

IMPLEMENTASI PERBANDINGAN ALGORITMA ANT COLONY SYSTEM DENGAN ALGORITMA SUBSET DYNAMIC PROGRAMMING PADA KASUS TRAVELLING SALESMAN PROBLEM

Tommi Poltak Mario

Program Studi Teknik Informatika, STTI RESPATI YOGYAKARTA,

E-mail: tommypm_hutapea@yahoo.com.au

ABSTRAKSI

Travelling Salesman Problem atau TSP merupakan permasalahan dalam menentukan dan mengatur bagaimana suatu tour atau titik – titik persinggahan yang terdekat dengan titik tujuan dan tercepat waktu tujuan, ketika seorang salesman akan melakukan perjalanan dari titik asal ke titik tujuan yang mempunyai beberapa pilihan jalan yang dapat dilewati untuk sampai ke titik tujuan dan kembali ke titik asalnya lagi.

Permasalahannya adalah bagaimana mengatur suatu tour atau rute supaya diperoleh panjang perjalanan secara keseluruhannya menjadi minimum/terpendek agar diperoleh waktu yang singkat

Pada penelitian ini membicarakan perbandingan algoritma Ant Colony System dengan Subset Dynamic programming untuk menyelesaikan kasus TSP dengan membandingkan kedua algoritma tersebut. Oleh karena itu dibuat implementasi program untuk mencari algoritma yang terbaik dari kedua algoritma tersebut dengan mencari penyelesaian hasil optimal waktu tercepat dan jarak terpendek

Kata kunci: TSP, Ant Colony, Subset Dynamic.

1. PENGENALAN

Graf linier atau lebih singkatnya disebut dengan graf didefinisikan sebagai graf $G(V,E)$ yang terdiri dari suatu himpunan objek $V=(V_1,V_2,...V_n)$ yang disebut verteks dan himpunan lain $E=(e_1,e_2,...e_n)$ yang elemen-elemennya disebut edge, sedemikian sehingga tiap-tiap edge e_k diidentifikasi dengan pasangan verteks tak urut (unordered)[1].

Karena sifatnya yang sederhana, maka teori graf mempunyai penerapan yang luas pada teknik, ilmu fisika, matematika, kimia, dan mungkin masih banyak bidang-bidang yang lain.

Salah satu topik yang menarik dari teori graf adalah konsep tentang Travelling Salesman Problem yang terdiri dari dua jenis, yaitu Symetric TSP dan Asymetric TSP. Konsep ini memecahkan suatu masalah perjalanan yang berangkat dari suatu titik awal dan kembali lagi ke titik awal tersebut dalam suatu rute tertentu dengan memiliki rute terpendek. Sebagai contoh dalam suatu peta perjalanan seseorang mengunjungi sejumlah n kota yang digambarkan oleh verteks $v_1, v_2, v_3, ...v_n$ melewati jaringan jalan dan akan kembali ke kota v_1 . Masalahnya adalah mencari jarak Cij yang terdekat untuk menempuh perjalanan langsung v_i dan v_j dan kembali ke v_i .

Untuk lebih cepat dan akuratnya dalam mencari jarak Cij maka perancangan Travelling Salesman Problem ini akan dibuat suatu program untuk menvisulisasiakannya. Dalam menuliskan program, pertama kali harus ditentukan langkah apa yang harus diambil terlebih dahulu. Lalu prosedur urutan langkah-langkahnya ditata, dan baru dituliskan. Prosedur tata urutan langkah tersebut disebut algoritma. *Travelling Salesman Problem* (TSP) merupakan salah satu permasalahan yang memerlukan algoritma. Dari arti katanya TSP dapat

diterjemahkan sebagai suatu perjalanan yang dilakukan oleh seorang sales dengan syarat setiap kota harus dikunjungi sekali dan akan kembali ke kota tempat ia memulai perjalanan tersebut, dalam hal ini yang menjadi masalah adalah bagaimana sales tersebut mendapatkan rute perjalanan yang paling cepat. Algoritma Ant colony dan Subset Dynamic Programming adalah algoritma yang dapat diterapkan untuk menyelesaikan kasus TSP. Dalam hal ini dibangun program untuk mensimulasikan dan membandingkan kecepatan antara Algoritma Ant Colony dan Algoritma Subset Dynamic Programming pada perutean seorang sales/penjual dari kota yang satu ke kota yang lainnya.

