

IMPLEMENTASI DAN ANALISIS PERFORMANSI RAID PADA DATA STORAGE INFRASTRUCTURE AS A SERVICE (IAAS) CLOUD COMPUTING

Delvia Santi¹, R. Rumani M², Yudha Purwanto³

^{1,2,3} Gedung N-203, Departemen Elektro & Komunikasi,
Fakultas Teknik - Universitas Telkom

Jl. Telekomunikasi No. 1, Dayeuhkolot, Bandung 40257

¹ee_dhelkyo@yahoo.com, ²rrm@ittelkom.ac.id, ³yup@ittelkom.ac.id

Abstrak

Dalam rangka untuk memaksimalkan utilisasi *hardware server*, dapat digunakan teknologi *cloud computing*. Teknologi ini, memungkinkan utilisasi yang tidak terpakai dapat digunakan untuk beberapa *virtual server* lainnya. Dengan media penyimpanan *cloud computing* yang sangat besar, kemungkinan terjadinya kesalahan atau kegagalan menjadi tinggi karena banyaknya data yang akan disimpan atau diproses. Akibatnya kinerja performansi *disk* pun menjadi menurun. Salah satu cara untuk meningkatkan kinerja performansi *disk* adalah dengan menggunakan RAID (*Redundant Array of Independent Disk*).

Dengan diterapkannya RAID pada *data storage Infrastructure as a Service (IAAS) Cloud Computing*, dari sisi performansi, RAID 0 dapat meningkatkan kecepatan baca sebesar 11.13%, dan kecepatan tulis sebesar 23.35% dibandingkan dengan tanpa RAID. Sedangkan untuk nilai IOPS (*Input Output Operation Per Second*) terjadi peningkatan sebesar 74.80% pada saat sebelum *resource* digunakan oleh *instance* dan 80.89% pada saat setelah *resource* digunakan oleh *instance*, dibandingkan dengan tanpa RAID.

Kata kunci: *IaaS, Cloud Computing, RAID, performansi, IOPS.*

1. Pendahuluan

1.1 Latar Belakang

Dengan adanya teknologi *cloud computing* kebutuhan investasi dari infrastruktur dapat dipenuhi tanpa diperlukan adanya investasi infrastruktur dan *platform* secara fisik. *Cloud Computing* mampu menyediakan sumber daya teknologi informasi dasar seperti media penyimpanan, *processing power*, *memory*, sistem operasi dan kapasitas jaringan yang dapat digunakan oleh pengguna *cloud client* untuk menjalankan aplikasi yang dimilikinya dengan cara memaksimalkan utilitas *hardware server* yang ada. *Infrastructure as a Service (IaaS)* adalah salah satu layanan dari *Cloud Computing* yang menyediakan akses virtualisasi komputer, sumber daya dari *hardware* termasuk mesin, jaringan maupun penyimpanan data yang dapat digunakan oleh pengguna *cloud client*. [5]

Untuk meningkatkan kinerja, dilibatkan banyak *disk* sebagai satu unit penyimpanan. Tiap-tiap blok data dipecah ke dalam beberapa sub blok, dan dibagi-bagi ke dalam *disk-disk* tersebut. Ketika mengirim data *disk-disk* tersebut bekerja secara paralel dan dengan sinkronisasi pada masing-masing *disk*, sehingga dapat meningkatkan kecepatan transfer dalam membaca atau menulis data. Cara ini dikenal sebagai RAID. Selain masalah kinerja, RAID juga dapat meningkatkan reliabilitas dari *disk* dengan cara melakukan redundansi data. [6]

1.2. Tujuan

Tujuan dari penelitian ini adalah:

1. Membuat *IaaS Cloud Computing* menggunakan *Eucalyptus*
2. Mengimplementasikan RAID pada *server Cloud Computing*
3. Mengetahui performansi *data storage* sistem *Cloud Computing* sebelum dan sesudah digunakannya RAID.
4. Mengetahui performansi *data storage* sistem *Cloud Computing* ketika menggunakan RAID 0 dan RAID 1.

