

ANALISIS SISTEM KOMPUTASI PARALEL PADA INFRASTRUKTUR *GRID COMPUTING*

Irawan, Irmawati¹⁾, Syarif Al Qadri Syahrir²⁾.

Abstrak: *Grid Computing* merupakan salah satu sistem komputasi paralel terdistribusi dengan menggunakan banyak sumber daya komputasi yang dikelola secara bersama dan terpisah secara geografis. *Grid computing* saat ini menjadi suatu bentuk solusi untuk melakukan komputasi dalam skala besar. Penelitian ini menggunakan beberapa komputer yang dibagi menjadi dua klaster yang terpisah. Menggunakan pustaka OpenMPI untuk lingkungan komputasi paralel dan Sun Grid Scheduler sebagai *gridengine*, Globus Toolkit sebagai *middleware*, Gridsphere dan Vine Toolkit sebagai portal grid. Penelitian ini dapat digunakan sebagai panduan untuk membangun sistem komputasi paralel pada infrastruktur *grid computing* dan mengetahui cara kerja sistem *grid computing* untuk dapat dilakukan pengembangan sistem untuk mendukung proses pembelajaran.

Kata Kunci: grid computing, parallel computing, globus toolkit, vine toolkit, opengrid scheduler

PENDAHULUAN

Pesatnya kebutuhan data dan informasi saat ini menyebabkan para ilmuwan dan peneliti baik di lembaga penelitian, pendidikan maupun perusahaan memasuki suatu masa di mana kebutuhan akan tenaga komputasi dan kapasitas penyimpanan data akan jauh melebihi dari kapasitas yang tersedia. Meningkatnya kesadaran peneliti akan manfaat komputer dalam penelitian, menyebabkan semakin banyaknya usaha yang dilakukan untuk menjadikan komputer sebagai alat bantu diberbagai bidang yang mampu memenuhi kebutuhan komputasi skala besar dalam hal pemrosesan data.[1]

Cluster Computer atau *Grid Computing* menjadi solusi terbaik yang banyak diminati karena dapat dikembangkan menggunakan sumber daya yang sudah ada dari berbagai institusi sehingga terbentuk lingkaran komputasi yang sangat besar dan terpisah secara geografis.[2]*Grid computing* adalah teknologi komputasi terdistribusi yang memanfaatkan sumber daya yang terhubung melalui jaringan komputer secara bebas tapi terkoordinasi dengan mekanisme tertentu (Widodo Budiharto., 2012)[1].

Implementasi grid computing telah dilakukan oleh sebagian besar negara di dunia, sebagai contoh negara Hongkong yang telah mengimplementasikan GRID dengan tujuan sebagai R&D Grid, yakni grid bagi institusi pemerintahan, dan industri serta grid untuk

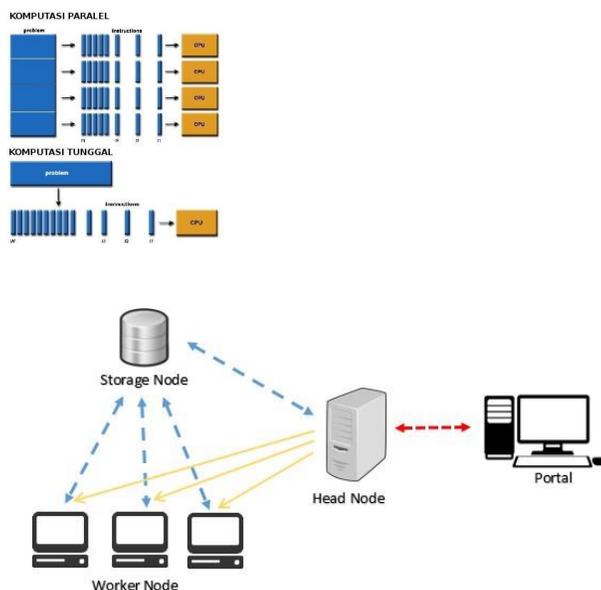
berhubungan dengan partner negara lain di lingkungan Cina dan Asia-Pasific. Di Indonesia sendiri sudah memulai riset mengenai grid computing pada tahun 2006 melalui usulan infrastruktur RI-GRID, yaitu infrastruktur komputasi grid skala nasional yang diberi nama InGrid (Inherent Grid).[2]

