

Media Pembelajaran untuk Metode Penjadwalan First Come First Serve-Ejecting Based Dynamic Scheduling (FCFS-EDS) untuk MPI Job dalam Sistem Grid System

Rusydi Umar

Program Studi Teknik Informatika, Fakultas
Teknologi Industri, Universitas Ahmad Dahlan
Yogyakarta, Indonesia
rusydi_umar@rocketmail.com

Ardi Pujiyanta

Program Studi Teknik Informatika, Fakultas
Teknologi Industri, Universitas Ahmad Dahlan
Yogyakarta, Indonesia
ardi_iin@yahoo.com

Abstract— *First Come First Serve-Ejecting Based Dynamic Scheduling (FCFS-EDS)* adalah metode untuk menjadwalkan untuk MPI Job berdasarkan reservasi atau pemesanan. *Message Passing Interface (MPI)* adalah spesifikasi API yang memungkinkan aplikasi paralel untuk berkomunikasi dengan sesama dengan saling berkiriman pesan. Penjadwalan FCFS-EDS masuk dalam kategori *Flexible Advance Reservation*, dinamik pada sudut pandang logik. Oleh karena itu akan sulit dipahami dan dibayangkan karena hanya bersifat logik saja. Untuk itulah perlu dibuat simulasi dari penjadwalan MPI job dengan metode FCFS-EDS, untuk membantu memahami proses penjadwalan MPI job dengan metode FCFS-EDS. Dengan simulasi ini pengguna dapat merubah-ubah parameter dalam penjadwalan MPI job dan melihat visualisasi dari penjadwalan MPI job dalam sudut pandang logik, dan juga mengetahui bagaimana nantinya sebuah MPI job akan dieksekusi.

Keywords—*media pembelajaran; FCFS-EDS; Penjadwalan; simulasi;*

I. PENDAHULUAN

Aplikasi paralel mempunyai kebutuhan *resource* yang sangat besar, dan *resource* itu berasal dari berbagai komputer paralel yang dapat dieksekusi dalam waktu yang sama. *Message Passing Interface (MPI)* adalah spesifikasi API yang mengijinkan aplikasi paralel untuk berkomunikasi dengan

dieksekusi dalam *cluster*, super komputer atau *system grid*. Aplikasi paralel ini dapat disebut dengan aplikasi MPI atau MPI job. Jika MPI job membutuhkan n simpul komputasi untuk dieksekusi maka n job ini harus mulai dieksekusi dalam waktu yang bersamaan. Aplikasi lain yang membutuhkan *resource* dari berbagai tempat yang berbeda adalah aplikasi *workflow*, dimana job dalam aplikasi *workflow* harus menunggu job lain sebelum job tersebut dibuat. Hal ini disebut ketergantungan dalam aplikasi *workflow*. Dalam aplikasi multimedia dan aplikasi *real time*, untuk menjamin kelancaran video dan audio

broadcast dalam jaringan, seperti konferensi video, membutuhkan jumlah *bandwidth* tertentu yang tidak mentolelir adanya penurunan *bandwidth* dalam transfer data di jaringan. Dengan contoh dari ketiga aplikasi diatas, maka diperlukan sebuah penjadwalan dengan sistem reservasi untuk menjamin bahwa *resources* tertentu akan siap untuk digunakan oleh aplikasi tersebut saat dibutuhkan [1].

Dalam kebanyakan sistem grid dengan *scheduler* tradisional, *job* yang di diterima akan diletakkan di antrian jika *resource* yang diperlukan tidak tersedia. Setiap sistem grid menggunakan algoritma penjadwalan yang berbeda, sebagai contoh ada yang menggunakan algoritma *First Come First Serve (FCFS)*, *Shortest Job First (SJF)*, *Earliest Deadline First (EDF)*, or *EASY Backfilling* [2], yaitu yang mengeksekusi *job* berdasarkan parameter yang berbeda seperti jumlah *resource*, waktu pengiriman, dan waktu eksekusi. Dengan algoritma-algoritma penjadwalan diatas, maka tidak ada jaminan tentang kapan sebuah *job* akan dieksekusi [3]. *Advance reservation* adalah proses untuk meminta *resource* yang akan digunakan pada waktu tertentu di kemudian hari [1]. *Resource* yang umum untuk direservasi/dipesan adalah mesin komputasi, *bandwidth* jaringan, tempat penyimpanan, dan kombinasi dari ketiganya.

