

JALUR OPTIMUM PENGIRIMAN BARANG DI KANTOR POS GRESIK DENGAN METODE *SIMULATED ANNEALING*

Santhi Yulia Retnani, Moch. Kautsar Sophan, Vivi Tri Widyaningrum

Prodi Teknik Informatika, Fakultas Teknik, Universitas Trunojoyo

Jl. Raya Telang, PO BOX 2 Kamal, Bangkalan - 69162

E-mail : santhi.maori@gmail.com, kautsar@if.trunojoyo.ac.id, vivie_7812@yahoo.com

ABSTRAK

Kantor pos memberikan suatu jasa pelayanan yaitu jasa pelayanan pengiriman barang. Dalam hal ini petugas pos harus mengirimkan banyak barang ke alamat-alamat pelanggan. Namun banyaknya jalan raya pada jaman sekarang terkadang menyulitkan petugas pos untuk memilih jalan mana yang dalam segi waktu tempuhnya lebih optimal. Oleh karena itu diperlukan suatu sistem informasi yang dapat menemukan jalur optimum untuk menghemat waktu dengan memperhitungkan antara panjang jalan dan kecepatan berkendara pada tiap ruas jalan. Sistem ini nantinya dibuat dengan menggunakan metode Simulated Annealing dan berbasis Web. Dengan memanfaatkan web untuk pengaksesannya, sistem ini mampu memberikan informasi yang cukup berguna bagi petugas pos mengenai jalur optimum dengan nilai total waktu tempuh paling optimal sehingga dalam mengantarkan barang dari kantor pos menuju alamat tujuan pelanggannya dapat menghemat waktu. Jalur optimum divisualisasikan dalam bentuk graf. Dari hasil uji coba yang dilakukan dapat diambil kesimpulan bahwa metode Simulated Annealing dapat digunakan untuk proses pencarian jalur optimal dengan alamat tujuan lebih dari satu.

Kata Kunci : Simulated Annealing, Pengiriman Barang, Jalur Optimum, Kantor Pos, Graf

ABSTRACT

Post offices provide a service that is service delivery. In this case the postman has to send stuff to the customer addresses. But many roads today sometimes make it difficult for the postman to choose which path in terms of travel time is more optimal. Therefore we need an information system that can find the optimal path to save time by measuring the length of the road and driving speed on each road segment. This system will be made by using the method of Simulated Annealing and Web-based. By making use of the web for accessing it, the system is able to provide information that is useful for the postman about optimum path with a total value of travel time so that the most optimal from the post office to deliver the goods from the post office to the customers address can save time. The optimum path is visualized in the form of a graph. From the results of experiments performed can be concluded that the method of Simulated Annealing can be used to search the optimal path to the destination address more than one.

Keyword: *Simulated Annealing, Shipping, Optimum Path, Post Office, Graph*

PENDAHULUAN

Seiring berkembangnya kemajuan teknologi sistem informasi membuat pelayanan jasa pengiriman barang yang dilakukan kantor pos kini lebih sistematis. Kantor pos tidak hanya mengantarkan surat tetapi juga mengantarkan barang paket dari pengirim yang ditujukan ke suatu alamat tertentu. Dalam hal ini petugas pos harus mengirimkan banyak barang ke alamat-alamat pelanggan dengan pertimbangan efisiensi waktu.

Sedangkan pada jaman sekarang, jumlah jalan raya yang sangat banyak terkadang menyulitkan petugas pos untuk dapat memilih jalur yang cepat ke alamat tujuan pelanggan. Biasanya jalur yang ditempuh hanyalah jalur yang dihafal dan sering dilalui serta dianggap terpendek, padahal belum tentu jalur tersebut lebih optimal dari segi waktu tempuhnya. Karena terkadang jalur yang pendek itu mempunyai tingkat kemacetan yang lebih tinggi sehingga waktu tempuh lebih lama dibanding jalur yang sedikit lebih panjang tetapi tingkat kemacetannya rendah. Untuk itu diperlukan sebuah sistem yang memanfaatkan teknologi informasi guna melakukan proses pencarian jalur optimum secara otomatis.

