IMPLEMENTASI ALGORITMA ANT COLONY DENGAN MULTI AGENT SYSTEM PADA KASUS PENCARIAN JALUR TERPENDEK

Winda Pitralisa[1], Tjokorda Agung Budi Wirayuda[2], Zk. Abdurahman Baizal[3]

[1]Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Graf sudah banyak diaplikasikan dalam kehidupan sehari-hari, terutama dalam bidang pemrograman komputer. Salah satu contoh aplikasi graf yang dibahas pada makalah ini adalah pencarian jalan atau lintasan terpendek dari titik awal menuju titik tujuan. Penelitian pada makalah ini menggunakan algoritma semut yaitu Ant Colony System (ACS) dengan Multi Agent System (MAS). Sistem yang dibangun memanfaatkan semut sebagai agen yang ditugaskan untuk mencari solusi dari permasalahan optimasi jarak terpendek. ACS dengan MAS merupakan pengembangan dari Ant System (AS) dimana terdapat perbedaan pada aturan transisi status dan pembaharuan feromon.

Penelitian ini dilakukan dengan pembuatan sistem yang menerapkan MAS di dalam ACS. Kemudian dilakukan pengujian pada parameter yang dapat mempengaruhi hasil dari jarak terpendek yaitu parameter yang digunakan dalam proses perhitungan (nilai pengatur bobot intensitas feromon (α) dan nilai pengatur bobot visibilitas (β)) dan parameter untuk menentukan aturan transisi status dimana ACS melibatkan proses ekplorasi dan ekploitasi untuk menghasilkan solusi dari proses perhitungan.

Hasil akhir yang diperoleh dari penelitian ini adalah ACS dengan MAS sebaiknya menggunakan nilai pengatur bobot intensitas feromon (α) yang bernilai lebih kecil daripada nilai pengatur bobot visibilitas (β) dengan tingkat perbedaan performansi hampir mencapai 50%, dan untuk memenuhi syarat q≤q0 pada aturan transisi probabilitas pemilihan simpul menggunakan q0=0.75 yang menunjukkan peluang proses eksploitasi lebih banyak agar hasil jarak terpendek mendekati minimum. Pada pengujian pembagian/pengaturan jumlah ekplorasi dan eksploitasi berdasarkan busur/jalur yang telah dilewati menunjukkan akan lebih baik jika ada keseimbangan antara kedua proses tersebut. Selain itu, memori dan waktu terlihat berubah secara linier atau lebih dipengaruhi oleh banyaknya simpul.

Kata Kunci : Graf, ACS, MAS, AS, Eksplorasi, Eksploitasi.

Abstract

Graf has been widely applied in everyday life, especially in the field of computer programming. One example application graph discussed in this paper is to seek the path or the shortest path from the starting point towards the destination point. Research on this paper using the ant algorithm is Ant Colony System (ACS) with the Multi Agent System (MAS). The system built use ants as an agent assigned to look for solutions of optimization problems the shortest distance. ACS with MAS is an extension of Ant System (AS) where there is a difference in the status transition rule and pheromone updating.

This research was done by making systems that apply MAS in the ACS. Then, examined using parameters that can affect the result of the shortest distance in terms of parameters used in the calculation (regulator of pheromone intensity (α) and visibility of the regulator (β)) and parameters to determine the transition state in which ACS involves the process of exploration and exploitation to generatesolutions.

The final result obtained from this study were ACS with MAS should use the value weighted intensity regulator of pheromone (α) value is smaller than the value of weight control visibility (β) approximately 50% performance level, and to qualify q ≤ q0 on the probability of choosing node transition rules using q0 = 0.75 which shows more opportunity exploitation process for the shortest distance close to the minimum. In the testing division / setting the number of exploration and exploitation based on arc / track has been skipped show would be better if there is a balance between both processes. In addition, memory and time appear to change linearly or more influenced by the number of nodes.

Keywords : Graf, ACS, MAS, AS, Exploration, Exploitation.

1. Pendahuluan

1.1 Latar Belakang

Dalam beberapa persoalan terutama dalam bidang pemrograman komputer, graf mempunyai peranan penting dalam membantu pemodelan untuk menyelesaikan berbagai persoalan optimasi seperti sistem penjadwalan, perjalanan, transportasi dan aliran jaringan. Sejak ditemukannya teori graf dalam ilmu matematika, ada banyak ditemukan solusi mengenai masalah rute penyebaran informasi. Dengan mengandaikan alur penyebaran informasi sebagai graf, masalah ini dapat dibuat menjadi lebih sederhana. Ditambah lagi dengan pengaplikasian teori graf ke dalam bidang pemrograman komputer, dapat diciptakan algoritma-algoritma yang dapat memecahkan masalah tersebut dalam waktu cepat serta cara untuk melakukan optimisasi yaitu salah satunya dengan menggunakan Algoritma Semut.

Secara umum, pencarian jalur terpendek dapat dibagi menjadi dua metode, yaitu metode konvensional dan metode heuristik. Metode konvensional cenderung lebih mudah dipahami daripada metode heuristik, tetapi jika dibandingkan, hasil yang diperoleh dari metode heuristik lebih variatif dan waktu perhitungan yang diperlukan lebih singkat . Metode heuristik terdiri dari beberapa macam algortima yang bisaa digunakan. Salah satunya adalah algoritma semut (*Ant Colony*, Antco*)*.