Penjadwalan berdasarkan pemesanan atau reservasi *First Come First Serve-Ejecting Based Dynamic Scheduling (FCFS-EDS)* termasuk pada kategori *Flexible Advance Reservation*, dinamik pada sudut pandang logik [4]. Oleh karena itu akan sulit dipahami dan dibayangkan karena hanya bersifat logik saja. Untuk itulah pada tahun pertama telah dibuat simulasi dari penjadwalan MPI job dengan metode FCFS-EDS, untuk membantu memahami proses penjadwalan MPI job dengan metode FCFS-EDS. Dengan simulasi ini pengguna dapat merubah-ubah parameter dalam penjadwalan dan melihat visualisasi dari penjadwalan dalam sudut pandang logik, dan juga mengetahui bagaimana nantinya sebuah MPI job akan dieksekusi. Disamping itu performansi dari

penjadwalan juga perlu dibuat visualisasinya berupa grafik secara *real time* (waktu nyata).

II. DASAR TEORI

Dalam bagian ini akan dijelaskan mengenai pustaka yang berkaitan dengan penjadwalan MPI *job*, dengan metode metode FCFS-EDS. Disamping itu juga dikaji hasil-hasil penelitian terdahulu yang berkaitan dengan penelitian yang akan dilakukan. Dalam bagian ini akan dikaji pula penelitian yang relevan dengan penjadwalan MPI *Job* pada sistem grid

A. Komputasi Grid

Komputasi grid meliputi hubungan antara alat-alat komputasi yang jumlahnya tak terbatas dan berada dimanapun dalam sebuah jaringan dan juga meliputi layanan jaringan. Pendekatan pada komputasi yang inovatif ini memiliki persamaan dengan jaringan listrik, seperti yang tersedia di rumah kita setiap harinya. Dengan pendekatan ini maka menumbuhkan kekuatan untuk menghubungkan komputer yang terdistribusi secara geografis untuk menyelesaikan masalah aplikasi komputasi intensif, aplikasi data intensif yang bersekala besar. Pendekatan ini menuju pada teknologi yang disebut komputasi jaringan *peer-to-peer* (P2P) *computing* [5] dan Grid [6]. Jaringan listrik sudah menjadi hal yang biasa bagi sebagian besar masyarakat Indonesia. Kita tahu hanya dengan masuk dalam sebuah ruangan dan menghidupkan TV atau alat listrik yang lain seperti kipas angin, maka listrik akan didapatkan oleh alat listrik tersebut pada saat itu juga. Dengan cara yang sama komputasi grid akan mencari alat komputasi ke dalam lingkungan grid. Komputasi grid dapat menambah alat komputasi dengan jumlah yang tak terbatas ke lingkungan grid manapun [7].

Platform komputasi Grid [8][9] dapat melakukan berbagi, pemilihan, dan penggabungan dari *resource* yang heterogen (komputer dan penyimpan data) yang dimiliki oleh organisasi administratif yang berbeda-beda (biasa disebut organisasi virtual), untuk menyelesaikan masalah dengan skala besar dalam bidang ilmu pengetahuan, rekayasa, dan perdagangan.

B. Berbagai macam Grid

Berdasarkan tujuan dan target domain aplikasi, Grid dapat diklasifikasikan ke beberapa tipe [10]:

- *Computational Grid*. *Computational grid* menyediakan fasilitas komputasi terdistribusi untuk mengeksekusi aplikasi yang sangat intensif untuk melakukan komputasi, seperti aplikasi *Bag-of-Tasks* (BoT) [11] dimana tiap aplikasi memiliki tugas yang mandiri, dan Simulasi Monte Carlo [12]. Aplikasi-aplikasi tersebut dijadwalkan pada mesin komputasi yang tersedia oleh proyek MyGrid [13], Nimrod-G [14], and SETI@home [15]
- *Utility grid*. *Utility grid* menyediakan satu atau beberapa layanan grid pengguna akhir sebagai alat-alat IT (*information technology*) berdasarkan pada keharusan membayar bila

akan mengakses utilitas tersebut. *Utility grid* mempunyai kerangka kerja untuk negosiasi dan pengembangan kontrak. Alokasi dari mesin komputasi berdasarkan pada permintaan pengguna. Contoh *utility grid* diantaranya adalah *Utility Data Center* [16], pada tingkat *enterprise* dan Gridbus [17] pada tingkat global.

