

## PENGUJIAN DAN IMPLEMENTASI SELF-ORGANIZING MAP PADA STUDI KASUS PARIWISATA

### TESTING AND IMPLEMENTATION OF SELF-ORGANIZING MAP ON TOURISM CASE STUDY

David Winalda

Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom  
[davidwinalda94@gmail.com](mailto:davidwinalda94@gmail.com)

---

#### Abstrak

Salah satu wilayah di Indonesia yang perkembangan wisatanya cukup cepat adalah wilayah Bandung Raya. Wilayah Bandung Raya yang luas, banyaknya tempat wisata baru yang bermunculan, hingga wisatawan yang baru pertama kali datang, membuat wisatawan cukup kesulitan dalam mengetahui tempat wisata apa saja yang harus dikunjungi dan rute mana yang harus dilalui yang menyesuaikan waktu wisata dari wisatawan. Untuk itu maka aplikasi yang akan dirancang adalah sebuah sistem yang memberikan rute wisata paling minimal dan mengetahui susunan rencana perjalanan yang menyesuaikan waktu wisatawan. Pendekatan yang digunakan dalam pencarian rute wisata adalah model dari Jaringan Saraf Tiruan yaitu *Self-Organizing Maps* (SOM). Pengujian memberikan hasil bahwa parameter yang baik untuk melatih jaringan SOM dalam pencarian rute wisata yang paling minimal adalah *learning rate*  $\alpha=0.7$ , jumlah *neuron* = 8 per city/lokasi/tempat wisata,  $N$  (iterasi) = 800, dan  $r_0=23.2$ . Setelah dilakukan perbandingan dengan TSP heuristik dan metode SOM sebelumnya, parameter yang telah ditetapkan membuat SOM yang dikembangkan mempunyai performansi yang baik yaitu mempunyai jarak tempuh yang lebih minimal. Lalu dalam penyusunan perjalanan, sistem dapat menyusun perjalanan menyesuaikan durasi wisata wisatawan dengan jam operasional wisata.

**Kata kunci :** jaringan saraf tiruan, *travelling salesman problem*, rute wisata, SOM

---

#### Abstract

Bandung Raya is one of the areas in Indonesia that the development of tourism is quite fast. The vast area of Bandung Raya, the number of new tourist spots emerging, until the first time tourists first come, making tourists quite difficult in knowing what place should be visited and which route to go which adjust resources owned by tourists like the time they have. To handle this then the designed application is a system that provides the most minimum route and know the arrangement of travel plans that adjust the time they have. The approach used in searching tourist route is *Artificial Neural Network* model that is *Self Organizing Maps* (SOM). The result that the best parameter to train SOM network in searching the most minimum route is *learning rate*  $\alpha=0.7$ , number of *neuron* = 8 per city,  $N$  (iteration) = 800, and  $r_0=23.2$ . After the comparison with the heuristic TSP and SOM method before, the parameters that have been established to make SOM developed has a good performance that has a more minimal mileage. Then in the preparation of travel, the system can arrange travel adjust the duration of tourist tours with hours of tourism operations.

**Keywords:** artificial neural network, *travelling salesman problem*, tourists route, SOM

---

#### 1. Pendahuluan

Pariwisata di Indonesia adalah salah satu sektor penting di bidang ekonomi. Pada tahun 2009, pariwisata menempati urutan ketiga dalam hal penerimaan devisa setelah komoditi minyak dan gas bumi serta minyak kelapa sawit [1]. Namun masih terdapat beberapa masalah di sektor wisata menurut Menteri Pariwisata Arif Yahya [2] dan Menteri Pariwisata sebelumnya Maria Elka Pangestu [3]. Masalah tersebut adalah destinasi wisata, sarana dan prasarana, infrastruktur, marketing, dan teknologi informasi. Masalah lainnya adalah terdapat beberapa wisatawan yang belum mengetahui tempat dan rute wisata di Bandung Raya yang sesuai dengan minat dan sumber daya yang dimilikinya seperti waktu wisata. Penyebabnya adalah banyaknya wisatawan yang baru pertama kali datang, tempat wisata baru bermunculan, dan juga wilayah Bandung Raya yang sangat luas.