Algoritma semut atau *Ant System* diinspirasi oleh tingkah laku koloni semut, bagaimana hewan yang hampir buta dengan kemampuan individu yang sederhana dapat menemukan jalan terpendek (sarang semut dengan makanan) jika bersama dalam suatu koloni. Semut juga mampu untuk beradaptasi dengan perubahan yang terjadi di dalam lingkungan mereka, sebagai contoh menemukan jalur terpendek yang baru ketika yang lama sudah tidak memungkinkan lagi karena munculnya rintangan. *Ant System* (AS) yang tidak optimum jika kajian kasusnya lebih luas selanjutnya dikembangkan atau dilakukan perbaikan dengan munculnya *Ant Colony System* (ACS) . Peningkatan performa AS mengarah pada penerapan ACS. ACS juga menerapkan *multi-agent system* (MAS) yang merupakan paradigma pengembangan sistem di mana dalam suatu komunitas terdapat beberapa *agent* yang saling berinteraksi, bernegosiasi, dan berkoordinasi satu sama lain dalam menjalankan pekerjaan. Konsep agen tersebut dianalogikan dengan konsep algoritma semut sehingga bisa membantu performansi optimal mendapatkan jalur terpendek dari suatu graf dalam waktu seminimal mungkin. Algoritma ini dapat dijadikan metodologi baru dalam mencari solusi optimum atau membuktikan keoptimalannya dibandingkan metodelogi lain dalam permasalahan graf. Oleh karena itu, diterapkan cara lain dengan menggunakan algoritma ini dalam pencarian solusi graf yang lebih memperlihatkan peran pada penerapan MAS didalamnya.

1.2 Perumusan Masalah

Perumusan masalah dari Tugas Akhir ini berdasarkan latar belakang masalah yang dipaparkan yaitu mengenai cara menerapkan *multi-agent system* dalam Algoritma Semut/ACS serta membuat implementasi yang dapat menyelesaikan permasalahan graf dengan Algoritma Semut/ACS.

1.3 Batasan Masalah

1. Program hanya akan mencari lintasan terpendek dalam graf dari satu titik ke titik lainnya.
2. Graf yang digunakan graf lengkap berbobot yang tertutup (graf tertutup) dan *bidirectional* (setiap jalur bisa ditempuh bolak-balik dua arah).
3. Graf direpresentasikan dalam bentuk matriks.
4. Input yang digunakan untuk graf yaitu jumlah titik dan titik awal serta titik tujuannya.
5. Fungsi – fungsi yang dibutuhkan adalah
   a. Fungsi untuk membuat matriks/graf
   b. Fungsi probabilitas untuk menentukan titik selanjutnya
   c. Fungsi untuk menyeleksi jarak terpendek.
6. Perangkat lunak dibangun menggunakan bahasa pemrograman Java
7. Kasus ini akan diselesaikan dengan menggunakan ACS.

1.4 Tujuan

Tujuan pengerjaan Tugas Akhir ini berdasarkan rumusan masalah di atas adalah:

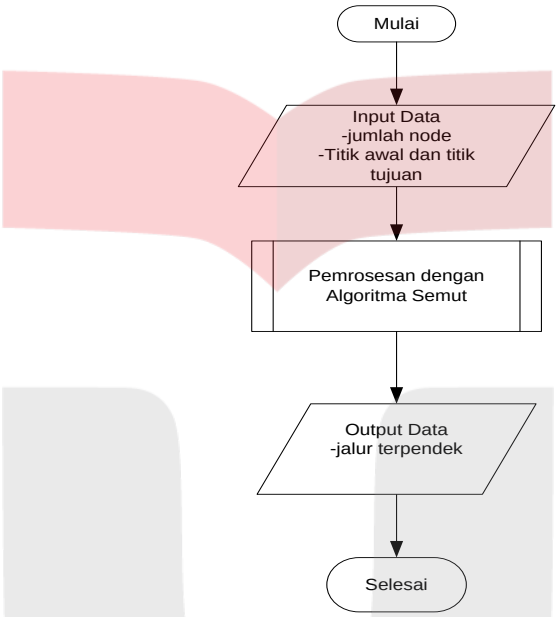
1. Pembuatan perangkat lunak untuk permasalahan graf menggunakan Algoritma Semut dengan penerapan Multi-Agent System.
2. Mengaplikasikan metode Algoritma Semut ini ke dalam program sederhana untuk mencari lintasan terpendek dalam graf dari suatu masukan (menganalisa dan menghitung lintasan dengan jarak paling minimal).
3. Menganalisa dan mengimplementasikan ACS, sebagai perbaikan dari *ant system* pada graf.
4. Melihat peran dari MAS yang diterapkan pada ACS dan menganalisa performansinya.

