

21898/4108



RSIF  
004.6  
Tuh  
1-1  
2008

TUGAS AKHIR - CI 1599

**ITS GRID**

DWIGHT RONNARD FITZGERALD TUHUSULA  
NRP 5101 109 005

Dosen Pembimbing  
Ir. F.X. Arunanto, M.Sc

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2008

<b>PERPUSTAKAAN ITS</b>	
<b>Tgl. Terima</b>	13-2-2008
<b>Terima Dari</b>	H
<b>No. Agenda Prp.</b>	230027



**FINAL PROJECT - CI 1599**

## **ITS GRID**

**DWIGHT RONNALD FITZGERALD TUHUSULA**  
NRP 5101 109 005

Advisor  
Ir. F.X. Arunanto, M.Sc.

DEPARTMENT OF INFORMATICS  
Faculty of Information Technology  
Institute of Technology Sepuluh Nopember  
Surabaya 2008

## ITS GRID

### TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer  
pada  
Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**DWIGHT RONNALD FITZGERALD TUHUSULA**  
NRP : 5101 109 005

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Ir. F. X. Arunanto, M.Sc.  
NIP: 131285253



**SURABAYA**  
**JANUARI 2008**

## ITS GRID

Nama Mahasiswa : Dwight Ronald Fitzgerald Tuhusula  
NRP : 5101 109 005  
Jurusan : Teknik Informatika FTIf-ITS  
Dosen Pembimbing : Ir F. X Arunanto., M.Sc.

### Abstrak

*Grid computing merupakan teknologi system terdistribusi yang flexible dan menawarkan sejumlah potensi dan keuntungan termasuk high performance computing, data federation dan kolaborasi, resource allocation dan optimisasi, dan banyak lainnya. Ethernet yang merupakan salah satu faktor pendukung terus berubah sebagai teknologi cost-efektif yang digunakan untuk membuat cluster computer, grid network kampus, grid cluster dan extended multi-side grid.*

*Didalam lingkungan grid, situasinya bertambah komplek. perbedaan antara client dan server semakin hilang, karena individual resource bertindak sebagai server di satu saat (karena menerima request) dan sebagai client di lain waktu (karena meminta request ke resource yg lain). Contohnya ketika kita meminta kode simulasi dijalankan di komputer lainnya, kita adalah client dan komputer lainnya adalah server. tapi di lain waktu, kode dan komputer yang sama bertindak sebagai client, ketika mereka meminta request atas permintaan kita ke komputer lain untuk mengakses input dataset dan untuk menjalankan komputasi tambahan.*

*Di masa depan ketika komputasi, penyimpanan data dan perangkat lunak tidak lagi berupa objek yang kita miliki, tetapi merupakan utility yang kita masukkan, maka semakin sukses komunitas peneliti dan ilmu pengetahuan dimana mereka yang berhasil mengefektifkan infrastruktur grid yang cocok sehingga pada akhirnya dapat menggabungkan metode pengembangan dan mengadopsi metode penyelesaian masalah yang baru ke dalam ilmu mereka.*

**Kata kunci : Grid, Grid Computing**

## ITS Grid

Name : Dwight Ronald Fitzgerald Tuhusula  
NRP : 5101 109 005  
Department : Teknik Informatika FTIf-ITS  
Lecturers : Ir F. X Arunanto., M.Sc.

### Abstract

*Grid computing is evolving as a very flexible distributed system technology and offers a number of potential uses and benefits including high performance computing, data federation and collaboration, resource allocation and optimization, and more. Ethernet continues to evolve as a cost-effective technology with which to build computer clusters, campus Grid networks, Grids-of-clusters, and extended multi-site Grids*

*In Grid environments, the situation is more complex. The distinction between client and server tends to disappear, because an individual resource can act as a server one moment (as it receives a request) and as a client at another (as it issues requests to other resources). For example, when I request that a simulation code be run on a colleague's computer, I am the client and the computer is a server. But a few moments later, that same code and computer act as a client, as they issue requests--on my behalf--to other computers to access input datasets and to run subsidiary computations.*

*In a future in which computing, storage, and software are no longer objects that we possess, but utilities to which we subscribe, the most successful scientific communities are likely to be those that succeed in assembling and making effective use of appropriate Grid infrastructures and thus accelerating the development and adoption of new problem solving-methods within their discipline.*

**Keywords : Grid, Grid Computing**

## KATA PENGANTAR

Penulis bersyukur kepada Tuhan Yesus. Seperti ada tertulis dalam “(Matius 7:7) *Mintalah maka akan diberikan kepadamu, carilah maka engkau akan mendapat, ketoklah maka pintu akan dibukakan bagimu.*”

Karena kasih dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul :

### **“ITS GRID”**

Tidak lupa penulis sampaikan ucapan terima kasih kepada pihak-pihak yang telah membantu penyelesaian Tugas Akhir ini :

- Bapak Ir. F. X. ARUNANTO, M.Sc selaku pembimbing penulis, yang dengan sabar membimbing dan memberikan pengarahan – pengarahan kepada penulis.
- Bapak Bobby Nazief, Phd yang telah mengizinkan penulis untuk melakukan riset Grid Computing pada jurusan Ilmu computer Universitas Indonesia.
- Rekan – rekan GCRG Ilkomp UI, terutama Fajran dan Fajar yang dengan sabar mengajari penulis mulai dari awal hingga akhir.
- Papa dan Mama yang dengan setia selalu mendoakan penulis mulai dari awal penulis berkuliah hingga detik – detik akhir penulis menyelesaikan sidang sarjana.
- Mami dan Oma yang senantiasa mendoakan penulis selama penulis kuliah hingga menyelesaikan kuliahnya.
- Mario dan Daniel kedua adik penulis yang juga senantiasa mendoakan penulis.
- Dani Wafa, yang rela membantu penulis disaat minggu terakhir penulis akan menghadapi ujian siding, jasanya tidak akan pernah penulis lupakan.
- Bapak Royana M I, S.Kom, M.Kom, yang telah membantu penulis memberikan masukan – masukan mengenai tugas akhir dan prosedurnya, serta ijin terhadap IP Public.

- Tandhy, yang selalu membantu penulis selama penulis mengerjakan tugas akhir (kutunggu sarjanamu).
- Angky yang juga menjadi motivator penulis yang selalu mengomentari penulis jika penulis malas. Thanks a lot for everythings hunbun.
- K' Jose, Bang Eli, Bang Ales, K' Rocky, Mas Aan, Afa, Alfred, Ino dan rekan – rekan pelayanan penulis di MDC Surabaya.
- Rekan – rekan tim Futsal Croet FC yang selalu memberikan dukungan pada penulis ketika IPL (Rossi, Firman, Ubed, Satria, Wajar, Iman, Dimas, Radit, Wibi).
- Rekan – rekan kuliah Annas, Robby, Daud, Khoirul Annas, Pii, Grawas, Aribi dan rekan – rekan lainnya.
- Buat Princea dan Irma (semangat buat ipul) yang selalu memberikan semangat juga, buat “Ganbatnya” Ayu (thanks for blueberry cheese cake-nya).
- Rekan seperjuangan TA : ody (trima kasih buat abstrak nya, kapan dy lulusnya?), aconk, priagusta, kholif, acil, adit, anggoro, hok cha, kiky, oyen, hendra, rino, pinky.
- Rekan – rekan IMC, semangat terus cari trik sulap baru
- Rekan – rekan GCL (Zul, Sule, Chika, Silas, Prasta (thanks for Laptop pas sidangnya), mieky+erny, Adit.
- Rekan-rekan TA-ers yang telah mendahului dan yang sedang mengerjakan : Galih, Leris, Connie dan semua rekan – reka yang telah sarjana.
- Semua pihak yang membantu penyelesaian Tugas Akhir ini yang penulis tidak mungkin dapat menyebutkan satu persatu.

Penulis mengharapkan adanya kritik dan saran yang membangun guna menambah manfaat serta mengurangi kesalahan dan kekurangan yang ada. Penulis berharap semoga laporan ini dapat memberikan manfaat.

Surabaya, February 2008

Penulis

## DAFTAR ISI

Abstrak.....	iii
Abstract.....	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Permasalahan.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	4
1.5 Metodologi.....	4
1.6 Sistematika Penulisan.....	5
BAB 2 TINJAUAN PUSTAKA.....	7
2.1 GRID History.....	7
2.1.1 Anatomy Grid.....	7
2.1.2 Komputasi Terdistribusi.....	8
2.1.3 Arsitektur Komputasi Terdistribusi.....	9
2.2 GRID.....	10
2.2.1 Arsitektur Grid.....	11
2.2.2 Layer Fabric.....	12
2.2.3 Layer Connectivity.....	13
2.2.4 Layer Resource.....	14
2.2.5 Layer Collective.....	15
2.2.6 Layer Aplikasi.....	16
2.3 Pelayanan GRID.....	16
2.3.1 Open Grid Services Architecture.....	17
2.3.2 Model Layanan OGSA.....	20
BAB 3 PERANCANGAN INFRASTRUKTUR DAN IMPLEMENTASI.....	25
3.1 ARSITEKTUR UMUM.....	25
3.1.1 Grid Clients.....	25
3.1.2 Grid Scheduler.....	25
3.1.3 Compute Nodes.....	26



3.1.4	Data Storage/ sources.....	26
3.2	ARSITEKTUR SISTEM.....	27
3.3	SPESIFIKASI SISTEM PERANTI KERAS DAN LUNAK.....	29
3.4	SKEMA.....	31
3.5	TOPOLOGI.....	32
3.6	KONFIGURASI CLUSTER DAN INSTALASI.....	34
3.6.1	Mesin Head Node.....	34
3.6.2	Mesin Data Storage.....	35
3.6.3	Mesin Head Node.....	36
3.6.4	Worker Node.....	37
3.6.5	Portal.....	37
BAB 4	UJI COBA DAN EVALUASI.....	39
4.1	Penggunaan Portal ITS Grid.....	39
4.2	Data Manager.....	40
4.3	Mengirim Job Generik.....	41
4.4	Melihat status job.....	43
4.5	Menu lainnya.....	44
4.5.1	Settings dan layout pada tab Welcome.....	44
4.5.2	Portlets, Users, Groups, Roles, Layout dan messaging pada tab Administration.....	45
4.5.3	Trusted Cluster list, Cluster access list, Application, Application pool, Pools admin, User admin, New release, email users pada tab Grid Admin.....	47
4.5.4	TeraGrid data manager, TeraGrid job manager, TeraGrid Application, TeraGrid Resources pada tab Other Grids.....	50
4.5.5	Development environtment, Qscript, Qhost, Qstat, Qsub, Qdel, SSH pada tab advanced.....	51
4.6	Evaluasi.....	53
BAB 5	KESIMPULAN, HAMBATAN DAN SARAN.....	55
5.1	KESIMPULAN.....	55
5.2	HAMBATAN.....	56
5.3	SARAN.....	56
LAMPIRAN	.....	59

DAFTAR PUSTAKA .....	78
BIODATA PENULIS .....	81

## DAFTAR GAMBAR

Gambar 2.1.1	Arsitektur umum Grid.....	25
Gambar 2.2.2	Arsitektur Sistem .....	27
Gambar 3.2.1	Cara kerja globus .....	28
Gambar 3.4	Skema ITS Grid.....	31
Gambar 3.5	Topologi.....	32
Gambar 4.1	Tampilan Portal ITS Grid .....	39
Gambar 4.2.2	Tampilan setelah login.....	40
Gambar 4.2	Tampilan Data Manager.....	40
Gambar 4.3	Tampilan Job Generik.....	41
Gambar 4.4	Tampilan tatus Job.....	43
Gambar 4.5.1	Tampilan Menu settings.....	44
Gambar 4.5.1.1	Tampilan layout.....	45
Gambar 4.5.2	Tampilan Menu portlets.....	45
Gambar 4.5.2.1	Tampilan Menu Users.....	46
Gambar 4.5.2.2	Tampilan Menu Groups.....	46
Gambar 4.5.2.3	Tampilan Menu Roles.....	46
Gambar 4.5.2.4	Tampilan Menu Layout.....	47
Gambar 4.5.2.5	Tampilan Menu Messaging.....	47
Gambar 4.5.3.1	Tampilan Menu Trusted cluster list.....	47
Gambar 4.5.3.2	Tampilan Menu Cluster access list.....	48
Gambar 4.5.3.3	Tampilan Menu Application.....	48
Gambar 4.5.3.4	Tampilan Menu Application pool.....	48
Gambar 4.5.3.5	Tampilan Menu Application Pools Admin.....	49
Gambar 4.5.3.6	Tampilan Menu User Admin.....	49
Gambar 4.5.3.7	Tampilan Menu Cluster News Release.....	49
Gambar 4.5.3.8	Tampilan Menu email users.....	50
Gambar 4.5.4.1	Tampilan Menu TeraGrid data manager.....	50
Gambar 4.5.4.2	Tampilan Menu TeraGrid job manager.....	50
Gambar 4.5.4.3	Tampilan Menu TeraGrid application.....	51
Gambar 4.5.4.4	Tampilan Menu TeraGrid Resources.....	51
Gambar 4.5.5.1	Tampilan Menu Development Environtment....	51
Gambar 4.5.5.2	Tampilan Menu Qscript.....	51

Gambar 4.5.5.3 Tampilan Menu Qhost.....	52
Gambar 4.5.5.3 Tampilan Menu Qstat.....	52
Gambar 4.5.5.4 Tampilan Menu Qsub.....	52
Gambar 4.5.5.5 Tampilan Menu Qdel.....	53
Gambar 4.5.5.6 Tampilan Menu SSH.....	53
<b>Gambar 5.1.b Tampilan Menu Application Submission.....</b>	<b>69</b>

# BAB 1

## PENDAHULUAN

Pada bab ini akan dibahas mengenai garis besar tugas akhir, yang meliputi Latar Belakang Masalah, Tujuan dan Manfaat Pembuatan, Permasalahan, Batasan Masalah, Metodologi Pembuatan Tugas Akhir, dan Sistematika Penulisan.

### 1.1 Latar Belakang

Perkembangan kecepatan prosesor berkembang sesuai dengan Hukum Moore, meskipun demikian bandwidth jaringan komputer berkembang jauh lebih pesat. Semakin cepatnya jalur komunikasi ini membuka peluang untuk menggabungkan kekuatan komputasi dari sumber-sumber komputasi yang terpisah. Perkembangan ini memungkinkan skala Komputasi Terdistribusi ditingkatkan lebih jauh lagi secara geografis, melintasi batas-batas domain administrasi yang sudah ada.

Inovasi sepanjang setengah abad telah meningkatkan kecepatan komputer dalam faktor jutaan. Tetapi kecepatan komputer masa kini masih terhitung rendah untuk berbagai aplikasi saintifik berskala raksasa. Salah satu solusi dari masalah kurangnya daya komputasi adalah dengan menjalankan aplikasi pada sejumlah komputer individual yang terhubung ke jaringan. Cara ini dalam terminologi teknis dikenal sebagai clustering. Walaupun clustering dapat menyediakan peningkatan daya komputasi secara signifikan, sebuah cluster membutuhkan fasilitas khusus dengan dibangun diatas sebuah lokasi tunggal. Ini memunculkan persoalan seberapa besar ruang yang dapat disediakan untuk menampung sekian banyak komputer yang bekerja secara bersamaan itu. Hal ini memunculkan persoalan baru dimana dibutuhkan investasi ekstra untuk membangun gedung-gedung baru, khusus sebagai lokasi cluster.

Grid computing sebenarnya merupakan sebuah aplikasi pengembangan dari jaringan komputer (network). Hanya saja, tidak seperti jaringan komputer konvensional yang berfokus pada

komunikasi antar piranti (device), aplikasi pada Grid computing dirancang untuk memanfaatkan sumber daya pada terminal dalam jaringannya. Grid computing biasanya diterapkan untuk menjalankan sebuah fungsi yang terlalu kompleks atau terlalu intensif untuk dikerjakan oleh satu sistem tunggal. Dalam pengertian yang lebih teknis, Grid computing merupakan sebuah sistem komputasi terdistribusi, yang memungkinkan seluruh sumber daya (resource) dalam jaringan, seperti pemrosesan, bandwidth jaringan, dan kapasitas media penyimpan, membentuk sebuah sistem tunggal secara virtual.

Setelah semua komputer yg dimiliki seorang "peneliti haus tenaga komputasi" dipergunakan secara terus menerus untuk memecahkan masalahnya, setelah berbagai cara untuk memecahkan masalah dicoba, dan dipilih yang paling efisien, tapi tetap masalahnya belum bisa dipecahkan juga, apa yang harus dia lakukan? **Komputasi Grid** adalah salah satu jawaban dari pertanyaan ini.

Dalam buku "**The Grid: Blue Print for a new computing infrastructure**" (Foster, Ian, may 2003. The University of Chicago), dijelaskan bahwa yang dimaksud dengan **komputasi grid** adalah

***"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."***

atau dengan kata lain,

***"infrastruktur perangkat keras dan perangkat lunak yang dapat menyediakan akses yang bisa diandalkan, konsisten, tahan lama dan tidak mahal terhadap kemampuan komputasi mutakhir yang tersedia."***

Seandainya kelak dikemudian hari teknologi yang dibutuhkan untuk mewujudkan visi paradigma ***Komputasi Grid***

ini sudah mapan, peluang akan semakin terbuka bagi kerjasama lintas organisasi, lintas benua dan lintas bangsa. Akan terbuka peluang bagi peneliti di Indonesia yang ingin melakukan komputasi yang sangat rumit, dengan menggunakan supercomputer tercepat di dunia, tanpa harus melakukan investasi besar-besaran dalam bidang teknologi informasi.

## **1.2 Rumusan Permasalahan**

Permasalahan yang diangkat dalam menyelesaikan penelitian ini adalah:

- Bagaimana melakukan koordinasi terhadap sumberdaya komputasi yang tidak berada dibawah suatu kendali terpusat. Seandainya sumber daya yang digunakan berada dalam satu cakupan domain administratif, maka komputasi tersebut belum dapat dikatakan komputasi grid.
- Bagaimana menggunakan standard dan protokol yang bersifat terbuka (tidak terpaud pada suatu implementasi atau produk tertentu). Komputasi grid dibutuhkan untuk mewujudkan komputasi bersama dalam skala besar. Kesepakatan dan standar yang dibutuhkan adalah dalam bidang autentikasi, otorisasi, pencarian sumberdaya, dan akses terhadap sumber daya.

