

ANALISIS KOMPUTASI PARALEL DAN SERIAL PADA ALGORITMA MERGE SORT

Machudor Yusman, Aristoteles* dan Anie Rose Irawati

Jurusan Ilmu Komputer FMIPA Universitas Lampung, Bandar Lampung 35145

*E-mail: aristoteles@unila.ac.id

ABSTRAK

Tulisan ini menjelaskan mengenai penerapan teknologi komputasi paralel pada algoritma *merge sort*. Data yang digunakan pada penelitian ini sebanyak 100 sampai 6 juta data. Data berupa angka yang dibangkitkan secara acak. Pada tahap pengujian, komputasi paralel menggunakan prosesor=2, prosesor=4 (simulasi komputer), dan 2 komputer yang terhubung secara *peer to peer*. Hasil penelitian menunjukkan bahwa jumlah data yang tidak terlalu besar, waktu komputasi serial lebih cepat bila dibandingkan dengan waktu komputasi paralel. Dengan menggunakan dua komputer *peer to peer* jumlah data yang mampu diurutkan mencapai 6 juta data. Nilai Speed Up rata-rata $p=2$, $p=4$, dan komputer=2 adalah 2.273738, 1.980198 dan 1.074942.

Kata kunci: *merge sort*, *peer to peer*, *speed up*

ABSTRACT

This paper describes the application of parallel computing technologies in the merge sort algorithm. The data used in this study as many as 100 to 6 million. Data in the form of randomly generated numbers. In the testing phase, parallel computing processor = 2, processor = 4 (computer simulation), and 2 computers connected in a peer to peer. The results showed that the amount of data that is not too big, serial computation time is faster when compared to the parallel computing time. By using two computer peer to peer the amount of data that can reach 6 million sorted data. The average of Speed Up value of $p = 2$, $p = 4$, and computer = 2 was 2.273738, 1.980198 and 1.074942.

Keywords : *merge sort*, *peer to peer*, *speed up*

1. PENDAHULUAN

Semakin meningkatnya kesadaran peneliti akan manfaat komputer dalam penelitian, menyebabkan semakin banyak usaha untuk menjadikan komputer sebagai alat bantu di berbagai bidang penelitian. Hal ini tidaklah mengherankan mengingat akan terjadi penghematan biaya dan pengurangan resiko besar-besaran ketika menggunakan komputer dibandingkan cara konvensional.

Namun untuk melakukan eksperimen tersebut dibutuhkan sumber daya komputasi yang besar. *Supercomputer* awalnya digunakan untuk mengatasi masalah tersebut. Namun kenyataan bahwa mahalnya harga serta penggunaannya yang harus bergantian menyebabkan solusi ini tidak populer.

Disisi lain kekuatan PC dan kecepatan jaringan bertambah baik. Muncul ide penggabungan beberapa sumber daya tersebut menjadi suatu sumber daya komputasi yang besar yang setara dengan *supercomputer*. Hal ini dikenal dengan komputer *cluster*, solusi ini sangat populer sehingga banyak yang mengadopsinya di departemen masing-masing.

Adanya *cluster-cluster* yang bertebaran melahirkan ide untuk mengintegrasikan sumber daya tersebut menjadi sumber daya komputasi yang lebih besar lagi. Hal ini dikenal sebagai *grid computing* (komputasi *grid*). Istilah *grid* diadopsi dari istilah ketenagalistrikan dimana pembangkit-pembangkit listrik dihubungkan satu dengan yang lain untuk menyediakan kebutuhan pengguna dimana saja.

Teknologi *grid computing* merupakan teknologi yang telah dikembangkan dalam waktu yang panjang. Secara evolusi pengembangan teknologi sejenis mulai dari Condor¹), kemudian diikuti oleh Paralel Virtual Machine²) dan Message Passing Interface³) sampai dengan Globus Toolkit⁴). Sejak awal, para peneliti di bidang komputasi berkinerja tinggi telah menggunakan dua pendekatan⁵) yaitu *supercomputer* dan *multicomputer*.

