

Desain Sistem Interkoneksi pada Arsitektur Komputasi *Cluster*

S. Y. J. Prasetyo

Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Jl. Diponegoro 52-60, Salatiga 50711, Indonesia
Email: srieyulianto@yahoo.com.au

Abstract

Architectural computation cluster is built on an integrative management and control over all applications and resources in one computational system. It is supported by the demand of good performance and vast media storage available in computer for quick-handling computation. The resources needed are hardware, operating system and communication software determined by the purpose of cluster building, compatibility, performance needed by the users and communication software like Message Passing Interface. MPI works using flat tree algorithm allowing non- blocking in message sending mechanism that enable one transmission for all messages in root to all the destinations set. It gives high speed internal communication working on parallel and super computers.

Key Words: Computer Cluster, Grid Computation, Network Architecture, Distribution Storage

1. Latar Belakang

Perkembangan material baru dan proses produksi pada industri berteknologi tinggi menuntut solusi terhadap kebutuhan kompleksitas komputasional. Mulai dari kebutuhan sumber daya komputasi, media penyimpanan dan kecepatan komunikasi yang meningkat secara eksponensial. Konsep *high-performance computing* merupakan alternatif solusi yang ditawarkan untuk memenuhi kebutuhan mendesak tersebut baik pada penelitian maupun pengembangan.

Arsitektural komputasi *cluster* dibangun berdasar pada konsep pengelolaan dan pengendalian keseluruhan aplikasi dan sumber daya secara terintegrasi dalam satu sistem dalam lingkungan komputasional. Ide dasar ini didukung oleh tuntutan kebutuhan penggunaan komputer dengan performa dan sumber daya media penyimpanan yang besar tersedia dalam satu lokasi geografis untuk menangani semua komputasi dalam skala waktu singkat [1]. *Cluster* dapat didefinisikan sebagai dua atau lebih komputer yang bekerja secara bersama-sama untuk memberikan solusi. Perbedaan mendasar pada arsitektur *client-server* adalah bahwa pada lingkungan *client-server* bekerja melalui mekanisme pengiriman layanan terhadap satu atau beberapa server sedangkan

pada *cluster* menyediakan sumber daya misalnya mesin untuk menyediakan lingkungan komputasi sebagaimana sistem tunggal. Di samping itu sistem *cluster* mempunyai konfigurasi konektivitas yang relatif kompleks pada komunikasi antar *node* (mesin) dalam *cluster* [2]. Sistem *cluster* menggunakan beberapa komputer, khususnya PC atau *workstation* dengan berbagai *platform* seperti Linux, Windows NT, dan UNIX, serta beberapa perangkat penyimpanan dan interkoneksi jaringan. Sistem ini menyediakan reliabilitas dan performa sistem yang tinggi baik pada komputer yang berfungsi sebagai *client* maupun komputer yang berfungsi sebagai server.

Saat ini, sebagian besar aplikasi yang digunakan untuk tujuan saintis dan bisnis pada berbagai skala membutuhkan puluhan atau bahkan ratusan prosesor untuk melakukan penyediaan solusi. Sebagian prosesor digunakan untuk proses komputasi dan sebagian lagi digunakan untuk mekanisme penyimpanan data. Tidak mungkin ada prosesor tunggal yang dapat menyediakan kebutuhan fungsionalitas aplikasi sebagaimana yang diperlukan [3]. Aplikasi yang membutuhkan komputer dengan puluhan dan ratusan prosesor harus dapat bekerja dengan performa tinggi (*high performance computer*). Spesifikasi komputer model demikian membutuhkan keseimbangan antara tiga faktor yang meliputi sumber daya komputer, ukuran memori, dan kapasitas I/O dan kapasitas komunikasi [4]. Pada praktiknya, rancang bangun arsitektural *cluster* harus melakukan konfigurasi komponen sumber daya (*resources*). Konfigurasi ini terkait dengan perangkat keras, sistem operasi, jaringan dan perangkat lunak komunikasi. Pemilihan tipe perangkat ditentukan oleh faktor tujuan pembangunan *cluster*, kompatibilitas, dan performa yang dibutuhkan oleh pengguna. Pemilihan perangkat keras, seperti prosesor Intel Pentium, Ppro, PI, PII, PIII, dan seterusnya memiliki pertimbangan yang berbeda dengan prosesor AMD seperti Sparc, Alpha, atau Mach. Pemilihan sistem operasi yang mendukung *cluster* seperti Solaris, WinNT, FreeBSD, NetBSD, dan Linux harus didasarkan pada tujuan pembangunan sistem. Pertimbangan penentuan teknologi jaringan yang digunakan seperti *bandwidth* dan performa, menggunakan ATM (*Asynchronous Transfer Mode*), Ethernet (normal, *fast* dan *gigabit*), Myrinet, SCI, SLIP, dan USB. Perangkat lunak komunikasi adalah *Message Passing System*, yang bekerja melalui mekanisme respon terhadap *message* yang terdistribusi antar *node*. Perangkat lunak sistem *message passing* adalah PVM (*Parallel Virtual Machine*) dan MPI (*Message Passing Interface*) [5].

