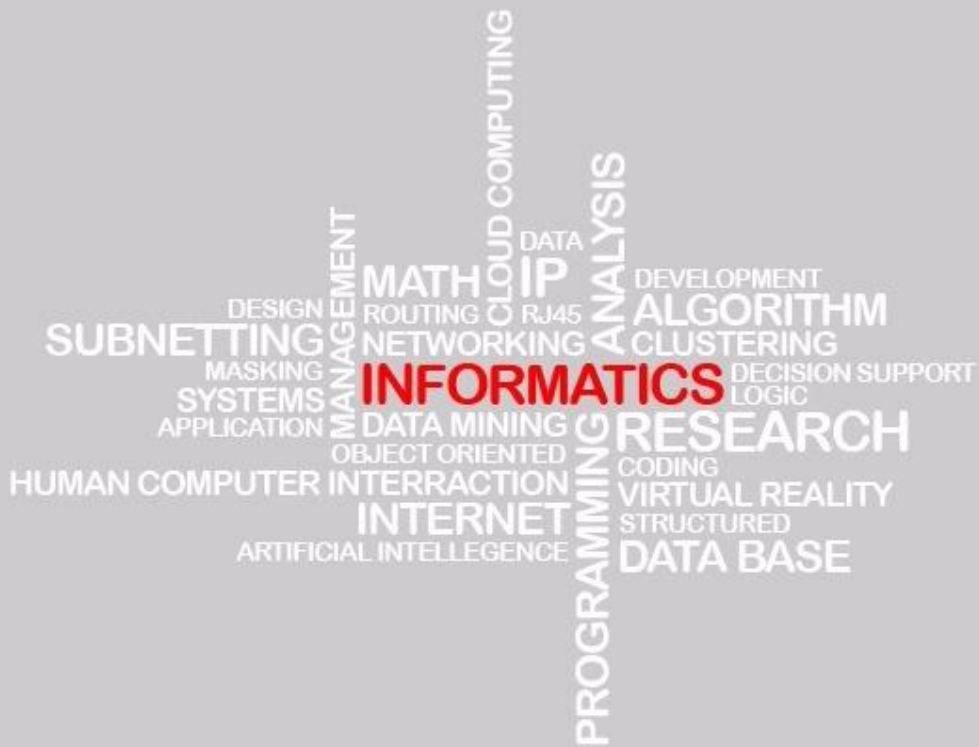


ISSN: 2503-250X

INFORMATICS JOURNAL

Volume 1, Number 1, April 2016



INFORMATION SYSTEM DEPARTMENT
UNIVERSITY OF JEMBER

2016

ISSN : 2503-250X



INFORMAL : Informatics Journal

**Program Studi Sistem Informasi
Universitas Jember**

Analisis Sistem Virtual Cluster Pada Komputasi Pararel
Menggunakan Layanan IAAS

Diah Ayu Retnani, Erick Irawadi Alwi

Analisa SWOT untuk Strategi Pengembangan Teknologi
Informasi pada Koperasi Simpan Pinjam XYZ

Oktalia Juwita, Arief Dwi Hartadi

Analisa dan Perancangan Sistem Informasi Inventori
Produk Non-BBM PT. Pertamina (Persero)

Glady Caren Rorimpandey, Olivia Kembuan

Penerapan Metode AHP dalam Sistem Informasi Pemetaan
Sebaran Strata Desa Siaga Aktif Dinas Kesehatan
Kabupaten Jember

Yanuar Nurdiansyah, Rizka Rohmatul Fitriah

Scenarios for Securing Content Delivery in the DRM
Environment

Antonius Cahya Prihandoko, Hossein Ghodosi, Bruce Litow

Penyisipan Citra Pesan Ke Dalam Citra Berwarna
Menggunakan Metode Least Significant Bit dan Redundant
Pattern Encoding

Rahmandhita Fikri Sannawira, Agus Sidiq Purnomo

Web:
<http://jurnal.unej.ac.id/index.php/INFORMAL>

Vol. 1 No. 1, April 2016



INFORMAL : Informatics Journal

Volume 1 Nomor 1, April 2016

Editorial Team

Editor in Chief

Dr. Saiful Bukhori, ST., M.Kom
University of Jember

Managing Editor

Drs. Antonius Cahya Prihandoko, M.App.Sc., Ph.D
University of Jember

Editorial Board

Achmad Maududie, ST., M.Sc - *University of Jember*
Dr. Daniel Siahaan - *Institut Teknologi Sepuluh November (ITS), Surabaya*
Dr. Adi Wijaya - *Telkom University, Indonesia*
Prof. Drs. Slamir, M.Comp.Sc, Ph.D - *University of Jember*

Layout Editor

Oktalia Juwita, S.Kom, M.MT - *University of Jember*
Fahrobby Adnan, S.Kom, M.MSI - *University of Jember*

Journal Contact

Program Studi Sistem Informasi Universitas Jember,
Jl. Kalimantan 37 Jember 68121, Indonesia.
Phone/Fax : +62.331.326911
Email: informal@unej.ac.id



INFORMAL : Informatics Journal

Volume 1 Nomor 1, April 2016

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Kuasa atas curahan kasih dan rahmatNya, sehingga edisi perdana INFORMAL (Informatics Journal), yakni volume 1 nomor 1 tahun 2016, dapat terbit. Kehadiran jurnal ini diharapkan dapat menambah referensi dalam kegiatan ilmiah. Informatics Journal merupakan salah satu media komunikasi bagi para peneliti di bidang Ilmu Komputer, khususnya pada disiplin Sistem Informasi dan Teknologi Informasi.

Tim Informatics Journal mengucapkan banyak terima kasih atas partisipasi para penulis pada edisi ini. Semoga kerja sama ini akan terus berlangsung pada kesempatan selanjutnya. Kami juga mengundang partisipasi para peneliti lain untuk dapat memberikan kontribusi pada terbitan INFORMAL berikutnya. INFORMAL dijadwalkan terbit setiap catur wulan, yakni pada bulan April, Agustus dan Desember.

Pada terbitan perdana ini masih banyak kekurangan di sana sini. Oleh karena itu, kami mohon saran dan kritik yang membangun untuk lebih baiknya jurnal ini. Akhirnya, semoga INFORMAL dapat memberikan manfaat bagi para penulis dan pembaca sekalian.

