

# Analisis Integrasi Elastic Net pada Self-Organizing Map untuk penyelesaian Travelling Salesman Problem

Muhammad Sulkifly Said<sup>1</sup>, Indah Soesanti<sup>2</sup>, Sri Suning Kusumawardani<sup>3</sup>  
Departement Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada  
Jl. Grafika No.2 Yogyakarta, Indonesia  
Corresponding Author: sulkifly.ti14@mail.ugm.ac.id<sup>1</sup>, indah@mti.ugm.ac.id<sup>2</sup>,  
suning@ieee.org<sup>3</sup>

## Abstrak

Dalam persoalan TSP Ketika jumlah kota semakin besar maka algoritme deterministik sudah tidak efektif lagi. Untuk persoalan dengan ruang pencarian yang sangat besar diusulkan metode berbasis probabilistik salah satunya adalah ANN (*Artificial Neural Network*), ANN terinspirasi oleh fenomena biologi dan alam, salah satunya adalah SOM (*Self-Organizing Maps*) atau bisa juga dikenal sebagai *topological preserving map*, Kohonen Feature Map berbasiskan pada *unsupervised learning* atau pembelajaran tak terawasi. Algoritma SOM dalam penelitian dimodifikasi untuk mempercepat proses konvergensi saat proses adaptasi dan *learning*. Dengan kata lain, disetiap iterasi terjadi proses pembaruan guna mendapatkan *neuron* dengan bobot paling dekat yang dapat merepresentasikan *vector*, permasalahan TSP yang diselesaikan dalam penelitian ini adalah TSP Cycle.

## Abstract

In TSP, when the number of cities grows larger, determining algorithms is no longer effective. To deal with big search spaces, it is suggested to apply probability base methods, one of which is ANN (*Artificial Neural Network*). The emergence of ANN was first inspired by one of biological and natural phenomena which was SOM (*Self-Organizing Maps*) or *topological preserving map*. Kohonen Feature Map is based on *unsupervised learning*. To be able to be applied in research, SOM algorithm is modified to accelerate convergence processes during adapting and learning processes. In other words, in each iteration process, regeneration processes to obtain neurons with the most intended mass. TSP problem that can be solved in this research is TSP Cycle.

**Keywords:** TSP; Travelling Salesman Problem; Combinatorial Optimization; Self-Organizing Map.

## PENDAHULUAN

*Travelling salesman problem* (TSP) merupakan salah satu cabang dari permasalahan optimasi, yaitu permasalahan optimasi bernilai diskrit. Definisi dari optimasi adalah suatu tindakan untuk mencapai hasil terbaik dari suatu keadaan tertentu yang mencari nilai *minimum* atau *maximum*. Optimasi dapat dijumpai di banyak bidang antara lain *design*, *construction*, *maintenance* dan lain-lain (J. Y. Å, Yang, Zhang, & Tang, 2009). Dalam kasus TSP tidak ada satupun metode secara general dapat memecahkan masalah secara efisien, sehingga diperlukan suatu algoritma yang dapat mengurangi pengecekan terhadap semua kombinasi yang ada agar pencarian solusi tidak meningkat secara eksponensial. Karenanya sejumlah metode telah diusulkan untuk persoalan tersebut dan salah satunya adalah dengan pendekatan *neural network* (P. H. S. Å, Teresinha, & Steiner, 2007). *Neural Network* merupakan representasi dari suatu sistem yang mengambil inspirasi dari riset atas perilaku saraf manusia secara biologis (Networks, Now, Are, & Networks, n.d.). Terdapat beberapa macam kelebihan dalam *neural network* diantaranya *error-tolerance*, *self-organizing*,

How to cite:

Said, M.S; Soesanti, I; Kusumawardani, S.S. (2016). Analisis Integrasi Elastic Net pada Self-Organizing Map untuk penyelesaian Travelling Salesman Problem. *Prosiding SNMIPA 2016 UNNES dna UIN Walisongo Semarang*, 1(1), 78-86.

*adaptive learning, parallel information processing dan real time operation* (Paper, Introduction, Models, & Maps, 1990). Inilah yang menyebabkan Neural Network dapat digunakan untuk menyelesaikan berbagai permasalahan misalkan *handwriting recognition, computer vision, speech recognition, medical* dan lain-lain (Kohonen, Barna, Chrisley, & Science, n.d.).