Untuk menangani masalah tersebut maka aplikasi yang akan dirancang adalah sebuah sistem yang dapat membantu wisatawan mengetahui rute terbaik untuk mencapai setiap tempat pariwisata yang akan dituju lalu mengetahui susunan rencana perjalanan yang menyesuaikan waktu wisatawan dan jam operasional wisata.

Rute wisata yang akan dirancang adalah bagaimana wisatawan mulai berangkat dari tempatnya saat ini misalnya hotel tempat menginap, kemudian mengunjungi tempat-tempat wisata yang ingin dikunjunginya tepat satu kali dan

kembali lagi ke tempat asal wisatawan berangkat. Maka pendekatan yang akan digunakan adalah *Travelling Salesman Problem* (TSP).

TSP yang memiliki pola dapat diimplementasikan menggunakan Jaringan Saraf Tiruan (JST) yang dapat mengenali pola tertentu. JST memiliki beberapa model yaitu model *backpropagation*, model *simulated annealing*, model *hopfield*, dan model *Self-Organizing Map* (SOM). Lalu metode SOM dipilih karena pada beberapa paper [4,5] menyimpulkan bahwa metode ini lebih baik dan efisien dari metode lainnya.

Rumusan masalah yang didapatkan adalah bagaimana penerapan SOM untuk pencarian rute untuk kasus TSP, bagaimana performansi metode SOM, dan bagaimana memberikan susunan rencana perjalanan pada wisatawan dalam waktu wisata yang tersedia.

Untuk mengetahui parameter SOM yang dapat digunakan untuk pencarian rute wisata, maka akan dilakukan pengujian beberapa parameter berbeda hingga didapatkan parameter yang cocok untuk kasus rute pariwisata. Lalu akan diukur performansi dari SOM menggunakan parameter yang telah didapatkan untuk memastikan apakah SOM sudah berjalan dengan baik dan dapat mencari rute yang minimum/terdekat. Lalu sistem memberikan penyusunan rencana perjalanan untuk wisatawan.

## 2. Dasar Teori /Material dan Metodologi/perancangan

### 2.1 Travelling Salesman Problem

*Travelling Salesman Problem* atau yang dikenal dengan TSP adalah sebuah problem untuk menentukan rute seorang salesman. Dalam menyelesaikan urusan bisnisnya ke beberapa kota, sales tersebut memulai dari sebuah kota, kemudian menuju kota-kota yang ingin ditujunya tepat satu kali dan kembali menuju kota awalnya tanpa melalui kota yang telah dikunjunginya. Rute dan jarak yang ditempuhnya harus optimal dan terdekat.

### 2.2 Self Organizing Map

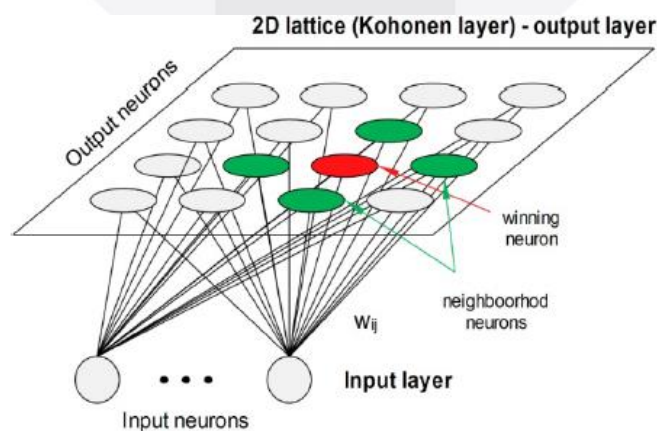
*Self-Organizing Map* atau SOM adalah salah satu model dari JST. Metode ini ditemukan oleh professor asal Finlandia bernama Tuevo Kohonen pada tahun 1982 sehingga metode ini disebut juga sebagai metode *Kohonen*.

SOM masuk dalam kategori *competitive learning*. SOM juga termasuk kedalam *unsupervised learning* dimana tidak ada bantuan atau pengawasan pada saat pembelajaran/*learning*. Hal tersebutlah yang membuat SOM menjadi menarik. SOM merepresentasikan data multidimensional ke dalam satu atau dua dimensi. Proses ini yaitu mengurangi dimensi vektor, pada dasarnya adalah teknik kompresi data yang dikenal sebagai *Vector Quantisation*.

