

IMPLEMENTASI ALGORITMA *SIMULATED ANNEALING* TERHADAP RUTE PERJALANAN PADA SISTEM REKOMENDASI OBJEK WISATA

Rovianty Nugracia,
Dr. Kemas Muslim Lhaksana, S.T., M.ISD.

Fakultas Informatika, Universitas Telkom, Bandung
Email: roviantynugracia@gmail.com, kemas.muslim@gmail.com

Abstrak

Bandung Raya memiliki berbagai objek wisata yang terkenal dan sering dikunjungi wisatawan. Banyaknya objek wisata dapat mempengaruhi keputusan wisatawan dalam melakukan perjalanan wisata. Wisatawan yang memilih menggunakan jasa dari biro perjalanan biasanya ingin mengunjungi beberapa lokasi sekaligus dengan mengupayakan waktu yang singkat. Melalui Tugas Akhir ini, penulis membangun sebuah sistem untuk menentukan rute perjalanan wisata, dengan menggunakan algoritma *Simulated Annealing* (SA).

Dalam menentukan rute atau jalur wisata, sistem harus mampu memberikan estimasi perjalanan terbaik, dengan melibatkan aspek kecepatan waktu dan jarak terpendek yang bisa dilalui, serta mengoptimalkan jumlah destinasi sebanyak mungkin. Rute wisata dirancang dengan titik keberangkatan dari hotel tempat menginap, kemudian menuju titik-titik yang menjadi lokasi wisata, hingga kembali lagi ke hotel tempat wisatawan menginap. Maka pendekatan yang digunakan adalah *Traveling Salesman Problem* (TSP).

Penulis memilih algoritma SA untuk digunakan dengan asumsi bahwa SA mampu menyelesaikan permasalahan TSP dengan cara lebih cepat dibandingkan dengan algoritma lainnya, dengan cara simulasi. Perhitungan TSP dengan menggunakan algoritma SA dapat digunakan untuk mencapai solusi yang sub-optimal diuji dengan melibatkan 5-10 titik destinasi wisata. Algoritma SA mengevaluasi parameter dengan mengukur dan membandingkan nilai jarak sebagai parameter. Parameter yang nilainya lebih besar dari nilai sebelumnya akan ditolak pada saat iterasi, namun apabila ditemukan parameter dengan nilai lebih kecil pada saat iterasi, maka parameter tersebut yang akan diterima untuk menggantikan nilai parameter sebelumnya. Semakin banyak titik lokasi yang dipilih oleh wisatawan maka semakin banyak proses *annealing* dilakukan dan semakin banyak pula iterasinya.

Kata kunci: rute, SA, simulated annealing, TSP, traveling salesman problem.

Abstract

Bandung Raya has various famous tourist attraction and is often visited by tourists. Those tourist attractions influence the decision to travel. Travelers who choose to use the services of travel agencies usually want to visit multiple locations at once by striving for a short time. Through this final project, the authors build a system for determining the travel route, using algorithms *Simulated Annealing* (SA).

In process of determining routes or travel lane, the system must be able to give an estimate of the best trips, involving aspects shortest time speed and distance that can be passed along as much as possible to optimize the number of destinations. These are designed with the tour departure point of the hotel where you stayed, then headed dots become tourist sites, to come back to the hotel where tourists stay. Then the approach used is *the Traveling Salesman Problem* (TSP).

The author chose this algorithm to be used with assumptions that SA can solve TSP problems faster than any other algorithm, by doing simulation. TSP calculation by using this algorithm can be used to achieve sub-optimal solutions tested by involving 5-10 tourist destination point. SA algorithm evaluates the parameters by measuring and comparing the value of distance as a parameter. The parameter value is greater than the previous value will be rejected during the iteration, but if found parameter to a smaller value at the time of the iteration, then the parameters that would be acceptable to replace the previous parameter value.

Keywords: route, SA, simulated annealing, TSP, traveling salesman problem.

1. Pendahuluan

Latar Belakang

Bandung Raya memiliki berbagai objek wisata yang terkenal dan sering dikunjungi wisatawan, mulai dari Gedung Sate, Museum Geologi Bandung, hingga wisata alam seperti Gunung Tangkuban Parahu, Kawah Putih, Situ Patenggang dan lain-lain. Banyaknya objek wisata turut mempengaruhi keputusan wisatawan dalam melakukan perjalanan wisata. Tidak sedikit para wisatawan mengalami kesulitan dalam menentukan list tempat yang ingin dikunjungi. Hal ini dapat didasari oleh minimnya pengetahuan para wisatawan terkait lokasi wisata di kawasan Bandung Raya, dengan alasan baru pertama kali datang atau kemunculan tempat-tempat wisata yang baru. Maka sebagian besar wisatawan memilih menggunakan jasa dari biro perjalanan.

Umumnya wisatawan ingin melakukan perjalanan wisata dengan mengunjungi beberapa lokasi sekaligus dengan mengupayakan waktu yang singkat. Oleh karena itu, biro perjalanan biasanya menyediakan jasa perjalanan wisata ke beberapa lokasi dalam bentuk paket wisata. Berdasarkan kondisi tersebut, wisatawan memerlukan suatu sistem atau aplikasi untuk mendukung dalam pemilihan rute perjalanan sehingga seluruh target objek wisata dapat dikunjungi secara optimal.