## **1.3 Batasan Masalah**

Asumsi dan ruang lingkup permasalahan yang dikerjakan dalam Penelitian ini adalah:

- Infrastruktur yg dibangun akan berbentuk web services dan didalam web tersebut bisa di pasang aplikasi – aplikasi berbasis MPI.
- Sistem akan diujicobakan pada lingkungan Laboratorium Grid Computing Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (*ITS*).

#### 1.4 Tujuan dan Manfaat

Membangun infrastruktur untuk memanfaatkan sumber daya – sumber daya yang terdapat pada lingkungan ITS, khususnya yang menggunakan komputer sebagai alat komputasi. Dalam tugas akhir ini system yang akan dipakai berjalan diatas platform Globus Toolkit.

Dalam tugas akhir ini juga akan diuji cobakan aplikasi paralel untuk mengetahui apakah sistem tersebut berjalan dengan baik berdasarkan pada tool yang ada seperti Globus.

#### 1.5 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini adalah:

##### 1. Pemahaman Sistem dan Studi Literatur

Mencari dan Mempelajari bahan literatur tentang *grid computing*, *globus*, *MPI programming* serta *operating system linux* yang mampu menjalankan globus sebagai tools.

##### 2. Pengumpulan dan analisis data

Instalasi *linux* sebagai *operating system* yang dipakai untuk menjalankan *globus* tools yang merupakan *header node* serta membagi beberapa komputer sebagai *cluster node*.

##### 3. Implementasi Sistem

Mengimplementasikan desain sistem yang sudah dibuat berdasarkan *header* dan *cluster node*, yang akan menjalankan beberapa perintah dari *globus tools*.

##### 4. Uji Coba dan Evaluasi

Melakukan pengecekan terhadap portal yang telah di implementasi dan pengecekan terhadap fungsi – fungsi yang siap di tambahkan aplikasi baru, agar dapat berfungsi dengan maksimal kelak dikemudian hari.



## **5. Penyusunan Buku Tugas Akhir**

Tahap akhir dari proses pembuatan tugas akhir ini adalah penyusunan laporan atau dokumentasi secara lengkap dan menyeluruh dari semua kegiatan yang telah dilakukan. Pada tahap ini pula dirumuskan kesimpulan dan saran untuk pengembangan lebih lanjut.

### **1.6 Sistematika Penulisan**

Buku tugas akhir ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut:

#### **BAB I. PENDAHULUAN**

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

#### **BAB II. TINJAUAN PUSTAKA**

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan tugas akhir ini.

#### **BAB III. PERANCANGAN INFRASTRUKTUR DAN KONFIGURASI**

Bab ini membahas konfigurasi dan setting pada mesin – mesin yang akan dibangun.

#### **BAB IV. UJI COBA SISTEM**

Uji coba dan evaluasi sistem yang telah dibuat, akan dilakukan uji coba berdasarkan parameter-parameter yang ditetapkan, dan kemudian dilakukan analisis terhadap hasil uji coba tersebut.

## BAB V. PENUTUP

Bab ini berisi kesimpulan dari hasil uji coba yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

### 1.6. Sistematisa Pengajaran

Penerapan sistem ini dapat meningkatkan hasil belajar siswa.

### BAB I. PENDAHULUAN

Bab ini berisi latar belakang masalah, tujuan dan manfaat penelitian, metode penelitian, dan sistematika penulisan tesis.

### BAB II. TINJAUAN PUSTAKA

Bab ini membahas penelitian tentang pembelajaran berbasis multimedia dengan model pembelajaran dan media pembelajaran yang akan digunakan.

### BAB III. PERANCANGAN INSTRUKSI DAN KONTEN KASUS

Bab ini membahas konsep-konsep dan strategi yang akan digunakan dalam pembelajaran.

### BAB IV. UJI COBA SISTEM

Bab ini berisi uji coba sistem yang telah dibuat dan hasil dari uji coba tersebut. Uji coba tersebut dilakukan dengan cara yang sistematis dan komprehensif.

## **BAB 2**

### **TINJAUAN PUSTAKA**

Pada bab ini akan dibahas mengenai tinjauan pustaka yang menjadi dasar dari pembuatan Tugas Akhir.

#### **2.1 GRID History**

“Grid” computing muncul sebagai area baru yang penting, yang berbeda dengan distribusi conventional yang focus pada sharing sumberdaya, aplikasi inovatif dan berorientasi pada performansi yang tinggi.

##### **2.1.1 Anatomy Grid**

Tujuan dari pembuatan tugas akhir ini adalah membahas konsep Grid. Dalam proses ini, penulis mengembangkan arsitektur secara detail dan membuat pmetaan untuk pengembangan teknologi grid selanjutnya. Grid teknologi berbeda dengan teknologi yang sedang trend saat ini seperti internet, bisnis, peer-to-peer computing dan teknologi lainnya yang dapat membawa keuntungan yang signifikan. Spesifik masalah dari konsep grid ini adalah bagaimana dalam mengkoordinasikan dalam sharing sumberdaya dan penyelesaian masalah secara dinamis, multi-institutional virtual organitation. Dalam sharing sumberdaya, tidak mengutamakan dalam pemilihan file tapi bagaimana cara mengakses secara langsung untuk computer, software, data dan sumberdaya yang lain dibutuhkan kolaborasi dalam penyelesaian masalah yang muncul dalam dunia industri, ilmu pengetahuan dan teknik.. Sharing sumberdaya ini membutuhkan control sumberdaya yang tinggi dengan menyediakan sumberdaya dan memberikan batasan pada client. Dalam suatu instansi, mendefinisikan dengan cara memanggil virtual organization, seperti aplikasi servis profider, penyimpanan servis providers, siklus providers dan lain sebagainya. Contoh

tersebut merepresentasikan penyelesaian masalah secara komputerisasi dengan kolaborasi antara proses komputerisasi dengan lingkungannya. Teknologi grid sebanding dengan teknologi computer terdistribusi yang telah ada, sebagai contoh distribusi bisnis dengan system computer dapat menggunakan teknologi grid untuk bias melakukan sharing sumberdaya; Dalam ASP/ SSP, Teknologi Grid dapat digunakan untuk membangun pasar secara dinamis dengan proses yang terkomputerisasi. Hubungan dalam sharing sumberdaya antara Pembangunan, management dan eksploitasi secara dinamis dengan cross-organizational virtual organisasi akan menghasilkan teknologi baru. Dalam mendefinisikan arsitektur Grid, dimulai dari perspektif dan efektivitas operasi Virtual Organization yang memungkinkan pembangunan sumberdaya dengan banyak partisipan. Yang menjadi masalah dalam hal ini adalah Pengalamatan interoperability. Dalam lingkungan jaringan, interoperability menggunakan protocol yang umum.

### 2.1.2 Komputasi Terdistribusi

Tujuan dari komputasi terdistribusi adalah menyatukan kemampuan dari sumber daya (sumber komputasi atau sumber informasi) yang terpisah secara fisik, ke dalam suatu sistem gabungan yang terkoordinasi dengan kapasitas yang jauh melebihi dari kapasitas individual komponen-komponennya.

Tujuan lain yang ingin dicapai dalam komputasi terdistribusi adalah transparansi. Kenyataan bahwa sumber daya yang dipakai oleh pengguna sistem terdistribusi berada pada lokasi fisik yang terpisah, tidak perlu diketahui oleh pengguna tersebut. Transparansi ini memungkinkan pengguna sistem terdistribusi untuk melihat sumber daya yang terpisah tersebut seolah-olah sebagai satu sistem komputer tunggal, seperti yang biasa digunakannya.

Salah satu masalah yang dihadapi dalam usaha menyatukan sumber daya yang terpisah ini antara lain adalah skalabilitas, dapat atau tidaknya sistem tersebut dikembangkan.

### 2.1.3 Arsitektur Komputasi Terdistribusi

Banyak arsitektur perangkat lunak dan keras yang bervariasi yang digunakan untuk komputasi terdistribusi. Pada tingkat yang lebih rendah, penghubungan beberapa CPU dengan menggunakan jaringan sangat dibutuhkan. Pada tingkat yang lebih tinggi menghubungkan proses yang berjalan dalam CPU tersebut dengan sistem komunikasi juga dibutuhkan. Arsitektur umum yang memungkinkan sistem terdistribusi antara lain:

1. Client-server: klien menghubungi server untuk pengambilan data, kemudian server memformatnya dan menampilkannya ke pengguna.
2. Arsitektur 3-tier: Kebanyakan aplikasi web adalah 3-Tier.
3. Arsitektur N-tier: N-Tier biasanya menunjuk ke aplikasi web yang menyalurkan lagi permintaan kepada pelayanan enterprise. Aplikasi jenis ini paling berjasa bagi kesuksesan server aplikasi.
4. Tightly coupled: biasanya menunjuk kepada satu set mesin yang sangat bersatu yang menjalankan proses yang sama secara paralel, membagi tugas dalam bagian-bagian, dan kemudian mengumpulkan kembali dan menyatukannya sebagai hasil akhir.
5. Peer-to-peer: sebuah arsitektur di mana tidak terdapat mesin khusus yang melayani suatu pelayanan tertentu atau mengatur sumber daya dalam jaringan. Dan semua kewajiban dibagi rata ke seluruh mesin, yang dikenal sebagai peer.
6. Service oriented di mana sistem diatur sebagai satu set pelayanan yang dapat diberikan melalui antar-muka standar.
7. Mobile code: berdasarkan prinsip arsitektur mendekatkan pemrosesan ke sumber data

Replicated repository: Di mana repository dibuat replikanya dan disebarakan ke dalam sistem untuk membantu

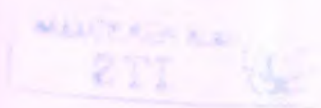


pemrosesan online/offline dengan syarat keterlambatan pembaharuan data dapat diterima

## 2.2 GRID

Grid adalah komputerisasi dan management data infrastructure yang menyediakan kemampuan electronic untuk masyarakat social dalam bisnis, pemerintahan, penelitian, ilmu pengetahuan dan hiburan.. The grid telah mantransformasikan ilmu pengetahuan, bisnis, kesehatan dan masyarakat. Konsep Grid computing pertama kali dieksplorasi pada tahun 1995 melalui eksperimen yang dikenal sebagai I-WAY, dimana jaringan berkecepatan tinggi digunakan untuk menghubungkan dalam waktu singkat, suber daya yang sifatnya high-end pada 17 situs di sepanjang Amerika bagian Utara. Selepas aktifitas ini, berkembang pula sejumlah proyek penelitian yang bertujuan untuk mengembangkan teknologi inti Grid computing untuk hal-hal yang lebih "produktif" bagi berbagai komunitas dan disiplin keilmuan. Tidak kurang dari badan bernama National Technology Grid bentukan US National Science Foundation (Lembaga Ilmu Pengetahuan AS), yang bekerjasama dengan Information Power Grid dari NASA (badan luar angkasa Amerika Serikat), bersama-sama membentuk sebuah infrastruktur Grid computing untuk melayani kegiatan para peneliti di NASA maupun berbagai universitas di Amerika Serikat. Di Eropa dan Amerika Serikat, European Data Grid, Particle Physics Data Grid, dan proyek Grid Physics Network (GriPhyN), berencana untuk membangun kerjasama dalam pengembangan aplikasi Grid computing untuk kepentingan analisis data pada eksperimen-eksperimen fisika.

Sementara itu, the Network for Earthquake Engineering Simulation Grid (NEESgrid) tengah berancang-ancang untuk menghubungkan para insinyur sipil dengan arsip data dan sistem simulasi komputer untuk mengembangkan bangunan dengan kekuatan yang lebih besar.

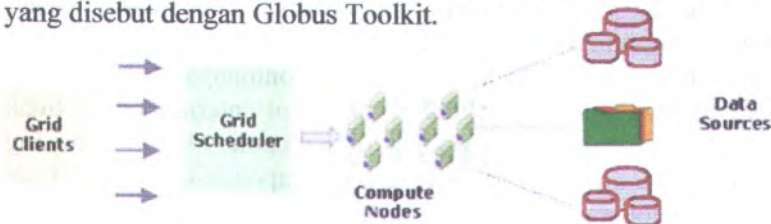


### 2.2.1 Arsitektur Grid

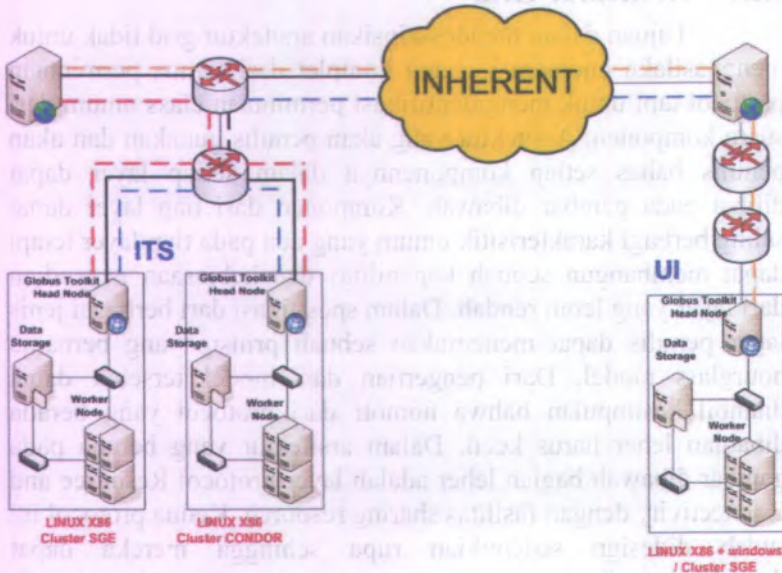
Tujuan dalam mendeskripsikan arsitektur grid tidak untuk menghasilkan enumerasi secara komplit dari semua permintaan protocol tapi untuk mengidentifikasi permintaan class umum dari suatu komponen. Arsitektur yang akan penulis gunakan dan akan penulis bahas setiap komponennya dalam setiap layer dapat dilihat pada gambar dibawah. Komponen dari tiap layer dapat saling berbagi karakteristik umum yang ada pada tiap layer tetapi dapat membangun sebuah kapabilitas dan kebiasaan penyediaan dari layer yang lebih rendah. Dalam spesifikasi dari berbagai jenis layer penulis dapat menemukan sebuah prinsip yang bernama hourglass model. Dari pengertian dari model tersebut dapat diambil kesimpulan bahwa nomotr dari protocol yang berada dibagian leher harus kecil. Dalam arsitektur yang berada pada gambar dibawah bagian leher adalah layer protocol Resource and Connectivity dengan fasilitas sharing resourch. Kedua protocol ini sudah didesign sedemikian rupa sehingga mereka dapat diimplementasikan untuk secara otomatis menemukan Fabric Layer dan dapat berjalan dalam sebuah konstruktor dari layanan global dan layer kolektif penulis sebut dengan aplikasi yang lebih spesifik karena pada layer ini dapat menentukan koordinat mereka.

Arsitektur yang penulis gunakan merupakan arsitektur yang dapat dikatakan sebagai arsitektur level tinggi . Dimana

semua prtokol yang berada pada arsitektur ni merupakan protocol standart yang disusun oleh sebuah komunitas terbuka yang disebut dengan Globus Toolkit.



**Gambar 2.1.1 Arsitektur umum Grid**



**Gambar 2.2.2** Arsitektur Grid ITS

### 2.2.2 Layer Fabric

Layer fabric yang berada pada arsitektur ini merupakan layer yang menyediakan sumberdaya dari grid protocol. Sumberdaya-sumberdaya tersebut contohnya : sumberdaya komputasional, system penyimpanan (storage system), catalog, sumberdaya jaringan(network resources), dan sensor. Dari sumberdaya-sumberdaya tersebut juga terdapat sumberdaya – sumberdaya yang berbentuk logika contohnya dalam kasus pengimplementasian sumberdaya protocol internal tetapi tidak terkonsentrasi pada Grid Arsitektur. Komponen-komponen yang berada pada layer Fabric merupakan komponen-komponen local yang mengoperasikan operasi pada level yang tinggi. Berikut ini merupakan beberapa kapabilitas yang terdapat pada layer ini :



- a. memulai program dan memonitor serta mengontrol hasil eksekusi suatu proses. Mekanisme control pada sumberdaya untuk alokasi proses sangatlah berguna. Dibutuhkan suatu fungsi yang dapat mengontrol karakteristik hardware dan software dengan baik.
- b. **Storage resources**: Mekanisme yang digunakan untuk mengambil dan mendapatkan file.
- c. **Network resources**: Mekanisme pengaturan yang berisi control sumberdaya yang dialokasikan kedalam transfer jaringan sehingga memiliki nilai lebih dalam penggunaannya. Fungsi ini diminta untuk menyediakan karakteristik jaringan dan mengambilnya.
- d. **Code repositories**: Form ini khusus untuk menyimpan sumberdaya untuk manage source dan object Catalogs: Tempat untuk menyimpan sumberdaya yang diimplementasikan dalam catalog query dan operasi update.
- e. **Connectivity layer** didefinisikan dalam core communication and authentication protocols yang diminta untuk spesifik grid dalam transaksi di jaringan. Authentication protocols dibangun untuk memberikan servis komunikasi dengan kriptografi sebagai keamanan dalam verifikasi dan identifikasi user dan sumberdaya.

### 2.2.3 Layer Connectivity

Layer yang kedua yang akan penulis bahas adalah layer connectivity. Layer ini merupakan layer yang mendefinisikan tentang inti dari komunikasi atau core communication dan pengidentifikasi protocol untuk transaksi yang terjadi pada jaringan Grid specific. Perlu juga penulis ketahui bahwa komunikasi protocol tidak dapat digunakan untuk melakukan perubahan terhadap data antara sumberdaya dengan layer fabric. Pengidentifikasi protocol dibangun dari layanan komunikasi oleh penyedia layanan kriptografi. Terdapat beberapa permintaan komunikasi yang dapat dilakukan yaitu : transport, routing dan

naming. Sebagai bagian dari penanggapan yang cepat layer ini menyediakan sebuah layanan keamanan yang merupakan layanan keamanan standrt yang dibangun dari konsep internet protocol. Solusi keamanan yang disediakan oleh grid ini telah mencakup untuk penyediaan keamanan yang mendukung fleksibilitas untuk proteksi dalam berkomunikasi. Maksud dari fleksibilitas disini adalah menyediakan bebrapa layanan pendukung seperti : control terhadap proteksi, menyediakan unit data independent untuk penentuan protocol, mendukung untuk protocol transport yang biasa digunakan seperti TCP.

