

Implementasi *Simulated Annealing* untuk Penentuan Rute pada Jaringan Mikrotik

Gusti Eka Yuliasuti, Citra Nurina Prabiantissa, Agung Mustika Rizki

Abstract—Recently computer networks are increasingly complex. It needs to be a supporting device for network management such as a router. Router is a device that plays an important role in the routing process. In a router stored information in the form of routing paths, where the information includes data and which routers will be passed. In certain cases, a router can have more than one gateway. This is because the router needs to send data packets to several networks that have different segments. Such cases can be handled by using the appropriate routing path selection rules. The routing problem can be regarded as a traveling salesman problem (TSP), where a mechanism is needed to determine the shortest route to be traversed. The author implements the Simulated Annealing Algorithm because it can produce an optimal solution with light computing, so that the routing process can be more effective and efficient.

Index Terms—Computer Network; Routing; Simulated Annealing; Travelling Salesman Problem.

Abstrak—Jaringan komputer saat ini semakin kompleks. Perlu adanya suatu perangkat pendukung untuk manajemen jaringan seperti *router*. *Router* merupakan perangkat yang berperan penting dalam proses *routing*. Pada sebuah *router* tersimpan informasi berupa jalur *routing*, dimana informasi tersebut mencakup data dan *router* mana saja yang akan dilewati. Pada kasus tertentu, *router* dapat memiliki lebih dari satu *gateway*. Hal ini disebabkan karena *router* perlu mengirimkan paket data ke beberapa jaringan yang memiliki segmen berbeda. Kasus tersebut dapat ditangani dengan menggunakan aturan pemilihan jalur *routing* yang tepat. Permasalahan *routing* dapat dikatakan sebagai suatu permasalahan *travelling salesman problem* (TSP), dimana diperlukan suatu mekanisme dalam menentukan rute terpendek untuk dilalui. Penulis mengimplementasikan Algoritma *Simulated Annealing* karena dapat menghasilkan solusi yang optimal dengan komputasi ringan, sehingga proses *routing* dapat lebih efektif dan efisien.

Kata Kunci— Algoritma *Simulated Annealing*; Jaringan Komputer; Penentuan Rute; Travelling Salesman Problem.

Gusti Eka Yuliasuti; Teknik Informatika, Institut Teknologi Adhi Tama Surabaya, Indonesia (email: gustiekay@itats.ac.id)

Citra Nurina Prabiantissa; Teknik Informatika, Institut Teknologi Adhi Tama Surabaya, Indonesia (email: citranurina@itats.ac.id)

Agung Mustika Rizki; Teknik Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Surabaya, Indonesia (email: agung.mustika.if@upnjatim.ac.id)

I. PENDAHULUAN

Seiring dengan perkembangan teknologi, jaringan komputer saat ini semakin kompleks. Perlu adanya suatu perangkat pendukung untuk manajemen jaringan, salah satunya adalah *router*. *Router* merupakan sebuah perangkat penghubung dalam proses *routing* [1]. Sedangkan *routing* merupakan suatu proses pengiriman paket data yang dikirimkan dari satu jaringan ke jaringan yang lain.

Pada sebuah *router* tersimpan informasi berupa jalur *routing*, dimana informasi tersebut mencakup data dan *router* mana saja yang akan dilewati. Pada kasus tertentu, *router* dapat memiliki lebih dari satu *gateway*. Hal ini disebabkan karena *router* perlu mengirimkan paket data ke beberapa jaringan yang memiliki segmen berbeda. Kasus tersebut dapat ditangani dengan menggunakan aturan pemilihan jalur *routing* yang tepat.

Permasalahan *routing* dapat dikatakan sebagai suatu permasalahan *travelling salesman problem* (TSP), dimana diperlukan suatu mekanisme dalam menentukan rute terpendek untuk dilalui [2].

Pada penelitian sebelumnya, permasalahan TSP telah diselesaikan dengan menggunakan beberapa algoritma antara lain: *Hill Climbing* [3], *Tabu Search* [4], *Particle Swarm Optimization* (PSO) [5], Algoritma Genetika [6] dan *Simulated Annealing* [7].