Pada mesin *cluster*, setiap *node* merupakan komputer dengan *singleprocessor* atau *multiprocessor*, seperti PC, *workstation*, SMP Server sesuai dengan memori, perangkat I/O, dan sistem operasi. *Cluster* yang setiap *node* memiliki spesifikasi yang seragam disebut homogen, sedangkan yang sebaliknya disebut sebagai heterogen. Setiap *node* memiliki interkoneksi dengan LAN (*Local Area Network*) menggunakan teknologi jaringan seperti Ethernet, Fast Ethernet, Gigabit Ethernet, Myrinet, *Quadrics Network* (QsNet), *InfiniBand Communication Fabric*, *Scalable Coherent Interface* (SCI), *Virtual Interface Architecture* (VIA), atau *Memory Channel*.

Kecepatan transfer data dalam bentuk *bandwidth* dan *latency* pada jaringan merupakan karakteristik yang memainkan peranan penting pada teknologi ini. *Bandwidth* menggambarkan seberapa besar informasi yang dapat dikirimkan melalui suatu koneksi jaringan tertentu, sedangkan *latency* menggambarkan waktu yang dibutuhkan untuk suatu perangkat jaringan untuk melakukan pemrosesan *frame* data. Sebagai perbandingan, berikut ini disajikan tabel perbandingan teknologi jaringan pada arsitektur *cluster*.

Tabel 1 Perbandingan Teknologi Jaringan pada Arsitektur *Cluster*

Teknologi Jaringan	Bandwidth(Mbyte/s)	Latency (µs/paket)
Ethernet	1.25	-
Fast Ethernet	12.5	158
Gigabit Ethernet	125	33
Myrinet	245	6
QsNet	340	2
InfiniBand	320, 1280 dan 3840	<10
SCI	400	1.5
VIA	150	8
Memory Channel	100	3

Arsitektur jaringan komputer lain yang memiliki kemiripan dalam tipe dan komponen penyusun adalah *grid* komputer. Komputasi *grid* merupakan infrastruktur dalam bidang komputasional baru yang menjadi pelopor teknologi untuk berbagi sumber daya (*resources sharing*) dan integrasi sistem terdistribusi pada skala besar. Layanan esensial teknologi *grid* adalah layanan informasi, pengelolaan sumber daya, transfer data, dan keamanan sistem [6]. Pada dasarnya, filosofi pemanfaatan komputasi *grid* adalah untuk kolaborasi lintas organisasi dan lokasi geografis dalam orientasi memberikan layanan sumber daya. Filosofi tersebut tidak hanya diterapkan untuk tujuan komputasional saintis namun juga diterapkan untuk tujuan komersial. Lingkup bisnis demikian dicirikan adanya pusat data yang menyediakan layanan sumber daya komputasional dalam skala besar. Sebagai contoh adalah industri berbasis teknologi informasi Hewlett-Packard yang menjalankan 120 pusat data di seluruh dunia dan menggunakan 7000 aplikasi untuk mengoperasikan bisnisnya. Infrastruktur dibangun dalam bentuk *grid enterprise* terdistribusi, terkoneksi dengan semua bentuk layanan sumber daya di pusat data tersebut [7].

Komputasional *grid* dapat dibangun menggunakan sistem yang bersifat heterogen, yaitu sistem yang dibangun dengan perbedaan spesifikasi sistem operasi, perangkat keras, perangkat lunak, kapasitas media penyimpanan, kecepatan CPU, konektivitas jaringan, dan teknologi. Eksploitasi komputasi *grid* merupakan fase penting sebagai jembatan antara pengguna (aplikasi sistem) dan *grid* (sumber daya). Tahap pertama dalam eksploitasi komputasional adalah penentuan sumber daya komputer dan konektivitas jaringan. Tahap kedua pemilihan sumber daya yang dibutuhkan dan tahap ketiga adalah eksploitasi sumber daya [8].

Berdasarkan pendekatan tersebut, sebagian ahli mendefinisikan komputasi *grid* dengan pendekatan dispersi secara geografis terhadap komputasi, penyimpanan, sumber daya jaringan, performa, kualitas layanan yang tinggi, penggunaan yang mudah, kemudahan akses data, penyediaan terhadap kolaborasi organisasi, aplikasi untuk berbagi sumber daya, dan data pada lingkungan heterogen.

Lingkungan jaringan *grid* pada dasarnya dapat dibedakan menjadi beberapa kategori yang meliputi: (1) *Computer Grid*, secara esensial merupakan himpunan sumber daya yang terdistribusi (dengan satu atau lebih lokasi). Proses interkoneksi sumber daya, pengaturan pengguna, penjadwalan instruksi, keamanan *grid*, dan autentikasinya. Keuntungan utama pada tipe ini adalah konstruksi infrastruktur yang efisien dalam pengaturan instruksi dan penggunaan sumber daya, (2) *Data Grid*, penyediaan sistem terdistribusi, akses keamanan data, pengelolaan basisdata terdistribusi pada lokasi geografis yang berbeda. Data *grid* memungkinkan eliminasi berbagai anomali dalam basisdata seperti replikasi, perpindahan, dan sentralisasi data. Dengan teknik ini dapat juga dilakukan pengelolaan pengguna berdasarkan hak aksesnya.