#### **2.2.4 Layer Resource**

Layer yang ketiga ini merupakan layer yang membangun layer connectivity untuk menemukan protocol seperti APIs dan SDKs untuk secure negotiation, inisialisasi, monitoring, control, accounting dan pembayaran terhadap sharing operation dalam sumberdaya yang individu. Implementasi dari layer resource untuk protocol ini dapat penulis sebut dengan fungsi dari Fabric layer untuk mengakses dan mengontrol dari sumberdaya local.

Terdapat dua kelas utama dari resource layer :

- a. information protocol yang digunakan untuk mneginformasikan tentang struktur dan langkah-langkah dari sumberdaya yang ada seperti konfigurasi, current load, dan usage policy
- b. management protocol yang digunakan untuk menegosiasikan untuk akses dalam sharing sumberdaya, penspesifikasian, sebagai contoh pencarian sumber daya untuk sebuah performansi seperti pembuatan sebuah proses atau mengakses data. Sejak management protocol dapat digunakan untuk menginisialisasi sharing relationship maka mereka harus mendapatkan "policy application point".

Sejak banyak macam protocol dapat diciptakan maka layer resource dari leher dan terdapat pembatasan terhadap focus set.

### 2.2.5 Layer Collective

Pada saat layer resource difokuskan pada interaksi antara tiap sumberdaya maka pada layer berikutnya dalam arsitektur ini yaitu layer collective lebih membahas pada sumberdaya secara global. Karena kolektive komponen dibangun dari sumber daya yang terbatas dan layer connectivity merupakan leher dalam protocol hourglass maka mereka dapat mengimplementasikan maka mereka dapat mengimplementasikan variasi dari sharing kemampuan tanpa pencarian permintaan yang baru. Sebagai contoh :

1. Layanan directory dengan partisipasi dari virtual organisasi untuk menemukan property dari sumberdaya virtual organisasi.
2. Co-allocation, penjadwalan, dan pemutusan layanan dengan virtual organisasi sebagai participant untuk meminta alokasi dari satu atau lebih sumberdaya untuk tujuan yang lebih spesifik dan tugas penjadwalan.
3. Grid-enabled programming systems memungkinkan model programing yang Familiar untuk digunakan dalam lingkungan Grid, menggunakan jenis layanan grid untuk menemukan alamat resource, security, resource alocation, dan jenis layanan yang lainnya.
4. Workload management systems and collaboration frameworks- juga untuk mengetahui solusi dari masalah dengan menyediakan deskripsi,kegunaan, dan langkah-langkah untuk manajemen,asynchronous, multi-component workflows contoh ini dapat memberikan suatu ilustrasi yang lebih luas mengenai jenis Collective layer protocol dan layanan yang diberikan. fungsi collectivitas dapat dimplementasikan sebagai layanan yang tersusun berlangsung, dengan associated protocols,atau dengan associated APIs designed untuk link dengan suatu aplikasi. pada beberapa kasus, implementasinya dapat dibangun dalam Resource layer protocol dan APIs. lebih dari itu, penulis mendefenisikan suatu Co-allocation API

untuk implemntasi operasi co-allocation dan mungkin menyediakan fungsi tambahan yang lain, seperti otorisasi, penanganan(toleransi) kesalahan, dan logging. Suatu aplikasi mungkin menggunakan layanan co-reservation protokol untuk meminta end-to-rnd network reservation.komponen dari Collective layer mungkin menyesuaikan dengan kebutuhan dari komunitas user, Virtual Organization, atau application domain. komponen Collective yang lain dapat memiliki tujuan yang umum.

### **2.2.6 Layer Aplikasi**

Layer terakhir dalam arsitektur Grid terdiri dari user application dijalankan didalam lingkungan virtual organisation(VO). Aplikasi dibangun oleh beberapa faktor, dan dari pemanggilan, layanan yang ada pada setiap layer. Pada masing-masing layer, terdapat protokol-protokol yang menyediakan akses pada beberapa layanan yang berguna seperti resource management, data access, resource discovery, dan sebagainya. Pada masing-masing layer, APIs mungkin juga mendefinisikan siapa yang mengimplementasi pergantian protocol messages dengan layanan yang tepat untuk melakukan suatu tindakan-tindakan.

### **2.3 Pelayanan GRID**

Di antara e-business dan e-science, penulis harus mengintegrasikan layanan pendistribusian diantara keduanya. Didalam sebuah perusahaan tunggal dan atau dari sumberdaya eksternal yang berbagi dan melayani hubungan antara layanan provider maka akan tercipta keheterogean dan dinamisasi dalam virtual organization. Pengintegrasian ini secara teknis akan menantang karena adanya kebutuhan untuk mencapai kualitas layanan terbaik pada saat melayani dari paltform yang berbeda. Pada bagian ini penulis akan menjelaskan tentang OGSA(Open Grid Services Architecture). Berdasarkan teknologi dan konsep dari layanan Grid dan Web maka akan tercipta suatu arsitektur

yang akan melakukan penyeragaman terhadap jasa dan layanan serta menemukan suatu mekanisme standar untuk menciptakan, penamaan dan menemukan pelayanan Grid dengan cepat juga menyediakan tempat yang terbuka, berbagai protokol yang mengikat dan mendukung pengintegrasian dengan dasar fasilitas platform yang asli. Dalam kaitannya dengan WSDL (Web Services Description Language), OGSA akan menggambarkan hubungan yang terkonvensi. Hal ini didasari oleh karena sebuah pelayanan yang baik dapat menjadi dukungan terhadap suatu kepercayaan, pengesahan, otorisasi dan pendelegasian. Bagian yang penulis jelaskan ini merupakan pelengkap dari artikel yang telah dibuat sebelumnya. Penulis akan menjelaskan bagaimana gambaran mekanisme Grid yang Service Oriented Architecture dan kemampuan Grid jika disatukan dengan sebuah layanan web atau secara garis besar bagaimana pelayanan yang dilakukan oleh grid.

### **2.3.1 Open Grid Services Architecture**

Sebelumnya penulis harus berargumentasi bahwa infrastruktur IT di internal suatu perusahaan, SP-ENHANCED infrastruktur, dan multi-organizational Grids, Kegiatan yang berkaitan dengan perhitungan akan terus meningkat terkait dengan makin meningkatnya pula kegiatan yang dilakukan oleh suatu ciptaan, manajemen dan sebuah aplikasi. Namun hal ini tergantung pula oleh bagaimana struktur dari perusahaan tersebut dan juga terkait dengan apakah perusahaan tersebut merupakan perusahaan yang homogen atau heterogen. Sebuah individu dalam perusahaan mungkin tersusun dari sebuah sistem hirarki yang kecil-kecil dan akan tumpang tindih dengan individu yang lain dalam perusahaan. Dengan mengabaikan perbedaan ini, para pengembang aplikasi untuk VOS menghadapi berbagai kebutuhan umum atau masalah yang telah umum yang memang harus dihadapi dalam sebuah perusahaan seperti masalah keamanan membuat struktur kerja, membagi-bagi struktur kerja dan manajemen sumber daya. Pembahasan penulis sekarang lebih

megarrah kepada sifat alamiah dari berbagai kebutuhan diatas dan bagaimana mekanisme untuk menjadikan semua kebutuhan diatas menjadi praktis. Dengan itu maka penulis akan melakukan pembahasan mengenai OGSA yang merupakan arsitektur yang mendukung penciptaan, pemeliharaan dan layanan aplikasi yang dipelihara VO's. Pembahasan akan penulis mulai mengenai kegunaan dari OGSA serta layanan apa yang terdapat dalam arsitektur ini

### **2.3.1.1 Orientasi Pada Layanan**

Ketika penulis menggambarkan sebuah VO's penulis dapat memusatkan pada sumber daya fisik yang sedang digunakan secara bersama yang juga didukung oleh layanan ini. Sebuah layanan dapat penulis sebut dengan sebuah jaringan yang tak terlihat yang akan menyediakan beberapa kemampuan. Sebuah terminologi obyek yang biasa dibantah juga dapat digunakan, tetapi penulis akan menghindari terminologi itu dalam kaitannya yang mempunyai arti terlalu berlebihan. Di dalam OGSA, penulis memusatkan pada jasa yaitu pengkomputerasian sumber daya, gudang/penyimpanan sumber daya, jaringan, program, database, dll diman semuanya akan diwakili oleh jasa. Sebuah pandangan yang berorientasi terhadap objek akan mengijinkan penulis untuk membuat kebutuhan akan defenisi mekanisme interface baku, Transparansi local, adaptasi ke jasa OS lokal. Suatu pandangan service-oriented juga menyederhanakan virtualization diamana encapsulation berada di belakang suatu alat penghubung umum dari implementasi yang berbeda. Virtualisasi akan mempertimbangkan konsistensi apabila sumber daya akan mengakses ke platform heterogen dengan penempatan yang remote atau lokal ketransparanan, dan memungkinkan pemetaan berbagai kejadian sumber daya logis ke fisik yang sama sumber daya dan manajemen sumber daya di dalam suatu VO's berdasarkan pada komposisi dari tingkat sumber daya yang lebih rendah. Virtualisasi mengijinkan komposisi jasa untuk membentuk layanan yang lebih canggih tanpa memperdulikan aturan bagaimana jasa yang sedang berjalan

diterapkan. Virtualisasi dari Jasa Grid juga memiliki kemampuan untuk memetakan layanan umum perilaku semantik ke fasilitas platform yang asli.

### **2.3.1.2 Proses Layanan Semantik**

Penulis juga memerlukan ilmu semantik standard untuk interaksi layanan sedemikian sehingga, sebagai contoh, jasa yang berbeda mengikuti konvensi yang sama untuk pemberitahuan kesalahan. Sampailah di Sini, OGSA menggambarkan apa yang penulis panggil dengan suatu layanan Grid yaitu suatu Web yang melayani itu akan menyediakan satu set alat penghubung yang akan dirumuskan dengan baik dan mengikuti konvensi yang spesifik. Alat penghubung menunjuk penemuan, ciptaan jasa yang dinamis, manajemen semumur hidup, pemberitahuan, dan kemampuan memanager. Konvensi yang dimaksud diatas merujuk pada penamaan. Tentunya penulis juga mengharapkan otorisasi dan kendalo concurency akan ditangani oleh OGSA. Kemampuan penulis terhadap virtualisasi dan menyusun jasa tergantung pada lebih dari satu defenisi interface baku. Dua isu penting lainnya adalah pengesahan dan jasa yang dapat dipercaya. Semua perhatian ini akan meningkatkan keadaan umum arsitektur tanpa kemampuan yang mencurigakan. Sebuah konvensi dan alat penghubung yang menggambarkan suatu layanan Grid terkait, khususnya dengan perilaku yang berhubungan dengan manajemen dari kejadian jasa yang temporer. Para peserta VO's secara statis akan memelihara kegigihan dalam pelayanan dan tidak bergantung kepada klien secara terus menerus. Mereka sering membutuhkan sebuah pelayanan instance yang dinamis, yang kemudian menangani manajemen dan hubungan interaksi dengan status yang diminta dai sebuah aktivitas. Apabila sebuah aktivitas yang tidak terlalu diperlukan maka jasa dapat dihancurkan. Sebagai contoh di dalam suatu sistem video konfrensi, penetapan dari suatu sesi video konfrensi mungkin melibatkan jasa pada untuk mengatur end-to-end arus data menurut Qos batasan atau didalam suatu Web yang melayani

lingkungan kejadian jasa boleh jadi instance yang dengan dinamis menyediakan layanan akan terpotong.

### 2.3.2 Model Layanan OGSA

Suatu pendapat dasar tentang OGSA adalah segalanya diwakili oleh suatu jasa yaitu suatu jaringan memungkinkan menyediakan beberapa kemampuan melalui pertukaran pesan. Komputerisasi sumber daya, sumber daya gudang/penyimpanan, jaringan, program, database, dan sebagainya adalah jasa. Adopsi ini berasal dari suatu model penyeragaman service-oriented yang berarti bahwa semua komponen berada pada lingkungan sebetulnya. Secara rinci, OGSA menghadirkan segalanya sebagai pelayanan Grid yaitu suatu pelayanan web menyesuaikan diri untuk satu set konvensi dan interface baku yang merupakan pendukung untuk mendapat tujuan seperti manajemen seumur hidup. Inti dari semua ini adalah alat penghubung yang konsisten dari semua layanan grid yang diterapkan, memudahkan konstruksi jasa higher-order yang dapat diperlakukan di suatu penyeragaman.

Sebab Jasa Grid adalah dinamis dan stateful, penulis memerlukan suatu cara untuk membedakan antara berbagai layanan dinamis yang diciptakan oleh tiap orang. Kemudian, tiap-tiap layanan Grid diberi nama yang unik. Hal ini dilakukan oleh GSH atau Grid Service Handle yang akan mencirikan secara spesifik dari semua layanan grid baik itu pada masa sekarang sampai masa yang akan datang. Jika terdapat pelayan jasa yang gagal maka GSH yang sama dapat digunakan kembali. Jasa Grid mungkin dapat diupgrade selama seumur hidup, sebagai contoh untuk mendukung protokol yang versi yang baru atau untuk menambahkan protokol alternatif. Kemudian, GSH tidak akan membawa protokol apapun tidak seperti pengalaman jaringan. Sebagai gantinya, informasi ini di enkapsulasi bersama dengan semua instance-specific yang memerlukan hubungan ke jasa yang spesifik, ke dalam abstrak tunggal memanggil suatu layanan Grid yang disebut dengan GSR atau Grid Service



Reference. Tidak sama dengan GSH yang invariant, GSR untuk suatu layanan Grid dapat bertukar service itu seumur hidup. Suatu GSR mempunyai suatu waktu habis yang eksplisit atau mungkin menjadi cacat pada setiap waktu selama service's seumur hidup, dan OGSA menggambarkan mekanisme pemetaan, karena perolehan suatu GSR yang dapat diperbaharui. Hasil dalam menggunakan suatu GSR yang seumur hidup tak tergambarkan. Perlu juga penulis ketahui bahwa memegang suatu GSR yang sah tidak menjamin akses suatu layanan grid yang mana kebijakan lokal atau mengakses kendali batas ( untuk contoh meminta jumlah nomor) mungkin service akan melarang.

### **2.3.2.1 Open Grid Services Infrastructure (OGSI)**

OGSI.NET adalah sebuah platform untuk grid computing pada .NET dan jembatan untuk solusi grid computing di mesin unix. Versi yang paling terakhir adalah OGSI.NET 2.17. ini berisi sebuah OGSI compliant hosting container dan banyak contoh2 layanan dan beberapa klien termasuk graphical browser. OGSI (Open Grid Services Infrastructure) adalah sekumpulan dari spesifikasi WSDL (Web Services Definition Language) yang mendefinisikan standard interfaces, kelakuan, dan skema dari konsistensi grid computing dengan OGSA vision. Interfaces dan behaviour ini mendefinisikan Grid Service. Grid Services adalah sebuah Web Services yang meng-conform kedalam set conventions (Interfaces dan Behaviors) yang mendefinisikan bagaimana client berinteraksi dengan sebuah Grid Service. Konvensi ini, dan mekanisme OGSI yang lain yang berhubungan dengan creation dan discovery Grid Service, menyediakan pengaturan, penanganan kesalahan, dan manajemen keamanan distribusi. Spesifikasi OGSI diimplementasikan dalam sejumlah platform yg berbeda termasuk .NET, java(GT3 dari globus project) dan lainnya.

Open Grid Services Architecture (OGSA) [Grid Physiology] mengintegrasikan teknologi key Grid dengan mekanisme Web Services untuk membentuk sistem framework

terdistribusi berdasarkan OGSi. Instance dari sebuah Grid Service adalah layanan transient yang berpotensi yang meng-conform menjadi suatu set konvensi, yang digunakan sebagai WSDL interfaces, extensions, dan behavior, untuk tujuan seperti manajemen siklus, pendefinisian karakter, dan notification. Grid Services menyediakan manajemen pengaturan distribusi. OGSi mendefinisikan sebuah komponen model yang menggunakan skema WSDL dan XML definition untuk melakukan konsep-konsep :

- Stateful Web Services
- Extension of Web Services interfaces
- Asynchronous notification of state change
- References to instances of services
- Collections of services instances, and

Services state data that augments the constraint capabilities of XML Schema Definition

### 2.3.2.2 Value dalam OGSi

Terdapat beberapa value yang ditambahkan kedalam OGSi yaitu :

Spesifikasi OGSi mendefinisikan sekumpulan dari tingkah laku dari web services yang relevant dengan grid computing. Dalam hal ini, the OGSi spesifikasi dan OGSi working group seperti the web services interoperability organization (ws-1), tetapi mengutamakan standardisasi cukup penting untuk memungkinkan truly large-scale grids. Sebagai contoh, OGSi mendefinisikan interface manajemen siklus untuk mengirim isi dari layanan grid.

OGSi mendefinisikan ide pengiriman, contohnya : short-lived, services. Sebagai contoh, sebuah computational job dapat ditampilkan sebagai service. Sementara itu, web services tidak mendukung notion ini.

Spesifikasi OGSi mendefinisikan sebuah two-level naming scheme berdasarkan Grid Service Handles(GSHs) dan Grid

Service References(GSRs). Tiap GSH adalah identifier global untuk instance dari layanan grid yang bersifat unik untuk tiap waktu. Ketika sebuah GSH adalah sebuah Global Unique handle untuk a grid service, ini tidak berisi semua kemungkinan information dinamis yang dibutuhkan untuk berkomunikasi dengan seorang client. OGSi menyediakan untuk sebuah GSH untuk diresolve ke dalam satu atau lebih GSRs yang berisi semua informasi yang dibutuhkan untuk berkomunikasi dengan client-client yang menggunakan satu atau lebih protocol binding.

OGSI menyediakan sebuah model untuk mengakses internal state yang sebuah grid pilih untuk memberitahu public. SDE yang kepanjangannya service Data Element adalah suatu model yang menyediakan mekanisme standard untuk mengquery, update dan menambahkan dan menghapus data yang berhubungan dengan tiap instance layanan grid.

## BAB 3 PERANCANGAN INFRASTRUKTUR DAN IMPLEMENTASI

Dalam mendefinisikan Open Grid Service Architecture dapat ditinjau dari dua teknologi yaitu Globus Toolkit, yang sudah dipakai secara meluas dalam pendekatan ilmiah dan teknikal computer, dan Web service yang munculkan sebagai standar framework dalam mengakses aplikasi jaringan.

### 3.1 ARSITEKTUR UMUM



*Gambar 3.1 Arsitektur umum*

#### 3.1.1 Grid Clients

*Grid clients* merupakan client yang mengirimkan sejumlah data untuk diproses oleh head node yang kemudian diproses oleh scheduler.

Setelah user mengirimkan data kepada head node, kemudian dalam hitungan waktu maka user akan menerima kembali hasil perhitungan dan eksekusi dari worker node.