Salam,

INFORMAL Editorial Team



Table of Contents

Analisis Sistem Virtual Cluster Pada Komputasi Pararel Menggunakan Layanan IAAS <i>Diah Ayu Retnani, Erick Irawadi Alwi</i>	1 – 9
Analisa SWOT untuk Strategi Pengembangan Teknologi Informasi pada Koperasi Simpan Pinjam XYZ <i>Oktalia Juwita, Arief Dwi Hartadi</i>	10 – 13
Analisa dan Perancangan Sistem Informasi Inventori Produk Non-BBM PT. Pertamina (Persero) <i>Glady Caren Rorimpandey, Olivia Kembuan</i>	14 – 24
Penerapan Metode AHP dalam Sistem Informasi Pemetaan Sebaran Strata Desa Siaga Aktif Dinas Kesehatan Kabupaten Jember <i>Yanuar Nurdiansyah, Rizka Rohmatul Fitriah</i>	25 – 31
Scenarios for Securing Content Delivery in the DRM Environment <i>Antonius Cahya Prihandoko, Hossein Ghodosi, Bruce Litow</i>	32 – 38
Penyisipan Citra Pesan Ke Dalam Citra Berwarna Menggunakan Metode Least Significant Bit dan Redundant Pattern Encoding <i>Rahmandhita Fikri Sannawira, Agus Sidiq Purnomo</i>	39 – 46

Analisis Sistem *Virtual Cluster* Pada Komputasi Paralel Menggunakan Layanan IAAS

Diah Ayu Retnani W , Erick Irawadi Alwi

Program studi sistem informasi Universitas Jember, Jln. Kalimantan 37 Kampus Bumi Tegalboto
Jember, Jawa Timur 68121 Indonesia

Program studi Teknik Informatika Universitas Muslim Indonesia, Jln. Urip Sumohardjo KM.05
Makassar 90231 Indonesia

diah.retnaniw@gmail.com, erickirawadimti15@gmail.com

Abstract. Industrial technology development needed solutions about increasing computing resource performance. The one of alternative solution could resolve this problem using the high performance computing's concept example cluster. Cluster could increase performance of computing using many integrated computers in the computer network. cluster had problem in infrastructure system example place and hardware. One of the alternative solutions was virtual computer for infrastructure. one of cloud system services was IAAS. IAAS could produce integrated virtual computation resource on network directly. This study discussed about virtual cluster's performance using the virtual computer to solve the parallel computing's problem. Virtual computer in this study used cloud services. The research method was by implementing OpenMPI on a Linux-based computer cluster using five virtual computer and analyzed the performance of the system by different scenario tests. The results of this study Showed that the cloud was able to facilitate user to provide integrated infrastructure's computing service on a network-based and the virtual cluster machine was suitable when large size of data. Ideally, the value of speedup increased when the number of processors increases. The value of Speed-up was ideal when the the data size of 1000 using two nodes (1,346) and three nodes (1.0465). The value of Efficiency decreases when increasing number of processors. Most of the Value of speed-up and efficiency were not ideal at this experiment Because there was the overhead in the parallel system, for example, the additional parallel computing, inter-processor communication and synchronization processes in the cloud as provider of virtual computers.

Keywords: Cluster, openMPI, speed-up, Efficiency, mergesort, cloud.

1 Introduction

Perkembangan teknologi di perindustrian khususnya dibidang teknologi komputer menuntut solusi terhadap kebutuhan mulai dari sumber daya komputasi, media penyimpanan dan kecepatan komunikasi. Konsep *high-performance computing* merupakan alternatif solusi dalam menyelesaikan masalah komputasi. Pendekatan *parallel computing* telah memberikan peningkatan kecepatan waktu eksekusi penyelesaian masalah. Komputasi paralel (*parallel computing*) merupakan teknik menjalankan program untuk suatu proses dengan menggunakan lebih dari satu unit komputasi [4], komputasi paralel dapat mempersingkat dengan membagi program menjadi task-task yang dapat dikerjakan secara terpisah untuk kemudian dieksekusi secara paralel [3]. implementasi aneka jenis komputasi paralel ini memerlukan infrastruktur mesin paralel yang terdiri dari banyak komputer dengan dihubungkan pada satu jaringan dan mampu bekerja secara bersama sama untuk menyelesaikan satu masalah seperti penyelesaian masalah komputasi, rendering dan sebagainya. Penggunaan komputasi paralel salah satunya adalah *cluster*. *Cluster* merupakan pilihan yang cukup handal untuk pengolahan data dalam jumlah besar dan banyak. *Cluster* termasuk dalam klasifikasi *Distributed Memory Multikomputer*, merupakan dua atau lebih komputer atau *node* yang dihubungkan menjadi suatu sistem terintegrasi yang mampu menyelesaikan masalah komputasi bersama sama.

Penelitian yang dilakukan dalam bidang komputasi paralel terutama cluster sudah cukup banyak. Penelitian yang telah dilakukan antara lain terkait penggunaan praktisnya, membahas *hardware* pemrosesan paralel, pengukuran kinerja, perbandingan pemrosesan paralel dengan sistem serial atau *sequential* maupun penelitian tentang komputasi paralel dengan mengimplementasikan MPI sebagai middlewarenya. Penelitian yang dilakukan oleh yudistira tentang analisis perbandingan *Multi-Core dengan PC-Clustering Terhadap Kinerja Pengolahan Paralel* menyatakan bahwa sistem *PC-cluster* sangat unggul pada *bandwidth* dan kapasitas memori, dari keunggulan-keunggulan yang dimiliki tersebut sehingga membuat sistem *PC-cluster* menjadi sistem terbaik untuk komputasi paralel dibandingkan dengan *multi-core* [5].

Sistem cluster membutuhkan lebih dari satu komputer yang tergabung dalam satu jaringan, sehingga hal ini menimbulkan permasalahan kembali. Permasalahan permasalahan yang terjadi antara lain permasalahan pada kebutuhan komputer yang banyak, kebutuhan ruang untuk penyimpanan piranti piranti pendukung sistem sehingga meningkatkan biaya biaya untuk membangun infrastrukturnya.

Sistem cluster yang membutuhkan banyak komputer ini dapat teratasi dengan adanya teknologi virtualisasi pada pembangunan infrastruktur sistem seperti penggunaan komputer virtual. Teknologi virtualisasi adalah sebuah teknik atau cara untuk membuat sesuatu dalam bentuk *virtual*. Perangkat lunak *virtualisasi* adalah sebuah program yang memiliki kemampuan untuk membuat sebuah komputer secara *virtual* [2], disebut komputer *virtual* karena komputer itu tidak ada secara fisik. Salah satu kelebihan dari penggunaan teknologi *virtual* adalah pengurangan biaya investasi hardware dan tempat, sehingga pada penelitian ini membahas analisa kinerja dari sistem *cluster* yang dibangun dengan menggunakan komputer *virtual*. Komputer virtual pada penelitian ini menggunakan layanan dari sistem cloud yang telah dibangun oleh peneliti sebelumnya. Sistem cloud merupakan sistem yang mampu menyediakan sumber daya teknologi informasi berbasis internet sehingga dapat memudahkan pengguna tanpa perlu menyediakan infrastruktur dalam membangun sistem. Penggunaan cloud sebagai penyedia piranti sistem cluster dipilih karena pada penelitian sebelumnya yang dilakukan oleh faisal bahwa *cloud computing* mampu menyediakan sumber daya - sumber daya (prosesor, memori, dan *storage*) oleh para penggunanya, dari hasil penelitian seiring dengan bertambahnya jumlah mesin virtual yang aktif tidak terjadi degradasi kinerja yang artinya skalabilitas server dalam virtualisasi mempunyai performance yang bagus mendekati skalabilitas server sebenarnya [1].

