



PROSIDING

Seminar Nasional MIPA 2016

Naskah diseminarkan pada 5 November 2016 dan dipublikasikan pada
<http://conf.unnes.ac.id/index.php/mipa/mipa2016/schedConf/presentations>



Rancang Bangun Penjadwalan Pekerjaan Pada *Grid Computing*

Ardi Pujiyanta¹, Selo²

Universitas Ahmad Dahlan¹, Universitas Gadjah Mada²
Institusi Tempat Bekerja
email: ardi_iin@yahoo.com¹

Abstrak

Menemukan model penjadwalan yang baik untuk proses pengajuan kerja, Jaminan pekerjaan dikerjakan, dan karakteristik pekerjaan memberi wawasan penting ke dalam pengoperasian sistem Grid yang dapat digunakan oleh para pengembang dan komunitas riset. Dengan mengamati perilaku sistem dengan nilai yang berbeda dari parameter yang terlibat, pengembang dapat mengevaluasi kinerja, yang tergantung pada pilihan parameter berbeda, dan mungkin mengidentifikasi masalah dengan mengusulkan rancangan serta metode baru Grid Computing untuk meningkatkan dan mengoptimalkan middleware yang digunakan sesuai kebutuhannya. Ketika Penjadwal menerima permintaan pekerjaan, Penjadwal mencoba untuk mengalokasikan slot-waktu bebas, yang dapat memenuhi kebutuhan permintaan, dari slot-waktu yang tersedia, Penjadwal akan merespon permintaan, kemudian kontrak pekerjaan dikatakan telah berhasil disetujui antara permintaan dan Penjadwal. Jika tidak, dikatakan bahwa permintaan pekerjaan ditolak. Jika Penjadwal tidak membuat sumber daya diakses untuk permintaan pekerjaan di waktu mulai, atau gagal untuk menjaga sumber daya diakses selama durasi pekerjaan, dikatakan permintaan pekerjaan ditolak. karena tidak adanya slot waktu kosong.

Kata Kunci: Rancang bangun, Penjadwalan pekerjaan, Grid computing

PENDAHULUAN

Komponen utama dari infrastruktur jaringan adalah keamanan, manajemen sumber daya, layanan informasi, dan manajemen data (Jacob,2003). Istilah Manajemen sumber daya dalam komputasi grid dapat didefinisikan sebagai operasi yang mengontrol cara atau kemampuan grid yang dibuat agar tersedia untuk entitas lain seperti pengguna, aplikasi, dan layanan (Foster dan Kesselman,2004). Tujuan dari manajemen sumber daya adalah untuk memastikan penggunaan yang efisien dari sumber daya komputer dan kinerja optimalisasi tugas-tugas tertentu (Isah dan Safwana,2012).

Saat ini masalah ilmiah sangat kompleks sehingga untuk mengatasinya kita perlu daya komputasi besar. Grid diciptakan pada pertengahan 90-an di dunia akademis, awalnya diusulkan untuk menunjukkan komputasi sistem terdistribusi yang akan memberikan layanan komputasi *on demand*, seperti listrik dan air (Luis Ferreira di el., 2005). Dalam lingkungan jaringan global, pengguna berinteraksi dengan broker sumber daya jaringan yang menyembunyikan kompleksitas pengelolaan sumber daya dan penjadwalan. Pada pengoperasian komputasi grid, broker sumber daya jaringan menemukan sifat sumber daya, dimana pengguna dapat mengakses dengan menggunakan Layanan Informasi Grid, negosiasi dengan(grid diaktifkan) sumber daya atau agen mereka yang menggunakan layanan middleware, penjadwalan tugas ke sumber daya, tahap aplikasi dan data untuk diproses (penyebaran) dan akhirnya mengumpulkan hasilnya (Buyya, dkk, 2000).

Grid memperkenalkan cara baru untuk berbagi sumber daya komputasi dan penyimpanan dengan arsitektur yang terpisah secara geografis dengan membentuk site

manajemen sumber daya global (L. Ferreira et al, 2005). Pengelolaan sumber daya istilah dalam komputasi grid dapat didefinisikan sebagai operasi yang mengontrol sumber daya jaringan dan layanan yang dibuat agar tersedia untuk digunakan oleh entitas seperti pengguna, aplikasi dan layanan (Khanli, & Analoui, 2008) untuk memastikan pemanfaatan sumber daya komputer yang efisien dan untuk kinerja optimalisasi tugas-tugas tertentu. Di grid, kita perlu mempertimbangkan status kondisi jaringan dan status sumber daya. Jika jaringan atau sumber daya tidak stabil, pekerjaan akan gagal atau total waktu komputasi akan sangat besar (Chang, Chang & Lin, 2009).