Penelitian ini bertujuan untuk melakukan studi tentang cara membangun infrastruktur grid computing multi klaster dalam lingkungan heterogen menggunakan metode komputasi paralel. Dikarenakan penelitian terkait grid computing masih sangat kurang dan informasi mengenai tahapan pengembangan infrastruktur belum memadai sehingga dianggap perlu untuk membuat sebuah panduan dasar pengembangan infrastruktur

grid computing. Selama proses instalasi juga dilakukan evaluasi terhadap infrastruktur tersebut sebagai media pembelajaran untuk mengetahui mekanisme kerja sistem agar dapat dilakukan penelitian lebih lanjut untuk pengembangan dan metode pemanfaatannya.

A. Komputasi Paralel

Komputasi paralel merupakan teknik komputasi secara bersamaan dengan memanfaatkan beberapa komputer secara simultan. Biasanya diperlukan saat kapasitas yang diperlukan sangat besar, baik karena harus mengolah data dalam jumlah besar ataupun karena tuntutan proses komputasi yang banyak. Dalam model komputasi ini, suatu masalah yang kompleks dibagi-bagi menjadi beberapa bagian yang kecil dan dijalankan secara paralel.[2]

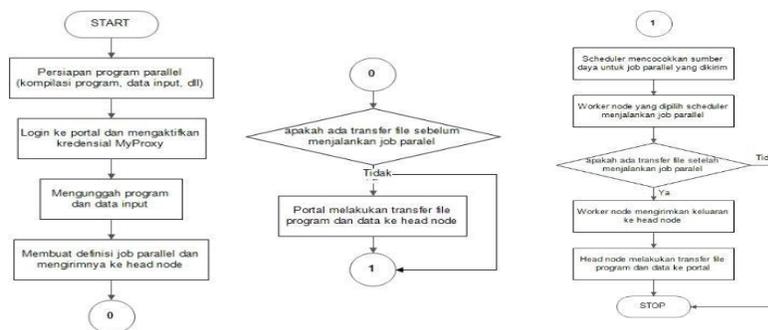


Gambar 1. Perbedaan antara Komputasi Paralel dan Komputasi Tunggal[2]

B. Parallel Job

Parallel job merupakan metode komputasi yang digunakan dalam sistem komputasi paralel dengan memanfaatkan lebih dari satu inti CPU atau slot untuk melakukan pekerjaan komputasi yang sudah ditentukan sebelumnya secara bersamaan. Metode ini dilakukan dengan menerima program paralel, data masukan, dan file definisi

parallel job yang dikirimkan oleh pengguna melalui web portal sesuai dengan format yang ditentukan oleh GRAM. Head node merespon dengan memanggil scheduler agar memilih worker node yang akan menjalankan parallel job sehingga menghasilkan file keluaran (standard output dan standard error) yang selanjutnya dikirimkan ke portal untuk didownload oleh pengguna yang menjalankan parallel job.



Gambar 2.Flowchart Job Paralel

C. Grid Computing

Grid Computing atau komputasi grid adalah penggunaan sumber daya yang melibatkan banyak komputer yang terpisah secara geografis untuk memecahkan persoalan komputasi dalam skala besar. Melalui Grid Computing kita dapat menyatukan dan mengkoneksikan sejumlah komponen perangkat keras (hardware) dan Dalam implementasinya, resources yang digunakan di dalam sebuah sistem Grid

tidaklah sedikit dan sifatnya pun heterogen.[3] Karena itu, dibutuhkan perangkat lunak (software) yang modular dan independen. Inti dari grid computing adalah untuk mencapai performa dan hasil yang lebih baik dengan cara memanfaatkan sumber daya komputasi dari berbagai organisasi yang berbeda secara bersama- sama.[8]

interface dan protokol standar yang bersifat

terbuka. Dengan cara inilah, resources yang ada tersebut dapat saling berkolaborasi untuk menyelesaikan proses komputasi tertentu.