2. Dasar Teori

2.1. Konsep RAID

RAID (*Redundant Array of Independent Disk*) merupakan salah satu cara untuk meningkatkan kinerja dan performansi *disk*. Metodenya dengan membentuk suatu sistem dari beberapa *hard disk* sehingga terbentuk satu partisi dari beberapa *hard disk*. Kegunaan RAID adalah sebagai perlindungan penyimpanan data sehingga kehandalan (*reliability*) data tetap terjaga. RAID merupakan gabungan beberapa *hard disk* fisik ke dalam sebuah unit logika penyimpanan, dengan menggunakan perangkat lunak atau perangkat keras khusus.[4]

2.2. Cloud Computing

Cloud Computing adalah sebuah model komputasi, dimana sumber daya seperti prosesor, *storage*, *network*, dan *software* menjadi abstrak dan diberikan sebagai layanan di jaringan atau *internet* menggunakan pola akses *remote*. Model *billing* dari layanan ini umumnya mirip dengan model layanan publik. Ketersediaan *on-demand* sesuai kebutuhan, mudah untuk dikontrol, dinamik dan skalabilitas yang hampir tanpa batas adalah beberapa atribut penting dari *cloud computing*. Sebuah *setup* infrastruktur model *cloud computing* biasanya di kenali sebagai *Cloud*. [2]

2.3. Eucalyptus

Eucalyptus adalah *software* yang tersedia untuk membuat dan mengatur *private Cloud* maupun *public Cloud*; *Eucalyptus* terdiri atas lima komponen, yaitu *Node Controller* (NC), *Cluster Controller* (CC), *Cloud Controller* (CLC), *Walrus Storage Controller* (WS3) dan *Storage Controller* (SC) [7]

3. Perancangan dan Implementasi

3.1. Kebutuhan Perangkat Keras Minimum

Untuk merancang layanan *Cloud Computing* menggunakan *Eucalyptus* terdapat dua cara. Pertama penginstalasian *CLC*, *Walrus*, *CC*, *SC*, dan *Node* dalam satu mesin disebut juga dengan *cloud in a box*. Yang kedua, setiap komponen *Eucalyptus* di *install* di mesin yang berbeda-beda.[3] Dalam penelitian ini, digunakan cara yang pertama, yaitu menginstall komponen-komponen *Eucalyptus* dalam satu mesin atau *cloud in a box*.

Berikut ini adalah spesifikasi minimum yang dibutuhkan untuk merancang *Eucalyptus cloud in a box* yaitu *server* dengan 200 GB *disk space* dan 4 GB *memory*, satu NIC per mesin, dan satu *common network switch*

Perangkat yang digunakan untuk mengimplementasikan RAID pada *Infrastructure as a Service Cloud computing* terdapat pada tabel berikut.

Tabel 1. Daftar Perangkat

Perangkat Keras	Spesifikasi yang digunakan	Jumlah
Laptop (client)	Processor Core 2 Duo, 1.8GHz, Memory 2GB, 1 NIC	1
Komputer (server)	Xeon, Memory 6GB, 1 NIC	1
Managable Switch	Support 100Mbps transfer rate	1
Harddisk	300GB	2

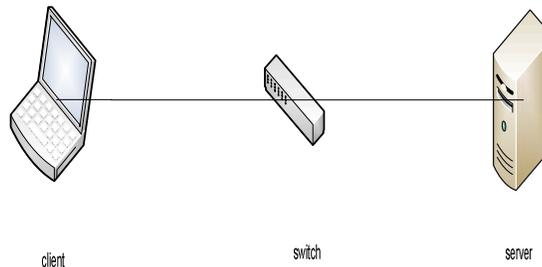
3.2. Kebutuhan Perangkat Lunak Minimum

Perangkat lunak minimum yang dibutuhkan adalah sebagai berikut:

1. Sistem Operasi
 - a. Centos 6.3 sebagai remote server
 - b. Windows 7 sebagai remote server
 - c. *CentOS 6.3* sebagai *virtual machine image*
2. Perangkat lunak pendukung
 - a. *Eucalyptus 3.2* sebagai *software infrastructure as a service cloud computing*
 - b. *Putty* sebagai *software* untuk *remote server*.
 - c. *Dd*, perangkat yang digunakan untuk pengukuran kinerja *block device*, secara umum sudah terdapat pada distribusi *linux*.
 - d. *Bonnie++ benchmarking*, digunakan untuk mengukur kinerja performansi *hard disk*.

