PENYELESAIAN MULTIPLE TRAVELLING SALESMAN PROBLEM (M-TSP) DENGAN ALGORITMA K-MEANS CLUSTERING-GENETIKA

Jihan [1] **Sri Mardiyati** [2]

[1] Matematika, FMIPA Universitas Indonesia
Kampus UI Depok, 16424, Indonesia
email : [1]jihan.thlb@gmail.com, [2]srimardiyati25@gmail.com

ABSTRACT

Multiple Travelling Salesman Problem (M-TSP) *is a problem of finding an optimal travel route from* n *cities by* m *salesmen with* $m < n$*, the condition is that each city can only be visited once and only by one salesman.* M-TSP *is a development of the* TSP *problem which involves more than one salesman.* M-TSP *Single Depot, where all the salesmen start travelling from the same city, will be discussed in this final project.* M-TSP *will be solved by using the K-Means Clustering-Genetic Algorithm that divides* n *cities to* m *clusters and applies the genetic algorithm to each cluster, then all the results obtained will be summed to determine the total mileage of the whole salesman. the results are expexted to be better than* TSPSG *software* .

Key words

Multiple Travelling Salesman Problem (M-TSP), *Travelling Salesman Problem* (TSP), *K-Means Clustering Algorithm, Genetics Algorithm.*

1. Pendahuluan

Transportasi digunakan manusia dalam kehidupan sehari-hari untuk menjalankan aktivitasnya, seperti bepergian dari satu tempat ke tempat lain, mengantarkan paket oleh kurir, distribusi bahan logistik, dan masih banyak lagi contoh lainnya. Dalam melakukan hal-hal tersebut, tentu manusia mempertimbangkan bagaimana agar jarak/waktu.biaya yang dihabiskan menjadi efisien sehingga semakin banyak yang bisa dilakukan agar makin banyak juga keuntungan yang diperoleh. Untuk mencapai hal itu, maka akan dicari rute perjalanan yang menyebabkan segala faktor yang dipertimbangkan menjadi efisien yang kemudian masalah ini lebih dikenal dengan nama *Travelling Salesman Problem* (TSP). TSP adalah permasalahan menentukan rute perjalanan mengunjungi sejumlah bergingga kota oleh seorang *salesman* dengan syarat masing-masing kota dikunjungi hanya satu kali dan *salesman* mengakhiri perjalanan pada kota yang sama saat ia memulai, sehingga rute yang didapatkan adalah rute yang paling efisien (Gupta, 2013).

Terdapat 3 faktor yang biasa akan dioptimalkan yaitu faktor biaya yang dikeluarkan, jarak yang ditempuh dan waktu yang dihabiskan. Dalam pembahasan ini, akan difokuskan pada pengoptimalah faktor jarak. Berdasarkan jenisnya TSP dibagi menjadi 2 yaitu simetris (TSP) dan asimetris (ATSP). TSP simetris yaitu TSP dimana jarak antar kota sama apabila ditempuh dari arah yang berbeda, sedangkan TSP asimetris adalah TSP dengan jarak antar kota berbeda jika ditempuh dari arah yang berbeda. Dikehidupan nyata tentunya tidak hanya satu orang saja yang menempuh perjalanan dari sekian banyak kota yang harus dilalui, hal itu menyebabkan masalah TSP mengalami perluasan dan modifikasi. Berdasarkan banyaknya orang yang terlibat dalam rute perjalanan, TSP dibagi menjadi 2 yaitu TSP dan *Multiple*-TSP (M-TSP). TSP adalah pencarian rute yang dilakukan oleh satu orang *sales*, sedangkan M-TSP ialah pencarian rute oleh beberapa orang *sales*. M-TSP dibagi menjadi 2 berdasarkan jenis depotnya yaitu, *single depot* dan *multiple depot.* M-TSP *single depot* adalah masalah pencarian rute ketika seluruh *sales* memulai dan mengakhiri perjalanan disatu kota yang sama, sedangkan M-TSP *multiple depot* adalah masalah pencarian rute ketika seluruh sales memulai dan mengakhiri perjalanan di *m* kota yang berbeda yaitu sebanyak jumlah *salesman* tersebut.

