Penerapan Algoritma Divide and Conquer Dalam Komputasi Paralel

Ivan Andrianto - 13513039
Program MagisterInformatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
andrianto.ivan@yahoo.co.id

***Abstract*— Komputer merupakan teknologi yang sering kita jumpai dalam kehidupan sehari-hari. Salah satu teknologi paling menarik sejak ditemukannya komputer pada tahun 1940-an adalah Komputasi Paralel. Pemrosesan paralel selalu berkembang dan mendapat perhatian bagi mereka yang bergerak di bidang pemrograman dengan melakukan banyak terobosan. Perkembangan Komputasi Paralel pada tahun 1990-an mengubah persepsi terhadap komputer. Perkembangan teknologi lainnya sangat berpengaruh terhadap perkembangan Komputasi Paralel.**

Komputasi paralel dapat mempersingkat penyelesaian suatu program dengan cara membagi suatu program menjadi bagian yang lebih kecil agar dapat dikerjakan secara bersamaan oleh masing-masing prosesor. Pemecahan masalah pada paralel tersebut memanfaatkan prinsip divide and conquer

***Index Terms*— Divide and Conquer, komputasi paralel, komputer**

I. PENDAHULUAN

Komputer merupakan suatu teknologi yang umum diguanakn pada saat ini, mulai di perusahaan untuk mendukung proses bisnis suatu perusahaan, pemerintahan, berbagai jenis lembaga, maupun digunakan secara perorangan. Penggunaan komputer dapat mempermudah suatu pekerjaan yang cukup rumit untuk dilakukan oleh manusia tanpa bantuan komputer. Misalnya, proses menghitung bilangan yang cukup besar, mengedit dokumen menjadi mudah dilakukan berkat keberadaan komputer.

Salah satu aspek yang penting dalam sebuah komputer adalah performa. Setiap komputer dapat memerlukan waktu yang berbeda untuk melakukan suatu tugas yang sama. Hal itu dapat dipengaruhi oleh spesfikasi hardware dari masing-masing komputer, algortima yang diterapkan dalam program, dan lain-lain. Kebutuhan akan komputasi yang cepat sangat diperlukan terutama untuk menyelesaikan proses yang kompleks dan memerlukan waktu yang lama. Proses optimasi sangat diperlukan dalam kasus tersebut.

Salah satu cara adalah dengan menggunakan beberapa prosesor sekaligus sehingga suatu pekerjaan bisa dibagi untuk dilakukan oleh setiap prosesor yang terdapat pada suatu sistem. Dengan demikian, diharapkan suatu persoalan dapat diselesaikan dengan lebih cepat. Teknik tersebut disebut komputasi paralel.

Komputasi Paralel adalah salah satu teknik pemrograman komputer secara bersamaan dengan beberapa komputer, baik dalam satu atau banyak prosesor. Hal ini digunakan apabila kapasitas yang diperlukan sangat besar, misalnya tuntutan untuk mengolah data dalam jumlah yang sangat besar atau untuk proses komputasi yang cukup banyak.

Pemrograman paralel adalah teknik pemrograman komputer yang memungkinkan untuk dilakukannya eksekusi perintah/operasi secara bersamaan. Untuk melakukan aneka jenis komputasi paralel ini diperlukan berbagai macam perangkat lunak yang berperan untuk mengatur distribusi pekerjaan dalam satu mesin paralel. Dengan mesin paralel semua program yang dijalankan diatasnya tidak secara otomatis akan diolah secara paralel, tetapi pemakai harus melakukan pemrograman paralel
.

Alogaritma Divide and Conquer merupakan algoritma dengan prinsip memecah-mecah persoalan yang dianggap terlalu besar menjadi beberapa persoalan yang lebih kecil dan lebih mudah untuk diselesaikan.

Divide and Conquer merupakan suatu varian dari beberapa strategi pemrograman yang cukup baik terutama untuk memecahkan permasalahan yang cukup besar atau kompleks. Strategi ini dilakukan secara berulang-ulang dalam menerapkan alogaritma yang sama dalam bagian-bagian masalah seperti pada masalah awalnya.