3.3. Gambaran umum Sistem

Secara umum, jaringan *Cloud Computing* yang dirancang dan diimplementasikan, dijelaskan melalui gambar 1. berikut. [1]



Gambar 1. Perancangan *cloud computing*

Gambar 3.1. di atas merupakan topologi fisik dari perancangan *Cloud Computing*. Terdapat *client* sebagai pengguna dari *cloud*, *switch* sebagai penghubung *client* dengan *server*, dan *server* sebagai mesin *cloud computing*. Pada *server* digunakan dua *hard disk* untuk pengimplementasian RAID 0 dan RAID 1.

3.4. Instalasi dan Konfigurasi Sistem

Pada penelitian ini, digunakan produk berupa *Fast Start* dari *Eucalyptus System, Inc.* untuk penginstalasian komponen-komponen *Eucalyptus*. Dengan menggunakan *Fast Start*, semua komponen dari *Eucalyptus* sudah “terbungkus” di dalamnya. Hal itu berarti akan memudahkan dalam penginstalasian karena tidak membutuhkan berkali-kali pengunduhan untuk penginstalasian yang membutuhkan waktu yang relatif lama. Pada *Fast Start* ini, sudah terdapat *Operating System* yaitu *Cent OS 6.3* beserta *virtual image* yang akan digunakan.

3.4.1. Konfigurasi RAID 0

Berikut cara untuk konfigurasi RAID 0:

1. Proses pertama sama dengan proses konfigurasi tanpa RAID.
2. Pada bagian buat, pilih partisi *raid*, kemudian atur *size* yang diinginkan. Atur *drive* yang akan digunakan. Pada penelitian ini digunakan *drive* sda dan sdb, dengan ukuran yang sama besar.
3. Buat *device* RAID, pilih *root (/)* untuk *mount point*, dengan tipe *filesystem* ext4, perangkat *raid* md0 dan *level raid* RAID 0. Centang *disk* yang akan digunakan.
4. Selanjutnya penginstalasian *Eucalyptus* dapat dimulai. *Software* di *install* dan *default Eucalyptus Machine Image* (EMI), dibangun. Ketika proses ini selesai, dilakukan *reboot* sistem.
5. Saat proses *reboot*, *accept* lisensi, buat *login non-root* dan aktifkan NTP.
6. Buka terminal, untuk melihat status RAID, ketikkan: # cat /proc/mdstat

3.4.2. Konfigurasi RAID 1

Berikut cara untuk konfigurasi RAID 1:

1. Proses pertama sama dengan proses konfigurasi tanpa RAID.
2. Pada bagian “buat”, pilih partisi RAID, kemudian atur *size* yang diinginkan dan diusahakan harus sama besar, karena jika tidak sama besar maka ukuran yang akan dipakai adalah ukuran *disk* yang lebih kecil. Atur *drive* yang akan digunakan. Pada penelitian ini digunakan *drive* sda dan sdb, dengan ukuran yang sama besar.
3. Buat *device* RAID, pilih *root (/)* untuk *mount point*, dengan tipe *filesystem* ext4, perangkat *raid* md0 dan *level raid* RAID 0. Centang *disk* yang akan digunakan.
4. Selanjutnya penginstalasian *Eucalyptus* dapat dimulai. *Software* di *install* dan *default Eucalyptus Machine Image* (EMI) dibangun. Ketika proses ini selesai, dilakukan *reboot* sistem.
5. Saat proses *reboot*, *accept* lisensi, buat *login non-root* dan aktifkan NTP.
6. Buka terminal, untuk melihat status RAID, ketikkan: #cat /proc/mdstat
7. Tunggu hingga *resync* selesai.

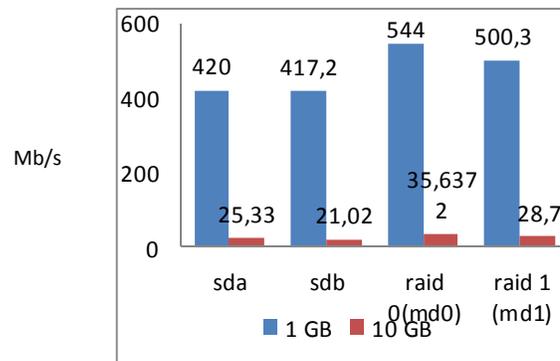
4. Analisis dan Hasil Implementasi

4.1. Analisis Hasil Pengukuran Kemampuan Baca Tulis *Disk* Pada *Drive*

Pada tahap pengujian ini, diperlihatkan pengukuran dari kinerja baca-tulis *disk* dengan menggunakan perangkat lunak *dd*. Pengukuran ini digunakan untuk mengetahui perbandingan dari kinerja *disk* berdasarkan *drive-drive* yang terdapat pada sistem *Cloud Computing*. Pada pengukuran ini, data diambil sebanyak sepuluh kali, kemudian data tersebut diambil rata-ratanya untuk mendapatkan hasil yang *valid*.