#### 3.1.2 Grid Scheduler

*Grid scheduler* adalah pembagi pekerjaan yang kemudian akan mendistribusikannya kepada worker node untuk diproses dan dikembalikan kepada scheduler. Fungsi utamanya adalah mengatur penjadwalan dan melakukan *remote machines* serta menjalankan perintah2 dalam unix

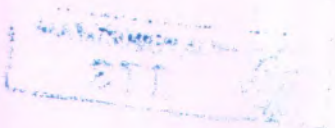
Ada beberapa jenis scheduler, antara lain SGE (*Sun Grid Engine*) dan Condor. Dua jenis scheduler inilah yang banyak dipakai dalam mesin scheduler yang terdapat pada headnode. Seperti yang digunakan pada cluster ITS saat ini menggunakan scheduler SGE (*Sun Grid Engine*) untuk cluster Informatika, sedangkan untuk cluster Elektro menggunakan Condor sebagai schedulernya

### 3.1.3 Compute Nodes

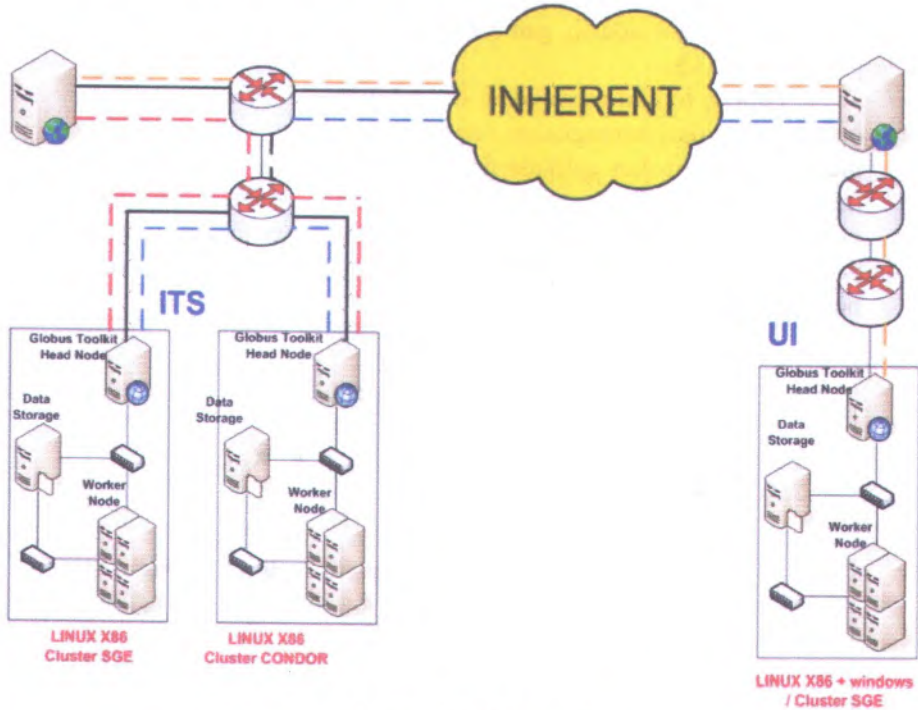
*Compute nodes* adalah pekerja dari system dalam *grid computing*. Selain itu fungsi dari setiap mesin yang ada akan berbeda ketika melakukan proses pengerjaan dan kompilasi data yang diolah. Setelah semua data yang diolah dan diproses, belum tentu data – data tersebut akan keluar secara bersamaan dari compute node itu secara langsung, karena scheduler yang akan mengaturnya secara bertahap.

### 3.1.4 Data Storage/ sources

*Data storages/ sources* merupakan tempat terjadinya pertukaran data dan penyimpanan data antara mesin worker dengan mesin headnode. Dimana mekanisme yang digunakan untuk mengambil dan mendapatkan file mengacu pada penggunaan *NFS (network file system)*. Agar antara headnode dan worker dapat saling melakukan *share* data. Agar ketika client melakukan *submit jobs*, maka akan terjadi pengaksesan data storage melalui headnode, dan saat *jobs* tersebut dijalankan oleh cluster, tiap2 worker akan mengacu pada data yang sama yang tersimpan didalam data storage tersebut.



### 3.2 ARSITEKTUR SISTEM



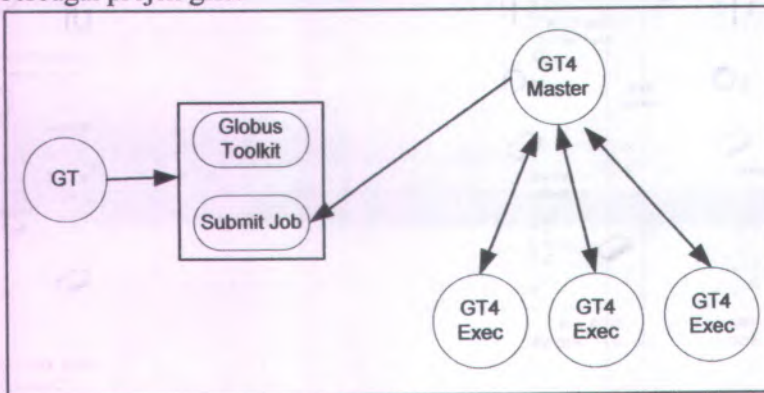
Gambar 3.2 Arsitektur Sistem

Pada arsitektur system ini, dijelaskan bahwa terdapat 2 cluster yang ada pada ITS Grid. Dimana cluster pertama adalah cluster Informatika menggunakan *scheduler* SGE dan cluster Elektro menggunakan *scheduler* Condor. Sedangkan pada headnode, kedua cluster ini tetap menggunakan GT4 (*Globus toolkit 4*). System yang terdapat pada ITS Grid juga menggunakan arsitektur yang sama seperti pada arsitektur umum system.

Arsitektur system yang digunakan pada ITS Grid juga serupa dengan arsitektur yang digunakan pada cluster UI. Mulai dari penggunaan *tools* pada *headnode* dan *scheduler* yang digunakan.

Berikut ini adalah gambaran mengenai cara kerja globus pada cluster ITS.

Globus toolkit merupakan dasar komunity, open-arsitektur, open source dari serangkaian service dan library software yang mendukung grid dan aplikasi grid. Toolkit adalah pengalaman masalah keamanan, penemuan informasi, manajemen resource, manajemen data, komunikasi, deteksi kesalahan, serta portabilitas. Mekanismenya adalah memakai ratusan situs dan berbagai projek grid.



**Gambar 3.2.1 Cara kerja globus**

Layaknya sebuah perusahaan atau organisasi, dalam GT4 terdapat MDS (*Monitoring and Discovery Service*), selain itu untuk dapat menggunakan MDS ini dengan benar, maka MDS ini harus dipasang secara terdistribusi. Caranya adalah kita membuat suatu struktur *virtual organization* (VO).

Sebuah VO terdiri dari individual – individual atau institusi yang membentuk satuan unit dan memiliki aturan – aturan spesifik dalam membagikan sumber daya yang dimiliki. Beberapa komponen MDS dipasang pada pusat – pusat VO, sementara lainnya dipasang pada *node – node* individual yang ada

Pada GT4, MDS diimplementasikan menggunakan WSRF. WS MDS GT4 itu sendiri terdiri dari :

1. Service Index  
Yang mengumpulkan informasi mengenai hasil monitor dan pencarian pada sumber daya Grid dan memublikasikannya pada satu lokasi tunggal.
2. Service Trigger  
Yang mengambil data dari sumber daya Grid yang ada dan bila kondisi dari data tersebut sesuai dengan aturan yang telah didefinisikan sebelumnya, maka akan dilakukan suatu aksi.
3. Agregator Framework  
Merupakan *framework software* yang digunakan untuk membangun servis yang mengumpulkan data
4. WebMDS  
Yang memungkinkan pengguna akhir untuk melihat informasi monitor pada Grid cukup dengan sebuah browser.

### 3.3 SPESIFIKASI SISTEM PERANTI KERAS DAN LUNAK

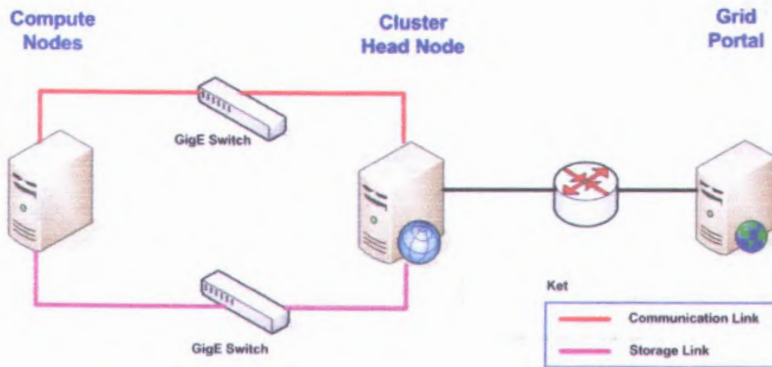
Sebenarnya tidak ada keterangan spesifik mengenai kebutuhan peranti keras untuk Globus Toolkit 4 (GT4), selama masih dapat menjalankan Java. Berdasarkan pengalaman, berikut ini adalah peranti minimum yang direkomendasikan:

- a. **Peranti Keras** (*minimum requirement*)
  - a. CPU dengan processor Pentium II 400 MHz
  - b. Memori 256 MB
  - c. Partisi harddisk sebesar 500 MB untuk instalasi minimum
  - d. Partisi harddisk sebesar 3 GB untuk instalasi penuh
- b. **Peranti Lunak**  
Untuk membangun ITS Grid itu sendiri aplikasi/ software yang digunakan adalah :
  - a. UCLA Grid portal



- Grid portal merupakan bentuk aplikasi web yang dipakai untuk melihat secara visual cara kerja *grid engine* dan *Globus Toolkit*
- b. Sun Grid Engine  
Mengatur penjadwalan dan melakukan *remote machines* serta menjalankan perintah2 dalam unix
  - c. MPICH  
Aplikasi berbasis MPI
  - d. Ant 1.5.1+  
*Framework* yang digunakan untuk mendukung J2SDK
  - e. Globus Toolkit 4.0.3  
Management Tools
  - f. JD2SDK 1.4.2+  
*Framework* java yang dipakai untuk menjalankan aplikasi berbasis java.
  - g. Apache Tomcat 5.0+  
Sebuah prasyarat bila kita ingin menggunakan fasilitas webMDS GT4
  - h. JUnit 3.8+  
Adalah sebuah *framework* untuk melakukan *testing*

### 3.4 SKEMA



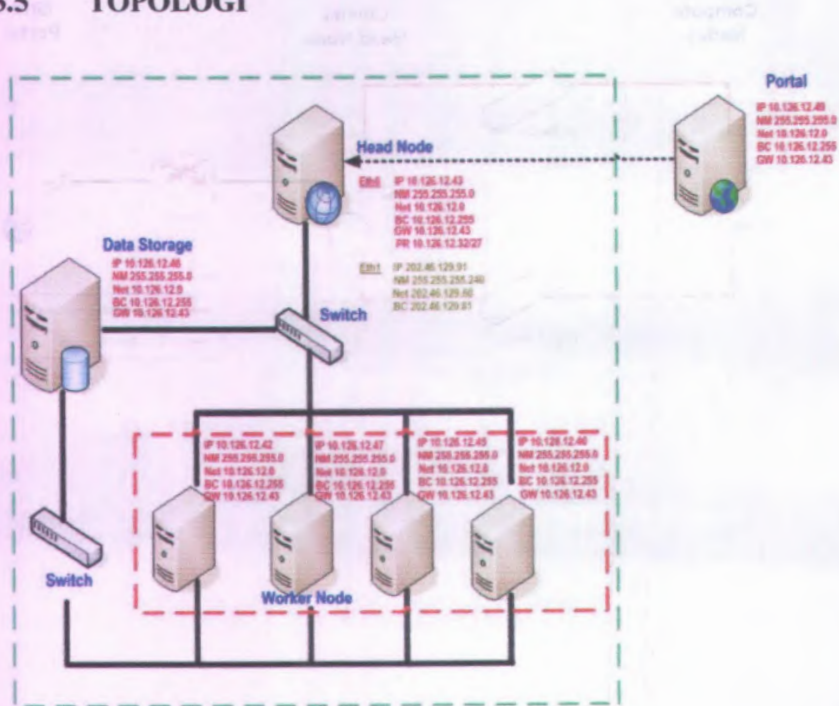
**Gambar 3.4 Skema ITS Grid**

Secara umum penjelasan mengenai skema ITS Grid adalah:

- Setiap node tersambung pada 2 GigE switch. Satu dipergunakan untuk storage link (khusus untuk NFS) dan lainnya dipergunakan untuk hal lainnya termasuk komunikasi saat menjalankan aplikasi MPI.
- Skema saat ini hanya menggambarkan kondisi pada 1 cluster saja yang terdapat pada cluster ITS yaitu

cluster Informatika, karena Cluster Elektro masih dalam pembuatan

### 3.5 TOPOLOGI



Gambar 3.5 Topologi

Topologi yang ada saat ini pada cluster Informatika yang dibangun adalah sebagai berikut :

**a. Head Node**

**eth0 :**

IP Address	10.126.12.43
Netmask	255.255.255.0
Network	10.126.12.0
Broadcast	10.126.12.255
Gateway	10.126.12.43

*Post Routing*    **10.126.12.32/27**

**eth1 :**

*IP Address*    **202.46.129.91**  
*Netmask*      **255.255.255.240**  
*Network*      **202.46.129.80**  
*Broadcast*    **202.46.129.81**

**b. Data Storage**

*IP Address*    **10.126.12.48**  
*Netmask*      **255.255.255.0**  
*Network*      **10.126.12.0**  
*Broadcast*    **10.126.12.255**  
*Gateway*     **10.126.12.43**

**c. Portal**

*IP Address*    **10.126.12.49**  
*Netmask*      **255.255.255.0**  
*Network*      **10.126.12.0**  
*Broadcast*    **10.126.12.255**  
*Gateway*     **10.126.12.43**

**d. Worker**

- Worker 1  
*IP Address*    **10.126.12.42**  
*Netmask*      **255.255.255.0**  
*Network*      **10.126.12.0**  
*Broadcast*    **10.126.12.255**  
*Gateway*     **10.126.12.43**
  
- Worker 2  
*IP Address*    **10.126.12.45**  
*Netmask*      **255.255.255.0**  
*Network*      **10.126.12.0**  
*Broadcast*    **10.126.12.255**

*Gateway* 10.126.12.43

- Worker 3  
*IP Address* 10.126.12.46  
*Netmask* 255.255.255.0  
*Network* 10.126.12.0  
*Broadcast* 10.126.12.255  
*Gateway* 10.126.12.43
- Worker 4  
*IP Address* 10.126.12.47  
*Netmask* 255.255.255.0  
*Network* 10.126.12.0  
*Broadcast* 10.126.12.255  
*Gateway* 10.126.12.43

Konfigurasi yang terdapat pada cluster ITS Grid ini menggunakan 1 IP Public yang terdapat digunakan pada Head node, bertujuan agar headnode pada cluster ITS khususnya *management tools* yang digunakan yaitu *Globus Toolkit* dapat berkomunikasi dengan *Globus Toolkit* lainnya yang terdapat pada cluster Elektro dan cluster UI.

### 3.6 KONFIGURASI CLUSTER DAN INSTALASI

#### 3.6.1 Mesin Head Node

Panduan Instalasi ini akan terbagi menjadi tiga bagian utama, yaitu panduan instalasi bagi Mesin Head Node, Mesin Data Storage, dan Mesin Worker. Seluruh petunjuk instalasi yang tertulis pada dokumen ini perlu dilakukan secara berurutan. Jangan melakukan suatu tahapan instalasi sebelum tahapan sebelumnya berhasil dengan baik.

### 3.6.1.1 Persiapan

Setidaknya ada dua hal yang perlu disiapkan, yaitu perangkat lunak dan perangkat keras. Perangkat lunak yang digunakan setidaknya terdapat pada dua lokasi, yaitu :

- a. Arsip perangkat lunak.
- b. Repository Debian GNU/Linux.

Dari sisi perangkat keras, hal – hal yang perlu dipersiapkan adalah:

- a. Mesin Head Node
- b. Mesin Data Storage
- c. Mesin Worker Node
- d. Switch Koneksi (24 port Gigabit Ethernet switch).
- e. Switch Data (24 port Gigabit Ethernet switch).

### 3.6.1.2 Instalasi

Urutan proses instalasi yang harus dilakukan adalah sebagai berikut :

- a. Instal sistem operasi pada Mesin Head Node.
- b. Lakukan instalasi sampai tahap Instalasi DNS Server.
- c. Lakukan instalasi sampai tahap Persiapan Storage dan Instalasi NFS Server.
- d. Lanjutkan instalasi Mesin Head Node.
- e. Lanjutkan instalasi Mesin Storage.
- f. Lakukan instalasi Mesin Worker.

### 3.6.2 Mesin Data Storage

Berikut ini adalah daftar aplikasi yang akan diinstal pada Mesin Data Storage.

- a. Sistem operasi Debian GNU/Linux 3.1r2 "Sarge".
- b. NFS Server.
- c. OpenSSH.
- d. Pustaka libnss-ldap dan libpam-ldap.

Berkas-berkas aplikasi SGE akan diletakkan pada mesin ini yang akan diakses oleh seluruh mesin lainnya di dalam cluster.

### 3.6.3 Mesin Head Node

Berikut ini adalah daftar aplikasi yang akan diinstal pada Mesin Head Node:

- a. Sistem operasi Debian GNU/Linux 3.1r2 "Sarge"
- b. inGrid Portal (SGE Appliance) yang berisi Globus Toolkit dan SGE Adapter.
- c. Sun Grid Engine (Master Host dan Submit Host).
- d. OpenLDAP.
- e. OpenSSH.
- f. Pustaka libnss-ldap dan libpam-ldap.
- g. NFS Client.
- h. DNS Server

Aplikasi-aplikasi cluster yang akan dipergunakan pada pengembangan berikutnya.

Beberapa patch maupun eksekusi script tambahan diperlukan agar konfigurasi default dari sistem yang terinstal dapat bekerja dengan baik. Berkas-berkas patch maupun script dapat dilihat <http://grid.ui.ac.id/ingrid/clusters/hastinapura/files/>.

Selain melakukan instalasi dan konfigurasi aplikasi, Mesin Head Node ini juga berfungsi sebagai gateway ke mesin kambing.vlsm.org dan haur.ui.edu. Oleh karena itu, konfigurasi NAT dan IP Forwarding juga harus dilakukan. Pada mesin ini, terdapat dua buah Network Interface Card (NIC). Berikut ini adalah susunan penggunaannya.