- *Data grid*. *Data grid* menyediakan infrastruktur untuk mengatur, mentransfer dan mengakses ke *dataset* yang sangat besar yang tersimpan di repositori [18] [19]. Dalam bidang astronomi [19], fisika energi tinggi [20], dan simulasi cuaca, dibutuhkan untuk menganalisis data yang sangat besar dan hasilnya harus didistribusikan. Bidang ini membutuhkan kolaborasi sains yang dapat ditemukan di data grid. Beberapa contoh proyek dalam data grid adalah Avaki EII [21], Biogrid [22], Virtual Observatory [23], dan LCGrid [24]
- *Knowledge grid*. *Knowledge grid* bekerja pada pemrosesan data, manajemen data, dan akuisisi pengetahuan. *Knowledge grid* ini juga menyediakan layanan analisa bisnis yang diintegrasikan dengan layanan data mining. Proyek yang berkenaan dengan knowledge grid diantaranya adalah EU *Data Mining Grid* [25] dan KnowledgeGrid [26]
- *Application service provisioning grid*. *Application service provisioning grid* menyediakan akses ke aplikasi yang berjarak jauh, modul dan pustaka yang disimpan dalam data center atau computational grid. Contohnya adalah NetSolve [27]
- *Interaction grid*. *Interaction grid* menyediakan layanan dan platform kepada pengguna untuk saling dalam system waktu nyata. Contohnya adalah AccessGrid [28]. *Interaction grid* sangat cocok untuk aplikasi multimedia dan aplikasi yang membutuhkan jaringan yang cepat, sebagai contoh adalah video conference.

C. Resource Management dalam Lingkungan Komputasi Grid

Komponen utama dari infrastruktur grid adalah keamanan, manajemen *resource*, layanan informasi, dan manajemen data [29]. Istilah manajemen *resource* dalam grid dapat didefinisikan sebagai operasi yang mengontrol tentang bagaimana caranya agar kemampuan grid dapat digunakan oleh entitas lain seperti pengguna, aplikasi dan layanan [30]. Tujuan dari manajemen *resource* adalah menjamin utilitas yang efisien dari komputer dan untuk mengoptimalkan performa dari task tertentu [31].

Terdapat dua macam manajemen *resource*, yaitu manajemen *resource* global dan manajemen *resource* lokal. Manajemen *resource* lokal menangani penjadwalan dan dan memenej *resource* pada site tertentu atau penyedia *resource* tertentu. Tidak seperti manajemen *resource* lokal, manajemen *resource* global tidak memiliki *resource* pada site tertentu sehingga tidak mempunyai kontrol terhadapnya. Manajer *resource* global harus mengambil keputusan

terbaik untuk memilih resource dan mengirimkan *job* kepadanya [32].

Manajemen *resource* dari grid memiliki beberapa lapis scheduler. Pada *level* yang paling tinggi adalah manajemen resource global yang mungkin punya sudut pandang yang lebih umum dari *resource* tetapi sangat jauh dari *resource* dimana akhirnya sebuah *job* dieksekusi. Pada *level* yang paling rendah, manajemen resource lokal yang memenej sekumpulan *resource* tertentu. Lapisan lain mungkin ada diantara manajemen resource, contohnya adalah manajemen resource yang menangani sekumpulan resource untuk proyek tertentu [33].

Dalam manajemen *resource* lokal, *resource* diakses, dialokasikan dan diberikan bergantung pada kriteria kualitas layanan, seperti reservasi, tenggat waktu dan biaya.

D. MPI Job

Alasan utama dibuat dan di pakainya komputer paralel adalah bahwa paralelisasi adalah cara yang paling ampuh untuk mengatasi masalah leher botol pada aplikasi yang besar bila dieksekusi dengan prosesor tunggal [34]. Ada tiga strategi yang bias digunakan untuk membuat aplikasi paralel. Strategi pertama adalah berdasarkan paralelisasi otomatis, dimana programmer tidak perlu mengurus tentang bagaimana membuat *task* paralel. Strategi kedua menggunakan *library parallel*, dimana kode program yang paralel dan umum digunakan dalam beberapa aplikasi dienkapsulasi dalam *library parallel*. Strategi ketiga adalah mengkode ulang secara awal dalam membuat aplikasi paralel. Programmer bebas memilih bahasa dan model pemrograman yang digunakan untuk membangun aplikasi paralel [35].