Sistem ini nantinya akan dijadikan suatu aplikasi baru yang dapat digunakan oleh kantor pos cabang Gresik dalam memberikan layanan jasa pengiriman barang. Penerapan metode *Simulated Annealing* pada sistem ini diharapkan mampu mendapatkan jalur optimum dari kantor pos ke alamat tujuan pelanggan dengan memperhitungkan antara panjang jalan dan kecepatan berkendara. Gambaran jalur optimumnya divisualisasikan dalam bentuk graf.

Dari penelitian sebelumnya, yang mengenai optimasi Distribusi Barang Berdasarkan Rute dan Daya Tampung Menggunakan Metode *Simulated Annealing* oleh Susilo Dwi J, Entin Martiana K, Arna Fariza, dan Ira Prasetyaningrum. Peneliti membuat aplikasi *Vehicle Routing Problem* (VRP) berbasis dektop dengan metode

Simulated Annealing. Dalam hal ini peneliti hanya memperhitungkan jarak terpendek tanpa memperhitungkan nilai waktu tempuhnya dan inputnya beberapa (lebih dari satu) alamat tujuan[1].

Sedangkan sistem yang akan dibuat pada penelitian ini adalah membuat aplikasi pencarian jalur optimal untuk pengiriman barang di kantor pos Gresik berbasis web dengan metode *Simulated Annealing*. Dalam hal ini peneliti tidak hanya memperhitungkan jarak melainkan juga memperhitungkan kecepatan rata-rata tiap ruas jalan sehingga diperoleh nilai waktu tempuhnya dan inputnya beberapa (lebih dari satu) alamat tujuan.

Agar dalam pembahasan penelitian ini tidak terlalu melebar maka diberikan sebuah batasan masalah yang meliputi sistem informasi ini dibangun untuk dapat menemukan jalur optimum, sistem informasi ini dibuat dengan metode *Simulated Annealing*, menggunakan PHP dan gambaran jalurnya divisualisasikan dalam bentuk graf, sistem informasi ini tidak dapat menemukan alamat secara detail, hanya dapat melakukan pendekatan alamat yang dituju, data yang digunakan berdasarkan data sebelumnya bukan data peramalan, wilayah pengiriman barang pada penelitian ini adalah kota Gresik, kecepatan rata-rata berkendara pada tiap-tiap ruas jalan tidak berubah-ubah (konstan) sesuai dengan data jalan yang ada, hasil pencarian jalur dihitung dari titik awal sampai ke tujuan akhir saja, dan tidak kembali ke titik awal.

METODE

Simulated Annealing (SA) adalah metode pencarian lokal yang acak, di mana solusi sementara dimodifikasi sehingga mengarah pada hasil yang lebih baik dengan beberapa kemungkinan. Mekanisme ini dapat mengantisipasi *local optima* yang buruk[2].

Metropolis menciptakan sebuah algoritma yang juga dikenal sebagai aturan metropolis dari probabilitas untuk

mensimulasikan pendinginan logam melalui serangkaian gerakan. Pada tiap pergerakan, sistem memiliki kemungkinan untuk merubah konfigurasi saat ini menjadi lebih buruk. Peluang perubahan konfigurasi dapat dihitung dari[3] :

$$P(\Delta WT, T) = \begin{cases} 1, & \text{Bila } \Delta WT \leq 0 \\ \text{Exp}\left(-\frac{\Delta WT}{T}\right), & \text{Bila } \Delta WT > 0 \end{cases} \quad (1)$$

Dimana :

ΔWT : Perubahan energi/konfigurasi

T : Temperatur

Persamaan (1) ini juga disebut kriteria metropolis. Kemampuan untuk mengubah konfigurasi membuat *Simulated Annealing* memungkinkan untuk melompat keluar dari lokal maksima dan minima dimana sebagian besar algoritma lain terjebak di dalamnya.

Pada algoritma *Simulated Annealing* pertama perlu ditentukan awal dan akhir dari suhu. Hal ini penting karena suhu (T) akan digunakan dalam persamaan probabilitas. Persamaan probabilitas, seperti yang didefinisikan oleh metropolis pada persamaan(1), ΔWT didapat dari[3] :

$$\Delta WT = \text{tot } WT_u - \text{tot } WT_a \quad (2)$$

Dimana :

tot WT_u : Konfigurasi baru (setelah *update state*)

tot WT_a : Konfigurasi saat ini (sebelum *update state*)

Simulated Annealing dapat diterapkan dalam kasus TSP. Algoritma akan dimulai pada suhu acak. Kemudian urutan bergerak akan diambil selama suhu tersebut. Dalam tiap gerakan akan diciptakan konfigurasi baru dan kemudian akan diterima atau ditolak sesuai dengan perhitungan persamaan probabilitas(1). Ketika sebuah gerakan diterima suhu akan berubah. Proses gerakan akan dilakukan secara berulang-ulang hingga suhu mencapai suhu akhir. Semakin besar tinggi suhu maka akan

semakin besar kemungkinan terjadi perpindahan yang kurang optimal[3].

