PARALEL ALGORITMA GENETIKA UNTUK MENYELESAIKAN TRAVELING SALESMAN PROBLEM MENGGUNAKAN MPI

PARALLEL GENETIC ALGHORITM TO SOLVE TRAVELING SALESMAN PROBLEM USING MPI

Rahadian Rizkina[1], Fhira Nhita[2], Fitriyani[3]

[1]Prodi S1 Ilmu Komputasi, Fakultas Informatika, Universitas Telkom
[2] Prodi S1 Ilmu Komputasi, Fakultas Informatika, Universitas Telkom
[3] Prodi S1 Ilmu Komputasi, Fakultas Informatika, Universitas Telkom
[1]**rrizkina@gmail.com**, [2]**farid.alchair@gmail.com**, [3]**fitriyani.y@gmail.com**

Abstrak

Traveling Salesman Problem baik dikenal sebagai masalah penting optimasi kombinatorial. Tujuannya adalah untuk menemukan jalur terpendek dari sebuah kota awal ke kota tujuan yang setiap kota hanya boleh dilewati tepat satu kali kemudian kembali ke kota awal. Metode untuk memecahkan TSP salah satunya adalah Genetic Algorithm (GA). GA adalah sebuah metode yang dapat menghasilkan solusi dan waktu yang optimal. Meskipun GA adalah metode yang optimal dan pendekatan yang baik untuk TSP namun saat jumlah kota pada TSP meningkat waktu yang dibutuhkan akan semakin besar. Hal ini disebabkan karena perhitungan nilai fitness setiap generasinya akan semakin banyak setiap jumlah kota bertambah. Oleh karena itu akan dibuat sebuah sistem untuk menangani masalah tersebut dengan memparalelkan metode GA. Sistem tersebut akan dijalankan pada Microsoft – MPI menggunakan Parallel Genetic Algorithm agar menghasilkan komplesitas waktu yang optimal. Hasil observasi menunjukan bahwa performa Paralel AG lebih baik daripada Serial AG. Parallel AG pada TSP menghasilkan nilai jarak terpendek sebesar 4697,18 dan nilai fitness 0,000213 untuk penggunaan 100 generasi, ukuran populasi sebanyak 100, probabilitas crossover 0,9 dan probabilitas mutasi 0,1 dengan waktu yang dibutuhkan sebesar 1,67 detik. Sedangkan Serial AG pada TSP menghasilkan nilai jarak terpendek sebesar 4801,91 dan nilai fitness 0,000208 untuk penggunaan 100 generasi, ukuran populasi sebanyak 100, probabilitas crossover 0,9 dan probabilitas mutasi 0,1 dengan perhitungan waktu 5,04 detik.

Kata kunci : Traveling Salesman Problem, Genetic Algorithm, Microsoft Cluster

Abstract

Traveling Salesman Problem well known as an important issue combinatorial optimization. The goal is to find the shortest path from an initial city to the destination city that each city can only be passed exactly once and then back into town early. Methods to solve TSP one of which is a Genetic Algorithm (GA). GA is a method that can produce a solution and the optimal time. Although GA is the optimal method and a good approach for TSP but when the number of cities in the TSP increased the time required will be greater. This is because the calculation of fitness value of each generation will be more and more every city number increases. Therefore, it will be created a system to deal with the issue parallelize GA method. The system will run on Microsoft - MPI using Parallel Genetic Algorithm to produce optimal time complexity. Observation results show that the performance is better than the AG Parallel Serial AG. Parallel AG on the TSP produce the shortest distance value at 4697.18 and the fitness value of 0.000213 for the use of 100 generations, the population size of 100, crossover probability mutation probability 0.9 and 0.1 with the required time of 1.67 seconds. While Serial AG on the TSP produce the shortest distance value at 4801.91 and the fitness value of 0.000208 for the use of 100 generations, the population size of 100, crossover probability and mutation probability, 0.9 and 0.1 with the timing 5.04 seconds.

Keywords: Traveling Salesman Problem, Genetic Algorithm, MPI Cluster.

1. Pendahuluan

Traveling Salesman Problem baik dikenal sebagai masalah penting optimasi kombinatorial. Tujuannya adalah untuk menemukan jalur terpendek dari sebuah kota awal ke kota tujuan yang setiap kota hanya boleh dilewati tepat satu kali kemudian kembali ke kota awal. Berbeda dengan definisi sederhana, sulit untuk memecahkan sebuah masalah TSP karena termasuk masalah NP-Complete problem [1].