Sifat *self-organizing* atau yang biasa juga dikenal dengan *topological preserving map* menggunakan metode pembelajaran *unsupervised learning* atau pembelajaran tidak terawasi

Tabel 2.1 pertumbuhan jumlah rute dan waktu berdasarkan jumlah Kota

no	Jumlah Rute	
5	$(5-1)!/2$	12
6	$(6-1)!/2$	60
7	$(7-1)!/2$	360
8	$(8-1)!/2$	2520
9	$(9-1)!/2$	20160
10	$(10-1)!/2$	181440
11	$(11-1)!/2$	1814400
12	$(12-1)!/2$	19958400
15	$(15-1)!/2$	43589145600
20	$(20-1)!/2$	$6.08225502 \times 10^{16}$
25	$(25-1)!/2$	$3.102242009 \times 10^{23}$

(Paper et al., 1990).. inilah yang menjadi sumber inspirasi dalam menyelesaikan permasalahan optimasi diskrit TSP dalam ruang pencarian yang besar, sekumpulan neuron akan melakukan proses *adaptation dan learning* secara sedirinya bertahap mengikuti property pola input data, sehingga membentuk suatu pemetaan yang merepresentasikan suatu solusi rute terpendek dengan total biaya yang mendekati sub optimum dengan waktu yang relatif cepat. Permasalahan *Traveling Salesman Problem (TSP)* menarik perhatian para ilmuwan dan matematikawan khususnya pada bidang *Computer Science dan Engineering*, TSP sangat mudah dideskripsikan tetapi sulit untuk dipecahkan. masalah dapat dinyatakan sebagai seorang salesman bepergian dan mengunjungi tepat sekali setiap daftar kota (Applegate, Bixby, Chvátal, & Cook, n.d.), fungsi tujuan yang ingin dicapai TSP adalah meminimalkan jarak atau biaya dari suatu set kota dari perjalanan seorang salesman. TSP masuk ke dalam kategori *NP-hard*. *NP-hard* adalah permasalahan yang memiliki solusi non polynomial dan tidak ada solusi polinomialnya. Sedangkan dalam persoalan keputusan TSP termasuk dalam kategori *NP-Complete*, yaitu permasalahan yang mungkin ditemukan solusi polinomialnya (Applegate et al., n.d.). Secara matematis TSP diformulasikan sebagai:

$$\min f(x) = \sum_{i=1}^{n-1} d(\pi(i), C\pi(i+1)) + d(\pi(i), C\pi(1))$$

$\pi$  adalah representasi dari permutasi  $d(\pi(i), \pi(i+1))$  adalah jarak kota  $i$  ke kota  $i+1$  dan  $d(\pi(n), \pi(1))$  adalah jarak dari kota  $n$  ke kota  $1$ , TSP menggunakan graph tertutup sehingga salesman melakukan perjalanan ke semua kota dan kembali ke kota pertama). Terdapat dua jenis TSP, yaitu TSP simetris dan TSP asimetris. TSP Simetris, jarak tempuh dari kota X ke kota Y sama dengan jarak tempuh dari kota Y ke kota X. sedangkan pada TSP Asimetris jarak antara kota X ke kota Y tidak sama dengan jarak kota Y ke kota X. Secara umum terdapat dua metode dalam menyelesaikan TSP yaitu *exact method* dan *heuristic method*. Menggunakan *exact method* untuk 5 kota jumlah rute yang dapat ditempuh sebanyak 12 rute kemungkinan, namun permasalahan mulai muncul jika kota yang akan di uji berjumlah banyak, misalkan untuk 10 kota maka jumlah rute yang didapatkan akan

sebanyak 181.440 rute kemungkinan, dengan begitu seiring dengan bertambahnya jumlah kota maka kandidat penyelesaian akan meningkat secara eksponensial. Pada table 2.1 menjelaskan jumlah rute dan waktu yang di dapatkan sesuai dengan jumlah kota yang di uji

Berdasarkan uraian table 2.1 menjelaskan dengan menggunakan *super computer* sekalipun masih sulit menyelesaikan persoalan TSP dengan jumlah kota yang sangat banyak, sehingga solusi sub optimal masih dapat diterima untuk menghindari tingkat kompleksitas pemrosesan yang lama.

