

ANALISA PENGGUNAAN SUMBER DAYA JARINGAN PADA MIGRASI APLIKASI DALAM *LINUX CONTAINER (LXD)* MENGGUNAKAN *LXD API*

ANALYSIS OF THE USE OF NETWORK RESOURCES IN APPLICATION MIGRATION IN LINUX CONTAINER (LXD) USING LXD API

Helmi Khairullah Setiana¹, Adityas Widjajarto, S.T., M.T.², Avon budiono, S.T., M.T.³.

^{1,2,3}Prodi S1 Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom

¹helmikhairullah@student.telkomuniversity.ac.id, ²avonbudi@telkomuniveristy.co.id,

³avonbudi@telkomuniversity.ac.id

Abstrak

Linux container merupakan salah satu container yang dibangun di atas linux fitur kernel. *Linux Container* ini dapat digunakan sebagai *virtual* di dalam suatu *host*. Kegunaan dari hal ini adalah kita bisa menggunakan *linux container* seperti *host* sendiri. Salah satu keuntungan dengan menggunakan *linux container* adalah yaitu lebih sederhana, lebih cepat, lebih aman, dan dapat diukur. Dengan menggunakan *container*, dapat digunakan dalam migrasi aplikasi dari satu *linux container* ke *linux container* yang lain. Selain hal mudah tersebut, *linux container* dapat juga digunakan di jaringan *eksternal*. Salah satu contoh penggunaannya adalah di dalam suatu *virtual private server*. didalam *virtual private server*, *linux container* dapat berjalan dengan baik. Dengan mulai berjalannya migrasi didalam suatu *virtual private server*, terdapat prokol jaringan yang ada didalamnya. Dengan adanya protokol jaringan tersebut dapat menghitung *quality of service* dari suatu *virtual private server* ini. *Quality of service* yang dapat di cari adalah *throughput*, *propagation delay*, *packet loss*, *jitter*.

Kata kunci: *linux container*, *quality of service*, *throughput*, *packet loss*, *end to end delay*, *jitter*.

Abstract

The Linux container is one container that is built on the Linux kernel feature. This Linux Container can be used as a virtual inside a host. The usefulness of this is that we can use linux containers like their own hosts. One of the advantages of using a linux container is that it is simpler, faster, safer, and can be used. By using a container, it can be used in promotional applications from one linux container to another linux container. Besides this, Linux containers can also be used on external networks. One example of its use is on a virtual private server. inside the private server the virtual linux container can run properly. By starting it on a virtual private server, there is a network protocol in it. With this network protocol, it can protect the quality of services from this virtual private server. The quality of service that can be searched for is throughput, propagation delay, packet loss, jitter.

Keywords: *linux container*, *quality of service*, *throughput*, *packet loss*, *end to end delay*, *jitter*.

1. Pendahuluan

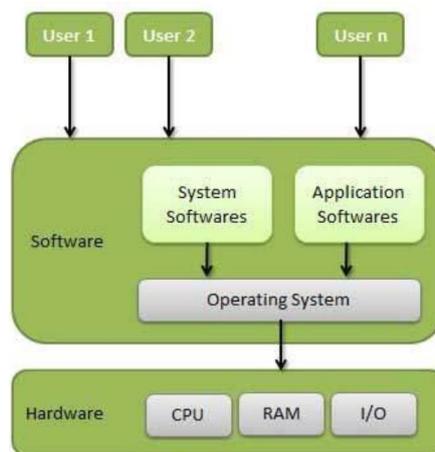
Linux container merupakan salah satu container yang dibangun di atas linux fitur kernel. *Linux Container* ini dapat digunakan sebagai *virtual* di dalam suatu *host*. Kegunaan dari hal ini adalah kita bisa menggunakan *linux container* seperti *host* sendiri. Selain hal tersebut *linux container* pun bisa digunakan dalam melakukan proses suatu migrasi aplikasi. Banyak orang lupa akan melakukan proses *backup* sehingga mereka kehilangan suatu hal yang disimpan di dalam suatu *host*. Maka dari itu *linux container* hadir untuk mengatasi solusi tersebut. Banyak sekali keuntung dengan menggunakan *Linux container*, yaitu lebih sederhana, lebih cepat, lebih aman, dan dapat diukur. Dengan beberapa keuntungan tersebut, *user* dapat menggunakan container ini dalam penginstalan aplikasi dan dapat dengan mudah melakukan migrasi aplikasi tersebut dengan sangat mudah.