Ada banyak metode untuk memecahkan TSP. Salah satunya adalah Genetic Algorithm (GA) atau algoritma genetika. AG adalah sebuah metode yang

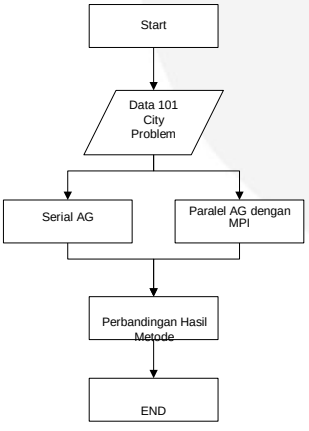
dapat menghasilkan solusi dan waktu yang optimal. Meskipun AG adalah metode yang optimal dan pendekatan yang baik untuk TSP, namun saat jumlah kota pada TSP meningkat, waktu yang dibutuhkan untuk menyelesaikan masalah akan cukup besar. Untuk menangani masalah waktu yang membesar, metode AG tidak dapat berdiri sendiri, sehingga salah satu cara untuk menangani masalah tersebut adalah memparalelkan metode AG supaya mempersingkat waktu yang dibutuhkan.

Algortima TSP dengan paralel AG perlu dijalankan dalam fasilitas komputasi kinerja tinggi untuk mencapai hasil yang sebanding dalam waktu yang singkat. Ada beberapa tools yang dapat digunakan untuk parallel genetic algorithm salah satunya adalah MPI. Untuk penggunaan MPI dalam AG sumber daya komputer sangat berpengaruh terhadap waktu total, peningkatan kecepatan, dan efisiensi. Microsoft MPI adalah salah satu open source yang menggunakan MPI dan open source yang mengizinkan terminal jaringan komputasi untuk melihat satu sama lain sebagai cluster yang bekerja bersama-sama untuk memecahkan masalah umum [2].

Tujuan dari tugas akhir ini adalah pengimplementasian optimasi TSP dengan menggunakan metode Parallel Genetic Algorithm menggunakan *Microsoft*-MPI dan membandingkannya dengan optimasi TSP serial Genetic Algorithm yang akan di implementasikan menggunakan Microsoft-MPI.

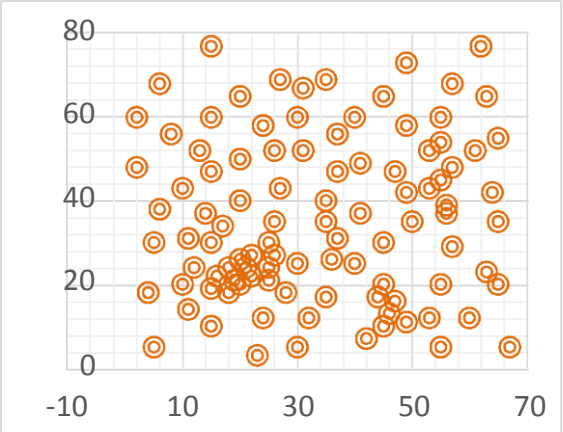
2. Metodologi

Secara spesifik, Pencarian solusi pada TSP kurva tertutup dengan menggunakan serial AG dan paralel AG terdiri dari beberapa tahapan, metode dan komponen. Berikut ini adalah *Flowchart* Perencanaan Sistem :



Gambar 2-1: Gambaran Umum Sistem

Data yang digunakan adalah data 101 kota yang didapatkan dari TSPlib. Berikut adalah visualisasi data 101 kota :