## METODE

*Self-Organizing Map* (SOM) diperkenalkan oleh seorang ilmuwan finlandia pada tahun 1982 bernama teuvo kohonen (Paper et al., 1990). SOM menggunakan metode pembelajaran tidak terbimbing (*unsupervised*), dengan pendekatan pengelompokan (*clustering*) yang tidak menggunakan kelas untuk mewakili setiap data, clustering dalam optimasi kombinatorial tidak membutuhkan data test (self test), Data keluaran merupakan sebuah simpul yang saling terkoneksi antara satu kota dengan kota lainnya. SOM dapat mereduksi data berdimensi tinggi ke dimensi rendah dan memvisualisasikannya ke bentuk topografi layaknya peta dua dimensi, sehingga memudahkan dalam pengamatan distribusi hasil pengelompokan tanpa merubah data asli. SOM menggunakan pelatihan ANN yang berbasiskan *winner takes all*, konsep winner takes all adalah hanya *neuron* pemenang yang akan diperbaharui bobotnya. SOM menggunakan NN layer tunggal (*single layer*) dengan jumlah unit *neuron* sama K kelompok. data memiliki hubungan dengan bobot *neuron* pemroses. Berbeda dengan *perceptron*, perhitungan keluaran sinyal dari data masukan dengan bobot pada setiap *neuron* dalam tidak menggunakan *inner-product* melainkan *Euclidian Distance* (square Euclidian) dan keluaran sinyal tidak perlu diaktiasi (fungsi aktivasi linear) karena fungsi aktivasi tidak mempengaruhi pada pemilihan *neuron* pemenang yang akan memperbaharui bobotnya. Jumlah kelompok K (*neuron*) didefinisikan sendiri oleh pengguna, dimana setiap Kelompok mewakili sebuah data hasil pemrosesan oleh indeks *neuron* dalam layernya. Untuk lebih jelasnya Gambar 3.1 mendeskripsikan arsitektur Self Organizing Map. Secara default Algoritma Self-Organizing Map dapat dijelaskan secara detail pada prosedur berikut:

Algoritma Self-Organizing Map	
1.	Inisialisasi bobot $W_{ij}$ , Tentukan parameter topologi ketetanggaan, Tentukan parameter laju pembelajaran, Tentukan jumlah maksimal iterasi pelatihan
2.	Selama jumlah maksimal iterasi belum tercapai lakukan langkah 3 – 7
3.	Untuk setiap data masukan X, lakukan langkah 4 – 6
4.	Untuk setiap neuron j, hitung $D_j = \sum_i (w_{ij} - x_i)^2$ , $i = 1, \dots, N$ , N adalah dimensi data (N)
5.	Cari indeks dari sejumlah neuron, yaitu $D_{ij}$ yang mempunyai nilai terkecil
6.	Untuk neuron j dan semua neuron yang menjadi tetangga j (yang sudah didefinisikan) dalam radius R dilakukan pembaruan bobot: $w_{ij}(\text{baru}) = w_{ij}(\text{lama}) + \eta(x_i - w_{ij}(\text{lama}))$ .
7.	Perbarui nilai laju pembelajaran

Algoritme yang diusulkan merupakan pengembangan Model optimasi yang didasarkan pada artikel "*an analogue approach to the travelling salesman problem using an elastic net method*" oleh Richard Durbin dan David Willshaw yang diterbitkan pada jurnal "Nature already back" pada tahun 1987 (Chen, 2007). Banyak solusi pendekatan untuk permasalahan TSP, diantaranya memulai dengan satu solusi *feasible* dan mencoba untuk mencari solusi yang lebih baik dari setiap proses iterasinya yang dikenal dengan metode *heuristic*.

