

## PENERAPAN TEKNOLOGI BLOCK STORAGE DALAM SISTEM VIRTUALISASI BERBASIS OPENSTACK

Oka Karma Putri, Irawan, Zawiyah Saharuna

Teknik Informatika dan Komputer, Politeknik Negeri Ujung Pandang, Indonesia

---

### Info Artikel

#### Riwayat artikel:

Received, (1 Juni 2024)

Revised, (15 Juni 2024)

Accepted, (22 Juni 2024)

#### Kata kunci:

Block storage,

Openstack,

Read performance,

Virtualisasi.

---

### ABSTRACT

*In agriculture, the drip irrigation system is still using a manual system, so it is necessary to develop an automatic irrigation system that can make it easier for farmers to control water supply. The use of wireless sensors in agriculture is a field that is experiencing an increase. Using a wireless network helps farmers when they have to carry out cable maintenance and repairs in areas that are difficult to reach. This study aims to implement a wireless sensor network based on ESP8266 for drip irrigation system automation that monitors changes in soil moisture and plant temperature, as well as automatic drip irrigation control. The method used in this study is the Waterfall Method, this model develops systematically from one stage to the next. The maximum range distance that can be achieved by the router node to send data to the sink node is 140 meters at the Line Of Sight (LOS) position. At the same time, the router node also controls drip irrigation automatically by sending commands to the actuator node (drip controller) to open or close the solenoid valve connected to the water reservoir which will circulate air by gravity. To implement it, you need ESP8266, YL-69 moisture sensor, DS18B20, Actuator (Relay, Solenoid Valve), Server.*

---

### ABSTRAK

Perkembangan teknologi informasi yang semakin canggih serta meningkatnya kebutuhan perangkat keras komputer untuk membangun berbagai sistem dalam suatu instansi yang bergerak dalam bidang teknologi informasi terkendala dalam pengadaan sistem ketika dilakukan proses *upgrade*. Kendala tersebut adalah biaya yang dibutuhkan tidak sedikit dan dapat diatasi dengan teknologi virtualisasi. Implementasi *block storage* pada virtualisasi berbasis *openstack* dapat mengefisiensi penggunaan sumber daya penyimpanan data pada infrastruktur server. *Block storage* merupakan penyimpanan skala besar yang memiliki performa tinggi dan mempercepat penanganan aliran data. Hasil yang diperoleh dari pengukuran menunjukkan bahwa jumlah rata-rata *read performance* mengalami kecepatan transfer data yang stabil pada *image raw* sebesar 3,6 MB/s serta jumlah rata-rata *write performance* mengalami kecepatan transfer data yang stabil terjadi pada *image raw* sebesar 3,5 MB/. Hasil dari analisis penelitian ini menunjukkan bahwa *block storage* dapat membuat volume/disk yang digunakan oleh *instance*/virtual mesin (VM) sebagai penyimpanan tambahan untuk menyimpan data atau file pada *openstack*.

---

### Penulis Korespondensi:

Irawan

Teknik Informatika dan Komputer, Politeknik Negeri Ujung Pandang, Jl. Perintis Kemerdekaan KM. 10, Makassar, 90245, Indonesia

Email: irawan@poliupg.ac.id

---

## 1. PENDAHULUAN

Virtualisasi merupakan strategi untuk mengurangi konsumsi daya pada pusat data. Dengan virtualisasi, satu *host server* fisik memiliki banyak *server* virtual [1]. Teknologi virtualisasi *server* ini bertujuan untuk menghindari pemborosan daya, proses yang mahal atau dengan kata lain meningkatkan efisiensi serta mengoptimalkan penggunaan prosesor dan penghematan lain seperti biaya listrik karena hanya menggunakan satu atau sedikit *server* saja [2].

Virtualisasi dapat diimplementasikan menggunakan *cloud computing*. *Cloud computing* merupakan sebuah model komputasi (*computing*), dimana sumber daya seperti *processor*, *storage*, *network* dan *software* menjadi abstrak dan diberikan sebagai layanan di jaringan internet menggunakan pola akses remote [3].