Dalam menentukan rute atau jalur wisata, sistem harus mampu memberikan estimasi perjalanan terbaik, dengan melibatkan aspek kecepatan waktu dan jarak terpendek yang bisa dilalui, serta mengoptimalkan jumlah destinasi sebanyak mungkin. Rute wisata dirancang dengan titik keberangkatan dari hotel tempat menginap, kemudian menuju titik-titik yang menjadi lokasi wisata, hingga kembali lagi ke hotel tempat wisatawan menginap. Maka pendekatan yang akan digunakan adalah *Traveling Salesman Problem (TSP)*.

TSP dikenal sebagai salah satu pendekatan optimasi yang banyak menarik perhatian para peneliti sejak beberapa dekade terdahulu. Pada mulanya, TSP dinyatakan sebagai permasalahan dalam mencari jarak minimal pada suatu perjalanan terhadap sejumlah n destinasi di mana setiap destinasi yang terlibat hanya dikunjungi sekali dengan destinasi awal = destinasi akhir. Pada kasus umum TSP, semua destinasi digambarkan dalam node yang sama nilainya, namun untuk kasus penentuan rute perjalanan wisata, setiap destinasi digambarkan dalam node berbeda berdasarkan kategori diantaranya penginapan seperti hotel, wisata alam, pendidikan contohnya museum, serta kuliner.

Dalam menyelesaikan masalah, metode pendekatan TSP didukung dengan algoritma tertentu. Terdapat berbagai algoritma yang dapat digunakan untuk mendukung TSP, salah satunya algoritma *Simulated Annealing (SA)*. SA merupakan metode pencarian yang memanfaatkan teori probabilitas untuk mencari global minimum dari suatu permasalahan optimasi. Penulis memilih algoritma SA untuk digunakan dengan asumsi bahwa SA mampu menyelesaikan permasalahan TSP dengan cara lebih cepat dibandingkan dengan algoritma lainnya, dengan cara simulasi. Penulis bermaksud menguji implementasi algoritma pencarian rute terpendek yang melibatkan *Simulated Annealing*, serta menganalisis performansi algoritma tersebut.

Topik dan Batasannya

Berdasarkan latar belakang yang telah dijelaskan, topik masalah yang dibahas pada tugas akhir ini mencakup penerapan algoritma SA untuk pencarian rute berdasarkan kasus TSP, kemudian analisis performansi algoritma SA dalam menghasilkan rute wisata terbaik, serta diharapkan luaran program dapat menghasilkan susunan rencana perjalanan wisata bagi wisatawan berdasarkan waktu yang tersedia.

Penelitian ini memiliki batasan masalah, yaitu wilayah rute wisata hanya di daerah Bandung Raya, Jawa Barat.

Tujuan

Melalui penelitian / tugas akhir ini diharapkan penulis dapat mengetahui dan membuktikan apakah algoritma *Simulated Annealing* bekerja baik untuk pencarian rute yang optimal berdasarkan kasus TSP, serta mengetahui dan mengukur performansi algoritma *Simulated Annealing* dalam pencarian rute terbaik dan jarak minimal. Selain itu program yang dibangun memiliki tujuan untuk menghasilkan susunan rencana wisata yang optimal bagi wisatawan.

Organisasi Tulisan

Jurnal TA berisi abstrak, pendahuluan, studi pustaka (*literature review*), sistem yang dibangun, evaluasi, dan kesimpulan.

Bagian abstrak menjelaskan secara ringkas mengenai tugas akhir mencakup alasan yang melatarbelakangi pemilihan topik, solusi masalah, juga kesimpulan. Pendahuluan membahas alasan penelitian dilakukan beserta permasalahan yang terjadi. Kemudian solusi yang ditawarkan dari permasalahan yang ada pada bidang wisata. Studi pustaka berisi kajian teori yang mendukung penelitian. Teori yang digunakan dapat diambil dari beberapa referensi.

Perancangan sistem membahas model konseptual beserta sistematika penelitian untuk pemecahan masalah. Evaluasi berisi hasil pengujian berdasarkan metode dan skenario pengujian, sekaligus menjelaskan analisis terhadap hasil pengujian tersebut. Hasil dari kegiatan analisis ini kemudian dijadikan sebagai dasar

pengambilan kesimpulan. Kesimpulan menjawab tujuan penelitian, dapat disertai saran yang diberikan pada penelitian ini untuk pengembangan sistem selanjutnya.

2. Studi Terkait

Sistem Rekomendasi

Sistem Rekomendasi merupakan suatu model aplikasi dari hasil observasi yang dapat mendukung pengguna dalam mencari dan memilih produk, layanan, atau informasi. Sistem rekomendasi memiliki potensi untuk mendukung dan meningkatkan kualitas keputusan konsumen saat melakukan pencarian dan memilih produk secara online. Pada dasarnya sistem rekomendasi memiliki 3 (tiga) teknik dalam pengaplikasiannya, yaitu *Collaborative Filtering*, *Content Based Filtering*, dan *Knowledge Based*.

Collaborative Filtering membuat rekomendasi berdasarkan rating dari pengguna. Sebagai contoh, pada *Recommended System Handphone*, rating yang diberikan oleh pengguna *handphone* merk A akan dibandingkan dengan rating yang diperoleh *handphone* merk B. *Handphone* yang memiliki rating paling tinggilah yang akan direkomendasikan.

Content Based Filtering membuat rekomendasi berdasarkan deskripsi dari profil pengguna, atau deskripsi produk. Sebagai contoh, apabila pengguna menyukai film Harry Potter maka berdasarkan deskripsi tersebut sistem dapat merekomendasikan film-film lainnya yang memiliki genre sejenis.