Penelitian membahas algoritma *Ant Colony System* dan *Subset Dynamic Programming* yang akan diterapkan untuk menyelesaikan masalah TSP. Cara yang digunakan untuk menerapkannya adalah dengan membandingkan dan mengimplementasikan kedua algoritma tersebut ke dalam sebuah program.

Program yang dibuat mempunyai batasan masalah yaitu:

- Program dapat menyelesaikan TSP dengan menggunakan algoritma Ant Colony System dan subset dynamic programming.
- Program hanya menyelesaikan masalah perjalanan terpendek dengan asumsi tidak dipengaruhi kendala lain, misalnya jalan rusak dan lain sebagainya
- Inputan berupa kota secara otomatis oleh sistem secara acak
- Jumlah kota maksimal 50 kota
- Jalur yang dibuat berupa garis lurus
- Sistem mencari semua jalur yang ada (graf lengkap)
- Output hasil berupa data perjalanan

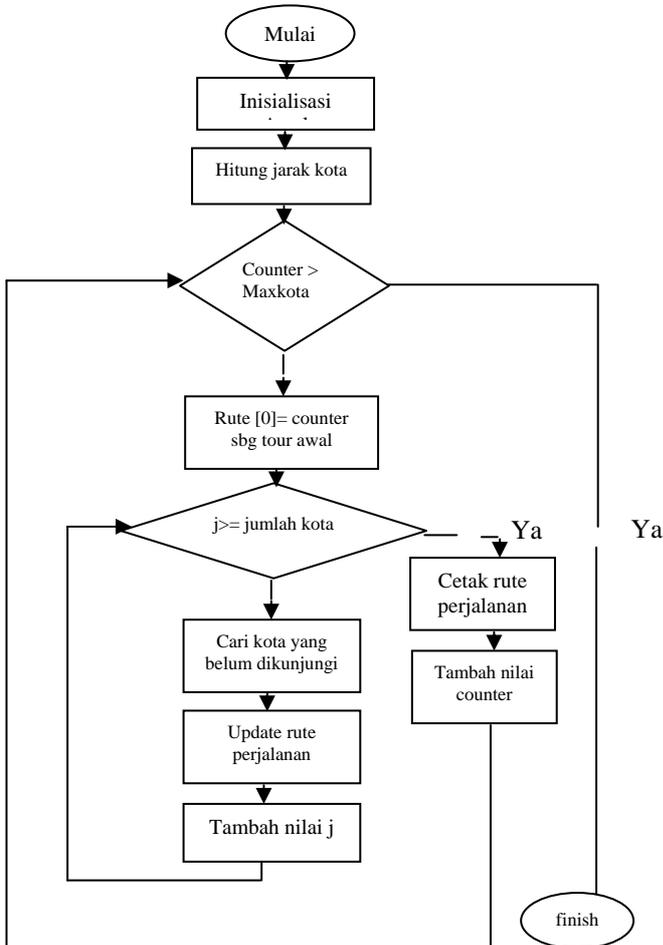
2. SISTEM KERJA

Proses algoritma pada beberapa metode mempunyai beberapa tahap penggambaran *Travelling Salesman Problem*. Pada bab ini penulis akan mencoba secara rinci proses simulasi TSP dengan metode *Ant Colony* dan *subset dynamic programming*.

Simulasi TSP metode *Ant Colony* dan *Subset Dynamic Programming* adalah suatu operasi Windows 32 bit. Aplikasi yang telah dikembangkan ini dibangun dengan perluasan algoritma dengan konsep untuk menerapkan metode *Ant Colony* dan *Subset Dynamic Programming*, dengan menyajikan suatu populasi objek dalam lingkungan grafis dimana suatu obyek dan obyek lainnya ditempatkan bersama-sama di dalam eksperimen populasi oleh aplikasi tersebut.