Grids berbasis jenis sumber daya yang terbagi dalam tiga jenis:

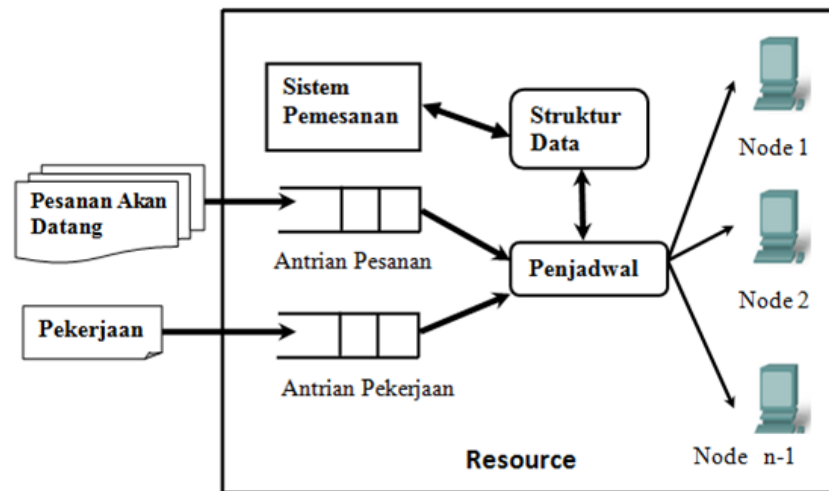
1. komputasi grid, komputasi grid adalah grid yang memiliki kekuatan pengolahan sebagai sumber daya komputasi yang utama, yang dibagi di antara node. Ini adalah jenis yang paling umum dari grid dan telah digunakan untuk melakukan komputasi kinerja tinggi untuk mengatasi pengolahan tugas (Luis Ferreira et al., 2005),
2. Data grid
3. Jaringan grid.

Waktu antar kedatangan pekerjaan, waktu eksekusi pekerjaan, waktu pekerjaan dihabiskan di berbagai fase pengolahan di Grids yang tidak diketahui, secara model probabilistik. Menemukan model probabilistik yang baik untuk proses pengajuan kerja, delay komponen Pekerjaan, dan karakteristik pekerjaan memberi kita wawasan penting ke dalam pengoperasian sistem Grid dan dapat digunakan oleh para pengembang dan komunitas riset. Dengan mengamati perilaku sistem dengan nilai yang berbeda dari parameter yang terlibat, pengembang dapat mengevaluasi kinerja, yang tergantung pada pilihan parameter yang berbeda, dan mungkin mengidentifikasi masalah dan mengusulkan metode baru untuk meningkatkan dan mengoptimalkan middleware yang digunakan.

Mengingat besarnya biaya yang terlibat dalam mendirikan/implementasi *hardware* yang sebenarnya, simulasi adalah alternatif yang dapat digunakan. Sebuah prasyarat yang diperlukan untuk memperoleh hasil yang bermanfaat adalah model yang memadai dari lalu lintas (yaitu proses kedatangan job) dan waktu pekerjaan dengan keadaan yang berbeda. Model probabilistik dapat digunakan untuk memfasilitasi dimensi sistem Grid dan prediksi kinerja grids, dengan skenario yang berbeda. Selain itu, simulasi dapat digunakan untuk mengevaluasi kualitas layanan baru dan kebijakan algoritma penjadwalan yang baik di Grid (*meta-scheduling*) dan di tingkat *site* (*metacomputing*).

Ada tiga jenis pendekatan arsitektur manajemen sumber daya yaitu terpusat, desentralisasi, dan hirarkis. Model penjadwalan terpusat: digunakan untuk mengelola sumber daya tunggal atau ganda yang terletak baik dalam satu atau beberapa domain, model ini hanya dapat mendukung kebijakan seragam dan untuk manajemen cluster (antrian batch). Pendekatan desentralisasi, dalam model ini antar penjadwal saling berinteraksi untuk menentukan sumber daya yang akan mengeksekusi pekerjaan. Model penjadwalan hirarkis: Model ini cocok untuk sistem jaringan karena memungkinkan pemilik sumber daya menentukan kebijakan mereka sendiri pada pengguna eksternal (Foster, A. Roy & V. Sander, 2000).