2 Research Method

Penelitian ini menggunakan metode eksperimen yaitu menganalisis kinerja dari sistem virtual cluster dengan melakukan serangkaian uji coba menyelesaikan masalah komputasi. Permasalahan yang digunakan sebagai bahan uji coba yaitu dengan pengurutan data acak menggunakan metode mergesort dengan interval ukuran data sebesar 100 sampai 100000 data acak. Mergesort digunakan karena relative mudah diparalelkan atau dapat bekerja secara paralel.

Alat-alat yang digunakan berupa hardware satu buah komputer yang digunakan sebagai remote kekomputer virtual. Komputer virtual yang digunakan untuk membentuk sistem cluster adalah 5 buah virtual komputer yang telah disediakan oleh sistem cloud yang berfungsi sebagai master node dan slave node dengan spesifikasi sebagai berikut:

- *Processor 1 Core.*
- *Memory 1 GB RAM.*
- *Harddisk 80 GB .*

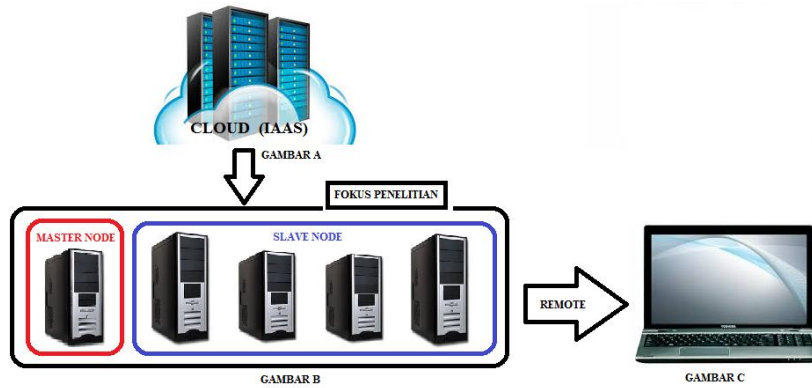
Perangkat lunak yang digunakan *master* dan *slave* dalam penelitian ini adalah sebagai berikut :

- Menggunakan *linux ubuntu v.10.04 – Lucid Lynx.*
- *Library komunikasi*
- *Network File System (NFS).*
- *Openssh server.*
- *Openssh client.*
- *Build-essential*

Tahapan pada penelitian ini ada dua :

1. Perancangan sistem virtual cluster

Dalam penelitian ini digunakan alat-alat perangkat keras komputer berupa *virtualisasi* yang membentuk sebuah sistem *cluster*. Komputer komputer tersebut terdiri dari *master* dan *slave*. *Master (head node)* merupakan node yang bertugas untuk mendelegasikan dan membagi beberapa *task* atau tugas yang akan dikerjakan oleh *slave (compute node)*. *Master* harus dapat diakses melalui jaringan yang terhubung dengan *slave computer node*, *master* juga harus menjalankan layanan seperti: NFS, SSH serta *library komunikasi*. Library yang digunakan pada penelitian ini adalah OpenMPI yang merupakan salah satu library yang digunakan untuk alat komunikasi komputasi paralel. Layanan ini diperlukan oleh *master* untuk mengintegrasikan seluruh *slave* di dalam *cluster*. *Slave (Compute node)* adalah komputer pekerja yang menerima *task* dari *master* dan memproses *task* tersebut. Komputer yang difungsikan sebagai *slave* hanya dapat diakses oleh *master* untuk melakukan proses komputasi. Perancangan yang digunakan dalam penelitian ditunjukkan pada Gambar 1.



Gambar 1 Perancangan sistem *cluster*; (a. Layanan IAAS *cloud computing*, (b) Fokus Penelitian perancangan *cluster*, (c) laptop sebagai *remote*

Pada Gambar 1 (a) adalah layanan IAAS *cloud computing* dimana dalam *cloud* tersebut dapat menyediakan layanan komputer *virtual* yang disesuaikan dengan permintaan kebutuhan penelitian ini antara lain processor, memori, jaringan dan lain lain, gambar 1 (b) adalah fokus dari penelitian ini yang merupakan gambar dari rancangan arsitektur sistem *cluster* dimana terdapat 5 komputer *virtual* yang telah dibuat oleh layanan *cloud*, dan gambar 1 (c) adalah sebuah *laptop* yang dijadikan *console* sebagai *remote* untuk seluruh komputer *virtual*, *remote* untuk akses *virtual* komputer tersebut menggunakan *virtual box*.

Pada penelitian ini menggunakan layanan *cloud* berupa IaaS. IaaS adalah layanan *cloud* yang menyediakan Infrastruktur komputasi berupa media penyimpanan, processing power, memory, sistem operasi, dan kapasitas jaringan. Penyedia *cloud* memberikan alamat IP untuk mengakses masing masing *virtual* komputer yang memiliki infrastruktur sesuai dengan kebutuhan yang diinginkan.

2. Analisis sistem *virtual cluster*

Ketika sistem *cluster* telah selesai dibangun maka untuk mengetahui kinerja dari sistem tersebut dilakukan beberapa pengujian dengan skenario yang berbeda beda. Analisis pengukuran kinerja sistem *cluster* dilakukan dengan menjalankan satu program komputasi paralel dengan skenario berbeda beda. Program yang dipilih adalah pengurutan angka dengan salah satu metode yang mudah diimplementasikan secara paralel yaitu mergesort. Program dieksekusi dengan menggunakan satu sampai lima komputer *virtual* dan dengan ukuran data yang berbeda beda. Range data yang digunakan antara 100 sampai 100000.

Kinerja komputasi paralel dipengaruhi oleh teknik pemrograman, arsitektur, atau keduanya. Parameter yang digunakan untuk mengukur kinerja sistem paralel pada penelitian ini, diantaranya adalah waktu total eksekusi, *speed-up* dan efisiensi.