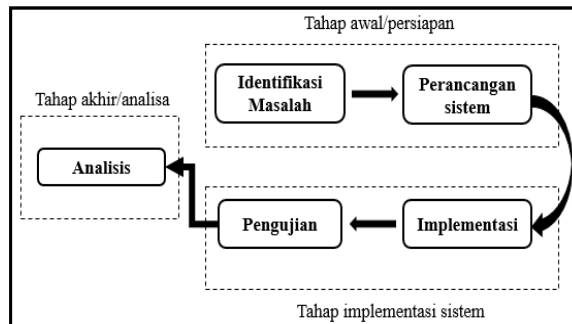
Penelitian sebelumnya yang menggunakan teknologi virtualisasi dilakukan oleh Muhammad Satir [4] dengan judul implementasi infrastruktur virtualisasi dengan openstack. Pada penelitian tersebut dilakukan pengujian terhadap kinerja layanan penggunaan CPU dan memori pada *server* fisik dan *virtual server*. Namun, penelitian tersebut belum menggunakan *block storage* yang merupakan salah satu layanan virtualisasi openstack yang bekerja sebagai *storage*.

*Block storage* merupakan penyimpanan skala besar yang memiliki performa tinggi khususnya untuk memproses database. Sistem blok penyimpanan dapat mengurangi *overhead* dan mempercepat penanganan aliran data, dengan menempatkan data ke dalam blok (*blocking*) dan *deblocking* pada saat ada permintaan data keluar [5].

Oleh karena itu pada penelitian ini dilakukan pembangunan *server storage (block storage)* menggunakan virtualisasi openstack. Sistem ini diharapkan mampu menjadi solusi dalam hal efisiensi penggunaan sumber daya penyimpanan data pada infrastruktur *server*.

## 2. METODE

Prosedur penelitian terdiri dari tiga bagian utama yaitu tahap awal/persiapan, tahap implementasi dan tahap akhir sehingga dalam proses pelaksanaan penelitian lebih terarah dan terstruktur serta hasil yang diperoleh sesuai dengan tujuan penelitian. Adapun proses penelitian yang telah dilakukan ditunjukkan pada Gambar 1.



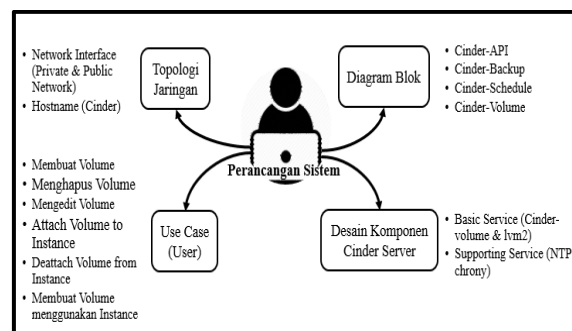
Gambar 1. Prosedur Penelitian

### 2.1 Identifikasi Masalah

Tahap pertama yang dilakukan dalam penelitian ini adalah identifikasi masalah serta cakupan sistem yang akan dibuat. Proses identifikasi masalah dilakukan dengan pemahaman dasar teori dan melakukan studi literatur melalui pustaka-pustaka yang berkaitan dengan penelitian berupa *link* resmi, jurnal, buku dan skripsi yang dijadikan literatur untuk mendukung judul yang diangkat. Pada penelitian sebelumnya belum terdapat *block storage*. Identifikasi masalah dilakukan untuk mengidentifikasi dan memahami konsep dari sistem *block storage*.

### 2.2 Perancangan Sistem

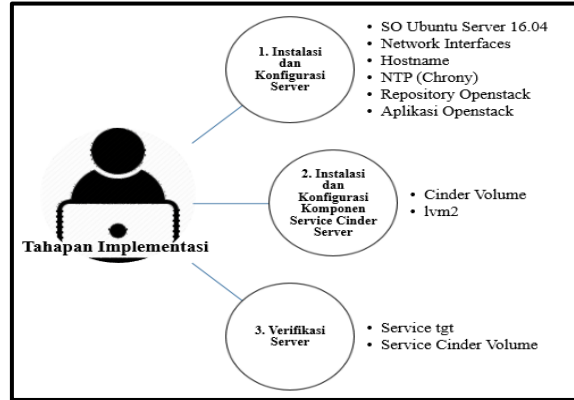
Perancangan sistem merupakan tahapan-tahapan secara konseptual bertujuan sebagai gambaran umum sistem yang dibangun seperti desain topologi jaringan, diagram blok, desain komponen, dan *use case* sistem. Perancangan sistem secara umum mengidentifikasi komponen-komponen sistem yang akan dirancang secara rinci seperti yang ditunjukkan pada Gambar 2.



