

OPTIMALISASI ALOKASI BEBAN SECARA DINAMIS PADA LINGKUNGAN SISTEM MULTI AGEN

Muhammad Rizka¹, Ismaniar Isa², Husaini³

^{1,2,3}Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

¹muhammad.rizka910@gmail.com

²isma_isa0508@yahoo.com

³husaini@yahoo.com

Abstrak— Software agent merupakan suatu program komputer yang dapat menangani suatu pekerjaan yang di delegasikan oleh user secara mandiri dan dapat bekerja dalam suatu network environment. Dalam suatu network environment beberapa agen dapat saling berinteraksi dan berkomunikasi dalam mencapai sebuah tujuan (*goal*). Sekumpulan agen yang saling bekerjasama dalam mewujudkan suatu tujuan disebut dengan sistem multi agen. Penanganan load distribution yang tidak optimal ke setiap dapat mengakibatkan unbalancing sistem dan bahkan dapat terjadi deadlock di dalam lingkungan sistem multi agen. Dalam penelitian ini dibangun sistem load balancing agar dapat menangani ketidak seimbangan *workload* diantara agen. Sistem multi agen yang dibangun terdiri dari agen load control dan agen worker. Agen load control bertugas dalam pendistribusian beban kesetiap agen worker secara dinamis berdasarkan kondisi agen worker. Agen worker memiliki tugas dalam menangani setiap beban yaitu melakukan proses enkripsi data yang diberikan oleh agen load control. Hasil pengujian menunjukkan bahwa pendistribusian beban secara dinamis oleh agen load control dapat meningkatkan kinerja sistem multi agen hingga mencapai 30,99 % dibandingkan dengan pendistribusi secara uniform.

Kata kunci— Software agent, load balancing, beban, jaringan, multi agen, optimalisasi.

Abstract— Software agent is a computer program that can handle a job that delegated by user independently and can work in a network environment. In a network environment some agents can interact and communicate in achieving a goal (*goal*). A set of agents that work together to create a goal is called a multi-agent system. Bad handling of load distribution to each can result in unbalancing system and even deadlock may occur within the multi-agent system environment. In this research a load balancing system is built in order to handle workload imbalances between agents. The load-control agent is tasked with distributing the load of each worker agent dynamically based on the condition of the worker agent. Agent worker has a duty in handling any load that is doing the data encryption process provided by the load control agent. The test results show that the dynamic load distribution by the load control agent can improve the performance of multi-agent systems up to 30.99% compared to uniform distributors.

Keywords— Software agent, load balancing, Network, multi agent

I. PENDAHULUAN

Perkembangan *Software agent* yang terus meningkat pesat membutuhkan sebuah kolaborasi efisiensi yang adaptif terhadap suatu aplikasi yang berjalan pada jaringan homogenous maupun heterogenous. Sebuah sistem efisiensi yang dapat menyediakan kostumisasi terhadap suatu perubahan lingkungan. Sistem multi agen merupakan sebuah teknologi yang didesain untuk memenuhi kebutuhan tersebut. Sistem multi agen merupakan sebuah paradigma dalam hal membangun suatu sistem dengan kompleksitas tinggi yang berbasis distributed, knowledge, computing dan adaptif. Sistem multi agen terdiri dari sekumpulan intelligent agent dan resource yang saling berinteraksi untuk mencapai suatu tujuan tertentu. Agen merupakan entitas autonomous yang dapat bertindak proactive dan flexible terhadap suatu lingkungan dalam keadaan tertentu

Dalam pendistribusian beban dapat saja terjadi ketidakseimbangan workload diantara agen didalam suatu network environment. Penelitian yang terkait dilakukan oleh Shin [1]. Metode *load balancing* yang diusulkan pada saat agen mengalami overload yaitu perpindahan agen dan workload ke komputer (*host*) yang lain. Dengan melakukan migrasi agen dan workload dari suatu komputer ke komputer lain memungkinkan terjadinya overload karena adanya penambahan agen dan beban pada komputer tujuan sehingga dapat mengakibatkan terjadi proses reload balancing pada

sistem yang pada akhirnya akan membuat sistem mengalami penurunan kinerja.