2. Dasar Teori

2.1 Sistem Operasi

Sistem operasi adalah suatu *software* yang mengatur perangkat keras dari suatu komputer [1]. Sistem operasi menyediakan beberapa aplikasi agar bisa digunakan oleh *user* dan sebagai perantara antara *user* dengan perangkat keras dari suatu komputer [1]. Tujuan dari sistem operasi adalah untuk membuat *user* dapat menjalankan program dengan nyaman dan efisien [1]. Sistem operasi pertama kali dibuat hanya sebagai pelaksana dan penanganan program aplikasi pengoperasian perangkat keras pada mesin [4]. Sistem operasi hanya menampilkan tampilan yang kurang menarik. Sistem operasi modern ini sudah menampilkan tampilan yang sangat menarik, sehingga para *user* hanya tinggal klik saja untuk membuka aplikasi tanpa harus mengetikkan perintah terlebih dahulu. Sudah banyak sekali sistem operasi yang ada di dunia ini. Berbagai macam – macam tipe sistem operasi pun sudah banyak di dunia ini.

Mulai dari tipe sistem operasi windows, linux, android, ios, dan lain-lain. Selain tipe yang beragam, *color os* dan versi dari setiap sistem operasi pun beragam. Secara tidak langsung, sistem operasi sudah sangat dekat dengan penggunaannya. Mulai dari komputer, *handphone*, *hand held device*, dan lain-lain. Perangkat keras tersebut pun sudah terpasang sistem operasi agar para *user* dapat berinteraksi dengan perangkat keras yang mereka gunakan.

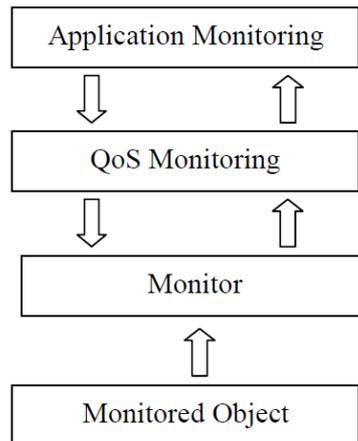


Gambar 2.1 Komponen dari sistem komputer.
(Sumber: (Tutorials Point, 2016)).

Berdasarkan gambar diatas terlihat bahwa komponen dari sistem komputer terbagi menjadi dua, yaitu *software* dan *hardware*. sistem operasi menghubungkan antara perangkat lunak dengan perangkat keras. Jika tidak ada sistem operasi, maka komponen sistem pada komputer tidak akan berjalan dengan baik. Sistem operasi pun juga membuat nyaman penggunaannya. Didalam sistem operasi berisi perangkat lunak yang dapat digunakan untuk menunjang kegiatan yang berada di dalam perangkat keras. Sedangkan komponen yang ada di dalam perangkat keras ada *CPU*, *RAM*, dan *I/O*.

2.2 Quality of service

Quality of Service adalah suatu metode yang digunakan dalam mengukur seberapa baik jaringan di dalam suatu layanan agar dapat menentukan karakteristik jaringan yang ada di dalam suatu layanan [2] [3]. Bisa di katakan bahwa *quality of service* adalah kemampuan dalam menyediakan jaminan kinerja dalam suatu jaringan di dalam suatu layanan [6]. Performa yang diukur dalam *quality of service* adalah pengiriman data dalam suatu jaringan. Seberapa baik atau buruk kah performa jaringan di dalam suatu layanan pada saat pengiriman data.



Gambar 2.3 Model Monitoring QoS.

(Sumber: (Hasanul, 2018)).

