RANCANG BANGUN LOW-COST CLUSTER - HIGH PERFORMANCE COMPUTING

Tiara Nursyahdini[1] Fitriyani, S.si.,M.T.[2] Izzatul Ummah S.T., M.T.[3]

[1,2,3]Prodi Ilmu Komputasi Telkom University, Bandung

[1]tiaranursyahdini@gmail.com [2]fitriyani.y@gmail.com [3]izzatul.ummah@gmail.com

Abstrak

Bidang komputasi mengalami kemajuan yang sangat signifikan setiap tahunnya, semua hal tersebut dibuat untuk satu tujuan, membuat ukuran perangkat semakin *compact* yang digunakan untuk pengolahan data yang semakin besar. Kebutuhan akan komputer berkinerja tinggi yang sangat erat hubungannya dengan *supercomputer* dan *massively parallel processors* (MPP) menurut beowulf telah dapat dipenuhi oleh komputer *cluster*. Supercomputer and MPP sangat kompleks untuk dikembangkan dan membutuhkan biaya yang besar, sedangkan komputer *cluster* dapat dibangun dari komputer-komputer sebagai *node* dengan harga yang lebih murah dengan jaringan berkecepatan tinggi. Perangkat lama yang semakin ditingggalkan dan menjadi tumpukan menjadi ide untuk memanfaatkan beberapa perangkat komputer lama menjadi satu kesatuan (pararel) yang memiliki performa lebih tinggi. Kinerja komputer secara pararel akan meningkatkan waktu komputasi, hal ini lah yang menjadi dasar dibuatnya komputer *cluster*.

Penelitian tugas akhir ini akan membahas bagaimana *low cost cluster* dapat digunakan untuk mengerjakan kasus komputasi kinerja tinggi, dalam hal ini rendering animasi. Pada *rendering*, mutlak membutuhkan kinerja komputer yang mumpuni. Jika sekuel Lord of The Ring dikerjakan dengan personal computer (PC) yang tercepat saat ini, dibutuhkan waktu lebih dari 10 tahun untuk menyelesaikan proses rendering. Bagi para artis grafis 3D atau animator, hal yang paling menyita waktu adalah menunggu proses rendering. Di Indonesia sendiri, produsen animasi rumahan sudah mulai menggeliat, membuat *renderfarm* sederhana yang berbiaya rendah, akan jadi pilihan yang efisien daripada menyewa atau membeli sejumlah komputer server yang berbiaya tinggi. Penelitian ini menggunakan PC desktop sejumlah 32 *node*, dengan 1 *master* untuk membuat *renderfarm*. Sistem operasi yang digunakan adalah Linux, dengan *rendering tools* yaitu blender dan brender.

Hasil dari penelitian ini menunjukkan bahwa peningkatan jumlah *node* dua kali lipat tidak linear dengan peningkatan waktu komputasi. Speed up terbaik pada ray tracing rendering didapat pada kenaikan 4 node menjadi 8 node, yaitu 93%, speed up cycles rendering terbaik terdapat pada peningkatan 8 node menjadi 16 node yaitu sebesar 91%, dan speed up motion capture terbaik ada pada peningkatan 16 node menjadi 32 node, yaitu sebesar 95%.

Kata kunci : komputer *cluster*, komputasi kinerja tinggi, *rendering*, *renderfarm*, blender, brender.

1. Pendahuluan

Komputasi berkinerja tinggi (*high performance computing*) dapat dikaitkan dengan sebuah metode untuk meningkatkan kinerja dari sebuah aplikasi. Hal ini meliputi pembagian sebuah pekerjaan (program aplikasi) ke dalam beberapa unit paralel yang memungkinkan dan berkerja secara simultan untuk meningkatkan kecepatan dalam penyelesaian pekerjaan tersebut[1].

