ISSN: 2339-2541
JURNAL GAUSSIAN, Volume 15, Nomor 2, Tahun 2026, Halaman 320 - 331
Online di: https://ejournal3.undip.ac.id/index.php/gaussian/

PERBANDINGAN ALGORITMA *NEAREST NEIGHBOR* DAN *ANT COLONY OPTIMIZATION* DALAM OPTIMASI RUTE WISATA SEMARANG BERBASIS *MULTI-ATTRIBUTE UTILITY THEORY*

Ashilah Tsuraya Izzati[1], **Hasbi Yasin**[1*], **Triastuti Wuryandari**[1]
[1] Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro
***e-mail**: hasbiyasin@live.undip.ac.id

DOI: 10.14710/j.gauss.15.2.320-331

Article Info:
Received: 2026-01-26
Accepted: 2026-08-24
Available Online: 2026-09-01

Keywords:
Muti-Attribute Utility Theory; Traveling Salesman Problem; Tourism Route Optimization; Branch and Bound; Nearest Neighbor; Ant Colony Optimization

Abstract: Tourism route planning is an important aspect of tourism development to improve travel efficiency and destination quality. This study aims to determine priority tourist destinations and to compare the performance of tourism route optimization methods in Semarang. Priority destinations are identified using the *Multi-Attribute Utility Theory* (MAUT) with criteria weights determined by the *Rank Order Centroid* (ROC). Tourism destinations with the highest utility values are selected as priority destinations and modelled as *Traveling Salesman Problem* (TSP). Tourism route optimization is carried out by comparing three optimization problem-solving method, namely exact method, heuristic method, and metaheuristic method. The exact method employs the *Branch and Bound* (B&B) algorithm as a benchmark to obtain the optimal solution. The heuristic method uses the *Nearest Neighbour* (NN) algorithm and metaheuristic method uses the *Ant Colony Optimization* (ACO) algorithm. Algorithm performance is evaluated based on total travel distance, computation time, and relative error (RE) to the optimal solution. The result show that the NN algorithm yields a relative error of 8.66%, while the ACO algorithm achieves a lower relative error of 2.2%. This indicates that ACO algorithm produces routes that are closer to the optimal solution.

1. PENDAHULUAN

Pertumbuhan ekonomi daerah tidak terlepas dari peran sektor pariwisata yang berkontribusi dalam meningkatkan pendapatan masyarakat, menyediakan lapangan kerja, sera menguatkan identitas dan budaya lokal. Kota Semarang dan Kabupaten Semarang memiliki karakteristik serta potensi pariwisata yang beragam, dengan persebaran destinasi yang relatif berjauhan. Kondisi tersebut menimbulkan tantangan bagi wisatawan dalam merencanakan perjalanan wisata yang efisien. Permasalahan tersebut dapat diatasi melalui penerapan Sistem Pendukung Keputusan (SPK). Dalam SPK, penentuan alternatif terbaik berdasarkan pertimbangan banyak kriteria dapat dilakukan menggunakan metode *Multi-Attribute Utility Theory* (MAUT) (Wira, 2019). Metode MAUT didefinisikan sebagai pendekatan pengambilan keputusan dengan mempertimbangkan berbagai kriteria penentuan destinasi, seperti harga tiket, jadwal perjalanan, dan ulasan pengguna. Pada umumnya, penentuan bobot setiap kriteria dalam metode MAUT dilakukan menggunakan metode *Rank Order Centroid* (ROC). Bobot yang diberikan berdasarkan urutan tingkat kepentingan relatif setiap kriteria sehingga kriteria yang dianggap lebih penting memperoleh bobot yang lebih besar dibandingkan dengan kriteria lain (Ilhami *et al*., 2024). Setelah diperoleh destinasi wisata prioritas, permasalan rute wisata dimodelkan sebagai *Traveling Salesman Problem* (TSP) untuk meminimalkan jarak tempuh. Optimasi rute wisata dilakukan dengan membandingkan tiga metode optimasi, yaitu metode eksak, heuristik, dan metaheuristik. Metode eksak menggunakan algoritma *Branch and Bound* (B&B) sebagai acuan pembanding untuk memperoleh solusi optimal pada jumlah data terbatas (Witin *et al*., 2025),

metode heuristik menggunakan algoritma *Nearest Neighbor* (NN), sedangkan metode metaheuristik menggunakan algoritma *Ant Colony Optimization* (ACO).