Untuk menangani luasnya variasi dalam aplikasi perangkat lunak dan perangkat keras yang digunakan dalam lingkungan jaringan, dengan berbagai macam bentuk jaringan grid, perangkat lunak yang dikenal sebagai middleware digunakan. Salah satu komponen yang paling penting dari middleware adalah manajer sumber daya yang menangani seleksi sumber daya dan penjadwalan pekerjaan (Foster, A. Roy & V. Sander, 2000) (Y. Juan, B. Yun & Q. Yuhui, 2007). Sumber daya komputer pada *grids* dikelola oleh *Reservation Manager* (RM), yang melakukan kontrol masukan dan melacak pemesanan pada semua sumber daya yang di bawah kendalinya. Ketika *Reservation Manager* menerima permintaan pemesanan, *Reservation Manager* mencoba untuk mengalokasikan slot-waktu bebas, yang dapat



Gambar 1. model sumber daya keseluruhan yang mendukung Model reservasi fleksibel.

memenuhi kebutuhan permintaan itu. Dari slot-waktu yang tersedia, RM akan merespon permintaan dengan respon konfirmasi. Kemudian, dikatakan bahwa kontrak pemesanan telah berhasil disetujui antara permintaan dan *Reservation Manager*. Jika tidak, dikatakan bahwa permintaan pesanan ditolak. Mengingat kontrak pesanan, jika *Reservation Manager* tidak membuat sumber daya diakses untuk permintaan pesanan di waktu mulai, atau gagal untuk menjaga sumber daya diakses selama durasi pesanan, ini dikatakan bahwa pelanggaran pesanan terjadi.

METODE

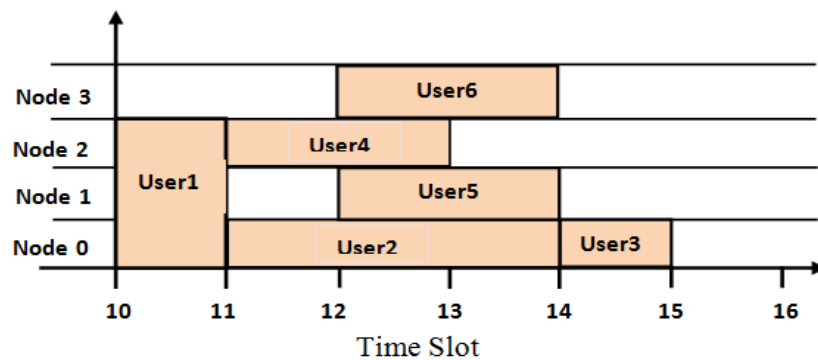
1. Perencanaan dan pemesanan

Perencanaan dan reservasi merupakan tugas penting yang harus dilakukan oleh Grid Sumber daya Manajemen Sistem. Perencanaan dan pemesanan adalah proses menganalisis pekerjaan dan menentukan sumber daya yang diperlukan untuk suksesnya pelaksanaan pekerjaan. Berdasarkan adanya sumber daya yang dicadangkan kepada pengguna.