Supercomputer yaitu membangun sebuah komputer dengan teknologi perangkat keras berkinerja tinggi. *Multicomputer* yaitu membangun sebuah sistem komputer dengan teknologi jaringan interkoneksi dan

perangkat lunak. Pendekatan pertama umumnya menghasilkan sebuah komputer yang berkinerja tinggi, tetapi berharga amat mahal sehingga hanya dapat dimiliki oleh segelintir pihak saja. Pendekatan kedua menghasilkan suatu sistem komputer yang kinerjanya bervariasi sesuai jumlah komputer yang tergabung dan konfigurasi perangkat lunak yang digunakan.

Harga untuk membangun *multicomputer* lebih terjangkau daripada *supercomputer*, namun pemakaiannya terbatas. Hal ini karena sistem komputer yang berbasis jaringan umumnya diterapkan pada komputer-komputer yang terhubung dalam suatu jaringan lokal (LAN). Salah satu penyebabnya adalah masalah keamanan jaringan yang belum tertangani dengan baik. Selain itu, sistem perangkat lunak pendukung yang memungkinkan komputer-komputer tersebut bekerja sebagai satu kesatuan umumnya memiliki konfigurasi yang kompleks sehingga pemakai harus memiliki keahlian khusus sebelum dapat memanfaatkan sistem komputer tersebut.

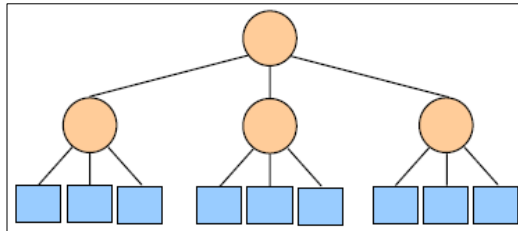
Teknologi komputasi *grid* adalah suatu cara penggabungan sumber daya yang dimiliki banyak komputer yang terhubung dalam suatu jaringan sehingga terbentuk suatu kesatuan sistem komputer dengan sumber daya komputasi yang besar. Menurut⁴⁾, suatu *grid* harus memenuhi tiga hal yaitu:

1. Mengkoordinasi sumber daya-sumber daya yang tidak dikendalikan secara terpusat.
2. Menggunakan protokol dan antarmuka yang standard, terbuka dan dapat digunakan untuk berbagai keperluan (general purpose).
3. Memberikan *Quality of Service* (QoS) yang tidak Trivial, yaitu tergantung kebutuhan pengguna.

2. METODE PENELITIAN

2.1. Algoritma Divide and Conquer

Algoritma divide and conquer sudah lama diperkenalkan sebagai sumber dari pengendalian proses paralel, karena masalah-masalah yang terjadi dapat diatasi secara independen. Banyak arsitektur dan bahasa pemrograman paralel mendesain implementasinya (aplikasi) dengan struktur dasar dari algoritma divide and conquer. Untuk menyelesaikan masalah-masalah yang besar, dan dibagi (dipecah) menjadi bagian yang lebih kecil dan menggunakan sebuah solusi untuk menyelesaikan problem awal adalah prinsip dasar dari pemrograman/strategi divide and conquer. Ilustrasi dengan teknik *divide* dan *conquer* terlihat pada Gambar 1.