Gambar 2. Perancangan Sistem

### 2.3. Implementasi

Tahap ini merupakan langkah-langkah dalam mengimplementasikan semua komponen yang telah dirancang sebelumnya. Proses instalasi dan konfigurasi *server* dalam membangun sistem openstack dilakukan secara bertahap karena masing-masing komponen dalam openstack memiliki fungsi yang berbeda namun saling terintegrasi satu sama lain, sehingga jika salah satu komponen dalam proses instalasi tidak berjalan maka akan mempengaruhi beberapa komponen yang lain ketika melakukan implementasi pada sistem. Penejelasan mengenai tahapan instalasi dan konfigurasi sistem ditunjukkan pada Gambar 3.



Gambar 3. Tahapan Instalasi dan Konfigurasi

### 2.4. Pengujian Fungsional Sistem

Pengujian sistem dilakukan untuk memenuhi tujuan yang ingin dicapai dalam penelitian ini, yaitu dengan melakukan pengujian terhadap sistem yang sudah berhasil terimplementasikan. Adapun skema pengujian fungsional sistem dilakukan bertujuan untuk menguji sistem openstack apakah telah sesuai dengan kebutuhan yang diinginkan.

Tabel 1. Daftar modul yang di uji

Pengujian	Bahan Uji
Black Box	a. Membuat volume
	b. Attach volume
	c. Membuat <i>instance</i> (VM) menggunakan volume

Metode yang digunakan adalah *black box testing* yaitu pengujian terhadap kesesuaian antara fungsionalitas sistem openstack yang dirancang berdasarkan kebutuhan pengguna dengan sistem openstack yang diimplementasikan. Digaram aktivitas adalah sebuah skema aktivitas yang terjadi pada pengujian sistem openstack pada *dashboard*.

### 2.5. Analisis Kinerja Sistem

Pada tahap ini dilakukan pengujian terhadap kinerja layanan pada sistem yang dibuat dengan pemasangan aplikasi FTP *server* pada setiap *instance* dan analisa terhadap kinerja layanan dengan melihat performa baca tulis *hardisk* pada virtual mesin (*instance*). Analisa dilakukan pada saat proses *upload download file* pada FTP *server* dalam *instance/VM*.

#### 2.5.1 Pemasangan Aplikasi FTP *server*

FTP *server* pada *instance* yang berhasil terpasang dengan menggunakan volume sebagai *disk* dapat diakses pada *web browser* menggunakan alamat *ip external* yang di *allocate* pada *instance*. *File* yang akan diuji pada FTV *server* diantaranya *file ISO*, video, dan teks.

No	VM/Instance	Image			Flavor	
		Nama	Format	Size	CPU	RAM
1	FTP Server	Ubuntu 16.04	QCOW	287.19 MB	1	1 GB
2	FTP Server	Ubuntu 16.04	RAW	2.20 GB	1	1 GB
3	FTP Server	Ubuntu 16.04	VMDK	274.62 MB	1	1 GB

Table 2. Spesifikasi Instance yang akan dibuat untuk pengambilan data

### 2.5.2 Analisis Penggunaan Harddisk

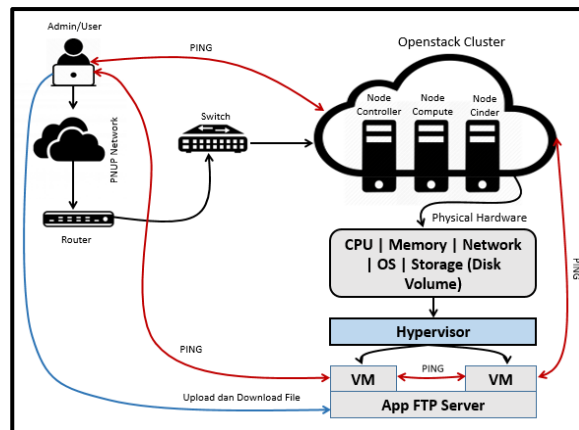
Setelah pemasangan aplikasi FTP server berhasil berjalan, pengujian selanjutnya adalah analisis untuk melihat performa baca tulis penggunaan harddisk pada cinder server saat upload dan download file pada aplikasi FTP server. Proses pengambilan data dilakukan setiap upload dan download file berbeda sebanyak 10 kali dalam setiap pengujian. Pada Tabel 3 menunjukkan file yang akan diupload dan didownload pada saat pengambilan data baca tulis penggunaan harddisk.