Waktu eksekusi dapat diartikan sebagai waktu berlangsungnya (*running*) program paralel pada arsitektur komputer paralel yang dituju. Waktu eksekusi sekuensial didefinisikan sebagai waktu *running* algoritma yang sama yang dieksekusi oleh satu prosesor. Selain waktu eksekusi, kinerja komputasi paralel juga diukur dengan membandingkan waktu proses algoritma paralel dengan waktu proses sekuensial, dengan mendefinisikan t_s dan t_p sebagai waktu proses algoritma paralel pada prosesor tunggal dan p prosesor, maka *speed-up* dapat dirumuskan sebagai berikut pada persamaan 2.3:

$$S_p = \frac{t_s}{t_p} \quad (1)$$

dimana :

- S_p adalah peningkatan kecepatan jika menggunakan multiprosesor
- t_s sebagai waktu proses menggunakan sistem prosesor tunggal (dengan algoritma sekuensial terbaik)
- t_p sebagai waktu proses untuk menyelesaikan problem yang sama menggunakan *multiprosesor*.

Speed-up pada satu prosesor adalah sama dengan satu, dan *speedup* pada p prosesor idealnya adalah p atau bernilai $1 \leq S_p \leq p$. Secara ideal *speed-up* meningkat sebanding dengan bertambahnya jumlah prosesor. Dalam beberapa kasus dapat terjadi *superlinear speedup* ($S_p > p$), hal ini disebabkan oleh fitur unik dari arsitektur paralel, misalnya ukuran *cache* yang lebih besar pada lingkup pemrograman paralel dibandingkan dengan ukuran *cache* pada lingkup pemrograman sekuensial [3,4].

Efisiensi merupakan suatu ukuran kinerja yang sangat erat hubungannya dengan *speedup*. Secara matematis efisiensi dinyatakan dengan

$$E = \frac{S_p}{p} \quad (2)$$

dengan kisaran nilai antara $(1/p) \leq E \leq 1$ efisiensi akan menurun jika jumlah prosesor meningkat. Nilai *speed-up* dan efisiensi yang tidak ideal ini dikarenakan adanya *overhead* pada sistem paralel, Hal ini berlaku untuk semua sistem paralel, dan gejala saturasi dari *speedup* dan efisiensi ini mengikuti Hukum Amdahl [1].

3 Results and Analysis

3.1 Analisis Perancangan Sistem Cluster

Pada dasarnya semua *node* pada sistem *cluster* harus saling terhubung dan dapat berkomunikasi dengan baik untuk dapat bekerjasama mengerjakan satu masalah. Masing-masing *node* harus saling terhubung didalam satu jaringan dan memiliki identitas dengan cara memberikan *IP Address* dan *Hostname*. Pemberian identitas ini digunakan agar masing masing *node* dapat dengan mudah saling mengenali satu dengan yang lain. Kelebihan dari penggunaan service cloud yaitu mampu menyediakan virtual komputer yang telah terintegrasi dengan jaringan internet dan spesifikasi virtual komputer dapat disesuaikan dengan kebutuhan pengguna sehingga pada penelitian ini user dimudahkan dalam konfigurasi infrastruktur jaringan. Laptop yang digunakan sebagai remote untuk mengakses virtual komputer harus memenuhi spesifikasi yang mendukung teknologi yang digunakan oleh cloud. Pada penelitian ini cloud dibangun dengan menggunakan hypervisor (software virtualization) KVM yang dapat mengoptimalkan virtualisasi secara penuh dengan pemanfaatan teknologi Intel-VT yang telah di-embed dalam prosesor, sehingga laptop yang digunakan sebagai remote harus sudah mempunyai prosessor yang mendukung secara default teknologi tersebut.

Pada tahap awal perancangan jumlah virtual komputer yang digunakan pada penelitian ini sebanyak lima buah. Alamat IP ini digunakan untuk mengakses virtual komputer di cloud dengan menggunakan jaringan publik. Lima buah virtual komputer ini satu virtual komputer untuk master node dan empat buah virtual komputer untuk slave node. Peneliti memperoleh dua alamat IP untuk masing masing virtual komputer. Satu alamat IP untuk jaringan publik dan yang satu lagi untuk jaringan lokal. Jaringan lokal ini digunakan agar masing masing virtual komputer dapat berkomunikasi didalam jaringan cloud agar tidak dapat diakses dari luar, jaringan lokal ini nantinya digunakan untuk alamat IP pada sistem cluster sedangkan jaringan publik digunakan apabila kita akan mengakses virtual mesin kita dari luar atau publik. Jaringan publik digunakan untuk mengakses virtual komputer yang ada dicloud dari luar. Pada saat pembangunan sistem cluster, peneliti menggunakan jaringan publik untuk mengakses virtual komputer pada layanan cloud sesuai kebutuhan, kemudian jaringan lokal digunakan agar masing masing virtual komputer yang digunakan sebagai anggota sistem cluster dapat berkomunikasi namun yang diluar jaringan lokal tidak dapat berkomunikasi atau terhubung. Pemberian IP statis digunakan agar IP masing masing node tidak berubah, hal ini dapat memudahkan identifikasi antar anggota cluster. Kekurangan penggunaan service cloud yaitu tidak dapat memastikan bahwa komputer virtual yang digunakan benar benar homogen atau heterogen, hal ini karena dipengaruhi oleh spesifikasi server dari arsitektur cloud dan tergantung pemakaian jaringan internet yang membangun cloud, apabila jaringan dalam keadaan trafik datanya tinggi maka akan mempengaruhi kinerja dari virtual komputer menjadi tidak optimal. Konfigurasi kemudian dilanjutkan dengan bagaimana agar antar node dapat saling berkomunikasi, dapat saling bertukar data atau dapat melakukan pendistribusian data dan mengerjakan satu tugas secara bersama sama. Masing masing node harus dapat berkomunikasi dan mampu bekerja sama secara otomatis tanpa melakukan otentikasi atau ijin terlebih dahulu.

Penggunaan SSH digunakan sebagai perantara komunikasi yang berbentuk *shell* antar *node*. SSH memfasilitasi agar *master node* dapat melakukan eksekusi pada *slave node* secara *remote*. Konfigurasi SSH dengan cara *generate rsh keygen* membuat *master node* dapat melakukan otentikasi ketika akan melakukan akses pada *slave node*. Pada implementasinya, *master node* melakukan *generate public key* yang merupakan angka acak. *Public key* ini kemudian akan dikirimkan ke dalam *folder* SSH pada *slave node* kemudian diubah menjadi *authorized key*. Ketika *master node* akan melakukan koneksi SSH pada *slave node*, SSH pada *slave node* akan memeriksa apakah komputer yang akan mengakses tersebut telah terdaftar dalam *authorized keys slave node*. *Master node* dapat melakukan akses ke seluruh slave yang terdapat di dalam daftar tanpa harus memasukkan *password slave node* terlebih dahulu. Konfigurasi SSH disemua *slave node*, terlebih dahulu membuat *directory* dengan perintah *.ssh* sebagai tempat untuk file *authorized_keys*.

```
$ sudo apt-get install openssh-server
$ ssh-keygen -t rsa
$ scp .ssh/id_rsa.pub ubuntu@server-132:/home/ubuntu/.ssh/authorized_keys
```

Gambar 2 Proses *generate public key* dan mengubahnya menjadi *authorized keys* pada *master node*

Konfigurasi *master* dapat dilakukan dengan perintah terlihat seperti contoh pada Gambar 2 untuk semua anggota *master*, sehingga *master* dapat berkomunikasi keseluruhan anggota tanpa otentikasi. Ketika seluruh node dapat berkomunikasi tanpa melakukan otentikasi terlebih dahulu maka konfigurasi selanjutnya dilanjutkan dengan konfigurasi

bagaimana agar antar komputer dapat melakukan pendistribusian data dengan *file sistem* global yang dapat diakses oleh semua *node* untuk keperluan pengaksesan berkas-berkas.