D. Arsitektur Sistem

Konsep dasar dalam pengembangan infrastruktur grid yakni keterlibatan banyak komputer dalam melayani job dan setiap komputer tersebut memiliki peran masing- masing. Seperti yang dijelaskan pada gambar dibawah ini.

Panah warna kuning menjelaskan komunikasi antara *head node* dengan *worker node* dalam mendistribusikan dan menjalankan *parallel job*. Panah dua arah warna biru menunjukkan akses bagi pakai data dan program paralel. *Head node* menyimpan *file* data dan program paralel yang dikirim dari portal pada NFS *mount* di *storage node*, dan *file* tersebut diakses oleh *worker node*. Panah dua arah warna merah menunjukkan komunikasi antara *head node* dengan portal melalui *middleware*.

Dalam infrastruktur *grid computing* arsitektur sistem dibagi menjadi beberapa komponen antara lain :

1. Head Node

Komputer yang berperan sebagai *head node*

memilih dan menjadwalkan setiap *parallel job* yang dikirimkan oleh portal. *Head node* dan portal berkomunikasi menggunakan *middleware* grid. Dalam tugas akhir ini, penulis menggunakan Globus Toolkit 4 dan Grid Engine yang disediakan dalam package debian.

2. Storage Node

Komputer storage node digunakan untuk menyimpan data yang dibutuhkan dalam menjalankan *parallel job*. Komputer ini akan menjalankan layanan bagi pakai file seperti Network File System (NFS) yang akan digunakan oleh Scheduler agar dapat berbagi pakai file dengan worker node sehingga worker node dapat menjalankan *parallel job*.

3. Worker Node

Komputer worker node adalah bagian terpenting dari sebuah infrastruktur grid. Komputer ini berkomunikasi dengan scheduler

4. Portal

Komputer yang berperan sebagai portal akan berkomunikasi dengan *scheduler* pada setiap klaster untuk mengirimkan *parallel job*.

agar dapat menjalankan parallel job, dan mengakses storage node melalui NFS mount untuk menjalankan executable dan data yang dibutuhkan parallel job tersebut.

E. Globus Toolkit

Open source Globus Toolkit ®

secara aman melintasi batas-batas perusahaan, institusi, dan geografis tanpa mengorbankan otonomi daerah. *Toolkit* ini meliputi *software services* dan *libraries* untuk pemantauan sumber daya, *discovery*, dan manajemen ditambah sistem keamanan dan manajemen *file*. [4]

scheduler yang dikembangkan oleh Sun Microsystem. Perangkat lunak ini berfungsi untuk mengatur eksekusi *parallel job*, mulai dari antrian *parallel job*, komunikasi dengan pustaka MPI yang disebut sebagai *parallel environment*, dan mengatur alokasi *worker node* untuk setiap *parallel job* yang sedang dalam antrian. Pengguna dapat mengirim banyak *parallel job* sekaligus tanpa harus mengetahui dimana, kapan, dan bagaimana *parallel job* itu dijalankan. Perangkat lunak ini mengumpulkan informasi mengenai sumber daya yang tersedia berdasarkan aturan yang dapat ditetapkan oleh administrator.

dalah teknologi yang memungkinkan untuk membangun “*grid*”, yang dapat membantu orang-orang untuk membagi sumber daya komputasi, database, dan alat-alat lainnya

F. OpenGrid Scheduler/Grid Engine

OpenGrid Scheduler/Grid Engine yang dikenal sebagai Sun Grid Engine adalah perangkat lunak

Informasi ini nantinya akan digunakan oleh Grid Engine untuk pengalokasian sumber daya kepada *parallel job* yang dikirimkan oleh pengguna. [6]

G. Message Passing Interface (MPI)

Sebuah program paralel yang menggunakan MPI akan menjalankan satu atau lebih proses pada setiap komputer. Proses tersebut

akan bekerja pada data yang ada di komputer tersebut. Suatu variable yang dibuat pada masing-masing proses pada setiap komputer tidak dapat mengakses variable yang dibuat pada komputer lain.