Konfigurasi atau Energi dalam *Simulated Annealing* merupakan biaya atau tujuan dari pencarian solusi permasalahan optimasi. Dalam TSP energi didapat dari jarak yang ditempuh saat melewati tiap node. Waktu tempuh dapat dicari dengan menggunakan persamaan[4] :

$$WT_i = \sum_{i=1}^n [(d_i / v_i) + 1 \text{ menit}] \quad (3)$$

Dimana :

WT_i : Fungsi waktu tempuh setelah iterasi

d_i : Jarak antara node(i) dan node(i+1)

v_i : Kecepatan rata-rata antara node(i) dan node(i+1)

n : Jumlah node

1 menit : Waktu penurunan (*dropping*) barang

Algoritma *Simulated Annealing* juga bergantung pada nilai temperatur suhu dan nilai temperatur suhu akan mengalami penurunan pada tiap iterasi. Penurunan suhu tidak bisa dilakukan dengan sembarangan. Terdapat fungsi *cooling schedule* (T) yang digunakan untuk mencari nilai suhu yang telah turun. Persamaannya adalah[3] :

$$T_i = T_0 \left(\frac{T_n}{T_0}\right)^{\frac{i}{N}} \quad (4)$$

Dimana :

T_i : Temperatur *cooling schedule* ke-I

T_0 : Temperatur awal

T_n : Temperatur *cooling schedule*

N : Jumlah iterasi

i : iterasi (nilai i : 1,2,3,...)

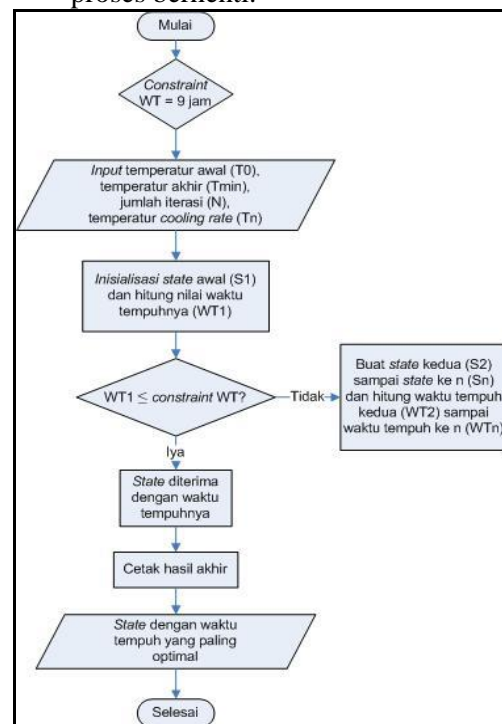
Kriteria penghentian proses simulasi. Ada beberapa metode yang biasa digunakan untuk mengontrol penghentian algoritma, yaitu dilihat dari[4] :

- a. Maksimum jumlah iterasi.
- b. Nilai minimum temperature.
- c. Nilai minimum fungsi objektif.
- d. Nilai minimum dari tingkat penerimaan.