Komputer paralel memiliki proses-proses yang cukup kompleks sehingga harus dipecah-pecah menjadi bagian-bagian masalah. Algoritma divide dan conquer diperkenalkan sebagai sumber dari pengendalian proses paralel yang cukup rumit hal ini karena masalah-masalah yang terjadi dapat diatasi secara independen. Pemrograman akan menjadi lebih sederhana jika masalah

dapat dipecah-pecah menjadi bagian yang kecil supaya mudah diselesaikan.

Komputasi paralel mempunyai prinsip bersesuaian dengan alogaritma Divide and Conquer, yaitu membagi-bagi proses menjadi bagian-bagian yang lebih kecil dan memungkinkan dikerjakan oleh sebuah unit komputasi.

Ada 2 klasifikasi utama komputasi paralel, yaitu:
1. Sebuah komputer dengan banyak unit komputasi internal yang dikenal sebagai Shared Memory Multiprosessor.
2. Beberapa komputer yang terhubung melalui sebuah jaringan yang dikenal sebagai Distributed Memory Multicomputer.

II. LANDASAN TEORI

Divide and Conquer merupakan pendekatan algoritma yang dilakukan dengan cara membagi permasalahan ke menjadi berukuran lebih kecil, sehingga dapat dipecahkan secara terpisah dan lebih mudah. Suatu masalah dipecah menjadi beberapa sub-masalah yang mirip namun dengan ukuran yang lebih kecil. Selanjutnya, sub-masalah dapat dipecahkan secara rekursif (perulangan). Hasil pemecahan dari setiap sub-masalah selanjutnya akan digabungkan kembali sehinga kita memperoleh solusi masalah secara keseluruhan. Pendekatan *divide and conquer*. Terdapat tiga langkah umum pada setiap tingkat perulangan, yaitu *Divide, Conquer, and Combine*.

Tiga langkah utama dalam algoritma divide dan conquer adalah :

1. *Divide* (Memecah)
Tahap *divide* merupakan tahap memecahkan sutau masalah yang dianggap masih terlalu besar sehingga menjadi lebih kecil. Biasanya dilakukan dengan menggunakan rekursif sampai ukuran dari masing-masing sub-masalah dianggap cukup kecil.

2. *Conquer* (Menaklukkan)
Masalah yang telah dipecahkan pada tahap pertama akan diselesaikan pada tahap *conquer*. Karena telah dibagi, pemecahan suatu sub-masalah menjadi lebih sederhana dibandingkan memecahkan masalah yang tidak dibagi.

3. *Combine* (Menggabungkan)
Hasil dari masing-masing pemecahan pada tahap divide yang telah diproses pada ahap conquer digabungkan pada tahap *combine* untuk memperoleh hasil akhir

Selain itu, terdapat empat hal penting yang harus dipahami dalam strategi divide dan conquer :

1. Branching Factor

Dalam suatu algoritma divide dan conquer, brancing factor merupakan jumlah dari sub masalah yang akan dibagi dari sebuah masalah awal. Ini adalah langkah nyata dari algoritma divide dan conquer, dalam pemecahan masalah menjadi beberapa bagian. Jumlah minmum branching factor adalah 2. Jika tidak, masalah tidak dapat dibagi.

2. Balance
Kondisi balance tercapai jika masalah awal persoalan divide and conquer dibagi menjadi sub-sub masalah yang memiliki ukuran sama. Artinya, jumlah dari keseluruhan sub masalah sama dengan masalah awal.

3. Data Dependence of Divide Function
Algoritma divide dan conquer menggunakan fungsi pembagian terhadap data yang memiliki ketergantungan. Hal itu berarti ukuran relatif dari sebuah sub masalah tergantung proses input datanya. Sebagai contoh, salah satu penerapan algoritma divide and conquer yant tidak seimbang adalah quicksort.

4. Control Parallelism of Sequentiality
Suatu Algoritma divide and conquer dikatakan berurutan apabila sub masalah di eksekusi sesuai dengan urutan perintah program. Parelisasi dari algoritma divide dan conquer yang terurut pertama kali didefinisikan oleh Mou's

Contoh umum dari penggunaan pendekatan ini dalah dalam algoritma Merge Sort. Pada algoritma Merge Sort, banyaknya angka yang membagi banyak angka menjadi dua bagian (Langkah Divide). Kemudian angka-angka tersebut dirutukan pada tiap sub-masalahnya (Langkah Conquer). Setelah proses mengurutkan angka-angka tersebut selesai, gabungkan dengan sub-masalah lainnya sehingga terbentuk solusi umum(Langkah Combine).