Untuk dapat membagi pakai data, dalam hal ini adalah nilai dari suatu variable local, diperlukan mekanisme komunikasi, yaitu menggunakan *message passing*. Mekanisme ini dilakukan dengan mengirim dan menerima pesan yang dilakukan antar computer yang menjalankan program paralel.[7]

H. Gridsphere

Gridsphere adalah perangkat lunak portal web

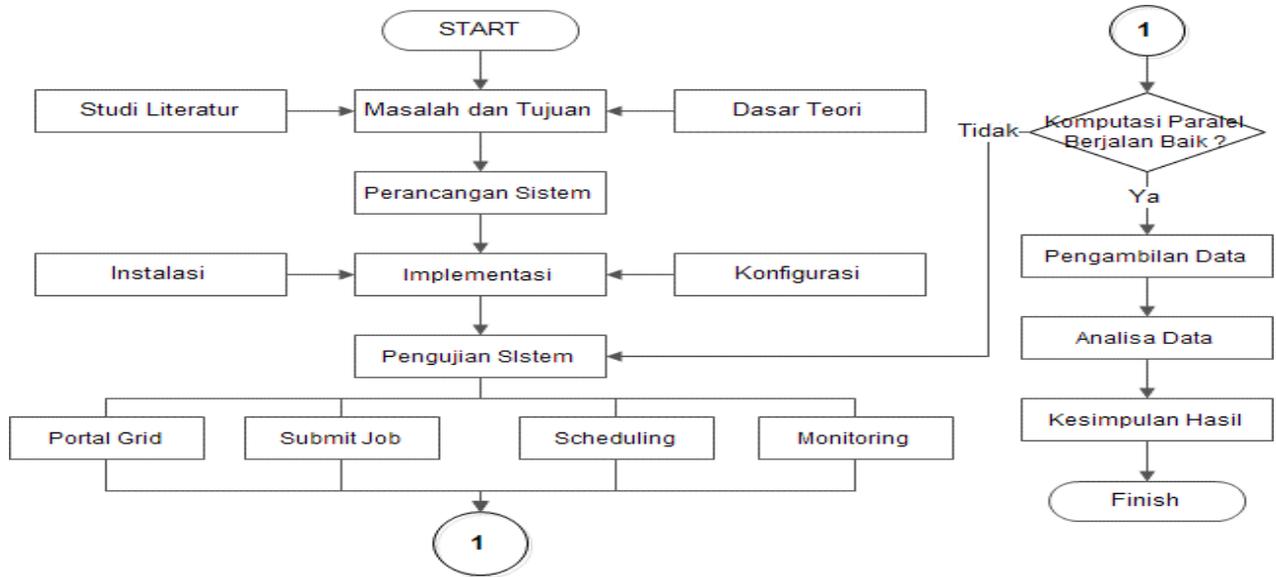
berbasis java yang paling banyak digunakan, dikembangkan oleh mahasiswa- mahasiswa di sebuah institut di German. Aplikasi ini digunakan untuk mengintegrasikan informasi dan aplikasi, serta merepresentasikannya kepada pengguna. Dalam kasus kali ini adalah informasi dan aplikasi yang disediakan oleh grid. Selain istilah portal, dikenal juga istilah portlet yang merupakan komponen web berbasis java yang bersifat portabel dan dapat menghasilkan konten web yang bersifat dinamis.[5]

METODE PENELITIAN

I. Tahapan Penelitian

Penelitian ini akan menggunakan metode analisa sistem dengan membangun infrastruktur *grid computing* untuk diuji coba menggunakan beberapa skenario yang menghasilkan data informasi untuk dapat

diambil keputusan yang digunakan dalam membuat dokumentasi penelitian. Fokus penelitian ini pada perancangan infrastruktur dan menganalisa bagaimana sistem komputasi paralel pada infrastruktur *grid computing* berjalan.