Selama proses pembelajaran unit *input* yang hampir sama dikelompokkan kedalam kelompok tertentu yang berada di unit *output*. SOM bergantung pada *weight* dimana *weight* akan berubah-ubah untuk mengurangi kesalahan/*error* pada sistem. Hal ini dilakukan berulang kali dan dengan banyak pasangan vektor sampai jaringan memberikan *output* yang sesuai.

### 2.3 Arsitektur Self-Organizing Map

Pada SOM, jaringan terdiri dari 2 layer, yaitu *input layer* dan *output layer*. Setiap *node* pada *input layer* terhubung dengan seluruh *node* pada *output layer*. Namun *node* pada *output layer* tidak terhubung satu sama lain. Pada gambar 1 berikut merupakan visualisasi dari arsitektur SOM.



Gambar 1 Arsitektur SOM [6]

Setiap *node* pada input mempunyai posisi topologi spesifik (koordinat x dan y). Jika sebuah *input layer* terdiri dari vektor  $V$  dari  $n$  dimensi ( $V_1, V_2, V_3, \dots, V_n$ ) maka setiap *node input* akan mempunyai bobot vektor ke masing-

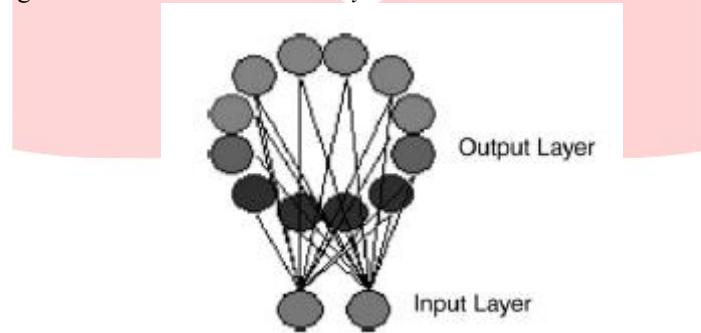
masing *output layer* sesuai jumlah *node* pada *output layer* yaitu *weight vector*  $W$  dari  $n$  dimensi ( $W_1, W_2, W_3, \dots, W_n$ ) [7].

### 2.4 Penerapan TSP pada SOM

Arsitektur yang digunakan tetap sama yaitu pada grup pertama, setiap *neuron* terhubung dengan setiap *neuron* lain dari kumpulan *neuron*. Lalu nilai *weight* tergantung pada jarak antar *neuron*.

Langkah-langkahnya adalah sebagai berikut [8].

1. Tempatkan *neuron* pada *output layer* untuk pola akhir yang diinginkan (*neuron* pada *output layer* akan memiliki sebuah atribut dari posisi  $x$  dan  $y$ . Topologi yang digunakan adalah 2 dimensi).
2. Didalam TSP, solusi akhir atau *output layer* adalah berupa *ring/cincin/lingkaran* terlihat pada gambar 2 yang menghubungkan seluruh kota/*city*/tempat wisata dan cincin ini harus yang terpendek. Lalu tempatkan  $k$  *neuron* secara seragam dalam format cincin.  $k \geq \text{city}$ .



Gambar 2 SOM pada TSP [9]

3. *Input* untuk kasus ini adalah koordinat dari sebuah titik/kota/*city*/tempat wisata. Jadi *input layer* akan berjumlah 2 *neuron* yaitu  $x_i$  dan  $y_i$ .  $N=2$ .
4. Lalu inialisasi seluruh *weight* dengan nilai *random*.
5. Pilih titik/kota/*city*/tempat wisata dan gunakan koordinatnya, lalu gunakan sebagai input.
6. Hitung nilai  $J$  yang membuat  $D(J)$  menjadi minimum.