1. NIC pertama (eth0) dihubungkan dengan segmen Inherent.
2. NIC kedua (eth1) dihubungkan dengan Switch Koneksi.

### 3.6.4 Worker Node

Berikut ini adalah daftar aplikasi yang akan diinstall pada mesin Worker.

- a. Sistem operasi Debian GNU/Linux 3.1r2 "Sarge".
- b. NFS Client.
- c. Sun Grid Engine (Execute Host).
- d. Pustaka libnss-ldap dan libpam-ldap.
- e. MPICH.

### 3.6.5 Portal

Berikut ini adalah daftar aplikasi yang akan diinstall pada Mesin Portal:

- a. Sistem operasi Debian GNU/Linux 3.1r2 "Sarge".  
inGRID Portal (Portal) yang berisi Globus Toolkit,  
Apache Tomcat, UCLA Grid Portlets,
- b. SimpleCA, dan GridFTP.
- c. inGRID Grid Portlets.
- d. OpenSSH.
- e. Apache2.
- f. MySQL Server.
- g. Modul apache2 jk dan php-mysql.
- h. Instalasi Drupal dan modulnya untuk website GCRG.

Penjelasan detail mengenai langkah – langkah penginstalan akan dijelaskan lebih lagi pada lampiran

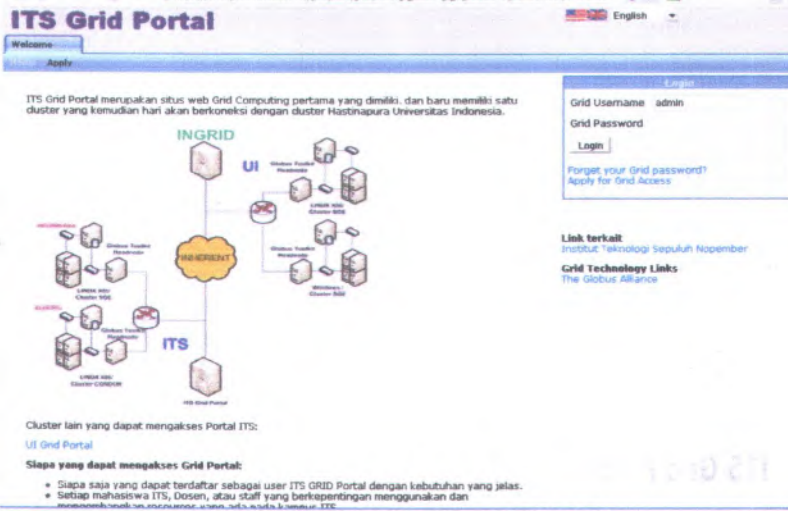


## BAB 4

### UJI COBA DAN EVALUASI

Pada bab ini akan dibahas tentang implementasi dari perancangan infrastruktur yang telah dilakukan pada bab 3.

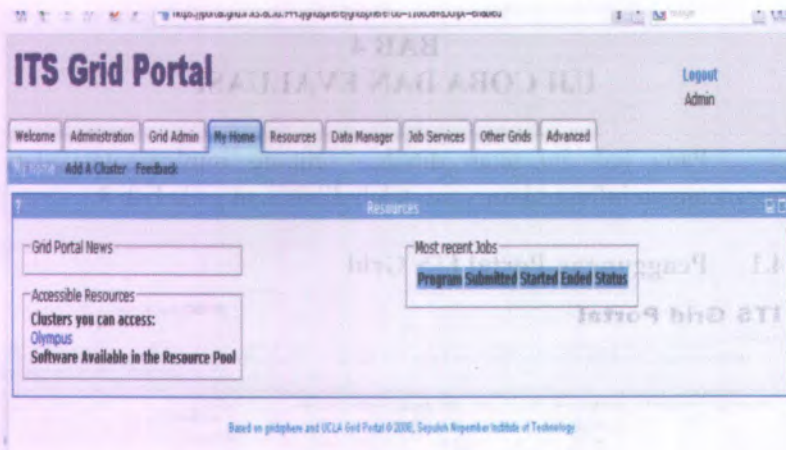
#### 4.1 Penggunaan Portal ITS Grid



**Gambar 4.1 Tampilan Portal ITS Grid**

ITS GRID Portal dapat diakses melalui URL <https://portal.grid.if.its.ac.id:9443/gridsphere>. Setelah Kita memasuki halaman tersebut, Kita akan menjumpai tampilan seperti pada gambar 4.1.

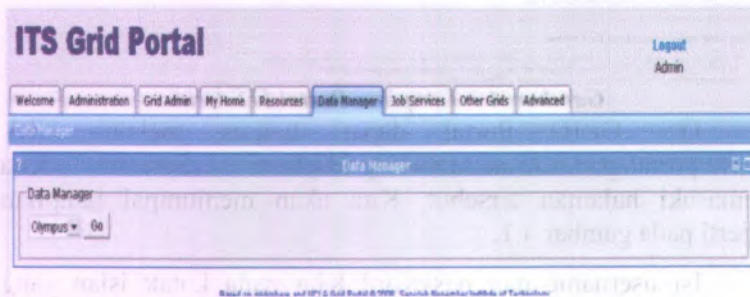
Isi username dan password Kita pada kotak isian yang disediakan. Setelah itu klik tombol Login. Kita akan menjumpai tampilan seperti pada Gambar 4.1.1. Pada tampilan ini, Kita dapat melihat cluster-cluster yang dapat Kita akses pada kotak berjudul Accessible Resources. Pada bagian atas, Kita akan menjumpai menu-menu utama, yaitu My Home, Resources, Data Manager, Job Services, Other Grids, dan Advanced.



**Gambar 4.2.2** Tampilan setelah login

## 4.2 Data Manager

Pada halaman ini pengguna diharuskan memilih data manager. Fungsi utama dari data manager adalah memilih cluster yang dipakai untuk mengerjakan jobs yang akan diminta oleh user



**Gambar 4.2** Tampilan Data Manager

### 4.3 Mengirim Job Generik

The screenshot shows a window titled "Generic Job Submission" within a "Job Services" application. The window contains the following sections and fields:

- Submit to:** A dropdown menu with "Olympus" selected.
- Job To Submit:** A section with the instruction "Required entries have bold labels."
  - Job Name:** An empty text input field.
  - Executable:** An empty text input field with a help icon and the text "Just a name you give this job so you can recognize it later. The file name of your executable required."
  - Arguments:** An empty text input field.
  - Directory:** A section with instructions: "Directory is the directory in which your job will run. Every filename used in your job which is not specified as an absolute path will be relative to this directory." It includes sub-points: "If Directory is omitted, your home directory will be used by default." and "If Directory is specified but does not include an absolute path, it will be relative to your home directory." Below this is an empty text input field.
  - JobType:** A dropdown menu with "Serial" selected.
- Environment Variables:** An empty text area.
- Stdin:** A dropdown menu with "JobInput" selected.
- Job Requirements:** A section with instructions: "For rental jobs, the number of processors must be one. Some schedulers terminate jobs that have reached their maximum CPU time, others use the maximum elapsed time."
  - Number of Processors:** An input field with the value "1".
  - Maximum Memory(MB):** An input field with the value "480".
  - Maximum Time (in minutes):** An input field with the value "60".
  - Queue Name (Optional):** An empty text input field.
- Save as Your Application:** A section with the text "Are you submitting the same job over and over again? Tired of filling in this form? Save your filled in form as an application. Then you can select and submit it from 'User Applications'." Below this is an empty text input field.

**Gambar 4.3 Tampilan Job Generik**

Pada menu Job Services, Kita dapat menemukan tampilan untuk melakukan pengiriman job yang akan mengeksekusi aplikasi yang Kita miliki. Selain itu Kita juga dapat melihat status dari setiap job yang pernah Kita kirimkan. Untuk mengirim job dengan konfigurasi rancangan Kita sendiri, setelah memilih menu Job Services, Kita perlu mengklik submenu Generic Jobs. Pada halaman tersebut, Kita dapat menentukan sendiri aplikasi yang akan dieksekusi, parameter, jenis job, jumlah prosesor, dan konfigurasi lainnya. Berikut ini adalah penjelasan dari setiap kotak isian yang tersedia.

#### a. Submit to

Berisi nama cluster yang ingin kita akses Cluster yang dipilih akan menerima job yang Kita kirim. Aplikasi akan berjalan pada mesin-mesin yang terdapat di dalam cluster tersebut.

**b. Job Name**

Berisi nama dari job yang ingin kita buat. Buatlah nama yang jelas yang mencerminkan job yang Kita buat sehingga Kita dapat mengenali job tersebut dengan mudah suatu saat nanti.

**c. Executable**

Berisi nama berkas aplikasi yang ingin Kita jalankan. Aplikasi ini harus berada pada cluster yang ingin Kita akses. Kita dapat memasukkan aplikasi melalui fasilitas manajemen data. Jika Kita ingin melihat dan memilih berkas aplikasi secara visual, Kita dapat mengklik ikon Folder di sebelah kanan kotak isian. Sebuah window akan muncul dan berisi halaman yang menampilkan daftar berkas yang berada pada cluster yang bersangkutan

**d. Arguments**

Berisi parameter dari aplikasi yang ingin Kita jalankan.

**e. Directory**

Berisi direktori tempat job Kita dijalankan. Jika Kita mengosongkan isian direktori ini, maka job Kita akan dijalankan pada home directory milik Kita pada cluster yang Kita pilih. Pada umumnya, Kita cukup mengosongkan isian ini agar job Kita berjalan pada home directory.

**f. JobType**

Berisi jenis dari aplikasi yang Kita jalankan. Pada saat ini terdapat dua buah jenis aplikasi yang dapat dijalankan, yaitu Serial dan MPI Paralel. Pemilihan Serial mengakibatkan aplikasi Kita akan dijalankan seperti biasa ketika Kita menjalankan aplikasi tersebut di komputer Kita. Sedangkan MPI Paralel perlu dipilih jika aplikasi yang Kita buat merupakan sebuah aplikasi parallel berbasis MPI. Aplikasi Kita akan dijalankan dengan bantuan aplikasi mpirun yang memang digunakan untuk menjalankan aplikasi berbasis MPI.

**g. Environment Variables**

Berisi daftar environment variables yang perlu diset pada lingkungan eksekusi aplikasi Kita.

**h. Number of Processors**

Berisi jumlah prosesor yang akan dialokasikan dan digunakan untuk mengeksekusi job yang Kita kirimkan.

**i. Maximum Memory (MB)**

Berisi jumlah memory maksimum yang diperbolehkan untuk digunakan oleh aplikasi.

**j. Maximum Time (in minutes)**

Berisi jumlah waktu eksekusi maksimum aplikasi yang akan dijalankan.

**k. Queue Name (Optional)**

Berisi nama antrian yang akan digunakan untuk menjalankan aplikasi.

Setelah Kita selesai mengatur konfigurasi job yang ingin Kita kirim, Kita dapat mengklik tombol Submit Job untuk mengirim job tersebut ke cluster yang telah Kita pilih.

#### 4.4 Melihat status job

Status dari seluruh job yang pernah Kita kirim dapat Kita lihat statusnya, apakah job tersebut masih dieksekusi, sudah selesai di eksekusi, gagal dieksekusi, dan sebagainya. Untuk melihat status ini, setelah Kita memilih menu Job Services, Kita perlu mengklik submenu Job Status.

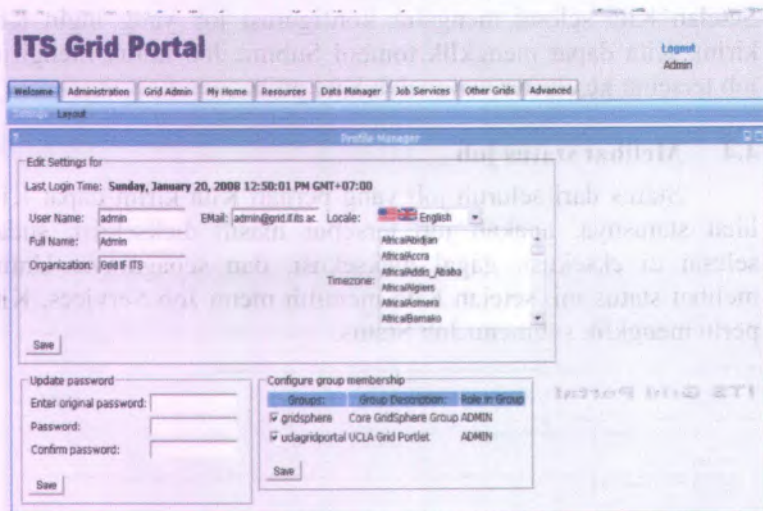


**Gambar 4.4 Tampilan status Job**

Keluaran dari aplikasi berupa stkitard output dan stkitard error dapat Kita peroleh melalui halaman ini juga. Kita cukup mengklik tulisan Stdout atau Stderr dari salah satu job untuk melihat stkitard output atau stkitard error dari job tersebut. Jika Kita perlu mendownload stkitard output maupun stkitard error menjadi sebuah berkas, klik saja ikon yang berada di sebelah tulisan Stdout atau Stderr. Jika Kita ingin menghapus status dari job yang ada pada daftar, klik saja ikon yang berada pada kolom Purge dari job yang ingin Kita hapus. Job tersebut akan segera terhapus dari daftar

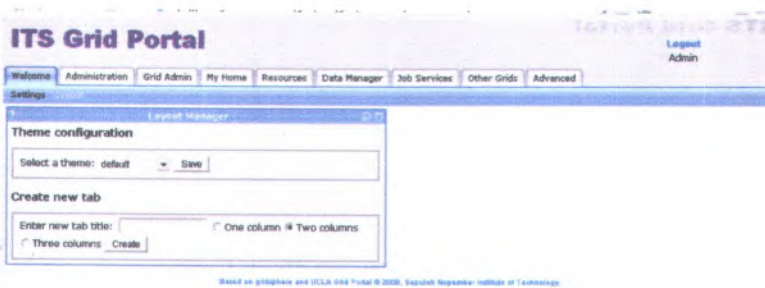
## 4.5 Menu lainnya

### 4.5.1 Settings dan layout pada tab Welcome



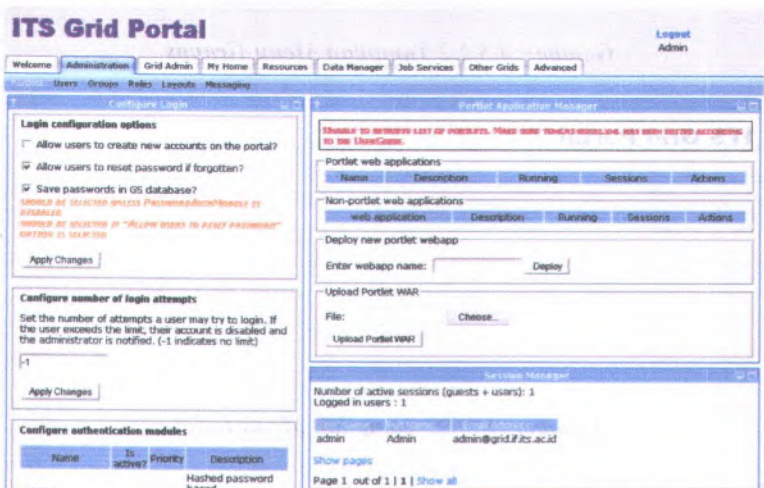
Based on gridsphere and UCLA Grid Portal ©2008, Sepuluh Nopember Institute of Technology

**Gambar 4.5.1** Tampilan Menu settings



Gambar 4.5.1.1 Tampilan layout

## 4.5.2 Portlets, Users, Groups, Roles, Layout dan messaging pada tab Administration



Gambar 4.5.2 Tampilan Menu portlets

**ITS Grid Portal** Logout Admin

Welcome Administration Grid Admin My Home Resources Data Manager Job Services Other Grids Advanced

Portlets Users Groups Roles Layouts Messaging

User Account Manager

User WAS SUCCESSFULLY DELETED

Display All Users

ID	Admin	admin@grid.itso.ac.id	Grid IT
admin	Admin	admin@grid.itso.ac.id	Grid IT

Show pages

Page 1 out of 1 | 1 | Show all

Create a New User

Based on gridSphere and UCLA Grid Portal © 2008, Republik Indonesia Institute of Technology

**Gambar 4.5.2.1 Tampilan Menu Users**

**ITS Grid Portal** Logout Admin

Welcome Administration Grid Admin My Home Resources Data Manager Job Services Other Grids Advanced

Portlets Users Groups Roles Layouts Messaging

Portlet Group Manager

Group Name	Manage Portlets	Edit Users	Group Description	Disable Groups
gridSphere	Select portlets	Edit Users	Core GridSphere Group	
udagridportal	Select portlets	Edit Users	UCLA Grid Portal	Details

Create new group Edit default groups

Based on gridSphere and UCLA Grid Portal © 2008, Republik Indonesia Institute of Technology

**Gambar 4.5.2.2 Tampilan Menu Groups**

**ITS Grid Portal** Logout Admin

Welcome Administration Grid Admin My Home Resources Data Manager Job Services Other Grids Advanced

Portlets Users Groups Roles Layouts Messaging

Role Manager

Display All Roles

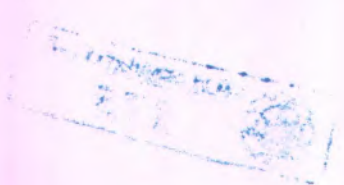
Role Name	Role Name	Role Name
GUEST	Guest	
USER	User	
ADMIN	Admin	
SUPER	Super	

Page 1 out of 1 | 1 | Show all

Create New Role

Based on gridSphere and UCLA Grid Portal © 2008, Republik Indonesia Institute of Technology

**Gambar 4.5.2.3 Tampilan Menu Roles**





**ITS Grid Portal** Logout Admin

Welcome Administration **Grid Admin** My Home Resources Data Manager Job Services Other Grids Advanced

Portlets: Users Groups Roles Messaging

Layout Manager

**Edit Banner**  
Below you can edit the portal banner HTML.  
 <div id="page-Logo">  
 <!-- Banner -->  
 <font size="6" color="#22003a" face="Arial Black">

**Edit Footer**  
Below you can edit the portal footer HTML.  
 <div style="color: #333333;>  
 font-family: lucida, helvetica, serif; font-size: 10px;  
 padding-left: 3px; padding-top: 5px; padding-bottom: 5px