Tabel 3. Daftar file yang di uji

No	File	Kapasitas
1	ISO	1.6 GB
2	Video	1.4 GB
3	Teks	2.5 GB

### 2.5.3 Pengujian Analisis Service

Pada pengujian analisis service menggambarkan proses pengujian sistem secara umum. Gambar 4 menunjukkan proses pengujian analisis service.



Gambar 4. Diagram Alur Pengujian Analisis Service

## 3. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah block storage dapat membuat volume/disk. Volume/disk dapat digunakan untuk menyimpan data atau file. Ada beberapa type image yang dapat digunakan untuk membuat volume/disk diantaranya image qcow, raw dan VMDK. Volume/disk tersebut dapat dijadikan penyimpanan tambahan oleh virtual mesin/instance dalam virtualisasi openstack.

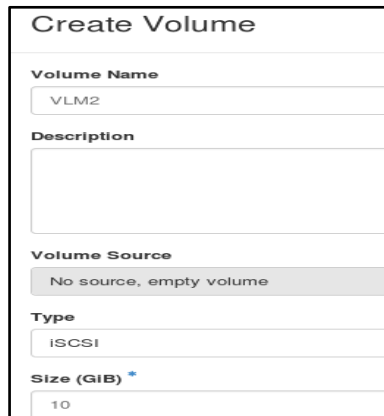
Adapun jenis virtualisasi yang digunakan yaitu full virtualisization. Full virtualisization berarti seolah-olah ada komputer lain didalam komputer yang disebut virtualisasi.

### 3.1 Hasil Pengujian Fungsional

Pengujian sistem dilakukan untuk memenuhi tujuan yang ingin dicapai dalam penelitian ini, dengan melakukan pengujian terhadap sistem yang sudah berhasil diimplementasikan.

#### 3.1.1 Halaman Membuat Volume

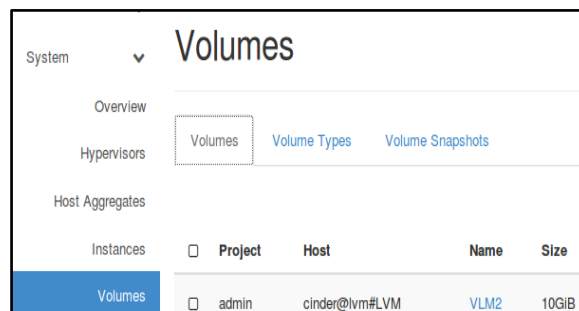
Volume merupakan storage tambahan dan juga dapat digunakan untuk partisi tersendiri di *openstack*. Satu volume tidak boleh digunakan oleh dua atau lebih virtual mesin/instance. Satu volume hanya boleh dipasang atau digunakan oleh satu *instance* yang aktif. Beberapa kolom yang harus dilengkapi untuk membuat volume seperti yang ditunjukkan pada Gambar 5.



The screenshot shows a 'Create Volume' form with the following fields and values:

- Volume Name:** VLM2
- Description:** (empty text area)
- Volume Source:** No source, empty volume
- Type:** ISCSI
- Size (GIB):** 10

Gambar 5. Halaman Membuat Volume



The screenshot shows the 'Volumes' dashboard with a table listing the created volume:

Instances	Project	Host	Name	Size
<input type="checkbox"/>	admin	cinder@lvm#LVM	VLM2	10GiB

Gambar 6. Tampilan Hasil Membuat Volume

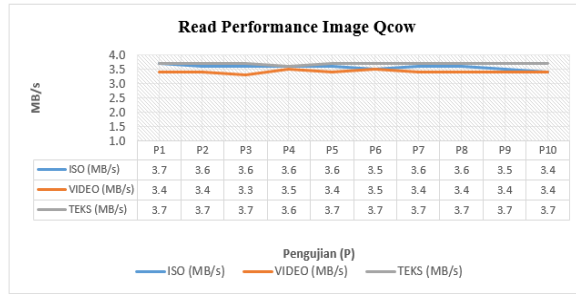
Pada Gambar 6 menunjukkan bahwa volume berhasil dibuat. Volume yang telah dibuat, dapat dipasang atau attach pada *instance* yang belum memiliki OS atau volume tambahan.