Dalam penelitian ini diusulkan suatu mekanisme *load balancing* beban ke setiap agen dengan cara pengontrolan distribusi beban. Pendistribusian beban keseluruhan agent worker pada setiap komputer dilakukan secara dinamis. Sistem multi agen terdiri dari agent load control dan agent worker. Agent worker bertugas sebagai pekerja yang melakukan proses eksekusi tugas. Agent load control bertanggung jawab dalam mengawasi kondisi load agent worker. Load balancing pengalokasian beban ditentukan berdasarkan pertimbangan kondisi *load agent worker*, antrian beban dan resource komputasi komputer dimana agent worker berada. Beban kerja yang diberikan ke agent worker berupa tugas enkripsi data dengan menggunakan algoritma kriptografi AES-128 (*Advanced Encryption Standard*).

II. METODOLOGI PENELITIAN

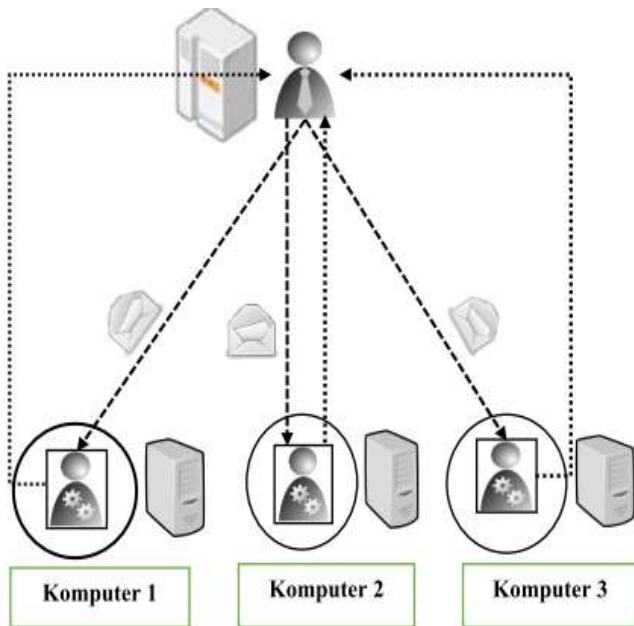
Load balancing yang di usulkan pada penelitian ini yaitu dengan pertimbangan kondisi load, antrian beban dan daya komputasi komputer. Dalam network environment yang dibangun ada dua jenis agen yang dideploy yaitu agent load control dan agent worker. Agent worker yang bertugas dalam melakukan proses enkripsi data sedangkan agent load

control bertanggung jawab dalam memantau kondisi agent worker pada setiap komputer (host) dan selanjutnya mengalokasikan beban ke agent worker. Agent load control melakukan proses pengecekan secara berkala kondisi agent worker yang sedang bertugas mengenkripsi data. Informasi terkait kondisi setiap agent worker dijadikan bahan pertimbangan dalam penentuan jumlah beban yang akan di alokasikan.

A. Kondisi Agent Worker.

Dalam memperkirakan kondisi agent worker, agent load control akan mengirimkan sebuah pesan ACL (Agent Communication Language) secara periodik ke setiap agent worker. Agen *load control* segera mencatat waktu forwarding $F(t)$ pesan yang dikirimkan ke agent worker dan waktu receiving $R(t)$ yaitu pesan balasan dari agent worker. Agent load control mengkalkulasi nilai round trip time (RTT) pesan berdasarkan persamaan 1 berikut ini:

$$RTT = \text{receiving time } R(t) - \text{forwarding time } F(t) \quad (1)$$



Gambar 1. Alur Pesan ACL

B. Antrian Beban

Untuk memperkirakan antrian beban pada agent worker, Agent load control mengirimkan sebuah pesan ACL secara periodik dalam rentang waktu lima detik ke setiap agent worker. Agent worker membalas pesan ACL dengan informasi antrian beban yang sedang terjadi. Antrian beban yang terjadi pada agent worker diklasifikasikan kedalam tiga tahapan yaitu: rendah, sedang, dan padat.