Keuntungan menggunakan pendekatan Divide and Conquer adalah sebagai berikut :
1. Dapat memecahkan masalah yang sulit.
Memecahkan masalah *Divide and conquer* merupakan cara yang sangat efektif jika masalah yang akan diselesaikan cukup rumit.

2. Memiliki efisiensi algoritma yang tinggi.
Pendekatan Divide and Conquer juga memiliki efisiensi algoritma yang cukup tinggi. Sebagai contoh, terdapat masalah dengan ukuran n memiliki sub-masalah p dengan ukuran n/p pada tiap perulangannya. Dengan demikian, kompleksitas algoritma untuk ukuran waktu konstan O akan menjadi O (n log n)
Algortima sekuensialuntuk sorting memiliki kompleksitas O (n^2). Dapat disimpulkan bahwa pendekatan *divide and cconquer* ini lebih efisien dalam menyelesaikan algoritma sorting.

3. Dapat bekerja secara paralel.
Divide and Conquer telah didisain untuk dapat bekerja dalam mesin-mesin yang memiliki banyak

prosesor. Terutama mesin yang memiliki sistem pembagian memori, dimana komunikasi data antar prosesor tidak perlu direncanakan terlebih dahulu, hal ini karena pemecahan sub-rutin dapat dilakukan di prosesor lainnya.

4. Akses memori yang cukup kecil.

Untuk akses memori, Divide and Conquer dapat meningkatkan efisiensi memori yang ada cukup baik. Hal ini karena, sub-rutin memerlukan memori lebih kecil daripada masalah utamanya. Disamping itu, Divide and Conquer hanya memerlukan satu bagian memori untuk menyelesaikan sub-rutin sub-rutin tersebut (karena berupa perulangan dan sudah memiliki pesanan memori untuk variabel-variabelnya).

Selain memiliki kelebihan, algoritma *Divide and Conquer* juga memiliki beberapa kekurangan yaitu:

1. Lambatnya proses perulangan

Proses pemanggilan sub-rutin yang berlebih akan menyebabkan call stack penuh. Hal ini dapat menjadi beban yang cukup signifikan pada prosesor. Pengulangan yang terus menerus pada sub-rutin yang cukup banyak dapat berakibat pada lambatnya proses pengulangan.

2. Lebih rumit untuk masalah yang sederhana

Untuk pemecahan masalah yang relatif sederhana, algoritma sekuensial terbukti lebih mudah dibuat daripada algoritma divide and conquer. Hal ini disebabkan karena algoritma sekuensial tidak perlu melalui ketiga langkah yang dilakukan oleh divide and conquer.

Algoritma divide dan conquer diperkenalkan sebagai sumber dari pengendalian proses paralel yang cukup rumit hal ini karena masalah-masalah yang terjadi dapat diatasi secara independen. Banyak arsitektur dan bahasa pemrograman yang mendesain implementasinya (apliaksi) dengan struktur dasar dari algoritma divide dan conquer.

Pemrograman betanggungjawab atas implementasi suatu solusi. Pembuatan program akan menjadi lebih sederhana jika masalah dapat dipecah menjadi sub-sub masalah yang dapat dikelola.

III. Pengeritan Komputasi Paralel

Komputasi paralel adalah salah satu teknik melakukan beerapa komputasi sekaligus secara bersamaan. Hal ini umum diperlukan saat melakukan perhitungan yang kompleks, misalnya harus mengolah data dalam jumlah besar seperti di industry keuangan dan bioinformatika atau karena tuntutan proses komputasi yang banyak. Kasus kedua seringkali ditemukan pada kalukasi numeric yang bertujuan menyelesaikan persamaan matematis di bidang fisika (fisika komputasi), kimia (kimia komputasi), dan lain-lain.

Untuk melakukan aneka jenis komputasi paralel ini diperlukan infrastruktur mesin paralel yang terdiri dari banyak komputer yang dihubungkan dengan jaringan dan mampu bekerja secara paralel untuk menyelesaikan satu masalah. Karena itu, diperlukan aneka perangkat lunak pendukung yang biasa disebut sebagai middleware. Middleware tersebut berperan dalam mengatur distribusi pekerjaan antar node dalam satu mesin paralel.