1.5 Metodologi Penyelesaian Masalah

Metode penelitian yang diterapkan pada pengerjaan Tugas Akhir ini adalah :

1. Melakukan studi literatur tentang konsep *multi-agent system*, algoritma semut, dan graf.
2. Mengidentifikasi masalah dan melakukan perancangan dan pemodelan pada sistem yang akan diuji.

3. Melakukan pengujian perangkat lunak yaitu dengan melihat performansi dan akurasi dari algoritma semut dengan penerapan *multi-agent system* .
4. Pengambilan kesimpulan dan penyusunan laporan tugas akhir.



Gambar 1-1 Deskripsi Sistem

1.6 Sistematika Penulisan

- **BAB I PENDAHULUAN**
  Bab ini berisi pembahasan masalah umum yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.
- **BAB II LANDASAN TEORI**
  Bagian ini memuat landasan teori yang berfungsi sebagai sumber atau alat dalam memahami permasalahan yang berkaitan dengan teori graph, teori jalur terpendek dan teori mengenai algoritma Ant Colony.
- **BAB III PERANCANGAN SISTEM**
  Bab ini mngenai perancangan sistem sesuai dengan spesifikasi yang telah dijabarkan pada latar belakang masalah dan kebutuhan yang telah dianalisa sebelumnya.
- **BAB IV PENGUJIAN DAN ANALISIS**

Bab ini berisikan penjelasan mengenai hasil implementasi dan analisis kinerja dari perangkat lunak. Pada bagian ini mengulas analisis hasil pengujian terhadap sistem yang dibandingkan dengan kebenaran dan kesesuaiannya dengan kebutuhan perangkat lunak yang telah dituliskan pada bagian sebelumnya.

- **BAB V PENUTUP**
  Memuat kesimpulan-kesimpulan yang merupakan rangkuman dari hasil dan pembahasan perangkat lunak pada bagian sebelumnya dan saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama pembuatan perangkat lunak.

4. Pengujian Sistem dan Analisa Hasil

4.1 Pengujian Sistem

Pada bab ini menjelaskan bagaimana pengujian dan analisis terhadap sistem yang merupakan implementasi algoritma semut yaitu ACS dan bertujuan mencari jalur/jarak terpendek pada suatu kasus graf *multi agent system* (MAS). Pembahasan pada bab ini meliputi tujuan pengujian, strategi/skenario pengujian, hasil pengujian dan analisis hasil pengujian

4.1.1 Tujuan Pengujian

Berdasarkan latar belakang dan tujuan sistem pada bab 1, maka tujuan pengujian yaitu :

1. Menganalisis faktor-faktor yang mempengaruhi hasil dari jarak/jalur terpendek dan kinerja sistem dalam proses ACS dengan MAS.
2. Membandingkan waktu pemrosesan dan memori yang digunakan pada parameter pengujian yang berbeda.

4.1.2 Skenario Pengujian

Strategi pengujian didasarkan pada tujuan pengujian 4.1.1 tersebut, antara lain:

1. Pengujian untuk mengetahui yang dapat mempengaruhi rumus/proses perhitungan beserta hasil jarak/jalur terpendek menggunakan ACS dengan MAS. Pengujian dilakukan pada intensitas feromon dan visibilitas (invers dari jarak antar simpul), yaitu parameter alfa (α) dan beta (β). Alfa digunakan untuk mengontrol Tij (intensitas feromon) dan Beta untuk mengontrol visibilitas. Kedua parameter tersebut diuji pada graf dengan jumlah node 20, 30, dan 50. Nila alfa dan beta diatur sebagai berikut :
   - α=β, dengan α=1 dan β=1.
   - α>β, dengan α=2 dan β=1.
   - α<β, dengan α=1 dan β=2.

   pada pengujian ini, Tij = 0.01, ρ=0.5, q0=0.8, jumlah siklus=50, dan diujikan pada graf dengan titik awal 1 menuju ke semua node lainnya. Jumlah semut disesuaikan dengan jumlah node.
2. Hasil dari pengujian 1 digunakan pada pengujian aturan transisi status yaitu q≤q0 untuk melihat pengaruh dari nilai q0 yang menjadi penentu transisi ekplorasi/eksploitasi. Pengujian juga dilakukan pada graf dengan jumlah node 20, 30, 50 node. Nilai q0 diatur sebagai berikut :
   - q0=0.25
   - q0=0.5
   - q0=0.75

pada pengujian ini, Tij = 0.01, ρ=0.5, jumlah siklus=50 dan diujikan pada graf dengan titik awal 1 menuju ke semua node lainnya. Jumlah semut disesuaikan dengan jumlah node.