Keduanya, baik TSP dan M-TSP dapat diselesaikan dengan metode eksak dan metode heuristik. Metode eksak digunakan untuk ukuran masalah yang relatif kecil sehingga hasil yang didapat merupakan hasil yang optimal. Sedangkan, metode heuristik digunakan apabila metode eksak tidak lagi efesien yaitu ketika ukuran permasalahan relatif besar sehingga hasil yang diperoleh merupakan

solusi pendekatan dari solusi yang optimal. Beberapa metode eksak untuk menyelesaikan TSP yaitu *Branch and Bound* (Wiener, 2003), dan *Branch and Cut* (Gekdenhuvs, 2012). Metode heuristik untuk menyelesaikan TSP antara lain Algoritma Geneika (Potvin,1996), Algoritma *Ant Colony* (Dorigo, 1996) dan Algoritma Genetika-2 *local search* (GA2OPT) (Tohrusman, 2006). Untuk masalah M-TSP metode eksak yang digunakan adalag *Lagrangian Relxation* (Yadlapali, 2008) dan *Branch and Bound* (2009). Sedangkan metode heuristik untuk M-TSP adalah *Modeified Genetic Algorithm* (Tang & Liu, 2000), *Modified Ant Colony* (Junjie & Dingwei, 2006) dan GA2OPT (Sedighpour, 2011).

Dalam tulisan ini akan dibahas mengenai pencarian rute yang dilakukan oleh banyak *salesman* dari depot yang sama dengan jarak antar kotanya tidak memperhatikan arah tempuh yaitu *Symetric Multiple Travelling Salesman Problem Single Depot* yang akan disebut sebagai (M-TSP). Masalah ini akan diselesaikan dengan menggunakan gabungan dari 2 metode heuristik yaitu algoritma *k-means clustering* ditahapan awal lalu dilanjutkan dengan algoritma genetika yang kemudian metode ini ditulis Algoritma *K-Means Clustering*-Genetika.

2. Sistem Persamaan Linier

2.1 Formulasi Masalah dan Model Matematis *Multiple Travelling Salesman Problem* (M-TSP)

Travelling Salesman Problem (TSP) adalah masalah penentuan rute perjalanan yang ditempuh oleh satu *salesman*, dalam aplikasinya tentu dapat digunakan lebih dari satu *salesman* yang melakukan perjalanan. TSP dengan lebih dari satu *salesman* dikenal dengan *Multiple Travelling Salesman Problem* (M-TSP), yang secara umum didefinisikan sebagai masalah pengaturan rute perjalanan $m > 1$ *sales* untuk mengunjungi sejumlah $n > m$ lokasi dimana *salesman* mengakhiri perjalanan dikota yang sama dengan kota awal perjalanan dan masing-masing lokasi hanya dikunjungi tepat satu kali sehingga total jarak yang dilalui menjadi minimum.[Carter, 2005].

Kota tempat *salesman* memulai dan mengakhiri perjalanannya disebut depot, dan kota yang lain disebut kota tujuan. Depot tiap *salesman* dapat sama atau berbeda, ketika depotnya sama maka permasalahannya disebut *Single Depot Multiple Travelling Salesman Problem* yang kemudian disebut sebagai M-TSP saja, sedangkan jika depotnya berbeda disebut sebagai *Multiple Depot-Multiple Travelling Salesman Problem (mM-TSP).*

Pada TSP,masalah dapat direpresentasikan dalam bentuk graf untuk memudahkan pemahamannya, maka masalah M-TSP juga dapat direpresentasikan dalam bentuk graf. M-TSP pada pembahasan ini direpresentasikan dengan graf komplit tidak berarah *(complete undirected graph).* Berikut ini akan diterangkan notasi yang digunakan dalam pembentukan model untuk permasalahan M-TSP yaitu :