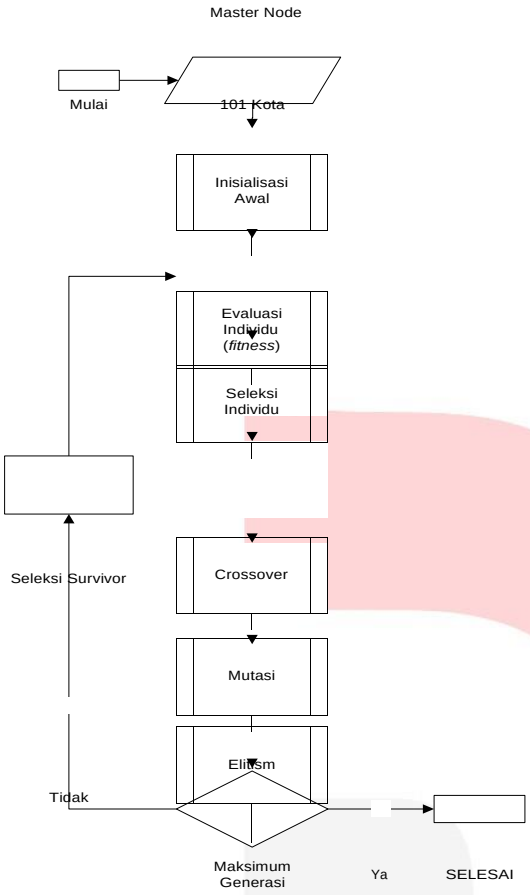
Skenario yang digunakan untuk pengujian sistem Serial AG dan Paralel AG sebagai berikut :

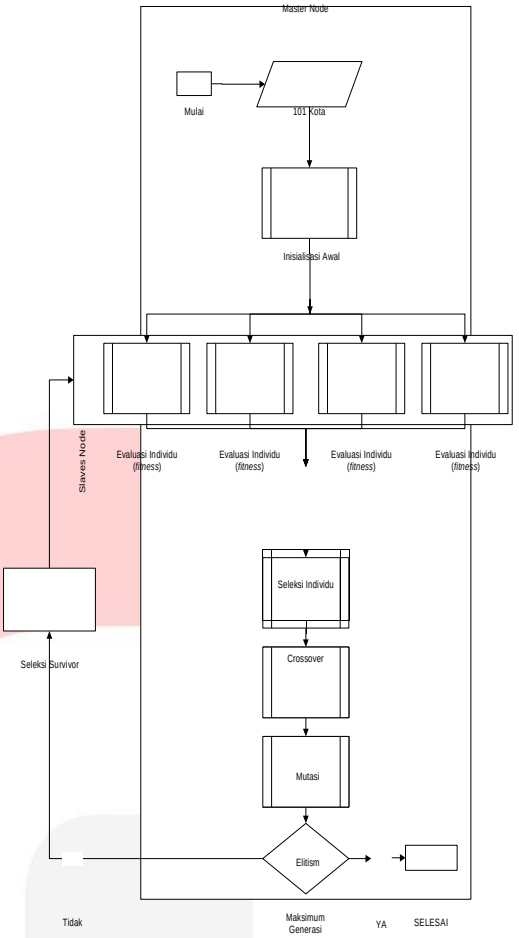
No	Ukuran Populasi	Probabilitas Crossover	Probabilitas Mutasi
1	100	0.9	0.5
2			0.3
3			0.1
4		0.7	0.5
5			0.3
6			0.1
7		0.5	0.5
8			0.3
9			0.1

2.1 Serial Algoritma Genetika

Pada sistem serial AG yang dibangun sama dengan sistem paralel AG, hanya berbeda pada tahap evaluasi individu. Dimana pada serial AG, evaluasi individu dilakukan pada master slave bukan di slave node.

Berikut ini adalah flowchart paralel AG untuk mencari solusi pada persoalan TSP.





2.2 Paralel Algoritma Genetika

Pada paralel master slave, tahap yang paling menghabiskan waktu, terutama saat ukuran populasi sangat besar adalah evaluasi individu, tahap tersebut akan dijalankan didalam paralel. Dalam masalah optimasi TSP dengan menggunakan paralel AG, semua tahap kecuali evaluasi individu akan dijalankan pada master node secara sekuensial. Master node akan mengirim individu-individu ke slave node yang akan menghitung nilai fitness dari individu yang dikirim dan hasilnya dikembalikan ke master node. Ketika master node mendapatkan nilai fitness dari semua individu didalam populasi, tahap elitisme, seleksi individu, crossover dan mutasi diterapkan secara global [7]. Berikut ini adalah flowchart paralel AG untuk mencari solusi pada persoalan TSP :

2.2.1 Inisialisasi Populasi

Tahapan pertama dalam paralel AG adalah inisialisasi populasi yakni melakukan penentuan nilai awal yang akan dijalankan pada master slave. Bagian penentuan nilai awal ini merupakan input yang dilakukan oleh pengguna sendiri. Input-input yang diperlukan dalam paralel AG pada tugas akhir ini meliputi:

1. Penentuan banyaknya node dalam setiap kromosom.
2. Penentuan besar populasi dalam satu generasi.
3. Penentuan banyak generasi yang akan dilakukan.
4. Penentuan besar crossover probability (peluang terjadinya kawin silang).
5. Penentuan besar mutation probability (peluang terjadinya mutasi).
6. Node awal dan tujuan.