Pada penelitian ini, penulis akan mengimplementasi Algoritma *Simulated Annealing* karena dapat menghasilkan solusi yang cukup optimal dengan komputasi ringan.

II. LANDASAN TEORI

Router pada jaringan mikrotik melakukan pemetaan rute untuk menentukan *traffic* jaringan [8]. Terdapat dua jenis *routing*, yakni *static routing* dan *dynamic routing*. *Static routing* merupakan penentuan rute yang dimasukkan secara manual oleh operator jaringan, sedangkan yang dimasukkan secara manual oleh operator jaringan, sedangkan *dynamic routing* merupakan penentuan rute yang dikumpulkan berdasarkan proses-proses dinamis yang berjalan pada jaringan tersebut.

Pada suatu jaringan mikrotik, terdapat beberapa titik perangkat yang nantinya akan dilewati untuk proses pengiriman data dari perangkat pengirim ke perangkat

penerima. Perlu rute yang optimal dalam proses pengiriman data agar efektif dan efisien, sehingga diperlukan implementasi suatu metode untuk melakukan penentuan rute.

Permasalahan penentuan rute atau selanjutnya disebut TSP tersebut dapat diselesaikan dengan implementasi beberapa metode yang pernah dilakukan pada penelitian-penelitian sebelumnya. Tujuan yang ingin dicapai pada permasalahan TSP adalah mencari jarak terpendek untuk dilalui [6].

Algoritma *Hill Climbing* digunakan oleh Irfan [3] untuk menyelesaikan permasalahan TSP. Studi kasus yang diselesaikan adalah pengiriman surat atau barang suatu perusahaan. Penelitian tersebut menghasilkan kesimpulan bahwa Algoritma *Hill Climbing* dapat digunakan untuk menyelesaikan permasalahan TSP.

Ramadhania dan Rani [4] menerapkan Algoritma Genetika dan *Tabu Search* untuk penyelesaian TSP. Cara kerja Algoritma *Tabu Search* adalah dengan menyimpan solusi terbaik saat tetap mencari solusi, agar solusi terbaik tidak hilang. Kesimpulan dari penelitian tersebut adalah penerapan kedua algoritma ini cukup optimal dalam menghasilkan *runtime* yang cenderung kecil.

Algoritma PSO diterapkan oleh Puteri dkk.[5] untuk mengoptimasi rute pendistribusian air minum di kota Malang. Hasil optimasi dari sistem tersebut dibandingkan dengan perhitungan manual, kemudian dapat disimpulkan bahwa penerapan Algoritma PSO cukup baik dalam menghasilkan solusi yang optimal.

Penelitian terkait permasalahan TSP dengan *time-window* telah diselesaikan dengan menerapkan Algoritma Genetika. Yuliasuti dkk. [6] mendapatkan solusi yang optimal setelah generasi ke-1000. Dalam penerapannya, solusi yang dihasilkan berupa rute jarak terpendek dengan paling sedikit Batasan waktu.

Penyelesaian permasalahan TSP yang terakhir dengan menerapkan Algoritma *Simulated Annealing* seperti yang dilakukan. Hidayati dkk.[7] berpendapat bahwa TSP didefinisikan sebagai pencari urutan lokasi dari keseluruhan lokasi dan setiap lokasi tidak boleh dikunjungi lebih dari satu kali. Tujuannya adalah untuk meminimalisir biaya yang dikeluarkan. Kesimpulan yang didapatkan dari penelitian tersebut adalah penerapan Algoritma *Simulated Annealing* akan mendapatkan solusi yang optimal. Selain itu didapatkan kesimpulan bahwa nilai batas maksimum iterasi dapat mempengaruhi probabilitas pencarian solusi optimal. Semakin besar nilai batas maksimum iterasi, maka semakin tinggi pula probabilitas menghasilkan solusi yang optimal.

Angresti dkk. [9] telah melakukan pengujian terhadap Algoritma *Simulated Annealing* dan didapatkan kesimpulan bahwa Algoritma *Simulated Annealing* nilai fungsi objektif lebih minimal jika parameter kontrol telah ditentukan sebelumnya.

