

APLIKASI KOHONEN *SELF ORGANIZING* PADA *TRAVELLING SALESMAN PROBLEM* (TSP) DENGAN PROGRAM MATLAB

Nurul Imamah Ah.¹⁾

¹⁾Prodi Pendidikan Matematika, FKIP, Universitas Muhammadiyah Jember

E-mail: ¹⁾nurul.imamah86@gmail.com

Abstract

Travelling salesman problem is a problem that founded with sales that must go to many city and rest in a other city until back to the first city. Travelling salesman problem need an optimal ways with minimum scale, untill the process didnt need much money and others. The process to get optimal ways is too difficult, so this article will give the sollution to get an optimal ways with kohonen self organizing and using MATLAB to get an easy program of kohonen self organizing. From the discussion about travelling salesman problem using kohonen self organizing with 10 city and the space, we get minimum space that needed to walk in all the city is 2.6907 meter.

Keywords: Travelling Salesman Problem, Kohonen Self Organizing, Matlab

1. PENDAHULUAN

Travelling Salesman Problem (TSP) dikenal sebagai salah satu permasalahan optimasi klasik yang berat untuk dipecahkan secara konvensional. Penyelesaian eksak terhadap persoalan ini akan melibatkan algoritma yang mengharuskan untuk mencari kemungkinan semua solusi yang ada. Sebagai akibatnya, kompleksitas waktu dari eksekusi algoritma ini akan menjadi eksponensial terhadap ukuran dari masukan yang diberikan. *Travelling Salesman Problem* melibatkan seorang *travelling salesman* yang harus melakukan kunjungan ke sejumlah kota dalam menjajakan produknya.

Rangkaian kota-kota yang dikunjungi harus membentuk suatu jalur sedemikian sehingga kota-kota tersebut hanya boleh dilewati tepat satu kali dan kemudian kembali lagi ke kota awal. Penyelesaian terhadap permasalahan TSP ini adalah untuk memperoleh jalur terpendek. Penyelesaian eksak terhadap masalah TSP mengharuskan untuk melakukan perhitungan terhadap semua kemungkinan rute.

Adapun perumusan masalah dalam artikel ini adalah:

1. Bagaimana menyelesaikan masalah TSP dengan menggunakan algoritma kohonen *self organizing*.
2. Bagaimana cara menyelesaikan masalah TSP menggunakan algoritma kohonen *self organizing* dengan bantuan program MATLAB.

Sedangkan tujuan penulisan artikel yaitu:

1. Untuk mengetahui cara penyelesaian masalah TSP dengan menggunakan algoritma kohonen *self organizing*.
2. Untuk mengetahui cara menyelesaikan masalah TSP menggunakan algoritma kohonen *self organizing* dengan program MATLAB.

Ruang Lingkup Penelitian terbatas pada permasalahan TSP (*Travelling Salesman Problem*) Sebagai bahan acuan dalam menyelesaikan kasus *travelling salesman problem* pada cabang jaringan syaraf tiruan.

1.1 Travelling Salesman Problem

Travelling Salesman Problem (TSP) adalah suatu masalah yang ditemukan oleh pedagang yang harus bepergian dan singgah di beberapa kota hingga kembali ke kota

semula. Dengan kata lain TSP didefinisikan sebagai permasalahan dalam mencari jalur terpendek dengan melakukan tour tertutup (yang dimulai dari suatu kota dan kembali tersebut). Sedangkan Menurut Moon (2002) menyebutkan bahwa permasalahan TSP adalah untuk menemukan jalur terpendek dengan suatu jalur hanya dapat ditempuh satu kali dan kembali pada jalur awal keberangkatan. Dalam kehidupan sehari-hari, kasus TSP ini dapat diaplikasikan untuk menyelesaikan kasus lain, yaitu:

1. Pak Pos mengambil surat di kotak pos yang tersebar pada n buah lokasi di berbagai sudut kota.
2. Lengan robot mengencangkan n buah mur pada beberapa buah peralatan mesin dalam sebuah jalur perakitan.
3. Produksi n komoditi berbeda dalam sebuah siklus.

1.2 Jaringan Saraf Kohonen *Self Organizing*

Jaringan Saraf Kohonen *Self Organizing* merupakan salah satu cabang pembelajaran jaringan syaraf tiruan. Jaringan Syaraf tiruan didefinisikan sebagai susunan dari elemen-elemen penghitung yang disebut neuron atau titik (node) yang saling terhubung guna dimodelkan untuk meniru fungsi otak manusia. (PHK TIK, 2008). Sedangkan menurut Fausett (2004) Jaringan Syaraf Tiruan (*artificial neural network*) adalah pemrosesan sistem informasi pada karakteristik tertentu dalam keadaan yang berhubungan dengan jaringan syaraf biologi.