2.2.2 Evaluasi Individu

Tahap kedua dari paralel AG adalah evaluasi individu, dimana proses ini akan menghitung nilai fitness dari setiap kromosom yang telah dibangkitkan secara random pada tahap inisialisasi populasi di atas.

Nilai fitness dari setiap kromosom dihitung berdasarkan panjang jalur linier yang dihasilkan dari jumlah jarak keseluruhan dari urutan node-node yang dilalui. Dalam masalah optimasi pada TSP,

individu (kromosom) yang bernilai fitness yang tinggi yang akan bertahan hidup atau yang akan terpilih dan kromosom yang bernilai rendah akan mati atau tidak terpilih pada tahap selanjutnya. Karena solusi yang dicari adalah meminimalkan sebuah fungsi h, maka nilai fitness yang dicari adalah kromosom yang memiliki panjang jalur yang pendek.

Oleh karena itu, rumus untuk mencari nilai fitness pada masalah minimasi ini adalah:

$$f = \frac{1}{h}$$

keterangan :
f = fungsi fitness
h = fungsi yang akan diminimasi (Total Jarak)

Untuk mendapatkan total jarak, hitung jarak antar node menggunakan rumus jarak kartesian, kemudian jumlahkan semua hasil jarak node yang telah dihitung menggunakan persamaan jarak kartesian. Oleh karena itu, rumus untuk mencari jarak kartesian adalah:

$$|A - B| = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

keterangan :
A, B = node A dan node B
X,Y = koordinat node (absis,ordinat)

Tahap evaluasi individu ini dilakukan (looping) sebanyak besar populasi dalam satu generasi. Sehingga didapat nilai fitness dari semua kromosom dalam satu populasi. Nilai fitness suatu kromosom ini kemudian akan dibandingkan dengan fitness kromosom yang lainnya yang ada pada semua generasi. Dimana nilai fitness paling tinggi yang akan terpilih.

2.2.3 Elitsme

Input pada prosedur ini adalah Populasi, indeks kromosom terbaik dan ukuran populasi akan disimpan di dalam master node. Kromosom terbaik yang telah disimpan di dalam master node akan tetap dipilih sebagai salah satu kandidat induk yang akan dipindah silangkan. Nilai fitness terbaik ini akan dibandingkan dengan nilai fitness kromosom-kromosom generasi berikutnya hasil pindah silang dan mutasi. Kromosom terbaik ini akan disertakan lagi pada generasi berikutnya.

2.2.4 Seleksi individu

Input pada prosedur ini adalah Populasi, indeks kromosom terbaik dan ukuran populasi akan disimpan di dalam master node. Kromosom terbaik yang telah disimpan di dalam master node akan tetap dipilih sebagai salah satu kandidat induk yang akan dipindah silangkan. Nilai fitness terbaik ini akan dibandingkan dengan nilai fitness kromosom-kromosom generasi berikutnya hasil pindah silang dan mutasi. Kromosom terbaik ini akan disertakan lagi pada generasi berikutnya.

2.2.5 Crossover

Prosedur crossover adalah prosedur untuk mengkawinkan dua induk yang telah dipilih pada proses roulette wheel, namun tidak semua induk akan mengalami crossover karena proses crossover ini banyak dikendalikan oleh beberapa bilangan random.

Crossover pada TSP dapat di implementasikan dengan skema order crossover. Pada skema ini, satu bagian kromosom di pertukarkan dengan tetap menjaga urutan kota yang bukan bagian dari kromosom tersebut. Pada skema order crossover digunakan teknik dua titik potong (two-point crossover), dimana titik potong ini menentukan gen mana saja yang akan dipertukarkan antarinduk. Titik potong diperoleh secara acak, gen-gen yang terletak diantara dua titik potong akan saling dipertukarkan antarinduk.

2.2.6 Mutasi

Pada kasus TSP ini skema mutasi yang digunakan adalah skema swap mutation. Dengan skema swap mutation ini mutasi dilakukan dengan cara menukarkan gen-gen yang dipilih secara acak dengan gen yang dipilih secara acak juga. Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh parameter probabilitas mutasi. Diperkirakan total gen yang mengalami mutasi pada seluruh generasi adalah probabilitas mutasi dikalikan dengan ukuran populasi dan banyaknya maksimum generasi.