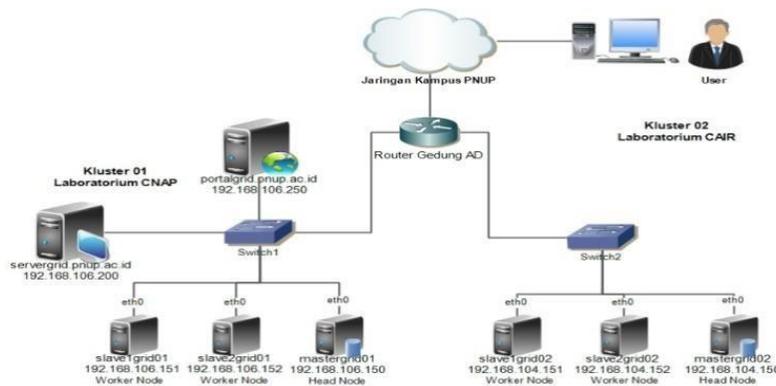


Gambar 3. Flowchart Proses Penelitian

J. Desain Sistem

Pada desain topologi jaringan yang dirancang ini dibangun dua buah klustersistem komputasi yang ditempatkan di dua lokasi yang berbeda yang dihubungkan

menghubungkan satu buah router pada jaringan kampus Politeknik Negeri Ujung Pandang. Rancangan desain topologi dari sistem yang akan digunakan pada penelitian ini dapat dilihat pada Gambar di bawah ini.



Gambar 4. Rancangan Infrastruktur *Grid Computing*

Berikut ini adalah tabel perencanaan pengalaman IP dan penamaan server yang

akan dikonfigurasi pada layanan DNS server.

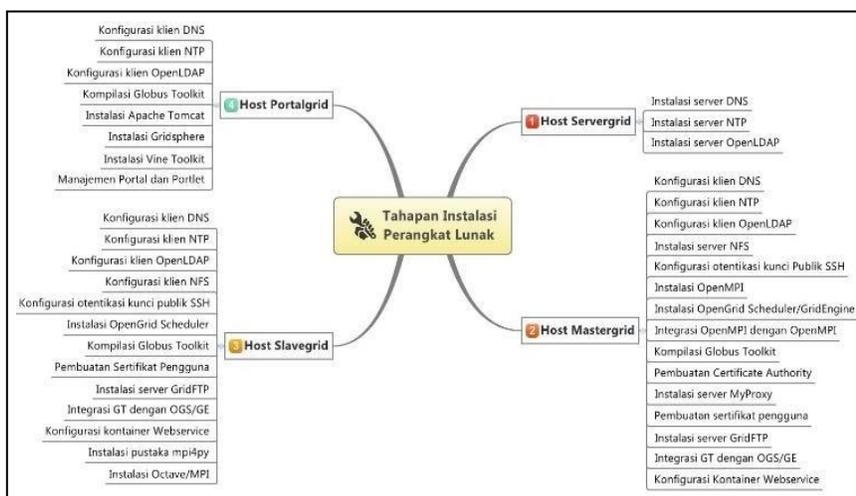
Tabel 1. Perencanaan Hostname dan IP Address Perangkat