Berdasarkan penelitian-penelitian yang telah dilakukan sebelumnya, penulis akan melakukan penelitian terkait permasalahan TSP dengan mengimplementasi algoritma *Simulated Annealing*.

Algoritma *Simulated Annealing* memiliki kompleksitas yang tidak terlalu tinggi dan waktu komputasi yang tidak terlalu lama, namun tetap dapat menghasilkan solusi yang optimal.

III. METODE PENELITIAN

Algoritma *Simulated Annealing* bekerja seperti proses pendinginan dan pemanasan logam untuk menjadi kristal dengan menggunakan energi seminim mungkin yang selanjutnya biasa disebut proses *annealing* [10].

Jika dianalogikan, ketika memanaskan logam kemudian didinginkan maka akan menghasilkan kristal dengan kualitas baik [11]. Hal tersebut berlaku sebaliknya, jika logam didinginkan dengan cepat maka kristal yang terbentuk tidak sempurna [12]. Algoritma ini menirukan proses pendinginan dengan menurunkan suhu secara bertahap sehingga konvergen pada kondisi beku dan stabil [13].

Terdapat beberapa parameter yang diperhatikan dalam Algoritma *Simulated Annealing*, antara lain [14]:

- a. Keadaan sistem; solusi yang memungkinkan
- b. Energi; besar fungsi tujuan dari kombinasi keadaan sistem atau bisa disebut sebagai nilai objektif
- c. Temperatur; nilai kendali yang membuat suatu keadaan dapat bergerak atau tidak, selanjutnya disebut sebagai parameter kontrol
- d. Laju Pendinginan; proses penurunan temperatur yang berfungsi untuk mengetahui seberapa cepat pencapaian solusi akhir yang didapat

$$T_t = \alpha \times T_0 \quad (1)$$

Keterangan:

T_0 = temperatur awal

T_t = temperatur baru

α = faktor reduksi suhu ($\alpha < 1$)

Alur dari Algoritma *Simulated Annealing* adalah sebagai berikut [15]:

1. Membangkitkan solusi awal dengan membangkitkan dua nilai acak untuk x_1 dan x_2 . Selain itu juga perlu menentukan nilai awal untuk temperature awal (T_0), temperature akhir (T_t), iterasi maksimum, faktor reduksi suhu (α) dan nilai objektif awal (E).
2. Mencari solusi baru dengan membangkitkan kembali dua nilai acak sekaligus menghitung nilai objektif dari solusi baru ini.
3. Mengecek parameter kontrol untuk mengetahui perlu diturunkan atau tidak. Jika parameter kontrol perlu diturunkan, maka lanjut ke langkah 4. Namun jika parameter kontrol tidak diturunkan, maka perlu mengulangi langkah 2 dan 3.
4. Menentukan solusi baru yang terbaik dari beberapa solusi baru yang diterima. Kemudian solusi baru yang terbaik tersebut akan menggantikan solusi awal.
5. Mengevaluasi solusi baru untuk mengetahui apakah solusi baru tersebut diterima atau tidak dengan menggunakan rumus berikut

$$\Delta E = E(X_{i+1}) - E(X_i)$$

(2)

Keterangan:

ΔE = selisih nilai objektif

$E(X_{i+1})$ = nilai objektif dari solusi baru

$E(X_i)$ = nilai objektif dari solusi awal

Jika nilai objektif dari solusi baru lebih kecil dari nilai objektif awal ($\Delta E \leq 0$), maka solusi baru diterima. Namun jika nilai objektif solusi baru lebih besar dari pada solusi awal ($\Delta E > 0$), maka solusi baru masih perlu dipertimbangkan dengan menggunakan probabilitas berikut

$$P(\Delta E) = e^{-\Delta E/T} > r$$

(3)