Menurut Puspitorini (2008) Jaringan Saraf Kohonen *Self Organizing* termasuk dalam pembelajaran tak terawasi (*unsupervised learning*). Pertama kali diperkenalkan oleh Prof. Teuvo Kohonen pada tahun 1982. Pada jaringan ini, suatu lapisan yang berisi neuron-neuron akan menyusun dirinya sendiri berdasarkan input nilai tertentu dalam suatu kelompok yang dikenal dengan istilah *cluster*. Selama proses penyusunan diri, *cluster* yang memiliki vektor bobot paling cocok dengan pola input (memiliki jarak yang paling dekat) akan terpilih sebagai pemenang. Neuron

yang menjadi pemenang beserta neuron-neuron tetangganya akan memperbaiki bobot-bobotnya. Tujuan pembelajaran dari algoritma ini adalah untuk mentransformasikan suatu pola sinyal masukan yang mempunyai dimensi yang berubah-ubah ke suatu peta yang berdimensi satu atau dua. Sifat pemetaan yang dimiliki Jaringan Saraf Kohonen *Self Organizing* meniru pada otak manusia yang tidak terdapat pada Jaringan Saraf Tiruan lainnya.

1.3 Algoritma *Travelling Salesman Problem* dengan Kohonen *Self Organizing*

Untuk selengkapnya, algoritma TSP dapat dijabarkan (Kusumadewi, 2003), sebagai berikut:

1. Tetapkan parameter-parameter:
 - a. Maksimum epoch (MaxEpoch).
 - b. *Learning rate* untuk perubahan bobot antar neuron (θ)
 - c. *Learning rate* untuk perubahan bobot antara koordinat kota dengan neuron (ϕ). Nilai θ dan ϕ bisa dibuat sama.
 - d. Pengurangan learning rate (momentum).
 - e. Faktor pengali untuk menuju koordinat kota (*near*).
2. Masukkan koordinat kota (X_i, Y_i), dengan $i = 1, 2, \dots, N$.
3. Cari jarak antar setiap kota ke- i dengan kota ke- j , D_{ij} , dengan $i, j = 1, 2, \dots, N$.
4. Tetapkan:
 - a. Jumlah neuron (Q), dengan $Q \geq N$.
 - b. Tetapkan koordinat awal setiap neuron (nX_i, nY_i), sedemikian hingga neuron-neuron membentuk lingkaran.

$$\text{Misal: } nX_i = 0.5 \cos(\alpha_i) \quad (8)$$

$$nY_i = 0.5 \sin(\alpha_i) \quad (9)$$

dengan $i=1, 2, \dots, Q$; $\alpha = 0$, dan $\alpha_i = \alpha_{i-1} + 2\pi/Q$, untuk $i > 1$

5. Tetapkan bobot awal antara koordinat kota (x, y) dengan setiap neuron, sebut sebagai wX_i dan wY_i secara acak, antara 0 sampai 1; dengan $i=1, 2, \dots, Q$.
6. Tetapkan bobot awal antar neuron, sebut

sebagai r_{ij} , dengan $i, j = 1, 2, \dots, Q$, dengan cara:

- a. Cari jarak setiap antar neuron, , dengan $i, j = 1, 2, \dots, Q$.

- b.
$$r_{ij} = e^{-\left[\frac{nD_{ij}^2}{2\theta^2}\right]}$$

7. Set epoch = 0

8. Kerjakan selama epoch < MaxEpoch:

- a. epoch = epoch+1
- b. Pilih sembarang kota secara random, misal kota terpilih adalah kota ke-idx.
- c. Set koordinat lokasi yang akan didekati (tX,tY):
 - i. Bangkitkan bilangan satu random r antara 0 sampai 1.
 - ii. $tX = X_{idx} + r * \text{Near} - \text{Near}/2 \dots\dots(2.10)$
 - iii. $tY = Y_{idx} + r * \text{Near} - \text{Near}/2 \dots\dots(2.11)$
- d. Cari jarak minimum antara (tX,tY) dengan bobot antara koordinat kota dan neuron (wX_i, wY_i), $i=1, 2, \dots, Q$. Misalkan jarak minimumnya jatuh pada jarak antara (tX,tY) dengan bobot ke-j (wX_j, wY_j).
- e. Perbaiki bobot antara koordinat kota dan setiap neuron(wX_i, wY_i), $i=1, 2, \dots, Q$:
 - i. $wX_i = wX_i + \varphi * r_{ij} (tX - wX_i) \dots\dots(11)$
 - ii. $wY_i = wY_i + \varphi * r_{ij} (tY - wY_i) \dots\dots(12)$
 dengan j adalah indeks terpilih pada (d).
- f. Perbaiki nilai θ dan φ :
 - i. $\theta = \theta * \text{momentum} \dots\dots(13)$
 - ii. $\varphi = \varphi * \text{momentum} \dots\dots(14)$
- g. Perbaiki bobot antar neuron (r_{ij}), dengan $i=1, 2, \dots, Q$; dan j adalah indeks terpilih pada (d):