- $G(V, E)$ adalah sebuah graf komplit tidak berarah dengan :
  - $V = \{1,2, \dots, n\}$ adalah himpunan kota.
  - $S = \{2, \dots, n\}$ adalah kumpulan kota yang dilalui dalam rute perjalanan masing-masing *salesman* selain depot, dimana $|S|$ adalah jumlah kotanya.
  - $E = \{(x_{ij}) : i,j \in V\}$ adalah himpunan jalan yang menghubungi kota i dan kota j. d_{ij} adalah bobot jalan yang merepresentasikan jarak kota i dan kota j, $d_{ij} \neq 0$ untuk $i \neq j$ dan $d_{ij} = 0$ apabila $i = j$. Artinya jika *sales* tidak berjalan maka jarak nya adalah nol, jika berjalan dari kota i ke kota j maka jaraknya tidak sama dengan nol. Karena M-TSP simetris maka $d_{ij} = d_{ji}$.
- Misalkan m adalah jumlah *salesman* yang menempuh perjalanan sesuai dengan rutenya masing-masing.
- $D = \{d_{ij} ;\ i = 1, \dots, n ;\ j = 1, \dots, n\}$ adalah himpunan variabel jarak dalam fungsi tujuan yang akan dioptimalkan.
- $Z_j ;\ j = 1,2, \dots, k$ adalah *centroid*. *Centroid* adalah pusat dari kluster. Sedangkan $Z_{ji},\ i = 1,2, \dots, n ; j = 1,2, \dots, k$ adalah jarak antara *centroid* j dan kota i.
- $n_j ;\ j = 1,2, \dots, k$ menyatakan jumlah kota didalam kluster ke-j

Dengan asumsi :

- Berlaku pertidaksamaan segitiga untuk variabel jarak $d_{ij} + d_{jk} \geq d_{ik}$, yaitu jarak kota i ke j ditambah jarak j ke k akan lebih besar sama dengan jarak kota i ke kota k.
- Karena M-TSP *single depot*, maka depot untuk semua *salesman* sama sehingga depot dikunjungi dan ditinggalkan lebih dari satu kali.
- Depot dipilih terlebih dahulu.

- Kota yang menjadi depot dianggap sebagai kota 1.
- Jika salah satu salesman telah melewati jalan dari kota i ke kota j, maka *salesman* yang lain tidak boleh melewati jalan tersebut, mereka harus melewati jalan dengan i dan j yang berbeda-beda. Artinya setiap kota (kecuali depot) dan jalan hanya boleh dikunjungi tepat satu kali dan oleh satu orang *sales* saja. Sehingga rute tiap *sales* berbeda.
- Rute antar *salesman* harus saling lepas, artinya dalam rute seorang *sales* tidak boleh terdapat *subtour sales* yang lain.

Variabel keputusan yang digunakan dalam model matematis masalah M-TSP adalah sebagai berikut :

- Variabel keputusan biner $x_{ij} \in \{0,1\}, \forall (i,j) \in E$

$$x_{ij} = \begin{cases} 1, \text{jika } \textit{salesman} \text{ berjalan dari kota i ke kota j} \\ 0, \text{jika tidak} \end{cases}$$

Pastikan pada rute seorang *salesman* tidak terdapat *subtour sales* yang lainnya, dan haya jalan yang berasal dari depot saja yang menuju kota tujuan yang pertama dikunjungi. Seluruh jalan yang menghubungkan antar dua kota dari n kota yang ada dituliskan secara lengkap. Jika sebuah jalan dari kota i ke kota j dilewati dalam rute perjalanan salesman, maka x_{ij} memiliki bobot sebesar jarak antar kotanya. Sedangkan jika tidak dilewati, maka tidak memiliki bobot karena nilai dari variabel keputusan binernya adalah nol. Sehingga fungsi objektif dari masalah M-TSP didapatkan dari penjumlahan dari perkalian antara seluruh variabel keputusan biner dengan variabel jaraknya yang dituliskan sebagai berikut : [Nallusamy R, 2012]

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij}$$

Dengan kendala ;

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad ,j = 2, \dots, n$$

$$\sum_{i=1}^{n} x_{ij} = m \qquad ,j = 1$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad ,i = 2, \dots, n$$

$$\sum_{j=1}^{n} x_{ij} = m \qquad ,i = 1$$

[Bektas, 2005]

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \qquad ,\forall S \subseteq V \backslash \{1\},\ S \neq \emptyset$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \geq 1 \qquad ,\forall S \subseteq V \backslash \{1\},\ S \neq \emptyset$$

dimana,

$$x_{ij} = \begin{cases} 1, \text{ jika } \textit{salesman} \text{ berjalan dari kota i ke kota j} \\ 0, \text{lainnya} \end{cases}$$