$$D(J) = \sqrt{(w_{xj} - x_i)^2 + (w_{yj} - y_i)^2} \quad (1)$$

7. Update bobot pemenang dengan rumus berikut.

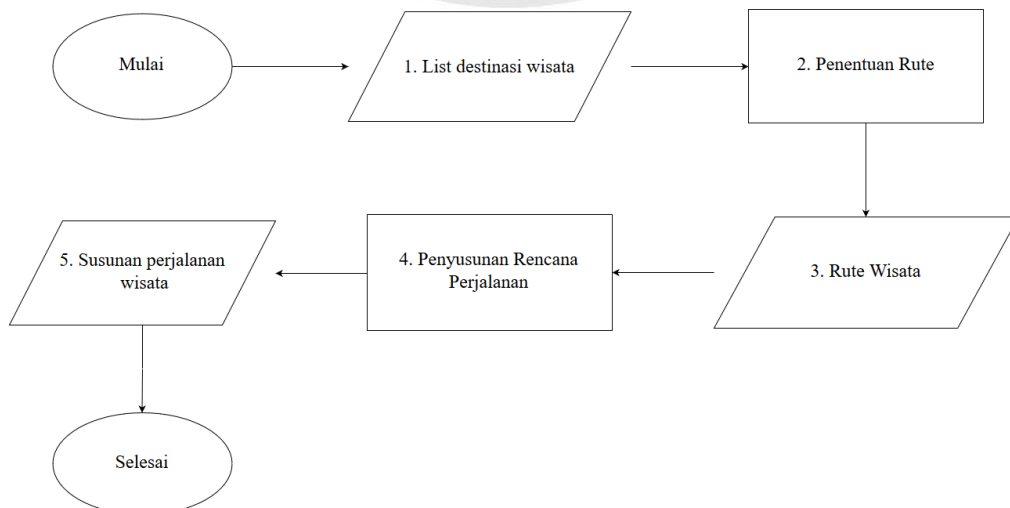
$$w_{xi} = w_{xi}(\text{old}) + \alpha \cdot \theta [x_i - w_{xi}(\text{old})] \quad (2)$$

$$w_{yi} = w_{yi}(\text{old}) + \alpha \cdot \theta [y_i - w_{yi}(\text{old})] \quad (3)$$

8. Lalu perbaharui *learning rate*,  $\theta$ ,  $\sigma$ .
9. Kembali ke langkah 5

### 2.5 Metode Pengembangan Sistem

Berikut adalah metode yang menjelaskan keseluruhan aplikasi yang akan dikembangkan. Metode yang akan dikembangkan tidak menggunakan sistem rekomendasi dengan ontologi. List destinasi wisata yang digunakan dianggap sudah direkomendasikan oleh sistem.



Gambar 3 Metode Pengembangan Sistem

Constraint dan batasan yang digunakan untuk menentukan daftar wisata, susunan destinasi wisata, dan rute wisata adalah jarak/waktu tempuh, waktu/jam operasional tempat wisata, *traffic*, dan tipe endaraan.

## 2.6 Metode Penyusunan Rencana Wisata

Setelah didapatkan rute wisata berdasarkan jarak antar lokasi wisata, kemudian akan dilakukan pengecekan pada jam operasional wisata dan waktu wisatawan.

Jika jam kunjungan pada setiap tempat wisata sesuai dengan jam operasional wisata dan sesuai dengan waktu wisatawan, maka susunan wisata tidak ada perubahan dan tetap seperti yang telah didapatkan sebelumnya.

Jika jam kunjungan pada suatu tempat wisata tidak sesuai dengan jam operasional wisata, misalnya lokasi wisata tersebut belum buka, maka lokasi wisata tersebut mengalami perubahan dan dipindahkan kunjungan terakhir.

Jika jam kunjungan pada suatu tempat wisata tidak sesuai dengan jam operasional wisata, misalnya lokasi wisata tersebut sudah tutup atau diluar waktu wisata, maka lokasi wisata tersebut mengalami perubahan dan direkomendasikan untuk kunjungan hari selanjutnya.

## 3. Pembahasan

### 3.1. Data Pengujian

SOM tidak memerlukan data pelatihan untuk proses pembelajaran, tetapi menggunakan data input sebagai pembelajarannya. Maka data wisata yang telah dikumpulkan digunakan untuk pelatihan dan pengujian secara bersamaan.