Save

**Choose Default Theme**  
Select from the list of available themes: metal Save

**Guest Layout**  
Edit guest layout

**Group Layouts**  
The following group layouts exist:

Group Name	Group Description	Edit layout	Delete layout
gridosphere	Core GridSphere Group	Edit	
uclagridportal	UCLA Grid Portal	Edit	Delete

Based on gridosphere and UCLA Grid Portal © 2006, Sapuh Nugent, Institute of Technology

Gambar 4.5.2.4 Tampilan Menu Layout

**ITS Grid Portal** Logout Admin

Welcome Administration **Grid Admin** My Home Resources Data Manager Job Services Other Grids Advanced

Portlets: Users Groups Roles Layouts

Messaging Services

The following messaging services are available:

Service to connect to Email		
mailserver		olympus.grid.fls.ac
mailport		25
mailaddress		portal@grid.fls.ac

Save and activate changed services

Based on gridosphere and UCLA Grid Portal © 2006, Sapuh Nugent, Institute of Technology

Gambar 4.5.2.5 Tampilan Menu Messaging

#### 4.5.3 Trusted Cluster list, Cluster access list, Application, Application pool, Pools admin, User admin, New release, email users pada tab Grid Admin

**ITS Grid Portal** Logout Admin

Welcome Administration **Grid Admin** My Home Resources Data Manager Job Services Other Grids Advanced

ClusterAccessList Application Application Pool Pools Admin User Admin New Release Email Users

TrustedCluster Table Manager

olympus.grid.fls.ac id =

Cluster Name:	True Name:	Default Scheduler:	Cluster Head Node:	Support Email:

Load

Add

Delete

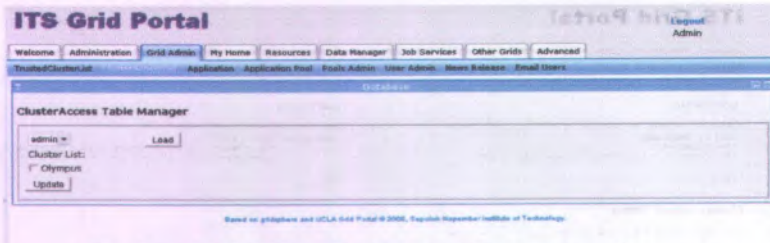
Save

Clear

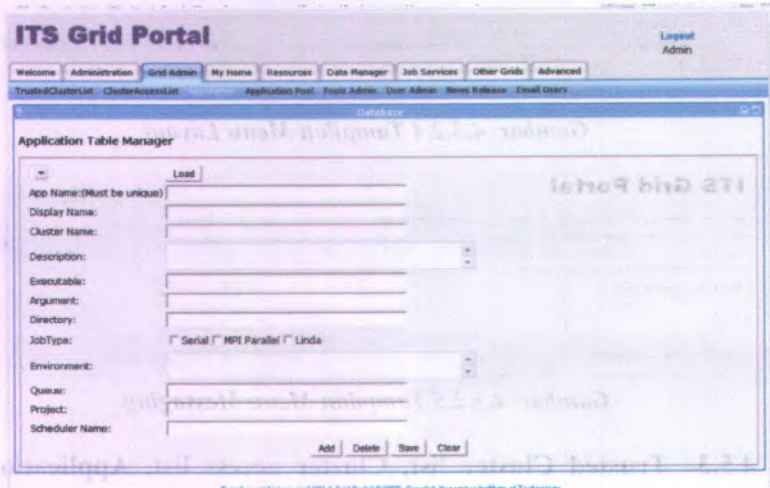
Based on gridosphere and UCLA Grid Portal © 2006, Sapuh Nugent, Institute of Technology

Gambar 4.5.3.1 Tampilan Menu Trusted cluster list

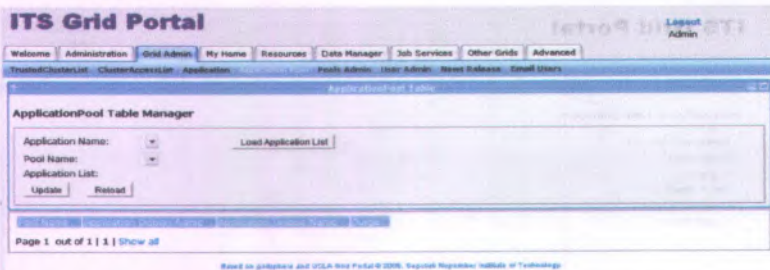




**Gambar 4.5.3.2 Tampilan Menu Cluster access list**



**Gambar 4.5.3.3 Tampilan Menu Application**



**Gambar 4.5.3.4 Tampilan Menu Application pool**

The screenshot shows the ITS Grid Portal interface. At the top, there is a navigation menu with items: Welcome, Administration, Grid Admin, My Home, Resources, Data Manager, Job Services, Other Grids, and Advanced. Below this is a sub-menu for Grid Admin containing: TrustedClusterList, ClusterAccessList, Application, Application Pool, PoolInfo Admin, User Admin, News Release, and Email Users. The main content area is titled 'PoolInfo Table Manager' and contains a form with the following fields: Pool Name (Must be unique), Contact Name, and EmailAddr. There are 'Add', 'Delete', 'Save', and 'Clear' buttons at the bottom of the form. A small copyright notice at the bottom reads: 'Based on gridshare and UCLA Grid Portal © 2006, Sepuluh September Institute of Technology'.

*Gambar 4.5.3.5 Tampilan Menu Application Pools Admin*

The screenshot shows the ITS Grid Portal interface with the 'User Admin' menu selected. The sub-menu includes: TrustedClusterList, ClusterAccessList, Application, Application Pool, PoolInfo Admin, User Admin, News Release, and Email Users. The main content area is titled 'Display All Users' and shows a table with the following data:
 

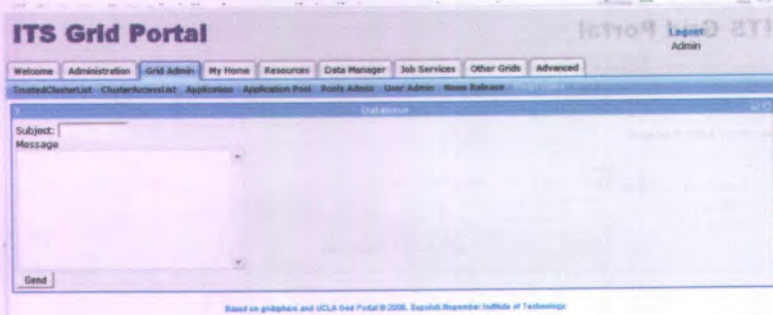
Username	Full Name	Group Access	Cluster Access	Cluster Admin
admin	Admin	admin@grid.its.ac.id	Grid IF	ITS

 Below the table, it indicates 'Page 1 out of 1 | Show all' and a 'Create a New User' link. A small copyright notice at the bottom reads: 'Based on gridshare and UCLA Grid Portal © 2006, Sepuluh September Institute of Technology'.

*Gambar 4.5.3.6 Tampilan Menu User Admin*

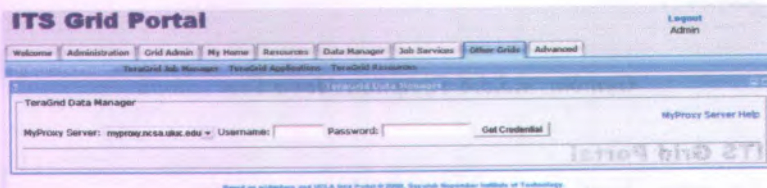
The screenshot shows the ITS Grid Portal interface with the 'News Release' menu selected. The sub-menu includes: TrustedClusterList, ClusterAccessList, Application, Application Pool, PoolInfo Admin, User Admin, News Release, and Email Users. The main content area is titled 'News Release' and contains a form with the following fields: Title, Owner, Date, and Content. There are 'Add', 'Save', and 'Clear' buttons at the bottom of the form. A small copyright notice at the bottom reads: 'Based on gridshare and UCLA Grid Portal © 2006, Sepuluh September Institute of Technology'.

*Gambar 4.5.3.7 Tampilan Menu Cluster News Release*

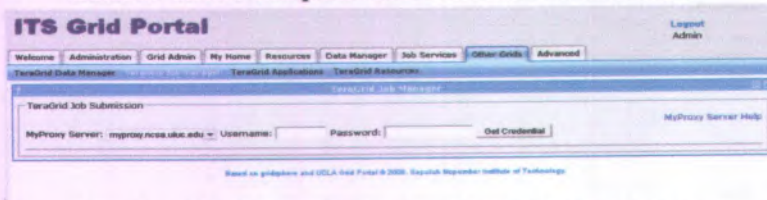


**Gambar 4.5.3.8 Tampilan Menu email users**

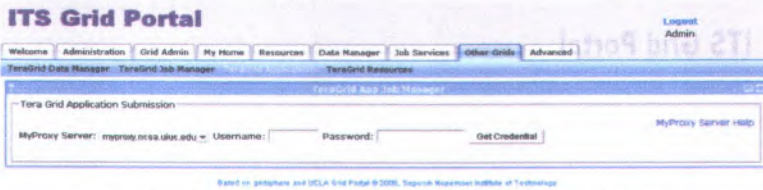
#### 4.5.4 TeraGrid data manager, TeraGrid job manager, TeraGrid Application, TeraGrid Resources pada tab Other Grids



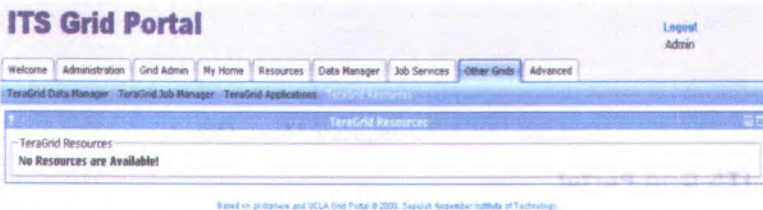
**Gambar 4.5.4.1 Tampilan Menu TeraGrid data manager**



**Gambar 4.5.4.2 Tampilan Menu TeraGrid job manager**

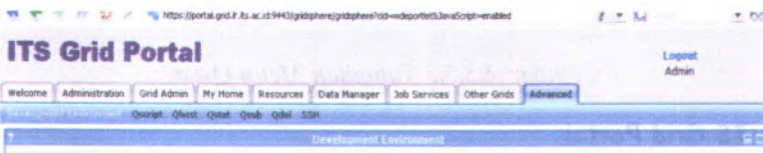


**Gambar 4.5.4.3 Tampilan Menu TeraGrid application**

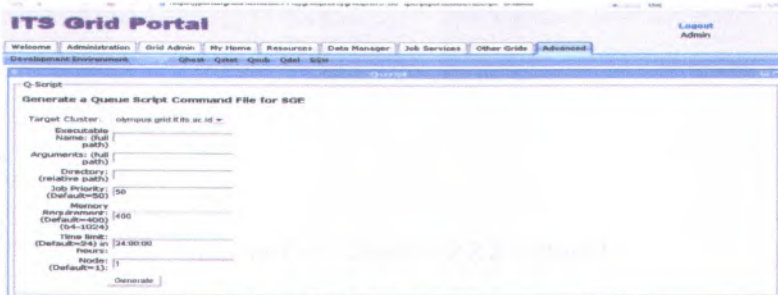


**Gambar 4.5.4.4 Tampilan Menu TeraGrid Resources**

## 4.5.5 Development environment, Qscript, Qhost, Qstat, Qsub, Qdel, SSH pada tab advanced



**Gambar 4.5.5.1 Tampilan Menu Development Environment**



**Gambar 4.5.5.2 Tampilan Menu Qscript**

ITS Grid Portal Logout  
Admin

Welcome Administration Grid Admin My Home Resources Data Manager Job Services Other Grids **Advanced**

Development Environment Qscript Qhost Qsub Qstat SSH

Qhost

Qhost Command

Qhost Command

Target Cluster: olympus.grid.its.ac.id

Show Qhost Result

Based on gridshare and UCLA Grid Portal © 2005, Sepuluh Nopember Institute of Technology

**Gambar 4.5.5.3 Tampilan Menu Qhost**

ITS Grid Portal Logout  
Admin

Welcome Administration Grid Admin My Home Resources Data Manager Job Services Other Grids **Advanced**

Development Environment Qscript Qhost Qsub Qstat SSH

Qstat

Qstat Command

Qstat Command

Target Cluster: olympus.grid.its.ac.id

User ID:

Show Qstat Results

Job Details

Job ID:

Show Job Details

Based on gridshare and UCLA Grid Portal © 2005, Sepuluh Nopember Institute of Technology

**Gambar 4.5.5.3 Tampilan Menu Qstat**

ITS Grid Portal Logout  
Admin

Welcome Administration Grid Admin My Home Resources Data Manager Job Services Other Grids **Advanced**

Development Environment Qscript Qhost Qsub Qstat SSH

Qsub

Qsub Command

Qsub Command

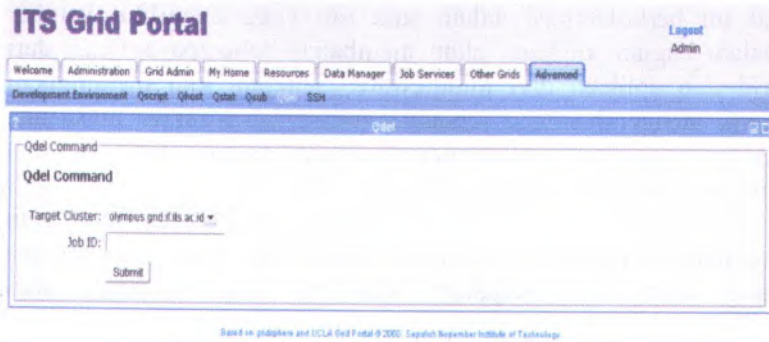
Target Cluster: olympus.grid.its.ac.id

Arguments:

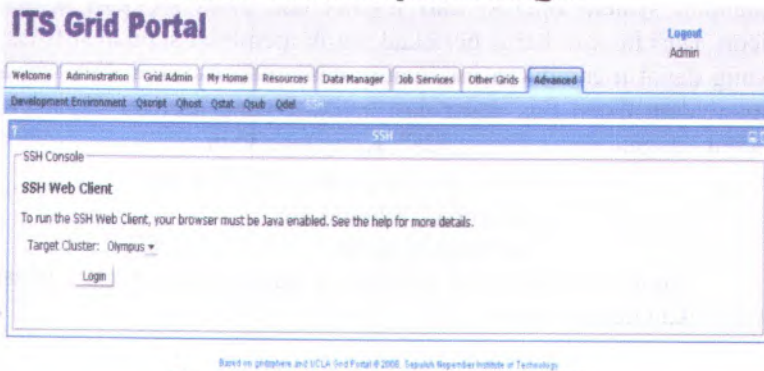
Submit

Based on gridshare and UCLA Grid Portal © 2005, Sepuluh Nopember Institute of Technology

**Gambar 4.5.5.4 Tampilan Menu Qsub**



**Gambar 4.5.5.5 Tampilan Menu Qdel**



**Gambar 4.5.5.6 Tampilan Menu SSH**

## 4.6 Evaluasi

Grid dapat menyajikan sebuah teknologi yang dapat dipilih untuk membuat sebuah aplikasi yang besar untuk penelitian, bisnis, hiburan, kesehatan dan berbagai raung lingkup lainnya. Bagaimanapun sebuah komunitas mempunyai permasalahan yang umum yaitu mengatur dari teknologi yang baru yaitu sebuah aplikasi membutuhkan untuk mengendalikan sebuah penelitian dan mengembangkan teknologi yang baru tetapi sebuah aplikasi sangat sulit untuk dikembangkan. Dalam komunitas grid, infrastruktur dari grid, pengembang aplikasi dan usaha middleware dapat melakukan proses secara bersama-sama dalam

hal ini berkolaborasi dalam satu tim yang memiliki disiplin. Dalam bagian ini kami akan membahas beberapa aplikasi dari grid dan aplikasi dari middleware. Sebelum kita melanjutkan untuk mengembangkan sebuah software infrastruktur maka ada baiknya bagi kita untuk merealisasikan potensi dari grid dan infrastruktur dari grid secara umum.

Aplikasi merupakan kata kunci menuju ke Grid, dan contohnya memberikan sebuah gambaran dari kesuksesan dan apa yang akan kita kerjakan hari ini. Tujuan utama dari pengembangan sebuah Grid adalah untuk mengembangkan sebuah aplikasi. Tetapi sebuah harapan ditujukan yaitu Grid dapat menjadi system operasi dari internet dan akan menjadi sebuah icon. Hari ini kita harus bertekad untuk membuat sebuah software yang dapat menjadikan ini semua menjadi kenyataan. Kita dapat mengidentifikasi tiga dasar dari aplikasi yang hari ini dapat kita sebut dengan "natural for grids". Ketiganya yaitu :

- a. Minimal communication application
- b. Staged/linked application
- c. Access to resources

Adapun yang dapat menjadi sebuah aplikasi Grid dimasa yang akan datang yaitu :

- a. Adaptive applications
- b. Real time and on demand applications
- c. Coordinated application
- d. Poly applications



## BAB 5

### KESIMPULAN, HAMBATAN DAN SARAN

Pada bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari tujuan pembangunan infrastruktur ITS Grid, Selain itu terdapat beberapa kendala yang dihadapi penulis dan juga saran untuk pengembangan lebih lanjut.

#### 5.1 KESIMPULAN

Dari permasalahan yang telah dibahas panjang lebar diatas maka ada beberapa kesimpulan yang dapat ditarik :

- a. Portal ITS Grid bisa diakses dari lingkungan ITS, dan kelak ada beberapa aplikasi yang bisa menampung dan menjawab setiap jobs yang diterima dari user.
- b. Didalam menu *Job Services* → *Application* pada portal ITS Grid, terdapat *Application Submission* yang dapat diisi aplikasi – aplikasi MPI yang berjalan pada infrastruktur Grid ITS.



**Gambar 5.1.b Tampilan Menu Application Submission**

- c. Didalam portal ITS Grid seluruh job yang pernah Kita kirim dapat Kita lihat statusnya, apakah job tersebut masih dieksekusi, sudah selesai di eksekusi, gagal dieksekusi, dan sebagainya. Hal ini berdasar pada *menu status job*.