Hostname	IP Address	Perangkat
mastergrid01.pnup.ac.id	192.168.106.150	Master Grid Klaster 01
slave1grid01.pnup.ac.id	192.168.106.151	Slave1 Grid Klaster 01
slave2grid01.pnup.ac.id	192.168.106.152	Slave2 Grid Klaster 01
mastergrid02.pnup.ac.id	192.168.104.150	Master Grid Klaster 02
slave1grid02.pnup.ac.id	192.168.104.151	Slave1 Grid Klaster 02
slave2grid02.pnup.ac.id	192.168.104.152	Slave2 Grid Klaster 02
portalgrid.pnup.ac.id	192.168.106.250	Server Portal Grid
servergrid.pnup.ac.id	192.168.106.200	Server VM Proxmox

K. Tahapan Instalasi Perangkat Lunak

Proses instalasi dan konfigurasi perangkat lunak pada setiap komponen *grid computing* perlu dilakukan secara bertahap karena masing-masing perangkat lunak tersebut saling terhubung dan

terintegrasi satu sama lain sehingga keberhasilan proses instalasi sebelumnya akan berpengaruh terhadap keberhasilan proses instalasi perangkat lunak selanjutnya.



Gambar 5. Tahapan Instalasi Perangkat Lunak

Tahapan ini dimulai dengan melakukan proses instalasi dan konfigurasi pada klaster pertama hingga berlanjut untuk klaster kedua. Pengerjaan klaster dimulai dari *head node* terlebih dahulu dalam hal ini *host mastergrid* karena konfigurasi perangkat lunak yang dilakukan pada *worker node* akan menggunakan hasil konfigurasi pada *head node*. Tahapan akhir dilakukan manajemen server portal untuk menghubungkan dua

buah klaster yang terdiri dari *head node* dengan beberapa *worker node* yang sudah dibuat sebelumnya.

L. Kebutuhan Hardware dan Software

Pada penelitian ini digunakan beberapa buah *computer node* sebagai *slave* dan *master* untuk masing-masing klaster. Sebagai gambaran mengenai spesifikasi perangkat komputer dan perangkat jaringan yang akan digunakan

ditampilkan pada Tabel.

Tabel 2. Spesifikasi Kebutuhan Hardware

Klaster I					
Hostname	mastergrid01		slave1grid01		slave2grid01
IP	192.168.106.150		192.168.106.151		192.168.106.152
Address	192.168.106.150		192.168.106.151		192.168.106.152
CPU	Intel(R) Core(TM)2 Duo E7600 @3.06GHz		Intel(R) Core(TM)2 Duo E7600 @3.06GHz		Intel(R) Core(TM)2 Duo E7600 @3.06GHz
RAM	2 GB		2 GB		2 GB
Hardisk	320 GB		320 GB		320 GB
NIC	10/100Mbps		10/100Mbps		10/100Mbps
Klaster II					
Hostname	mastergrid02		slave1grid02		slave2grid02
IP	192.168.104.150		192.168.104.151		192.168.104.152
Address	192.168.104.150		192.168.104.151		192.168.104.152
CPU	Intel(R) Core(TM)2 Duo E4500 @2.20GHz		Intel(R) Core(TM)2 Duo E4500 @2.20GHz		Intel(R) Core(TM)2 Duo E4500 @2.20GHz
RAM	1 GB		512 MB		1 GB
Hardisk	320 GB		320 GB		320 GB
NIC	10/100Mbps		10/100Mbps		10/100Mbps

Beberapa aplikasi yang digunakan untuk mendukung operasional kerja sistem komputasi paralel pada infrastruktur *grid computing* adalah sebagai berikut.

- **Sistem Single Sign-On :** OpenLDAP
- **Sistem Hypervisor :** Proxmox
- **Sistem Klaster/Paralelisasi :** OpenMPI, MPICH2.
- **Sistem Grid :** Sun Grid Engine, Globus Toolkit
- **Sistem Akses Portal :** Vine Toolkit, Gridsphere.

- **Sistem Monitoring** :Globus Toolkit.

- **Perangkat Lunak Pendukung** : OpenJDK, Apache ANT, Bind, NFS menggunakan bahasa C untuk mencetak identitas proses yang dilakukan.