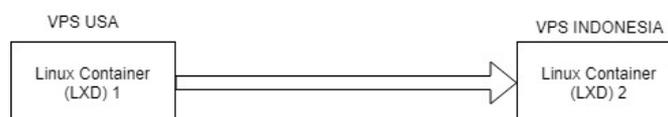
Berdasarkan gambar diatas bisa diketahui bahwa *quality of service* memiliki model dalam memonitoringnya. Hal pertama yang dilakukan adalah memonitor objek yang akan di monitor. Langkah kedua adalah memulai monitor. Langkah ketiga yaitu mulai menentukan dan melakukan *qos* monitoring apa yang akan di teliti. Langkah terakhir adalah aplikasi monitoring apa yang di pakai dalam menganalisa *qos*. Berdasarkan hal tersebut, dapat diketahui bahwa kinerja dari jaringan komputer terdapat beberapa masalah yang ada. Contoh masalah yang ada adalah *throughput* yang besar, *packet loss* yang besar, *End to end delay* yang lama, dan *jitter* yang lama [6]. Maka dari itu *quality of service* ini sangat penting dilakukan dalam menentukan kinerja jaringan komputer agar para *user* dapat mengetahui dan mengantisipasi hal tersebut.

3. Skenario dan Hasil Penelitian

3.1. Skenario

Skenario merupakan tahapan proses bagaimana cara dalam melakukan riset didalam suatu penelitian yang dilakukan. Terdapat empat skenario yang dilakukan dalam penelitian ini. Empat scenario tersebut adalah sebagai berikut:

3.1.1 Skenario 1



Gambar 3.1.1 Skenario satu.

Pada skenario pertama ini yaitu dengan mengirim container satu ke container dua. Pada saat proses migrasi ini dilakukan pengambilan data jaringan dengan menggunakan *tcpdump*. Setelah didapatkan data melalui *tcpdump*, lalu di lihat dan di analisis dengan menggunakan *wireshark*.

3.1.2 Skenario 2

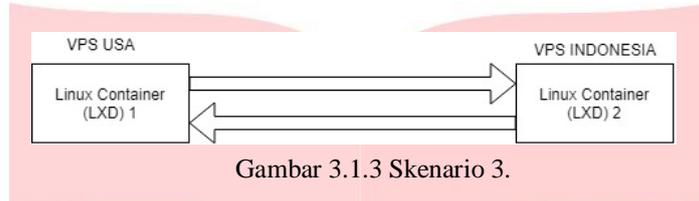


Gambar 3.1.2 Skenario dua.

Pada skenario kedua ini yaitu dengan mengirim kembali container yang telah di migrasi pada skenario satu. Pada saat proses migrasi ini dilakukan pengambilan data jaringan dengan

menggunakan *tcpdump*. Setelah didapatkan data melalui *tcpdump*, lalu di lihat dan di analisis dengan menggunakan *wireshark*.

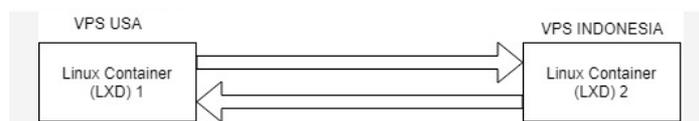
3.1.3 Skenario 3



Gambar 3.1.3 Skenario 3.

Pada skenario ketiga ini yaitu dengan cara saling mengirim container. Pada saat proses migrasi ini dilakukan pengambilan data jaringan dengan menggunakan *tcpdump*. Setelah didapatkan data melalui *tcpdump*, lalu di lihat dan di analisis dengan menggunakan *wireshark*.

3.1.4 Skenario 4



Gambar 3.1.4 Skenario 4.

Pada skenario keempat ini yaitu dengan cara saling mengembalikan container yang telah di migrasi pada skenario tiga. Pada saat proses migrasi ini dilakukan pengambilan data jaringan dengan menggunakan *tcpdump*. Setelah didapatkan data melalui *tcpdump*, lalu di lihat dan di analisis dengan menggunakan *wireshark*.