Kebutuhan akan komputer berkinerja tinggi yang sangat erat hubungannya dengan *supercomputer* dan *massively parallel processors* (MPP) telah dapat dipenuhi oleh komputer *cluster*. Supercomputer and MPP sangat kompleks untuk dikembangkan dan membutuhkan biaya yang besar, sedangkan komputer kluster dapat dibangun dari komputer-komputer sebagai *node* dengan harga yang lebih murah dengan jaringan berkecepatan tinggi. Komputer kluster dapat dibuat dari komputer- komputer *node* yang masing-masing terdiri dari satu atau lebih *processor, memory*

yang dibagi (di-*shared*) oleh semua *processor* di dalam *node*, dan *device* lainnya seperti *disk*, dan terhubung dengan sebuah jaringan yang mengijinkan perpindahan data antar-*node* suatu sistem tersebut [2].

Benchmark yang dilakukan oleh Bailey, D; Barton, J; Lasinski, T; & Simon, H.[3] dalam *The NAS Parallel Benchmarks* menggunakan teori beowulf mengemukakan bahwa kluster yang dibangun dari PC bisa bekerja sama baiknya dengan *supercomputer* maupun MPP.

Apabila sedikit dikaitkan dengan isu lingkungan saat ini, komputer *cluster* dapat mengurangi sampah elektronik yaitu komputer, karena sampah komputer yang awalnya tidak dapat digunakan karena kemampuannya sudah tertinggal dibandingkan komputer-komputer yang ada saat ini dapat digabungkan menjadi suatu kluster computer yang kinerjanya lebih baik. Dengan ini salah satu konsep penghijauan lingkungan yaitu penggunaan kembali dapat diimplementasikan.

Untuk itu dalam studi tugas akhir ini penulis akan menerapkan langkah langkah dan kebutuhan yang diperlukan dalam pembangunan *low cost* PC kluster dengan analisis performansi dengan studi kasus rendering menggunakan sistem operasi *open source* linux.

2. Tinjauan Pustaka

2.1 Komputer *Cluster*

Secara harafiah, *clustering* berarti pengelompokan. PC-*clustering* dapat diartikan pengelompokan beberapa buah PC menjadi satu kesatuan dan mampu memproses dengan interkoneksi jaringan baik itu lokal maupun internet.

Penggunaan PC klustering jamak dilakukan pada institusi pendidikan maupun perusahaan sebagai pengembangan dari komputer multi prosesing. Pada dasarnya pada sistem ini, sebuah PC dapat diibaratkan satu buah elemen pemroses, untuk dapat menggunakan PC sebagai bagian dari suatu elemen pemroses dibutuhkan software yang sebelumnya telah didesain untuk keperluan tertentu.[7]

Suatu komputer yang ter-cluster memiliki suatu arsitektur tertentu, dimana arsitektur tersebut memungkinkan suatu komputer dapat berkomunikasi antar komputer satu dengan lainnya. Arsitektur tersebut terdiri dari dua komponen yaitu komponen hardware dan software. Sekumpulan komputer (pada suatu jaringan komputer) independen yang beroperasi dan terlihat oleh client jaringan seolah-olah komputer-komputer tersebut adalah satu buah unit komputer. Clustering dirancang untuk meningkatkan kemampuan kinerja dari komputer- komputer yang berada pada suatu jaringan komputer untuk dapat meningkatkan hal-hal berikut:

1. Toleransi kesalahan (fault tolerance), yang dapat menyebabkan node lainnya (misal komputer B) akan mengambil alih kerja node utama (sebutan untuk node yang melakukan eksekusi program tertentu, misal komputer A) ketika node 4 tersebut mengalami kegagalan. Client tidak akan melihat pergantian peran ini. Dengan begitu, downtime pun dapat dikurangi secara drastis.
2. Penyerataan beban (load-balancing), yang dapat mendistribusikan beban satu node ke semua node anggota cluster. Dengan begitu, kinerja dan skalabilitas node utama pun menjadi relatif lebih baik. Bagian terpenting dari komputer cluster adalah adanya sebuah aplikasi middleware yang dapat menggabungkan seluruh anggota dalam cluster sehingga dapat bekerja sama. Tugas utama dari aplikasi middleware ini adalah untuk komunikasi dan sinkronisasi antar komputer.