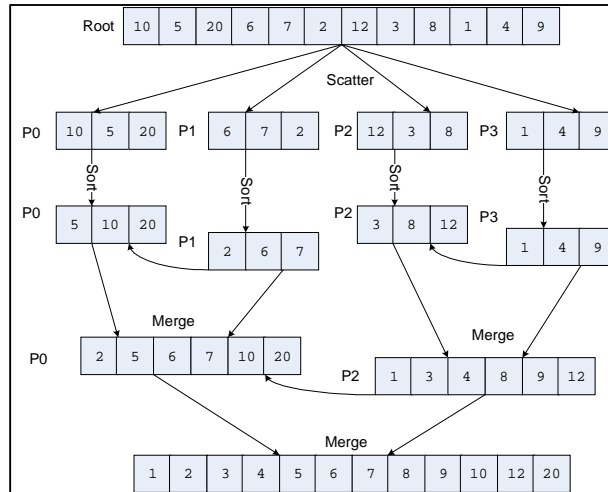
Gambar 1. Ilustrasi pemecahan masalah dengan divide dan Conquer

Divide and conquer adalah varian dari beberapa strategi pemrograman topdown, tetapi keistimewaannya adalah membuat sub-sub problem dari problem yang besar, oleh karena itu strategi ini ditunjukkan secara berulang-ulang (*recursively*), didalam menerapkan algoritma yang sama dalam sub-sub problem seperti yang diterapkan pada masalah aslinya (*original problem*). Sebagaimana prinsip dasar algoritma perulangan dibutuhkan sebuah kondisi untuk mengakhiri perulangan tersebut. Biasanya untuk mengecek apakah problem sudah cukup kecil untuk diselesaikan dengan metode secara langsung. Mungkin dari segi ilustrasi kita, bahwa proses-proses pada komputer paralel tentunya memiliki proses/*problem/job* yang cukup kompleks sehingga harus dipecah-pecah menjadi sub-sub problem. Salah satu penerapan algoritma *divide and conquer* adalah pengurutan data dengan metode *merge*.

Metode penelitian yang digunakan untuk eksperimen komputasi paralel yaitu algoritma *merge sort* paralel. Penelitian ini mengurutkan data sebanyak n bilangan yang dibuat dari bilangan *random* atau acak. Berikut ini ilustrasi atau alur cara kerja algoritma *merge sort* paralel dengan beberapa tahapan yakni :

1. master (induk) membangkitkan bilangan acak sebanyak n bilangan integer.
2. master (induk) mendistribusikan n/p data ke tiap prosesor.
3. untuk setiap proses melakukan sorting lokal.
4. hasil sorting dari setiap proses dilakukan *merge* ke master (induk)

Ilustrasi atau alur cara kerja algoritma *merge sort parallel* dapat dilihat pada Gambar 2.



Gambar 2. Ilustrasi proses pengurutan secara paralel menggunakan metode merge

3. HASIL DAN PEMBAHASAN

3.1. Pengujian

Pada bagian ini akan disampaikan mengenai hasil pengujian dan analisa kedua program yakni pengurutan dengan menggunakan Algoritma Merge Sort baik yang dilakukan secara serial dan secara paralel. Untuk melakukan pengujian digunakan suatu skenario pengujian yang dapat membandingkan hasil dari kedua program tersebut. Detail mengenai skenario pengujiannya dijelaskan sebagai berikut.

3.2. Spesifikasi Hardware

Dalam penelitian ini, pengujian proses paralel dilakukan pada satu komputer dengan spesifikasi seperti pada Tabel 1.

Tabel 1. Spesifikasi satu komputer

Komponen	Spesifikasi
Processor	Intel Core2 Dua 2,1 GHz
RAM	2 GB
Sistem Operasi	Linux

Di samping itu, dilakukan pula pengujian spesifikasi dua komputer dengan spesifikasi masing-masing seperti pada Tabel 2.

Tabel 2. Spesifikasi dua komputer

Komponen	Spesifikasi
Komputer 1	
Processor	Intel Core2 Dua 2,1 GHz
RAM	2 GB
Sistem Operasi	Linux
Komputer 2	
Processor	Intel Pentium 4 Dual Core 1.6 GHz
RAM	1 GB
Sistem Operasi	Linux

3.3. Skenario Pengujian

Pengujian dilakukan dengan membandingkan nilai Speed Up dan Efisiensi dari sejumlah prosesor dengan banyaknya data yang diurutkan. Pengujian juga dilakukan dengan memperlihatkan hasil pengujian dengan menggunakan satu komputer dan dua komputer. Pada pengujian satu komputer dilakukan dengan

melakukan simulasi terhadap dua prosesor dan empat prosesor. Sedangkan pengujian lainnya dilakukan benar-benar menggunakan dua komputer yang terhubung pada suatu jaringan *peer to peer*.