### 3.2 Hasil Analisis Penggunaan Hardisk

Analisis ini dilakukan untuk melihat performa baca tulis harddisk (volume/disk) yang dipasang pada setiap *instance* dengan menggunakan type image yang berbeda-beda yaitu image qcow, raw dan VMDK. Untuk setiap *instance* yang diuji masing-masing menggunakan disk berkapasitas 50 GB dan RAM 1 GB dengan beban FTP server.

#### 3.2.1 Analisis Image Qcow

Dari hasil pengujian yang dilakukan sebanyak 10 kali pada masing-masing *file* didalam virtual mesin yang menggunakan *image* tipe qcow, maka hasil yang didapatkan untuk *read performance* dan *write performance* adalah sebagai berikut :



Gambar 7. Grafik hasil pengujian *read performance image qcow*

Pada Gambar 7 merepresentasikan hasil pengujian *read performance* pada *image qcow*. Pada grafik tersebut dapat dilihat kecepatan transfer data tertinggi terjadi pada *file* teks, kecepatan transfer setiap kali pengujian cenderung stabil dan hanya mengalami penurunan kecepatan transfer data pada pengujian ke 4 yaitu 3.6 MB/s. Sedangkan dari hasil pengujian *file* ISO dan video kecepatan transfernya mengalami fluktuasi.

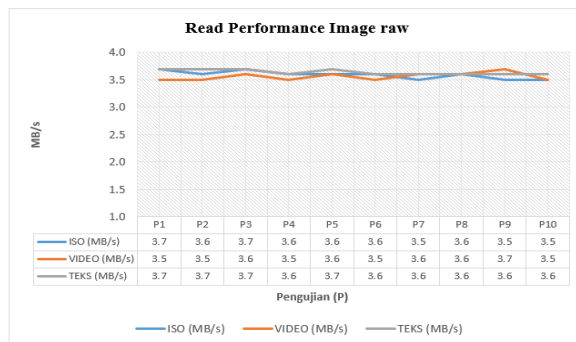


Gambar 8. Grafik hasil pengujian *write performance image qcow*

Dalam penelitian ini dapat dilihat grafik hasil pengujian *write performance image qcow* pada Gambar 8. Dari pengujian 1 sampai 8 kecepatan transfer data pada *file* video cenderung stabil dan mengalami penurunan kecepatan transfer data pada pengujian ke 9 yaitu 3.3 MB/s sedangkan kecepatan transfer data pada *file* ISO dan teks mengalami fluktuasi.

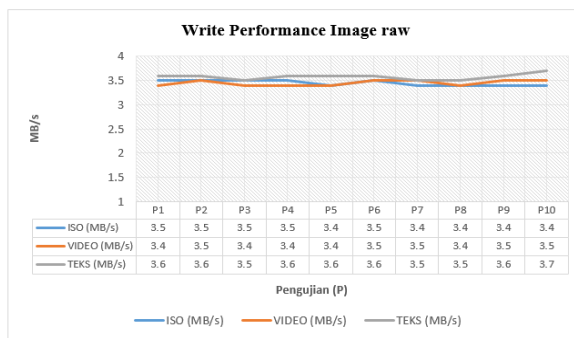
### 3.2.2 Analisis *Image Raw*

Dari hasil pengujian yang dilakukan sebanyak 10 kali pada masing-masing *file* didalam virtual mesin yang sudah terdapat aplikasi *FTP server* dengan menggunakan *image* tipe *raw*, maka hasil yang didapatkan untuk *read performance* dan *write performance* adalah sebagai berikut :



Gambar 9. Grafik hasil pengujian *read performance image raw*

Pada Gambar 9 merepresentasikan grafik hasil pengujian *read performance* pada *image raw*. Pengujian kecepatan transfer data setiap *file* mengalami fluktuasi. Pengujian kecepatan transfer data tertinggi terjadi pada *file* teks dan kecepatan transfer data terendah terjadi pada *file* ISO dan video.