3.2. Hasil Penelitian

Hasil penelitian didapatkan setelah melakukan proses riset sesuai dengan skenario yang telah dilakukan. Hasil penelitian ini didasarkan pada aplikasi yang diisikan pada container. Penelitian ini bertujuan untuk mengetahui *quality of service* dari suatu jaringan *virtual private server*. Berikut ini merupakan hasil penelitian yang telah dilakukan :

3.2.1 Container dengan tidak ada aplikasi

A.) Skenario 1

Througput	678.114 bit
Packet loss	7,30%
End to end delay	14,714 ms

B.) Skenario 2

Througput	31.301.018 bit
Packet loss	1,65%
End to end Delay	0,706 ms

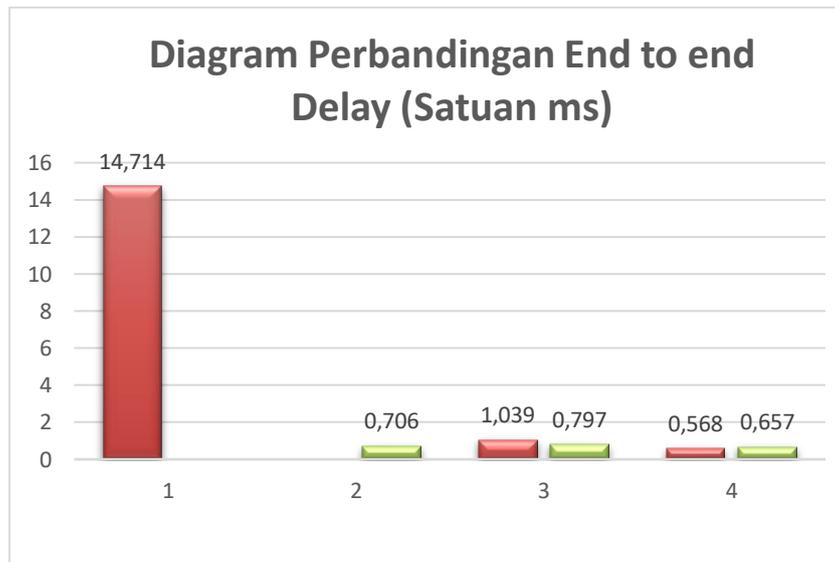
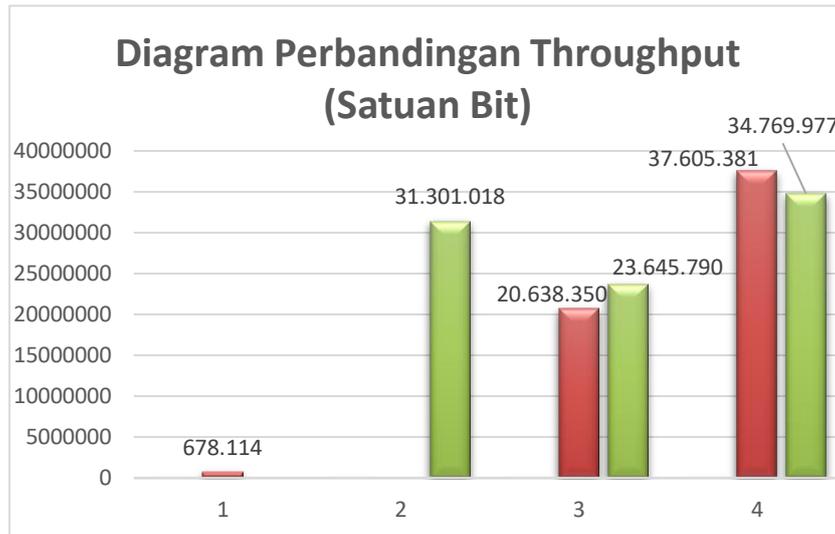
C.) Skenario 3

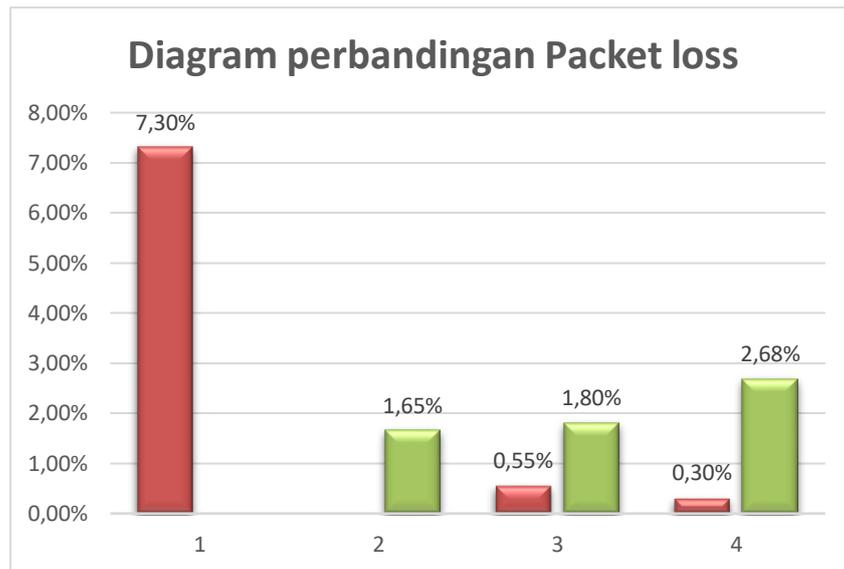
	VPS USA	VPS INDONESIA
Througput	20.638.350 bit	23.645.790 bit
Packet loss	0,55%	1,80%
End to end Delay	1,039 ms	0,797 ms