### 3.2. Hasil Pengujian Skenario Pertama

Pada skenario pertama pengujian menggunakan data set TSP dari *University of Waterloo, Canada* [10]. Keduanya menyediakan hasil riset, pengujian, dan kumpulan data terbaru mengenai TSP. Hasil risetnya telah digunakan oleh banyak kalangan karena merupakan salah satu hasil riset yang terbaik di bidang TSP. *Map/Data set: Western Sahara (29 Cities)*. Parameter yang diujikan adalah sebagai berikut:

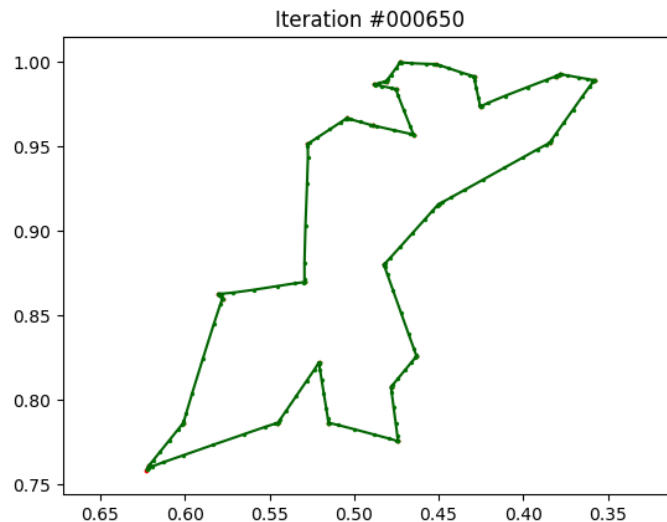
- *Neuron per city*: 2, 4, 6, 8, 10;
- *Learning rate*  $\alpha$ : 0.2, 0.5, dan 0.7

Berikut adalah hasil pengujian terbaik pada skenario pertama:

Tabel 1 Pengujian SOM dengan *Neuron = 8 per city*,  $\alpha=0.7$ , dan  $r_0=23.2$

<i>Map</i>	<i>Decay</i>	<i>Neurons</i>	<i>N</i>	$\alpha_0$ ( <i>decay</i> )	$r_0$ ( <i>decay</i> )	TSP <i>Dist.</i>
<b><i>Western Sahara</i></b>	<i>Exponential</i>	8 <i>Neuron per City</i>	500	0.7 (0.999)	23.2 (0.995)	27601.173
			500			27610.293
			500			27739.383
			500			28010.572
			650			27601.173
			800			27610.293
			800			28802.077

Gambar pada tabel 1 menunjukkan jarak tempuh paling minimal sebanyak lebih dari 1 kali pada iterasi ke 500 dan 650. Namun rute yang sudah terbentuk dengan sangat baik yaitu pada iterasi ke-650 terlihat pada gambar 4. *Neuron* sudah membentuk rute dan tidak ada titik inputan yang belum dikunjungi oleh *neuron*.



Gambar 4 Rute optimal pada iterasi ke-650 pada pengujian SOM dengan  $Neuron = 8$  per city,  $\alpha=0.7$ , dan  $r_0=23.2$

Dari beberapa pengujian yang telah dilakukan pada skenario pertama, maka didapatkan hasil analisis sebagai berikut.

1. *Learning rate* yang rendah tidak dapat digunakan untuk kasus pencarian rute pada SOM. Maka didapatkan *learning rate* yang cukup baik untuk digunakan pada pengujian selanjutnya yaitu *learning rate*  $\alpha=0.7$ .
2. Jumlah *neuron* juga mempengaruhi dalam pembentukan rute yang baik dan jarak yang minimal. Jumlah *neuron* yang cukup stabil adalah 8 *neuron per city*.
3. *Neighborhood variance* yang digunakan adalah  $r_0=23.2$ .
4. Komputasi pada SOM cukup cepat dalam pencarian rute yang optimal.

### 3.3. Hasil Pengujian Skenario Kedua

Pada skenario kedua berikut ini adalah perbandingan hasil optimal rute TSP menggunakan algoritma SOM dengan TSP heuristik dari hasil riset, pengujian, dan kumpulan data *Faculty of Mathematics University of Waterloo*.

Tabel 2 Perbandingan rute optimal menggunakan SOM dan TSP heuristik

<i>Map</i>	Jumlah Titik/ <i>Cities</i>	TSP Dist. SOM	TSP Dist. Heuristic
<b><i>Western Sahara</i></b>	29 <i>Cities</i>	27601.17	27603
<b><i>Djibouti</i></b>	38 <i>Cities</i>	6659.91	6656

Pada *dataset* Western Sahara, algoritma SOM menghasilkan optimal *tour/distance* yang lebih baik. Lalu pada *dataset* Djibouti hanya berbeda sedikit sekali.