(1)

2.2 *Network File System*

Network File System (NFS)[6] adalah sarana untuk menghubungkan disk-disk pada sistem jarak jauh dengan sistem lokal, serta memvisualisasikan penempatan disk tersebut seakan pada lokasi fisik yang sama. Dengan demikian memungkinkan untuk melakukan mount file-file dari komputer yang berbeda melalui jaringan TCP/IP. NFS merupakan protokol tak bertempat. Setiap permintaan dari *compute node*dan *head node*bersifat unik atau berlaku untuk dirinya sendiri. Artinya NFS memiliki sifat yang mapan. Bila *head node*down, *compute node*tidak harus reboot. Berdasarkan desainnya, NFS termasuk lingkungan yang tidak aman. Bila ingin berbagi data atau ruang disk dengan aman, perlu pertimbangan lain sebagai alternatif NFS. Ada 3 komponen dasar dari NFS [6]:

1. Serangkaian protokol TCP/IP
2. *head node*yang menggunakan proses mengekspor sistem file (NFS *head* node)
3. *compute node* yang bisa di-mount otomatis dengan file /etc/fstab saat booting (NFS *compute node*)

2.3 *Bash Shell*

Shell adalah command executive, artinya program yang menunggu instruksi dari pemakai, memeriksa sintak dari instruksi yang diberikan, kemudian mengeksekusi perintah tersebut. Shell ditandai dengan prompt. Untuk pemakai menggunakan prompt $ dan untuk superuser menggunakan prompt #. Bash script adalah file yang berisi koleksi program yang dapat dieksekusi. Untuk eksekusi bash script menggunakan tanda . (dot) sebelum file bash script yang berarti eksekusi shell dan tanda ./ berarti file bash script barada pada direktori aktual. Secara default dalam Linux menggunakan bash shell.

2.4 *Distribute Manager (DRM)*

Distributed Resource Management (DRM) adalah suatu sistem yang dapat mengatur pemanfaatan sumber daya terdistribusi untuk menjalankan suatu job. Penggunaan sekumpulan mesin yang terhubung dalam sebuah jaringan demi penyediaan sumber daya komputasi memerlukan sebuah sistem yang dapat mengatur penggunaan sumber daya yang ada tersebut. Pengaturan diperlukan agar sumber daya yang ada dapat digunakan secara optimal. DRM ini sering juga disebut sebagai sistem penjadwalan job (job scheduler) karena tugasnya memang melakukan penjadwalan eksekusi job dalam sumber daya yang tersedia. Saat sebuah job dikirimkan, job akan masuk ke dalam sebuah antrian job (jobs queue). Saat ada sumber daya yang dapat digunakan, job dalam antrian tadi akan diberikan 10 ke sumber daya untuk dieksekusi. Dalam disebutkan bahwa ada beberapa komponen dalam suatu DRM, yaitu :

1. *Batch queueing* sebagai tempat antrian job yang dikirimkan. *Job* akan berada pada antrian ini sampai ada mesin yang siap untuk menjalankan job tersebut.
2. *Resource management* yang bertugas untuk mengirimkan *job* yang berada dalam antrian ke sumber daya lalu menjalankannya.

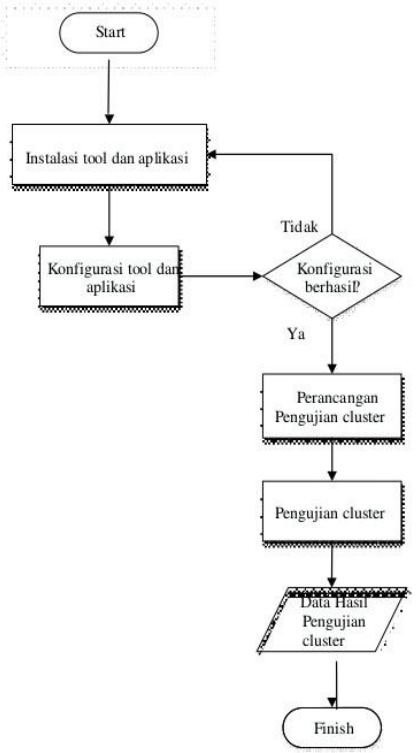
3. Perancangan Sistem

Perancangan sistem dibagi menjadi 3 buah tahap utama. Tahap pertama adalah menyiapkan komponen-komponen hardware pendukung, kedua adalah instalasi cluster, dalam hal ini khususnya renderfarm, ketiga adalah pengolahan data pada cluster, yaitu rendering animasi.