Selain diperlukan dukungan hardware, pemrogram atau pengguna komputer harus dapat memanfaatkan pemrograman paralel untuk merealisasikan komputasi. Meskipun semua hardware yang diperlukan sudah tersedia, komputasi paralel tidak akan berjalan sendirinya.

Di dalam komputasi paralel ada yang dinamakan dengan pemrograman paralel. Pemrograman paralel merupakan suatu teknik pemrograman komputer yang memungkinkan eksekusi perintah/operasi secara bersamaan (komputasi paralel). Bila komputer yang digunakan secara bersamaan tersebut dilakukan oleh komputer-komputer terpisah yang terhubung dalam suatu jaringan komputer, hal itu dikenal dengan suatu istilah yang disebut sistem terdistribusi (distributed computing).

Tujuan utama dari pemrograman paralel adalah untuk meningkatkan performa komputasi sehingga persoalan dapat diselesaikan dengan lebih cepat. Keberhasilan pemrograman paralel diukur dari berapa banyak peningkatan kecepatan yang diperoleh dengan menggunakan teknik paralel. Semakin banyak hal yang bisa dilakukan secara bersamaan (dalam waktu yang sama), semakin banyak pekerjaan yang bisa diselesaikan.

Peningkatan Kecepatan

Peningkatan kecepatan dapat diformulasikan dalam persamaan berikut ini

$$S = \frac{T_1}{T_j}$$

Dimana T_1 adalah waktu yang dibutuhkan untuk menyelesaikan pekerjaan (program komputer) apabila program tersebut hanya dijalankan dalam satu komputer. Dan T_j adalah waktu yang dibutuhkan jika pekerjaan dikerjakan bersamaan oleh beberapa komputer (dibagi ke beberapa komputer).

Paralel processing tidak dapat disamakan dengan multitasking, Pada multitasking, sebuah CPU mengeksekusi beberapa program sekaligus. Paralel processing disebut juga paralel computing. Pada sistem komputasi paralel, terdapat beberapa unit prosesor dan beberapa unit memori yang melakukan pekerjaan secara bersama-sama

Untuk lebih memperjelas pemahaman mengenai perbedaan komputasi tunggal yang menggunakan 1 processor dengan komputasi paralel (menggunakan

beberapa processor, berikut adalah penjelasan mengenai beberapa mengenai model komputasi komputer.

Ada 4 model komputer jika dikelompokkan berdasarkan jenis komputasi yang digunakan. Keempat jenis tersebut adalah:

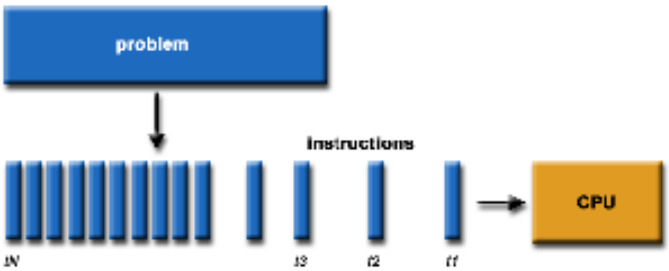
1. Single Instruction, Single Data (SISD)
   SISD merupakan model yang hanya menggunakan 1 processor saja. Dengan kata lain, model ini merupakan dapat dikatakan sebagai modeluntuk komputasi tunggal.
   Contoh: UNIVAC1, IBM 360, CDC 7600, Cray 1, dan PDP 1.

2. Single Instruction, Multiple Data (SIMD)
   Model ini menggunakan banyak processor dengan instruksi yang sama, namun data yang diolah di setiap prosesor berbeda.
   Contoh: ILLIAC IV, MasPar, Cray X-MP, dan Cray Y-MP.