Berdasarkan perspektif topologi, tipe *grid* dapat dikategorikan dalam beberapa jenis yang meliputi sistem *cluster*, *Intra Grids*, *Extra Grids*, dan *Inter Grid* [9].

Di samping tujuan komersial, performa, kemampuan komputasi data secara intensif, dan teknologi jaringan dalam komputasi *grid* menjadi solusi terbaik untuk beberapa proyek penelitian ilmiah skala besar seperti astronomi, ruang angkasa, dan proyek gen manusia (*human genome project*). Sebagai contoh adalah proyek riset *Grid Data Farm Petascale Data Intensive Computing* yang dibangun oleh *High Energy Accelerator Research Organization*, *ETL/TACC (Electrotechnical Laboratory/Tsukuba Advanced Computing Center)*, *University of Tokyo*, dan *Tokyo Institute of Technology (Titech)* Jepang [10].

Perbedaan mendasar antara arsitektur komputer *grid* dan *cluster* adalah sebagaimana berikut ini.

Localization, sumber daya komputer *cluster* berada dalam satu ruangan atau dalam beberapa gedung, sedangkan arsitektur *grid* tersebar melalui areal yang lebih luas dan terkoneksi secara global.

Ownership, pada arsitektur *cluster* dan *grid*, karena terkait dengan sumber daya yang besar umumnya tidak dapat dimiliki oleh individual, namun secara ideal dimiliki oleh publik dengan pemerintah sebagai regulatornya.

Genesis, intentions, pada arsitektur *grid*, tidak dapat dipilih-pilih sumber daya/komponen yang akan digunakan, berbeda dengan *cluster*, bisa dilakukan pemilihan sumber daya/komponen sesuai dengan yang dibutuhkan oleh pihak pengelola.

Granularity, baik arsitektur *grid* dan *cluster* umumnya secara struktural dalam bentuk granular.

Parallelism, arsitektur *grid* dan *cluster* merupakan komputasi paralel dalam hal aturan, sumber daya komputasional, dan sekuensial *job*.

Interconnection, ditinjau dari performa sistem multiprosesornya, *cluster* membutuhkan interkoneksi berkecepatan tinggi.

Centralization, berdasarkan sistem pengelolaan jaringan, *grid* bersifat desentralisasi sehingga sangat fleksibel, konsekuensinya pada kontrol informasi membutuhkan mekanisme komunikasi *message* yang lebih besar antar *node*-nya, sedangkan *cluster* bersifat sentralisasi.

2. Tinjauan Teoritis

Komputasi *cluster* adalah teknologi yang di dalamnya terdiri atas beberapa komputer, yang disebut juga *node cluster*, yang bekerja bersama-sama untuk menyelesaikan suatu masalah komputasi. Setiap *node cluster* dapat berupa komputer yang memiliki prosesor tunggal atau multiprosesor seperti PC, *workstation*, atau SMP Server lengkap dengan perangkat lainnya seperti memori, perangkat I/O, dan sistem operasi. *Node cluster* terkoneksi melalui jaringan LAN menggunakan Ethernet, Fast Ethernet, Gigabit, Ethernet, Myrinet, *Quadrics Network*, *InfiniBand Communication Fabric*, *Scalable Coherent Interface*, *Virtual Interface Architecture*, atau *Memory Channel*.

Pada dasarnya, hampir semua sistem operasi seperti Linux, Solaris, FreeBSD, dan Windows dapat digunakan untuk pengelolaan *node cluster*. Namun demikian masih dibutuhkan aplikasi yang ditulis sebagai *clustering libraries* pada sistem *middleware* seperti PVM (*Parallel Virtual Machine*) dan MPI (*Message Passing Interface*) [11].

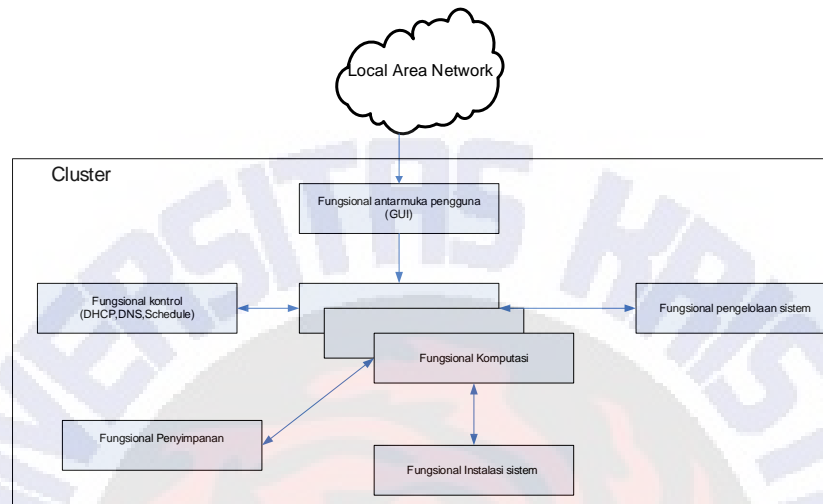
Koneksi PC *Cluster* dapat digunakan sebagai sistem server dengan skalabilitas besar dan performa komputer paralel yang tinggi. Koneksi PC *cluster* yang diimplementasikan basisdata, mengakibatkan sejumlah besar data dapat ditransfer secara kontinu dari satu *node cluster* menuju *node cluster* lainnya. Di samping itu dapat pula dilakukan eksekusi secara paralel, distribusi beban, dan pengelolaan sistem.