Penelitian oleh Sholihaningtias (2023) menunjukkan bahwa metode MAUT dengan pembobotan *Rank Order Centroid* (ROC) efektif dan objektif pada seleksi karyawan berdasarkan kriteria dan bobot yang ditentukan. Penelitian Aqila *et al*. (2024) mengintegrasikan metode MAUT untuk penentuan destinasi dan optimasi rute menggunakan *Particle Swarm Optimization* (PSO) untuk menentukan rute wisata yang efisien. Penyelesaian TSP menggunakan algoritma B&B menghasilkan solusi optimal dengan jarak 148,6 km pada jumlah data terbatas (Witin *et al*., 2025), sedangkan algoritma NN memberikan solusi dalam waktu yang relatif singkat, tetapi solusi yang diperoleh tidak selalu optimal (Putra *et al*., 2023). Algorima ACO terbukti efektif dalam optimasi rute logistik dengan pengurangan jarak hingga 37% (Sembiring dan Chailes, 2020). Isnaeny *et al*. (2025) melakukan penelitian dengan hasil algoritma B&B menunjukkan solusi optimal dengan peningkatan waktu komputasi pada dataset besar, sedangkan algoritma NN menghasilkan solusi dengan *gap* 15-25% terhadap solusi optimal. Perbandingan ACO dan NN oleh Prasetio dan Suseno (2024) menunjukkan bahwa algoritma NN menghasilkan rute terpendek 46,7 km, sedangkan algoritma ACO menghasilkan rute 53,2 km, keduanya lebih efisien dibandingkan rute awal 65,1 km. Penelitian ini disusun berdsarkan kajian terhadap penelitian terkait, dengan tujuan menentukan destinasi wisata prioritas di Kota Semarang dan Kabupaten Semarang menggunakan metode MAUT dengan pembobotan ROC, serta melakukan perbandingan kinerja algoritma NN dan ACO dengan algoritma B&B sebagai acuan pembanding.

2. TINJAUAN PUSTAKA

Sistem Pendukung Keputusan (SPK) memanfaatkan *Rank Order Centroid* sebagai metode untuk menentukan bobot kriteria berdasarkan urutan tingkat kepentingannya sehingga kriteria dengan prioritas lebih tinggi memperoleh bobot lebih besar (Ilhami *et al*., 2024). Metode ROC sering dikombinasikan dengan metode SPK lain untuk menghasilkan bobot yang lebih objektif dan terukur. Wira (2019) menjelaskan bahwa metode *Multi-Attribute Utility Theory* (MAUT) diimplementasikan untuk menentukan alternatif terbaik berdasarkan beberapa kriteria. Dalam MAUT, setiap kriteria dinilai pada skala utilitas 0-1 yang merepresentasikan preferensi pengambil keputusan. Implementasi metode MAUT yang dikombinasikan dengan pembobotan ROC dilakukan melalui tahapan berikut:

1. Mengidentifikasi kriteria yang akan ditetapkan dalam pengambilan keputusan.
2. Memberikan peringkat pada setiap kriteria berdasarkan tingkat kepentingan relatifnya.
$$C_1 > C_2 > C_3 > \cdots > C_j \quad (1)$$
$$W_1 > W_2 > W_3 > \cdots > W_j \quad (2)$$
dengan C_j adalah kriteria ke-j dan W_j adalah nilai bobot ke-j.
3. Menghitung bobot kriteria dengan rumus ROC pada Persamaan (3) (Sholihaningtias, 2023) .
$$W_j = \frac{1}{j}\sum_{p=1}^{j}\left(\frac{1}{p}\right) \quad (3)$$
dengan W_j menyatakan bobot ke-j, j adalah banyaknya kriteria, p merupakan urutan prioritas, dan $\sum_{p=1}^{j}\left(\frac{1}{p}\right)$ menunjukkan total nilai dari pembagian kriteria.

4. Membuat matriks keputusan.

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mj} & \cdots & x_{mn} \end{bmatrix} \quad (4)$$

dengan X merupakan matriks keputusan, x_{ij} adalah elemen dari matriks keputusan yang merepresentasikan alternatif ke-i pada kriteria ke-j, m menyatakan banyaknya alternatif, dan n menyatakan banyaknya kriteria.