Message Passing Interface (MPI) adalah spesifikasi API yang memungkinkan aplikasi paralel untuk berkomunikasi dengan sesamanya dengan saling berkiriman pesan. Hal ini biasanya digunakan oleh program paralel yang dieksekusi dalam cluster atau super computer. Aplikasi paralel ini dapat disebut dengan aplikasi MPI atau *MPI job*. Jika *MPI job* membutuhkan n simpul komputasi untuk dieksekusi maka n *job* ini harus mulai dieksekusi dalam waktu yang bersamaan.

E. Media Pembelajaran

Dasar teori dari pemakaian multimedia dalam pendidikan telah dirangkum oleh [36][37]. Teori ini mengatakan bahwa manusia mempunyai beberapa kanal untuk mengkomunikasikan data. Jika informasi dikomunikasikan dengan dua tau lebih kanal tersebut, maka akan ada peningkatan penguatan, ingatan dan pemahaman [38].

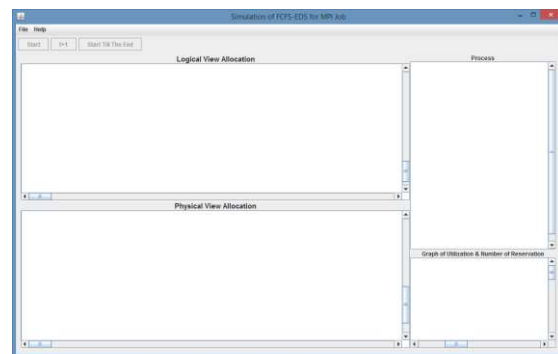
Gaya belajar mempunyai dukungan potensial terhadap penggunaan multimedia. Ada empat macam gaya belajar perasa, penganalisa, pekerja, dan pencipta [39]. In Riding, R. dkk [40] menyarankan penggunaan multimedia untuk ke empat gaya belajar tersebut.

Multimedia menarik perhatian lebih dari satu dari lima indra, dengan memanfaatkan dua indra yang utama untuk menerima informasi, yaitu pendengaran dan penglihatan. Karena gerakan dan suara, dapat juga meningkatkan minat, perhatian, dan motivasi dalam proses belajar [41].

III. HASIL DAN PEMBAHASAN

Penjadwalan berdasarkan pemesanan atau reservasi yang disebut *First Come First Serve - Ejecting Based Dynamic Scheduling* (FCFS-EDS) termasuk pada category *Flexible Advance Reservation*, dinamik, pada sudut pandang logik [4]. Penjadwalan ini mengambil keuntungan dengan menggeser *job* yang sudah terjadwal sebelumnya (tentunya tetap pada batasan yang telah diberikan oleh pengguna) untuk memberikan ruang pada *job* yang akan dijadwalkan dalam sekumpulan *resource*.

Program Simulasi Penjadwalan dengan *First Come First Serve - Ejecting Based Dynamic Scheduling* (FCFS-EDS) telah dibuat dan dipublikasikan dalam [41] namun belum lengkap karena belum terdapat visualisasi performansi dari penjadwalan tersebut. Hasil lengkap dari media pembelajaran untuk Simulasi Penjadwalan dengan *First Come First Serve - Ejecting Based Dynamic Scheduling* (FCFS-EDS) adalah perangkat lunak yang diberi nama *Simulation of FCFS-EDS for MPI Job*. Tampilan Awal dari *Simulation of FCFS-EDS for MPI Job* dapat dilihat pada Gambar 1.



Gambar 1. Tampilan awal Simulasi FCFS-EDS untuk *MPI Job*

Perangkat Lunak ini terdiri dari beberapa bagian, yaitu bagian menu dan bagian visualisasi simulasi..

A. Menu pada Simulasi

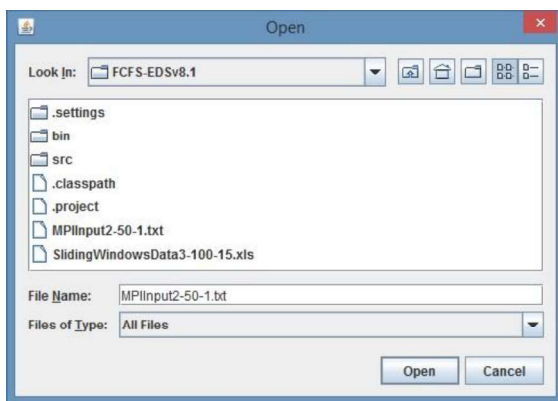
Menu pada *Simulation of FCFS-EDS for MPI Job* terdiri dari dua menu utama yaitu menu *File* dan Menu *Help*. Menu *File* berguna untuk membuka *file* (*Open*), yaitu *file* input dari program simulasi ini yaitu urutan dari *MPI Job* yang datang di *Local Resource Manager* pada tiap *time slot*. Urutan kedatangan *job* ini mengikuti distribusi *Poisson*. Menu *Close*, berguna untuk menutup *file* kedatangan dari *MPI Job* yang telah dibuka dan divisualisasikan, hal ini digunakan bila kita akan mengganti *file* input dengan *file* yang lain. Menu *Exit* digunakan untuk keluar dari Program