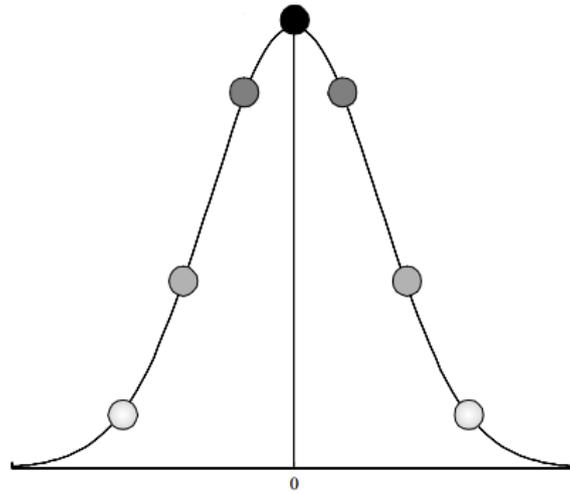
Pendekatan dari Durbin dan Willshaw berbeda, dalam algoritme ini tidak mulai dengan *feasible solution* melainkan dengan *Elastic Net*, yaitu lingkaran node di mana jumlah node merupakan kelipatan dari jumlah kota TSP. Lingkaran ini secara bertahap menghampiri hingga menempati seluruh kota dan mendapatkan rute terpendek untuk *salesman*. Untuk selanjutnya *Elastic Net* dapat diistilahkan sebagai *Adaptive Ring*. Terdapat beberapa parameter yang digunakan dalam algoritma ini antara lain *Epoch* (jumlah maximum epoch), *Neuron* (jumlah neuron yang digunakan), *Near* (faktor pengali untuk menuju koordinat kota), *Momentum* (pengurangan *learning rate* variabel theta dan phi (*learning reduction*)), *Theta* (*learning rate* bobot input dan kota), *Phi* (*learning rate* antar bobot neuron) (Aras & Oommen, 1999). Neuron dalam *Adaptive Ring* bersifat dinamis yang jumlahnya berubah-ubah pada setiap iterasinya, tujuannya agar mempercepat proses konvergensi dari seluruh kota yang terpilih dapat didekati oleh simpul neuron secara acak, sehingga nantinya akan dapat membentuk sebuah garis simpul penghubung sekaligus sebagai urutan kota yang disinggahi. Berikut basic cara kerja algoritma tersebut (Brocki, n.d.):

Procedure Train SOM
<p><b>Begin</b></p> <p>Randomize weight for all neurons</p> <p><b>for</b> (i=1 to iteration_numer) <b>do</b></p> <p>  <b>begin</b></p> <p>    take one random input pattern</p> <p>    find the winning neuron</p> <p>    find neighbors of the winner</p> <p>    modify synaptic weight of neurons</p> <p>    reduce learning rate and neighborhood radius</p> <p>  <b>end</b></p> <p><b>End</b></p>

Topologi ketetanggaan yang digunakan adalah fungsi ketetanggaan gaussian dalam proses bobot antar *neuron*, Topologi ketetanggaan gaussian maksimum berada pada titik  $D_{ij}=0$ , dalam proses training SOM, tinggi atau rendahnya kekuatan amplitudo ditandai dengan jarak antar kedekatan *neuron* terhadap koordinat kota. *winning neuron* adalah neuron dengan kekuatan amplitudo terbesar. fungsi tetangga Gaussian dapat dilihat pada Gambar 1.

Persamaan fungsi *Gaussian neighbourhood* digunakan sebagai fungsi ketetanggaan neuron,  $r_c$  menyatakan posisi diskrit neuron pemenang dan  $r_i$  adalah posisi diskrit neuron ke-i. penelitian oleh linde at al merekomendasikan nilai 0.2 sebagai nilai default untuk penurunan  $\alpha(t)$  secara linear (Brocki, n.d.) yang dapat dinyatakan dengan:

$$h_{ci}(t) = \exp\left(-\frac{(\|r_c - r_i\|^2)}{(0.2 \cdot \sigma^2(t))}\right)$$



Gambar 1. Fungsi Ketetanggaan Gaussian

Adapun Langkah-langkah Algoritme Elastic Net untuk sejumlah 5 kota adalah sebagai berikut

1. Pemilihan Node Kota secara Random

Pemilihan Node Kota secara Random digambarkan seperti pada Gambar 2. Inisialisasi awal adalah Pemilihan kota secara acak, icon berwarna biru adalah adaptive ring dan icon merah merupakan kota yang terpilih secara random