Perancangan sistem dimulai dengan pemilihan hardware dengan spesifikasi tertentu sesuai dengan kebutuhan cluster, spesifikasi pada setiap komputer didalam cluster harus memiliki spesifikasi yang sama agar cluster dapat berjalan dengan baik.

Tahap selanjutnya adalah proses networking, networking dilakukan pada komputer-komputer yang akan digabungkan menjadi sebuah cluster agar setiap komputer dapat terhubung kedalam sebuah jaringan dan dapat saling berkomunikasi satu dengan lainnya.

Tahap selanjutnya adalah instalasi cluster renderfarm, jika tahap instalasi sudah selesai maka proses selanjutnya adalah proses rendering. Setelah rendering dilakukan didapatkan hasil yang selanjutnya akan dianalisis. Ilustrasi dapat dilihat pada gambar dibawah ini:



Gambar 3-1 Perancangan Sistem

3.1 **Spesifikasi Perangkat Keras dan Perangkat Lunak**

Sistem yang digunakan untuk implementasi klaster komputer perlu di *set up* terlebih dahulu. Perangkat yang akan digunakan sesuai dengan prinsip low cost adalah sebagai berikut :

1. *33 PC quad core* yang dihubungkan dengan switch.
2. *Linux Ubuntu 12.4* (untuk master node)
3. *Centos 7* (untuk slave node)

4. *SSH Client-Server*
5. *Blender 2.68*
6. *Brender*

3.2 *Networking*

Pada pembuatan tugas akhir ini, penulis merancang PC cluster yang bertujuan untuk melakukan proses *rendering* secara paralel untuk dibandingkan dengan proses *rendering* pada komputer tunggal. *Cluster* yang difungsikan untuk paralel *rendering* sering disebut *render farm*.

Topologi jaringan yang digunakan adalah topologi star (bintang) dan menggunakan internet protocol (IP) kelas C (192.168.1.0/24). Jumlah nodes berjumlah 32, dan satu master node.

3.3 *Installasi Cluster*

Berikut ini adalah tahapan-tahapan utama untuk membangun Cluster Renderfarm:

3.4.1 Persyaratan

Berikut ini merupakan beberapa persyaratan sebelum melakukan instalasi cluster renderfarm:

1. PC master sudah terinstall sistem operasi Ubuntu, dan PC node diinstall sistem operasi centos.
2. Install php dan mysql pada Master Node dan Slave Node. Untuk menginstall, ketikkan yum install php php-mysql httpd . dan install juga browser pada komputer master.
3. Pastikan IPTABLES dalam keadaan disable melalui chkconfig ketikan perintah iptables –L –v –n untuk memastikan IPTABLES sudah dalam keadaan disable, hal ini dilakukan untuk mematikan firewall, dan akan secara otomatis tetap dalam kondisi mati, walaupun router dimatikan.

3.4.2 Konfigurasi Jaringan

Untuk memudahkan proses instalasi sampai dengan proses penggunaan cluster maka IP (Internet Protocol) pada Master Node dan seluruh Slave Node dibuat menjadi IP Static, ini berfungsi agar kita tidak perlu lagi melakukan setting IP jika Master Node atau Slave Node mati atau direstart. Setelah memastikan seluruh IP pada Master Node dan Slave Node adalah IP Static kemudian edit file /etc/hosts pada Master Node dan Slave Node dengan mengetikan perintah cat /etc/hosts. Tambahkan IP dan hostname komputer-komputer lain pada masing-masing komputer agar setiap komputer dapat melakukan komunikasi.

3.4.3 Konfigurasi Secure Shell (SSH)

Lakukan konfigurasi SSH pada Master Node dan Slave Node agar dapat mengendalikan node-node atau mentransfer file antara node. Berikut ini adalah langkah-langkah untuk melakukan konfigurasi SSH:

Masuk sebagai root pada Centos dengan mengetikan perintah su pada terminal.