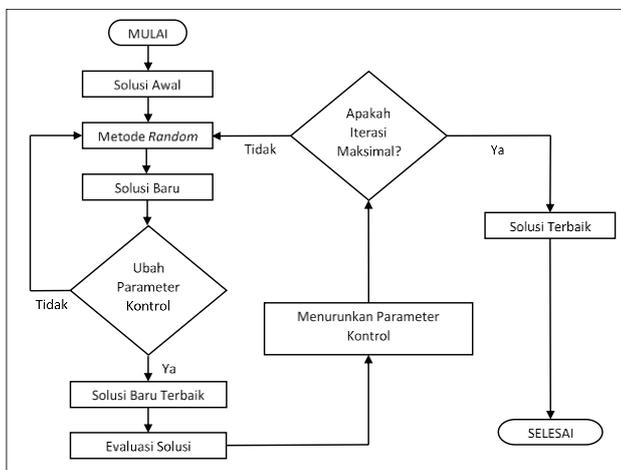
Keterangan:

T = parameter kontrol

r = bilangan random antara 0 dan 1

6. Menurunkan parameter kontrol setelah mendapatkan solusi baru dan solusi baru tersebut diterima. Parameter kontrol diturunkan dengan menggunakan rumus persamaan (1).
7. Mengecek iterasi maksimum apakah sudah selesai atau belum. Jika iterasi maksimum selesai, artinya solusi terbaik merupakan solusi yang optimal. Namun jika iterasi tidak maksimum, maka perlu mengulangi langkah 1 hingga 7.

Skema dari implementasi Algoritma *Simulated Annealing* pada permasalahan TSP seperti ditunjukkan pada Gambar 1.



Gambar 1 Skema Algoritma Simulated Annealing

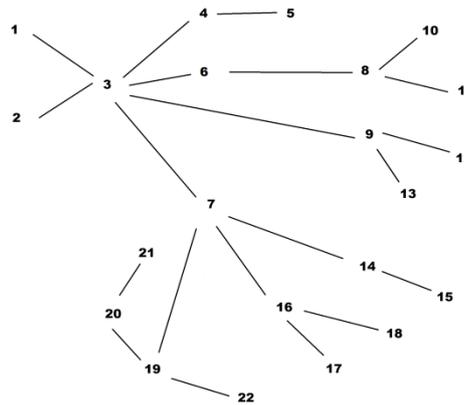
IV. HASIL DAN PEMBAHASAN

Studi kasus pada penelitian ini adalah salah satu topologi jaringan pada sebuah perguruan tinggi di Surabaya [1]. Berdasarkan gambaran topologi jaringan tersebut, penulis mentransformasi topologi tersebut menjadi sebaran beberapa titik dan garis penghubung dengan menggunakan pengkodean permutasi sebagai contoh ditunjukkan pada Gambar 2.

Pengkodean permutasi ini merupakan pengkodean yang dituliskan secara berurutan mulai dari angka 1 hingga seterusnya tanpa ada angka yang diulang. Pada

Gambar 2 terdapat 22 titik yang menunjukkan posisi router dan hotspot pada setiap area.

Dalam permasalahan TSP, pengkodean permutasi yang dimaksud adalah setiap titik hanya akan dilalui sebanyak satu kali saja [16]. Namun, pada penelitian ini permutasi untuk rute yang akan dicari tetap sesuai dengan gambaran topologi jaringannya atau sesuai dengan titik-titik yang terhubung. Jika terdapat calon solusi dengan titik-titik yang tidak terhubung maka akan ada sebuah *penalty* yang akan terus bertambah pada nilai objektifnya. Matriks jarak dengan satuan waktu antar titik-titik tersebut ditunjukkan seperti Tabel 1.



Gambar 2 Pengkodean Permutasi

Tabel 1 Matriks Jarak dengan Satuan Waktu

	1	2	3	4	5	...	22
1	0	1000	39	1000	1000	...	1000
2	1000	0	42	1000	1000	...	1000
3	39	42	0	38	1000	...	1000
4	1000	1000	38	0	41	...	1000
5	1000	1000	1000	41	0	...	1000
...
22	1000	1000	1000	1000	1000	...	0

Pada Tabel 1, terdapat nilai 1000 yang menunjukkan antar kedua titik tersebut tidak terhubung sehingga menjadi *penalty*. Satuan waktu yang digunakan pada matriks jarak yakni milisekon.