2.2 Penyelesaian M-TSP dengan Algoritma K-Means Clustering – Genetika

Algoritma *K-Means Clustering* adalah salah satu algoritma sederhana yang dapat menyelesaikan masalah pengelompokkan/pengklusteran. Prosedurnya sederhana dan mudah diaplikasikan terhadap himpunan data yang akan dikelompokkan menjadi sejumlah kluster. Penyelesaian masalah M-TSP diawali dengan membagi n kota yang ada menjadi m kluster. Pada proses pembuatan kluster, dipilih satu buah kota yang menjadi depot untuk semua *sales* dan dipilih lagi sejumlah $m-1$ kota secara random dari sejumlah kota yang ada selain depot. Depot dan $m-1$ kota yang akan dijadikan sebagai *centroid*, dimana *centroid* merupakan pusat dari tiap kluster. Kota selain *centroid* merupakan kota tujuan, inilah yang akan dipartisi menjadi beberapa kluster. Proses kerja algoritma *K-Means Clustering* dijelaskan dalam tahapan dibawah ini dan diberikan juga diagram alurnya.

- Tentukan jumlah kluster (k) yang diinginkan, yang sesuai dengan jumlah *salesman* (m).
- Pilih kota yang akan dijadikan sebagai depot.
- Pilih sejumlah $k-1$ kota yang akan dijadikan *centroid* secara random. Depot yang dipilih pada langkah (2) juga merupakan *centroid*. Sehingga jumlah *centroid* adalah k buah. Beri nomor urut untuk seluruh *centroid*, *centroid* ke-1 hingga k.
- Hitung jarak dari setiap kota yang ada dengan masing-masing *centroid*. Perhitungan jarak menggunakan jarak Euclid.

$$Z_{ji} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad ;\ i = 1,2, \dots, n\ , j = 1,2, \dots, k$$

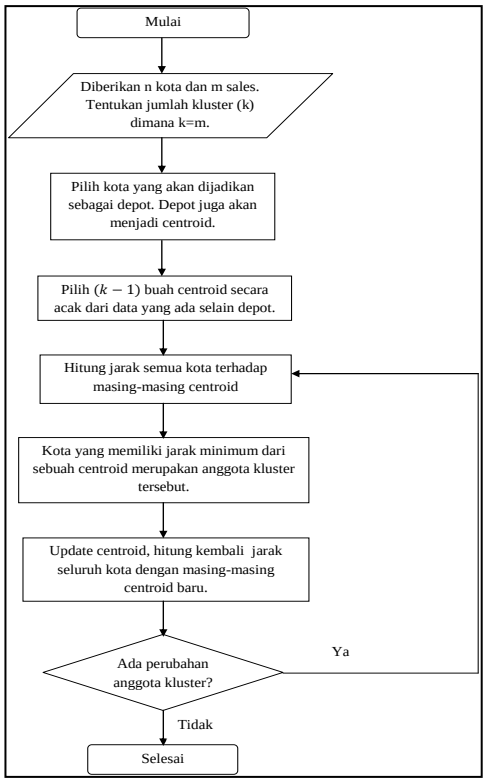
Z_{ji} = jarak kota i terhadap *centroid* j

- Kelompokkan kota kedalam kluster yang memiliki jarak paling minimum.
- Setelah seluruh kota telah masuk kedalam kluster-kluster, hitung *centroid* yang baru dengan cara menjumlahkan koordinat kota dalam satu kluster dibagi dengan jumlah kota yang ada dalam kluster tersebut. Lakukan hal yang sama pada kluster yang lain.

$$Z_j^* = \frac{1}{n_j} \sum_{x_i \in C_j} x_i \quad , j = 1,2, \dots, k$$

n_j = jumlah kota dalam kluster ke-j

- Jika terdapat perubahan anggota kluster maka ulangi langkah 3 hingga tidak ada perubahan anggota pada setiap kluster. Jika *centroid* yang baru tidak berubah dari sebelumnya $Z_j^* = Z_j$ maka proses berhenti. Karena *centroid* yang tidak berubah menyebabkan anggota kluster juga tidak berubah