HASIL DAN PEMBAHASAN

Pada skenario ini dilakukan proses eksekusi program paralel

```

syarif@mastergrid02:~$ mpicc program.c -o program
syarif@mastergrid02:~$ mpirun -machinefile /datagrid/machinefile.txt -np 3 program
Hello world from processor mastergrid02, rank 0 out of 3 processors
Hello world from processor slave1grid02, rank 1 out of 3 processors
Hello world from processor slave2grid02, rank 2 out of 3 processors
syarif@mastergrid02:~$
    
```

Gambar 6. Hasil Eksekusi Program Hello World

Pengujian selanjutnya dilakukan dengan menggunakan matriks menggunakan ordo lebih banyak yakni 5000x4000 dikali 4000x3000 sehingga menghasilkan matriks 5000x3000. Perkalian

matriks ini dilakukan dengan 3 tahapan yang terdiri dari perkalian matriks pertama, perkalian matriks kedua, dan perkalian matriks hasil. Proses tersebut ditampilkan pada gambar dibawah ini

```

syarif@mastergrid01:~$ mpicc mat5000.c -o mat5000
syarif@mastergrid01:~$ mpirun --machinefile /datagrid/machinefile.txt -np 3 mat5000
.....
5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0
5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0
5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0
5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0
5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0 5000.0
5000.0 5000.0 5000.0
Time taken = 270.14223 seconds
syarif@mastergrid01:~$
    
```

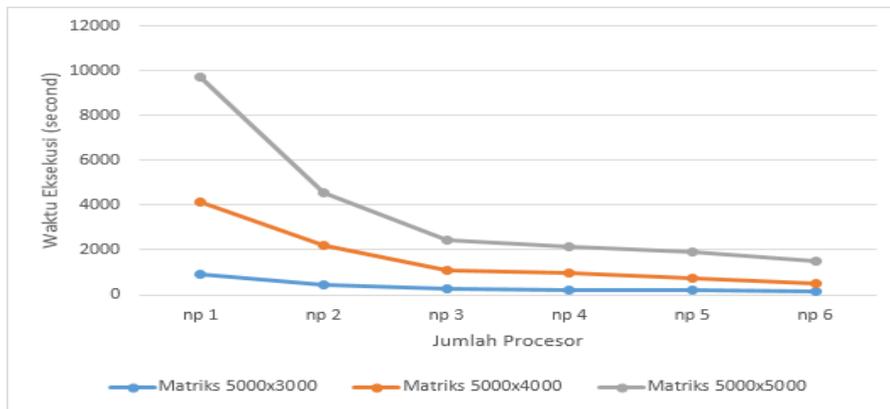
Gambar 7. Hasil Eksekusi Program Perkalian Matriks 5000x3000

Proses tersebut hanya dilakukan percobaan eksekusi dengan menggunakan satu jenis perkalian matriks. Percobaan selanjutnya dilakukan percobaan eksekusi program paralel dengan 3

macam jumlah perkalian matriks dengan 6 kali percobaan jumlah prosesor yang digunakan. Hasil percobaan tersebut ditampilkan dalam bentuk tabel dan grafik seperti gambar dibawah ini.

Tabel 3. Hasil Eksekusi Program Paralel Perkalian Matriks

Ukuran Matriks	Waktu Eksekusi (Detik)					
	1 NP	2 NP	3 NP	4 NP	5 NP	6 NP
5000x3000	912	433	270	210	180	166
5000x4000	4127	2213	1080	976	713	496
5000x5000	9731	4539	2410	2133	1887	1391



Gambar 8. Grafik Hasil Eksekusi Program Paralel Perkalian Matriks

Dengan demikian jenis program paralel, jumlah proses yang harus dikerjakan pada program paralel akan mempengaruhi waktu

yang dibutuhkan untuk melakukan

eksekusi program paralel.
Selain itu jumlah processor
atau komputer akan
mempengaruhi waktu proses
sebuah program paralel.