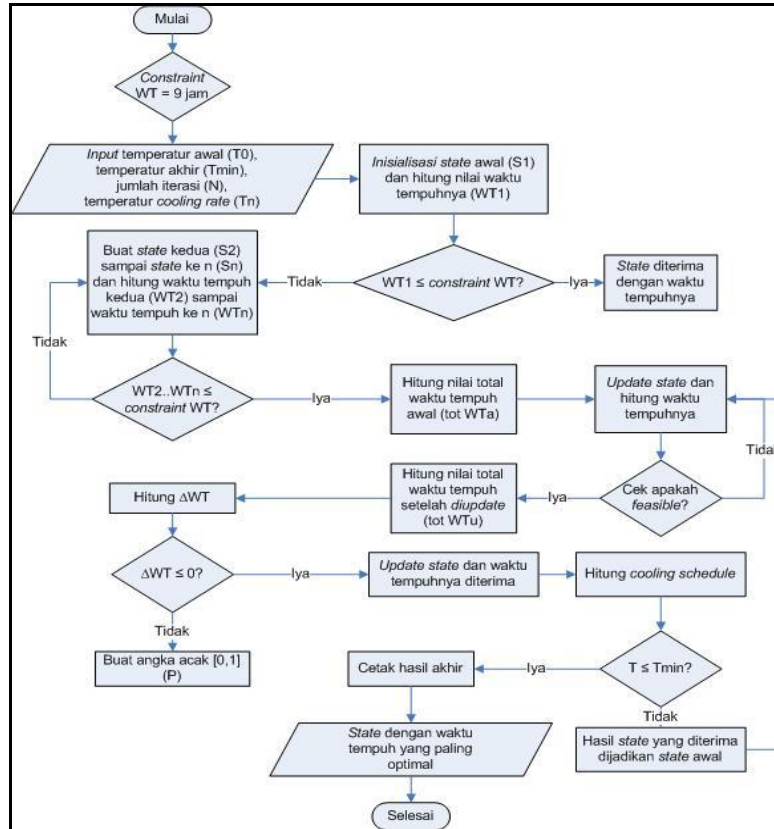
Berikut adalah beberapa langkah proses perhitungan langkah proses perhitungan algoritma *Simulated Annealing* sebelum mencapai solusi yang optimal adalah :

1. *Konstant* \rightarrow *Constraint* waktu tempuh.
2. Input temperatur awal (T_0), temperatur akhir (T_{min}), iterasi (N) dan *cooling rate* (T_n).
3. *Inisialisasi state* awal (S_1) dan hitung waktu tempuh awal (WT_1) yang dihasilkan pada kondisi awal. Dalam hal ini yang dimaksud *state* adalah hasil rute yang akan dilewati. *State* ini dibentuk dengan cara memilih koneksi jalan secara *random*, namun jika dalam koneksi tersebut terdapat jalan yang menjadi tujuan maka jalan tersebutlah yang akan langsung dipilih.
4. Cek apakah nilai $WT_1 \leq \text{constraint}$ WT. Jika benar maka *state* diterima dan langsung dijadikan hasil akhir (*flowchartnya* seperti gambar 1), namun jika tidak maka pecah *state* tersebut ($S_1 \dots S_n$) sampai nilai waktu tempuh tidak melebihi *constraint* (*flowchartnya* seperti gambar 2).
5. Hitung total waktu tempuh awal (Tot WTa).
6. Lakukan *update state*. Yang dimaksud *update state* ini adalah menukarkan salah satu *node* yang ada pada *state* awal (S_1) dengan *node* yang ada pada *state* kedua (S_2) atau *state* ke- n (S_n).
7. Cek apakah *update statenya feasible*. Dalam hal ini yang dimaksud *feasible* adalah :
 - i. Apakah *node* yang ditukarkan itu saling terkoneksi?
 - ii. Apakah hasil waktu tempuhnya tidak melebihi *constraint*?
 - iii. Apakah jalan yang menjadi tujuan itu tidak hilang?
 Jika iya maka *state* diterima dan hitung total waktu tempuh setelah *diupdate*, namun jika tidak maka *update state* kembali sampai memenuhi syarat *feasiblenya*.

8. Hitung total waktu tempuh *update* (Tot WTu).
9. Cek apakah $\Delta WT \leq 0$. Jika iya maka *update state* dan total waktu tempuh *update* diterima, jika tidak maka bangkitkan bilangan *random* berdistribusi *uniform* $[0,1]$, dilambangkan dengan (P).
10. Turunkan temperatur (T) dengan fungsi *cooling schedule*.
11. Cek apakah hasil $T \leq T_{min}$. Jika benar maka proses berhenti dan cetak *state* akhir dengan hasil waktu tempuh paling optimal, namun jika tidak ulangi langkah 5 yaitu buat *state* kedua (S_2) dan hitung nilai waktu tempuhnya (WT_2) kembali sampai mencapai kriteria (dengan catatan hasil *state* yang diterima akan dijadikan sebagai *state* awal (S_1) beserta nilai waktu tempuhnya (WT_1) dan hasil T (*Cooling Schedule*) dijadikan sebagai T_0 , dan begitu seterusnya).
12. Jika telah mencapai kriteria maka proses berhenti.