Pada kasus kunjungan wisata diperkirakan jumlah wisata tidak melebihi dari batas jumlah maksimal data set diatas. Jadi untuk kasus rute wisata dapat menggunakan parameter pada *dataset* Western Sahara dan Djibouti yang sudah cukup baik.

### 3.4. Hasil Pengujian Skenario Ketiga

Pada skenario ketiga dilakukan perbandingan terhadap 3 algoritma SOM heuristik yang telah dilakukan pengujian untuk mengetahui apakah SOM yang telah dirancang lebih baik. SOM heuristik tersebut adalah *PKN-Pure Kohonen Network* [11,12], *GN-Guilty Net*[13], dan *AVL- The procedure of Ange'niol, Vaubois and La Texier*[14].

Tabel 3 Hasil pengujian dari PKN, GN, AVL, dan SOM yang telah diuji

Map	PKN	GN	AVL	Current SOM
Eil51	443.9	470.7	443.5	442.946
St70	692.8	755.7	693.3	690.687

Hasil tabel 3 diatas menunjukkan bahwa pengujian skenario 3 sudah sangat baik. Hasil jarak yang paling minimal sudah lebih baik dibandingkan 3 SOM heuristik sebelumnya untuk data set yang sama.

Pada *Eil51* parameter yang digunakan mengacu kepada parameter terbaik yang telah didapatkan pada skenario pertama yaitu dengan jumlah 8 *neuron per city*,  $\alpha=0.7$ ,  $N=1500$ , dan *neighborhood variance* sebesar 61.2. Pada *St 70* digunakan 8 *neuron per city*,  $\alpha=0.8$ ,  $N=2000$ , dan *neighborhood variance* sebesar 61.2.

Perbedaan hanya pada iterasi yang lebih besar untuk menyesuaikan *dataset* dengan jumlah *city* yang lebih besar.

### 3.5. Hasil Pengujian Skenario Keempat

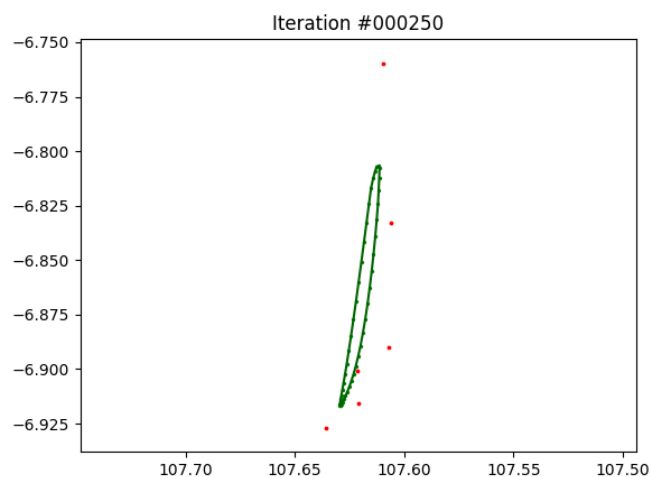
Pada skenario keempat menguji sistem penyusunan rencana perjalanan. Lokasi yang digunakan sebagai data uji adalah berupa 5 lokasi wisata yang dianggap sudah direkomendasikan oleh sistem.

Parameter yang digunakan adalah mengacu pada parameter optimal yang telah didapatkan pada skenario pertama yaitu dengan jumlah neuron 8 *per city*,  $\alpha=0.7$ ,  $N=800$ , dan *neighborhood variance* sebesar 23.2.

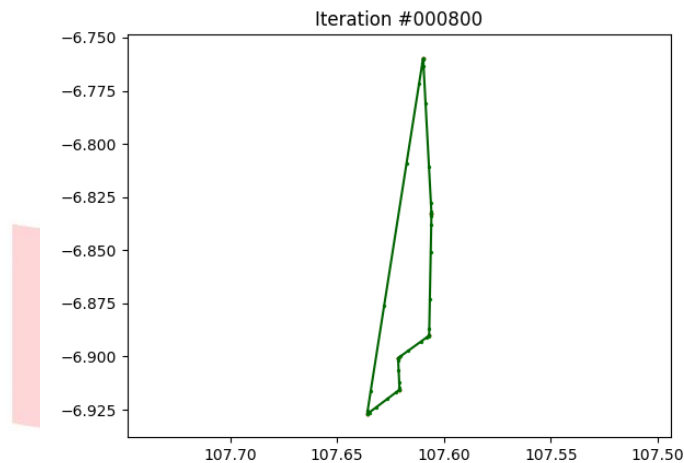
Berikut adalah pengujian untuk wisata 1 hari.