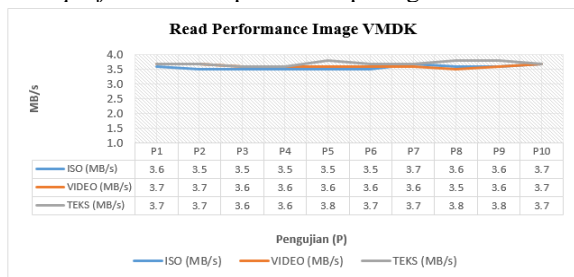


Gambar 10. Grafik hasil pengujian *write performance image raw*

Pada Gambar 10 merepresentasikan grafik hasil pengujian *write performance* pada *image raw*. Pengujian kecepatan transfer data setiap *file* mengalami fluktuasi. Pengujian kecepatan transfer tertinggi terjadi pada file teks dan kecepatan transfer terendah terjadi pada *file* video dan ISO. Dari pengujian 1 sampai pengujian 4 pada *file* ISO cenderung stabil yaitu 3.5 MB/s dan mengalami penurunan pada pengujian 5 yaitu 3.4 MB/s lalu kembali stabil pada pengujian 6 kemudian pada pengujian 7 sampai 10 kembali mengalami penurunan kecepatan transfer.

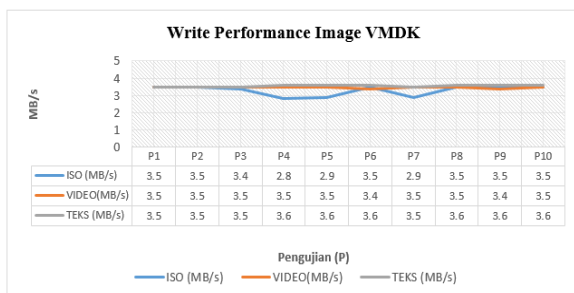
### 3.2.3 Analisis *Image VMDK*

Dari hasil pengujian yang dilakukan sebanyak 10 kali pada masing-masing *file* didalam virtual mesin yang sudah terdapat aplikasi FTP server dengan menggunakan *image* tipe VMDK, maka hasil yang didapatkan untuk *read performance* dan *write performance* dapat dilihat pada grafik berikut :



Gambar 11. Grafik hasil pengujian *read performance image VMDK*

Berdasarkan Gambar 11 grafik hasil pengujian *read performance* pada *image VMDK* kecepatan transfer setiap *file* mengalami fluktuasi. Kecepatan transfer *file* tertinggi terjadi pada *file* teks sedangkan kecepatan transfer *file* terendah terjadi pada file ISO dan video.



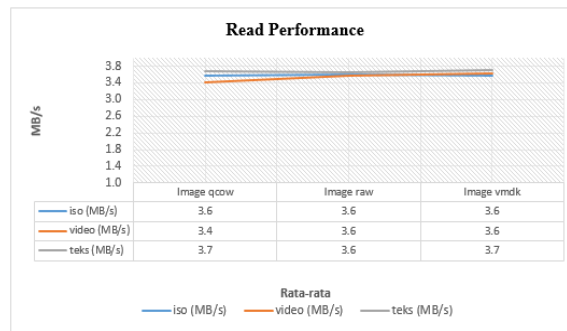
Gambar 12. Grafik hasil pengujian *write performance image VMDK*

Pada Gambar 12 merepresentasikan hasil pengujian *write performance* pada *image VMDK*. Kecepatan transfer data pada *file* video dan teks mengalami fluktuasi. Pada pengujian 5 ke pengujian 6 pada *file* video mengalami penurunan kecepatan transfer data dan kembali stabil pada pengujian 7 sampai pengujian 8. Adapun pengujian 3 ke pengujian 4 pada *file* teks mengalami peningkatan kecepatan transfer data dan kembali mengalami penurunan transfer pada pengujian ke 7 yaitu 3.5 MB/s kemudian kembali stabil pada pengujian 8 yaitu 3.6 MB/s,

sedangkan pada *file* ISO mengalami penurunan kecepatan transfer yang signifikan pada pengujian 4,5 dan 7. Hal ini disebabkan banyaknya situs lain yang diakses secara bersamaan pada saat proses *upload file* sehingga kecepatan transfer data dalam jaringan menurun,

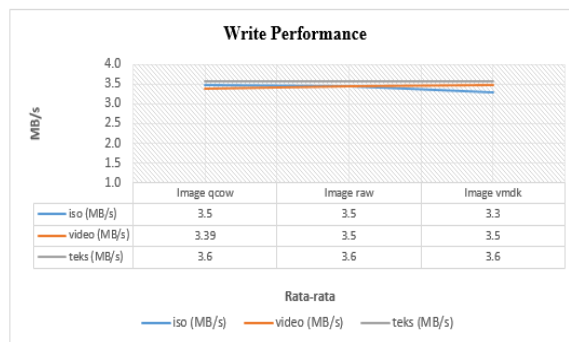
### 3.2.4 Analisis Rata-rata Gabungan *Image* Qcow, Raw dan VMDK