Pada pengujian dengan dua komputer ini, setiap PC/Notebook terinstall Sistem Operasi Linux secara virtual dengan menggunakan Virtual Box versi 2.2. Adapun skenario pengujian yang dilakukan dapat dilihat pada Tabel 3.

Tabel 3. Skenario Pengujian

No	Jumlah Processor (p)	Banyak Data (n)	Parameter Analisis	
1	p = 2	10, 100, 1000, 10000, 100000, 1000000, 2000000, 4000000, 5000000, 5200000		
2	P = 4	10, 100, 1000, 10000, 100000, 1000000, 2000000, 4000000, 5000000, 5200000, 5300000, 5350000	Speed Up (S)	Efisiensi (E)
3	Komp = 2	10, 100, 1000, 10000, 100000, 1000000, 2000000, 4000000, 5000000, 5200000, 5300000, 5350000, 5500000, 6000000		

3.4. Hasil Pengujian

Hasil pengujian dilakukan dengan menghitung nilai Speed Up dan Efisiensi penggunaan pemrograman secara serial dan paralel. Waktu eksekusi secara serial dan paralel ini yang akan dijadikan sebagai parameter untuk mendapatkan nilai Speed Up dan Efisiensi Algoritma Merge Sort secara paralel. Perbandingan nilai Speed Up dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian dengan dua prosesor (p = 2)

Banyak Data (n)	(Tp)	(TS)	Speed Up (S) Ts/Tp	Efisiensi (S/p)
100	0.009781	0.00006	0.006134	0.003067
1.000	0.01884	0.000447	0.023726	0.011863
10.000	0.022327	0.005258	0.2355	0.11775
100.000	0.070323	0.062971	0.895454	0.447727
1.000.000	0.904371	0.998713	1.104318	0.552159
2.000.000	2.295663	5.094314	2.219104	1.109552
4.000.000	6.246864	40.72526	6.519313	3.259656
5.000.000	9.598989	50.29295	5.239401	2.619701
5.200.000	12.55685	52.99855	4.22069	2.110345
Rata-rata	3.52489	16.6865	2.273738	1.136869

3.5. Analisis Pengujian dengan Dua Prosesor

Pada Tabel 3 dapat dilihat bahwa untuk jumlah data 100 sampai dengan 100.000 waktu komputasi secara serial masih lebih kecil. Namun, secara umum nilai Speed Up dari jumlah data tersebut makin meningkat. Ini berarti bahwa proses paralel yang dilakukan memiliki hasil yang lebih baik seiring dengan bertambahnya jumlah data.

Pada jumlah data 1.000.000 sampai dengan 5.000.000, waktu komputasi secara paralel jauh lebih kecil dibandingkan dengan waktu komputasi serial. Dengan waktu komputasi ini, akan memberikan pengaruh pada nilai Speed Up yang cenderung meningkat dari mulai 1.104318 pada data 1.000.000 sampai 6.519313 pada jumlah data sebanyak 4000.000.

Dengan bertambahnya jumlah data, ternyata tidak selalu meningkatkan nilai Speed Up apabila jumlah prosesor yang digunakan selalu tetap. Pada jumlah data sebanyak 5.000.000 terlihat bahwa Speed Up mengalami penurunan dari 6.519313 menjadi 5.239401. Terjadinya penurunan ini diakibatkan oleh peningkatan jumlah data sebesar 1000.000 data yang berimbas pada peningkatan waktu komputasi paralel

sebesar 3.352125 dan sedangkan waktu komputasi serial meningkat sebesar 9.56769. Apabila penambahan jumlah data ini terus dilakukan dengan jumlah prosesor tetap, maka dapat diperkirakan bahwa nilai Speed Up akan terus mengalami penurunan.