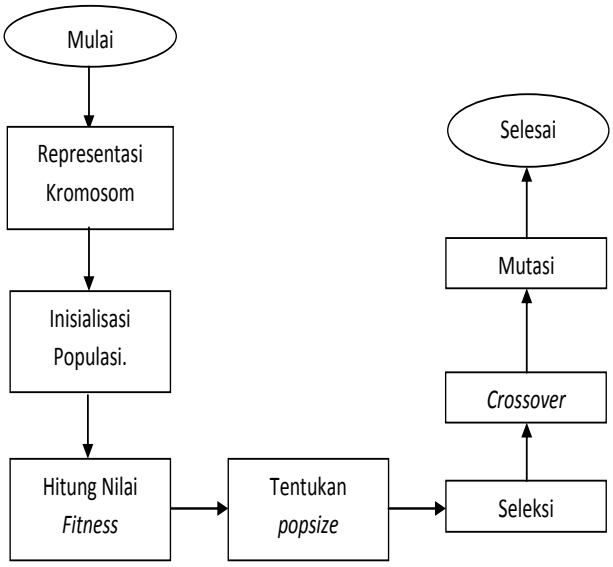


Gambar 2.1 Diagram Alur Algoritma *K-Means Clustering*

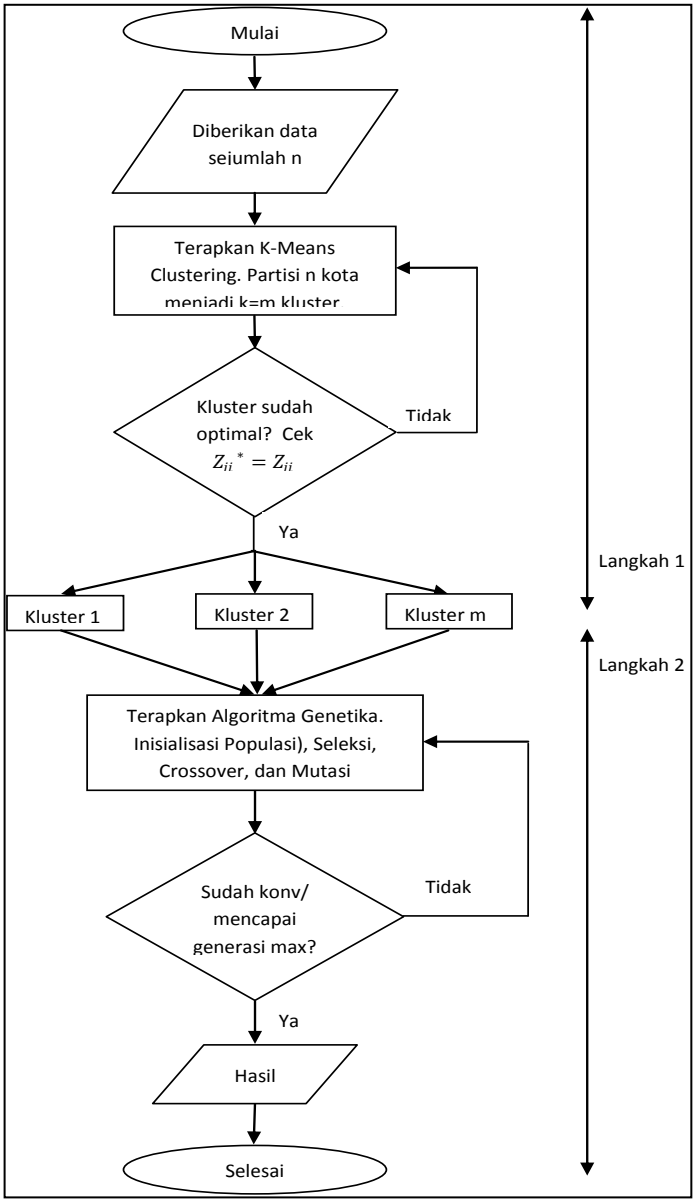
Setelah mendapatkan hasil berupa kota-kota yang sudah dipartisi menjadi beberapa kluster, kemudian masing-masing kluster tersebut akan diselesaikan dengan algoritma genetika. Masalah yang sebelumnya adalah M-TSP, pada tahapan ini menjadi sejumlah masalah TSP, karena masing-masing kluster yang didapatkan merupakan masalah TSP.