4.1.1. Analisis Kecepatan *Disk* Pada Saat Menulis Data

Perbandingan nilai dari kecepatan *disk* pada saat menulis data 1 GB dan 10 GB menggunakan RAID 0, RAID 1 dan tanpa RAID, dapat dilihat pada grafik dibawah ini.



Gambar 2. Grafik kecepatan *disk* untuk menulis data

Dari grafik di atas dapat dilihat kecepatan penulisan data 1 GB pada RAID 0 (md0) sebesar 544 MB/s, sedangkan pada *drive* sda sebesar 420 MB/s, untuk *drive* sdb sebesar 417.2 MB/s, dan *drive* RAID 1(md1) sebesar 500.3 MB/s. Jika dibandingkan dengan RAID 0, *drive* sda mengalami penurunan kecepatan penulisan sebesar 22.79%, sdb mengalami penurunan sebesar 23.35%, dan RAID 1 (md1) sebesar 7.54%.

Kecepatan penulisan 10 GB pada RAID (md0) sebesar 35.6372 MB/s, sedangkan pada sda sebesar 25.33 MB/s, untuk sdb sebesar 21.02 MB/s dan pada *drive* RAID 1 (md1) sebesar 28.7 MB/s. Jika dibandingkan dengan RAID 0, *drive* sda mengalami penurunan kecepatan sebesar 28.92%, *drive* sdb mengalami penurunan sebesar 41 %, dan untuk RAID 1 (md1) sebesar 19.47%.

Pada RAID 0, penulisan data 10 GB mengalami penurunan sebesar 92.45% dibandingkan dengan penulisan 1 GB. Pada sda, penulisan data 10 GB mengalami penurunan sebesar 93.96% dibandingkan dengan penulisan 1 GB. Pada sdb, penulisan data 10 GB mengalami penurunan sebesar 94.91% dibandingkan dengan penulisan 1 GB. Pada RAID1 (md1), penulisan data 10 GB mengalami penurunan sebesar 95.86% dibandingkan dengan penulisan 1 GB.

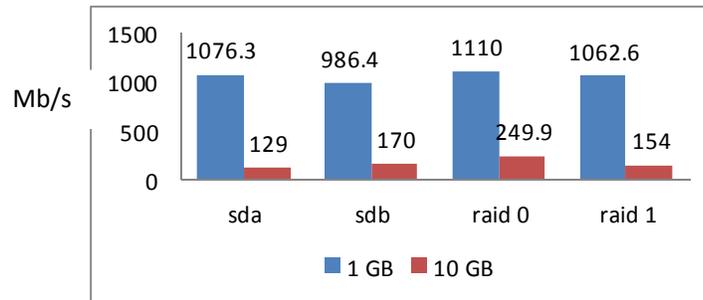
Dilihat dari penulisan data sebesar 1 GB dan 10 GB, kecepatan penulisan data semakin menurun, hal itu karena semakin besar data yang ditulis, maka kecepatan untuk menulis pun menjadi berkurang.

Penulisan data 1 GB dan 10 GB pada sda dan sdb memiliki nilai yang hampir sama, karena pada *drive* tersebut, tidak menggunakan RAID. Artinya, penulisan data pada *drive* sda dan sdb hanya dilakukan pada satu *disk* dalam satu waktu, dengan akibat performa *disk* menjadi tidak maksimal.

RAID 0 memiliki kecepatan paling tinggi dibandingkan *drive* lainnya, baik untuk penulisan data 1 GB maupun 10 GB. Hal itu karena pada RAID 0 menggunakan *data stripping* untuk akses ke blok-bloknya. Dengan *data stripping* penulisan data dapat dilakukan secara paralel, artinya dalam satu waktu data dapat ditulis sebanyak n-kali *disk*; *disk* yang digunakan pada penelitian ini adalah sebanyak dua *disk*.

4.1.2. Analisis Kecepatan Disk Pada Saat Membaca Data

Perbandingan nilai dari kecepatan *disk* pada saat membaca data 1 GB dan 10 GB menggunakan RAID 0 , RAID 1 dan tanpa RAID, dapat dilihat pada grafik dibawah ini.