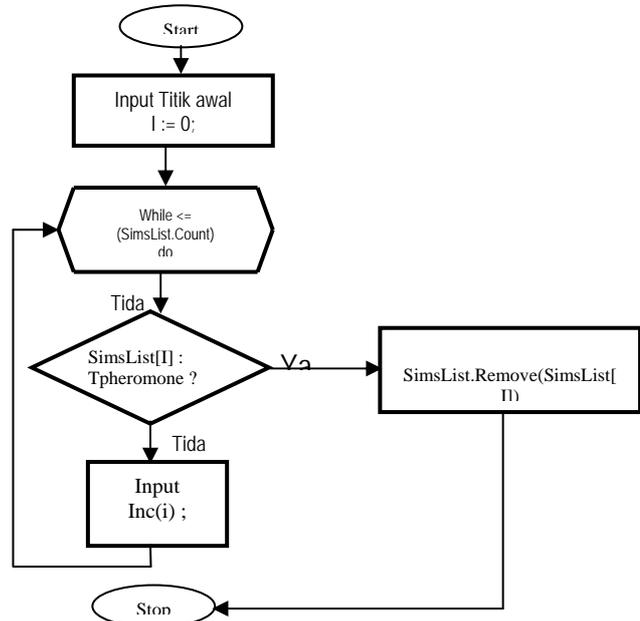
Berikut adalah model flow chart serta penjelasan tentang model flowchart *subset dynamic programming*.

Untuk penjelasan gambar dibawah flowchart subset dynamic programming yaitu Inisialisasi awal pada sistem yang dibuat memberikan nilai awal pada variabel tertentu yang digunakan untuk menjalankan algoritma subset dynamic programming yaitu counter = 0; jarak terpendek = 999999; jarak terakhir = 0; MaxArr subset = 0; Redim Arrsubset (1,1,0); rute [0]= counter menandakan bahwa kota awal dimulai dari array ke-0.



Gambar 1. Flowchart subset dynamic programming

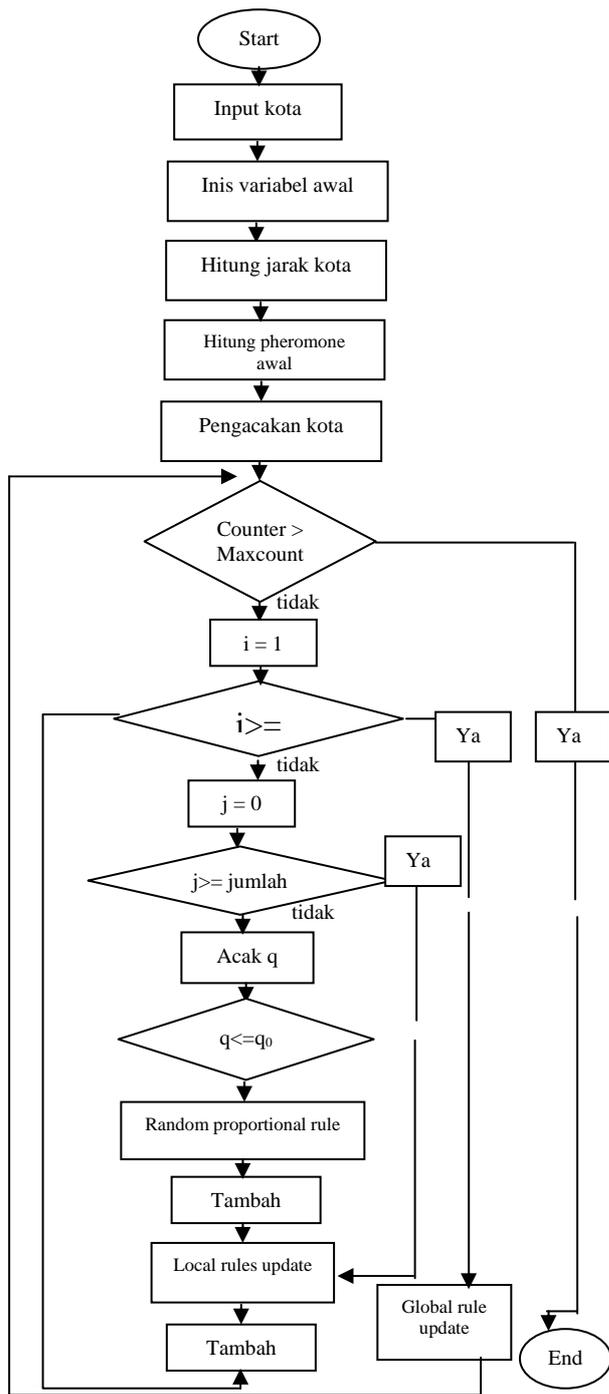
Secara spesifik kita dapat melihat alur lintasan travelling subset dynamic programming