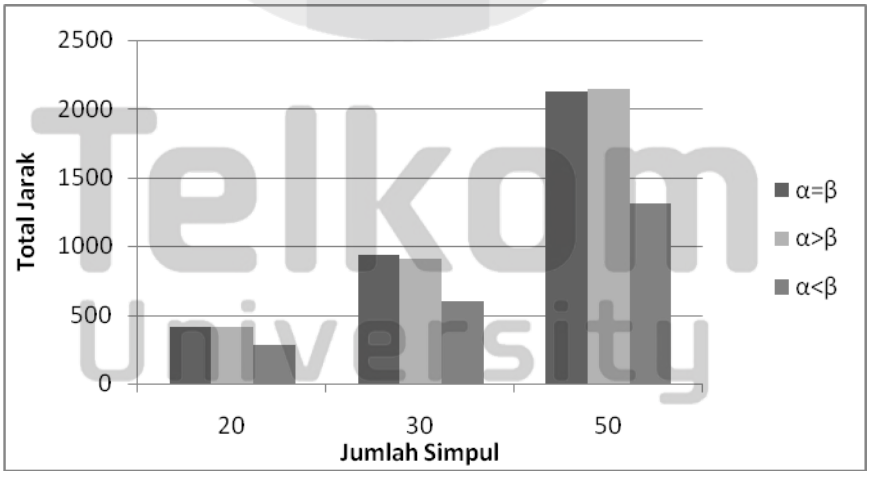
3. Pengujian untuk mengetahui pengaruh proses eksploitasi, dimana eksploitasi diatur persentasinya pada keseluruhan proses yaitu 50%, 80%, dan 90%. Parameter yang digunakan Tij=0.01, α=1, β=2, ρ=0.5, dan jumlah semut disesuaikan dengan jumlah node.
   Pengujian dilakukan dengan titik awal 1 ke semua titik terhubung lainnya pada masing-masing graf. Misalnya pada graf dengan 20 simpul, maka pengujian dilakukan dengan titik awal simpul ke-1 dan titik tujuan simpul ke-2 sampai simpul ke 20. Begitu pula pada graf 30 simpul dan 50 simpul.
4. Jarak yang dihasilkan dari masing-masing simpul awal 1 ke semua titik terhubung lainnya kemudian dijumlahkan agar digunakan untuk perbandingan. Waktu dan memori yang dipakai untuk perbandingan yaitu nilai rata-ratanya.
5. Hasil terbaik dari pengujian 1, pengujian 2, dan pengujian 3 dibandingkan dengan hasil dari algoritma Djikstra untuk melihat selisih nilai terbaik yang dihasilkan dari masing-masing pengujian.

4.2. Analisa Hasil Pengujian

Analisa hasil pengujian diterangkan sebagai berikut :

4.2.1. Analisis pengaruh α dan β tehadap jarak

Pengujian dilakukan sesuai dengan skenario pengujian. Hasil jarak yang diperoleh dari masing-masing pengujian (pengujian jika α=β, α<β, α>β) simpul ke-1 menuju semua simpul yang terhubung dicatat dan dijumlahkan semuanya. Lalu total jarak dari masing pengujiaan dibandingkan.



Gambar 4-1 Diagram pengaruh nilai α dan β terhadap jarak

Pada Gambar 4-1, total jarak yang dihasilkan memiliki nilai lebih kecil jika menggunakan parameter α>β. Hal itu menunjukkan bahwa nilai visibilitas lebih berpengaruh terhadap proses perhitungan daripada nilai

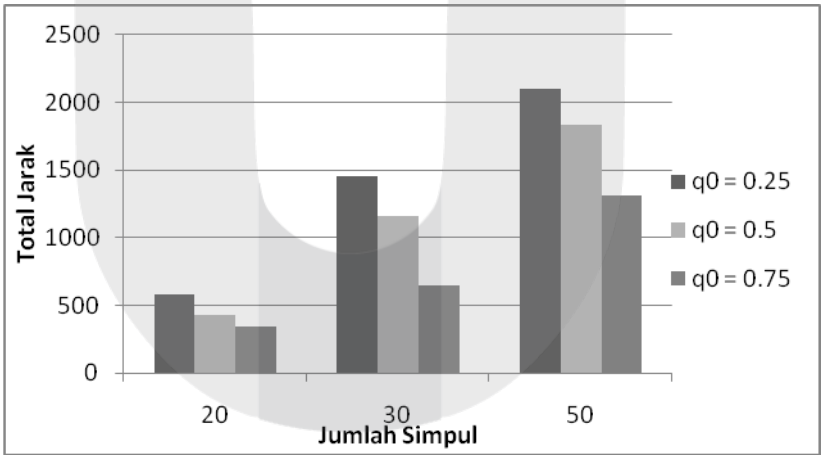
intensitas feromonnya (Tij), dimana β bertujuan untuk mengontrol nilai visibilitas tersebut. Visibilitas merupakan invers dari jarak. Tujuan dari adanya visibiltas juga terbukti yakni mempengaruhi sensitivitas dari pembagian feromon antara simpul. Nilai Tij yang kecil dan α yang hanya digunakan pada proses ekplorasi juga menjadi penyebab nilai visibiltas lebih berperan dalam perhitungan untuk menghasilkan jarak terpendek.

5.2.2. Analisis pengaruh q0

Pengujian pada 4.2.1 menunjukkan bahwa sebaiknya digunakan $\alpha<\beta$. Hasil tersebut selnajutnya dipakai untuk pengujian q0.

a. Pengaruh q0 yang diset tehadap jarak

Pengujian dilakukan pada masing=masing graf yang akan diuji (pengujian jika q0=0.25, q0=0.5, q0=0.75) dengan titik awal simpul ke-1 menuju semua simpul yang terhubung kemudian mencatat jarak yang dihasilkan dan dijumlahkan semuanya. Total jarak dari masing pengujiaan dibandingkan.