- Lokasi awal: Trans Luxury Hotel Bandung
- Lokasi tujuan destinasi wisata:
  1. Tangkuban Perahu
  2. Museum Geologi
  3. Farm House Lembang
  4. Kebun Binatang Bandung
  5. Toko Sepatu Brodo Footwear
- Masukkan program:
  1. Nama wisatawan: David Winalda
  2. Jam mulai wisata: 08.00 WIB
  3. Jam akhir wisata: 17.00 WIB
  4. Waktu tempuh (Google Maps API)
  5. Lokasi destinasi wisata
- Keluaran program:

Dibawah ini adalah beberapa ilustrasi dari pengujian untuk mencari rute dan jarak yang paling minimal:



Gambar 5 Proses pencarian rute pada iterasi ke-250



Gambar 6 Proses pencarian rute pada iterasi ke-800

Pada kasus dibawah, rencana perjalanan yang direkomendasikan telah sesuai dengan jam operasional wisata. Namun pada kasus tertentu jika jam kunjungan wisata tidak sesuai dengan jam buka lokasi wisata, maka kunjungan ke lokasi wisata akan dikunjungi pada urutan terakhir dari susunan wisata yang telah direkomendasikan.

Jika saat kunjungan akhir, lokasi wisata telah tutup maka lokasi wisata tersebut akan direkomendasikan untuk esok hari.

Lalu susunan rencana perjalanannya adalah sebagai berikut.

Tabel 4 Hasil penyusunan rencana perjalanan wisata SOM

No	Tujuan	Jam Tiba	Durasi Wisata	Selesai Wisata	Waktu Tempuh Lokasi Berikutnya	Jam Berangkat ke lokasi Berikutnya
1	<i>Trans Luxury Hotel</i>	-	-	-	97 Menit	08.00
2	Tangkuban Perahu	09.37	60 menit	10.37	50 Menit	10.37
3	<i>Farm House Lembang</i>	11.27	60 menit	12.27	28 Menit	12.27
4	Kebun Binatang Bandung	12.55	60 menit	13.55	08 Menit	13.55
5	Museum Geologi	14.03	60 menit	15.03	09 Menit	15.03
6	Toko Sepatu Brodo Footwear	15.12	60 menit	16.12	12 Menit	16.12
7	<i>Trans Luxury Hotel</i>	16.24	-	-	-	-

Berdasarkan tabel diatas, sistem telah dapat menyusun rencana perjalanan dengan baik yaitu dapat menyesuaikan durasi waktu wisata dari wisatawan, waktu tempuh tempat wisata, dan jam operasional wisata sehingga setiap lokasi wisata dapat dikunjungi.

#### 4. Kesimpulan

Dari beberapa pengujian dan analisis yang dilakukan, kesimpulan yang dapat diambil adalah:

1. Hasil dari skenario pertama menunjukkan bahwa parameter yang optimal adalah menggunakan 8 neuron per city/lokasi/tempat wisata, learning rate  $\alpha=0.7$ , iterasi  $N=650-1000$ , dan neighborhood function  $r_0=23.2$ . Learning rate ( $\alpha$ ) yang rendah membuat jaringan SOM belajar dan berubah dengan cukup lambat. Jumlah neuron yang sedikit juga mempengaruhi dikarenakan tidak dapat menyebar dengan baik untuk membentuk rute pada titik input neuron. Lalu iterasi yang baik berada pada  $N = 800$ . Range iterasi 650 – 1000 dapat digunakan juga menyesuaikan jumlah lokasi wisata.