Pada dasarnya *cluster* secara tipikal dibangun dengan sejumlah *node*, yang mana setiap *node* tersebut memiliki fungsional logika yang berbeda [2]. Adapun struktur logika fungsional *node* dalam suatu *cluster* komputer adalah sebagaimana pada Gambar 1.

Berdasarkan tujuan desain dan fungsionalnya, komputasi *cluster* dapat dikategorikan dalam beberapa jenis. Jangkauan komputasi *cluster* adalah kemampuan dalam komputasional secara intensif atau akuisisi data secara intensif sebagaimana yang digunakan dalam aplikasi dan pemodelan.

Cluster High Availability, didesain untuk penyediaan data atau layanan yang tidak dapat ditunda pada masyarakat pengguna akhir. Tujuan kategori *cluster* ini adalah penyediaan layanan suatu aplikasi berjalan hanya pada suatu *node cluster*, namun ketika terdapat *cluster* lain yang dalam jangka panjang tidak melakukan eksekusi barulah dilakukan distribusi beban. Kategori ini banyak diterapkan untuk aplikasi basisdata, *mail*, web, atau aplikasi server. *Cluster Aware*, aplikasi didesain secara spesifik untuk digunakan dalam lingkungan *cluster*. Hal ini dapat diidentifikasi melalui mekanisme komunikasi

antar *node cluster*. Sebagai contoh adalah jika menjalankan aplikasi basisdata, yang mana di dalamnya terjadi modifikasi data. Namun demikian *cluster* tidak tahu pada *cluster* atau *node* yang manakah proses modifikasi data dalam basisdata tersebut berjalan.



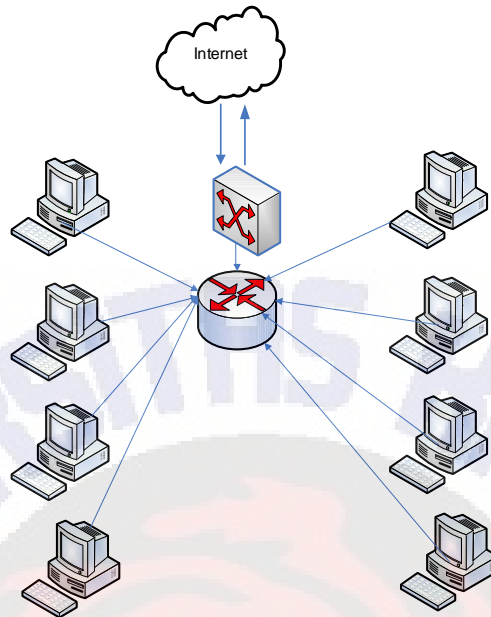
Gambar 1 Struktur Logika Fungsional Komputasi *Cluster*

3. Desain Arsitektur

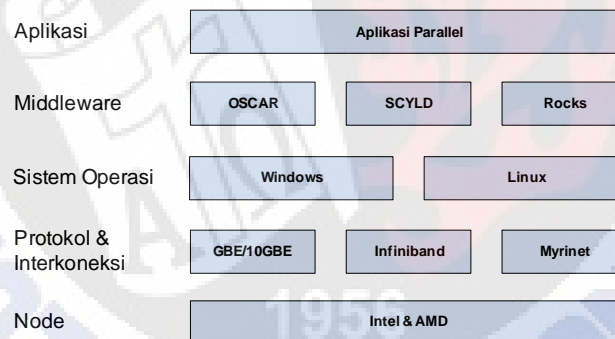
Cluster Load Balancing, tipe ini melakukan mekanisme distribusi sumber daya ke seluruh *node* dan menjalankannya secara bersamaan. Jika ada *node cluster* yang terganggu dan tidak dapat melakukan pemrosesan instruksi maka akan mengirimkan kembali sumber daya tersebut pada *node* yang tersedia. Kategori ini dapat diimplementasikan untuk pencapaian sistem dengan reliabilitas, skalabilitas, dan ketersediaan aplikasi dan sumber daya data secara luas seperti web, *mail*, *news*, atau layanan FTP. Arsitektural umum kategori ini adalah sebagaimana pada Gambar 2.

Bangunan dasar *cluster* dapat dibagi dalam beberapa kategori yaitu: *node cluster*, sistem operasi *cluster*, jaringan *switch hardware*, dan *node/interkoneksi switch*. Adapun komponen *cluster* adalah sebagaimana pada Gambar 3 dan Gambar 4.

