabel 2. Hasil Pengujian Skalabilitas

<i>Hypervisor</i>	<i>Throughput berdasarkan jumlah VNF yang dijalankan (Gbps)</i>				
	1	2	3	4	5
ESXi	7,18	7,01	6,97	6,81	6,75
XEN	9,83	9,35	8,21	7,29	6,42

Hypervisor	Throughput berdasarkan jumlah VNF yang dijalankan (Gbps)				
	1	2	3	4	5
KVM	19,29	19,21	17,15	13,21	11,40



Gambar 9. Grafik Hasil Pengujian Skalabilitas Pada Hypervisor VMware ESXi

#### IV. KESIMPULAN

##### A. Kesimpulan

Dari hasil pengujian dan analisis performansi NFV pada hypervisor XEN, KVM, dan VMware ESXi, VNF dapat diimplementasikan di ketiga hypervisor dengan perbedaan performansi berdasarkan parameter throughput dan skalabilitas. Berdasarkan parameter throughput, KVM merupakan hypervisor yang memiliki komunikasi internal network yang paling baik dari segi kecepatan dengan kecepatan maksimum 19.29 GB/s. Selain itu, throughput mempengaruhi beban CPU dalam mengolah paket data, semakin banyak data yang dilewatkan maka semakin besar konsumsi CPU-nya. Pada parameter skalabilitas hasilnya menunjukkan bahwa VMware ESXi memiliki skalabilitas yang sangat baik yang ditunjukkan dengan tren penurunan throughput paling sedikit dibanding dengan XEN dan KVM dengan demikian konsumsi ekstrem pada suatu VNF tidak begitu mengganggu performansi VNF yang lainnya.

##### B. Saran

Sebagai kelanjutan dari penelitian ini, disarankan untuk melakukan optimasi pada masing-masing hypervisor agar mendapatkan hasil performansi yang lebih baik. Selain itu, penelitian selanjutnya disarankan untuk menggunakan VNF yang terbaru sehingga kernel yang digunakan dapat memaksimalkan kemampuan hardware dari server.

#### DAFTAR PUSTAKA

- [1] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A. "Xen and the art of virtualization". In ACM SIGOPS Operating Systems Review (Vol. 37, No. 5, pp. 164-177). ACM. 2003.
- [2] Linux Foundation. "Xen To Become Linux Foundation Collaborative Project | Linux Foundation". Web. 20 Dec. 2016.
- [3] Garg, A., "Network Function Virtualization (Doctoral dissertation, Indian Institute of Technology Bombay)".
- [4] Deshane, T., Shepherd, Z., Matthews, J., Ben-Yehuda, M., Shah, A. and Rao, B., "Quantitative comparison of Xen and KVM. Xen Summit". Boston, MA, USA, pp.1-2. 2008.
- [5] European Telecommunications Standards Institute. Network Functions Virtualisation (NFV); White Paper #2. 2013.
- [6] Thekkedath, Balamurali. "Network Functions Virtualization For Dummies®, Hewlett Packard Enterprise Special Edition". 1st ed. John Wiley & Sons, Inc., Web. 16 Oct. 2016.
- [7] European Telecommunications Standards Institute. "Network Functions Virtualisation (NFV); White Paper #1". 2012.
- [8] European Telecommunications Standards Institute. Network Functions Virtualisation (NFV); Architectural Framework. GS NFV 002 (V1.1.1). 2013.
- [9] European Telecommunications Standards Institute. 2013. "Network Functions Virtualisation (NFV); Use Cases". GS NFV 001 (V1.1.1).
- [10] European Telecommunications Standards Institute. 2014. "Network Functions Virtualisation (NFV); White Paper #3".
- [11] V. K. Quintana Rodriguez and F. Guillemin, "Performance analysis of resource pooling for network function virtualization," 17th International Telecommunications Network Strategy and Planning Symposium (Networks), Montreal, QC, 2016, pp. 158-163. 2016.
- [12] J. Anderson, H. Hu, U. Agarwal, C. Lowery, H. Li and A. Apon, "Performance considerations of network functions virtualization using containers," International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, 2016, pp. 1-7. 2016.
- [13] C. Wang, O. Spatscheck, V. Gopalakrishnan, Y. Xu and D. Applegate, "Toward High-Performance and Scalable Network Functions Virtualization," in IEEE Internet Computing, vol. 20, no. 6, pp. 10-20, Nov.-Dec. 2016.
- [14] Perera, P.M. and Keppitiyagama, C., September. "A performance comparison of hypervisors". In Advances in ICT for Emerging