## 5.2 HAMBATAN

Setelah penulis melakukan implementasi dan evaluasi terhadap infrastruktur ITS Grid, penulis menemui beberapa hambatan dan kendala yang dihadapi yaitu :

Selama penulis melakukan instalasi terhadap mesin Headnode, Storage, Portal dan Worker, kendala yang dihadapi adalah :

- a. Pada saat instalasi *openLDAP* yang berfungsi sebagai direktori, penulis kesulitan menguji apakah fungsi *openLDAP* tersebut sudah berhasil dijalankan (*run*) dalam mesin headnode tersebut.
- b. Penulis mengalami kesulitan untuk mengintegrasikan konektivitas antara mesin *headnode* dan *worker* ketika memerlukan akses file yang menggunakan spasi bersama secara handal. Dimana prinsip *NFS* menggunakan salah satu storage pada server yang diperbolehkan untuk dimount oleh komputer-komputer yang lain seolah-olah storage tersebut berada pada disk lokal di masing-masing komputer yang memount-nya.
- c. Penulis juga mengalami kendala melakukan integrasi antara *Globus toolkit* dan *SGE (sun grid engine)*.
- d. Kesulitan pengaktifan *gridsphere* ketika portal aktif juga dialami penulis.

## 5.3 SARAN

Setelah melakukan pembahasan diatas ada beberapa saran yang penulis ingin berikan, secara umum penulns menyarankan agar kedepan nanti, pengembangan infrastruktur ini di dukung dengan peralatan yang lebih baik lagi secara khusus :

- a. Mesin Headnode
  - a. Mesin headnode sebaiknya memiliki standar sebagai mesin server, karena komunikasi antar *Globus (management tools)* adalah melalui mesin ini

- b. Didalam mesin Headnode terdapat SGE (*sun grid engine*) yang bertugas untuk membagi *jobs* yang diterima dari user melalui Globus
- b. Mesin Storage  
Pada mesin storage ini, harus memiliki alokasi harddisk yang besar untuk menyimpan data komunikasi antara headnode dengan worker.
- c. Mesin Portal  
Mesin portal harus memiliki alokasi *RAM* dan *processor* yang besar, karena melalui mesin portal inilah user dapat mengakses infrastruktur ITS Grid.
- d. Mesin Worker  
Mesin worker sebaiknya mempunyai kemampuan yang hampir sama dengan mesin portal, karena mesin worker banyak melakukan aktivitas *running program*
- e. Dalam melaksanakan sebuah pembangunan infrastruktur Grid Computing, harus mempersiapkan beberapa hal yang perlu diperhatikan, meliputi :
  - a. Pemilihan Network Adapter (Kartu Ethernet).  
Rata-rata, *network adapter* masa kini sudah mendukung kecepatan 100 Mbps dan harganya tidak terlalu mahal. Gunakanlah pada seluruh komputer yang terintegrasi dalam sistem *NFS* ini. Jika pemakaian *NFS* dimaksudkan hanya terjadi dalam *subnet* internal, gunakanlah *network adapter* yang terpisah dengan *subnet* eksternal (internet). Ini akan membuat *load* jaringan pada masing-masing *subnet* akan menjadi lebih ringan.
  - b. Pemakaian Switch atau Router yang tepat. Pemakaian *switch* bila dibandingkan dengan pemakaian *hub* akan membantu kinerja jaringan secara keseluruhan. Gunakanlah *switch* yang mempunyai *speed* yang sama dengan *network adapter*, yaitu minimal 100 Mbps. Bila dimungkinkan, aturlah MTU (Maximum Transfer Unit) pada *router* atau *switch* mendekati nilai umum pada *wsize / rsize* Perintah `tracpath nmaserver/ 2049` dari komputer klien

akan menunjukkan nilai MTU yang berlaku saat ini. optimal yang akan diuraikan di bawah ini (pada poin 2).

c. Meminimalkan/menghindari *collision* (tumbukan).

Beberapa *network adapter* datang dengan setting Auto (Autonegotiation) untuk memilih kecepatan antara 10 dengan 100 Mbps, serta *full duplex* dengan *half duplex*. Jika anda sudah mengetahui seluruh infrastruktur jaringan harus dibangun dengan kecepatan 100 Mbps \*dan\* full-duplex, lakukan setting secara *strict* pada *network adapter*. Banyak kasus terjadi, kinerja NFS menjadi amat sangat lambat bahkan putus hanya karena banyaknya *collision* pada jaringan. *Collision* dapat dicek dengan perintah **ifconfig** dengan memperhatikan baris *collision*.

## LAMPIRAN

### Panduan Instalasi Portal inGRID :

Berikut ini adalah daftar aplikasi yang akan diinstall pada Mesin Portal :

1. Sistem operasi Debian GNU/Linux 3.1r2 "Sarge".
2. inGRID Portal (Portal) yang berisi Globus Toolkit, Apache Tomcat, UCLA Grid Port-lets, SimpleCA, dan GridFTP.
3. inGRID Grid Portlets.

Instal paket yang dibutuhkan:

- a. `# apt-get install sudo xinetd # apt-get install zlib-bin zlib1g  
zlib1g-dev # apt-get install gcc g++ g77 manpages-dev make #  
apt-get install zip unzip`
- b. Persiapkan database MySQL.
  - i. `# mysql -u root -p mysql> create database gridsphere mysql>  
grant all privileges on gridsphere.*  
to gridportal@localhost identified by 'password-  
gridportal';`
  - ii. `mysql> grant all privileges on gridsphere.* to  
gridportal@localhost.localdomain identified by  
'password-gridportal';`
- c. Buat user globus
  - i. `# adduser globus`
- d. Download berkas instalasi dari URL berikut.  
`http://grid.ui.ac.id/files/inGRID-Installer/inGRID-Installer 0.1-  
debian.tar.gz`
- e. Extract berkas instalasi  
`$ tar xzf inGRID-Installer_0.1-debian.tar.gz`
- f. Edit berkas `build.properties`, ubah nilai variabel-variabel berikut  
(a) `uc.portal.hostname` menjadi alamat dari portal, yaitu  
`portal.grid.if.its.ac.id`
- g. Edit berkas `portal.properties`, ubah data-data berikut.
  - i. `db.username` berisi `gridportal`.
  - ii. `db.password` berisi `password` untuk `gridportal`

- iii. mail.server berisi mail.its.ac.id
  - iv. mail.port berisi 25
  - v. mail.address berisi [portal@grid.if.its.ac.id](mailto:portal@grid.if.its.ac.id)
  - vi. uc.distinct.name berisi "OU=IF, OU=GRID, O=Institut Teknologi Sepuluh Nopember, C=ID"
  - vii. uc.ca.subject berisi "ou=IF, ou=GRID,o= Institut Teknologi Sepuluh Nopember,c=ID"
  - viii. uc.ca.email berisi [portal@grid.if.its.ac.id](mailto:portal@grid.if.its.ac.id)
  - ix. uc.ca.pass berisi password untuk Certificate Authority
- h. Jalankan script instalasi  
\$ ./install.sh portal
  - i. Jalankan script post-instalasi sebagai user root  
# cd ~/globus/inGRID-Installer/3rdParty/ # ./postinstall.sh portal
  - j. Tandatangani sertifikat mesin oleh Certificate Authority
  - k. Patch script startup Apache Tomcat agar memasukkan nilai \$GLOBUS LOCATION dan mengalokasikan memori yang lebih besar  
\$ cd /home/globus/UCPortal/apache-tomcat-5.5.17/bin \$  
wget http://grid.ui.ac.id/files/inGRID-Installer/catalina.sh.patch \$ patch < catalina.sh.patch
  - l. Restart xinetd  
# /etc/init.d/xinetd restart
  - m. Hidupkan Globus Toolkit  
\$ cd /home/globus/UCPortal/GT4 \$ ./globus-start.sh \$ cd  
etc/globus\_wsrf\_mds\_index/DefaultIndex \$ ./srun.sh
  - n. Hidupkan Apache Tomcat  
\$ cd /home/globus/UCPortal/apache-tomcat-5.5.17/bin \$  
./startup.sh
4. OpenSSH.
    - a. # apt-get install ssh
    - b. Tunggu sampai proses selesai dan aplikasi OpenSSH sudah dapat digunakan.
  5. Apache2.
    - a. Instal paket MySQL Server dan client.

- b. # apt-get install mysql-server mysql-client
  - c. # mysqladmin password 'password-root-yang-baru'
6. MySQL Server.
- a. Instalasi mySQL server dan client
    - i. Instal paket MySQL Server dan client.
    - ii. Ubah password root.
    - iii. # apt-get install mysql-server mysql-client
    - iv. # mysqladmin password 'password-root-yang-baru'
7. Modul apache2 jk dan php-mysql.
8. Instalasi Drupal dan modulnya untuk website GCRG.

## **Panduan Instalasi Cluster Informatika :**

### **1. Pendahuluan**

Panduan Instalasi ini akan terbagi menjadi tiga bagian, yaitu panduan instalasi bagi Mesin Head Node, Storage, Worker. Seluruh petunjuk instalasi yang tertulis pada dokumen ini perlu dilakukan secara berurutan. Jangan melakukan suatu tahapan instalasi sebelum tahapan sebelumnya berhasil dengan baik.

#### **1. Persiapan**

Setidaknya ada dua hal yang perlu disiapkan, yaitu perangkat lunak dan perangkat keras. Perangkat lunak yang digunakan setidaknya terdapat pada dua lokasi, yaitu

- a. Arsip perangkat lunak
- b. Repository Debian GNU/Linux

Dari sisi perangkat keras hal – hal yang perlu dipersiapkan adalah sebagai berikut :

- a. Mesin Head Node
- b. Mesin Data Storage
- c. Mesin Worker Node (sebanyak 4 buah)
- d. Switch Koneksi (24 port Gigabit Ethernet switch)
- e. Switch Data (24 port Gigabit Ethernet switch)

#### **2. Instalasi**

Urutan proses instalasi yang harus dilakukan adalah sebagai berikut:

- a. Instal sistem operasi pada Mesin Head Node.
- b. Lakukan instalasi sampai tahap Instalasi DNS Server.
- c. Instal sistem operasi pada Mesin Storage
- d. Lakukan instalasi sampai tahap Persiapan Storage dan Instalasi NFS Server
- e. Lanjutkan instalasi Mesin Head Node
- f. Lanjutkan instalasi Mesin Storage
- g. Lakukan instalasi Mesin Worker



## 2. Mesin Data Storage

Berikut ini adalah daftar aplikasi yang akan diinstal pada Mesin Data Storage

- a. Sistem operasi Debian GNU/Linux 3.1r2 "Sarge".
- b. NFS Server
- c. OpenSSH
- d. Pustaka libnss-ldap dan libpam-ldap

### i. Instalasi Debian GNU/Linux

Setelah instalasi selesai, berikut ini adalah hal-hal yang perlu dilakukan sebagai root

1. Upgrade sistem yang ada dengan menjalankan perintah-perintah berikut

```
# apt-get update
# apt-get upgrade
```

2. Konfigurasi jaringan sebagai berikut

```
address 10.126.12.48
netmask 255.255.255.0
network 10.126.12.0
broadcast 10.126.12.255
gateway 10.126.12.43
dns-nameservers 10.126.12.43
dns-search grid.if.its.ac.id
```

### ii. Instalasi OpenSSH

```
# apt-get install ssh
```

Tunggu sampai proses selesai dan aplikasi OpenSSH sudah dapat digunakan.

### iii. Persiapan Berkas Sun Grid Engine

Berikut ini adalah tahapan yang perlu dilakukan

1. Buat user administrator Sun Grid Engine
2. Download Sun Grid Engine. Paket instalasi Sun Grid Engine dapat di download di URL berikut :
  - a. <http://grid.ui.ac.id/files/sge/n1ge6update4/n1ge-60u4-bin-linux24-i586.tar.gz>
  - b. <http://grid.ui.ac.id/files/sge/n1ge6update4/n1ge-60u4-common.tar.gz>

3. Buat dan persiapkan direktori instalasi Sun Grid Engine
  - a. `# mkdir /opt/sge`
  - b. `# chown sgeadmin:sgeadmin /opt/sge`
4. Extract paket instalasi Sun Grid Engine
  - a. `# cd /opt/sge`
  - b. `# tar xzf ~/n1ge-6_0u4-bin-linux24-i586.tar.gz`
  - c. `#tar xzf ~/n1ge-6_0u4-common.tar.gz`
  - iv. Persiapan Storage dan Instalasi NFS Server. Berikut ini adalah tahapan yang perlu dilakukan :
    1. Konfigurasi logical volume manager
      - a. `# pvcreate /dev/hdb1 /dev/hdc1 /dev/hdd1`
      - b. `# vgcreate grid_vg /dev/hdb1 /dev/hdc1 /dev/hdd1`
      - c. `# lvcreate gridlv`
      - d. `# mkfs.ext3 -c -c /dev/mapper/grid_vg-gridlv`
    2. Konfigurasi filesystem table. Tambahkan data berikut dalam berkas /etc/fstab
      - a. `/export /dev/mapper/grid_vg-gridlv ext3 defaults 0 0`
    3. Install aplikasi NFS Server.
 `# apt-get install nfs-server`
    4. Konfigurasi NFS Server. Ubah berkas /etc/exports dan isilah data berikut.
      - a. `/export`  
192.168.1.0/24(rw,async) 192.168.0.1/32(rw,async)
      - b. `/opt/sge`  
192.168.1.0/24(rw,async)  
192.168.0.1/32(rw,async)
  - v. Instalasi PAM LDAP
 

Berikut ini adalah langkah-langkah yang perlu dilakukan

    1. Install paket yang diperlukan
 `# apt-get install libpam-ldap libnss-ldap ldap-utils`
    2. Ikuti layar konfigurasi yang ditampilkan sesuai data berikut
      - a. LDAP Server host: 10.126.12.43

- b. The distinguished name of search base: ou=IF, ou=Grid, ou=olympus,o=Institut Teknologi Sepuluh Nopember,c=ID
  - c. LDAP version to use: 3
  - d. Database requires login: no
  - e. Make configuration readable/writable by owner only: no
  - f. Make local root database admin: no
  - g. Database requires logging in: no
  - h. Local crypt to use when changing password: crypt
3. Konfigurasi name service
- ```
# vi /etc/libnss-ldap.conf
base ou=Olympus ,ou=Grid,ou=IF,o=Institut
Teknologi Sepuluh Nopember,c=ID
uri ldap:// 10.126.12.43
ldap_version 3
scope sub
nss_base_passwd ou=Olympus
,ou=Grid,ou=IF,o=Institut Teknologi Sepuluh
Nopember,c=ID
nss_base_shadow ou=Olympus
,ou=Grid,ou=IF,o=Institut Teknologi Sepuluh
Nopember,c=ID
nss_base_group
ou=Olympus ,ou=Grid,ou=IF,o=Institut Teknologi
Sepuluh Nopember,c=ID
```
4. Konfigurasi konfigurasi LDAP
- ```
# vi /etc/ldap/ldap.conf
BASE ou=Hastinapura,ou=GCRG,o=Universitas
Indonesia,c=ID
URI ldap://10.126.12.43
```
5. Konfigurasi nsswitch
- ```
# vi /etc/nsswitchconf
passwd: files ldap compat
group: files ldap compat
```

- ```
shadow: files ldap compat
```
6. Konfigurasi PAM
 

```
# vi /etc/pam.d/common-account
account sufficient pam_ldap.so
account required pam_unix.so
# vi /etc/pam.d/common-auth
auth sufficient pam_ldap.so
auth required pam_unix.so nullok_secure
# vi /etc/pam.d/common-password
password sufficient pam_ldap.so
password required pam_unix.so nullok obscure min=4
max=8 md5
```
  3. **Mesin Mesin Head Node**
    - a. Instalasi Debian GNU/Linux
      - i. **eth0**

```
address 10.126.12.43
netmask 255.255.255.0
network 10.126.12.0
broadcast 10.126.12.255
up iptables -t nat -A POSTROUTING -s 10.126.12.32/27
-j MASQUERADE
down iptables -t nat -F
#p ip r add 10.0.0.0/8 via 10.126.12.1
#up ip r add 202.46.129.43 via 10.126.12.1
```
      - ii. **eth1**

```
address 202.46.129.91
netmask 255.255.255.240
network 202.46.129.80
broadcast 202.46.129.95
gateway 202.46.129.81
# dns-* options are implemented by the resolvconf
package, if installed
# dns-nameservers 202.46.129.2
# dns-search its.ac.id
```

- b. Instalasi OpenSSH
  - # apt-get install ssh
- c. Instalasi DNS Server
  - i. Instal aplikasi BIND
  - ii. Konfigurasi Zona. Tambahkan data berikut pada berkas /etc/bind/named.conf.local
    - ...
    - zone "portal.grid.if.its.ac.id" {
      - type master;
      - file "/etc/bind/zones/db.portal.grid.if.its.ac.id";
      - };
      - zone "10.192.in-addr.arpa" {
        - type master;
        - file "/etc/bind/zones/db.168.192";
        - };
    - iii. Buat definisi zona. Berkas zona dapat didownload pada URL berikut
      - [http://grid.ui.ac.id/files/hastinapura/db.hastinapura\\_grid.ui.ac.id](http://grid.ui.ac.id/files/hastinapura/db.hastinapura_grid.ui.ac.id)
      - <http://grid.ui.ac.id/files/hastinapura/db.192.168>
      - # mkdir /etc/bind/zones
      - # cd /etc/bind/zones
      - # wget
        - http://grid.ui.ac.id/files/hastinapura/db.hastinapura\_grid.ui.ac.id
        - # wget http://grid.ui.ac.id/files/hastinapura/db.192.168
    - iv. Atur forwarders
      - # vi /etc/bind/named.conf.options
      - ...
      - forwarders {
        - 202.46.129.91;
        - };
    - v. Ubah konfigurasi nameserver
      - # vi /etc/network/interfaces
      - ...

```

dns-nameservers 202.46.129.2
...
# vi /etc/resolv.conf
...
nameserver 202.46.129.2
...