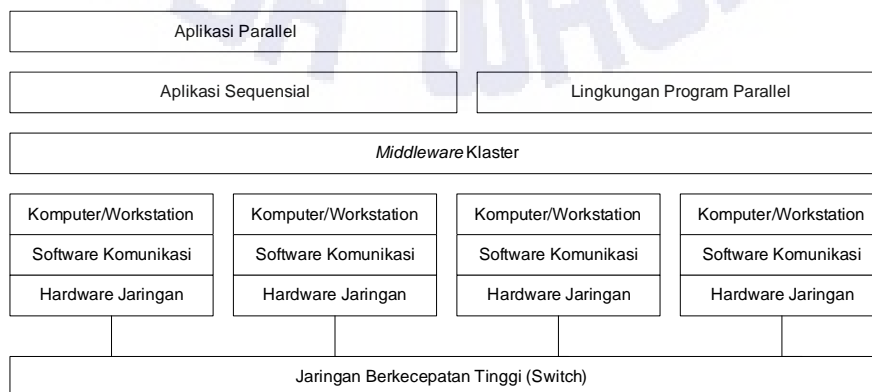
Penentuan kategori arsitektural *cluster* yang akan digunakan, *High Performance Computing* (HPC) atau *High Availability Computing* (HAC) berdasarkan pada aplikasi yang akan digunakan dan budget yang tersedia. Rekombinasi kedua arsitektural tersebut menghasilkan sistem dengan performa yang sangat tinggi. Perbedaan mendasar kedua arsitektural tersebut adalah bahwa pada HPC, setiap *node* dalam *cluster* akan melakukan eksekusi setiap instruksi yang diberikan, dengan demikian dimungkinkan beberapa *node* siap untuk melakukan eksekusi instruksi yang sama.



Gambar 2 Arsitektural Umum Kategori *Cluster Load Balancing*

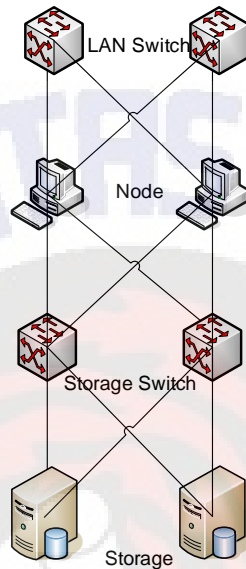


Gambar 3 Komponen *Cluster*

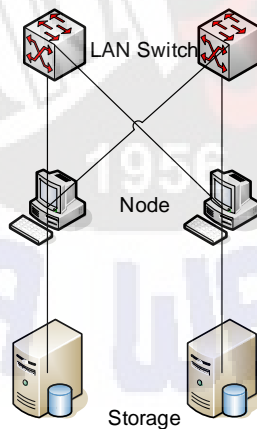


Gambar 4 Komponen *Cluster* dalam Jaringan

Sedangkan HA umumnya digunakan pada aplikasi yang berorientasi pada layanan *end-user* dalam beberapa *node* komputasi. Contoh aplikasi seperti ini adalah aplikasi *web services*, *e-commerce*, basisdata, dan sebagainya [11]. Adapun arsitektur dasar adalah sebagaimana pada Gambar 5 dan Gambar 6 berikut ini.



Gambar 5 Arsitektur Dasar *Storage Cluster* Terdistribusi



Gambar 6 Arsitektur Dasar *Cluster* Terdistribusi

Secara matematis *cluster* dapat didefinisikan sebagai sejumlah prosesor dan sumber daya terkait yang akan digunakan untuk beberapa tujuan, ditambahkan dengan sejumlah sumber daya lainnya. Dapat diformulasikan sebagai berikut:

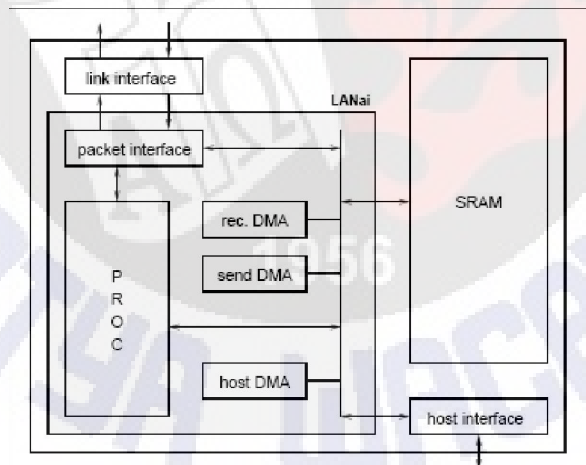
$$N_n(t) = N_0 + N_1(t)$$

dimana *node* merupakan himpunan bagian dari *cluster* prosesor (dengan mekanisme memori terdistribusi) dengan mekanisme komunikasi internal yang lebih cepat dibandingkan dengan *node* komunikasi. N_0 adalah konstanta jumlah *node* dalam *cluster*; t adalah waktu, dan $N_1(t_0)$ adalah variabel jumlah *node*, penggunaan sumber daya mulai dari $t = t_0$, sedangkan $N_n(t_0)$ adalah jumlah seluruh *node* yang tersedia pada waktu tertentu.