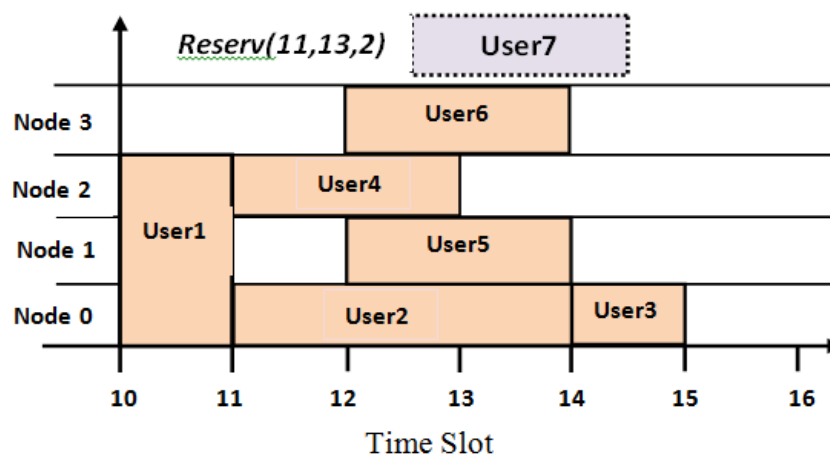
2 Model pemesanan diawal

Gambar 1 menunjukkan model jaringan antrian terbuka dari sumber daya yang diterapkan untuk pekerjaan fleksibel. Dalam model ini, ada dua antrian: satu dicadangkan untuk pekerjaan *advance reservation*, sementara yang lain adalah untuk pekerjaan paralel dan *independent*. Setiap antrian pekerjaan *independent* menunggu untuk dieksekusi oleh salah satu *compute nodes*. Semua *compute nodes* terhubung dengan jaringan berkecepatan tinggi. *compute nodes* di sumber daya dapat homogen atau heterogen. Di dalam model arsitektur penjadwalan ini, diasumsikan bahwa sumber daya memiliki *compute nodes* homogen, masing-masing memiliki kekuatan pemrosesan yang sama, memori dan *hard disk* (Sulistio, dkk,2007).

Dalam model ini, setiap sumber daya memiliki Sistem Reservasi(RS), yang bertanggung jawab untuk menangani permintaan pemesanan *online* dan permintaan pesanan yang akan datang. Ketika *Reservation Manager* menerima permintaan pemesanan *online* atau permintaan pesanan yang akan datang, Sistem Reservasi akan mencari ketersediaan sumber daya. Lebih khusus lagi, Sistem Reservasi memeriksa struktur data untuk permintaan ini. Oleh karena itu, Peran utama dari struktur data untuk menyimpan dan memperbarui informasi tentang ketersediaan *compute nodes* sebagai waktu berjalan. Kemudian, *scheduler* sumber daya bertanggung jawab untuk mengelola pekerjaan yang masuk dan menugaskan mereka untuk *compute nodes* yang tersedia.



Gambar 2. Contoh Diagram *time slot* dengan 4 *compute node*, dimana 6 *user* telah ditempatkan pada masing-masing *node time slot*.

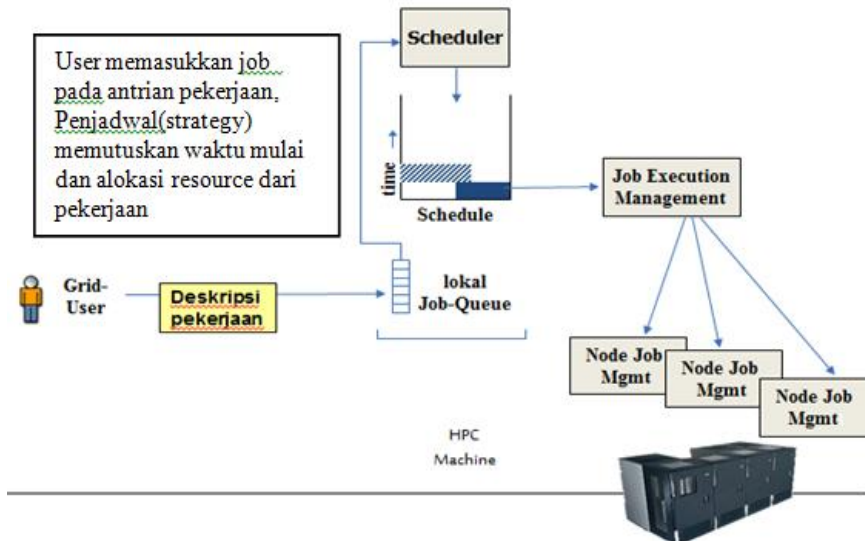


Gambar 3. Contoh *Advance Reservation* fleksibel. Sebuah kotak putus-putus menunjukkan permintaan baru.