Setelah mengetahui matriks jarak yang digunakan, maka selanjutnya dapat dilakukan pengujian Algoritma *Simulated Annealing* terhadap studi kasus penelitian.

Pengujian pada penelitian ini dilakukan sebanyak sepuluh kali, agar didapatkan hasil rata-ratanya. Sebagai contoh, satu rute yakni 1→3→6→8→10 dengan nilai objektif awal sebesar 100. Nilai tersebut memiliki arti total waktu yang diperlukan untuk menempuh rute tersebut dengan satuan milisekon. Hasil pengujian untuk rute tersebut dapat dilihat pada Tabel 2.

Tabel 2 Hasil Pengujian

Iterasi	Rute	Nilai Objektif	Solusi
1	1→3→6→8→10	100	Awal
2	1→3→6→8→11	124 > 100	Ditolak

3	1→3→7→14→15	153 > 100	Ditolak
4	1→3→7→12→13	425 > 100	Ditolak
5	1→3→9→12	37 < 100	Diterima

Berdasarkan tabel hasil pengujian tersebut, salah satu contoh rute yang pendek adalah rute pada iterasi ke 5. Dengan kata lain, rute tersebut memiliki waktu yang paling singkat dibandingkan dengan rute lainnya yakni hanya membutuhkan waktu selama 37 milisekon untuk melakukan *routing*. Rute tersebut diprioritaskan dibanding dengan rute lainnya karena dapat membuat proses *routing* menjadi lebih efektif dan efisien. Pencarian rute terbaik akan terus berlanjut untuk rute-rute lainnya, hingga didapatkan solusi rute yang optimal.



Gambar 3 Hasil Pengujian

Pada Gambar 3 menunjukkan bahwa selama pengujian sebanyak 10 kali, didapatkan hasil yang fluktuatif. Titik optimal yang didapatkan berdasarkan hasil pengujian tersebut berada pada nilai 37. Hal tersebut menunjukkan bahwa dengan mengimplementasi Algoritma Simulated Annealing akan didapatkan nilai obyektif disertai *penalty* seminim mungkin.

V. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, penulis dapat menarik kesimpulan bahwa Algoritma *Simulated Annealing* dapat diimplementasikan untuk menyelesaikan permasalahan TSP. Kelebihan dari Algoritma ini adalah tingkat kompleksitasnya yang rendah, dengan begitu waktu komputasi yang diperlukan tidak terlalu lama. Hal tersebut yang tidak didapatkan dalam implementasi metode lainnya. Walaupun tingkat kompleksitas dari Algoritma *Simulated Annealing* cukup rendah, namun Algoritma *Simulated Annealing* dapat menghasilkan solusi yang optimal. Optimal yang dimaksud pada penelitian ini adalah didapatkan jarak dengan waktu minimum berdasarkan nilai obyektif dan *penalty*. Algoritma *Simulated Annealing* membuat proses *routing* menjadi lebih efektif dan efisien. Proses *routing* dikatakan efektif karena telah mencapai tujuan yang diharapkan yakni mendapatkan rute dalam jaringan. Selain itu proses *routing* ini dapat dikatakan efektif karena dengan mengimplementasi salah satu metode sederhana yakni

Algoritma *Simulated Annealing*, didapatkan hasil yang terbaik dengan minim waktu komputasinya.