2. Cari node terdekat (“winning neuron”)

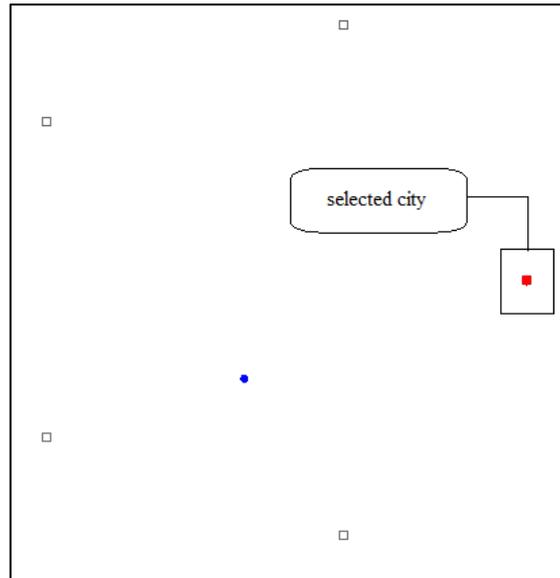
Pada langkah selanjutnya, algoritma melakukan proses looping ke semua neuron pada Adaptif Ring untuk mencari neuron terdekat, yaitu neuron dengan jarak terkecil dan bergerak mendekati kota yang terpilih secara acak pada langkah pertama kemudian melakukan update pada neuron tetangga. Parameter (*learning rate*) atau laju tingkat pembelajaran mendefinisikan seberapa jauh neuron akan dipindahkan. *Learning rate* akan menurun dari waktu ke waktu untuk mendapatkan fleksibilitas maksimum guna menstabilkan solusi pada tahap selanjutnya dari algoritma. Mencari node terdekat digambarkan seperti pada Gambar 3.

3. Adaptasi neuron pada intermediate stage

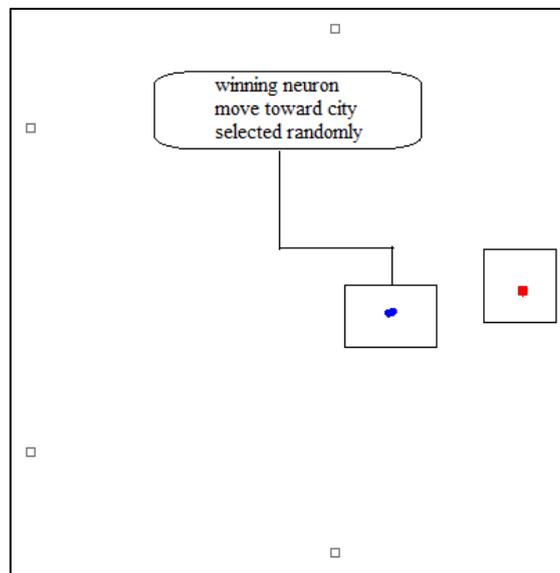
Pada iterasi ke-100 neuron yang mengalami pembaharuan fungsi ketetanggaan akan Semakin jauh dari neuron lainnya, aturan inilah yang membuat efek "rubber band" atau efek elastis pada adaptive ring. Adaptasi neuron pada intermediate stage digambarkan pada Gambar 4.

4. Adaptasi neuron pada final stage

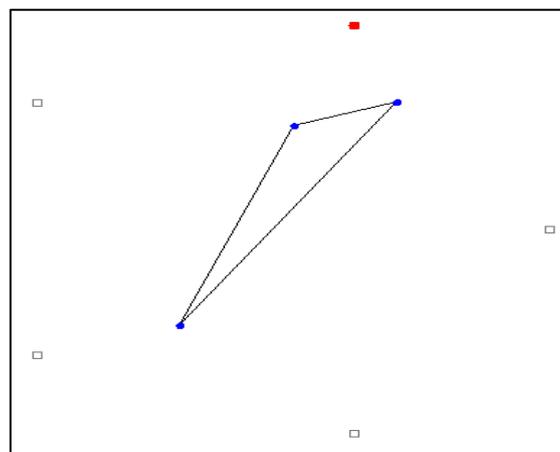
Pada final iterasi, neuron pada topologi elastic net telah berhasil mendekati seluruh kota dan membentuk sebuah simpul jalur yang merupakan cost yang mendekati sub optimal dari rute TSP Cycle. Data yang digunakan pada penelitian ini dibangkitkan secara acak dengan fungsi random Double pada java, Perhitungan luas lingkaran membutuhkan nilai fungsi class Math.PI, yaitu sebuah konstanta dengan nilai berkisar 3.14159 perhitungan jarak menggunakan Euclidian Distance. Adaptasi neuron pada final stage digambarkan pada Gambar 5.