Dari hasil pengujian yang dilakukan sebanyak 10 kali pada setiap virtual mesin yang menggunakan tipe *image* qcow, raw dan VMDK pada masing-masing *file*, maka hasil yang didapatkan untuk rata-rata *read performance* dan *write performance* adalah sebagai berikut :



Gambar 13. Grafik hasil rata-rata *read performance image* qcow, raw, dan VMDK

Dari Gambar 13 menunjukkan grafik hasil rata-rata *read performance* dengan kecepatan baca *type image* qcow menggunakan *file* ISO memiliki kecepatan rata-rata 3.6 MB/s, *file* video 3.4 MB/s, dan *file* teks 3.7 MB/s. Adapun untuk *type image* raw menggunakan *file* ISO memiliki kecepatan rata-rata 3.6 MB/s, *file* video 3.6 MB/s, dan *file* teks 3.6 MB/s, sedangkan untuk *type image* VMDK menggunakan *file* ISO memiliki kecepatan rata-rata 3.6 MB/s, *file* video 3.6 MB/s, dan *file* teks 3.7 MB/s.



Gambar 14. Grafik hasil rata-rata *write performance image* qcow, raw, dan VMDK

Pada Gambar 14 menunjukkan grafik hasil rata-rata *write performance* dengan kecepatan tulis *type image* qcow menggunakan *file* ISO memiliki kecepatan rata-rata 3.5 MB/s, *file* video 3.39 MB/s, dan *file* teks 3.6 MB/s. Adapun untuk *type image* raw menggunakan *file* ISO memiliki kecepatan rata-rata 3.5 MB/s, *file* video 3.5 MB/s, dan *file* teks 3.6 MB/s, sedangkan untuk *type image* VMDK menggunakan *file* ISO memiliki kecepatan rata-rata 3.3 MB/s, *file* video 3.5 MB/s, dan *file* teks 3.6 MB/s.

## 4. KESIMPULAN

Berdasarkan hasil pengujian dapat disimpulkan bahwa :

1. Implementasi *block storage* pada virtualisasi berbasis openstack dapat menggunakan arsitektur jaringan yang sederhana. *Block storage* dapat membuat volume/disk yang digunakan oleh *instance/virtual mesin* (VM) sebagai penyimpanan tambahan untuk menyimpan data atau *file* pada openstack.
2. Pengujian dilakukan pada tiga *instance* dengan *type image* yang berbeda-beda yakni *image* qcow, raw dan VMDK menggunakan aplikasi FTP server. Pengambilan data berlangsung pada saat proses *upload* dan *download file* pada aplikasi FTP server dengan mengukur kecepatan transfer data. Jumlah situs yang diakses dapat mempengaruhi kecepatan transfer data, semakin banyak situs yang diakses pada saat proses *upload*



dan *download* file maka kecepatan transfer data akan semakin menurun dan begitupun sebaliknya.

## 5. REFERENCES

- [1] Hartanto. (2012). Implementasi Virtual Private Server Menggunakan Xen Hypervisor, 1–9.
- [2] Arfriandi. (2012). Perancangan, implementasi, dan Analisis Kinerja Virtualisasi Menggunakan Proxmox Esx, Vmware dan Openstack. *Jurnal Teknologi*, 5 nomor 2, 182–191. <https://doi.org/10.1016/j.matdes.2011.11.021>
- [3] Nugraha, D. (2015). Implementasi Private Cloud Computing Sebagai Layanan Infrastructure As a Service ( IaaS ). *Studi, Program Informatika, Teknik Komputer, Jurusan Ilmu*, 8(2), 7–14.
- [4] Satir. (2018). Implementasi Infrastruktur Virtualisasi dengan Openstack.
- [5] Mobnasesemka. (2016). Mengenal Block Storage Sebagai Konsep Penyimpanan di Data Center. Retrieved from <https://mobnasesemka.com/block-storage-data-center/>
- [6] Kumar. (2014). Cloud computing using open source solution – OpenStack. *International Journal of Applied Engineering Research*, 9(27 Special Issue), 9636–9639. <https://doi.org/10.13140/2.1.1695.9043>
- [7] Openstack.org. (2017). Storage, Layanan Blcok. Retrieved from <https://docs.openstack.org/newton/id/install-guide-ubuntu/cinder.html>