4. Desain Topologi

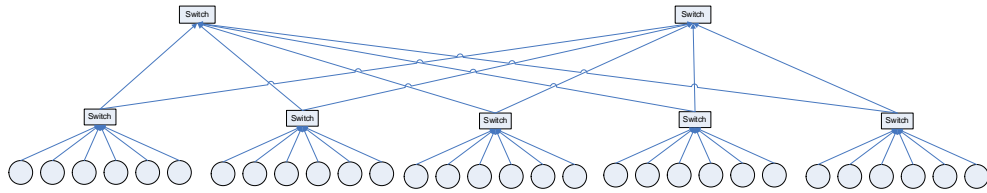
Interkoneksi perangkat keras *node cluster* dapat dilakukan dengan beberapa teknologi, yaitu Fast Ethernet, FDDI, ATM, Fiber-channel, Myrinet, dan SCI. Myrinet adalah perangkat yang digunakan untuk komunikasi berkecepatan tinggi untuk jaringan komputer, diproduksi secara komersial oleh Myricon Inc. Myrinet menyediakan *tool* untuk antar muka perangkat keras dan perangkat lunak jaringan komputer berbasis *cluster*. Myrinet NIC bekerja berdasarkan *chip* LANai 4.1 dengan prosesor RISC, dua DMA Controller untuk penanganan *bit stream* dari dan ke dalam jaringan dan satu DMA Controller untuk transfer *message* dari dan ke *host*. *Layout* Myrinet adalah sebagaimana pada Gambar 7.

Myrinet merupakan jenis LAN yang memiliki kecepatan komunikasi tinggi didasarkan pada teknologi komunikasi paket data dan *switch* untuk MPPs (*Massively Parallel Processors*) [13].

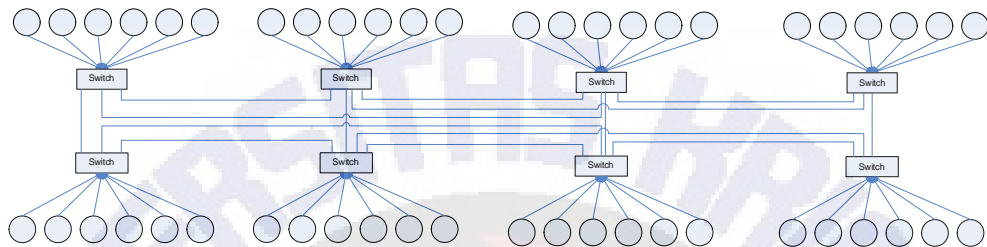


Gambar 7 Layout Myrinet

Switch Myrinet secara mendasar terdiri atas *mesh* 8x8 dengan *latency* sebesar 0.5μ . Karakteristik penting *switch* ini adalah kemampuannya dalam kontrol aliran paket data. Paket dikirimkan oleh *switch* melalui jalur yang sama pada saat permintaan pengiriman, *flow control* menangani perangkat keras dengan melakukan *blocking packet* pada saat *output port* sedang dalam keadaan sibuk. Paket dapat diblok dengan waktu yang singkat, kemudian didrop. Topologi Myrinet adalah sebagaimana pada Gambar 8 dan Gambar 9.