Network File System (NFS) merupakan sistem *shared directory* yang memungkinkan setiap *node* terhubung pada jaringan tersebut, membagi (*share*) *directory* pada komputer yang telah ditentukan. Semua *node* di dalam jaringan yang mengaplikasikan NFS dapat mengakses *shared directory* pada NFS *server* dengan cara melakukan *mounting* NFS tersebut pada NFS *client*. NFS yang telah dilakukan proses *mounting*, dianggap oleh NFS *client* sebagai *local file sistem*.

Pada sistem *cluster* ini, *directory* yang akan di *share* adalah *directory* data. *Directory* ini digunakan untuk menyimpan *file-file* yang digunakan untuk sampel pengujian. NFS berfungsi untuk melakukan *shared directory* yang akan digunakan untuk *sharing file*. Nama-nama *directory* yang akan di *share* harus sudah ada dalam *slave* dan *master*. Konfigurasi NFS yang dilakukan pada *master node* ada di tiga lokasi *file* konfigurasi, yaitu **/etc/exports**, **/etc/hosts.allow**, dan **/etc/hosts.deny**. Prinsip kerja pada konfigurasi ketiga *file* tersebut yaitu:

- Pada *file* konfigurasi **/etc/exports**, *master node* akan melakukan identifikasi *shared directory* yang diizinkan untuk diakses.
- Ketika *slave node* akan melakukan akses *shared directory* yang ter-*mounting* dari *master node*, maka *master node* akan melakukan pengecekan apakah *slave node* yang ingin melakukan akses terdaftar pada *list* yang terdapat pada **/etc/hosts.allow** (Gambar 3) dan tidak terdaftar pada **/etc/hosts.deny** (Gambar 4).

```
portmap: 192.168.4.0/24
```

Gambar 3 Konfigurasi /etc/hosts.allow pada master node

Konfigurasi yang terlihat pada Gambar 2 yaitu *slave node* yang memiliki *IP address* dengan *range* dari 192.168.4.0 hingga 192.168.4.255 dapat melakukan akses ke komputer *master node*.

```
portmap: ALL
```

Gambar 4 Konfigurasi /etc/hosts.deny pada master node

Pada Gambar 4 memperlihatkan bahwa selain yang berada pada daftar yang tertera di **/etc/hosts.allow** maka aksesnya ditolak. Jika semua persyaratan tersebut terpenuhi maka *slave node* dapat melakukan akses ke *shared directory* *master node* tersebut. *File* konfigurasi **/etc/exports** ditampilkan pada Gambar 5. *Master node* mengidentifikasi bahwa *directory* yang dibagikan kesemua *node* terdapat pada **/home/master/data** yang merupakan tempat *file* sampel program berada.

```
/home/ubuntu/data 192.168.4.0/24 (rw, sync, no_root_squash, no_subtree_check)
```

Gambar 5 Konfigurasi /etc/exports pada master node

Pada Gambar 4, kolom pertama menyatakan lokasi *shared directory*. Kolom kedua merupakan *IP Address* yang diizinkan untuk melakukan akses NFS ke *master node*. Pada Gambar 5 bahwa *slave node* yang memiliki *IP address* dengan *range* dari 192.168.4.0 hingga 192.168.4.255 dapat melakukan akses ke komputer *master node*. Kolom ketiga terdapat beberapa opsi. Opsi **rw** merupakan opsi yang menyatakan bahwa *file* yang berada pada *directory* tersebut dapat dibaca dan ditulis (*read and write*). Opsi kedua **sync**, menyatakan bahwa *directory* tersebut akan disinkronisasi ketika terjadi *interrupt* pada sistem operasi seperti *reboot* atau *shutdown* secara tiba-tiba. Opsi ketiga **no_root_squash** menyatakan bahwa *user root* pada *slave node* memiliki hak akses yang sama seperti pada *master node*. Opsi yang keempat **no_subtree_check**, menyatakan bahwa *sub shared directory* diakses langsung melalui *slave node*. Pada *slave node*, konfigurasi yang dilakukan tidak jauh berbeda seperti yang dilakukan pada *master node*. *File* konfigurasi NFS untuk *slave node* terletak pada *directory* **/etc/fstab**.

```
server-132:/home/ubuntu/data /home/ubuntu/data nfs rw, sync, hard, intr 0 0
```

Gambar 6 Konfigurasi /etc/fstab pada slave node pada sistem cluster

Gambar 6 menunjukkan konfigurasi **/etc/fstab** yang dilakukan pada *slave node*1. Pada kolom pertama merupakan parameter yang menyatakan *IP address* komputer yang di-*mounting* oleh *slave node*, dalam penelitian ini merupakan *master node*. Kolom kedua adalah lokasi *shared directory* pada *master node*. Kolom ketiga adalah tempat *directory* hasil *mounting* *master node* pada *slave node*. Perbedaan opsi di **/etc/fstab** pada *slave node* dengan **/etc/exports** pada *master node* adalah pada *slave node* terdapat opsi **hard** dan **intr**. Opsi **intr** adalah sistem akan memberikan *interrupt* jika pada saat *mounting* terdapat gangguan, sedangkan opsi **hard** adalah program yang mengakses *file* pada *directory* hasil *mounting*. Hang terjadi jika mesin *master node* mengalami *crash*. Konfigurasi *fstab* digunakan agar permanen sehingga jika komputer di *restart mount* tidak akan lepas. NFS akan melakukan *mounting* ketika sistem operasi dilakukan proses *restart*, setelah konfigurasi pada *master* dan *slave node* dilakukan.