Knowledge Based membuat rekomendasi dengan memanfaatkan *personalization rule*. *Personalization rule* merupakan aturan-aturan berbasis pengetahuan yang dibuat berdasarkan skala prioritas. Level skala prioritas diatur dengan 6 memprediksikan prioritas kebutuhan pengguna terhadap layanan atau produk. Produk yang memiliki level prioritas paling tinggi akan direkomendasikan bagi pelanggan.

Traveling Salesman Problem

Permasalahan tentang TSP dikemukakan pada tahun 1800 oleh William Rowan Hamilton. Pada saat itu, Hamilton menemukan sebuah permainan dimana pemain harus menyelesaikan perjalanan dari 20 titik dengan hanya melewati jalur-jalur tertentu. Sedangkan bentuk umum TSP sendiri baru dipelajari untuk pertama kalinya pada tahun 1930.

TSP termasuk persoalan yang dipahami melalui kasus seorang salesman yang berkeliling mengunjungi sejumlah kota. Deskripsi persoalannya adalah sebagai berikut: Seorang salesman menggunakan waktunya untuk mengunjungi n kota (*nodes*) secara siklus perputaran. Didalam satu kali perjalanan keliling, ia harus menentukan urutan dari sejumlah kota yang harus dilaluinya, setiap kota hanya boleh dilalui sekali dan hanya sekali dalam perjalanan, dan perjalanan berakhir pada kota awal dimana ia memulai perjalanan.

Kebanyakan TSP merupakan suatu simetris yang berarti untuk dua kota A dan B, jarak dari kota A ke kota B adalah sama dengan jarak dari kota B ke kota A. Dalam hal ini, kita akan mendapatkan panjang perjalanan keliling yang sama persis jika kita membalikkan rute perjalanan tersebut. Jadi tidak ada perbedaan antara suatu perjalanan keliling dan kebalikannya.

Simulated Annealing

Simulated Annealing adalah salah satu metode pencarian acak yang mempunyai kelebihan dalam kemampuan untuk menghindari jebakan optimal lokal, yaitu dengan cara mencari solusi model secara acak dari ruang model sehingga diharapkan solusi yang diambil tidak terjebak dalam minimum lokal. Metode ini mengadopsi dari proses termodinamika dalam pembentukan substansi atau kristal dengan penurunan temperatur yang perlahan sehingga didapatkan bentuk kristal yang homogen. Dengan demikian metode *Simulated Annealing* dikembangkan dengan analogi dari prinsip kristalisasi pada logam dengan proses pendinginan dan pembekuan, sehingga akan diperoleh energi minimum.

Simulated Annealing tidak hanya menerima perubahan fungsi obyektif yang turun akan tetapi juga menerima beberapa perubahan fungsi obyektif yang naik. Penerimaan perubahan fungsi obyektif ini ditentukan oleh persamaan state. Persamaan state adalah probabilitas penerimaan perubahan fungsi obyektif. Beberapa hal yang menjadi komponen penting pada implementasi SA, diantaranya:

State: didefinisikan sebagai kombinasi nilai dari penyelesaian yang mungkin. Pada permasalahan TSP state juga didefinisikan sebagai urutan nomor kota yang dilalui.

Energi: didefinisikan sebagai seberapa besar fungsi tujuan minimal. Dalam hal ini SA digunakan untuk meminimumkan energi dari kombinasi state. Pada permasalahan TSP yang disebut energi adalah jarak yang harus dilalui pada state. Energi dapat dinyatakan dalam urutan nomor kota yang dilalui.

Temperatur: didefinisikan sebagai suatu nilai kontrol yang membuat suatu state acak bisa bergerak naik atau tidak. Seperti halnya analogi pada kejadian thermal, ion-ion akan bergerak bebas pada temperatur yang tinggi, dan semakin terbatas gerakannya ketika temperatur turun. Pada saat iterasi awal diperlukan nilai temperature yang tinggi agar proses pencarian acak punya range yang lebar. Semakin bertambah jumlah iterasi maka temperatur semakin turun, namun nilai temperatur tidak boleh sampai nol. Pada saat nilai temperatur tinggi maka peluang model untuk diterima semakin besar akan tetapi ketika temperatur menurun atau bernilai kecil maka model banyak yang ditolak.

Perancangan SA terhadap TSP

Pada proses simulasi *annealing* dalam permasalahan optimasi, hal-hal yang harus diperhatikan dalam pelaksanaan proses SA yaitu sebagai berikut:

- 1) Inisialisasi rute awal
Pemilihan rute awal dilakukan secara random. Rute awal menjadi start node untuk proses iterasi dalam Simulasi *Annealing*.
- 2) Parameter awal
Parameter awal diharapkan memiliki nilai yang cukup besar sehingga mampu terhindar dari bad local optimal.
- 3) Mekanisme pertukaran
Menentukan destinasi wisata yang dibutuhkan untuk pertukaran solusi yang dianggap sebagai iterasi.
- 4) Fungsi objektif permasalahan
Mengevaluasi setiap fungsi biaya yang berubah karena proses iterasi dari mekanisme pertukaran.
- 5) *Annealing schedule*
Fungsi *annealing schedule* yang umum digunakan adalah

$$T_{i+1} = \alpha \times T_i \quad (2.1)$$
 Dengan α sebagai konstanta untuk menurunkan parameter control. Nilai $\alpha < 1$.
- 6) Kriteria penghentian proses *Simulated Annealing* (SA)
 Penghentian proses algoritma SA dapat dilihat berdasarkan kriteria sebagai berikut.
 - Maksimum jumlah iterasi
 - Nilai minimum parameter control
 - Nilai minimum fungsi objektif.
 - Nilai minimum dari tingkat penerimaan.