5. Melakukan normalisasi pada matriks keputusan menggunakan Persamaan (5) dan (6) (Nuroji, 2022).

$$r_{ij}^{*} = \frac{x_{ij} - \min(x_{ij})}{max(x_{ij}) - \min(x_{ij})}; \text{untuk kriteria } benefit \quad (5)$$

$$r_{ij}^{*} = 1 + \left(\frac{\min(x_{ij}) - x_{ij}}{max(x_{ij}) - \min(x_{ij})}\right); \text{untuk kriteria } cost \quad (6)$$

dengan r_{ij}^{*} menyatakan nilai hasil normalisasi untuk alternatif x_{ij} alternatif ke-i pada kriteria ke-j, $\min(x_{ij})$ dan $\max(x_{ij})$ masing-masing menyatakan merupakan nilai terkecil dan nilai terbesar dari seluruh alternatif pada kriteria ke-j.

6. Menghitung nilai utilitas marjinal menggunakan Persamaan (8) (Nuroji, 2022).

$$u_{ij} = \frac{e^{(r_{ij}^{*})^2} - 1}{e - 1} \quad (7)$$

$$u_{ij} = \frac{e^{(r_{ij}^{*})^2} - 1}{1{,}71} \quad (8)$$

dengan u_{ij} menyatakan nilai utilitas marjinal untuk alternatif ke-i pada kriteria ke-j dan e menyatakan bilangan eksponensial.

7. Menghitung nilai utilitas total menggunakan Persamaan (9) (Nuroji, 2022).

$$u_i = \sum_{i=1}^{n} u_{ij} \,.\, w_j \quad (9)$$

dengan u_i merupakan nilai utilitas total untuk alternatif ke-i.

Apriyani (2024) mendefinisikan *graph* sebagai representasi diagram yang terdiri dari tiitk-titik yang disebut *node* dan saling dihubungan oleh garis-garis yang disebut sisi (*edge*). Suatu *graph* G dinotasikan sebagai sebagai $G = \big(V(G), E(G)\big)$ dengan $V(G)$ merupakan himpunan *node* dan $E(G)$ merupakan himpunan *edge*. *Graph* dapat dikategorikan berdasarkan arah dan bobot pada setiap sisinya menjadi empat jenis, yaitu (Munir, 2009)

1. *Graph* berarah dan berbobot didefinisikan sebagai *graph* dengan sisi-sisinya diberi orientasi arah dan bobot.
2. *Graph* tidak berarah dan berbobot didefinisikan sebagai *graph* dengan sisi tanpa arah, tetapi setiap sisi memiliki bobot.
3. *Graph* berarah dan tidak berbobot didefinisikan sebagai *graph* dengan sisi-sisinya memilihi arah tertentu tanpa disertai bobot.
4. *Graph* tidak berarah dan tidak berbobot didefinisikan sebagai *graph* dengan sisi-sisinya tidak memiliki arah maupun bobot.

Zukhruf dan Frazila (2021) menjelaskan bahwa optimasi merupakan suatu proses dalam pengambilan keputusan yang berfokus pada penentuan alternatif paling optimal di antara berbagai pilihan yang ada dengan mempertimbangkan tujuan yang ingin dicapai serta batasan yang ada. Dalam penyelesaiannya, permasalahan optimasi dapat diselesaikan dengan menggunakan metode eksak dan metode pendekatan. Penyelesaian permasalahan

optimasi berskala besar dengan pendekatan eksak membutuhkan waktu yang cukup lama (Abbaspour *et al*., 2022). Oleh sebab itu, permasalahan optimasi yang kompleks umumnya diselesaikan menggunakan metode pendekatan berbasis heuristik dan metaheuristik. Metode heuristik menekankan efisiensi pencarian solusi, tetapi berpotensi terjebak pada solusi optimal lokal. Dengan demikian, metode metaheuristik menerapkan proses pencarian iteraitf yang lebih fleksibel sehingga mampu menghindari jebakan solusi lokal optimal.

Traveling Salesman Problem (TSP) mulai dikaji sejak abad ke-18 oleh dua matematikawan, yaitu Sir William Rowam Hamilton dari Irlandia dan Thomas Penyngton Kirkman dari Inggris. Gunawan *et al*. (2022) menyatakan TSP sebagai permasalahan dalam menentukan jarak minimal dari suatu tur lintasan tertutup yang melewati *n* kota, dengan setiap kota hanya dapat dikunjungi satu kali. TSP direpresentasikan sebagai *graph* dengan setiap kota merupakan representasi dari *node* dan garis yang menghubungkan dua *node* disebut sisi (*edge*). Dalam teori *graph*, perjalanan pulang-pergi yang melewati seluruh kota tepat satu kali disebut sebagai tur atau siklus Hamilton.

