PERANCANGAN APLIKASI PENJADWALAN TRAVELING SALESMAN PROBLEM DENGAN ALGORITMA GENETIKA

Hendy Tannady; Andrew Verrayo Limas

Industrial Engineering Department, Faculty of Engineering, Binus University
Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
hendy_tan3003@yahoo.com

ABSTRACT

Supply chain management plays an important role in enhancing the efficiency and effectiveness of manufacturing industry business process. In this research, the problem is taken from a sales division in a company in determining the optimal sequence when delivering goods into nine cities. This problem is often referred as travelling salesman problem. This problem is considered important since the optimal sequence can cut off operational cost. Creating an artificial intelligence for the company in determining the location and the optimal sequence of delivering goods is the main objective of this research. A genetic algorithm is utilized to determine the location and the optimal sequence. While for processing the data and concluding the result, researcher designed a Java-based application that provides the capability of automatic computing. The result of this computation is a sequence of locations with a fitness number for each. The best fitness number for the sequence location will be used for the final result and the conclusion to answer the company's problem.

***Keywords**: supply chain management, traveling salesman problem, genetic algorithm*

ABSTRAK

Manajemen rantai pasok memegang peran penting dalam peningkatan efisiensi dan efektivitas proses bisnis industri manufaktur. Dalam penelitian ini, permasalahan diambil dari sebuah divisi penjualan dari sebuah perusahaan yang mengalami masalah dalam menentukan urutan yang optimal ketika melakukan pengiriman barang ke sembilan kota dan permasalahan ini sering disebut sebagai permasalahan perjalanan penjual. Permasalahan ini sangat penting karena biaya operasional dapat ditekan melalui urutan yang optimal dan dijadikan sebagai tujuan dari penelitian, seperti menciptakan suatu intelegensi buatan bagi perusahaan dalam menentukan lokasi dan urutan pengiriman yang optimal. Peneliti menggunakan algoritma genetika sebagai metode dalam menentukan lokasi dan urutan yang optimal. Sedangkan untuk mengolah data dan menyimpulkan hasil, dirancang aplikasi berbasis Java yang memberikan kapabilitas untuk melakukan komputasi secara otomatis. Hasil dari komputasi ini adalah urutan-urutan lokasi yang masing-masing memiliki angka fitness. Urutan lokasi yang memiliki angka fitness terbagus akan dijadikan sebagai hasil akhir dan kesimpulan dalam menjawab masalah perusahaan.

***Kata kunci**: manajemen rantai pasok, permasalahan perjalanan penjual, algoritma genetika*

PENDAHULUAN

Besarnya pengaruh dari transportasi terhadap kesinambungan proses kerja *supply chain management* turut menempatkan transportasi sebagai media *supporting* yang harus mendapat tempat strategis dalam pembahasan, diskusi dan rencana kerja pelaku industri dalam penanganan *continous improvement* yang berkelanjutan. Bertambahnya beberapa faktor yang tidak dapat ditanggulangi dalam hal transportasi seperti kemacetan, jarak yang jauh, dan sebagainya membuat para manajer harus mengambil jalan lain untuk mengatasi tingginya biaya operasional yang diakibatkan oleh transportasi ini.

Menanggapi permasalahan di atas, peneliti bertujuan untuk memecahkan salah satu permasalahan dalam perusahaan seperti menentukan urutan lokasi pengiriman barang yang akan diolah dengan menggunakan metode algoritma genetika (*Genetic Algorithm*). Kylie Bryant (2000), menyebutkan bahwa algoritma ini cocok dalam memecahkan masalah penentuan urutan lokasi. Dalam pengolahan datanya, Larranaga et. al (1999) mengatakan bahwa teknik penyandian merupakan salah satu kunci utama yang menentukan keberhasilan dalam menentukan urutan lokasi ini. Lebih lanjut lagi, Kusumadewi & Purnomo (2005) memberikan langkah-langkah yang detail dalam merumuskan setiap langkah pengolahan data dengan menggunakan teknik pemrograman Java yang akan digunakan. Harapan dari peneliti adalah urutan lokasi ini dapat digunakan oleh perusahaan dalam menentukan jadwal pengiriman dan sekaligus memberikan dampak bagi pengurangan biaya operasional yang kian hari kian meningkat. Adapun manfaat dari penelitian ini adalah memberikan wawasan bagi para peneliti lainnya dalam memberikan solusi alternatif terkait dengan masalah perjalanan penjual, serta manfaat yang besar bagi divisi penjualan di perusahaan.