1. Install open ssh server dan client dengan mengetikan perintah yum –y install openssh-server openssh-clients.
2. Setelah proses instalasi selesai lakukan konfigurasi pada sshd_config dengan mengetik perintah cat /etc/ssh/sshd_config.

3. Ganti Permit Root Login yang semula "YES" ubah menjadi "NO"
4. Ganti Permit Empty Password yang semula "YES" ubah menjadi "NO" Buat user authentication untuk komputer-komputer agar dapat mengakses ssh server. Contoh: Allow User Master Node.
5. Save ssh_config.
6. Restart ssh dengan mengetikan perintah /etc/ini.d/ssh restart.
7. Lakukan hal yang sama untuk setiap node.

3.4.3 *Installasi* Blender

Berikut adalah tahapan installasi Blender, sebagai software utama untuk rendering animasi:

1. Blender akan diinstall di setiap node, pastikan service network sudah dikonfigurasi, node telah terhubung dengan masternya. Edit file ifcfg-eth0 , lalu cek dengan ping ke master, ketikkan perintah pada terminal, ping 192.168.1.254. Setelah itu, berikan nama pada setiap node, node 1 dengan nama dom01,dan seterusnya. Dilakukan hal yang sama pada node 2 sampai dengan node 40.
2. Buat directory pada setiap node, dengan nama /media/farm/blender, ketikkan perintah chown dom01:down01 /media/
3. Buat akses pada folder blender agar bisa baca dan eksekusi (*full access*), ketikkan perintah chmod 755 blender. 7 menandakan hal akses untuk user, 5 untuk akses grup dan 5 untuk akses lainnya..
4. Buat akses agar folder bisa mounting secara permanen, install red hat package manager, ketikkan perintah rpm –ivh SDL-1.2.12-3.el6.x86_64.rpm, setelah terinstall , lakukan konfigurasi pada mount –a
5. Lakukan hal yang sama untuk setiap node.

3.4 Pengecekan Sistem

Dilakukan langkah-langkah pengecekan sistem,sebagai berikut:

1. Buka web browser, akses brender di 192.168.1.254 sesuai dengan konfigurasi yang telah dilakukan
2. Jalankan brender dengan job yang sudah tersedia
3. Periksa kinerja sistem ram, cpu, dengan mengetikkan pada terminal, Ktikkan perintah top –b –n 1 | grep "blender" untuk mengecek kinerja sistem blender.

4. Pengujian dan Analisis

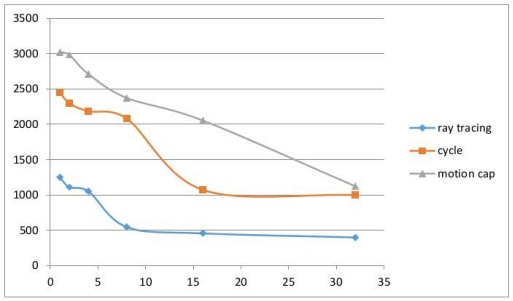
4.1 Skenario Pengujian

Pada tahap ini dilakukan pengujian terhadap setiap node. Node yang diuji adalah node 1,2,4,8,16,dan 32. Dilakukan rendering dengan 3 buah metode berbeda pada masing masing node, sebanyak 3 kali setiap metodenya dengan jumlah frame yang berbeda. Hasil uji terdapat pada lampiran, berikut adalah nilai waktu (menit) rata-rata hasil uji pada 1500 frame:

No	Metode	1 Node	2 Node	4 Node	8 Node	16 Node	32 Node
1	Ray Tracing	1249,33	1109,11	1050,98	545,64	456,12	398,08
2	Cycle	1698,87	1793,21	1984,49	2083,43	1071,83	987,85
3	Motion Capture	2872,34	2982,58	2710,92	2374,31	2053,90	1018,95

Tabel 4-1 Rata-rata Hasil Uji (menit)

Uji sistem dilakukan dengan 3 metode, metode ray tracing(abu-abu), metode cycle (merah), dan metode motion capture(biru), berikut rata-rata hasil uji dalam grafik:



Gambar 4-1 Rata-Rata Hasil Ujii (menit)

4.2 Uji Processors

Dari hasil uji rendering, dilakukan *monitoring* pada penggunaan RAM dan Processor, dan didapatkan hasil uji sebagai berikut:

1. Hasil uji menyatakan bahwa ketika rendering mulai dijalankan Processor bekerja 85% .
2. Ketika proses render selesai, processor dalam posisi idle, dan hanya penggunakan source sebesar 10% saja.