Gambar 3. Grafik kecepatan *disk* untuk membaca data

Dari grafik di atas dapat dilihat kecepatan pembacaan data 1 GB jika dibandingkan dengan RAID 0, *drive* sda mengalami penurunan kecepatan pembacaan sebesar 3.04%, sdb mengalami penurunan sebesar 11.13%, dan RAID 1 (md1) sebesar 4.27%.

Kecepatan pembacaan 10 GB jika dibandingkan dengan RAID 0, *drive* sda mengalami penurunan kecepatan sebesar 48.19%, *drive* sdb mengalami penurunan sebesar 31.73 %, dan untuk RAID 1 (md1) sebesar 38.15%.

Pada RAID 0, pembacaan data 10 GB mengalami penurunan sebesar 77.49% dibandingkan dengan pembacaan 1 GB. Pada sda, pembacaan data 10 GB mengalami penurunan sebesar 88.02% dibandingkan dengan pembacaan 1 GB. Pada sdb, pembacaan data 10 GB mengalami penurunan sebesar 82.77% dibandingkan dengan pembacaan 1 GB. Pada RAID1 (md1), pembacaan data 10 GB mengalami penurunan sebesar 85.50% dibandingkan dengan pembacaan 1 GB.

Berdasarkan persentase yang di dapat pada pembacaan data 1 GB, kecepatan antara *drive* RAID 0 (md0) dengan *drive* lainnya, tidak mengalami penurunan yang signifikan. Hal itu terjadi karena data 1 GB masih terlalu kecil jika dibandingkan dengan RAM yang ada, akibatnya data masih dapat dibaca pada *cache* di dalam *kernel*.

Berdasarkan persentase yang di dapat pada pembacaan data 10 GB, kecepatan antara *drive* RAID 0 (md0), tanpa RAID dan RAID 1 mengalami penurunan yang signifikan. Hal itu terjadi karena data 10 GB lebih besar daripada kapasitas memori yang tersedia, sehingga data tersebut dibaca melalui *disk*, bukan melalui *cache*. RAID 0 memiliki kecepatan membaca lebih baik daripada yang lainnya, karena RAID 0 menggunakan *data stripping* untuk akses ke blok-bloknya. Dengan *data stripping* pada RAID 0, waktu *seek* pun menjadi lebih cepat karena menggunakan dua *head* untuk membaca data, dengan akibat meningkatnya kecepatan membaca data.

4.2. Analisis Pengukuran IOPS Saat Sebelum dan Sesudah Semua *Resource* Digunakan

IOPS (*Input Output Operation Per Second*) merupakan pengukuran kinerja yang umum digunakan untuk *harddisk drive*, SSD (*Solid State Drive*), dan SAN (*Storage Area Networks*).

Bonnie++ Benchmark merupakan *tools* untuk mengukur nilai IOPS saat sebelum dan sesudah semua *resource* digunakan oleh *instance*. *Resource* ini berupa jumlah CPU, besar RAM dan kapasitas *disk*.

Nilai IOPS menggunakan *bonnie++* didapatkan dengan membuat data berukuran dua kali lebih besar dari RAM yang digunakan. Karena RAM yang digunakan pada penelitian ini sebesar 6 GB, sehingga ukuran data yang dibuat adalah sebesar 12 GB. Tujuan ukuran dibuat dua kali lebih besar daripada RAM adalah untuk menghentikan kerja *cache* sehingga data diambil langsung dari memori utama, agar tingkat keakuratan data menjadi lebih tinggi.

Untuk mengetahui ketersediaan dari *resource* yang ada pada *Cloud Computing* dapat dilakukan dengan perintah: `#euca-describe-availability-zones verbose`.

4.2.1. Pengukuran IOPS Menggunakan Bonnie++

Berikut cara untuk melakukan pengukuran nilai IOPS dengan menggunakan *Bonnie++*:

```
#bonnie++ -u root -d /opt/bonnie_test -s 12G.
```

Dengan melakukan perintah *Bonnie++* diatas, dilakukan *test* terhadap *device*

`/opt/bonnie_tes` dengan ukuran data sebesar 12 GB.