Gambar 2. Alur lintasan travelling subset dynamic programming

Untuk penjelasan gambar flowchart diatas yaitu inisialisasi awal pada sistem yang dibuat adalah memberikan ρ nilai parameter – parameter yang telah ditentukan yang digunakan untuk menjalankan algoritma ant colony system, yaitu α (alfa) = ρ = 0.1, β (betha) = 2, q_0 = 0,9.

Objek melakukan semua perjalanan dan kembali ke kota awal dilakukan proses global updating, tetapi hanya jalur terpendek yang mengalami proses ini sehingga secara otomatis diikuti oleh objek-objek lain. Pada saat user melakukan perpindahan maka pheromone setiap jalur menggunakan local updating . Proses ini dilakukan sehingga setiap jalur yang dilewati objek mengalami pengurangan pheromone.

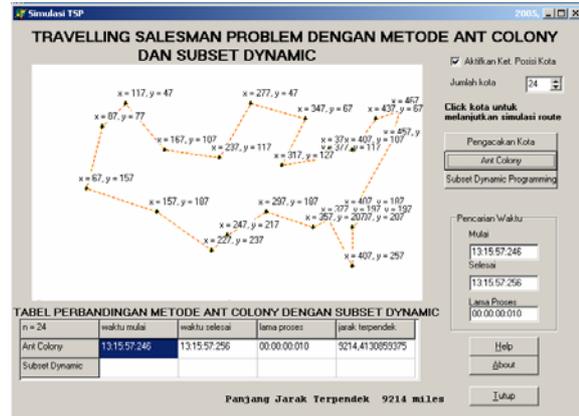


Gam1

3. IMPLEMENTASI PROGRAM

Pada saat menekan tombol Ant Colony, program membuat rute perjalanan yang dapat dimulai dari kota manapun juga, karena akan melalui rute yang sama. Pada tampilan di atas kita dapat melihat tampilan *Ant Colony* yang menampilkan hasil waktu yang diperoleh pada saat running dan menampilkan jarak terpendek yang ditempuh pada saat kita menggunakan algoritma *Ant Colony*. Hasil perjalanan tersebut adalah hasil perjalanan terbaik yang ditempuh dalam melakukan

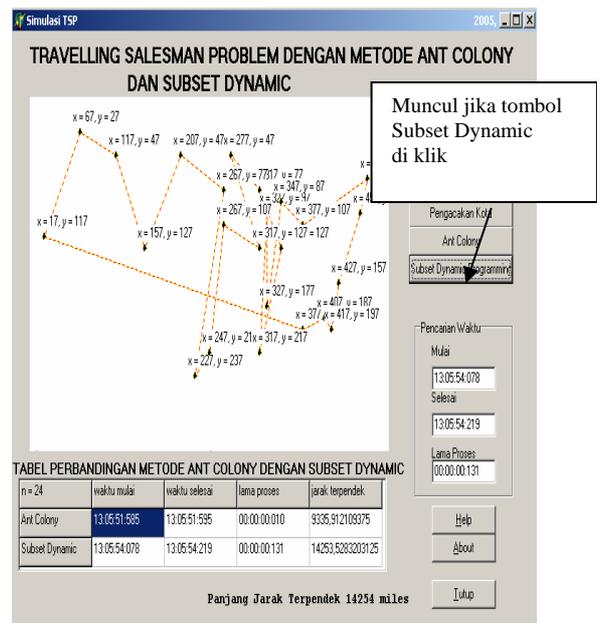
perjalanan. Untuk lebih jelasnya seperti terlihat pada gambar dibawah.