2. Hasil dari skenario kedua adalah SOM yang digunakan untuk kasus rute TSP heuristik dari hasil riset, pengujian, dan kumpulan data *Faculty of Mathematics Univesity of Waterloo* ini sudah cukup baik di dua *dataset* yang telah diuji yaitu Western Sahara dan Djibouti. Hasilnya pada data set Western Sahara mempunyai jarak tempuh yang lebih minimal dibandingkan TSP heuristik tersebut.
3. Pada perbandingan terhadap 3 SOM heuristik yang telah dilakukan pengujian dan dijadikan pembandingan untuk mengetahui apakah SOM yang dirancang lebih baik, hasilnya adalah lebih baik dari 3 SOM heuristik sebelumnya.
4. Selama masa pengujian, rute yang dihasilkan telah sesuai dengan jam operasional wisata. Untuk wisata 1 hari jumlah lokasi wisata yang optimal adalah 5-6 wisata, lalu untuk wisata 2 hari jumlah lokasi wisata yang optimal adalah 8-10 lokasi wisata, dan untuk wisata 3 hari jumlah lokasi yang optimal adalah 11-13 lokasi wisata. Optimal disini adalah sesuai dengan durasi waktu wisata wisatawan, jam operasional, dan waktu tempuh.

Saran yang dapat diberikan untuk pengembangan sistem pencarian rute berbasis TSP dengan SOM adalah:

1. Memodifikasi SOM untuk menyesuaikan waktu wisata wisatawan dengan jam operasional wisata.
2. Memperbaiki tampilan rute dengan menggunakan tampilan *Google Maps*.

#### Daftar Pustaka:

- [1] Pariwisata, Kementrian (n.d.). Ranking Devisa Pariwisata Terhadap Komoditas Ekspor Lainnya tahun 2004-2009. (Kementrian Kebudayaan dan Pariwisata RI) Retrieved Juni 21, 2011, from [https://www.budpar.go.id/filedata/5436\\_1695-Rankingdevisa.pdf](https://www.budpar.go.id/filedata/5436_1695-Rankingdevisa.pdf).
- [2] Rahmawati, Y. (2014, Oktober 28). Home: Berita Nasional. Retrieved from Bisnis Wisata: <http://bisniswisata.co.id/arief-yahya-lima-permasalahan-pariwisata-yang-harus-dihadapi/>
- [3] Nursastri, S. A. (2014, Februari 26). detikTravel. Retrieved from Detik Travel:<http://travel.detik.com/read/2014/02/26/152056/2509137/1382/ini-dia-7-masalah-utama-pariwisata-di-indonesia>
- [4] Maire, B. F., & Mladenov, V. M. (2012). Comparison of Neural Network for Solving the Travelling Salesman Problem.
- [5] Budinich, M., & Rosario, B. (1995). A Neural Network for the Travelling Salesman Problem with a Well Behaved Energy Function.
- [6] Wikimedia. (n.d.). Retrieved from Wikimedia: [https://upload.wikimedia.org/wikipedia/commons/thumb/4/46/Colored\\_neural\\_network.svg/2000px-Colored\\_neural\\_network.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/4/46/Colored_neural_network.svg/2000px-Colored_neural_network.svg.png)
- [7] Kohonen's Self Organizing Feature Maps. (n.d.). Retrieved from Ai-Junkie: [www.ai-junkie.com/ann/som/som1.html](http://www.ai-junkie.com/ann/som/som1.html)
- [8] Neural Network: Travelling Salesman Problem. (n.d.). Retrieved from Travelling Salesman Problem:<http://www.patol.com/java/TSP/index.html>
- [9] Bai Y, Wendong Z, Zhen J. An New Self-Organizing Maps Strategy for Solving the Travelling Salesman Problem. Elsevier 2005.
- [10] Faculty of Mathematics Univesity of Waterloo, C. N. (2016). Traveling Salesman Problem. Retrieved from Traveling Salesman Problem:<http://www.math.uwaterloo.ca/tsp/index.html>
- [11] Hueter GJ. Solution of the traveling salesman problem with an adaptive ring. Proc IEEE Int Conf Neural Networks (I-85-92) 1988.
- [12] Fort JC. Solving combinatorial problem via self-organizing process: an application of the Kohonen algorithm to the traveling sales-man problem. Biol Cyber 1988;59:33-40.
- [13] Burke LI, Damany P. The guilty net for the traveling salesman problem. Comput Oper Res 1992;19:255-65.
- [14] Angeniol B, Vaubois C, Le Texier JY. Self-organizing feature maps and the traveling salesman problem. Neural Networks 1988;1:289-93.