```

**d. Instalasi OpenLDAP**

**i. Install paket OpenLDAP server dan utilities.**

```
# apt-get install slapd ldap-utils
```

**ii. Ikuti layar konfigurasi OpenLDAP server. Masukkan data-data berikut (pada dasarnya konfigurasi ini akan diubah pada langkah berikutnya)**

1. DNS domain name: its.ac.id
2. Name of your organization: GRID
3. Admin password: silakan pilih
4. Allow LDAPv2 Protocol: No

**iii. Tambahkan skema inGRID yang dapat diambil dari URL**

```
http://grid.ui.ac.id/files/ingrid.schema
```

```
# cd /etc/ldap/schema
```

```
# wget http://grid.ui.ac.id/files/ldap/ingrid.schema
```

**iv. Ubah konfigurasi OpenLDAP server pada berkas /etc/ldap/slapd.conf**

```
# vi /etc/ldap/slapd.conf
```

```
...
```

```
include /etc/ldap/schema/ingrid.schema
```

```
...
```

```
suffix " ou=Olympus ,ou=Grid,ou=IF,o=Institut Teknologi  
Sepuluh Nopember,c=ID "
```

```
...
```

```
access to attrs=userPassword
```

```
by dn="cn=admin, ou=Olympus ,ou=Grid,ou=IF,o=Institut  
Teknologi Sepuluh Nopember,c=ID " write
```

```
by anonymous auth
```

```
by self write
```

- ```

by * none
...
access to *
by dn="cn=admin, ou=Olympus ,ou=Grid,ou=IF,o=Institut
Teknologi Sepuluh Nopember,c=ID " write
by * read
...
rootdn "cn=admin, ou=Olympus ,ou=Grid,ou=IF,o=Institut
Teknologi Sepuluh Nopember,c=ID "

```
- v. Buat password untuk administrator LDAP

```

# slappasswd
New password: <passwd>
Re-enter new password: <passwd>
{SSHA}z9gP0KGOxypK3pMGArS/N5MGO2B49DFo

```
  - vi. Salin sebuah string yang dihasilkan
  - vii. Buka kembali konfigurasi OpenLDAP server untuk memasukkan password administrator

```

# vi /etc/ldap/slapd.conf
...
rootpw
{SSHA}z9gP0KGOxypK3pMGArS/N5MGO2B49DFo
Masukkan string yang sebelumnya dihasilkan setelah kata
rootpw

```
  - viii. Restart OpenLDAP server

```

# /etc/init.d/slapd restart

```
  - ix. Ubah konfigurasi LDAP utilities

```

# vi /etc/ldap/ldap.conf
...
URI ldap://202.46.129.91
BASE ou=Olympus ,ou=Grid,ou=IF,o=Institut Teknologi
Sepuluh Nopember,c=ID
...

```
  - x. Inialisasi isi dari LDAP dengan berkas LDIF yang didapatkan dari URL <http://grid.ui.ac.id/files/ldap/init.ldif>

```

# wget http://grid.ui.ac.id/files/init.ldif

```

```
# ldapadd -x -D "cn=admin, ou=Olympus
,ou=Grid,ou=IF,o=Institut Teknologi Sepuluh
Nopember,c=ID " \
-W -f init.ldif
```

**e. Konfigurasi NFS**

**i. Install paket NFS Client**

```
# apt-get install nfs-common
```

**ii. Tambahkan data berikut ke dalam berkas /etc/fstab**

```
10.126.12.43:/export /export nfs defaults 0 0
```

```
10.126.12.43:/opt/sge /opt/sge nfs defaults 0 0
```

**iii. Mount ulang**

```
# mount -a
```

**f. Instalasi Sun Grid Engine**

**i. Instal aplikasi yang dibutuhkan**

```
# apt-get install binutils
```

**ii. Buat user administrator untuk Sun Grid Engine**

```
# adduser sgeadmin
```

**iii. Pastikan direktori /opt/sge yang berada pada mesin Storage telah di-mount**

**iv. Tambahkan informasi port yang digunakan oleh Sun Grid Engine dengan menambahkan dua baris berikut ke dalam berkas /etc/services**

```
sge_qmaster 536/tcp
```

```
sge_execd 537/tcp
```

**v. Lakukan instalasi master host. Secara umum, isian default dapat digunakan. Khusus untuk port range yang digunakan, isilah dengan 20000-20100**

**vi. Konfigurasi Sun Grid Engine**

Sebelum melakukan konfigurasi, jalankan script persiapan environment dari Sun Grid Engine.

```
# source /opt/sge/default/common/settings.sh
```

Setelah itu, lakukan langkah-langkah berikut

**1. Aktifkan fungsi reporting**

```
# qconf -mconf
```

```
...
```



```
reporting_params accounting=true reporting=true \
flush_time=00:00:15 joblog=true sharelog=00:00:00
```

...

2. Jalankan job sederhana agar fungsi reporting bekerja untuk pertamakalinya

```
# qsub /opt/sge/examples/jobs/simple.sh
```

3. Tambahkan parallel environment untuk MPI

```
# qconf -ap mpich
pe_name mpich
slots 9999
user_lists NONE
xuser_lists NONE
start_proc_args /opt/sge/mpi/startmpi.sh $pe_hostfile
stop_proc_args /opt/sge/mpi/stopmpi.sh
allocation_rule $round_robin
control_slaves FALSE
job_is_first_task TRUE
urgency_slots min
```

4. Masukkan parallel environment ke dalam queue

```
$ qconf -mq all.q
...
pe_list make mpich
...
shell /bin/sh
...

```

#### g. Instalasi inGRID SGE Appliance

- i. Instal paket yang dibutuhkan
 

```
# apt-get install sudo xinetd
# apt-get install zlib-bin zlib1g zlib1g-dev
# apt-get install gcc g++ g77 manpages-dev make
# apt-get install zip unzip
```
- ii. Buat user globus
 

```
# adduser globus
```
- iii. Download berkas instalasi dari URL berikut



- [http://grid.ui.ac.id/files/inGRID-Installer/inGRID-Installer\\_0.1-debian.tar.gz](http://grid.ui.ac.id/files/inGRID-Installer/inGRID-Installer_0.1-debian.tar.gz)
- iv. Extract berkas instalasi  

```
$ tar xzf inGRID-Installer_0.1-debian.tar.gz
```
  - v. Edit berkas `build.properties`, ubah nilai variabel-variabel berikut
    1. `uc.portal.hostname` menjadi alamat dari portal, yaitu `grid.ui.ac.id`
    2. `uc.appliance.hostname` menjadi alamat head node, yaitu `hastinapura.grid.ui.ac.id`
  - vi. Edit berkas `provider.properties`, sesuaikan dengan informasi cluster
  - vii. Download berkas instalasi CA dari URL berikut  

```
http://grid.ui.ac.id/files/globus_simple_ca_73eb1296_setup-0.19.tar.gz
```

 (TODO)
  - viii. Pindahkan berkas tersebut ke direktori `~globus/inGRID-Installer/3rdParty/TrustedCA`  

```
$ mv globus_simple_ca_73eb1296_setup-0.19.tar.gz \
~globus/inGRID-Installer/3rdParty/TrustedCA
```
  - ix. Ubah berkas `~globus/inGRID-Installer/3rdParty/build.xml` untuk menyesuaikan berkas instalasi CA yang digunakan
    1. Ubah nilai pada baris 23  

```
<property name="simpleca.file.1"
value="globus_simple_ca_ff3e505e_setup-0.19.tar.gz" />
```
    2. Menjadi  

```
<property name="simpleca.file.1"
value="globus_simple_ca_73eb1296_setup-0.19.tar.gz" />
```
  - x. Jalankan script untuk menyesuaikan environment Sun Grid Engine dan PATH untuk MPICH  

```
$ export SGE_ROOT=/opt/sge
$ source $SGE_ROOT/default/common/settings.sh
$ export PATH=$PATH:/opt/mpich/bin
```
  - xi. Jalankan script instalasi  

```
$ ./install.sh sgeappliance
```
  - xii. Jalankan script post-instalasi sebagai user root

- ```

# cd ~globus/inGRID-Installer/3rdParty/
# export SGE_ROOT=/opt/sge
# source $SGE_ROOT/default/common/settings.sh
# export PATH=$PATH:/opt/mpich/bin
# ./postinstall.sh sgeappliance

```
- xiii.** Tandatangan sertifikat mesin oleh Certificate Authority
- ```

# cd /etc/grid-security
# scp hostcert_request.pem globus@portal:
globus@portal$ grid-ca-sign -in hostcert_request -out
hostcert.pem
globus@portal$ scp hostcert.pem root@headnode:/etc/grid-
security/hostcert.pem
# cp hostcert.pem containercert.pem
# cp hostkey.pem containerkey.pem
# chown globus:globus container*

```
- xiv.** Tambahkan data berikut dalam sudoers
- ```

# visudo
globus ALL=(ALL) NOPASSWD: \
/home/globus/UCPortal/GT4/libexec/globus-gridmap-and-
execute \
-g /etc/grid-security/grid-mapfile \
/home/globus/UCPortal/GT4/libexec/globus-job-manager-
script.pl *
globus ALL=(ALL) NOPASSWD: \
/home/globus/UCPortal/GT4/libexec/globus-gridmap-and-
execute \
-g /etc/grid-security/grid-mapfile \
/home/globus/UCPortal/GT4/libexec/globus-gram-local-
proxy-tool *

```
- xv.** Tambahkan berkas adapter Sun Grid Engine bagi MDS
- ```

$ cd ~globus/UCPortal/GT4/libexec
$ wget http://grid.ui.ac.id/files/sge/globus-scheduler-
provider-sge
$ chmod +x globus-scheduler-provider-sge

```
- xvi.** Patch berkas Sun Grid Engine adapter di

```

http://grid.ui.ac.id/files/inGRID-Installer/sge.pm.patch
$ cd
~/globus/UCPortal/GT4/lib/perl/Globus/GRAM/JobManager
$ wget http://grid.ui.ac.id/files/inGRID-
Installer/sge.pm.patch
$ patch < sge.pm.patch

```

**xvii.** Tambahkan konfigurasi GridFTP pada berkas  
`/etc/xinetd.d/grid-ftp`  
`# vi /etc/xinetd.d/grid-ftp`  
`...`  
`env += GLOBUS_TCP_PORT_RANGE=40000,40500`  
`...`

**xviii.** Restart xinetd  
`# /etc/init.d/xinetd restart`

**xix.** Hidupkan Globus Toolkit  
`$ cd /home/globus/UCPortal/GT4`  
`./globus-start.sh`  
`$ cd etc/globus_wsrp_mds_index/DefaultIndex`  
`./srun.sh`

**h. Konfigurasi Tahap Akhir Mesin Head Node**

**i. Konfigurasi NFS**

- 1.** Buat direktori mount-point  
`# mkdir /export`  
`# mkdir /opt/sge`
- 2.** Tambahkan data berikut pada berkas `/etc/fstab`  
`10.126.12.43:/export /export nfs defaults 0 0`  
`10.126.12.43:/opt/sge /opt/sge nfs defaults 0 0`
- 3.** Mount ulang dengan menuliskan perintah berikut  
`# mount -a`
- 4. Mesin Worker**
  - a.** Instalasi Debian GNU/Linux
  - b.** Instalasi OpenSSH
  - c.** Instalasi PAM LDAP

**i. Instalasi paket yang diperlukan**  
`# apt-get install libpam-ldap libnss-ldap ldap-utils`

- ii. **Ikuti layer konfigurasi**  
 LDAP Server host: 10.126.12.43  
 The distinguished name of search base: ou=Olympus ,ou=Grid,ou=IF,o=Institut Teknologi Sepuluh Nopember,c=ID  
 LDAP version to use: 3  
 Database requires login: no  
 Make configuration readable/writable by owner only: no  
 Make local root database admin: no  
 Database requires logging in: no  
 Local crypt to use when changing password: crypt
- iii. **Konfigurasi name service**  

```
# vi /etc/libnss-ldap.conf
base ou=Olympus ,ou=Grid,ou=IF,o=Institut Teknologi
Sepuluh Nopember,c=ID
uri ldap://10.126.12.43
ldap_version 3
scope sub
nss_base_passwd ou=Olympus ,ou=Grid,ou=IF,o=Institut
Teknologi Sepuluh Nopember,c=ID
nss_base_shadow ou=Olympus ,ou=Grid,ou=IF,o=Institut
Teknologi Sepuluh Nopember,c=ID
nss_base_group ou=Olympus ,ou=Grid,ou=IF,o=Institut
Teknologi Sepuluh Nopember,c=ID
```
- iv. **Konfigurasi konfigurasi LDAP**  

```
# vi /etc/ldap/ldap.conf
BASE ou=Olympus ,ou=Grid,ou=IF,o=Institut Teknologi
Sepuluh Nopember,c=ID
URI ldap://10.126.12.43
```
- v. **Konfigurasi nsswitch**  

```
# vi /etc/nsswitchconf
passwd: files ldap compat
group: files ldap compat
shadow: files ldap compat
```
- vi. **Konfigurasi PAM**

- ```

# vi /etc/pam.d/common-account
account sufficient pam_ldap.so
account required pam_unix.so
# vi /etc/pam.d/common-auth
auth sufficient pam_ldap.so
auth required pam_unix.so nullok_secure
# vi /etc/pam.d/common-password
password sufficient pam_ldap.so
password required pam_unix.so nullok obscure min=4 max=8
md5

```
- d. Konfigurasi NFS
    - i. Install paket NFS Client
 

```
# apt-get install nfs-common
```
    - ii. Tambahkan data berikut ke dalam berkas /etc/fstab
 

```
10.126.12.43:/export /export nfs defaults 0 0
10.126.12.43:/opt/sge /opt/sge nfs defaults 0 0
```
    - iii. Mount ulang
 

```
# mount -a
```
  - e. Instalasi MPICH
    - i. Download berkas sumber dari MPICH dari URL berikut  
<http://grid.ui.ac.id/files/mpich/mpich-1.2.7p1.tar.gz>
    - ii. Ekstrak berkas tersebut
 

```
$ tar xzf mpich-1.2.7p1.tar.gz
```
    - iii. Lakukan konfigurasi dan kompilasi
 

```
$ cd mpich-1.2.7p1
$ ./configure --prefix=/opt/mpich
$ make
```
    - iv. Lakukan instalasi sebagai user root
 

```
# make install
```

## 5. Instalasi Sun Grid Engine

Pada mesin worker, Sun Grid Engine yang diinstal adalah bagian execution daemon. Berikut ini adalah langkah-langkah instalasinya. Pastikan mesin worker sudah menjadi administrative host dari Sun Grid Engine.

- a. Jika belum, tambahkan mesin tersebut ke dalam daftar administrative host  
portal# qhost -sh  
portal# qhost -ah node01.portal.grid.if.its.ac.id
- b. Instal aplikasi yang dibutuhkan  
# apt-get install binutils
- c. Buat user administrator untuk Sun Grid Engine  
# adduser sgeadmin
- d. Pastikan direktori /opt/sge dari Mesin Storage sudah di mount pada Mesin Worker
- e. Tambahkan data berikut pada berkas /etc/services  
# vi /etc/services  
...  
sge\_qmaster 536/tcp  
sge\_execd 537/tcp
- f. Lakukan instalasi execution daemon. Secara umum, isian default dapat digunakan  
# source /opt/sge/default/common/settings.sh  
# cd /opt/sge  
# ./install\_execd

## DAFTAR PUSTAKA

- Sun Microsystems, 2002, **Sun Grid Engine 5.3 and Sun Grid Engine, Enterprise Edition 5.3 Reference Manual**, USA
- Ian Foster, Open Grid Service Architecture : tutorial,. 2003, Argonne National Laboratory, University of Chicago, USA
- Heru Haryadi SY, Wendi Gunawan Yusuf, Sylvina Andriani, Happy Indah K, Jakarta Ritonga, Angga Agus H, Fitri, Tyo., **GRID COMPUTING THE NEW INOVASION**, 2005, Sekolah Tinggi Teknologi Telekomunikasi, Bandung. <[URL:http://wiki.sttelkom.ac.id/](http://wiki.sttelkom.ac.id/)>
- Kejian Jin, Prakashan Korambath,.2006, **UCPort Installation Guide**, Academic Technology Services, University of California, Los Angeles
- Grid Computing Research Group., 2006, **Tutorial Penggunaan inGRID Portal**, Fakultas Ilmu Komputer, Universitas Indonesia, Jakarta
- Grid Computing Research Group., 2007, **Panduan Instalasi Cluster Hastinapura**, Fakultas Ilmu Komputer, Universitas Indonesia, Jakarta
- Grid Computing Research Group., 2007, **Panduan Instalasi Portal inGRID**, Fakultas Ilmu Komputer, Universitas Indonesia, Jakarta
- Grid Computing Research Group., 2007, **Rancangan Infrastruktur inGRID**, Fakultas Ilmu Komputer, Universitas Indonesia, Jakarta
- Wikimedia Foundation, Inc., 2007, **Komputasi Grid**, <[URL:http://id.wikipedia.org/wiki/Komputasi\\_grid](http://id.wikipedia.org/wiki/Komputasi_grid)>



Wikimedia Foundation, Inc., 2007, **Komputasi Terdistribusi**,  
 URL:[http://id.wikipedia.org/wiki/Komputasi\\_terdistribusi](http://id.wikipedia.org/wiki/Komputasi_terdistribusi)>

Wikimedia Foundation, Inc., 2008, **Pemrograman Paralel**, <URL:  
[http://id.wikipedia.org/wiki/Pemrograman\\_paralel](http://id.wikipedia.org/wiki/Pemrograman_paralel)>

Grid Computing Research Group., 2007, **Portal inGRID**, Fakultas Ilmu  
 Komputer, Universitas Indonesia, Jakarta <URL:  
<https://grid.ui.ac.id/gridsphere/gridsphere>>

## BIODATA PENULIS



**Dwight Ronald Fitzgerald Tuhusula**, biasa dipanggil dwight, dilahirkan di ibu kota Jakarta pada 2 february 1981. Penulis merupakan anak pertama dari tiga bersaudara dan dibesarkan di tiga dua kota berbeda, yaitu Cilegon dan Bekasi.

Selama berkuliah di Jurusan Teknik Informatika - Institut Teknologi Sepuluh Nopember Surabaya, penulis aktif pada Tim Basket ITS, dan tidak hanya itu saja, penulis juga aktif sebagai pemain basket pada tim UKM Basket ITS 2002-2003 dan sempat memperkuat tim LIBAMA ITS. Selain kiprah penulis di basket, penulis juga pernah menjadi asisten pelatihan pada saat kunjungan mahasiswa ISTJ Jayapura pada tahun 2001, serta mengikuti seminar – seminar mengenai Grid computing baik yang diselenggarakan di kampus ITS maupun yang ada diluar kampus ITS.

Selain pada olahraga, penulis juga mempunyai hobby sulap atau lebih tepatnya street magics dan mentalits. Dan selama penulis mengerjakan Tugas Akhir inilah, penulis banyak mengaplikasikan sulap bersama rekan – rekan IMC (*Informatics Magicians Community*).

Ketertarikan penulis pada dunia IT meliputi desain pada photoshop, Networking dan Sistem Operasi Linux Debian. Selain itu penulis juga tertarik dalam mengembangkan teknologi Grid yang telah dibangun saat ini, dan juga penulis bisa dihubungi melalui email atau IM pada [dwight\\_fitzgerald@yahoo.com](mailto:dwight_fitzgerald@yahoo.com)