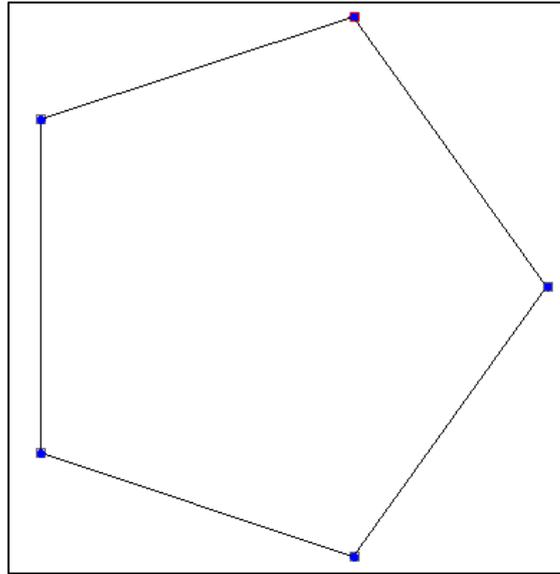
Gambar 2. Pemilihan Node Kota secara Random



Gambar 3. mencari node terdekat



Gambar 4. Adaptasi neuron pada intermediate stage



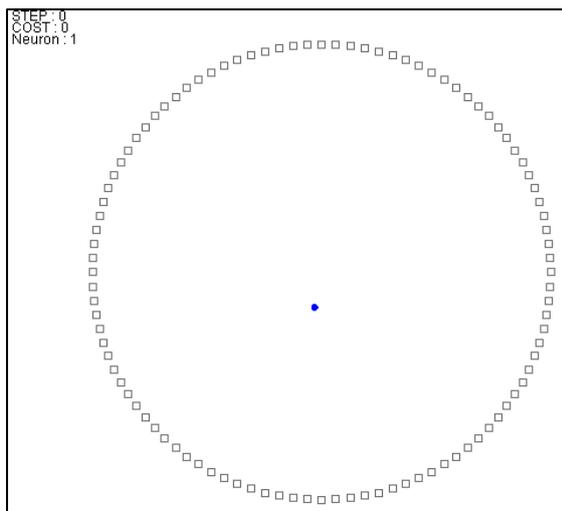
Gambar 5. Adaptasi neuron pada final stage

## HASIL DAN PEMBAHASAN

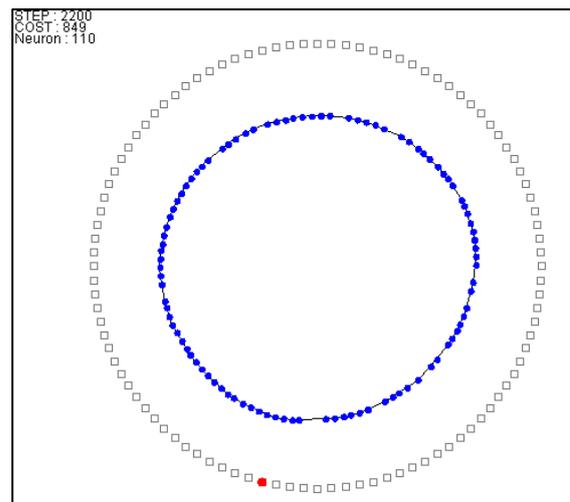
Hasil uji coba akhir adalah mengimplementasikan algoritme SOM terhadap sejumlah data yang dibangkitkan secara acak, hasil pembangkitan ini merupakan posisi koordinat kota yang terdiri dari koordinat x dan y. Pengujian dilakukan dengan 100 kota, dengan tiga gambar yang menggambarkan evolusi algoritma, mulai dari tahap awal dari cincin (Gambar 6), mencapai tahap peralihan setelah beberapa iterasi (Gambar 7) dan berhenti di tahap akhir akhir (Gambar 8).