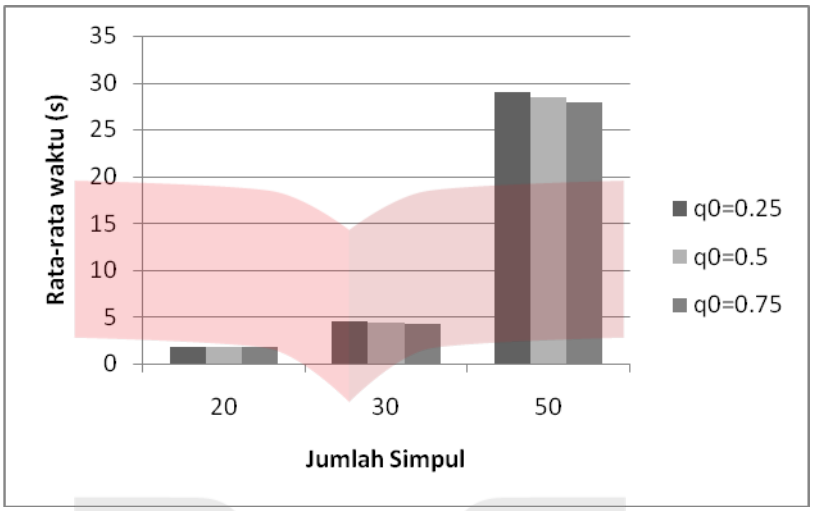
Gambar 4-2 Diagram pengaruh nilai q0 terhadap jarak

Gambar 4-2 memperlihatkan dengan jelas bahwa nilai q0 terbaik yaitu 0.75 yang dapat diartikan akan lebih baik jika proses eksploitasi mendominasi pemrosesan untuk menghasilkan jarak terpendek. Semakin kecil nilai q0 maka jumlah ekplorasi akan lebih besar daripada jumlah eksploitasi berdasarkan kondisi $q \leq q0$. Dominasi ekploitasi lebih baik karena pada ekplorasi pemilihan simpul tujuan dengan membangkitkan suatu bilangan random sehingga membuat hasilnya menjadi kurang akurat, sedangkan pada ekploitasi pemilihan simpul didasarkan pada nilai simpul yang lebih besar untuk dipilih.

b. Pengaruh q0 yang diset terhadap waktu pemrosesan

Pengujian dilakukan pada masing=masing graf yang akan diuji (pengujian jika q0=0.25, q0=0.5, q0=0.75) dengan titik awal simpul

ke-1 menuju semua simpul yang terhubung kemudian mencatat besarnya waktu dibutuhkan hingga mencapai solusidan dihitung nilai rata-ratanya agar dibandingkan dengan hasil dari q0 yang lain.

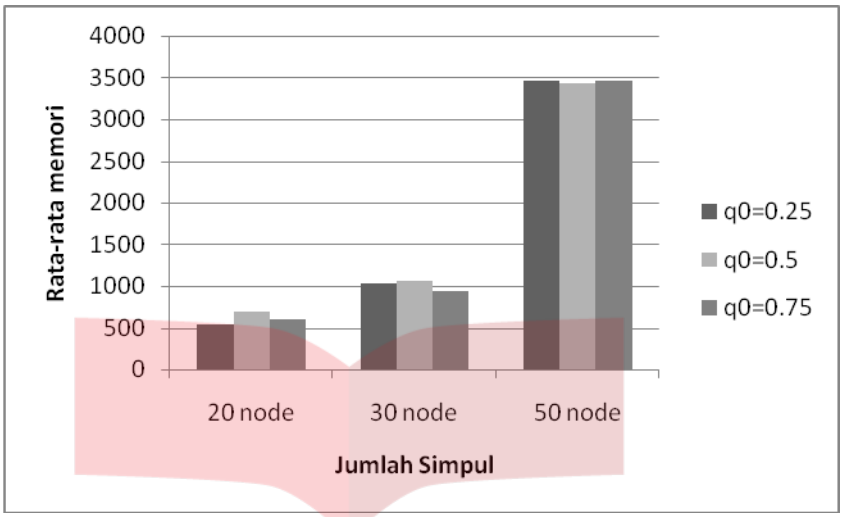


Gambar 4-3 Diagram pengaruh nilai q0 terhadap waktu pemrosesan

Nilai q0 tidak berpengaruh banyak terhadap waktu pemrosesan. Besarnya waktu pemrosesan lebih dipengaruhi oleh jumlah simpul. Namun sedikit terlihat pada Gambar 4-3 bahwa dengan q0=0.75 maka waktu pemrosesannya berkurang dibandingkan dengan nilai q0 yang lebih kecil.

c. Pengaruh q0 yang diset terhadap memori yang terpakai

Pengujian dilakukan pada masing=masing graf yang akan diuji (pengujian jika q0=0.25, q0=0.5, q0=0.75) dengan titik awal simpul ke-1 menuju semua simpul yang terhubung kemudian mencatat besarnya memori yang terpakai hingga mencapai solusi. Besarnya memori dari masing-masing titik yang diujikan kemudian dihitng nilai rata-ratanya agar dibandingkan dengan hasil dari q0 yang lain.