3.6. Analisis Pengujian dengan Empat Prosesor

Pada Tabel 5 dapat dilihat bahwa untuk jumlah data 100 sampai dengan 1000.000 waktu komputasi secara serial masih lebih kecil. Namun, secara umum nilai Speed Up dari jumlah data tersebut makin meningkat.

Ini berarti bahwa proses paralel yang dilakukan memiliki hasil yang lebih baik seiring dengan bertambahnya jumlah data. Nilai Speed Up pada penggunaan empat prosesor cenderung meningkat dengan bertambahnya jumlah data yang diurutkan sampai pada jumlah data 5.200.000. Setelah itu, nilai Speed Up mengalami penurunan pada jumlah data lebih dari sama dengan 5.300.000. Ini berbeda dengan hasil yang diperoleh pada pemrosesan dengan dua prosesor. Berdasarkan pengujian yang dilakukan, penggunaan dua prosesor tidak mampu mengeksekusi sampai dengan jumlah data lebih dari 5.200.000.

Bila dibandingkan dengan menggunakan dua prosesor sampai dengan jumlah data 5.200.000, secara umum nilai rata-rata Speed Up dari penggunaan empat prosesor adalah sebesar 1.980198, lebih kecil dari rata-rata nilai Speed Up dua prosesor sebesar 2.273738. Hal ini diperkirakan dipengaruhi oleh waktu komunikasi yang lebih banyak pada penggunaan empat prosesor dibandingkan dengan dua prosesor dengan jumlah data yang sama.

Tabel 5. Hasil Pengujian dengan empat prosesor ($p = 4$)

Banyak Data (n)	(Tp)	(TS)	Speed Up (S) Ts/Tp	Efisiensi (S/p)
100	0.028786	0.000077	0.002675	0.000669
1.000	0.026555	0.000469	0.017661	0.004415
10.000	0.050265	0.00524	0.104247	0.026062
100.000	0.132036	0.065502	0.496092	0.124023
1.000.000	0.867007	0.725553	0.836848	0.209212
2.000.000	1.798548	3.652118	2.030592	0.507648
4.000.000	5.440498	23.50407	4.320206	1.080051
5.000.000	12.60952	59.75159	4.738609	1.184652
5.200.000	12.84354	67.74773	5.274851	1.318713
Rata-rata sampai jumlah data 5.200.00	3.755195	17.27248	1.980198	0.495049
5.300.000	14.70851	58.05524	3.947051	0.986763
5.350.000	18.85945	59.90778	3.176539	0.794135
Rata-rata sampai jumlah data 5.350.000	6.124065	24.85594	2.267761	0.56694

3.7. Analisis Pengujian dengan Dua Komputer

Pada pengujian proses pengurutan dengan menggunakan dua komputer, terlihat bahwa jumlah data yang mampu untuk dijalankan dan dieksekusi lebih banyak dibandingkan dengan dua hasil sebelumnya yang menggunakan satu komputer. Ini menunjukkan bahwa penggunaan sumber daya secara nyata memiliki kinerja yang lebih baik bila dibandingkan dengan satu komputer yang dijadikan seakan akan sebagai dua prosesor yang berlainan.

Nilai rata-rata dari Speed Up dari penggunaan dua komputer adalah 1.074. Nilai ini memang lebih kecil dibandingkan dengan hasil pengujian pada dua skenario sebelumnya yaitu 2.273 dan 1.980. Nilai ini sangat dipengaruhi oleh *running time* yang dimiliki oleh proses paralel dan serial yang dihasilkan skenario ini yang juga lebih kecil dibandingkan dengan dua skenario sebelumnya.