Berdasarkan uraian gambar diatas, Pada final stage neuron telah fit terhadap kota dan membentuk simpul jalur terpendek pada TSP dengan total 1227 dan jumlah iterasi sebanyak 12300, neuron bersifat adaptif mengikuti jumlah kota yang telah di deklarasikan yaitu 100 kota.

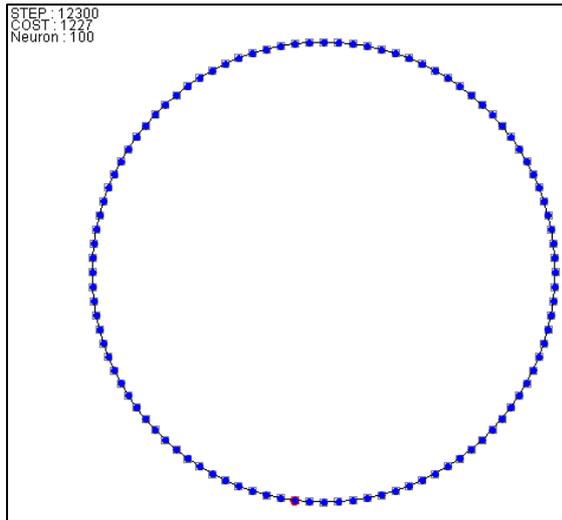
Berikut grafik konvergensi iterasi pada algoritma elastic net



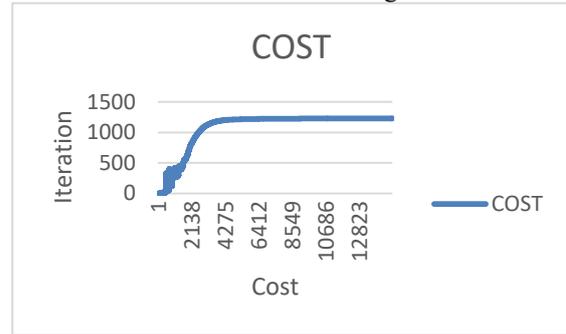
Gambar 6. Initial stage



Gambar 7. Intermediate stage



Gambar 8. Final stage



Gambar 9. Grafik Pembelajaran Elastic Net

## SIMPULAN

Berdasarkan hasil analisa dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, maka penulis dapat menarik kesimpulan sebagai berikut:

1. Elastic Net dapat digunakan dalam menyelesaikan persoalan travelling salesman problem.
2. Tingkat Keberhasilan Elastic Net dalam mendapatkan nilai sub optimal sangat bergantung pada penentuan nilai *learning rate*, *learning reduction* dan *Phi*

## DAFTAR PUSTAKA

- Ã, J. Y., Yang, G., Zhang, Z., & Tang, Z. (2009). *Neurocomputing An improved elastic net method for traveling salesman problem*, 72, 1329–1335. <http://doi.org/10.1016/j.neucom.2008.09.011>
- Ã, P. H. S., Teresinha, M., & Steiner, A. (2007). *A new approach to solve the traveling salesman problem*, 70, 1013–1021. <http://doi.org/10.1016/j.neucom.2006.03.013>
- Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (n.d.). No Title.
- Aras, N., & Oommen, B. J. (1999). *The Kohonen network incorporating explicit statistics and its application to the travelling salesman problem* 8, 12, 1273–1284.
- Brocki, L. (n.d.). *Kohonen Self-Organizing Map for the Traveling Salesperson Problem*, 2–8.
- Chen, J. (2007). *Solving the Traveling Salesman Problem Using Elastic Net Integrate with SOM*, 2234–2237.
- Kohonen, T., Barna, G., Chrisley, R., & Science, I. (n.d.). *Statistical Pattern Recognition with Neural Networks: Benchmarking Studies*, (x), 61–68.
- Networks, W. N., Now, W., Are, H., & Networks, N. (n.d.). No Title.
- Paper, I., Introduction, I., Models, N., & Maps, S. (1990). *The Self-organizing Map*, 78(9), 1464–1480.