Gambar 4. Pengecekan kota menggunakan metode Ant Colony

Pada saat menekan tombol Subset Dynamic programming, program membuat rute perjalanan yang dapat dimulai dari kota manapun juga, karena akan melalui rute yang sama. Pada tampilan di atas kita dapat melihat tampilan Subset Dynamic programming yang menampilkan hasil waktu yang diperoleh pada saat running dan menampilkan jarak terpendek yang ditempuh pada saat kita menggunakan algoritma Subset Dynamic programming. Hasil perjalanan tersebut adalah hasil perjalanan terbaik yang ditempuh dalam melakukan perjalanan. Untuk lebih jelasnya seperti terlihat pada gambar dibawah ini

Tampilan simulasi yang dilaksanakan dalam Subset Dynamic Programming:



Gambar 5. Tampilan Menu tabel hasil perbandingan antara algoritma Ant Colony dengan Subset Dynamic Programming yang menampilkan hasil waktu dan jarak terpendek yang ditempuh

Kita dapat melihat hasil waktu dan jarak terpendek yang ditempuh oleh kedua algoritma. Pada tampilan diatas kita dapat melihat tampilan hasil kedua algoritma yang menampilkan perbedaan hasil waktu yang diperoleh pada saat running dan menampilkan perbedaan jarak terpendek yang ditempuh kedua algoritma tersebut. Perbedaan yang nyata tersebut dapat kita lihat pada tabel hasil perbandingan diatas. Untuk lebih jelasnya seperti terlihat pada gambar 4, 5.

4. ANALISIS HASIL PERCOBAAN

Setelah kita melakukan percobaan dengan menggunakan kedua metode antara *ant colony* dengan *subset dynamic* maka akan terlihat hasil yang cukup jelas menandakan bahwa dari kedua metode ini terdapat perbedaan yang sangat signifikan sehingga dalam penelitian ini dapat diambil suatu kesimpulan metode mana yang sebaiknya digunakan dalam kasus *Travelling Salesmen Program* ini.

Dalam pengujian analisis hasil diadakan lima kali analisis mulai dari 10 kota samapai 50 kota yang masing-masing dilakukan 5 kali pengujian. Diasumsikan bahwa dengan 5 kali pengujian ini dapat menghasilkan suatu hasil analisa yang sangat baik untuk diambil suatu keputusan pemilihan metode yang baik.

Dibawah ini akan diuraikan hasil-hasil analisa pengujian seperti yang terlihat dibawah ini:

Tabel 1. Analisis Hasil Percobaan Algoritma Ant Colony dan Subset Dynamic Programming

Banyak Kota	Ant Colony		Subset Dynamic	
	Lama Proses (M sec)	Jarak Tempuh (mile)	Lama Proses (M sec)	Jarak Tempuh (mile)
10	10	7908,95	100	7908,95
	10	6964,3	80	7056,3
	10	7075,19	80	7075,19
	10	6956,65	70	7224,95
	10	7009,57	90	8230,68
20	10	9273	110	11738,74
	10	8911,67	110	9299,13
	20	9681,44	240	11382,69
	10	9810,98	191	11773,79
	10	9139,61	210	13369,13
30	20	11242,54	320	18338,63
	31	11160,34	290	15459,62
	20	11489,12	241	16188,26
	20	11400,49	301	20270,05
	10	10827,11	210	17236,51
40	40	11558,19	350	25744,15
	41	12587,76	410	23973,43
	20	11458	381	23789,05
	40	11901,53	420	21582,31
	50	12059,62	351	23174,50
50	40	14299,13	510	30111,18
	91	13919,39	380	30899,92
	60	14161,58	440	26756,86
	40	13430,7	341	25560,15
	70	14155,31	341	29696,18