Gambar 4-4 Diagram pengaruh nilai q0 terhadap memori yang terpakai

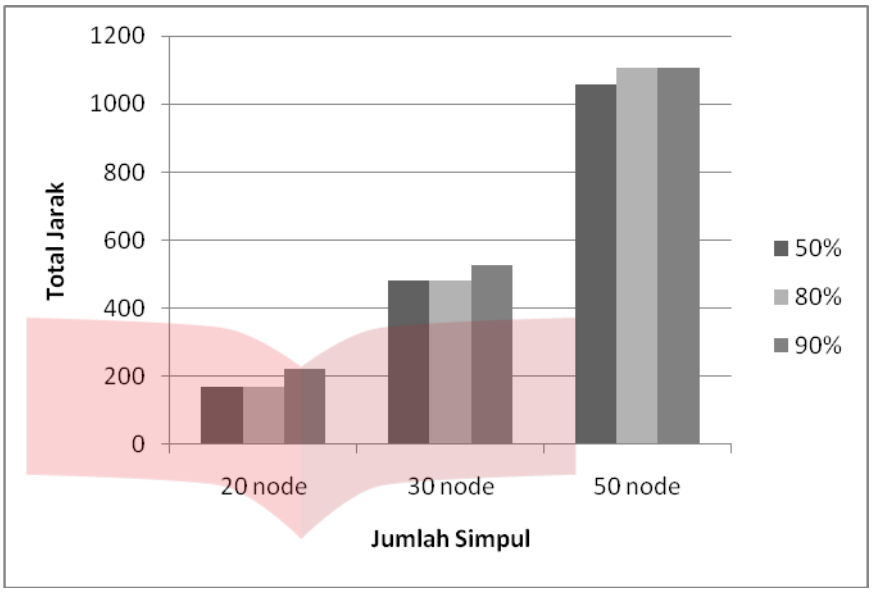
Seperti halnya pengaruh q0 terhadap waktu pemrosesan, Gambar 4-4 menunjukkan q0 juga tidak berpengaruh besar terhadap perubahan memori yang terpakai. Hal ini bisa memperkuat pengaruh semakin besarnya jumlah simpul yang mempengaruhi jumlah memori yang terpakai saat proses perhiutungan.

4.2.3. Analisis pengaturan persentase eksploitasi

Jika pada pengujian sebelumnya proses ekplorasi dan eksploitasi dipengaruhin kondisi q≤q0, maka pada percobaan selanjutnya mengatur secara otomatis pembagian jumlah antar proses eksplorasi dan eksploitasi tanpa harus bergantung dengan jumlah siklus.

a. Pengaruh persentasi ekploitasi terhadap jarak

Pengujian dilakukan sesuai dengan skenario pengujian. Hasil jarak yang diperoleh dari masing-masing simpul ke-1 menuju semua simpul yang terhubung dicatat dan dijumlahkan semuanya. Lalu total jarak dari masing pengujiaan dibandingkan.

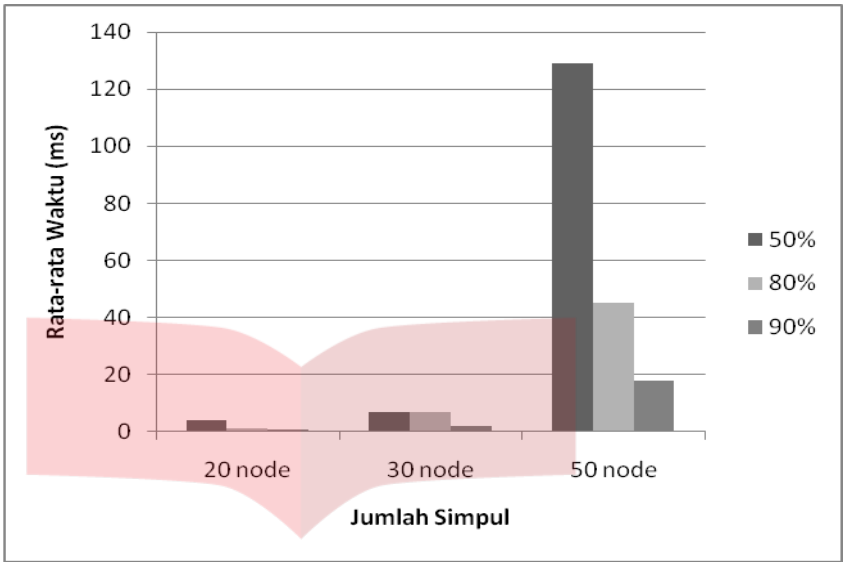


Gambar 4-5 Diagram pengaruh persentase ekploitasi terhadap jarak

Pada Gambar 4-5 menunjukkan jika jumlah ekploitasi diatur sama dengan jumlah eksploitasi maka solusi jarak terpendeknya bernilai lebih kecil. Dengan eksploitasi terlalu banyak mendominasi proses dan ekploitasi yang terlalu kecil justru membuat hasilnya tidak lebih baik dan cenderung buruk. Adanya keseimbangan antara proses eksplorasi dan eksploitasi menunjukkan peran masing-masing sama pentingnya, dimana ekplorasi berperan untuk mengeksplor simpul-simpul yang belum dikunjungi dan eksploitasi memanfaatkan hasil dari ekplorasi agar langsung dipakai untuk memperoleh solusi yang terbaik pada saat itu. Jika proses ekplorasi terlalu sedikit, kemungkinan ada simpul yang belum dijelajahi padahal bisa saja simpul tersebut merupakan salah satu solusi yang diinginkan.