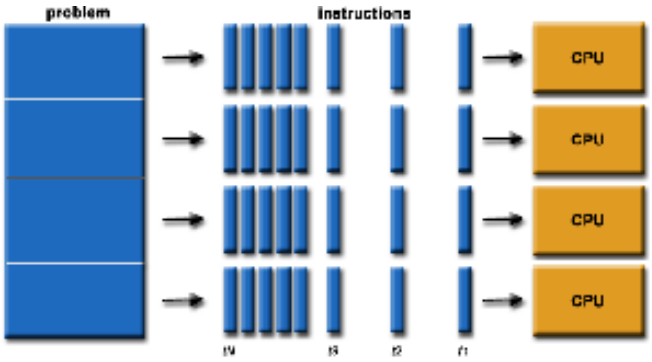
3. Multiple Instruction, Single Data (MISD)
   Model ini menggunakan banyak processor dengan setiap processor menggunakan instruksi yang berbeda namun mengolah data yang sama. Hingga saat ini, belum pernah terdapat komputer yang menerapkan model ini.

4. Multiple Instruction, Multiple Data (MIMD)
   Model ini menggunakan banyak processor dengan setiap processor memiliki instruksi yang berbeda dan mengolah data yang berbeda.
   Contoh: IBM POWER5, HP/Compaq AlphaServer, Intel IA32, dan AMD Opteron

Perbedaan antara komputasi tunggal dengan komputasi paralel, bisa digambarkan pada gambar di bawah ini:



Gambar 1. Penyelesaian Sebuah Masalah pada Komputasi Tunggal
Sumber:https://computing.llnl.gov/tutorials/parallel_comp/



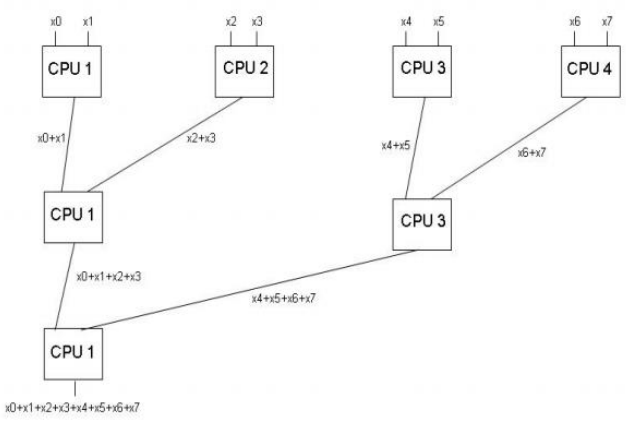
Gambar 2. Penyelesaian Sebuah Masalah pada Komputasi Paralel
Sumber:https://computing.llnl.gov/tutorials/parallel_comp/

Dari kedua gambar, kita dapat melihat bahwa komputasi paralel lebif efektif sehingga dapat menghemat waktu untuk pemorsesan data. Kesimpulan dari perbedaan kedua gambar di atas adalah bahwa kinerja komputasi paralel lebih efektif dan dapat menghemat waktu untuk pemrosesan data yang banyak daripada komputasi tunggal. Dari penjelasan-penjelasan di atas, kita bisa mendapatkan jawaban mengapa dan kapan kita perlu menggunakan komputasi paralel. Jawabannya adalah karena komputasi paralel jauh lebih menghemat waktu dan sangat efektif ketika kita harus mengolah data dalam jumlah yang besar. Namun keefektifan akan hilang ketika kita hanya mengolah data dalam jumlah yang kecil, karena data dengan jumlah kecil atau sedikit lebih efektif jika kita menggunakan komputasi tunggal.

Komputasi paralel diperlukan saat kapasitas yang diperlukan sangat besar, baik karena mengolah data dalam jumlah besar atau proses komputasi yang banyak. Sehingga semakin banyak hal yang bisa dilakukan dan diselesaikan dalam waktu yang sama, dengan demikian waktu yang dibutuhkan semakin singkat. Dengan adanya komputasi paralel dapat mempermudah menyelesaikan permasalahan yang rumit dengan waktu yang cepat dan efisien karena mampu melakukan beberapa proses komputasi secara bersamaan.

Adapun kekurangan dari komputasi paralel adalah dibutuhkan perangkat keras lainnya, kebutuhan daya yang lebih tinggi, dibutuhkan lebih banyak prosesor sehingga biaya lebih mahal. Melihat kelebihan dan kekurangan yang ada tentunya pengembangan komputasi paralel sanagt berperan penting dalam kemajuan dibidang komputer.