Tabel 1. Rumus pada Algoritma SA

	Rumus	Variabel	Keterangan
1	$kT = T_0 \cdot k$	kT = temperature saat ini T_0 = temperature awal k = konstanta penurunan temperatur	Menghasilkan temperature yang digunakan untuk simulasi kemungkinan rute selanjutnya
2	$\Delta E = E_{new} - E_{old}$	E_{new} = energy yang sedang dihitung E_{old} = energy dari node sebelumnya	Menghitung perubahan energy antar kemungkinan solusi rute
3	$p = e^{-\Delta E/kT}$	kT = temperature saat ini ΔE = perubahan temperature	Menghitung probabilitas dan menentukan arah evaluasi node pada setiap iterasi

Dalam implementasinya, algoritma *Simulated Annealing* tidak selalu menemukan solusi yang terbaik. Target SA adalah untuk menemukan solusi yang bagus dan layak diterima. Metode SA mencari solusi dengan berpindah dari solusi yang satu ke solusi yang lainnya, dan apabila solusi baru yang diuji mempunyai nilai fungsi energi yang lebih kecil, maka solusi yang sedang diuji akan menggantikan solusi yang lama. Umumnya solusi baru yang dipilih merupakan solusi yang ada di dekat/sekitar solusi yang lama.

3. Sistem yang Dibangun

Pengumpulan Data

Metode pengumpulan data dilakukan dengan cara studi kepustakaan. Metode ini dilakukan dengan cara menghimpun informasi yang relevan dengan topik dan masalah yang akan atau sedang diteliti. Informasi diperoleh dari buku-buku ilmiah dan non ilmiah, laporan penelitian, dan karangan-karangan ilmiah.

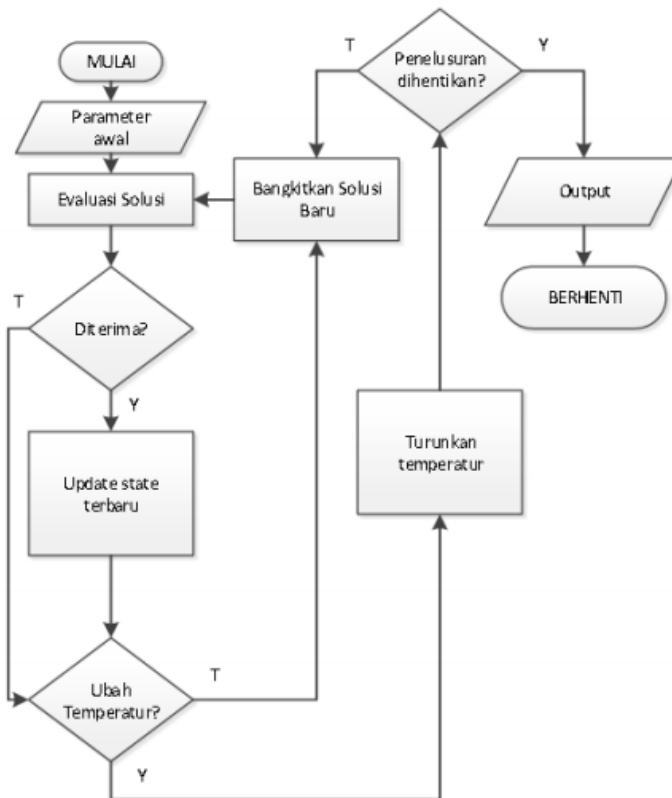
Berdasarkan hasil studi pustaka, penulis memilih pendekatan SA untuk digunakan dengan alasan bahwa SA menyelesaikan permasalahan TSP dengan cara simulasi atau menguji setiap kemungkinan rute yang ada, dimulai dari nilai parameter terbesar (jangkauan lokasi wisata terluas) hingga parameter terkecil. Selain itu, algoritma SA juga tidak biasa ditemukan dalam proses belajar sehari – hari sehingga menjadikan metode ini menarik untuk dianalisis.

Pengembangan Algoritma SA untuk Penentuan Rute Wisata

Metode algoritma SA untuk mencari rute dan penerapannya pada model TSP, adalah sebagai berikut.

1. Inisialisasi parameter awal.

2. Evaluasi rute awal. Apabila rute awal merupakan tujuan, maka pencarian selesai dan keluar. Apabila bukan, maka lanjutkan pencarian dengan menetapkan rute awal sebagai rute yang sekarang.
3. Inisialisasi BEST-SO-FAR (pilihan rute dengan solusi paling layak saat itu) dengan rute yang sekarang.
4. Inisialisasi parameter kontrol T sesuai dengan *annealing schedule*.
5. Ulangi proses hingga solusi ditemukan atau sudah tidak ada lagi aturan yang bisa diaplikasikan ke rute sekarang.
6. BEST-SO-FAR adalah solusi minimum global yang diharapkan.



Gambar 1. Flowchart Metode Algorima SA

4. Evaluasi

4.1 Hasil Pengujian

Pengujian dilakukan dengan scenario bahwa wisatawan memilih 6 (enam) destinasi kunjungan termasuk lokasi awal keberadaan wisatawan.