C. Kekuatan Komputasi Komputer (host)

Sistem multi agen dapat terdiri dari beberapa agen yang ditempatkan dalam suatu network environment. Network environment terdiri dari beberapa komputer yang saling terhubung. Komputer yang berada pada network memiliki spesifikasi yang berbeda sehingga akan berdampak

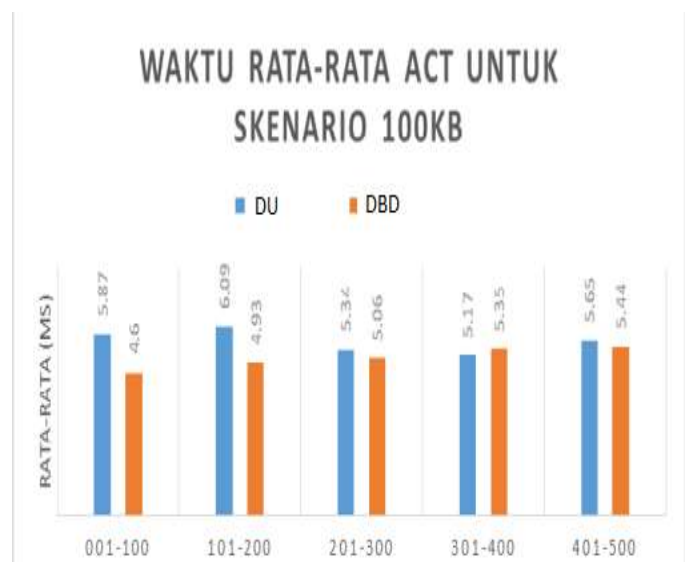
terhadap kinerja agen. Mekanisme alokasi beban juga mempertimbangkan daya komputasi pada komputer agent worker berada. Tingkat komputasi komputer didasari dari nilai million instruction per second (MIPS). Kekuatan komputasi komputer diklasifikasikan kedalam tiga bagian yaitu lambat, sedang dan cepat.

III. HASIL DAN PEMBAHASAN

Dalam pengujian yang dilakukan menerapkan dua skenario pendistribusian beban yaitu secara dinamis dan uniform. Distribusi beban secara dinamis diberi nama distribusi beban dinamis (DBD). DBD mempertimbangkan tiga parameter dalam pengalokasian beban yaitu Round Trip Time (RTT), antrian beban dan kekuatan komputasi. Parameter yang diuji dari kedua skenario tersebut adalah waktu rata-rata Actual Completion Time (ACT).

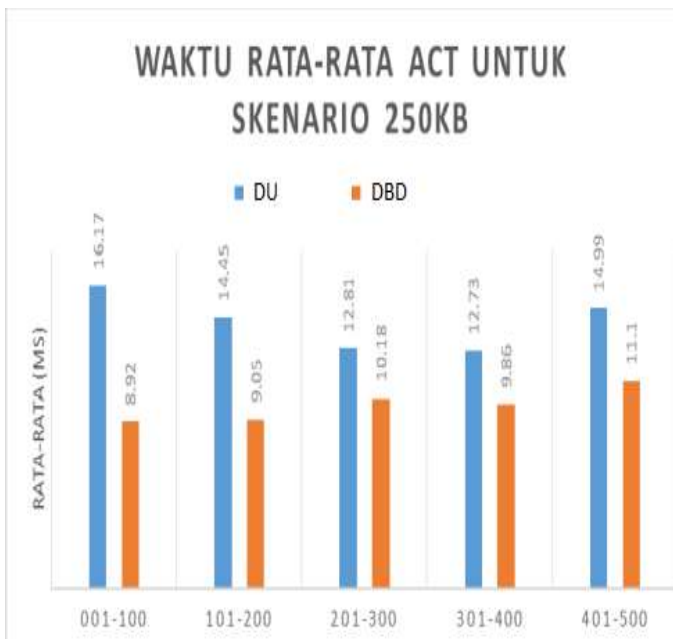
Skenario yang dilakukan pada penelitian ini adalah setiap komputer pekerja hanya memiliki satu agent worker. Setiap agent worker akan dialokasikan 500 job. Pengujian dilakukan dengan lima tahapan yaitu:

1. Sistem Multi agen diuji dengan 500 job dengan ukuran setiap job 100 KB.
2. Sistem Multi agen diuji dengan 500 job dengan ukuran setiap job 250 KB.
3. Sistem Multi agen diuji dengan 500 job dengan ukuran setiap job 500 KB.
4. Sistem diuji dengan 500 job dengan ukuran setiap job 750 KB.
5. Sistem Multi agen diuji dengan 500 job dengan ukuran setiap job 1000 KB.



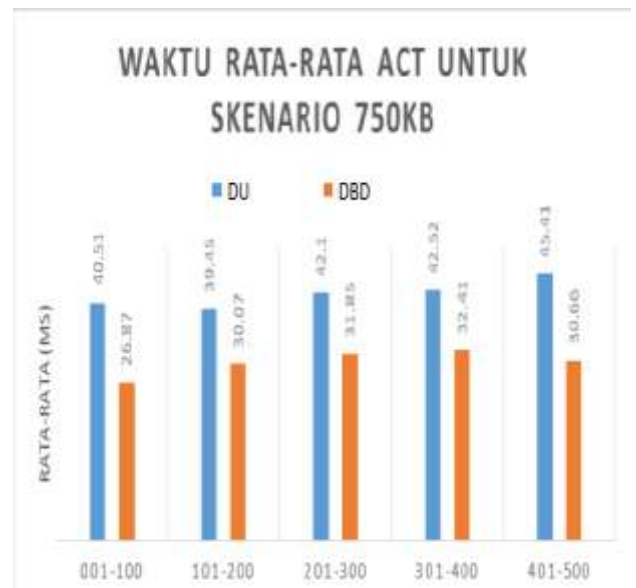
Gambar 2. Hasil Dari Skenario Pertama

Pada Gambar 2 dapat dilihat pada skenario dengan ukuran job 100 KB pada rentang job id 001-100 dan 101-200 terjadi perbedaan nilai ACT yang jelas diantara kedua metode pengujian. Metode DBD memiliki nilai rata-rata ACT lebih rendah dari metode DU.



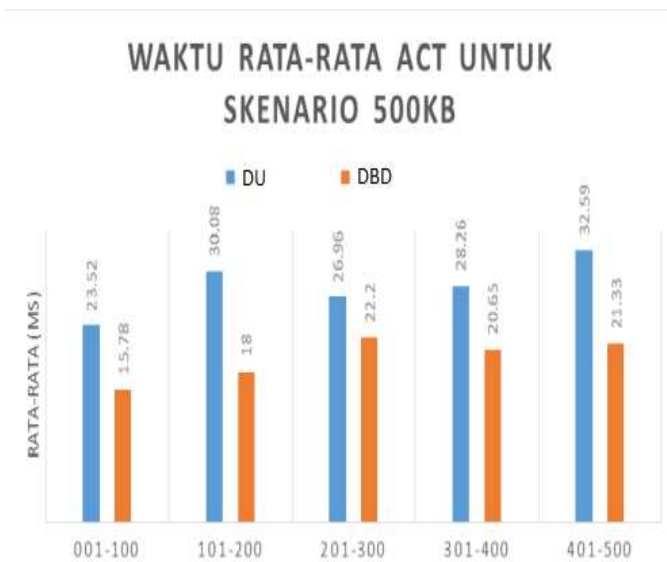
Gambar 3. Hasil Dari Skenario Kedua

Pada Gambar 3 dapat dilihat bahwa skenario dengan ukuran job 250 KB pada semua rentang job id. Metode Distribusi Beban Dinamis(DBD) memiliki nilai rata-rata ACT lebih rendah dari metode DU. Perbedaan waktu yang sangat signifikan terjadi pada 100 job pertama yang di eksekusi.