2.2.7 Seleksi Survivor / Pergantian Individu

Untuk pergantian populasi dalam suatu generasi digunakan general replacement yaitu pergantian populasi secara keseluruhan. Populasi pada generasi sebelumnya yang merupakan parent diganti seluruhnya dengan populasi baru yang merupakan anak atau turunannya. Populasi pada generasi berikutnya adalah kromosom bentuk baru dari hasil pindah silang dan mutasi serta ditambah kromosom hasil elitisme.

Prosedur yang sama akan berlaku untuk populasi baru, yakni akan mengalami tahapan yang sama dengan populasi sebelumnya. Apabila perhitungan dilanjutkan sampai ke maksimum generasi maka akan didapatkan nilai fitness tertinggi dari seluruh genersi yang menunjukkan kromosom terbaik yang akan diambil sebagai solusi.

2.2.8 Perbandingan Hasil AG

Untuk pergantian populasi dalam suatu generasi digunakan general replacement yaitu pergantian populasi secara keseluruhan. Populasi pada generasi sebelumnya yang merupakan parent diganti seluruhnya dengan populasi baru yang merupakan anak atau turunannya. Populasi pada

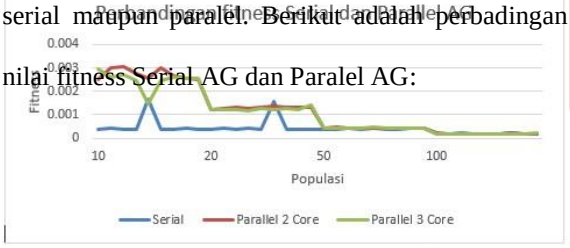
generasi berikutnya adalah kromosom bentuk baru dari hasil pindah silang dan mutasi serta ditambah kromosom hasil elitisme.

Prosedur yang sama akan berlaku untuk populasi baru, yakni akan mengalami tahapan yang sama dengan populasi sebelumnya. Apabila perhitungan dilanjutkan sampai ke maksimum

generasi maka akan didapatkan nilai fitness tertinggi dari seluruh generasi yang menunjukkan kromosom terbaik yang akan diambil sebagai solusi.

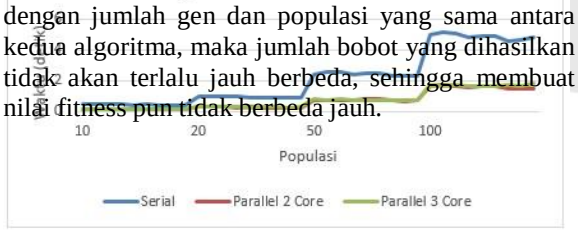
3. Hasil dan Analisis

Berdasarkan skenario yang telah dijelaskan sebelumnya, telah dilakukan observasi untuk pengujian kombinasi parameter berikut diatas secara serial maupun paralel. Berikut adalah perbadingan nilai fitness Serial AG dan Paralel AG:



Gambar 3-1 Perbandingan Fitness Serial AG dan Paralel AG

Berdasarkan grafik diatas, dapat dilihat bahwa perubahan fitness antara Serial AG maupun parallel tidaklah banyak memilili perbedaan, hal ini menandakan bahwa baik Serial AG maupun paralel sama-sama berpacu pada probabilitas bilangan random, sehingga tidak dapat diprediksi apakah fitness yang didapat Serial AG akan lebih baik atau tidak daripada AG parallel. Hal lain yang mempengaruhi nilai fitness adalah jumlah gen, dengan jumlah gen dan populasi yang sama antara kedua algoritma, maka jumlah bobot yang dihasilkan tidak akan terlalu jauh berbeda, sehingga membuat nilai fitness pun tidak berbeda jauh.