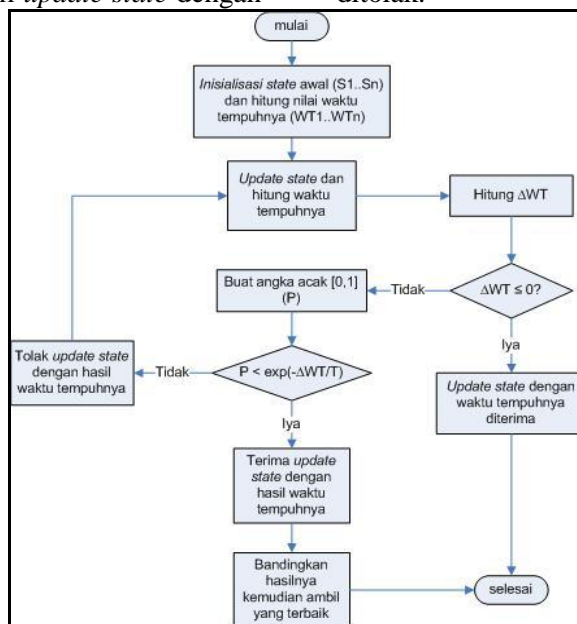
Gambar 1. *Flowchart* metode *simulated annealing* ketika *state* awalnya tidak melebihi *constraint*



Gambar 2. Flowchart metode *simulated annealing* ketika *state* awalnya melebihi *constraint*

Jika hasil $\Delta WT \leq 0$, maka bangkitkan bilangan *random* berdistribusi *uniform* $[0,1]$, dilambangkan dengan (P). Setelah itu uji kriteria, bila $p < \exp(-\Delta WT/T)$ maka hasil *update state* dengan

hasil waktu tempuhnya dan bandingkan hasilnya kemudian ambil yang terbaik, namun jika tidak maka *update state* dengan hasil waktu tempuhnya ini ditolak.



Gambar 3. Flowchart pembentukan angka acak (*random*)

HASIL DAN PEMBAHASAN

Pada sistem ini, jika pengguna ingin melakukan proses pencarian jalur optimal baru maka pengguna harus memilih tanggal pengiriman (sesuai dengan data pengiriman yang ada pada sistem), memilih *start node* (*start nodenya* tidak hanya dari pos saja, tetapi bisa dipilih sesuai yang diinginkan), memasukkan nilai *constraint time* (jam), suhu awal (T_0), suhu akhir (T_{min}), jumlah iterasi (N), dan nilai *cooling rate* (T_n). Parameter-parameter tersebut nilainya dapat ditentukan sendiri oleh pengguna.

Ketentuan-ketentuan dalam menggunakan parameter-parameter ini adalah :

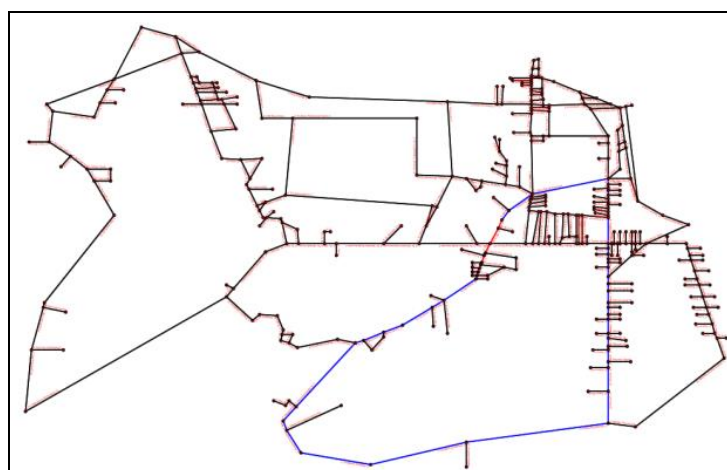
- Nilai *constraint*, T_0 , T_{min} , iterasi, dan T_n yang dimasukkan boleh bernilai berapa saja yang penting tidak bernilai negatif.