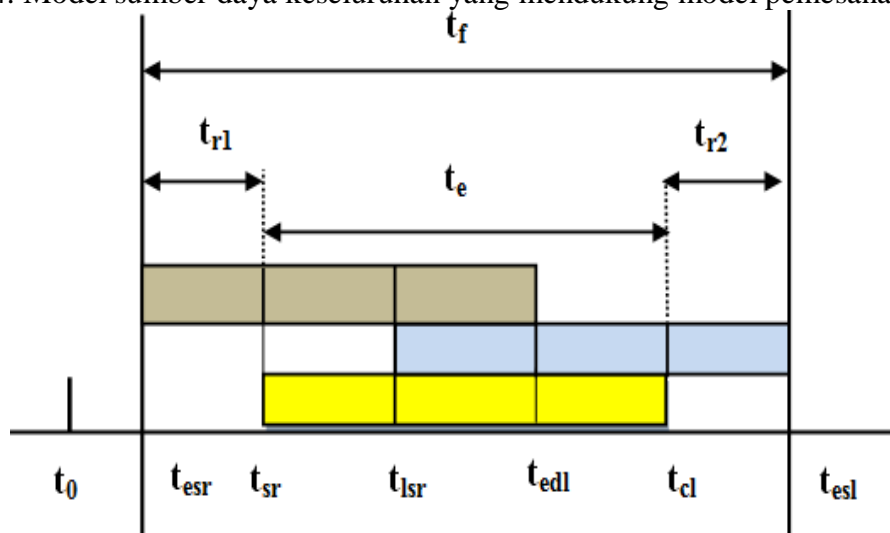
3. Time slot

Timeslot dapat digunakan untuk melacak alokasi pesanan saat ini dan pesanan yang akan datang dari kapasitas yang tersedia dari sumber. Ide dasar dari *Timeslot* adalah menggunakan kerangka acuan sumbu x dan sumbu y ($x \geq 0$, $y \geq 0$) untuk menyajikan jumlah seluruh kapasitas sumber daya yang tersedia, di mana sumbu x merupakan sumbu waktu, dan sumbu y menunjukkan kapasitas node yang tersedia pada setiap titik waktu (H. Chunming, H. Jinpeng & W. Tianyu, 2006). Fungsi Kapasitas (t) adalah kapasitas sumber daya maksimum yang tersedia pada *Timeslot* t dapat diwakili oleh struktur data. Gambar 2 menunjukkan contoh diagram *time slot* dengan 4 *compute node*, dimana 6 *user* telah ditempatkan pada masing-masing *node time slot*. Cara yang sederhana adalah untuk menyimpan semua permintaan pesanan yang diterima ke dalam sebuah array. Ketika permintaan baru diterima, record baru bisa ditambahkan ke array. Dengan cara itu, setiap kali pesanan masuk, kita harus mencari ke semua anggota array apakah ada *Timeslot* kosong atau tidak dengan melakukan perbandingan, ini tidak efisien. Dalam rangka meningkatkan efisiensi, diperlukannya struktur data lain yang dapat mendukung model *Timeslot* yang sesuai model arsitektur yang kami usulkan, Gambar 3 contoh dari model *Advance Reservation* yang fleksibel. Sebuah kotak putus-putus menunjukkan permintaan baru yang datang (*user7*), yang membutuhkan panjang 2 slot waktu, dimulai dari slot 11 sampai slot 13.

Keuntungan utama dari suatu struktur data adalah bahwa semua *timeslot* dapat diurutkan berdasarkan kerangka waktu yang diwakili oleh record data dan menunjukkan jumlah yang tersedia dari kapasitas. Ketika permintaan baru tiba, kami dapat dengan cepat menemukan semua *timeslot* yang terkait dan memeriksa kapasitas yang tersedia



Gambar 4. Model sumber daya keseluruhan yang mendukung model pemesanan fleksibel.



Gambar 5. Model usulan pemesanan akan datang yang Fleksibel

Gambar 4. menunjukkan user yang melakukan submit job pada antrian pekerjaan, scheduler akan memutuskan apakah submit job tersebut dilayani atau tidak berdasarkan ketersediaan slot pada sumber daya. Jika ternyata pada sumber daya ada ruang kosong maka user akan diberitau bahwa jobnya sukses dan jika tidak ada ruang kosong maka jobnya gagal.

HASIL DAN PEMBAHASAN

Gambar 5. menunjukkan model penjadwalan pada slot waktu pemesanan akan datang yang fleksibel. Jika user1 melakukan permintaan pesanan yang memerlukan slot waktu tertentu dan ternyata ada ruang sesuai yang dibutuhkan user, maka scheduler akan menempatkan slot sesuai pesanan, kemudian jika user2 melakukan permintaan pesanan pada ruang yang masih tersisa dan beririsan dengan slot user1, maka slot pesanan user1 akan digeser sehingga user2 dapat menempatkannya (Rusydi, dkk,2012). Pergeseran slot bisa kekiri dan kekanan, sehingga jika slot pesanan awal sudah terlanjur digeser kekanan, dan slot waktu pesanan sebelah kiri membatalkan pesannya, kemudian ada slot pesanan baru masuk yang akan menempati sebelah kanan, maka slot pesanan yang ada disebelah kanan yang sudah

terlanjur digeser, dapat digeser kekiri kembali, untuk membuat ruang agar slot pesanan baru dapat menempatnya.