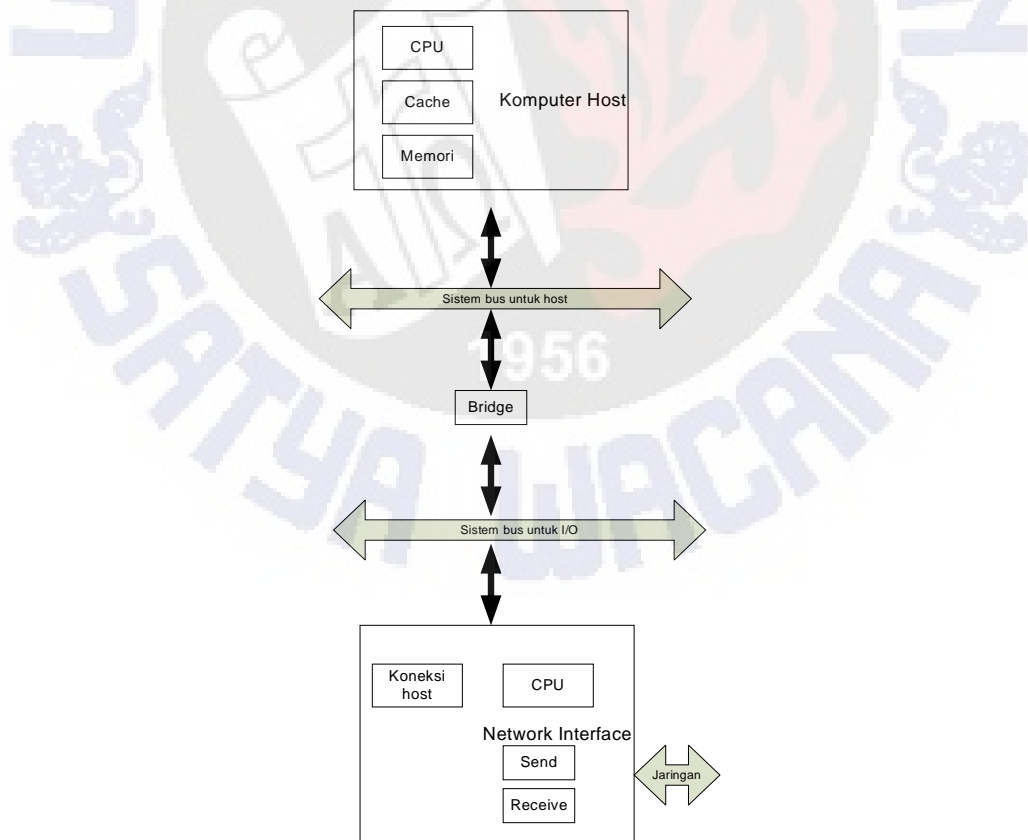


Gambar 8 Topologi Fat-tree Myrinet



Gambar 9 Topologi Cube Myrinet

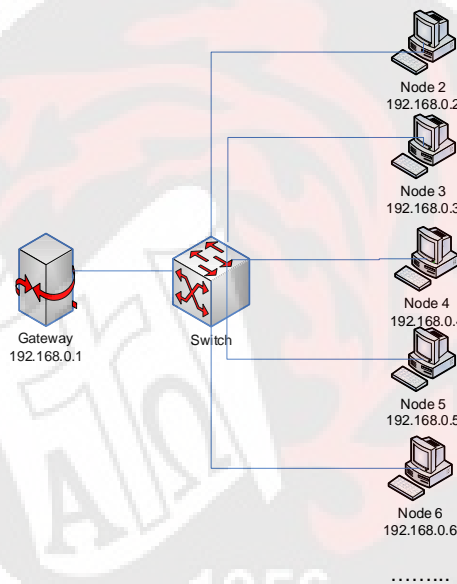
Arsitektur *interface* antara jaringan dan *host* pada Myrinet dapat digambarkan pada diagram berikut ini.



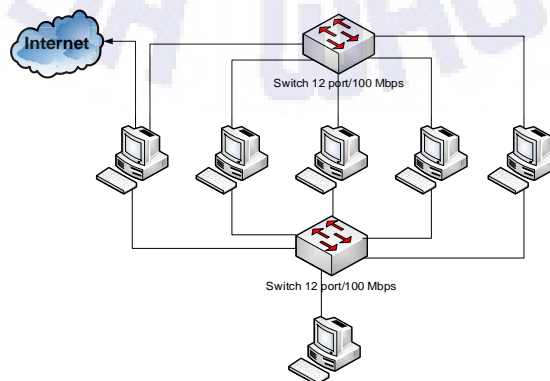
Gambar 10 Arsitektur *Interface* antara Jaringan dan *Host* pada Myrinet

Setiap komputer (*host*) memiliki NI (*Network Interface*) yang mana di dalamnya terdapat prosesor dan memori untuk pengendalian program dan data. NI dikoneksi dengan I/O pada host melalui sistem bus [14].

Kebutuhan dasar perancangan dan pembangunan sistem komputasi *cluster* adalah penyediaan beberapa komputer sebagai *cluster node*. Sebagian komputer dikelompokkan sebagai *master node* dan sebagian komputer dikelompokkan sebagai *slave node*. *Master* melayani sistem sebagai *gateway* untuk komunikasi keluar dari jaringan. Untuk itu diperlukan dua *ethernet card*, satu sebagai *gateway*, untuk penyangga alamat IP statik, yang akan berperan dalam koneksi Internet, dan satu *ethernet card* lagi untuk koneksi dengan *slave* pada *switch* berkecepatan tinggi. Koneksi antar *node* adalah sebagaimana pada Gambar 11. Bentuk koneksi alternatif dalam skala eksperimental adalah sebagaimana pada Gambar 12.



Gambar 11 Koneksi Antar *Node* dan Gateway



Gambar 12 Bentuk Koneksi Alternatif dalam Skala Eksperimental

Pemanfaatan arsitektur *cluster* sebagai solusi kebutuhan sistem didasarkan pada pendekatan, kebutuhan komputasi performa tinggi terhadap penyimpanan data operasional perusahaan, pemrosesan dan analisis secara *online*, dan visualisasi informasi. Kedua, arsitektur *cluster* tidak memerlukan perangkat komputer yang memiliki spesifikasi tinggi namun dengan spesifikasi standar dapat dicapai performa komputasi yang tinggi.

Beberapa faktor yang menjadi pertimbangan dalam pemilihan perangkat untuk perancangan sistem adalah sebagai berikut: (1) Kebutuhan kecepatan tinggi untuk proses komputasi numerik, (2) Kebutuhan memori secara fisik untuk proses komputasi, (3) Perangkat keras yang dapat ditingkatkan performanya melalui mekanisme *upgrade* dan penambahan komponen, (4) Jaringan berkecepatan tinggi antar *node cluster*, (5) Akses yang aman pada *pra registry* pengguna, (6) Biaya pembangunan dan pemeliharaan yang rendah.