METODE

Penelitian ini dilakukan dengan menggunakan algoritma genetika (*Genetic Algorithm*). Pada algoritma ini, teknik pencarian solusi dilakukan secara bersamaan dalam suatu populasi. Entitas yang berada di dalam populasi ini dinamkan kromosom, di mana kromosom ini merupakan solusi yang akan diolah kembali. Kromosom-kromosom ini akan melalui beberapa iterasi dan berevolusi menjadi kromosom anak. Proses ini dinamakan sebagai generasi. Selanjutnya, setiap generasi ini akan melalui beberapa evaluasi dengan alat ukur yang dinamakan fungsi *fitness*. Fungsi ini berguna untuk menentukan kualitas dari kromosom dalam populasi. Generasi berikutnya disebut sebagai anak (*offspring*) di mana generasi ini merupakan penyilangan dari dua kromosom induk sebelumnya. Penyilangan ini dinamakan sebagai *crossover* dan kromosom ini akan melalui modifikasi melalui operator mutasi. Hasil dari algoritma ini adalah kromosom terbaik dari beberapa generasi yang akan dijadikan solusi bagi permasalahan. Pengolahan data dilakukan dengan membuat aplikasi pemrograman yang dikerjakan dengan bahasa pemrograman JAVA 1.7 dan NETBEANS 7.1 sebagai *supporting* fungsionalnya. Pemrograman membatasi kuantitas kota sebanyak sembilan (9) kota, dan memerlukan data simulasi jarak antar kota bukan sebagai riil data, melainkan sebagai pemicu simulasi.

HASIL DAN PEMBAHASAN

Pembahasan dimulai dengan melakukan pengumpulan data lokasi pada sembilan kota yang memiliki jarak yang berbeda-beda. Adapun asumsi yang digunakan bahwa setiap kota X_i denganX_{i+1} dan kebalikannya memiliki jarak yang sama. Untuk data jarak pada perancangan aplikasi ini, digunakan matriks *Euclidean* sebesar 9 x 9 seperti yang digambarkan pada tabel 1:

Tabel 1
Matriks Jarak antar Kota

		Jarak Kota(h_j)								
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Jarak Kota (h_i)	**1**	0.00	15.07	20.80	20.47	11.54	19.71	16.68	14.81	19.88
	2	15.07	0.00	16.27	16.81	11.28	11.45	12.36	20.59	15.62
	3	20.80	16.27	0.00	21.33	19.53	11.63	11.70	20.85	17.82
	4	20.47	16.81	21.33	0.00	10.26	15.17	17.04	21.75	18.51
	5	11.54	11.28	19.53	10.26	0.00	12.79	21.65	10.89	13.61
	6	19.71	11.45	11.63	15.17	12.79	0.00	17.24	14.69	21.49
	7	16.68	12.36	11.70	17.04	21.65	17.24	0.00	12.82	20.80
	8	14.81	20.59	20.85	21.75	10.89	14.69	12.82	0.00	10.22
	9	19.88	15.62	17.82	18.51	13.61	21.49	20.80	10.22	0.00

Dari data di atas, digunakan algoritma genetika (*Genetic Algorithm*) untuk memecahkan masalah optimisasi *Traveling Salesman Problem* di sembilan kota. Pemecahan masalah dilakukan dengan menggunakan bahasa pemrograman Java 1.7. Enam komponen selanjutnya akan dibahas selanjutnya. Representasi dari gen dan kromosom menggunakan struktur penomoran kota yang tertera pada tabel 1. Prosedur inisialiasi pada aplikasi ini menggunakan ukuran populasi sebesar 80 dengan peluang penyilangan sebesar 0.45 dan peluang mutasi sebesar 0.01.Nilai *fitness* pada aplikasi ini menggunakan fungsi total jarak dari kesembilan kota. Adapun total jarak kota dapat dihitung dengan menggunakan rumus:

$$h_t = \sum_{j=1}^{n} \sum_{i=1}^{n} h_{ij}$$

di mana:
h_t = Total jarak pengiriman
h_{ij} = Jarak dari kota i ke kota j
i = kota domain
j = kota destinasi

Algoritma dimulai dengan membangkitkan populasi awal dengan 80 individu. Nilai *fitness* dihitung dengan menggunakan rumus sebagai berikut:

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

di mana:
f = nilai *fitness*
Setelah itu dilakukan seleksi pada populasi dengan menggunakan metode *Roulette Wheel* yang sering disebut sebagai *Stochastic Sampling with Replacement.* Probabilitas dari setiap *fitness* dari individu dihitung dan digunakan sebagai indikator dalam memilih urutan individu dalam populasi. 80 buah bilangan acak dibangkitkan dan digunakan sebagai penentu dalam pengurutan individu tersebut.