$$r_{ij} = e^{-\left[\frac{nD_{ij}^2}{2\theta^2}\right]}$$

9. Cari jarak minimum antara setiap kota ke-i dengan setiap bobot koordinat kota ke neuron ke-j, sebut sebagai mJ_i , dan indeksnya sebut sebagai L_i , dengan $i=1, 2, \dots, N$:

- a.
$$mJ_i = \sqrt{(X_i - wX_j)^2 + (Y_i - wY_j)^2}$$
 ;
dengan $j=1, 2, \dots, Q \dots\dots(16)$

b. $L_i = j$, sedemikian hingga

$$\sqrt{(X_i - wX_j)^2 + (Y_i - wY_j)^2} \dots\dots(17)$$

c. Bentuk matriks P berukuran $N \times Q$ dengan kolom pertama adalah mJ_i dan kolom kedua berisi L_i .

10. Urutkan naik matriks P, berdasarkan kolom pertama.

11. Jalur terpendek adalah matriks P terurut pada kolom kedua. Cari panjang jalur terpendek.

2. METODE

Adapun metode yang dilakukan dalam penulisan artikel ini adalah metode penelitian kepustakaan (*library research*) yang dilakukan dengan beberapa tahap sebagai berikut:

- a. Studi literatur
Pada tahap ini penulis mempelajari tentang algoritma kohonen self organizing dan cara kerja TSP.
- b. Pengumpulan data
Pada tahap ini penulis mengambil contoh kasus pada literatur.
- c. Aplikasi kasus pada program
Pada tahap ini penulis mengaplikasikan penyelesaian kasus TSP pada program MATLAB.
- d. Analisis hasil
Setelah mengaplikasikan penyelesaian kasus TSP pada program MATLAB maka penulis menganalisis hasil.
- e. Kesimpulan
Tahap terakhir dari penelitian ini adalah pengambilan kesimpulan berdasarkan hasil running program pada tahap 4.

3. HASIL DAN PEMBAHASAN

Jaringan kohonen *self organizing* dapat digunakan untuk penyelesaian kasus *traveling salesman Problem* (TSP). TSP berikut akan dipergunakan untuk mencari jalur terpendek dimana kata tujuan yang ditempuh sama dengan kota asal (sirkular). Pada penelitian ini diambil salah satu contoh kasus “Diberikan

sepuluh kota dan jarak antar kota disajikan dalam bentuk koordinat. Tentukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal keberangkatan”. TSP pada 10 kota dengan koordinat jalur kota disajikan pada tabel 1 berikut:

Tabel 1. Koordinat Jalur Kota

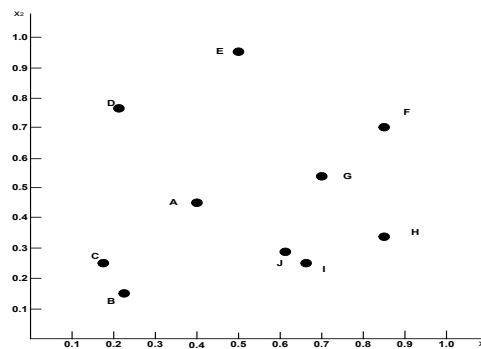
	x_1	x_2
A	0,4000	0,4439
B	0,2439	0,1463
C	0,1707	0,2293
D	0,2293	0,7610
E	0,5171	0,9414
F	0,8732	0,6536
G	0,6878	0,5219
H	0,8488	0,3609
I	0,6683	0,2536
J	0,6195	0,2634

Jarak antar kota diberikan dalam matriks dengan jarak simetri berikut :

Tabel 2. Koordinat Jarak Antar Kota

	Posisi									
	A	B	C	D	E	F	G	H	I	J
A	.0000	.3361	.3141	.3601	.5111	.5176	.2982	.4564	.3289	.2842
B	.3361	.0000	.1107	.6149	.8407	.8083	.5815	.6418	.4378	.3934
C	.3141	.1107	.0000	.5349	.7919	.8207	.5941	.6908	.4982	.4501
D	.3601	.6149	.5349	.0000	.3397	.6528	.5171	.7375	.6710	.6323
E	.5111	.8407	.7919	.3397	.0000	.4579	.4529	.6686	.7042	.6857
F	.5176	.8083	.8207	.6528	.4579	.0000	.2274	.2937	.4494	.4654
G	.2982	.5815	.5941	.5171	.4529	.2274	.0000	.2277	.2690	.2674
H	.4564	.6418	.6908	.7375	.6686	.2937	.2277	.0000	.2100	.2492
I	.3289	.4378	.4982	.6710	.7042	.4494	.2690	.2100	.0000	.0498
J	.2842	.3934	.4501	.6323	.6857	.4654	.2674	.2492	.0498	.0000