D.) Skenario 4

	VPS USA	VPS INDONESIA
Throughput	37.605.381 bit	34.769.977 bit
Packet loss	0,30%	2,68%
End to end Delay	0,568 ms	0,657 ms

Berdasarkan hasil penelitian di atas, berikut ini adalah diagram hasil penelitiannya :





3.2.2 Container dengan ada 1 aplikasi (odoo)

A.) Skenario 1

Througput	40.588.020 bit
Packet loss	0,29%
End to end delay	0,545ms

B.) Skenario 2

Througput	44.472.474 bit
Packet loss	1,83%
End to end delay	0,585 ms

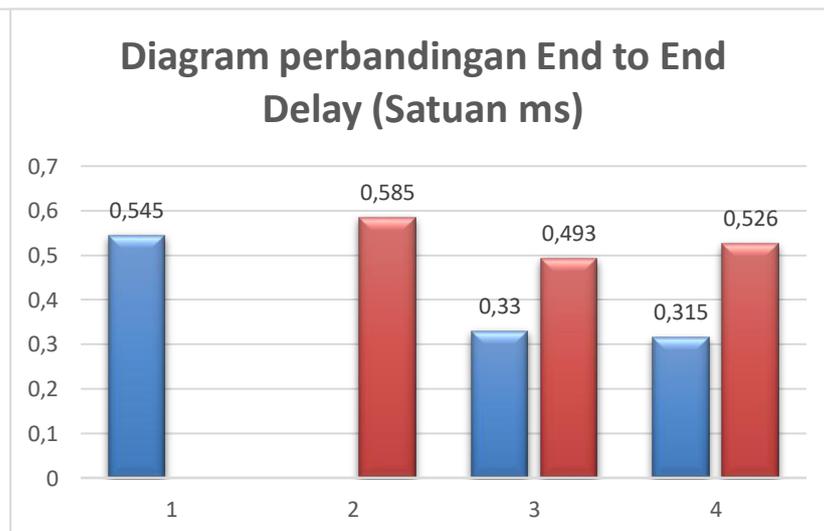
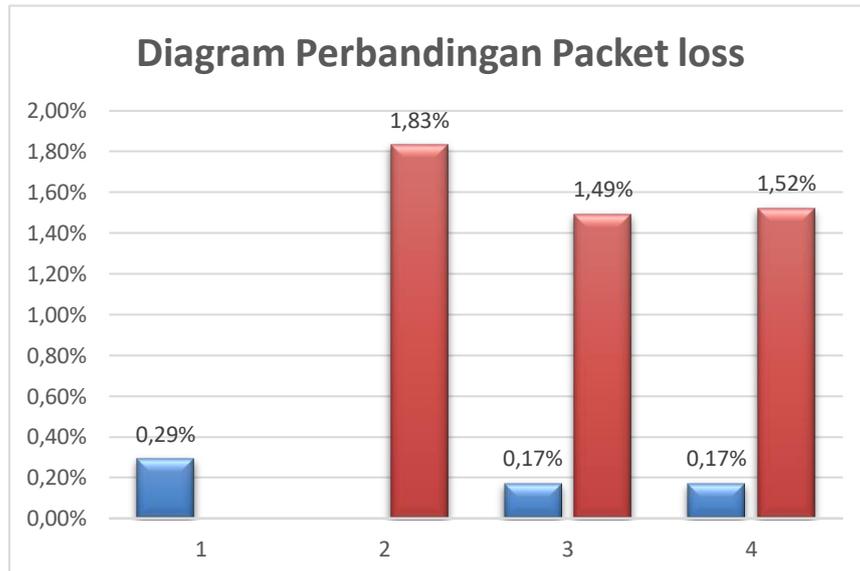
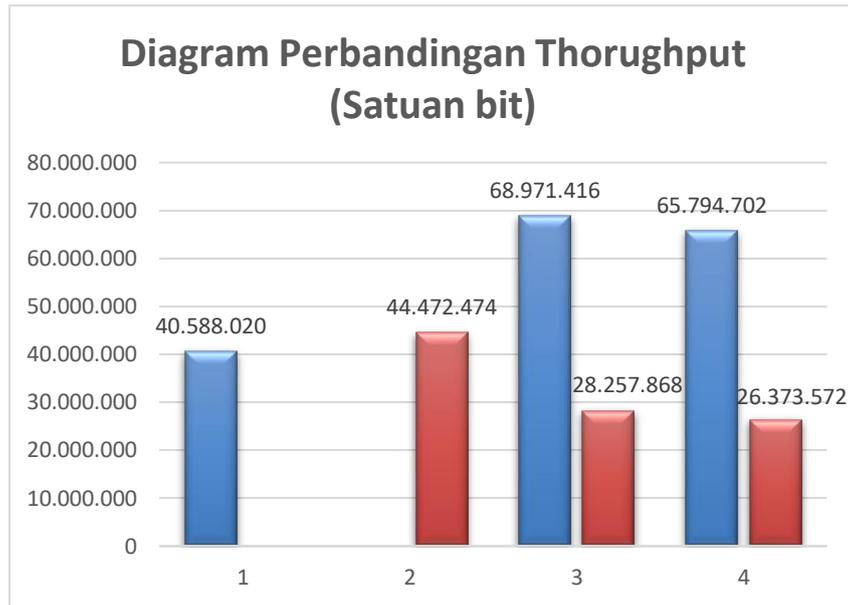
C.) Skenario 3