- Nilai T_n yang dimasukkan tidak boleh lebih besar dari nilai T_{min} .
- Nilai T_{min} yang dimasukkan harus lebih kecil dari T_0 .
- Nilai *constraint* yang dimasukkan sebaiknya jangan terlalu besar agar mendapat hasil yang optimal (maksimal 9 jam).

Gambar 4. Form proses pencarian jalur optimal

Tanggal Pengiriman	Rute Pengiriman	Jarak (KM)	Waktu (Jam)
2013-07-07	Jl. DR. Sutomo 2 - Jl. DR. Sutomo 1 - Jl. Jaksa Agung Suprpto - Jl. Panglima Sudirman 12 - Jl. Panglima Sudirman 11 - Jl. Panglima Sudirman 10 - Jl. Panglima Sudirman 9 - Jl. Panglima Sudirman 8 - Jl. Panglima Sudirman 7 - Jl. Panglima Sudirman 6 - Jl. Panglima Sudirman 5 - Jl. Panglima Sudirman 4 - Jl. Panglima Sudirman 3 - Jl. Panglima Sudirman 2 - Jl. Panglima Sudirman 1 - Jl. Veteran 1 - Jl. Veteran 2 - Jl. Veteran 3 - Jl. Veteran 4 - Jl. Veteran 5 - Jl. Veteran 6 - Jl. Veteran 7 - Jl. Veteran 8 - Jl. Veteran 9 - Jl. Veteran 10 - Jl. Veteran 11 - Jl. Veteran 12 - Jl. Veteran 13 - Jl. Veteran 14 - Jl. Veteran 15 - Jl. Veteran 16 - Jl. Veteran 17 - Jl. Veteran 18 - Jl. Veteran 19 - Jl. Veteran 20 - Jl. Mayjend Sungkono 1 - Jl. Mayjend Sungkono 2 - Jl. Mayjend Sungkono 3 - Jl. Mayjend Sungkono 4 - Jl. Mayjend Sungkono 5 - Jl. Mayjend Sungkono 6 - Jl. Mayjend Sungkono 7 - Jl. Sunan Prapen 3 - Jl. Sunan Prapen 2 - Jl. Sunan Prapen 1 - Jl. Sunan Giri 11 - Jl. Sunan Giri 10 - Jl. Sunan Giri 9 - Jl. Sunan Giri 8 - Jl. Sunan Giri 7	7.640	0.989
2013-07-07	Jl. DR. Sutomo 3 - Jl. DR. Sutomo 4 - Jl. Sunan Giri 1 - Jl. Sunan Giri 2 - Jl. Sunan Giri 3 - Jl. Sunan Giri 4 - Jl. Sunan Giri 5 - Jl. Sunan Giri 6	0.580	0.146

Gambar 5. Form hasil pencarian jalur optimal



Gambar 6. Form visualisai peta hasil pencarian jalur optimal

Dalam penelitian ini, peneliti melakukan tiga kali uji coba dengan menggunakan parameter-parameter yang

telah dijelaskan pada sebelumnya. Uji coba tersebut jika dimasukkan kedalam tabel adalah sebagai berikut :

Tabel 1. Data uji coba *simulated annealing*

Parameter	Uji coba ke-1	Uji coba ke-2	Uji coba ke-3
<i>Constraint</i>	0.5	0.5	1
Suhu Awal	10	1000	10
Suhu Akhir	2	1	2
Jumlah Iterasi	10	100	10
<i>Cooling Rate</i>	1	1	1
Jumlah Tujuan	44 Tujuan	44 Tujuan	44 Tujuan
Hasil Tujuan	3 Jalur	2 Jalur	2 Jalur
Lama Proses	450 detik	1800 detik	2000 detik

Dari beberapa uji coba yang telah dilakukan dapat diambil kesimpulan bahwa metode *Simulated Annealing* dapat digunakan untuk menyelesaikan masalah pencarian jalur optimal untuk tujuan lebih dari satu seperti pada uji coba di atas. Namun proses pencarian jalur optimal menggunakan metode *Simulated Annealing* bergantung pada beberapa parameter antara lain nilai *constraint*, nilai suhu awal (T_0), nilai suhu akhir (T_{min}), jumlah iterasi (N), dan nilai *cooling rate* (T_n).