Simulation of FCFS-EDS for MPI Job. Menu *File* dapat dilihat pada Gambar 2.



Gambar 2. Menu *File*

Bila menu *Open* pada Gambar 2. di klik maka kita akan diarahkan pada jendela untuk memilih *file* yang akan dibuka yang berisi urutan kedatangan dari MPI Job, yang berupa *file* dengan ekstensi *.txt*. Jendela untuk membuka *file* terlihat pada Gambar 3.



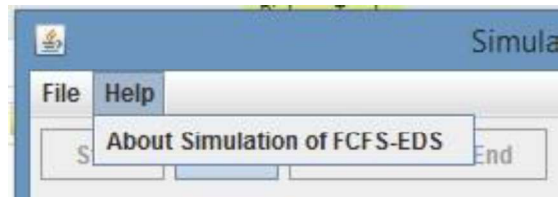
Gambar 3. Jendela untuk membuka *file* input

Bila menu *Exit* pada Gambar 2. di klik maka kita akan diarahkan pada jendela untuk memastikan apakah kita akan keluar dari program Simulasi of FCFS-EDS untuk MPI Job. Jendela *Exit* dapat dilihat pada Gambar 4.



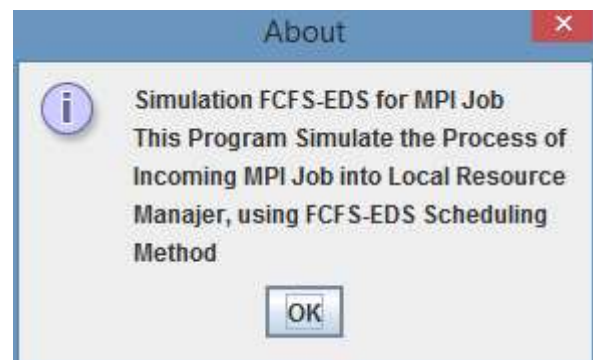
Gambar 4. Jendela dialog *Exit* dari Program

Menu berikutnya adalah menu *Close*, yang berguna untuk menutup *file* input yang berisi urutan kedatangan dari MPI Job yang telah mengikuti distribusi Poison. Bila kita *close file* yang telah dibuka maka kita harus memilih *file* input lain yang akan disimulasikan. Sedangkan Menu *Help* berisi *About Simulation of FCFS-EDS* seperti yang terlihat pada Gambar 5.



Gambar 5. Menu *Help*

Pada Gambar 5. bila menu *About Simulation of FCFS-EDS* diklik maka akan terdapat jendela yang berisi tentang keterangan dari perangkat lunak Simulation of FCFS-EDS untuk MPI Job seperti terlihat pada Gambar 6.