Ketika sistem cluster sudah dapat berkomunikasi tanpa otentikasi dan dapat saling bertukar data atau melakukan pendistribusian data maka masing masing node harus memiliki sebuah library komunikasi sebagai alat komunikasi sehingga masing masing *node* dapat berkomunikasi untuk melakukan proses paralel. Pada penelitian ini menggunakan OpenMPI karena library ini merupakan library yang sering digunakan pada sistem cluster dan merupakan library yang memiliki kinerja yang baik dan bersifat *direct* sehingga tidak perlu menggunakan *diamond* untuk mendaftarkan anggotanya dalam satu sistem *cluster*. Seluruh komputer yang terdapat dalam satu jaringan dan sudah terinstall *openMPI* dapat menjadi anggota sistem *cluster*. Pada saat sebelum menginstall OpenMPI terlebih dahulu menginstall *build-essential*. *Build-essential* berisi paket *GCC*, *G++*, dan *Gfortran* yang merupakan *compiler* program bahasa pemrograman *C*, *C++*, dan *fortran* pada platform linux. Langkah instalasi aplikasi OpenMPI menggunakan perintah \$ *sudo apt-get install libopenmpi-dev openmpi-bin openmpi-doc*. Proses instalasi dilakukan diseluruh *node*.

Program paralel dijalankan di *master node* dengan kondisi komputer *slave* telah ter-*mounting*, sehingga memiliki satu direktori yang terbagi sama untuk menyimpan program yang akan dieksekusi. Program paralel terlebih dahulu harus dikompilasi dengan menggunakan perintah **Mpicc.openmpi -o nama_file_kompilasi nama_file** agar dapat berbagi proses / tugas secara paralel. Pemrograman paralel dapat dilakukan secara paralel dengan membagi tugas / proses ke semua *slave node*, dengan perintah berikut ini:

Mpirun.openmpi -np jumlah_proses -H nama_hostname_slave_node nama_file_kompilasi

Perintah mengkompilasi dan menjalankan pemrograman paralel di dalam *cluster* dapat dilakukan dalam beberapa opsi, yaitu :

- *mpicc* merupakan kompilasi program bahasa C, *mpicc* akan memanggil kompilasi gcc (*GNU C Compiler*).
- *mpirun* merupakan perintah untuk menjalankan program yang telah dikompilasi
- *-np* adalah *number of process* merupakan opsi yang mengidentifikasi berapa banyak prosesor yang akan digunakan dalam menjalankan proses komputasi ini.
- *-H* merupakan perintah untuk mengidentifikasi berapa *hostname* yang akan digunakan untuk menjalankan program tersebut serta digunakan untuk mendefinisikan host host yang terdaftar sebagai anggota sistem *cluster*.

3.2 Analisis Pengujian Sistem Cluster

Tujuan pengujian sistem *cluster* dalam penelitian ini untuk menguji kinerja sistem virtual *cluster* yang telah dibangun dalam menyelesaikan suatu masalah komputasi yaitu pengurutan data acak menggunakan komputasi paralel. Kinerja suatu sistem cluster pada komputasi paralel dapat diketahui dari besarnya waktu yang digunakan dalam menyelesaikan suatu program, *speed-up* dan efisiensi. Pengujian *cluster* pada penelitian ini dilakukan dengan menyelesaikan masalah pengurutan data acak menggunakan metode *mergesort* dengan interval jumlah data acak yaitu antara 100 sampai dengan 100000 kemudian dijalankan dengan menggunakan *sequensial* dan paralel antara dua *node* sampai dengan lima *node*. Pemilihan jumlah data antara 100 sampai dengan 100000 bilangan acak bertujuan agar dapat memperlihatkan hasil perbedaan waktu eksekusi yang signifikan. Skenario ini bertujuan untuk melihat pengaruh perubahan jumlah *node* dengan banyaknya jumlah data terhadap waktu eksekusi pada saat menggunakan virtual cluster dari sistem cloud. Metode *mergesort* dipilih sebagai program yang digunakan untuk pengujian sistem cluster karena merupakan algoritma pengurutan data yang telah didisain untuk dapat bekerja dalam mesin-mesin yang memiliki banyak prosesor dan mampu dieksekusi pada komputasi paralel. Dari hasil pengujian diperoleh tabel 3 yang menunjukkan waktu total eksekusi, *speed-up* dan efisiensi, setiap pengujian yaitu dari interval banyaknya data 100 sampai 100000 yang dijalankan menggunakan satu *node* (*sequensial*) sampai lima *node*.

Tabel 1 waktu total, *speed-up* dan efisiensi pengujian cluster

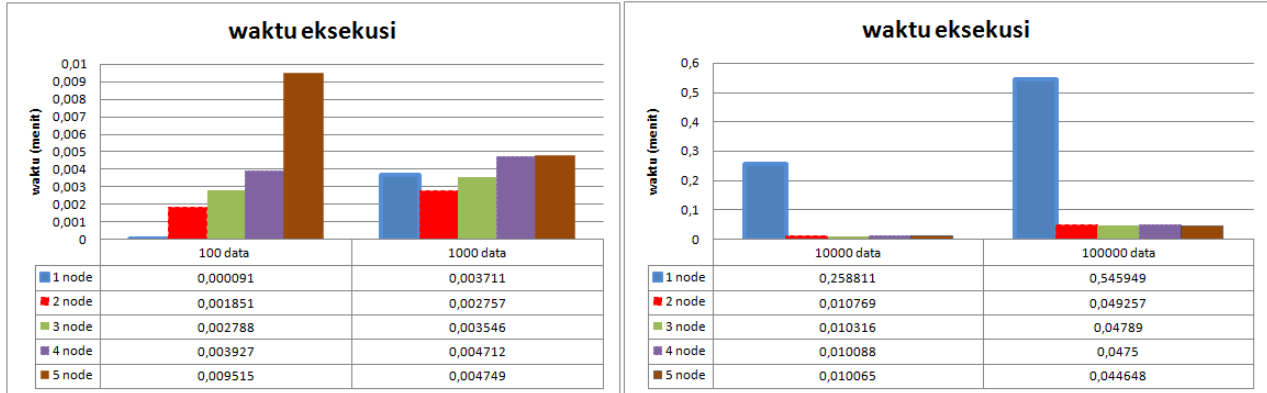
Mergesort	node	ukuran data openMPI			
		100	1000	10000	100000
Waktu Total Eksekusi (detik)	1	0.000091	0.003711	0.258811	0.545949
	2	0.001851	0.002757	0.010769	0.049257
	3	0.002788	0.003546	0.010316	0.04789
	4	0.003927	0.004712	0.010088	0.0475
	5	0.009515	0.004749	0.010065	0.044648
Speed-up	2	0,049163	1,346028	24,032965	11,083684
	3	0,032640	1,046531	25,088309	11,400063
	4	0,023173	0,787564	25,655333	11,493663
	5	0,009564	0,781428	25,713959	12,227849
efisiensi	2	0,024581	0,673014	12,016482	5,541842
	3	0,010880	0,348844	8,362770	3,800021
	4	0,005793	0,196891	6,413833	2,873416
	5	0,001913	0,156286	5,142792	2,445570