b. Pengaruh persentasi ekploitasi terhadap waktu pemrosesan

Setelah perbandingan total jarak yang dihasilkan, waktu pemrosesan pada. dari masing-masing pengujian simpul ke-1 menuju semua simpul yang terhubung juga dicatat dan dihitung rata-rata waktu yang dibutuhkan. Hasil rata-rata waktu dari pengujian tersebut dibandingkan dan berlaku pada semua graf yang digunakan.



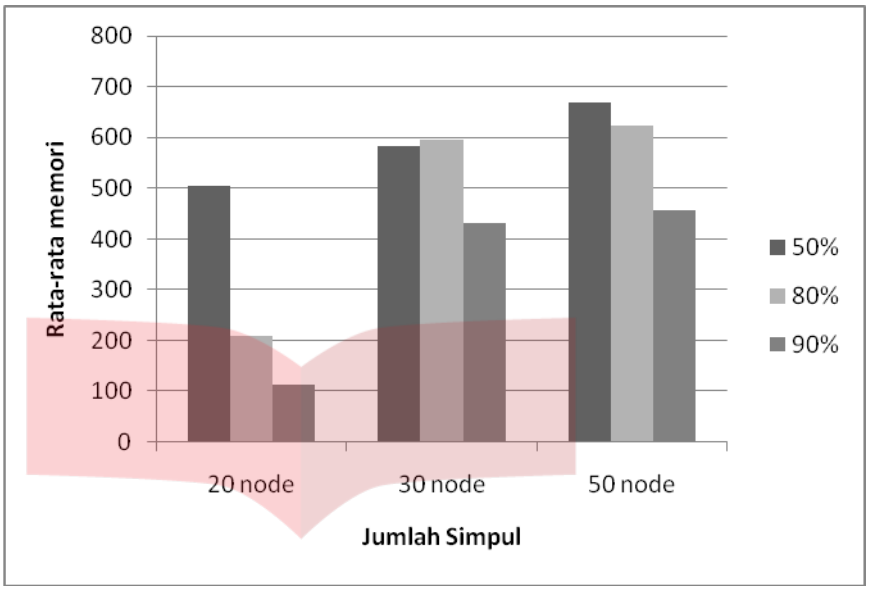
Gambar 4-6 Diagram pengaruh persentase eksploitasi terhadap waktu pemrosesan

Proses ekplorasi berpengaruh terhadap banyaknya waktu yang digunakan cukup terlihat pada Gambar 4-8. Semakin banyak ekploitasi dan sedikitnya ekplorasi maka waktu yang terpakai semakin kecil. Eksplorasi memakan waktu lebih banyak karena proses yang dijalankannya juga lebih bertahap daripada ekploitasi. Jika pada ekploitasi cukup dengan mengambil nilai hasil perkalian yang besar, berbeda dengan eksplorasi yang perlu menghitung probabilitas tiap simpul yang bisa dikunjungi kemudian menghitung probabilitas kumulatifnya dan membangkitkan bilangan random untuk pemilihan simpul. Pada graf dengan 50 simpul terlihat dengan jelas perbedaan waktu yang digunakan dimana jika ekploitasi mencapai 90% maka lebih cepat mengahsilkan solusi.

.

c. Pengaruh persentase ekploitasi terhadap memori yang terpakai

Selain waktu pemrosesan, memori yang terpakai saat pengujian simpul ke-1 menuju semua simpul yang terhubung pada masing graf juga dicatat dan dihitung rata-rata memori yang telah digunakn. Hasil rata-rata memori dari pengujian tersebut dibandingkan dan berlaku pada semua graf yang digunakan.