Algoritma genetika pertama kali ditemukan oleh John Holland dari Universitas Michigan. John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi dapat diformulasikan dengan genetika. Metode ini banyak diaplikasikan untuk menyelesaikan masalah dibeberapa bidang dikarenakan kemudahan impelementasi algortima genetika pada setiap masalah optimasi. Secara umum, langkah kerja algoritma genetika terdiri dari representasi kromosom, inisialisasi populasi, nilai fitness, seleksi, perkawinan silang (crossover) dan mutasi. Untuk lebih detailnya, langkah-langkah algoritma genetika dijelaskan sebagai berikut.

- Bangkitkan sejumlah populasi awal yang berisi sejumlah kromosom yang didalamnya terdapat untaian gen-gen yang merupakan urutan perjalanan kota-kota.
- Hitung nilai *fitness* dari setiap kromosom.
- Tetapkan *popsize* yang diinginkan, kemudian seleksi kromosom pada populasi awal hingga mendapatkan kromosom sebanyak *popsize*.
- Setelah mendapatkan kromosom hasil seleksi, tetapkan probabilitas *crossover* (p_c) dalam hal ini digunakan adalah 0.7. Bangkitkan bilangan random [0.1] pada setiap kromosom, kromosom dengan bilangan random kurang dari p_c maka akan dilakukan *crossover*. Jika kromosom hasil *crossover* memiliki *fitness* lebih baik dari kromosom awal, maka kromosom awal digantikan oleh kromosom hasil *crossover*.
- Tetapkan probabilitas mutasi (p_m), dalam hal ini digunakan p_m=0.1. bangkitkan bilangan random [0.1] pada setiap kromosom, kromosom yang memiliki bilangan random kurang dari p_m maka akan dilakukan mutasi. Jika kromosom hasil mutasi memiliki *fitness* lebih baik dari kromosom awal, maka kromosom awal digantikan oleh kromosom hasil mutasi.



Gambar 2.2 Diagram Alur Algoritma Genetika



Gambar 2.3 Diagram Alur *K-Means Clustering*-Genetika

3. Hasil Percobaan

Penyelesaian masalah M-TSP dengan Algoritma *K-Means Clustering*-Genetika diimplementasikan terhadap beberapa dataset yang diambil dari web TSPLIB95 yaitu pr76.tsp, rat99.tsp, dan pr124.tsp. jumlah *salesman* yang akan menempuh perjalanan adalah 6 orang pada setiap dataset.

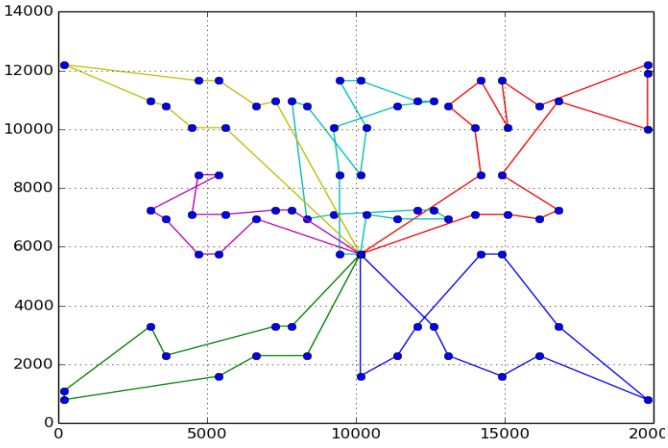
Program yang telah dibuat membutuhkan beberapa input yaitu: jumlah kluster, kota mana yang menjadi depot, jumlah populasi awal dan jumlah populasi orang tua (*popsize)*. Selain input, pada listing program juga telah ditetapkan langsung probabilitas *crossover*, probabilitas mutasi, dan maksimum generasinya. Probabilitas *crossover* dan mutasi yang digunakan secara berturut-turut adalah 0.7 dan 0.1. Kota yang menjadi depot pada tiap dataset berbeda-beda, untuk dataset pr76.tsp berada di kota 27, rat99.tsp di kota 50 dan pr124.tsp di kota 67.

Selain diselesaikan dengan Algoritma *K-Means Clustering*-Genetika, masalah M-TSP dengan dataset yang disebutkan diatas juga diselesaikan dengan sebuah *software* TSPSG yang hasilnya ditampilkan dalam tabel dibawah ini beserta dengan error relatifnya. *Software* TSPSG bekerja berdasarkan metode eksak yaitu *Branch and Bound*. Hasil keduanya ditulis lalu dihitung selisihnya untuk mendapatkan error relatif dengan hasil yang didapatkan pada TSPSG dijadikan sebagai acuan karena metodenya merupakan metode eksak.