Namun, kelebihan dari skenario ini adalah kemampuan dalam menyelesaikan jumlah data yang lebih banyak sampai dengan 6.000.000 data lebih banyak dari dua skenario sebelumnya. Di sisi lain, rata-rata waktu komputasi paralel pada dua komputer ini lebih kecil bila dibandingkan dengan skenario $p=4$ yaitu sebesar

3.684742. Namun nilai sedikit lebih besar bila dibandingkan dengan skenario $p=2$ yang memiliki rata-rata 3.52489. Hasil pengujian dengan dua komputer terlihat pada Tabel 6.

Tabel 6. Hasil Pengujian dengan dua komputer

Banyak Data (n)	(Tp)	(TS)	Speed Up (S)	Efisiensi
			Ts/Tp	(S/p)
1.00	0.013607	0.00012	0.00881899	0.0044095
1.000	0.055189	0.000435	0.007882005	0.003941
10.000	0.067543	0.0022393	0.033153695	0.0165768
100.000	0.222993	0.063866	0.286403609	0.1432018
1.000.000	1.176509	0.802954	0.682488617	0.3412443
2.000.000	2.239071	2.036682	0.90961028	0.4548051
4.000.000	6.010457	15.118139	2.515306074	1.257653
5.000.000	13.566412	32.640861	2.406005435	1.2030027
5.200.000	9.810895	27.713888	2.824807319	1.4124037
Rata-rata sampai jumlah data 5.200.000	3.684742	8.708798	1.074942	0.537471
5.300.000	8.830886	26.338159	2.9825047	1.4912523
5.350.000	8.186201	21.779998	2.660574545	1.3302873
5.500.000	8.455643	24.794032	2.932246785	1.4661234
6.000.000	12.766057	36.337325	2.846401594	1.4232008
Rata-rata sampai jumlah data 6.000.000	5.49242	14.43298	1.622785	0.811392

4. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan dapat disimpulkan bahwa untuk jumlah data yang tidak terlalu besar, waktu komputasi serial justru berjalan lebih cepat bila dibandingkan dengan waktu komputasi paralel. Hal ini disebabkan oleh waktu komunikasi untuk paralel yang menjadi *overhead*. Di sisi lain, skenario pengujian dengan dua komputer ternyata memberikan kehandalan dari sisi jumlah data yang mampu dieksekusi. Hasil pengujian menunjukkan bahwa dengan menggunakan dua komputer yang secara *peer to peer* jumlah data yang mampu diurutkan mencapai 6.000.000 data. Adapun, skenario dengan dua prosesor dan empat prosesor masing-masing hanya mampu mengurutkan data sampai dengan 5.200.000 dan 5.350.000 data. Dari ketiga skenario program yang dieksekusi didapatkan bahwa nilai Speed Up rata-rata dari ketiga skenario masing-masing adalah 2.273738, 1.980198 dan 1.074942. Dengan demikian, secara umum dapat disimpulkan bahwa proses komputasi untuk Algoritma Merge Sort yang dilakukan memiliki keberhasilan dalam menyelesaikan masalah pengurutan dengan algoritma ini.

DAFTAR PUSTAKA

1. Litzkow, M., Livney, M. & Mutka, M. 1988. Condor - a hunter of idle workstations. in Proc. 8th Int'l Conf. on DCS.
2. Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. & Sunderam, V.S. 1994. PVM: Parallel Virtual Machine : A User's Guide and Tutorial for Network Parallel Computing. MIT Press.
3. Gropp, W., Lusk, E. & Skjellum, A. 1995. Using MPI: Portable Parallel Programming with the Message Passing Interface. MIT Press.
4. Foster, I. & Kesselman, C. 1998. Globus: A Metacomputing Infrastructure Toolkit. *Int. J. Supercomput. Appl.*, **11** (2): 1-16.
5. Reed, D. & Fujimoto, R. M. 1987. Multicomputer Networks: Message-Based Parallel Processing. MIT Press.