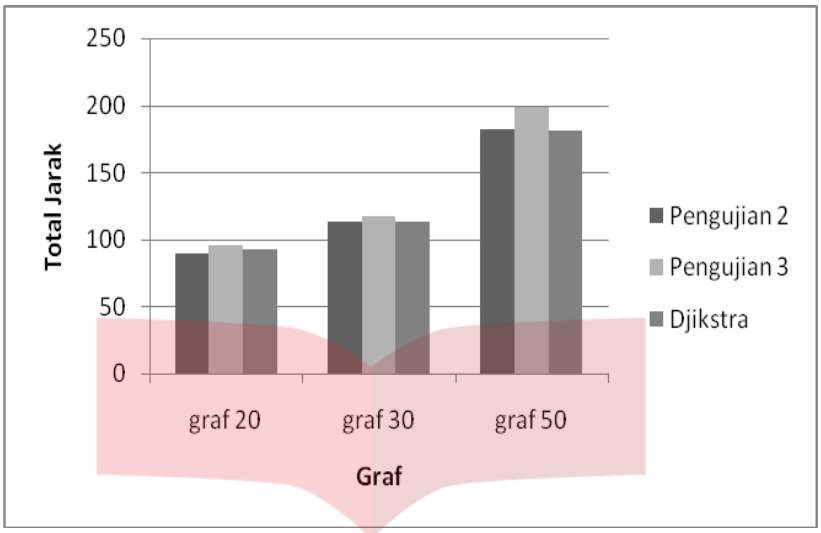


Gambar 4-7 Diagram pengaruh persentase eksploitasi terhadap memori yang terpakai

Sama halnya dengan pengaruh persentase terhadap waktu pemrosesan, begitu pula pada memori. Memori yang terpakai akan lebih sedikit jika ekploitasi mendominasi proses dan eksplorasinya hanya sebagian kecil. Hal ini terkait dengan penggunaan array pada proses. Pada ekplorasi array yang digunakan lebih banyak daripada saat eksploitasi, sesuai dengan tahapan ekplorasi yang memang lebih banyak sampai menemukan solusi yang diinginkan.

4.2.4. Analisa perbandingan ACS terhadap Djikstra

Perbandingan antara ACS dengan algoritma Djikstra untuk melihat keoptimalan dari solusi yang dihasilkan dan membuktikan bahwa hasil perhitungannya benar. Pada pengujian ini, data graf yang digunakan sama. Kemudian hasil perhitungan dari Djikstra dibandingkan dengan hasil terbaik yang diperoleh dari pengujian sebelumnya yaitu q0=0.75 (Pengujian 2), persentase eksploitasi 50% (Pengujian 3) dengan masing-masing pengujian menggunakan parameter α=1 dan β=2.



Gambar 4-8 Diagram Perbandingan ACS Dengan Algoritma Djikstra

Pada Gambar 4-8, setelah dibandingkan dengan Algoritma Djikstra maka dapat dibuktikan bahwa ACS cukup optimal dalam pencarian jalur terpendek. Perbandingan jarak total yang diperoleh dari ACS pada pengujian q0 (Pengujian 2) dan pengujian persentase eksploitasi (Pengujian 3) mendekati nilai yang dihasilkan dari Algoritma Djikstra. Pengujian 2 ternyata lebih baik daripada Pengujian 3, hanya saja kelemahan dari Pengujian 2 yaitu dari segi waktu dan jumlah siklus yang tidak tetap agar dapat langsung memperoleh hasil terbaik.

Daftar Pustaka

[1] Astuti , Yuni D. *Logika dan Algoritma*. Tersedia : http://yuni_dwi.staff.gunadarma.ac.id/Downloads/files/12671/Bab+1+-+Dasar+Teori+Graf.pdf [30 November 2009]

[2] Budianto. 2005. *An Overview and Survey on Multi Agent System. Intelligent Systems and Control Systems* (Jurnal Seminar Nasional : *Soft Computing, Intelligent Systems and Information Technology*" (SIIT 2005)) Indonesia Informatics Institute, Surabaya.

[3] Colorni, Alberto. Dorigo, M. Maniezzo, V. 1991. *Distributed Optimization by Ant Colonies*. Proceedings of ECAL91 – European Conference on Artificial Life, Paris, France, F. Varela and P. Bourgine (Eds.), Elsevier Publishing, hal. 134–142.

[4] Dorigo, M. Gambardella, L.M. 1997. *Ant Colonies For The Traveling Salesman Problem*. Journal of Université Libre de Bruxelles, Belgia.

[5] Mutakhiroh, Iing. 2007. *Pencarian Jalur Terpendek Dengan Algoritma Semut*.Jurnal Seminar Nasional Aplikasi Teknologi Informasi 2007 (SNATI). Laboratorium Pemrograman dan Informatika Teori UII, Yogyakarta.

[6] Refianti, Rina. Mutiara, A,B. *Solusi Optimal Travelling Salesman Problem Dengan Ant Colony System (ACS).* Makalah Jurusan Teknik Informatika, Universitas Gunadarma [Online]. Tersedia : http://paper.abmutiara.info/Ant_Colony_System/paper_J_GND_2005_3doc.pdf [13 Oktober 2009]

[7] Suyanto. 2010. *Algoritma Optimasi*. Graha Ilmu, Yogyakarta, hal. 206-221.

[8] Wahono, Romi S. 2001. *Multi Agent System: Beberapa Isu, Pendekatan dan Tantangan*. Proceedings IECI Japan Series Vol. 3, No.2, 2001, pp. 22-37. Department of Information and Mathematical Sciences Saitama University, Jepang.

[9] Wardy, Ibnu S. *Penggunaan Graf Dalam Agoritma Semut Untuk Melakukan Optimasi*. Makalah Program Studi Teknik Informatika ITB, Bandung. Tersedia : http://elesys.fsaintek.unair.ac.id/admin/makalah/Makalah0607-93.pdf [01 Desember 2009]