IV. Contoh Penerapan *Divide and conquer* Dalam Komputasi Paralel



Gambar 3 Penerapan Divide and conquer Dalam Komputas Paralel
http://www.cs.berkeley.edu/~demmel/cs170_spr07/LectureNotes/Lecture_Parallelism_DC.pdf

Penerapan *Divide and conquer* pada komputasi paralel dapat dilakukan dengan membagi suatu pekerjaan ke beberapa prosesor yang tersedia. Dengan cara ini, suatu persoalan dapat dipecah menjadi beberapa persoalan. Pembagian dilakukan hingga persoalan dianggap cukup kecil untuk diselesaikan. Dalam implementasinya, batasan seberapa kecil persoalan harus dipecah juga bergantung pada jenis persoalan yang diselesaikan dan banyaknya prosesor yang dapat digunakan.

Misalkan kita ingin menjumlahkan 2p angka pada suatu sistem yang terdiri dari p buah prosesor. Untuk memudahkan, kita bisa mengasumsikan jumlah p merupakan kelipatan hasil perpangkatan angka 2 ($p = 2^k$). Pada tahap pertama, setiap prosesor menambahkan sepasang angka yang berbeda, mengurangi 2p menjadi p. Kemudian, p/2 prosesor menambahkan pasangan angka yang berbeda, p menjadi p/2. Pada tahap berikutnya, sebanyak p/4 prosesor melakukan p/4 penjumlahan. Pada tahap ke-j, p/2j-1 prosesor melakukan p/2j-1 penjumlahan. Proses tersebut berulang sampai tahap ke-k, dimana $k=_{log2p}$. Langkah berikutnya adalah menggabugkan hasil penjumlahan yang dilakukan oleh masing-masing prosesor. Terdapat prosesor yang melakukan penjumlahan final. Dengan demikian, proses penjumlahan angka sebanyak 2p yang dikerjakan oleh p prosesor memerlukan $\log_2 p$ langkah.

Dengan cara yang serupa, kita dapat menghitung penjumlahan n buah angka pada p prosesor. Asumsi p|n dan p = 2k diberlakukan untuk mempermudah penjelasan. Proses komputasi dibagi ke masing-masing prosesor. Setiap prosesor secara independen dan bersamaan menjumlahkan n/p input yang berbeda. Sebanyak p/2 prosesor menggabungkan hasil penjumlahan yang dilakukan oleh prosesor yang berbeda dan secara keseluruhan memerlukan log2p + 1 komputasi. Dengan demikian dapat disimpulkan bahwa kompleksitas untuk menghitung n buah angka adalah O(n/p + log2 p).

Contoh berikut adalah penerapan *devide and conquer* untuk sebuah program yang menerapkan algoritma *devide and conquer* tanpa komputasi paralel.

```
function sum(a(1..n) )
//penjumlahan n buah bilangan yang dilakukan secara sekuensial
    if(n == 1)
            →a(1)
    x = sum(a(1…n/2))
    y =  sum(a(n/2+1…n))
    ➔  (x + y)
```

Kompleksitas dari program tersebut adalah
T(n) = 2T(n/2) + 1 = O(n)

Berikut adalah program yang sama dengan memanfaatkan komputasi paralel

```
function sum(a(1..n))
//penjumlahan n buah bilangan yang dilakukan secara paralel
    if(n == 1)
            →a(1)
    lakukan secara parelel {
            x = sum(a(1…n/2))
            y =  sum(a(n/2+1…n))
    }
(x + y)
```

Bagian program yang berada dalam blok ‘lakukan secara paralel’ akan dikerjakan pada prosesor yang berbeda jika masih terdapat prosesor yang tersedia. Jika tidak ada prosesor lain yang tersedia, proses akan dilakukan pada prosesor yang sama.

Dengan asumsi terdapat beberapa prosesor yang tersedia dengan jumlah sedikitnya n, kompleksitas menjadi $T(n) = T(n/2) + 1 = \log_2 n$. Namun, prosesor sebanyak n hampir tidak mungkin diterapkan karena memerlukan biaya yang cukup mahal jika n besar. Meskipun demikian, pemecahan komputasi secara paralel pada program yang menggunakan algoritma *devide and conquer* memiliki pengaruh dalam kecepatan penyelesaian persoalan meskipun jumlah p lebih kecil dari n.