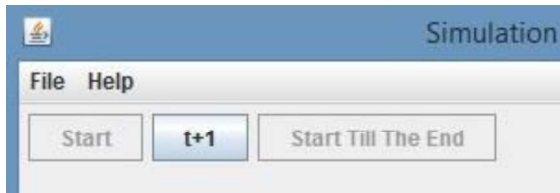


Gambar 6. About Simulation of FCFS-EDS

B. Bagian Simulasi dari FCFS-EDS untuk MPI Job

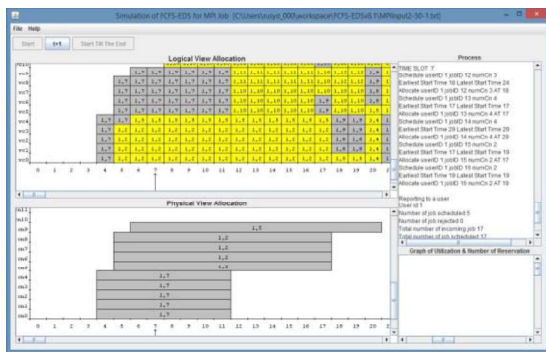
Pada bagian ini terdiri dari 4 bagian. Bagian pertama berguna untuk menampilkan penjadwalan dari FCFS-EDS dalam *Logical View*. Bagian kedua berguna untuk menampilkan penjadwalan FCFS-EDS dalam *Physical View*. Bagian 3 ketiga berisi Proses yang sedang dilakukan oleh sistem penjadwalan. Dibagian ini akan ditampilkan *job* apa saja yang datang pada *time slot* tertentu yang berupa teks. Ukuran *time slot* adalah 5 menit. Disamping itu juga akan ditampilkan dimana *job* bersangkutan akan dialokasikan. Visualisasinya ada di bagian *logical view* dan *physical view*. Bagian terakhir akan menampilkan *graph efisiensi* dari proses penjadwalan seiring dengan jalannya waktu, jadi *graph* dapat digambarkan secara waktu nyata.

Diatas bagian *logical view* terdapat 3 tombol, yaitu *start*, *t+1*, dan *Start till the End* Seperti terlihat pada Gambar 7. Tombol *Start* hanya bisa di klik bila telah ada *file* yang dibuka, bila tidak maka tombol ini masih *disable* (tidak bisa di klik). Bila *file* input telah dibuka maka tombol *Start* dan *Start till the End* akan *enabled* (*active*). Bila tombol *Start* yang di klik, maka kita selanjutnya harus mengklik tombol *t+1*, yaitu untuk melanjutkan ke *time slot* berikutnya. Bila tombol *Start till The End* yang di klik, maka simulasi akan berjalan dari *time slot* pertama sampai *time slot* terakhir.



Gambar 7. Tombol pada Simulation of FCFS-EDS for MPI Job

Contoh simulasi yang telah dijalankan dapat dilihat pada Gambar 8. Dalam Gambar 8. Terlihat pada bagian *logical view MPI Job-MPI Job* yang telah dijalankan. Terlihat bahwa Job dengan JobId 1.7 telah di pesankan di *time slot 4* pada *virtual compute node* (vc0 sampai vc4) dan pada *time slot 5* telah dipesankan pada vc4 sampai vc8. Saat dieksekusi, artinya saat t berada pada time slot 4, maka JobId 1.7 dieksekusi pada *compute node* (cn0 sampai cn4) pada time slot 4 dan seterusnya sampai JobId 1.7 selesai dieksekusi yaitu pada time slot 11.



Gambar 8. Simulasi FCFS-EDS untuk MPI Job

Pada bagian tampilan *graph* efisiensi memperlihatkan performansi dari penjadwalan FCFS-EDS, performansi disini digambarkan dengan persentasi dari utilisasi *computing resource*. Persentasi dari utilisasi *computing resource* dapat dihitung dengan rumus

$$UF = \frac{\text{computing resource terpakai}}{\text{Seluruh computing resource tersedia}} \quad (1)$$

Rumus diatas dihitung pada tiap *time slot*. Faktor utilisasi (utilization factor, UF) dihitung dalam sliding window berukuran 12 time slots (1 jam) sebagai rata-rata dari setiap 12 time slot. Dengan demikian grafik yang digambar menjadi lebih halus. Kemudian dibuat grafiknya seperti terlihat pada gambar 9. Dari Gambar 9. terlihat pada sumbu Y (bahwa persentasi utilisasi maksimal adalah 100%, yang berarti seluruh *computing resource* telah terpakai pada timeslot tertentu, dan 0% berarti tidak ada satupun *computing resource* yang terpakai pada *time slot* tertentu.

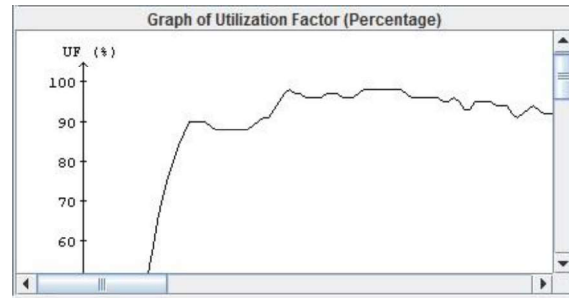


Figure 9. Grafik UF dari FCFS-EDS untuk MPI Job

IV. KESIMPULAN

Setelah dilakukan pengujian, maka didapatkan hasil bahwa program simulasi untuk penjadwalan FCFS-EDS untuk MPI job dapat digunakan untuk media pembelajaran, sehingga semakin mudah untuk memahami langkah-langkah penjadwalan MPI job, walaupun dilaksanakan di sudut pandang logik.