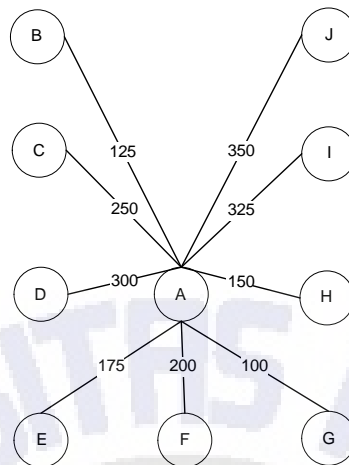
5. Desain Algoritma Cluster

Desain komunikasi pada sebagian besar lingkungan jaringan bekerja di bawah program *message passing system*. Sebagai gambaran adalah konfigurasi koneksi komunikasi internal berkecepatan tinggi (*high speed internal communication configuration*) yang bekerja dalam lingkungan mesin komputer paralel dan superkomputer menggunakan algoritma *binomial tree* dalam MPI [12]. Pada penelitian ini dilakukan perancangan pemanfaatan algoritma *flat tree* sebagai konfigurasi koneksi komunikasi internal berkecepatan tinggi yang bekerja pada lingkungan komputasi *cluster*. Sebagai contoh adalah desain topologi mesin *cluster* dengan matriks antar *node* dalam (*unit time*) adalah sebagaimana pada Tabel 2, dan penyelesaian menggunakan diagram algoritma *flat tree* pada mesin *cluster* adalah sebagaimana pada Gambar 13.

Tabel 2 Matriks Besarnya Nilai dalam Satuan Unit Waktu (*Time Unit*) antar *Node*

	A	B	C	D	E	F	G	H	I	J
A	0	125	250	300	175	200	100	150	325	350
B	125	0								
C	250		0							
D	300			0						
E	175				0					
F	200					0				
G	100						0			
H	150							0		
I	325								0	
J	350									0

Algoritma *flat tree* secara optimal dapat memberikan keuntungan dalam bentuk tidak terjadinya *non-blocking* pada mekanisme pengiriman *message*. Dengan demikian dimungkinkan terjadinya transmisi hanya dalam sekali pengiriman terhadap semua *message* yang ada dalam *root* menuju semua tujuan.



Gambar 13 Algoritma *Flat Tree* pada Desain Mesin *Cluster*

6. Simpulan

Simpulan yang didapat dari penelitian ini adalah (1) Arsitektur *cluster* menyediakan sistem dengan performa tinggi untuk komputasi penyimpanan, pemrosesan, analisis, dan visualisasi informasi, (2) Interkoneksi sistem dapat dilakukan dengan beberapa perangkat keras, perangkat lunak dan topologi sesuai kebutuhan keseluruhan sistem, dan (3) Algoritma *flat tree* potensial diterapkan dalam mesin *cluster* pada topologi torus dengan jumlah sembilan *node*.

7. Daftar Pustaka

- [1] Evans J. dkk, 2004, Network Adaptability in Clusters and Grids, Department of Electrical and Computer Engineering Technology Purdue University dan Department of Computer Science Illinois Institute of Technology, 10 W. 31st Street, Chicago
- [2] Hochstetler dan Bob beringer, 2004, Linux Clustering with CSM and GPFS, International Business Machines Corporation, US
- [3] Robert Ross dkk, 2001, *A Case Study in Application I/O on Linux Clusters*, Mathematics and Computer Science Division Argonne National Laboratory dan Scientific Data Technologies Group, NCSA Center on Astrophysical Thermonuclear Flashes University of Illinois at Urbana-Champaign University of Chicago Champaign, IL 61820, USA Chicago
- [4] Francis A. Vaughan Duncan A. Grove Paul dan D. Coddington, 2003, *Communication Performance Issues for Two Cluster Computers*, Department of Computer Science University of Adelaide, Adelaide, SA 5005, Australia
- [5] Mahmoud A dkk, 1998, Optimizing TCP in a *Cluster* of Low-End Linux Machines, Dept. of Computer Science, The American University in Cairo,

- P.O.Box 2511, Cairo
- [6] Junwei dkk, 2002, *GridFlow: Workflow Management for Grid Computing*, Research Laboratories, NEC Europe Ltd., Sankt Augustin, Germany, NASA Ames Research Centre, Moffett Field, California, USA
 - [7] Graupner S.dan Akhil Sahai, 2004, *Policy-based Resource Topology Design for Enterprise Grids*, Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto.
 - [8] Lacour Sébastien, Christian Pérez dan Thierry Priol, 2004, *A Network Topology Description Model for Grid Application Deployment*, Institut National de Recherche en Informatique et En Automatique, Campus Universitaire de Beaulieu, France
 - [9] Heger, 2006, *An Introduction to Grid Technology-Vision, Architecture, & Terminology* Fortuitous Technology, Austin 7-15, Institute of Solid State Physics, University of Latvia, Kengaraga Street 8, LV-1063 Riga, Latvia
 - [12] Park Kyung Lang, Joo Lee, Oh Young Kwon, Sung Yong Park, Hyung Woo Park, dan Shin Dug Kim, 2000, *An Efficient Collective Communication Method for Grid Scale Networks*, Dept. of Computer Science, Yonsei University, Dept. of Computer Engineering, Korea University of Technology and Education, Dept. of Computer Science, Sogang University, Korea Institute of Science and Technology Information, Korea