Sedangkan untuk teknik penyilangan (*crossover*), digunakan metode *Order* (OX). Metode ini membentuk kromosom anak (*offspring*) dengan memilih sebagian jalur dari induk, kemudian menata ulang jalur anak berdasarkan urutan induk lainnya yang diacak. Penyilangan dimulai dengan dua induk sebagai berikut:

p_1	=	6	9	7	2	4	5	1	3	8
p_2	=	9	2	3	6	5	4	7	8	1

Setelah itu, diambil dua bilangan acak dengan rentang 1 sampai dengan n-1. Misalkan untuk posisi persilangan didapatkan nilai $r_1 = 4$ dan $r_2 = 8$, sehingga target penyilangan dapat dilihat dibawah ini:

p_1	=	6	9	7	2	4	5	1	3	8
p_2	=	9	2	3	6	5	4	7	8	1

Tempat yang diarsir ini akan ditempatkan pada setiap kromosom anak (o_1 dan o_2) dengan urutan yang sama. Sedangkan tanda x merupakan angka-angka yang akan diisi oleh persilangan kedua induk awal (p_1 dan p_2).

o_1	=	x	x	x	x	4	5	1	3	x
o_2	=	x	x	x	x	5	4	7	8	x

Dimulai dari induk kedua (p_2), urutan diambil dari setelah titik persilangan kedua (r_2) sampai dengan ujung dari induk kedua dan dilanjutkan dengan urutan individu pertama sampai dengan titik persilangan pertama (r_1). Sehingga didapatkan urutan dari penyilangan ini yaitu:

Urutan Penyilangan Kesatu: 1 – 9 – 2 – 3 – 6 – 5 – 4 – 7 – 8

Karena gen ke-4, 5, 1, dan 3 sudah ada pada anak pertama (o_1), individu-individu ini dihapus sehingga urutan ini menjadi:

Urutan Penyilangan Kedua: 9 – 2 – 6 – 7 – 8

Urutan penyilangan kedua ini akan ditempatkan pada anak pertama dan dimulai dari titik persilangan kedua. Proses ini juga berlaku untuk anak kedua, sehingga anak pertama (o_1) dan anak kedua (o_2) menjadi:

o1	=	2	6	7	8	4	5	1	3	9
o2	=	9	2	1	3	5	4	7	8	6

Hal penting yang harus diperhatikan pada teknik penyilangan ini adalah peluang yang ditetapkan pada prosedur inisialisasi. Semakin tinggi peluang penyilangan, maka rasio dari anak yang dihasilkan juga akan lebih bervariasi. Kemudian algoritma ini dilanjutkan dengan mutasi yang akan mengubah urutan salah satu gen dengan gen lainnya pada suatu individu untuk memperoleh nilai *fitness* yang tinggi dari suatu individu. Misalkan individu satu memiliki gen sebagai berikut:

i_1	=	6	9	7	2	4	5	1	3	8

Diambil dua bilangan acak yaitu $r_1 = 3$ dan $r_2 = 5$ dan kemudian gen ke-3 diganti dengan gen ke-5, atau mengganti gen ke-5 dengan gen ke-3. Individu satu ini akan bermutasi menjadi:

i_1	=	6	9	4	2	7	5	1	3	8

Proses seleksi, penyilangan, dan mutasi dari 80 individu ini akan digenerasi sebanyak 100 kali dengan menggunakan bahasa pemrograman Java 1.7, dengan Netbeans 7.1 sebagai media untuk me-*running* fungsipemrograman, di mana nilai *fitness* terbaik, terburuk, dan rata-rata dihitung setiap

generasi. Hasil dari pengolahan algoritma ini adalah urutan lokasi pengiriman yang akan digunakan perusahaan untuk mengirim barang. Lebih lanjut lagi, hasil dari algoritma ini dapat digambarkan pada Gambar 1 dan 2:

G	K1	K2	K3	K4	K5	K6	K7	K8	K9	+	-	X
1	4	6	3	7	2	5	9	8	1	0.0082	0.0056	0.0068
2	4	6	3	7	2	5	9	8	1	0.0082	0.0063	0.0069
3	4	6	3	7	2	5	9	8	1	0.0082	0.0063	0.0070
4	4	6	3	7	2	5	9	8	1	0.0082	0.0063	0.0071
5	4	6	3	7	2	5	9	8	1	0.0082	0.0065	0.0071
6	4	6	3	7	2	5	9	8	1	0.0082	0.0061	0.0069
7	4	6	3	7	2	5	9	8	1	0.0082	0.0061	0.0071
8	4	6	3	7	2	5	9	8	1	0.0082	0.0061	0.0070
9	4	6	3	7	2	5	9	8	1	0.0082	0.0057	0.0070
10	4	6	3	7	2	5	9	8	1	0.0082	0.0057	0.0073
11	4	6	3	7	2	5	9	8	1	0.0082	0.0063	0.0075
12	4	6	3	7	2	5	9	8	1	0.0082	0.0064	0.0075
13	4	6	3	7	2	5	9	8	1	0.0082	0.0064	0.0077
14	4	6	3	7	2	5	9	8	1	0.0082	0.0062	0.0076
15	4	6	3	7	2	5	9	8	1	0.0082	0.0062	0.0075
16	4	6	3	7	2	5	9	8	1	0.0082	0.0062	0.0075
17	4	6	3	7	2	5	9	8	1	0.0082	0.0064	0.0075
18	4	6	3	7	2	5	9	8	1	0.0082	0.0067	0.0079
19	4	6	3	7	2	5	9	8	1	0.0082	0.0066	0.0079
20	4	6	3	7	2	5	9	8	1	0.0082	0.0064	0.0079

Gambar 1. Hasil pengolahanalgoritma genetika pada 9 kota.

80	4	6	3	7	2	5	9	8	1	0.0082	0.0073	0.0080
81	4	6	3	7	2	5	9	8	1	0.0082	0.0073	0.0080
82	4	6	3	7	2	5	9	8	1	0.0082	0.0071	0.0079
83	4	6	3	7	2	5	9	8	1	0.0082	0.0073	0.0080
84	4	6	3	7	2	5	9	8	1	0.0082	0.0073	0.0080
85	4	6	3	7	2	5	9	8	1	0.0082	0.0073	0.0080
86	5	6	3	7	2	4	9	8	1	0.0083	0.0073	0.0081
87	5	6	3	7	2	4	9	8	1	0.0083	0.0074	0.0082
88	5	6	3	7	2	4	9	8	1	0.0083	0.0074	0.0082
89	4	6	3	7	2	5	9	8	1	0.0082	0.0064	0.0081
90	4	6	3	7	2	5	9	8	1	0.0082	0.0064	0.0081
91	4	6	3	7	2	5	9	8	1	0.0082	0.0070	0.0081
92	4	6	3	7	2	5	9	8	1	0.0082	0.0070	0.0080
93	4	6	3	7	2	5	9	8	1	0.0082	0.0071	0.0081
94	4	6	3	7	2	5	9	8	1	0.0082	0.0064	0.0080
95	2	6	3	7	4	5	9	8	1	0.0086	0.0064	0.0078
96	2	6	3	7	4	5	9	8	1	0.0086	0.0064	0.0074
97	2	6	3	7	4	5	9	8	1	0.0086	0.0064	0.0073
98	2	6	3	7	4	5	9	8	1	0.0086	0.0061	0.0073
99	2	6	3	7	4	5	9	8	1	0.0086	0.0061	0.0075
100	2	6	3	7	4	5	9	8	1	0.0086	0.0061	0.0073

Gambar 2. Hasil pengolahan algoritma genetika pada 9 kota (lanjutan).

PENUTUP

Setelah melakukan perancangan aplikasi dengan Algoritma Genetika, diperoleh satu intelegensi buatan yang mampu memvisualisasikan rangkaian jarak antar kota domain dan destinasi di mana batasan kota adalah sembilan kota. Hasil dari aplikasi merupakan sebuah prioritas tempuh antar

kota yang dimulai dari yang menjadi prioritas tertinggi hingga terendah dan menggunakan data jarak antar kota sebagai fundamental data yang harus diketahui dan pemicu dari skala prioritas. Hasil penelitian ini berupa urutan lokasi kota yang akan ditempuh oleh penjual sebagaimana telah digambarkan pada Gambar 2. Perjalanan dimulai dari kota dua sampai ke kota satu dan balik lagi ke kota dua. Aplikasi dalam penelitian ini dapat digunakan oleh baik individu maupun organisasi yang memiliki permasalahan dengan prioritas jarak tempuh, yang umumnya dibutuhkan oleh divisi penjualan dalam menentukan urutan lokasi pengiriman barang.

DAFTAR PUSTAKA

Kusumadewi. S, Purnomo. H. (2005). *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik.* Yogyakarta: Graha Ilmu.

Kylie Bryant, A. B. (2000). *Genetic Algorithms and the Traveling Salesman Problem.* Claremont: Harvey Mudd College.

Larranaga, P. et. al. (1999). *Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators.* Netherlands: Kluwer Academic Publishers.