t_0 : Waktu sekarang

t_{esr} : batas bawah untuk waktu mulai dari pekerjaan (waktu mulai paling awal)

t_{sr} : Batas atas untuk mengakhiri waktu untuk menjalankan pekerjaan

t_e : Waktu eksekusi/durasi pekerjaan

t_{r1}, t_{r2} : t_{r1} (left hole), t_{r1} (right hole), t_r (waktu Relax), didefinisikan oleh $t_r = t_{r1} + t_{r2} = t_d - t_{es} - t_e$

t_{lsr} : Batas atas untuk memulai pelaksanaan pekerjaan (waktu mulai terakhir), didefinisikan sebagai $t_{lsr} = t_{esl} - t_e = t_{esr} + t_r$

t_{edl} : batas bawah sampai waktu untuk mengakhiri pelaksanaan pekerjaan, didefinisikan sebagai $t_{edl} = t_{esr} + t_e$

t_{sr} : Waktu mulai untuk menjalankan pekerjaan ($t_{esr} \leq t_{sr} \leq t_{lsr}$)

t_{cl} : Waktu penyelesaian untuk menjalankan pekerjaan ($t_{edl} \leq t_{cl} \leq t_{esl}$)

t_f : Waktu fleksibilitas, didefinisikan sebagai $t_f = t_{esl} - t_{esr}$

f : Tingkat fleksibilitas, didefinisikan sebagai $f = \frac{t_f}{t_e}$, di sini $f \geq 1$, (jika $f = \infty$, pekerjaan

dianggap sebagai mode non job reservasi Jika $t_s = t_0$ dan $f = 1$ pekerjaan untuk reservasi dipertimbangkan dengan prioritas paling atas mengarah ke modus pemesanan langsung yaitu, modus penjadwalan.)

userid : Identifikasi pengguna

jobid : Identifikasi pekerjaan

numj : Jumlah pekerjaan

numCN: Jumlah node komputasi yang dibutuhkan

maxCN: Jumlah total node penghitung

Jika diasumsikan bahwa:

1. Pada waktu user melakukan submit maka $t_{esr} \neq t_{sr}$ dan nilai $t_{sr} = t_{lsr}$
2. Pada waktu penjadwalan bisa saja terjadi $t_{esr} = t_{sr} = t_{lsr}$
3. Jika ditentukan nilai hole/lubang(t_r)= $t_r = t_e = t_{r1} + t_{r2}$, agar slot dapat digeser kekiri.

Definisi 1. permintaan pemesanan dapat ditandai sebagai 4-tuple $\langle t_{esr}, t_{sr}, t_{lsr}, t_e \rangle$, di mana t_{esr} adalah batas bawah untuk waktu mulai dari pekerjaan (waktu mulai paling awal), t_{sr} adalah Batas atas untuk mengakhiri waktu untuk menjalankan pekerjaan, t_{lsr} adalah Batas atas untuk memulai pelaksanaan pekerjaan (waktu mulai terakhir), didefinisikan sebagai $t_{lsr} = t_{esl} - t_e = t_{esr} + t_r$, t_e adalah Waktu eksekusi/durasi pekerjaan.

Definisi 2. Jika diberikan *time-slot* $Slot_k$ dan permintaan pesanan Pes_i , dapat memenuhi persyaratan Pes_i jika $t_{esr} \leq t_{esl} \cap t_{esl} \geq t_{esl} \cap t_e \geq t_e$

Definisi 3. Diberikan *time-slot* $slot_k$ dan permintaan pesanan Pes_i , ditentukan kejadian random E_i , menyatakan bahwa tidak ada pelanggaran pesanan yang terjadi pada Pes_i . Maka $slot_k$ dapat memenuhi persyaratan fleksibel Pes_i dengan probabilitas $\Pr\{E_i\}$ sebagai berikut $\Pr\{E_i\} = \Pr\{t_{esr} \leq t_{esl} \cap t_{esl} \geq t_{esl} \cap t_e \geq t_e\}$