Gambar 3-2 Perbandingan Waktu Serial AG Dan Paralel AG

Analisis yang di dapat dari gambar adalah, bahwa terjadi peningkatan yang signifikan untuk selisih waktu eksekusi antara serial dan Paralel AG 2 core. Hal ini membuktikan dengan parameter yang sama hasil perhitugan yang dilakukan dengan proses core waktu yang dibutuhkan sama dengan proses serial, hal ini disebabkan kurangnya jumlah populasi yang akan dihitung, dan waktu yang dibutuhkan proses Send dan Recv terpakai percuma. Evaluasi perfomansi, bisa didapatkan Speedup, dan Performance Improvement. Berikut adalah hasil dari Speedup:

$$S = \frac{0.4}{0.11} = 3{,}64$$

Keterangan:

s = hasil perhitungan serial (waktu)

p = hasil perhitungan parallel 2 dan 3 core (waktu)

Berikut adalah hasil dari Perfomance Improvement:

$$= \frac{0.4 - 0.11}{0.4} * (100\%) = 72.5\%$$

Dari perhitungan Evaluasi Perfomansi dapat dikatakan menyelesaikan AG dengan menggunakan proses paralel mampu meningkatkan perfomansi lebih dari 50%.

paralel menggunakan 2 core jauh lebih cepat dari proses perhitungan serial. Namun pada Paralel AG 3

4. Kesimpulan

Analisis dan hasil implementasi pada bab sebelumnya memberikan beberapa hal yang dapat disimpulkan dari penerapan Parallel Algoritma Genetika menggunakan MPI pada TSP. Berikut adalah kesimpulan yang dapat diambil:

a. Berdasarkan hasil observasi didapatkan, Paralel AG lebih cepat dibandingkan Serial AG untuk semua Probabilitas Crossover selain itu nilai fitness yang di dapatkan Paralel AG jauh lebih baik dari Serial AG.
b. Perbedaan implementasi Serial AG dan Paralel AG terletak pada proses perhitungan nilai fitness, perhitungan fitness pada Serial AG hanya dilakukan pada master node menggunakan 1 core sedangkan Paralel AG dilakukan pada slave node menggunakan 2 core atau lebih.
c. Pada perhitungan Evaluasi Perfomansi menghasilkan peningkatan sebesar 3.14 (Speedup) hal ini dapat dijadikan acuan bahwa menggunakan Paralel AG memang efektif.

5. Saran

a. Penggunaan populasi dan generasi yang lebih besar, dan core yang lebih banyak memberikan nilai fitness yang lebih baik dan memberikan perbandingan yang lebih signifikan antara Paralel dan Serial AG. Oleh karena itu penelitian selanjutnya diharapkan menggunakan populasi dan generasi yang lebih besar, dan core yang lebih banyak.
b. Penggunaan AG pada penelitian ini dianggap cukup baik, namun pada perkembangannya banyak metode yang mengkombinasikan 2 atau lebih metode, sehingga akan menarik apabila di implementasikan di cluster-MPI.

Daftar Pustaka:

[1] E. L. Lawler, J. K. Lenstra, A. H. G. RinnocoyKan, and D. B. Shymoys, The Traveling Salesman Problem, John Wiley & Sons, Chichester, (1985).

[2] Izzatin Abdul Aziz, Low Tan Jung, and Mazlina Mehat, "Parallelization Of Traveling Salseman Problem Using Rocks Cluster," dalam *Procedings of the International MultiConference of Engineers and Computer Scientists*, Hongkong, 2008.

[3] C. Elison, ""101 City Problem," TSP," [Online]. Available: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/eil101.tsp. [Diakses 15 Oktober 2014].

[4] G. Federico, "Travelling Salesman Problem," In-teh, Australia, 2008.

[5] Entin, Diktat Kuliah Kecerdasan Buatan, Institut Teknologi Sepuluh Nopember.

[6] Suyanto, Soft Computing: Membangun Mesin Ber-IQ Tinggi, Bandung: Informatika, 2008.

[7] H. R. Er dan Prof. Dr. Nadia Erdogan, "Parallel Genetic Algorithm to Solve Traveling Salesman Problem on MapReduce Framework using Hadoop Cluster," *JSCSE*, 2013.

[8] D. Chappell, Introducing Windows HPC Server - Running Parallel Application on Cluster, San Fransisco: Chappell & Associates, 2011.

[9] R. K. M. S. a. f. Jeremy Fischer, Methods For Creating XSEDE Compatible Cluster, New York, 2008.

[10] University of California, "Rocks Cluster Distribution," dalam *User Guide*, California, University of California, 2006.

[11] P. S. Pacheco, A User Guide to MPI, California: University of San Fransisco, 1998.

[12] S. J. K. Edward, CUDA by Example, US : NVIDIA, 2011.