ACKNOWLEDGMENT

Penelitian ini didanai oleh Direktorat Jendral Pendidikan Tinggi Republik Indonesia, dengan nomor kontrak: PPT-076/SP3/LPP-UAD/2017, tanggal 17 April 2017.

REFERENSI

- [1] W. Smith, I. Foster and V. Taylor, "Scheduling with Advanced Reservations," in 14th IEEE International Symposium on Parallel and Distributed Processing, pp. 127–132. IEEE Press, Cancun, 2000.
- [2] A. W. Mu'alem and D. G. Feitelson, "Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling," IEEE Transactions on Parallel and Distributed Systems, vol. 12, pp. 529-543, 2001
- [3] A. Sulistio and R. Buyya, "A Grid simulation infrastructure supporting advance reservation," in 16th International Conference on Parallel and Distributed Computing and Systems, pp. 1–7. ACTA Press, Calgary, 2004
- [4] U. Rusydi, A. Arun and C. R. Rao, "Advance Planning and Reservation in a Grid System," in NDT 2012. CCIS/LNCS, vol. 293, pp. 161-173. Springer, Heidelberg, Dubai, 2012
- [5] A. Oram, Peer-to-Peer: Harnessing the Power of Disruptive Technologies, California: O'Reilly Press, 2001
- [6] R. Buyya and A. Sulistio, "Service and Utility Oriented Distributed Computing Systems: Challenges and Opportunities for Modeling and Simulation Communities," in Proceeding of the 41st Annual Simulation Symposium, 2008 (ANSS-41 '08), pp. 68-81. IEEE Computer Society, Washington DC, 2008
- [7] J. Joshy and F. Craig, Grid Computing, New Jersey: Prentice Hall PTR, 2003
- [8] M. Daniel, A Networking Approach to Grid Computing, Hoboken, New Jersey: John Wiley & Sons, Inc, 2005
- [9] I. Foster and C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure, San Fransisco: Morgan Kaufmann, 1999
- [10] C. S. Yeo, R. Buyya, M. D. de Assuncao and J. Yu, Utility Computing and Global Grids, New York: John Wiley & Sons, 2007
- [11] W. Cirne, D. Paranhos, L. Costa and E. Santos-Neto, "Running Bag-of-Tasks applications on computational grids: the MyGrid approach," in The 32nd International Conference