PENUTUP

Kesimpulan

1. Infrastruktur *grid computing* dengan metode komputasi paralel dapat dibangun dari beberapa klaster yang terpisah secara geografis yang terdiri dari satu *master* dan beberapa *slave* kemudian dihubungkan menggunakan jaringan internet dengan memanfaatkan fungsi FQDN (*Fully Qualified Domain Name*).
2. Infrastruktur *grid* dibangun menggunakan beberapa perangkat lunak komputasi dan perangkat lunak pendukung yang diinstal secara bertahap dan saling terintegrasi satu sama lain.
3. Cara kerja sistem komputasi paralel pada infrastruktur *grid computing* dilakukan lebih mudah dengan adanya portal web. Pengguna melakukan *submit job* kemudian sistem akan mendistribusikan job ke beberapa *worker node* berdasarkan program yang digunakan hingga status eksekusi job dapat ditampilkan pada portal.
4. Pembangunan infrastruktur *grid computing* belum memiliki standar

baku dalam hal penggunaan perangkat lunak, masing-masing instansi pengembang memiliki standar untuk digunakan pada lingkungan sendiri.

5. Infrastruktur *grid computing* pada dasarnya menitik beratkan proses paralel pada program yang dibuat. Infrastruktur digunakan untuk melakukan manajemen dan eksekusi file program paralel. Aplikasi penggunaan infrastruktur *grid* untuk kebutuhan lain perlu menggunakan perangkat lunak tambahan yang bisa diintegrasikan dengan infrastruktur yang sudah ada.

Saran

1. Dalam mengembangkan infrastruktur *grid computing* sebaiknya dilakukan lebih dari satu orang agar diperoleh hasil penelitian yang jauh lebih baik dan terkordinasi.
2. Penelitian yang dilakukan selanjutnya diharapkan dapat mengarah pada pengaplikasian infrastruktur *grid computing* untuk kebutuhan belajar mengajar, seperti pembuatan sistem komputasi paralel untuk proses rendering atau

integrasi dengan sistem pembelajaran online.

3. Perlu untuk menerapkan mata kuliah pemrograman paralel pada perkuliahan agar sistem ini dapat dikembangkan lebih lanjut.

DAFTAR PUSTAKA

Anindito, Benediktus. 2013. Implementasi Portal Komputasi Grid untuk Multikluster. Surabaya: ITS Library.

Anonim. 2005. *GT 4.2 Quickstart Admin Guide*.

Budiharto, Widodo. 2012. Implementasi dan Evaluasi Penerapan Globus Toolkit untuk Aplikasi Grid Computing. Jurnal Computer Science Binus University, Vol 3 no 1. Jakarta: Comtech.

Ferreira, Luis, dkk. (2003, October 06). *Introduction to Grid Computing with Globus*. diakses pada 30 Januari 2015, <http://www.redbooks.ibm.com>
<http://toolkit.globus.org/toolkit/docs/4.2/4.2.1/admin/quickstart>. Diakses pada tanggal 10 Juli 2015.

Lai, Chuan-Lin., Yung, Chao-Tung (July, 2003). *Construct a Grid Computing Environment on Multiple Linux PC Cluster*. Tunghai Science vol. 5 : 107-124. diakses 3 Februari 2015.
<http://thuir.thu.edu.tw>

London e-Science Centre. 2005. Sun Grid Engine Integration with Globus Toolkit 4.
<http://www3.imperial.ac.uk/lesc/projects/archived/sungrid/gtb>. Diakses pada tanggal 15 Agustus 2015.

Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard*.
<http://www.mpi-forum.org/docs/mpi-11-html/mpi-report.html>. Diakses pada tanggal 27 Agustus 2015.

Rumagit, Arthur, (2013). *Implementasi Grid Computing untuk High Throughput Computing*. Jurnal Teknik Informastika Universitas Sam Ratulangi, vol 2 no 1. diakses 2 Februari 2015.
<http://www.portalgaruda.org/>