Pertimbangkan permintaan pesanan Pes_i yang dinotasikan sebagai $\langle t_{esr}, t_{sr}, t_{lsr}, t_e \rangle$ yang datang pada RM(*Reservation Manager*), Untuk memenuhi persyaratan Pes_i *Reservation Manager* harus menemukan slot bebas/kosong yang memenuhi $t_f \geq t_e$, yang berada dalam

periode $[t_{esr}, t_{esl}]$, di mana t_f adalah jumlah kapasitas bebas. Jika tidak ada *time-slot* yang dapat memenuhi persyaratan P_{es_i} , harus ada satu atau lebih pemesanan yang digeser dengan yang ada pada waktu pemesanan P_{es_i} . Seperti yang ditunjukkan pada Gambar 5 diatas. diasumsikan bahwa semua permintaan pemesanan adalah *independent*, sehingga probabilitas Ei dapat dinyatakan sebagai:

$$\Pr\{E_i\} = \Pr\{E_i^s\} \cdot \Pr\{E_i^e\}$$

Untuk pekerjaan dengan permintaan m pemesanan, probabilitas sukses reservasi untuk pekerjaan ini dapat dinyatakan sebagai:

$$\Pr\{E\} = \prod_{i=1}^m \Pr\{E_i^s\} \cdot \Pr\{E_i^e\}$$

di mana E adalah varian acak yang mewakili kebijakan reservasi fleksibel yang tidak mengakibatkan pelanggaran reservasi.

SIMPULAN

Advance Reservation(AR) dalam komputasi Grid adalah penting pada area penelitian karena memungkinkan pengguna untuk mendapatkan akses bersama ke sumber daya dengan memungkinkannya aplikasi dapat dieksekusi secara paralel. Ini juga memberikan jaminan pada ketersediaan sumber daya pada waktu yang ditentukan di masa yang akan datang. Kami memperkenalkan model fleksibel, dengan mempertimbangkan pemanfaatan sumber daya saat memproses permintaan reservasi. Dengan model ini, pengguna dapat melakukan query tentang ketersediaan sumber daya pada interval waktu tertentu. Pendekatan ini memungkinkan tingkat fleksibilitas kepada pengguna untuk memilih pilihan terbaik dalam pemesanan pekerjaan yang akan datang mereka, sesuai parameter *Quality of Service* (QoS).

DAFTAR PUSTAKA

- Jacob(2003), *Enabling Applications for Grid Computing with Globus*, USA: IBM Corporation.
- Foster dan Kesselman(2004), *The Grid 2: Blueprint for a New Computing Infrastructure*, Los Altos, California: Morgan Kaufmann.
- Isah dan Safwana(2012), "Resource Management in Grid Computing: A Review," *Greener Journal of Science Engineering and Technological Research*, vol. 2, no. 1.
- Buyya, dkk,(2000), "An Economy Driven Resource Management Architecture for Global Computational Power Grids," in *The 2000 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2000)*, CSREA Press, Las Vegas, USA.
- L. Ferreira at el.(2005). *Grid Computing in Research and Education. International Technical Support Organization*.
- Khanli, K.L, and Analoui, M.,(2008). *An approach to grid resource selection and fault management based on ECA rules. Future Generation Computer Systems*,24(4),296-316.
- I. Foster, A. Roy and V. Sander,(2000), "A quality of service architecture that combines resource reservation and application adaptation," in *8th International Workshop on Quality of Service (IWQOS 2000)*, IEEE Press, Pittsburgh.
- Y. Juan, B. Yun and Q. Yuhui, 2007, "A decentralized resource allocation policy in minigrid," *Future Generation Computer Systems*, vol. 23, no. 3.
- Chang, R., Sh. Chang, J., and Sh. Lin, P.,(2009). *An ant algorithm for balanced job scheduling in grids. Future Generation Computer Systems*, 25, 20–27

- Sulistio , dkk,(2007), "On Incorporating an On-line Strip Packing Algorithm into Elastic Grid Reservation-based Systems," in *13th International Conference on Parallel and Distributed Systems (ICPADS'07)*, IEEE Press, Hsinchu.
- H. Chunming, H. Jinpeng and W. Tianyu , 2006, "Flexible Resource Reservation Using Slack Time for Service Grid," in *12th International Conference on Parallel and Distributed Systems, IEEE Press*, Washington.
- Rusydi, dkk,(2012), "Advance Planning and Reservation in a Grid System," in *NDT 2012. CCIS/LNCS, vol. 293. Springer, Heidelberg* , Dubai.