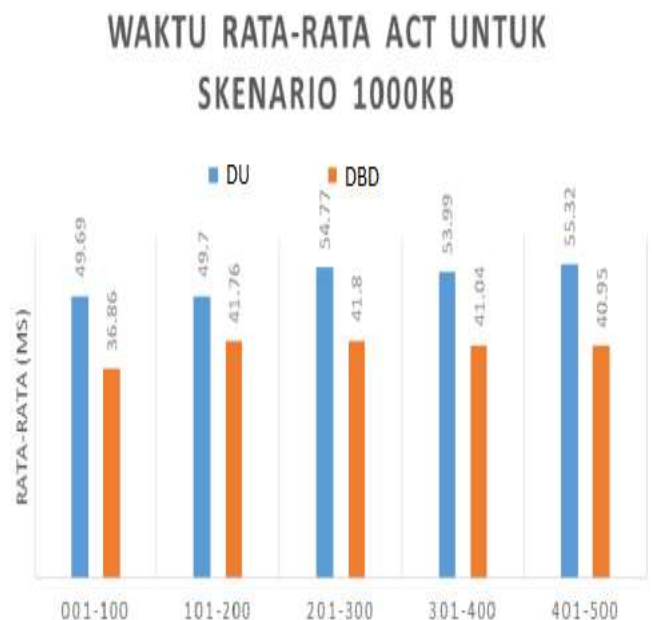
Gambar 5. Hasil dari Skenario Ke-empat

Pada Gambar 5 dapat dilihat bahwa skenario dengan ukuran job 750 KB pada semua rentang job id. Metode Distribusi Beban Dinamis(DBD) memiliki nilai rata-rata ACT lebih rendah dari metode DU.



Gambar 4. Hasil dari Skenario ketiga

Pada Gambar 4 dapat dilihat bahwa skenario dengan ukuran job 500 KB pada semua rentang job id. Metode Distribusi Beban Dinamis(DBD) memiliki nilai rata-rata ACT lebih rendah dari metode DU.



Gambar 6. Hasil dari Skenario Ke-lima

Pada Gambar 6 dapat dilihat bahwa skenario dengan ukuran job 1000 KB pada semua rentang job id. Metode Distribusi Beban Dinamis(DBD) memiliki nilai rata-rata ACT lebih rendah dari metode DU.

IV. KESIMPULAN

Distribusi beban secara dinamis dapat mengurangi waktu Actual Completion Time (ACT) agen worker dalam menyelesaikan suatu task. Distribusi Beban Dinamis (DBD) mempertimbangkan tiga parameter yaitu kondisi load, antrian beban dan kekuatan komputasi komputer dimana agent worker berada. DBD mengaplikasikan fuzzy logic dalam memproses tiga parameter masukan (kondisi load, antrian beban, dan kekuatan komputasi komputer). Penelitian ini menggunakan lima skenario pengujian yaitu 500 job untuk setiap skenario dengan ukuran job mulai dari 100 KB, 250 KB, 500 KB, 750 KB dan 1000 KB. Berdasarkan hasil pengujian yang dilakukan DBD dapat meningkatkan kinerja sistem multi agen dalam network environment. Peningkatan kinerja agen terjadi pada semua skenario pengujian.

REFERENSI

- [1] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
- [2] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
- [3] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.
- [4] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
- [5] (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [6] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/>
- [7] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
- [8] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
- [9] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [10] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
- [11] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.