Tabel 2. Simulasi Daftar Kunjungan

Index	Nama Lokasi	Alamat	Koordinat
0	Trans Luxury Hotel	Jl. Gatot Subroto No.289, Cibangkong, Batununggal, Kota Bandung, Jawa Barat	-6.925357 , 107.636738
1	Situ Patenggang	Patengan, Rancabali, Bandung, Jawa Barat	-7.166304 , 107.357877
2	Tangkuban Parahu	Cikahuripan, Lembang, Kab. Bandung, Jawa Barat	-6.758444 , 107.609914
3	Farm House	Jl. Raya Lembang No.108, Gudangkahuripan, Lembang, Kab. Bandung, Jawa Barat	-6.830162 , 107.606516

4	Gedung Sate Bandung	Jl. Diponegoro No.22, Citarum, Bandung Wetan, Kota Bandung, Jawa Barat	-6.902477, 107.618856
5	Cihampelas Walk	Jl. Cihampelas No.160, Cipaganti, Coblong, Kota Bandung, Jawa Barat	-6.892856, 107.605356

Tabel 3 berikut adalah matriks data jarak antar titik yang diperoleh dari *Google Maps*. Mode perjalanan yang dipilih oleh penulis untuk pengujian adalah mode *driving* dengan arah standart dan sudah sesuai dengan jalur satu arah maupun dua arah, hal ini memungkinkan jarak dari lokasi A ke B tidak sama dengan jarak dari lokasi B ke A. Pada implementasinya dengan *python*, data jarak antar titik diperoleh melalui rumus yang melibatkan data koordinat setiap lokasi. Jarak dihasilkan dalam satuan kilometer (km).

Tabel 3. Matriks Data Jarak antar Titik (km)

Asal	Tujuan						
	Index	0	1	2	3	4	5
	0	0	57	30	18	6.8	11
	1	51	0	80	63	57	56
	2	28	79	0	17	23	27
	3	16	62	17	0	12	9.3
	4	4	57	23	12	0	3.8
	5	6.6	56	28	11	2.7	0

Penyelesaian untuk scenario bahwa wisatawan memilih 6 (enam) destinasi kunjungan termasuk lokasi awal keberadaan wisatawan, dengan proses iterasi sebanyak 10 kali. Ada beberapa operator yang digunakan pada algoritma simulasi *annealing*, diantaranya:

- Node lokasi disimpan pada list atau larik L
- Bangkitkan 2 (dua) bilangan random 1 sampai NC, dengan $N1 < N2$
- Baris depan = L(1) sampai L(N1-1)
- Baris tengah = L(N1) sampai L(N2)
- Baris belakang = L(N2+1) sampai L(NC)
- Bangkitkan bilangan random r
 - Apabila $r < 0.5$, maka
 - a. DepanBaru = Baris depan
 - b. TengahBaru = Baris tengah dengan urutan dibalik dari belakang ke depan
 - c. BelakangBaru = Baris belakang
 - Apabila $r > 0.5$ maka
 - a. Baris sementara = [Depan Belakang]
 - b. Misalkan baris sementara memiliki M elemen, bangkitkan bilangan random r dengan nilai antara 1 sampai M
 - c. DepanBaru = Baris sementara (1:r)
 - d. TengahBaru = Tengah
 - e. BelakangBaru = Sementara (r+1:M)

Misalkan asumsi rute terpendek sementara adalah

$L_{min} [0-1-2-3-4-5-0]$ dengan energi = $57 + 80 + 17 + 12 + 3.8 + 6.6 = 176.4 \text{ km}$

Parameter:

$NC = 5$ (titik yang dikunjungi)

$M_{Iterasi} = 2$, $MSukses = 5$, Konstanta penurunan suhu = 0.95

$MaxIterasi = M_{Iterasi} * NC = 10$

$MaxSukses = MSukses * NC = 25$

Evaluasi solusi di T_0 ;

Misalkan $LSukses$ terlama yang di dapat sampai iterasi ke M_{Temp} adalah $[0-2-1-4-5-3-0]$ dengan jumlah energi 196.8 km

$L_{Max} = 196.8$, $T_0 = 2 * L_{Max} = 393.6$

Untuk suhu $T = 393.6$ dengan parameter awal [1-2-3-4-5]

Iterasi 1:

Bilangan random $N1 = 3, N2 = 4$

Baris depan = [1,2]

Baris tengah = [3,4]

Baris belakang = [5]

Bilangan random $r = 0.05$ (kurang dari 0.5), maka

Baris depan baru = [1,2]

Baris tengah baru = [4,3]

Baris belakang baru = [5]

Diperoleh jalur baru: [0-1-2-4-3-5-0]

Energi jalur baru = 188

188 km > 176.4 km, maka **jalur baru ditolak**

Iterasi 2:

Bilangan random $N1 = 4, N2 = 5$

Baris depan = [1,2,3]

Baris tengah = [4,5]

Baris belakang = []

Bilangan random $r = 0.27$ (kurang dari 0.5), maka

Baris depan baru = [1,2,3]

Baris tengah baru = [5,4]

Baris belakang baru = []

Diperoleh jalur baru: [0-1-2-3-5-4-0]

Energi jalur baru = 170

170 km < 176.4 km, maka **jalur baru diterima**

Iterasi 3:

Bilangan random $N1 = 1, N2 = 2$

Baris depan = []

Baris tengah = [1,2]

Baris belakang = [3,4,5]

Bilangan random $r = 0.4$ (kurang dari 0.5), maka

Baris depan baru = []

Baris tengah baru = [2,1]

Baris belakang baru = [3,4,5]

Diperoleh jalur baru: [0-2-1-3-4-5-0]

Energi jalur baru = 194.4

194.4 km > 176.4 km, maka **jalur baru ditolak**

Iterasi 4:

Bilangan random $N1 = 1, N2 = 2$

Baris depan = []

Baris tengah = [1,2]

Baris belakang = [3,4,5]

Bilangan random $r = 0.67$ (lebih dari 0.5), maka buat dulu baris sementara

Baris sementara = [baris depan] + [baris belakang]

Baris sementara = [3,4,5]