REFERENSI

- [1] G. E. Yuliasuti, C. N. Prabiantissa, S. Agustini, and D. H. Sulaksono, "Optimasi Rute Jaringan Mikrotik dengan Algoritme Genetika," in *Prosiding Seminar Nasional Sains dan Teknologi Terapan VIII*, 2020, no. October, pp. 209–216.
- [2] G. E. Yuliasuti, W. F. Mahmudy, and A. M. Rizki, "Implementation of Genetic Algorithm to Solving Travelling Salesman Problem with Time Window (TSP-TW) for Scheduling Tourist Destinations in Malang City," *J. Inf. Technol. Comput. Sci.*, vol. 2, no. 1, pp. 1–10, 2017.
- [3] M. Irfan, "Penyelesaian Travelling Salesman Problem (TSP) Menggunakan Algoritma Hill Climbing dan MATLAB," *J. Mat.*, vol. 17, no. 1, pp. 13–20, 2018, doi: 10.29313/jmtm.v17i1.3090.
- [4] S. E. Ramadhania and S. Rani, "Implementasi Kombinasi Algoritma Genetika dan Tabu Search untuk Penyelesaian Travelling Salesman Problem," *Automata*, 2021.
- [5] R. N. Puteri, A. W. Widodo, and I. Cholissodin, "Optimasi Multiple Travelling Salesman Problem Pada Pendistribusian Air Minum Menggunakan Algoritme Particle Swarm Optimization (Studi Kasus : UD . Tosa Malang)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 1, no. 9, pp. 842–848, 2017.
- [6] G. E. Yuliasuti, W. F. Mahmudy, and A. M. Rizki, "Penanganan Fuzzy Time Window pada Travelling Salesman Problem (TSP) dengan Penerapan Algoritma Genetika," *MATICS J. Ilmu Komput. dan Teknol. Inf.*, vol. 9, no. 1, pp. 38–43, 2017.
- [7] R. Hidayati, I. Guntoro, and S. Junianti, "Penggunaan Metode Simulated Annealing untuk Penyelesaian Travelling Salesman Problem," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 4, no. 2, pp. 217–221, 2019, [Online]. Available: <https://jurnal.unimed.ac.id/2012/index.php/cess/article/view/13626>.
- [8] D. Edi, "Kajian Algoritma Routing dalam Jaringan Komputer," *J. Inform. UKM*, vol. II, no. 3, 2006.
- [9] N. D. Angresti, A. Djunaidy, and A. Mukhlason, "Penerapan Hiperheuristik Berbasis Metode Simulated Annealing untuk Penyelesaian Permasalahan Optimasi Lintas Domain," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 5, no. 1, pp. 33–40, 2019, doi: 10.25077/teknosi.v5i1.2019.33-40.
- [10] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science (80-.)*, vol. 220, no. 4598, pp. 671–

- 680, 1983.
- [11] G. E. Noviardianto, M. Novel, and M. B. Legowo, "Penggunaan Metode Simulated Annealing untuk Optimasi Penempatan Posisi Access Point pada Jaringan WI-FI," *J. AL-AZHAR Indones. SERI SAINS DAN Teknol.*, vol. 5, no. 1, p. 10, 2019, doi: 10.36722/sst.v5i1.318.
- [12] J. B. C. Chagas, U. E. F. Silveira, M. P. L. Benedito, and A. G. Santos, "Simulated Annealing Metaheuristic for the Double Vehicle Routing Problem with Multiple Stacks," pp. 1311–1316, 2016, doi: 10.1109/ITSC.2016.7795726.
- [13] G. E. Yuliasuti, A. M. Rizki, W. F. Mahmudy, and I. P. Tama, "Optimization of Multi-Product Aggregate Production Planning using Hybrid Simulated Annealing and Adaptive Genetic Algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, pp. 484–489, 2019, doi: 10.14569/IJACSA.2019.0101167.
- [14] G. E. Yuliasuti, A. M. Rizki, W. F. Mahmudy, and I. P. Tama, "Determining Optimum Production Quantity on Multi-Product Home Textile Industry by Simulated Annealing," *J. Inf. Technol. Comput. Sci.*, vol. 3, no. 2, pp. 159–168, 2018, doi: 10.25126/jitecs.20183264.
- [15] A. Khumaidi, R. Raafi'udin, and I. P. Solihin, "Simulation Of Traveling Salesman Problem For Distribution Of Fruits In Bogor City With Simulated Annealing Method," *J. Mantik*, vol. 3, no. 4, pp. 611–618, 2020.
- [16] B. D. Setiawan and A. Pinandito, "Optimasi Kunjungan Objek Wisata dengan Menggunakan Algoritma Genetik," in *Prosiding Seminar Nasional Teknologi dan Rekayasa Informasi 2016*, 2016, p. 71.