Pada Tabel 3 menunjukkan bahwa pada sistem virtual cluster dengan ukuran data kecil contoh pada pengujian ini ukuran data 100 data acak tidak cocok digunakan karena semakin banyak node yang digunakan maka waktu eksekusi total yang dibutuhkan semakin besar namun ketika ukuran data acak semakin besar sistem virtual cluster mulai bekerja dengan baik seperti pada data berukuran 10000. Hal ini dipengaruhi karena semakin bertambah jumlah *node* yang digunakan maka semakin bertambah pula waktu komunikasi dan sebaliknya semakin berkurang untuk waktu komputasi yang dilakukan masing masing node sesuai dengan prinsip kerja dari sistem cluster. Prinsip kerja dari sistem cluster yaitu setiap langkah komputasi yang dilakukan oleh sistem akan dibagi kesemua anggota sehingga setiap prosesor pada setiap komputer anggota mengerjakan langkah komputasi yang kecil atau sedikit dan sebaliknya semakin banyak jumlah komputer berarti semakin banyak langkah komunikasi yang dilakukan selama menyelesaikan masalah pengurutan data, komunikasi untuk pendistribusian data antar prosesor dan kanal komunikasi pada sistem cluster seperti pada gambar 7. Pada gambar 7 merupakan gambar software HTOP yang digunakan untuk memonitoring kinerja dari CPU virtual komputer seluruh anggota cluster. Pada software HTOP terlihat bahwa pada saat awal bekerja node komputer master bekerja terlebih dahulu untuk membagikan tugas kepada anggotanya atau komputer komputer slavenya, kemudian ketika komputer slave bekerja maka komputer master menunggu hasil kerja masing masing anggotanya sehingga terlihat seperti pada gambar 7.

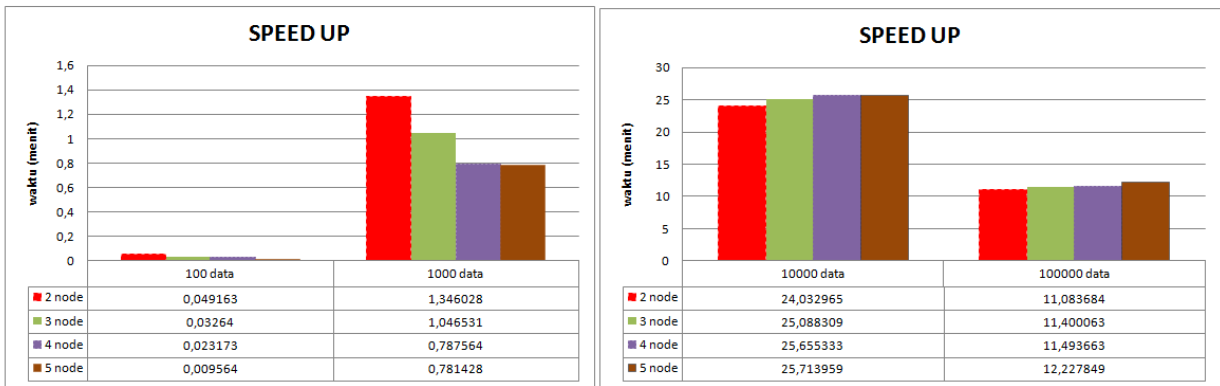


Gambar 7 Monitoring sistem cluster menggunakan HTOP

Ketika jumlah data acak berukuran kecil kemudian diselesaikan dengan menggunakan sistem cluster maka semakin banyak node yang bekerja mengakibatkan kinerja dari sistem cluster menjadi semakin turun. Waktu eksekusi yang dibutuhkan dalam pengurutan angka semakin lama dibandingkan dikerjakan secara sequential, hal ini terlihat ketika data yang diurutkan sebanyak 100 data acak. Waktu eksekusi yang dibutuhkan semakin lambat ketika menggunakan banyak node atau banyak komputer dibandingkan dengan satu komputer ini disebabkan karena waktu yang dibutuhkan untuk eksekusi lebih banyak dihabiskan untuk berkomunikasi. Langkah komputasi tambahan akibat komunikasi antar node yang lebih banyak dibandingkan langkah komputasinya sendiri bisa menyebabkan waktu eksekusi semakin lama dibandingkan sequential. Ketika jumlah data acak semakin besar yaitu seperti pada saat 100000 data kemudian diselesaikan dengan menggunakan banyak komputer akan semakin cepat dibandingkan dengan dikerjakan secara sequential menggunakan satu komputer. Hal ini disebabkan karena ketika diselesaikan menggunakan satu komputer maka beban kerja prosesor tidak seimbang yang menyebabkan lambatnya waktu eksekusi. Namun ketika dikerjakan dengan menggunakan banyak komputer beban masing masing prosesor semakin berkurang dan seimbang sehingga mempercepat waktu eksekusi seperti yang terlihat pada Gambar 8. Kesimpulan dari hasil pengujian diperoleh bahwa kinerja dari virtual cluster menggunakan layanan infrastruktur cloud mulai nampak pada saat pemakaian data dengan ukuran besar atau komputasi yang besar atau rumit, sedangkan data berukuran kecil yang memiliki langkah komputasi sederhana lebih cepat menggunakan satu komputer virtual saja.



Gambar 8 Grafik perbandingan waktu eksekusi total pada data acak interval 100 sampai 100000