- on Parallel Processing, 2003 (ICPP 2003), pp. 407-416. IEEE Computer Society, Kaohsiung, Taiwan, 2003.
- [12] D. Abramson, J. Giddy and L. Kotler, "High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?," in The 14th International Parallel and Distributed Processing Symposium (IPDPS'00), pp. 520-528, IEEE Press, Cancun, 2000
- [13] L. B. Costa, L. Feitosa, E. Araujo, C. Mendes, R. Coelho, W. Cirne and D. Fireman, "MyGrid: A complete solution for running bag-of-tasks applications," in 22nd Brazilian Symposium on Computer Networks (SBRC'04), IEEE Press, Brazil, 2004.
- [14] R. Buyya, D. Abramson and J. Giddy, "Architecture for a Resource Management and Scheduling System in a Global Computational Grid," in 4th International Conference & Exhibition on High Performance Computing in Asia-Pacific Region (HPC Asia'00), pp. 283-289, IEEE Press, Beijing, 2000
- [15] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky and D. Werthimer, "SETI@home: an experiment in public-resource computing," *Communications of the ACM*, vol. 45, no. 11, pp. 56-61, 2002
- [16] S. Graupner, J. Pruyne and S. Singhal, "Making the Utility Data Center a Power Station for the Enterprise Grid," Hewlett-Packard Laboratories, Internet and Computing Platforms Research Center, Palo Alto, 2003
- [17] R. Buyya and S. Venugopal, "The Gridbus Toolkit for Service Oriented Grid and Utility Computing: An Overview and Status Report," in 1st International Workshop on Grid Economics and Business Models (GECON'04), pp. 19-66, IEEE Press, Seoul, 2004
- [18] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury and S. Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets," *Network and Computer Applications*, vol. 23, pp. 187-200, 2001.
- [19] W. Hoschek, F. J. Jaen-Martinez, A. Samar, H. Stockinger and K. Stockinger, "Data management in an international data grid project," in 1st International Workshop on Grid Computing (Grid'00), pp. 77-79, Springer-Verlag, London, 2000.
- [20] M. J. Mineter, C. H. Jarvis and S. Dowers, "From stand-alone programs towards grid-aware services and components: a case study in agricultural modeling with interpolated climate data," *Environmental Modelling and Software*, vol. 18, no. 4, pp. 379-391, 2003.
- [21] Avaki EII - Enterprise Data Integration Software, [Online]. Available: http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.avaki_7.0/title.htm.
- [22] Biogrid Project, [Online]. Available: <http://biocodenv.com/biogridarticles/53-biogrid>.
- [23] International Virtual Observatory Alliance, [Online]. Available: <http://www.ivoa.net>.
- [24] LCG Computing Fabric Area, [Online]. Available: <http://lcg-computing-fabric.web.cern.ch>.
- [25] M. Cannataro and D. Talia, "The Knowledge Grid," *Communications of the ACM*, vol. 46, no. 1, pp. 89-93, 2003.
- [26] K. Seymour, A. YarKhan, S. Agrawal and J. Dongarra, "NetSolve: Grid Enabling Scientific Computing Environments," *Grid Computing and New Frontiers of High Performance Processing*, vol. 14, pp. 33-51, 2005.
- [27] K. Seymour, A. YarKhan, S. Agrawal and J. Dongarra, "NetSolve: Grid Enabling Scientific Computing Environments," *Grid Computing and New Frontiers of High Performance Processing*, vol. 14, pp. 33-51, 2005.
- [28] L. Childers, T. Disz, R. Olson, M. E. Papka, R. Stevens and T. Udeshi, "Access Grid: Immersive Group-to-Group Collaborative Visualization," in The 4th International Immersive Projection Technology Workshop, Ames, USA, 2000.
- [29] B. Jacob, L. Ferreira, N. Bieberstein, C. Gilzean, J. Girard, R. Strachowski and S. Yu, *Enabling Applications for Grid Computing with Globus*, USA: IBM Corporation, 2003
- [30] I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*, Los Altos, California: Morgan Kaufmann, 2004
- [31] A. Z. Isah and H. Safwana, "Resource Management in Grid Computing: A Review," *Greener Journal of Science Engineering and Technological Research*, vol. 2, no. 1, pp. 24-31, 2012
- [32] M. S. Jennifer, "Ten Actions When Grid Scheduling: The User as a Grid Scheduler," in *Grid Resource Management: State of the Art and Future Trends*, Norwell, MA, USA, Kluwer Academic Publishers, 2004, pp. 15-24
- [33] S. Uwe and Y. Ramin, "Attributes for Communication Between Grid Scheduling Instances," in *Grid Resource Management: State of the Art and Future Trends*, Norwell, MA, USA, Kluwer Academic Publishers, 2004, pp. 41-52
- [34] K. Hwang and Z. Xu, *Scalable Parallel Computing: Technology, Architecture, Programming*, New York: WCB/McGraw-Hill, 1998
- [35] P. T. Matthew and S. K. W. Johnny, "A Task Migration Algorithm for Heterogeneous Distributed Computing Systems," *System and Software*, vol. 41, pp. 175-188, 1998
- [36] Bagui, S, "Reasons for increased learning using multimedia," *Journal of Educational Multimedia and Hypermedia*, vol 7, pp. 3-18, 1998
- [37] Daniels, L. "Audio vision: Audio-visual interaction in desktop multimedia. Imagery and visual literacy", Selected readings from the annual conference of the international visual literacy association (26th, Tempe, Arizona). pp. 57-63, 1995
- [38] Ellis, T., "Animating to Build Higher Cognitive Understanding: A Model for Studying Multimedia Effectiveness in Education", *Journal of Engineering Education*, Vol. 93, No. 1, pp. 59-64, 2004
- [39] McCarthy, B. "A tale of four learners: 4MAT's learning styles" *Educational Leadership*, Vol 54, No. 1, pp. 46-52, 1997
- [40] Riding, R. & Grimley, M., "Cognitive style, gender and learning from multimedia materials in 11-year old children", *British Journal of Educational Technology*, Vol 30, No 1, pp. 43-56, 1999
- [41] Umar, R., & Pujiyanta, A, "Development of First Come First Serve-Ejecting Based Dynamic Scheduling (FCFS-EDS) Simulation Scheduling Method for MPI Job in a Grid System", *Journal of Engineering and Applied Sciences*, 12(8), 1972-1978., 2017