	VPS USA	VPS INDONESIA
Througput	68.971.416 bit	28.257.868 bit
Packet loss	0,17%	1,49%
End to end Delay	0,330 ms	0,493 ms

D.) Skenario 4

	VPS USA	VPS INDONESIA
Througput	65.794.702 bit	26.373.572 bit
Packet loss	0,17%	1,52%
End to end Delay	0,315 ms	0,526 ms

Berdasarkan hasil penelitian di atas, berikut ini adalah diagram hasil penelitiannya :



4. Kesimpulan

- 1.) *Linux Container (LXC)* bisa digunakan ke dalam *virtual private server* dan dengan berbeda *server negara*.
- 2.) Dalam proses migrasi antar server berbeda, didapatkan hasil *Throughput*, *packet loss*, dan *end to end delay* yang bagus. Sehingga *LXC* sangat cocok digunakan jika *data center* yang digunakan berbeda wilayah negara.

Daftar Pustaka:

- [1] ABRAHAM, S., PETER, B. G., & GREG, G. (2018). *OPERATING SYSTEM CONCEPTS tenth edition*. United States of America.
- [2] Hasanul, F. (2018). ANALYSIS QOS (QUALITY OF SERVICE) MEASUREMENT OF DELAY , JITTER, PACKET LOST AND THROUGHPUT TO GET GOOD QUALITY OF RADIO STREAMING WORK. *TEKNOLOGI INFORMASI DAN KOMUNIKASI*, 100 - 101.
- [3] Sukri, & Jumiaty. (2017). ANALISA BANDWIDTH MENGGUNAKAN METODE ANTRIAN PER CONNECTION QUEUE. *Jurnal Teknologi dan Sistem Informasi Univrab*, 247 - 248 .
- [4] Sumanta, K., Prof. Subhashis, K. C., & Anjali, G. (2018). Operating System a Case Study. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, 166.
- [5] Tutorials Point. (2016). *Operating System Fundamental OS Concepts*.
- [6] WINARNO, S., JAZI, E. I., KHABIB, M., & AHMAD, A. (2015). The Impact of QoS Changes towards Network Performance. *International Journal of Computer Networks and Communications Security*, 1,49.