Kita dapat melihat hasil analisis percobaan antara kedua algoritma diatas, yang mana algoritma *ant colony* masih lebih baik dari algoritma *subset dynamic programming*. Hasil analisis menunjukkan hasil perbedaan waktu dan jarak terpendek yang ditempuh dengan menggunakan algoritma *ant colony* jauh lebih baik.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan proses pembuatan, hasil percobaan dan pembahasan secara keseluruhan dari sistem ini, dapat diambil kesimpulan, sebagai berikut:

- o Algoritma *Ant Colony System* merupakan algoritma yang dapat dibandingkan dengan algoritma lain, yaitu algoritma *Subset Dynamic Programming*, yang digunakan untuk mendapatkan tingkat pembandingan waktu TSP.
- o Hasil percobaan membuktikan bahwa banyaknya jumlah kota sangat mempengaruhi waktu proses penyelesaian, semakin banyak kota semakin lama waktu yang diburuhkan dan begitu juga sebaliknya.
- o Waktu TSP algoritma *Ant Colony* relatif lebih cepat daripada waktu TSP algoritma *Subset Dynamic Programming*, karena algoritma *Ant Colony* menggunakan logika pembesaran radius titik koordinat OUT terhadap obyek objek yang mendekati wilayah titik akhir simulasi TSP. Sedangkan prinsip kerja *Subset Dynamic Programming* yang berlawanan dengan *Ant Colony* yakni menghapus jejak (*pheromone*) atau route yang terdahulu dibangkitkan dalam simulasi TSP membuat subset dynamic lebih lambat .
- o Sistem yang dibuat mudah dijalankan, karena tampilan yang menarik dan mudah dioperasikan serta informasi hasil yang diberikan jelas.

5.2 Saran

Sistem yang telah dibuat ini masih harus lebih banyak diperdalam, dan sistem memerlukan pengembangan lebih baik, antara lain:

- o Sistem ini masih dapat dikembangkan untuk menyelesaikan masalah kombinatorial selain TSP, misalnya *Annalaing Method, Tabu Search, Branch and Branch*.
- o Sistem ini masih perlu banyak perbaikan karena sistem ini sangat dibutuhkan dalam berbagai bidang seperti transportasi mengetahui perjalanan singkat dan cepat dari satu kota ke kota lain.
- o Sebagai kelanjutan dari proses analisis perbandingan kedua algoritma ini, sistem ini dapat juga dibandingkan dengan metode/ algoritma lainnya, untuk mengetahui ketepatan hasil yang didapat jauh lebih baik lagi . Serta dapat juga membandingkan lama waktu proses yang dibutuhkan antara kedua algoritma diatas dengan algoritma - algoritma lain. Sehingga

untuk pengembangan kasus yang lain dapat dipilih algoritma yang lebih sesuai dan lebih efektif.

DAFTAR PUSTAKA

- [1] Pramono, Joko, *Mudah menguasai Delphi 5*, Jilid 1 dan 2, PT Elex Media Komputindo, Bandung, 1999
- [2] Baase, Sara, *Computer Algorithm, Introduction to Design and analysis*, Second Edition, 1988.
- [3] Deo, Narsingh; Paryono Petrus (penterjemah). *Teori Graf, Diktat Kuliah Fakultas Teknik Informatika UKDW, Yogyakarta, 1989*
- [4] Even, Shumm, *Graph Algorithm*, United Stated of America: Computer Science Press, Inc, 197
- [5] Nugroho, Eko, Ir; *Pemrograman Terstruktur Dengan Pascal*, Andi Offset, Yogyakarta, 1996
- [6] Dorigo, M and Gambardella, L.M. (1997b), *Ant Colony System; A Cooperative learning Approach to The Travelling Salesman Problem*, *IEEE Transaction on Evolutionary Computer* Vol 1, Universite Libre de Bruxelles, Belgium, [http:// iridia.ulb.ac.be/ dorigo/dorigo.html](http://iridia.ulb.ac.be/dorigo/dorigo.html) (10 Juli 2005).