Koordinat jarak pada tiap kota diberikan pada grafik berikut:



Gambar 1. Titik Koordinat Jarak Kota pada TSP

3.1 Aplikasi Kohonen Self Organizing Pada Kasus TSP dengan Program MATLAB

Sesuai dengan algoritma kohonen self organizing maka didapatkan Output Program sebagai berikut:

Neuron = 1 2 3 4 5 6 6 8 9 10

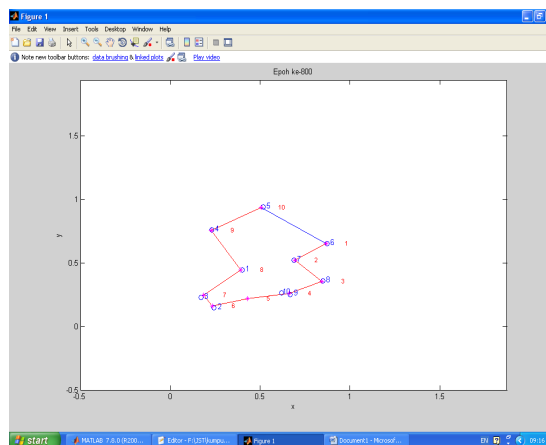
Jalur = 4 5 6 7 8 9 10 1 3 2

Panjang Jalur = 2.8362

Neuron = 1 1 3 5 6 8 9 9 10 10

Jalur = 2 3 1 4 5 6 7 8 9 10

Panjang Jalur = 2.6907



Gambar 2. TSP dengan kohonen self organizing

Berdasarkan hasil uji coba kasus TSP dengan kohonen self organizing menggunakan program MATLAB terlihat bahwa pada epoch ke 800 didapatkan jalur terpendek dari 10 kota dengan koordinat yang disajikan pada gambar 2.

Adapun urutan rute jalur optimal perjalanan yang diperoleh yaitu 2 3 1 4 5 6 7 8 9 10. Angka angka tersebut menunjukkan urutan rute yang dapat ditempuh dengan jarak tempuh minimal 2.6907 adalah rute kunjungan B C A D E F G H I J. Hasil tersebut dapat disajikan dalam tabel 3 berikut:

Tabel 3. Rute TSP dengan Kohonen *Self Organizing*

Rute/Koordinat	x1	x2
B	0,2439	0,1463
C	0,1707	0,2293
A	0,4000	0,4439
D	0,2293	0,7610
E	0,5171	0,9414
F	0,8732	0,6536
G	0,6878	0,5219
H	0,8488	0,3609
I	0,6683	0,2536
J	0,6195	0,2634

4. KESIMPULAN

Penyelesaian masalah *travelling salesman problem* menggunakan metode jaringan saraf kohonen untuk mendapatkan rute perjalanan terpendek sangat dipengaruhi oleh parameter pelatihan seperti *phi*, *theta*, *momentum* dan *near*, dan keadaan lainnya. Sehingga jika proses pelatihan dilakukan beberapa kali dengan data koordinat kota yang sama akan memungkinkan mendapatkan jalur dan panjang jalur perjalanan yang berbeda. Adapun rute terpendek yang dapat ditempuh dengan jarak tempuh minimal 2.6907 adalah rute kunjungan B C A D E F G H I J.

DAFTAR PUSTAKA

- Arhani, M. 2005. *Pemrograman MATLAB*. Yogyakarta: Perpustakaan Nasional.
- Chiung, M. 2002. *An Efficient genetic Alghorithm for Travelling salesman Problem with Precedence Constraints*. Korea: Department of Industrial Engineering.
- Fausett, L. 2004. *Dasar-Dasar Jaringan Syaraf Tiruan*. Tanpa Penerbit
- Puspitorini, S. 2008. *Penyelesaian Masalah Travelling Salesman Problem dengan*

Jaringan Saraf Self Organizing. Jambi: STMK Nurdin Hamzah.

PHK TIK. 2008. *Jaringan Syaraf Tiruan*. Malang: Universitas Widyagama.

Refianti, R. Tanpa Tahun. *Solusi optimal Travelling Salesman Problem dengan Ant colony System (ACS)*. Jurusan teknik Informatika: Universitas Gunadarma.