4.3 Uji Memory

Dari hasil uji rendering, dilakukan *monitoring* pada penggunaan RAM dan Processor, dan didapatkan hasil uji sebagai berikut:

1. Hasil uji menyatakan bahwa ketika rendering mulai dijalankan, RAM dan processor bekerja 100% .
2. Ketika proses render selesai, RAM dan processor dalam posisi idle, dan hanya penggunakan source sebesar 5% saja.

4.4 Uji Jalur Komunikasi

Dari hasil uji rendering, dilakukan *monitoring* pada jalur komunikasi, dan didapatkan hasil uji sebagai berikut:

1. Hasil uji menyatakan bahwa ketika rendering mulai dijalankan, ada komunikasi dua arah dari pc master ke setiap node .
2. Ketika proses render selesai, dalam posisi idle, dan tidak ada komunikasi.

4.5 Analisis *Speed Up*

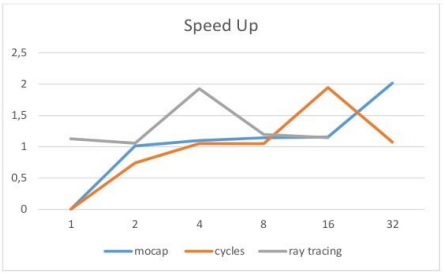
Dihitung *Speedup*, dan *Performance Improvement* dari penggunaan node. Rata-rata hasil dari pengujian terdapat pada tabel berikut (dalam menit) :

No	Metode	1 Node	2 Node	4 Node	8 Node	16 Node	32 Node
1	Ray Tracing	0	1,126 42569	1,055 31028	1,926 14178	1,1962 6414	1,145 79984
2	Cycle	0	1,043 20767	1,051 6502	1,048 69854	1,9438 0639	1,074 1394
3	Motion Capture	0	1,013 12287	1,100 20952	1,141 77171	1,1560 0078	2,015 70244

Tabel 4-5 Analisis *Speed Up*

Untuk *speedup*, dapat dihitung dengan membagi antara waktu eksekusi node sebelumnya dan node sekarang, dan hasilnya akan didapat kecepatan dari kedua waktu tersebut.

Berikut grafik speed up yang didapat dari data:



Gambar 4-5 Analisis *Speed Up*

Dari grafik dapat terlihat bahwa speed up terbaik pada ray tracing rendering didapat pada kenaikan 4 node menjadi 8 node, yaitu 93%, speed up cycles rendering terbaik terdapat pada peningkatan 8 node menjadi 16 node yaitu sebesar 91%, dan speed up motion capture terbaik ada pada peningkatan 16 node menjadi 32 node, yaitu sebesar 95%. sinkronisasi seberapa banyak pc yang digunakan, disesuaikan dengan beban kerja, dalam hal ini, untuk kasus rendering, motion capture memiliki beban kerja besar, maka penggunaan node sebanyak 32 menjadi lebih efisien . sedangkan untuk beban kerja sedikit seperti ray tracing, cukup digunakan 8 node saja.

4.6 Analisis *Performance Improvement*

Performance Improvement digunakan untuk menghitung hasil performansi antara node sebelumnya dan node sekarang, hasilnya terdapat dalam tabel berikut:

Tabel 4-6 Performance Improvement

Pada tabel terlihat kenaikan performansi setiap metode berbeda nilainya. Nilai ini yang menjadi rujukan

node	ray tracing	cycle	motion cap
1	0	0	0
2	0,112236159	0,062709313	0,012952888
4	0,052411393	0,049113477	0,091082217
8	**0,480827418**	0,04643712	0,124168179
16	0,164064218	**0,485545471**	0,13494868
32	0,127247216	0,06902214	**0,453804956**

berapa jumlah node yang dibutuhkan untuk optimasi *low cost*.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan uji dan analisis yang telah dilakukan, penelitian tugas akhir ini dapat disimpulkan seperti berikut:

1. Dari grafik dapat terlihat bahwa speed up terbaik pada ray tracing rendering didapat pada kenaikan 4 node menjadi 8 node, yaitu 93%, speed up cycles rendering terbaik terdapat pada peningkatan 8 node menjadi 16 node yaitu sebesar 91%, dan speed up motion capture terbaik ada pada peningkatan 16 node menjadi 32 node, yaitu sebesar 95%. sinkronisasi seberapa banyak pc yang digunakan, disesuaikan dengan beban kerja, dalam hal ini, untuk kasus rendering, motion capture memiliki beban kerja besar, maka penggunaan node sebanyak 32 menjadi lebih efisien . sedangkan untuk beban kerja sedikit seperti ray tracing, cukup digunakan 8 node saja.
2. Pembangunan cluster HPC, dari hasil implementasi sistem, mempunyai tiga aspek signifikan yang mempengaruhi cost, yaitu kemampuan processor, besar memory, dan komunikasi antar node. Dalam kasus ini, yaitu rendering, aspek optimal yang bisa digunakan sudah diuji coba, semakin banyak processor, semakin cepat hasil eksekusi. Namun, Speed up yang terjadi, tidak linear dengan penambahan dua kali lipat processor. Maka dari itu penambahan node yang disesuaikan dengan metode yang digunakan, akan sangat mempengaruhi *cost* yang dikeluarkan.
3. Dari hasil uji dan analisis, jalur komunikasi yang digunakan tidak dapat dioptimalkan lagi dari segi arsitekturnya untuk mengrangi cost, karena cluster ditempatkan di satu workstation. Jalur komunikasi dapat di optimalkan lagi hanya jika mengganti jenis komponen penghubung.
4. Otomatisasi jaringan menggunakan script, sesuai hasil analisis cost, dapat mengurangi biaya daya listrik yang digunakan.

5.2 Saran

1. Perlu dilakukan lebih banyak skenario dalam uji coba, agar didapat hasil yang lebih baik.
2. Perlu dilakukan uji coba menggunakan jaringan grid, untuk pengoptimalan source komputer di lab.
3. Ide untuk menyalurkan panas/ over heating pada komputer cluster, menjadi alternatif penghangat ruangan atau pemanas air bisa menjadi penelitian lanjutan.

Daftar Pustaka:

(1) klustering Tutorial. Diakses 15 Oktober 2014 dari scfbio-ittd. http://www.scfbio-iitd.res.in/doc/klustering.pdf

(2) Designing a kluster Computer. Diakses 20 Oktober 2014 , dari ameslab http://www.scl.ameslab.gov/Projects/parallel_computing/kluster_design.html

(3) Bailey, D; Barton, J; Lasinski, T; & Simon, H. January (1991). The NAS Parallel Benchmarks, ReportRNR-91-002, NASA/Ames Research Center.

(4) Barney, B. (2009). "Introduction to Parallel Computing" Diakses tanggal 27 Oktober 2014 , dari https://computing.llnl.gov/tutorials/parallel_comp/

(5) Buyya, R (ed.). (1999). High Performance kluster Computing, Volume 2: Programming an *Applications, New Jersey : Prentice Hall, Inc.*

(6) Sumaryono, Sujoko Perbandingan OpenMPI dan LAM/MPI sebagai Middleware *jaringan kluster head node.pdf diakses 25 Oktober 2014 dari lib.ugm.ac.id*

(7) *Ajinogoro,B.I.(2005).Aplikasi Sistem Paralel Menggunakan Prosesor Host 486*

Berbasis Linux Debian. Bandung: STT Telkom.

(8) *Andrian, H. R. (2008). Sistem Komputer. Bandung: Politeknik Telkom.*

(9) *Desrochers, George R. (1988). Principlesof Parallel and Multiprocessing. New York: McGraw-Hill.*

(10) *Lukmanul, H. (2007, Agustus 08). Desain dan Implementasi High Throughput Computing Environment Menggunakan Condor. di Institut Teknologi Sepuluh November. Surabaya.*