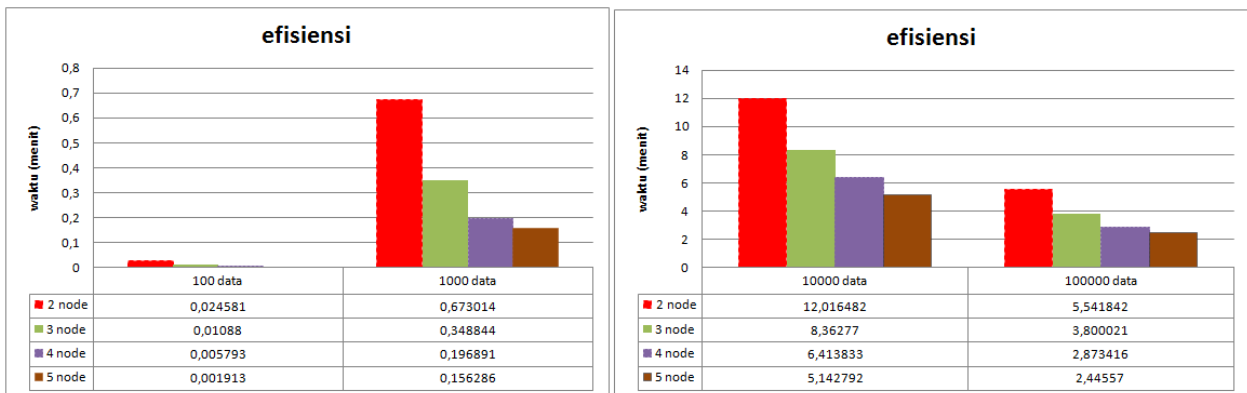


a. speed up pada interval data 100 sampai 1000

b. speed up pada interval data 10000 sampai 100000

Gambar 9 Grafik perbandingan speed-up pada data acak

Kinerja dari paralel komputing dapat dilihat dari nilai *speed-up* dan *efisiensi*. Nilai *speed-up* semakin menurun ketika jumlah anggota pada sistem cluster atau jumlah node komputer semakin bertambah ketika ukuran data jumlahnya sedikit yaitu 100 sampai 1000 data acak seperti terlihat pada gambar 9a, namun ketika ukuran data diperbesar hingga 100000, nilai *speed-up* smakin meningkat seperti terlihat pada gambar 9b. Secara ideal nilai *speedup* meningkat sebanding dengan bertambahnya jumlah prosesor, dari hasil pengujian tersebut maka sistem cluster cocok digunakan ketika ukuran data besar atau banyak dan tidak cocok digunakan ketika ukuran data sedikit atau kecil. Nilai *speed-up* yang ideal ketika nilai *speed-up* berada diantara lebih dari atau sama dengan satu atau kurang dari atau sama dengan jumlah prosesor, namun dari hasil pengujian hanya ketika ukuran data sebesar 1000 data acak dan pada saat penggunaan dua sampai tiga node komputer saja yang sesuai dengan syarat ideal nilai *speed-up* seperti pada saat penggunaan dua node sebesar 1,346 dan tiga node sebesar 1,0465. Ketika data diatas 100000 terjadi *superlinear speedup* nilai *speedup* jauh lebih besar dari jumlah prosesor yang digunakan.



Gambar 10 Grafik perbandingan efisiensi pada data acak

Efisiensi merupakan suatu ukuran kinerja yang sangat erat hubungannya dengan *speed-up*. Semakin besar jumlah node komputer atau anggota dalam sistem cluster menyebabkan semakin menurunnya efisiensi seperti terlihat pada gambar grafik 10 dan tabel 3. Nilai *speedup* dan efisiensi yang tidak ideal ini dikarenakan adanya *overhead* pada sistem paralel, misalnya komputasi tambahan yang hanya dibutuhkan pada sistem paralel, komunikasi antar-prosesor, dan proses sinkronisasi pada sistem cloud sebagai penyedia sarana komputer virtual.

4 Conclusion

Dari hasil pengujian dan analisis dapat ditarik kesimpulan bahwa:

1. Penggunaan layanan cloud sebagai penyedia sumber daya sistem dapat memudahkan dalam pembangunan sistem cluster karena sumber daya yang telah disediakan sudah terdapat dalam satu jaringan sehingga tidak perlu melakukan konfigurasi infrastruktur jaringan antar anggota cluster.
2. Masing masing virtual komputer memiliki dua alamat IP yaitu alamat untuk jaringan publik dan jaringan lokal. Jaringan publik digunakan untuk mengakses layanan IAAS berupa virtual komputer dari jaringan cloud dan penggunaan alamat IP jaringan lokal digunakan agar komputer virtual yang bukan anggota cluster tidak dapat berhubungan atau berkomunikasi sehingga kinerja dari jaringan sistem cluster dapat optimal.
3. Proses komputasi pengurutan data acak dengan metode mergesort yang dijalankan secara paralel dengan menggunakan virtual komputer dari layanan IAAS cloud dapat berjalan dengan baik ketika ukuran data yang dieksekusi besar dan rumit yaitu semakin banyak jumlah node yang digunakan maka waktu eksekusi semakin kecil hal ini terlihat ketika sistem melakukan pengurutan angka diatas 10000 angka, namun ketika data yang diurutkan memiliki ukuran kecil sistem cluster tidak berjalan dengan baik, karena semakin banyak node yang digunakan mengakibatkan waktu eksekusi total semakin lambat contohnya ketika data yang diurutkan sebanyak 100 angka.
4. Kinerja virtual cluster dengan menjalankan komputasi pengurutan data acak memiliki nilai *speed-up* yang ideal ketika ukuran data acak sebesar 1000 data dengan menggunakan dua sampai tiga node hal ini dikarenakan adanya *overhead* pada sistem paralel, berupa komputasi tambahan yang hanya dibutuhkan pada sistem paralel, komunikasi antar-prosesor, dan proses sinkronisasi pada sistem cloud sebagai penyedia sarana komputer virtual.

Beberapa saran yang dapat digunakan sebagai pengembangan penelitian ini lebih lanjut antara lain yang perlu diperhatikan:

1. Pada pembangunan sistem cluster perlu diketahui spesifikasi server cloud (privat atau publik) penyedia komputer virtual homogen atau heterogen, sehingga dapat lebih spesifik dalam melakukan pengujian kinerja dari sistem cluster tersebut.
2. Sistem cluster membutuhkan library komunikasi untuk berkomunikasi agar dapat melakukan kerja sama antar anggota dalam menyelesaikan suatu tugas, maka diperlukan penelitian tentang analisis karakteristik library pada sistem cluster, sehingga pengguna dapat menggunakan library pada sistem cluster sesuai kebutuhan.

Acknowledgements

Penulis ucapkan terimakasih kepada segenap dosen pengajar di MTI UGM terutama Bapak Teguh Bharata Adji, S.T., M.T., M.Eng., Ph.D, dan Bapak Ir. Sujoko Sumaryono, M.T. selaku pembimbing dalam penelitian cloud dan cluster di MTI UGM, dan segenap tim riset cloud dan cluster MTI UGM atas bantuannya hingga tersusun jurnal tentang virtual cluster ini.

References

- [1] Faisal Nggilu., "Analisis *Overhead* Sebagai Salah Satu Faktor Skalabilitas *Private Cloud Computing* Untuk Layanan *IAAS*" Tesis, Magister Teknologi Informasi., Universitas Gadjah Mada, Yogyakarta, 2012.
- [2] Purnomo, Membangun Virtual PC dengan VirtualBox. Penerbit Andi Yogyakarta. Yogyakarta 2010
- [3] Syaputra. D., "*Uji Kinerja OCTAVE Dengan MPITB Menggunakan Pustaka LAM/MPI dan OpenMPI*" Tesis, Magister Teknologi Informasi., Universitas Gadjah Mada, Yogyakarta, 2010.
- [4] Wilkinson, B. dan Allen, M., "Parallel Programming", New Jersey: Prentice-Hall, Inc. 1999.
- [5] Yudistira, R. 2008. Multi-Core VS. PC-Clustering Terhadap Kinerja Pengolahan Paralel.