$M = 3, r = 1$, sehingga

Baris depan baru = [3]

Baris tengah baru = [1,2]

Baris belakang baru = [4,5]

Diperoleh jalur baru: [0-3-1-2-4-5-0]

Energi jalur baru = 193.4

193.4 km > 176.4 km, maka **jalur baru ditolak**

Iterasi 5:

Bilangan random $N1 = 2, N2 = 4$

Baris depan = [1]

Baris tengah = [2,3,4]

Baris belakang = [5]

Bilangan random $r = 0.68$ (lebih dari 0.5), maka buat dulu baris sementara

Baris sementara = [1,5]

$M = 2, r = 2$

Baris depan baru = [1,5]

Baris tengah baru = [2,3,4]
 Baris belakang baru = []
 Diperoleh jalur baru: [0-1-5-2-3-4-0]
 Energi jalur baru = 174
 174 km < 176.4 km, maka **jalur baru diterima**

Iterasi 6:

Bilangan random $N1 = 2, N2 = 3$
 Baris depan = [1]
 Baris tengah = [2,3]
 Baris belakang = [4,5]
 Bilangan random $r = 0.17$ (kurang dari 0.5), maka
 Baris depan baru = [1]
 Baris tengah baru = [3,2]
 Baris belakang baru = [4,5]
 Diperoleh jalur baru: [0-1-3-2-4-5-0]
 Energi jalur baru = 170.4
 170.4 km < 176.4 km, maka **jalur baru diterima**

Iterasi 7:

Bilangan random $N1 = 1, N2 = 3$
 Baris tengah = [1,2,3]
 Baris depan = []
 Baris belakang = [4,5]
 Bilangan random $r = 0.58$ (kurang dari 0.5), maka buat dulu baris sementara
 Baris sementara = [4,5]
 $M = 2, r = 1$
 Baris depan baru = [4]
 Baris tengah baru = [1,2,3]
 Baris belakang baru = [5]
 Diperoleh jalur baru: [0-4-1-2-3-5-0]
 Energi jalur baru = 170.1
 170.1 km < 176.4 km, maka **jalur baru diterima**

Iterasi 8:

Bilangan random $N1 = 1, N2 = 5$
 Baris depan = []
 Baris tengah = [1,2,3,4,5]
 Baris belakang = []
 Bilangan random $r = 0.09$ (kurang dari 0.5), maka
 Baris depan baru = []
 Baris tengah baru = [5,4,3,2,1]
 Baris belakang baru = []
 Diperoleh jalur baru: [0-5-4-3-2-1-0]
 Energi jalur baru = 172.7
 172.7 km < 176.4 km, maka **jalur baru diterima**

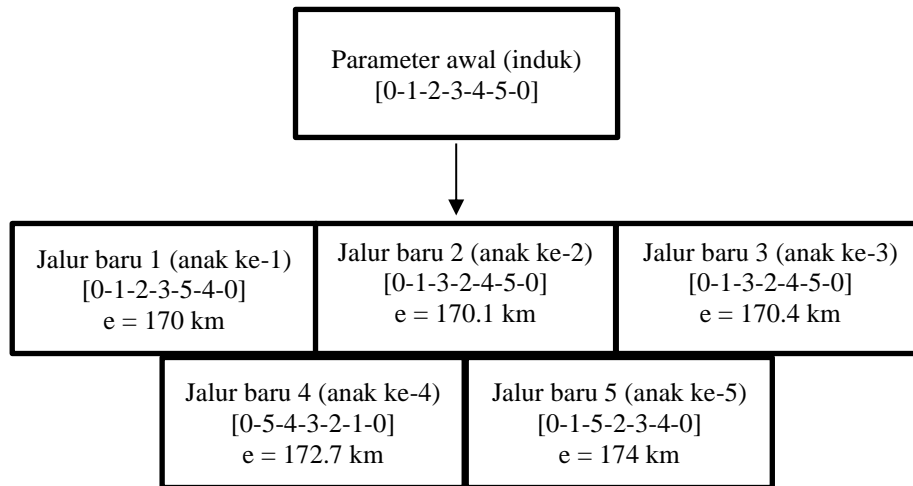
Iterasi 9:

Bilangan random $N1 = 1, N2 = 3$
 Baris tengah = [1,2,3]
 Baris depan = []
 Baris belakang = [4,5]
 Bilangan random $r = 0.71$ (lebih dari 0.5), maka buat dulu baris sementara
 Baris sementara = [4,5]
 $M = 2, r = 2$
 Baris depan baru = [4,5]
 Baris tengah baru = [1,2,3]
 Baris belakang baru = []
 Diperoleh jalur baru: [0-4-5-1-2-3-0]
 Energi jalur baru = 179.6
 179.6 km > 176.4 km, maka **jalur baru ditolak**

Iterasi 10:

Bilangan random $N1 = 3, N2 = 5$
 Baris tengah = [3,4,5]
 Baris depan = [1,2]

Baris belakang = []
 Bilangan random $r = 0.15$ (kurang dari 0.5), maka
 Baris depan baru = [1,2]
 Baris tengah baru = [5,4,3]
 Baris belakang baru = []
 Diperoleh jalur baru: [0-1-2-5-4-3-0]
 Energi jalur baru = 194.7
 $194.7 \text{ km} > 176.4 \text{ km}$, maka **jalur baru ditolak**



Berdasarkan hasil evaluasi solusi di T_0 , diperoleh 5 jalur baru yang memiliki jumlah energi lebih kecil dibandingkan dengan jumlah energi di parameter induknya. Selanjutnya masing-masing jalur baru dievaluasi dengan proses yang sama seperti parameter induk, pada keadaan suhu yang baru T_1 .