Penentuan nilai suhu awal (T_0), nilai suhu akhir (T_{min}), jumlah iterasi (N), dan nilai *cooling rate* (T_n) yang digunakan mempengaruhi lama proses saat dijalankan. Semakin tinggi nilai suhu awal maka proses iterasi pun semakin banyak sehingga mengakibatkan proses semakin lama namun hasilnya pun semakin optimal seperti terlihat pada hasil uji coba ke-2 yang menjadi hasil uji coba terbaik.

Penentuan nilai *constraint* yang digunakan mempengaruhi solusi yang dihasilkan. Semakin rendah nilai *constraint* maka solusi dihasilkan semakin optimal seperti terlihat pada hasil uji coba ke-2 yang menjadi hasil uji coba terbaik.

SIMPULAN

Dari pembahasan yang telah diuraikan pada bab sebelumnya, dapat

diambil suatu kesimpulan tentang Metode *Simulated Annealing* untuk menyelesaikan masalah Pengiriman Barang di Kantor Pos diantaranya adalah yang pertama dari pengujian yang dilakukan ternyata metode *Simulated Annealing* dapat digunakan untuk menyelesaikan masalah pencarian jalur optimal untuk tujuan lebih dari satu.

Kedua, penentuan nilai suhu awal, nilai suhu akhir, jumlah iterasi, dan nilai *cooling rate* yang digunakan memiliki pengaruh yang signifikan, semakin tinggi nilai suhu awal, semakin rendah nilai suhu akhir, dan semakin tinggi jumlah iterasi yang dipergunakan maka proses iterasi pun semakin banyak dan akan melakukan perbandingan terhadap tiap solusi sehingga metode ini akan menghasilkan solusi yang memiliki nilai mendekati optimal.

Ketiga, penentuan nilai suhu awal, nilai suhu akhir, jumlah iterasi, dan nilai *cooling rate* yang digunakan mempengaruhi lama proses saat dijalankan.

Keempat, semakin kecil nilai *constraint* maka solusi yang dihasilkan juga semakin optimal.

Kelima, banyaknya jumlah tujuan dan jumlah *node* yang digunakan pada *database* juga mempengaruhi lamanya proses saat dijalankan.

Keenam, dalam penelitian ini nama jalannya tidak diletakkan pada nodenya melainkan diletakkan pada busurnya sehingga ketika melakukan *update state* ada pengecekan-pengecekan kondisi kembali yaitu cek kondisi *feasiblenya*. Hal itulah yang membuat prosesnya pun semakin lama.

SARAN

Saran yang dapat disampaikan terkait dengan penelitian ini antara lain yaitu sebaiknya dilakukan pengembangan terhadap perhitungan yang ada untuk mengoptimalkan metode *Simulated Annealing* yang digunakan sehingga dapat menghasilkan jalur yang lebih optimal, sebaiknya juga dilakukan pengembangan untuk menemukan alamat

secara detail, bukan pendekatan alamat yang dituju, dan diharapkan ada penelitian yang menggunakan metode heuristik lain yang digunakan untuk masalah pengiriman barang atau masalah-masalah pencarian jalur optimal yang mempunyai tujuan lebih dari satu.

DAFTAR PUSTAKA

- [1] Dwi Susilo, Martiana Entin, Fariza Arna, Prasetyaningrum Ira. 2010. **“Optimasi Distribusi Barang Berdasarkan Rute dan Daya Tampung Menggunakan Metode *Simulated Annealing*”**. Surabaya : Jurusan Teknik Informatika, PENS-ITS.
- [2] Wahyu Andi, F. Allen. 2009. **“Aplikasi Desktop Pencarian Rute Jalan dengan Algoritma *Simulated Annealing*”**. Bandung : Jurusan Teknik Informatika Fakultas Teknologi Informasi, Universitas Kriten Maranatha.
- [3] Aji, Kristian. 2011. **“Program Aplikasi Optimasi Rute Pendistribusian di Kantor Pos MPC Yogyakarta dengan Metode *Simulated Annealing*”**. Yogyakarta : Matematika dan Informatika, Universitas Bina Nusantara.
- [4] Chibante, Rui, Araujo, Armando and Carvalho, Adriano. 2010. **“*Simulated Annealing Theory with Applications*”**. Sciyo.