Sebagai contoh, kita akan menjumlahkan 400 buah bilangan. Jika proses perhitungan tidak menggunakan komputasi paralel, maka diperlukan waktu sebanyak n-1, yaitu 399 kali. Jika menerpakan komputasi paralel dengan 4 prosesor, diperlukan waktu sebanyak sebanyak $n/p + \log_2 p$, yaitu $400/4 + \log_2 4 = 100 + 2 = 102$ kali. Perbedaan waktu yang diperlukan akan semakin besar jika prosesor yang digunakan semakin banyak dan banyaknya bilangan semakin besar. Penghematan waktu tersebut sangat berharga jika diterapkan dalam sistem komputer di suatu perusahaan yang memerlukan komputasi dalam

jumlah besar, terlebih jika proses komputasi yang besar perlu dilakukan secara berulang-ulang.

V. KESIMPULAN

Alogaritma *Divide and conquer* sudah lama diperkenalkan sebagai sumber dari pengendalian proses paralel, untuk mengatasi masalah-masalah yang terjadi secara independent

Pemrosesan paralel dipergunakan untuk memudahkan user dalam berinteraksi dari satu sistem ke sistem yang lain, dengan tujuan untuk membagi beban yang terdapat pada suatu sistem sehingga satu masalah dipecahkan secara bersama-sama.

Sistem pemrosesan parallel banyak digunakan oleh komputer-komputer saat ini, terutama untuk proses komputasi yang kompleks dan memerlukan waktu yang lama. Banyak perusahaan yang menggunakan komputasi paralel pada komputer-komputer mereka. Contoh industri yang sangat bergantung pada komputasi parelel adalah industri perfilman yang memerlukan komputer berkecepatan tinggi untuk melakukan rendering film.

Dengan pemrosesan paralel, kinerja komputer menjadi jauh lebih cepat dibandingkan tanpa membagi tugas. Setiap proses komputasi akan dilakukan oleh prosesor yang berbeda sehingga jauh lebih cepat daripada hanya menggunakan prosesor tunggal. Penggunan komputasi paralel tentu memerlukan biaya yang lebih besar. Namun, perbedaan waktu untuk menyelesaikan persoalan menjadi lebih cepat. Jika perhitungan sangat sederhana, misalkan menjumlahkan angka dari 1 sampai 10, kecepatan tidak akan terasa. Namun perbedaan waktu untuk menyelesaikan persoalan besar akan cukup terasa, terlebih jika proses komputasi dilakukan secara berulang-ulang atau terus-menerus.

Dapat disimpulkan bahwa komputasi paralel merupakan hal yangpenting untuk meningkatkan keefektifan dan keefisienan pemrosesan data yang banyak sehingga sangat menghemat waktu. Namun keefektifan akan hilang ketika kita hanya mengolah data dalam jumlah yang kecil, karena data dengan jumlah kecil atau sedikit lebih efektif jika kita menggunakan komputasi tunggal.

REFERENSI

[1].http://mahadisuta.blogspot.com/2012/12/implementasi- komputasi-paralel-dalam.html. Diakses pada 3 April 2015 pukul 17.20

[2].andikafisma.wordpress.com/alogaritma-divide-and-conquer/. Diakses pada 3 April 2015 pukul 17.37

[3].http://repository.usu.ac.id/bitstream/123456789/27329/3/Chapter%20II.pdf. Diakses pada 3 April 2015 pukul 18.21

[4].http://thesis.binus.ac.id/Asli/Bab2/2007-2-00251-IF-Bab%202.pdf. Diakses pada 3 April 2015 pukul 18.23

[5].http://kornelius-pg.blogspot.com/2012/04/kinerja-komputasi-dengan-parallel.html. Diakses pada 3 April 2015 pukul 19.11

[6].http://bertzzie.com/knowledge/analisis-algoritma/DivideAndConquer.html. Diakses pada 3 April 2015 pukul 19.19

[7].http://www.scribd.com/doc/33185692/Algoritma-Bucket-sort#scribd Diakses pada 3 April 2015 pukul 20.01

[8].http://blog.yogaprihastomo.com/wp-content/uploads/2008/01/divideconquer.pdf. Diakses pada 3 April 2015 pukul 20.03

[9].http://www.cs.berkeley.edu/~demmel/cs170_spr07/LectureNotes/Lecture_Parallelism_DC.pdf. . Diakses pada 4 April 2015 pukul 18.03

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 5 Mei 2015

Ivan Andrianto, 13513039