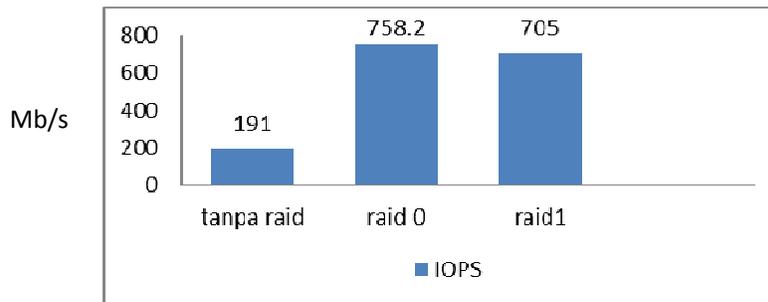
Dibawah ini adalah tabel dari hasil pengukuran IOPS menggunakan *Bonnie++*.

Tabel 2. Hasil Pengukuran IOPS Menggunakan *Bonnie++*

	Tanpa RAID	RAID 0	RAID 1
Sebelum	191 IOPS	758.2 IOPS	705 IOPS
Sesudah	120.8 IOPS	628 IOPS	441.5 IOPS

4.2.2. Analisis Pengukuran IOPS Pada Saat Sebelum Penggunaan Resource Oleh Instance

Pengukuran IOPS ini diambil ketika *resource* yang ada belum digunakan oleh *instance*. Berikut grafik yang menunjukkan hasil pengukuran IOPS pada saat sebelum *resource* digunakan oleh *instance* dengan ukuran *file* yang dibuat sebesar 12 GB.



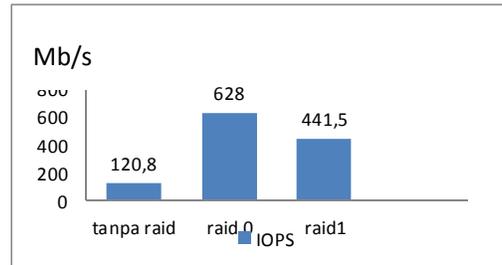
Gambar 4. Grafik IOPS sebelum penggunaan Resource oleh Instance

Jika dibandingkan dengan RAID 0, nilai IOPS tanpa RAID mengalami penurunan sebesar 74.80%. Pada RAID 1, nilai IOPS mengalami penurunan sebesar 6.99% jika dibandingkan dengan RAID 0.

Nilai IOPS dari RAID 0 merupakan yang paling tinggi dibandingkan dengan yang lainnya, karena RAID 0 menggunakan *data stripping* untuk akses ke blok-bloknya. Dengan menggunakan *data stripping*, data dapat ditulis dan dibaca melalui dua *disk* dalam waktu yang bersamaan (paralel). Selain itu, dengan menggunakan RAID 0 kapasitas *disk* pun menjadi dua kali lipat, sehingga dapat meminimalisasi terjadinya kepenuhan (*overload*) data yang dapat mengakibatkan penurunan performansi *disk*.

4.2.3. Analisis Pengukuran IOPS Pada Saat Sesudah Penggunaan Resource Oleh Instance

Pengukuran IOPS ini diambil ketika *resource* yang ada sudah digunakan oleh *instance* secara maksimal. Berikut grafik yang menunjukkan hasil pengukuran IOPS pada saat sesudah semua *resource* digunakan oleh *instance* dengan ukuran *file* yang dibuat sebesar 12 GB.



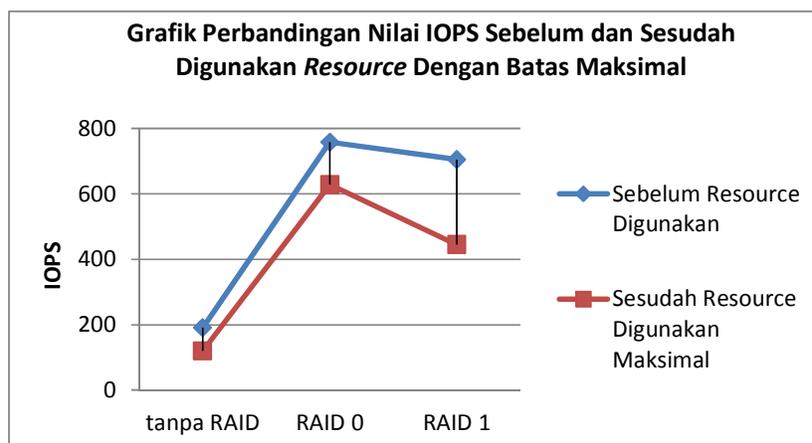
Gambar 5. Grafik IOPS setelah penggunaan *Resource* oleh *Instance*

Jika dibandingkan dengan RAID 0, nilai IOPS tanpa RAID mengalami penurunan sebesar 80.89%. Pada RAID 1, nilai IOPS mengalami penurunan sebesar 29.1% jika dibandingkan dengan RAID 0.