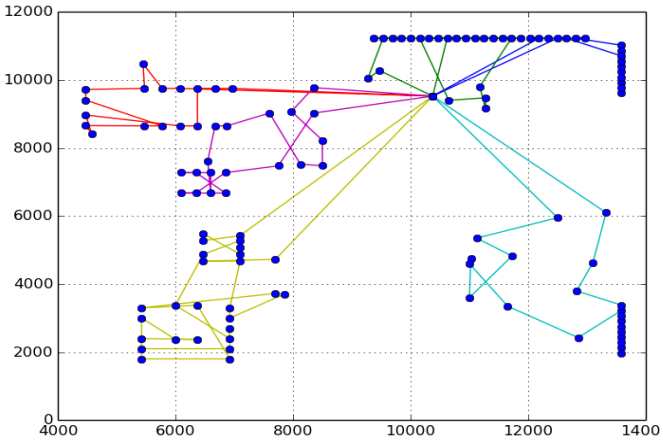
Dibawah ini akan ditampilkan gambar output rute perjalanan salesman untuk masing-masing dataset, tabel jarak tempuh dan tabel komparasi jarak tempuh antara Algoritma *K-Means Clustering*-Genetika dan *software* TSPSG

Tabel 3.1 Tabel Komparasi *K-Means Clustering*-Genetika & TSPSG

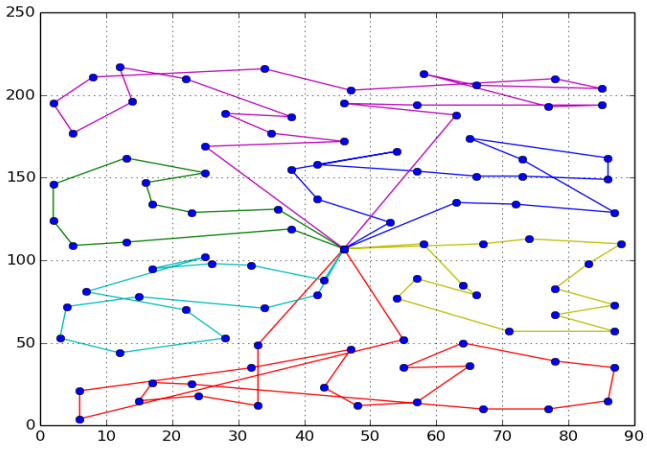
Nama Dataset	AKCG	TSPSG	Error Relatif
Pr76.csv	166712 .788383	154673.55	0.07783644
Rat99.csv	1979.6095014	1837.3008	0.07745531
Pr124.csv	120423. 16855	92177.983	0.30642009



Gambar 3.1 Rute Perjalanan pr76.tsp



Gambar 3.2 Rute Perjalanan rat99.tsp



Gambar 3.3 Rute Perjalanan pr124.tsp

Tabel 3.2 Output dataset pr76.tsp

	Sales	Jarak Tempuh	Total
Algoritma *K-Means Clustering*-Genetika	1	29633. 5	166712 .78
	2	25086.44	
	3	35598.98	
	4	31723.56	
	5	19111.05	
	6	25559.23	
TSPSG	1	27509.1	154673. 55
	2	25086. 5	
	3	31827. 97	
	4	25702.34	
	5	18988.44	
	6	25559.2	

Tabel 3.3Output dataset rat99.tsp

	Sales	Jarak Tempuh	Total
Algoritma *K-Means Clustering* - Genetika	1	243.11	1979.60
	2	337.55	
	3	472.61	
	4	372.35	
	5	282.75	
	6	271.21	
TSPSG	1	261.79	1837.30
	2	331.10	
	3	433.80	
	4	296.15	
	5	259.51	
	6	254.92540	

Tabel 3.4 Output dataset pr124.tsp

	Sales	Jarak Tempuh	Total
Algoritma *K-Means Clustering*-Genetika	1	11283. 96	120423.16
	2	14830.39	
	3	18087.35	
	4	24217.89	
	5	18133.14	
	6	33870.41	
TSPSG	1	9194.61	92177.98
	2	9040.75	
	3	15383.13	
	4	20238.72	
	5	13209.18	
	6	25111.58	