$$T_1 = 393.6 \times 0.95 = 373.92$$

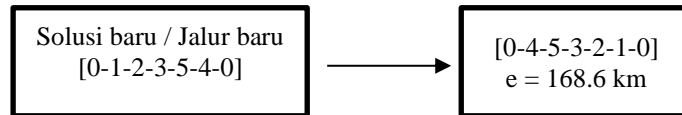
Lmin (jalur baru 1) = [0-1-2-3-5-4-0] dengan energi = 170 km

Evaluasi solusi di $T_1 = 373.92$

Tabel 4. Hasil Pengujian SA pada T_1 Jalur 1

Iterasi	N1	N2	Depan	Tengah	Belakang	r	Jalur Baru	Energy
1	1	4		1,2,3,5	4	0.36	0-5-3-2-1-4-0	179 km > 170 km Maka solusi rute ditolak
2	2	3	1	2,3	5,4	0.14	0-1-3-2-5-4-0	170.7 km > 170 km Maka solusi rute ditolak
3	1	4		1,2,3,5	4	0.66	0-4-1-2-3-5-0	176.7 km > 170 km Maka solusi rute ditolak
4	2	3	1	2,3	5,4	0.62	0-1-5-2-3-4-0	174 km > 170 km Maka solusi rute ditolak
5	1	3		1,2,3	5,4	0.68	0-5-4-1-2-3-0	183.7 km > 170 km Maka solusi rute ditolak
6	1	5		1,2,3,5,4		0.49	0-4-5-3-2-1-0	168.6 km < 170 km Maka solusi rute diterima
7	2	5	1	2,3,5,4		0.46	0-1-4-5-3-2-0	173.8km > 170 km Maka solusi rute ditolak

8	3	5	1,2	3,5,4		0.96	0-1-3-5-4-2-0	183 km > 170 km Maka solusi rute ditolak
9	3	4	1,2	3,5	4	0.97	0-1-2-4-5-3-0	190.8km > 170 km Maka solusi rute ditolak
10	3	4	1,2	3,5	4	0.17	0-1-2-5-3-4-0	191 km > 170 km Maka solusi rute ditolak



Evaluasi solusi di T_1 , hanya memperoleh 1 jalur baru.

$$T_2 = 373.92 \times 0.95 = 355.22$$

$L_{min} = [0-4-5-3-2-1-0]$ dengan energi = **168.6 km**

Evaluasi solusi di $T_2 = 355.22$

Tabel 5. Hasil Pengujian SA pada T_2

Iterasi	N1	N2	Depan	Tengah	Belakang	r	Jalur Baru	Energy
1	3	4	4,5	3,2	1	0.95	0-4-3-2-5-1-0	169.8 km > 168.6 km Maka solusi rute ditolak
2	4	5	4,5,3	2,1		0.27	0-4-5-3-1-2-0	191.6 km > 168.6 km Maka solusi rute ditolak
3	1	2		4,5	3,2,1	0.4	0-5-4-3-2-1-0	172.7 km > 168.6 km Maka solusi rute ditolak
4	1	2		4,5	3,2,1	0.67	0-3-2-4-5-1-0	168.8 km > 168.6 km Maka solusi rute ditolak
5	2	4	4	5,3,2	1	0.68	0-4-5-3-2-1-0	168.6 km = 168.6 km Maka solusi rute ditolak
6	2	3	4	5,3	2,1	0.17	0-4-3-5-2-1-0	186.1 km > 168.6 km Maka solusi rute ditolak
7	1	3		4,5,3	2,1	0.58	0-2-1-4-5-3-0	196.8 km > 168.6 km Maka solusi rute ditolak
8	1	3		4,5,3	2,1	0.09	0-3-5-4-2-1-0	183 km > 168.6 km Maka solusi rute ditolak
9	1	3		4,5,3	2,1	0.61	0-2-4-5-3-1-0	180.8 km > 168.6 km Maka solusi rute ditolak
10	3	5	4,5	3,2,1		0.15	0-4-5-1-2-3-0	179.6 km > 168.6 km Maka solusi rute ditolak

Setelah evaluasi solusi di T_2 tidak ditemukan lagi rute yang lebih pendek, maka penelusuran dihentikan. Rute **[0-4-5-3-2-1-0]** dengan panjang **168.6 km** adalah hasil optimum dari semua permutasi kemungkinan. Jika

dilihat dari semua kemungkinan rute, algoritma *Simulated Annealing* untuk kasus ini telah memberikan solusi rute yang optimal.

4.2 Analisis Hasil Pengujian

Hasil pengujian dengan 10 (sepuluh) kali iterasi menggunakan algoritma simulasi *annealing* pada 3 (tiga) suhu berbeda menghasilkan solusi rute terbaik [0-4-5-3-2-1-0] dengan jumlah energi sebesar 168.6 km. Dengan penerapan metode simulasi *annealing*, proses pencarian rute terpendek menjadi lebih singkat dibandingkan dengan beberapa metode lain misalnya metode *brute force* (hasil evaluasi *brute force* terlampir).

Melalui simulasi *annealing*, setelah beberapa kali iterasi apabila state atau rute yang dihitung selanjutnya terus menghasilkan energi dalam jumlah besar maka iterasi dapat dihentikan sehingga dipilih state dengan minimum energi untuk kemudian disarankan kepada user (wisatawan). Sedangkan dalam penggunaan *brute force* tentunya akan dicari semua kemungkinan jarak antar titik untuk selanjutnya dihitung dan ditentukan rute yang paling optimal yaitu rute dengan jarak paling minimum.