Nilai IOPS RAID 0 paling tinggi dibandingkan yang lainnya karena RAID 0 menggunakan *data stripping*. Nilai IOPS tanpa RAID paling rendah karena hanya menulis satu *disk* dalam satu waktu. Nilai IOPS RAID 1 lebih rendah dibandingkan dengan RAID 0, akan tetapi lebih tinggi jika dibandingkan tanpa RAID, karena RAID 1 menggunakan *mirroring*.

4.2.4. Analisis Perbandingan Nilai IOPS Sebelum dan Sesudah Digunakan Resource Dengan Batas Maksimal

Berikut ini grafik yang menunjukkan perbandingan nilai IOPS sebelum dan sesudah digunakan *resource* oleh *instance* dengan batas maksimal.



Gambar 6. Grafik perbandingan nilai IOPS sebelum dan sesudah digunakan *Resource* dengan batas maksimal

Dilihat dari grafik di atas, perbandingan nilai IOPS pada saat sebelum *resource* digunakan dengan yang telah digunakan hingga batas maksimal mengalami penurunan.

Hal itu karena pada saat sebelum *resource* (CPU, RAM dan *Disk*) digunakan *instance*, *resource* masih sepenuhnya digunakan oleh sistem lokal. Akibatnya pembacaan dan penulisan data menjadi lebih tinggi. Akan tetapi pada saat semua *resource* digunakan secara maksimal, dimana CPU, RAM dan *Disk* pada sistem lokal menjadi berkurang karena *resource* yang ada digunakan oleh *instance* (mesin *virtual*), nilai IOPS pun menjadi menurun.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil penelitian dan analisa yang telah dilakukan adalah sebagai berikut :

1. Dengan menggunakan RAID 0 kemampuan baca-tulis pada *drive* sistem *Cloud Computing* menjadi meningkat. RAID 0 lebih unggul karena menggunakan *data stripping* untuk akses ke blok-bloknya.
2. Nilai pengukuran IOPS menggunakan *Bonnie++* menunjukkan bahwa RAID 0 pada saat sebelum dan sesudah digunakannya *resource* secara maksimal oleh *instance* masih lebih baik dibandingkan dengan tanpa RAID dan RAID 1.
3. Penggunaan RAID pada *storage Cloud Computing* dapat meningkatkan performa *disk* dalam hal ini dengan menggunakan RAID 0.

5.2 Saran

Saran yang dapat diberikan untuk penelitian lebih lanjut mengenai topik ini adalah:

1. Diharapkan untuk penelitian selanjutnya dapat meneliti bagaimana performansi dari level RAID lainnya, seperti RAID 5, 6, 1+0 dan 0+1.
2. Untuk penelitian selanjutnya dapat menggunakan *software Infrastructure as a Service (IaaS)* lainnya seperti *Nimbus* dan *Cloud stack*.
3. Diharapkan untuk penelitian selanjutnya, selain menguji performansi *data storage* juga dapat menguji tentang keamanan data dari *Cloud Computing*.

Referensi

- [1.] Bento, Alberto M, Anil K. Aggarwal., 2013, *Cloud Computing Service and Deployment Model*, IGI Global.
- [2.] Furht, Borko, 2010, *Handbook of Cloud Computing*, Springer.
- [3.] Hurwitz, Judith, Robin Bloor, and Marcia Kaufman, 2010, *Cloud Computing for Dummies*, Wiley Publishing.
- [4.] Johnson, D., 2010, *Eucalyptus Installation Guide*, Vol. 1, UEC Publishing.
- [5.] Josyula, Venkata, Malcolm Orr, and Greg Page, 2012, *Cloud Computing : Automating the Virtualized Data Center*, Cisco Press.
- [6.] Suryatama, Indra, 2012, *Membangun Infrastruktur Komputasi Awan Privat Menggunakan Ubuntu Enterprise Cloud*, Penerbit Andi.
- [7.] Velte, Anthony T, 2010, *Cloud Computing – Apractical Approach*, McGraw Hill.