4. Kesimpulan

Setelah dilakukan penyelesaian terhadap masalah M-TSP dengan Algoritma *K-Means Clustering*-Genetika secara manual untuk contoh sederhana dan program untuk ukuran masalah yang lebih besar serta mencobanya juga pada software *TSP Solver and Generator* (TSPSG) didapatkan beberapa kesimpulan sebagai berikut :

1. Penggunaan Algoritma *K-Means Clustering*-Genetika dalam menyelesaikan masalah M-TSP relatif lebih mudah dikarenakan dengan melakukan pengklusteran dapat diketahui kota yang dilalui masing-masing *salesman*.
2. Menyelesaikan M-TSP menggunakan Program Algortitma *K-Means Clustering*-Genetika pada tiga dataset pr76.tsp, rat99.tsp dan pr124.tsp yang diambil di TSPLIB95 hanya memakan waktu 3 sampai 5 menit saja. Sangat jauh jika dibandingkan dengan penyelesaian menggunakan *software*

TSPSG yang memakan waktu 10 hingga 40 kali lipat lebih lama.

3. Hasil yang didapatkan pada program Algoritma *K-Means Clustering-Genetika* tidak lebih baik dari hasil yang diperoleh dengan *software* TSPSG. Meskipun begitu, ada faktor yang menjadi pertimbangan mengapa penulis menyimpulkan lebih praktis jika masalah M-TSP menggunakan Algoritma *K-Means Clustering* yaitu, lamanya waktu dan tenaga yang dihabiskan untuk memasukkan satu persatu entri dari matriks jarak untuk semua *salesman*, dan selisih hasil yang didapatkan juga tidak terlalu jauh hanya berkisar 7-30 persen saja.

REFERENSI

[1] Jacob, Bill. 1990. *Linear Algebra.* United States of America: W.H Freeman and Company New York.

[2] Kardi Teknomo,PhD ,"K-Means Clustering Tutorial" Teknomo ,Kardi,K-means Clustering Tutorials. http://people.revoledu.com\kardi\tutorial\kmeans\. Diakses pada 27 Februari 2015 11:31.

[3] Mahmudy, WF 2013, *Algoritma Evolusi, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang.*

[4] Michalewicz, Zbigniew. (1995). *Genetic Algorithms + Dara Structures = Evlution Programs 3th ed.* United States of America: Springer-Verlag Berlin Heidelberg New York.

[5] Nallusamy, R. : "Optimization of Non-Linear Multiple Traveling Salesman Problem Using K-Means Clustering, Shrink Wrap Algorithm and Meta-Heuristics". International Journal of Nonlinear Science. No.4,pp.480-487. 2009.

[6] Obitko, 1998. Dipetik April 2012, dari www.obitko.com/tutorials/genetic-algorithms

[7] Purnamasari, Riska Hardini. (2010). Impelementasi Algoritma Genetika Untuk Pencarian Rute Minimum Dalam *Travelling Salesman Problem.* Bandung: JBPTUNIKOMPP.

[8] Taha, H., 2007. *Operational Research: An Introduction.* New Jersey: Priceton Edicution, Inc.

[9] Tohirin, Akhrmad. (2011). Aplikasi Algoritma Genetika Untuk Optimasi Penjadwalan Mata Kuliah Di Jurusan Teknik Komputer UNIKOM Bandung: JBPTUNIKOMPP.

[10] Tohrusman, Qfandy Desaindo. (2006). Kinerja Hibrida Algoritma Genetik dan 2-opt *Local Search* Dalam Menyelesaikan *Travelling Salesman Problem"*. Depok. Departemen Matematika. Universitas Indonesia.

[11] Whitley, 2002. *Genetic Algorithm Evolutionary Computing.* Van Nostrand's Scientific Encyclopedia

[12] Winston, W., 2003. *Operational Research: Application and Algorithm (4th edition).* Cengage Learning

[13] www.lecturer.pens.ac.id. Kecerdasan Buatan Buku Bab 7 AlgoritmaGenetika diakses pada 12 Maret 2015 20:09.