Proses *brute force* dapat membutuhkan waktu yang lebih lama karena yang dicari adalah semua kemungkinan rute dan dihitung semua kemungkinan jaraknya, namun hasil yang disarankan dapat dipastikan sudah optimal. Melalui evaluasi dengan *brute force* ditemukan 5 rekomendasi rute dengan nilai panjang rute yang paling kecil, berkisar antara 166.7 km – 168.6 km. Simulasi *annealing* memiliki waktu eksekusi lebih sedikit jika dibandingkan dengan *brute force* dengan hasil yang disarankan bersifat sub-optimal, baik namun masih ada kemungkinan rute yang lebih baik akan ditemukan apabila iterasi dilakukan hingga seluruh kemungkinan jarak dihitung.

5. Kesimpulan

Berdasarkan hasil pengujian dan analisis maka penulis dapat menyimpulkan bahwa *Simulated Annealing* bekerja baik untuk pencarian rute berdasarkan kasus TSP. Penulis menggunakan algoritma *Brute Force* sebagai algoritma pembanding dalam menganalisis performansi dari algoritma *Simulated Annealing*. *Simulated Annealing* memiliki waktu eksekusi lebih sedikit jika dibandingkan dengan *Brute Force* dengan hasil yang disarankan bersifat sub-optimal. Setelah beberapa kali iterasi apabila state atau rute yang dihitung selanjutnya terus menghasilkan energi dalam jumlah besar maka iterasi dapat dihentikan sehingga dipilih state dengan minimum energi untuk kemudian disarankan kepada user (wisatawan). Hasil yang disarankan kepada wisatawan adalah rute yang baik, namun masih ada kemungkinan rute yang lebih baik akan ditemukan apabila iterasi dilakukan hingga seluruh kemungkinan jarak dihitung.

Adapun sebagai *future works* penulis berharap dapat melakukan perbandingan antara metode *Simulated Annealing* terhadap algoritma – algoritma lainnya dalam menyelesaikan persoalan TSP, sehingga dapat diketahui algoritma mana yang paling optimal dan memiliki tingkat kecepatan baik dalam memproses TSP.

Daftar Pustaka

- [1] Agusta, Yudi. 2010. *Simulasi Annealing*. [PDF] Denpasar : Data Mining and Clustering Research.
- [2] Arjani, Jayanti. 2009. *Implementasi Algoritma Genetika untuk Travelling Salesman Problem (TSP)*. Skripsi. Universitas Sumatera Utara : Medan
- [3] Chibante R, Araujo A, and Carvalho A. 2010. *Simulated Annealing Theory with Applications*. Croatia: Sciyo.
- [4] Jannach, Dietmar. 2014. *Recommender System An Introduction*. [PDF] Germany : Dietmar Jannach, TU Dortmund, Germany.
- [5] Kirkpatrick S, Gelatt C. D, Vecchi M. P. 1983. *Optimization by Simulated Annealing*, *Science*, Vol 220, Number 4589, Pages 671 – 680.
- [6] Larasati, Tuti. 2011. *Aplikasi Annealing untuk Menyelesaikan Traveling Salesmen Problem*. Universitas Sumatera Utara, Program Studi Matematika, Paper. [PDF] Medan: USU.
- [7] M Ikhsan, Rahma A, Wibisono L. 2015. *Traveling Salesman Problem*. Institut Teknologi Bandung, Program Studi Teknik Informatika, Paper. [PDF] Bandung: ITB.
- [8] Mayasari, Eka. 2012. *Penyelesaian Multiple TSP Dengan Simulated Annealing*. Universitas Jember, Program Studi Matematika, Jurnal. [online]
- [9] Nainggolan Leonardo. 2016. *Penyelesaian TSP Dengan Simulated Annealing, Studi Kasus: Technical Support BTSSOFT*. Universitas Sumatera Utara, Program Studi Ilmu Komputer, Jurnal. [online]
- [10] Pangaribuan, Goltiandy. 2009. *Studi Travelling Salesman Problem (TSP) dengan menggunakan Program Dinamik*. Medan : Universitas Sumatera Utara
- [11] 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.
- [12] Rahma Amin, Aulia & Mukhamad Ikhsan & Lastika Wibisono. *Travelling Salesman Problem*. Makalah. Bandung: Institut Teknologi Bandung
- [13] Simha, Rahul. The Traveling Salesman Problem (TSP). (Online) <http://www.seas.gwu.edu/~simhawe/champalg/tsp/tsp.html>

- [14] Suryawan B, Yuhana Umi, Fajrin N. 2013. *Pencarian Objek Wisata Berbasis Semantik*. Institut Teknologi Sepuluh November, Program Studi Teknik Informatika, Paper. [PDF] Surabaya: ITS.
- [15] Sarno R, Anistyasari Y, and Rahimi F. 2012. *Semantic Search*. Andi Offset.
- [16] Samana E, Prihandono B, Noviani E. 2015. *Aplikasi Simulated Annealing untuk Menyelesaikan Travelling Salesman Problem*. [PDF] Yogyakarta: Bimaster.
- [17] Sanjaya, Donny. 2014. *Implementasi Mobile Tracking dengan Metode Ant Colony Optimization dan Google API*. Skripsi. Medan: Universitas Sumatera Utara
- [18] Wahyu, Andi & Allen F. Aritonang (2007). *Aplikasi Desktop Pencarian Rute Jalan dengan Algoritma Simulated Annealing*. Jurnal. Universitas Kristen Maranatha: Bandung