A. H. Land dan A.G. Doig memperkenalkan *Branch and Bound* (B&B) pada tahun 1960. B&B adalah metode optimasi eksak yang bekerja dengan memecah permasalahan menjadi submasalah melalui proses percabangan (*branching*) dan menerapkan pembatasan (*bounding*) untuk menentukan simpul yang layak dieksplorasi (Margiyani dan Mussafi, 2014). Simpul yang tidak memenuhi kriteria dieliminasi melalui proses pemangkasan (*pruning*). Algoritma B&B disajikan dengan tahapan berikut (Suyanto, 2010):

1. Simpul akar dimasukkan ke dalam antrean Q. Jika simpul akar merupakan simpul tujuan (*goal node*), solusi diperoleh. Proses berakhir.
2. Jika antrean Q kosong, tidak diperoleh solusi . Proses berakhir.
3. Jika antrean Q tidak kosong, memilih simpul *i* dengan *cost* $\hat{c}(i)$ terkecil. Jika terdapat lebih dari satu simpul yang memenuhi, memilih salah satu simpul. Fungsi heuritik untuk menghitung perkiraan biaya sebagai berikut (Suyanto, 2010):
$$\hat{c}(i) = f(i) + \hat{g} \tag{10}$$
dengan $\hat{c}(i)$ adalah biaya perkiraan untuk simpul *i*, $f(i)$ adalah biaya nyata dari simpul akar ke simpul *i*, dan $\hat{g}$ adalah biaya perkiraan dari simpul *i* ke simpul tujuan.
4. Jika simpul *i* merupakan simpul tujuan (*goal node*), solusi berhasil diperoleh. Jika hanya satu solusi yang diinginkan, proses pencarian dihentikan. Pada masalah optimasi dengan pendekatan *least cost search*, *cost* semua simpul aktif diperiksa. Jika memiliki *cost* lebih besar daripada *cost* simpul solusi, simpul tersebut dipangkas (*pruning*).
5. Jika simpul *i* bukan merupakan simpul tujuan, simpul *i* diekspansi dengan membangkitkan seluruh simpul anak. Jika tidak dihasilkan simpul anak, proses kembali ke langkah 2.
6. Setiap simpul anak *j* yang dihasilkan simpul *i* dihitung *cost* $\hat{c}(j)$, kemudian semua anak simpul dimasukkan ke dalam antrean Q.
7. Proses diulangi dari langkah 2.

Nearest Neighbor (NN) merupakan metode heuristik yang menerapkan prinsip algoritma *Greedy*. Pada tahun 1951, J.G. Skellam memperkenalkan konsep jarak tetangga terdekat dalam analisis spasial dan dikembangkan oleh P.J. Clark dan F.C. Evans. Algoritma NN mampu menemukan lintasan yang relatif pendek, tetapi hasil yang diperoleh belum tentu optimal. Hal tersebut disebabkan penerapan algoritma *Greedy* menyebabkan pemilihan solusi pada setiap tahap hanya didasarkan pada kondisi lokal tanpa mempertimbangkan keseluruhan rute sehingga terkadang rute yang lebih pendek dapat terlewat (Khan dan Agrawal, 2016). Algoritma *Nearest Neighbor* disajikan dengan tahapan berikut (Sutoyo, 2018):

1. Menentukan *node* awal sirkuit yang sekaligus menjadi *node* akhir dari sirkuit.

2. Mentukan *node-node* lain yang terhubung dengan *node* awal dan belum pernah dikunjungi.
3. Memilih lintasan atau garis dengan bobot terkecil yang menghubungkan *node* awal dengan *node* yang belum pernah dikunjungi.
4. *Node* dari garis yang terpilih menjadi *node* awal pencarian selanjutnya dan ditandai sebagai telah dikunjungi.
5. Ulangi tahapan 2 hingga 4 hingga semua *node* ditandai sebagai telah dikunjungi.

Marco Dorigo memperkenalkan *Ant Colony Optimization* (ACO) pada tahun 1992 melalui penelitian disertasinya, kemudian dipublikasikan sebagai *Ant System*. ACO didefinisikan sebagai metode metaheuristik yang diadaptasi dari cara koloni semut menemukan jalur terpendek antara sarang dan sumber makanan melalui proses alami yang tidak melibatkan indra penglihatan (Karjono *et al*., 2016). Semut-semut akan meninggalkan jejak berupa zat kimia yang disebut *pheromone* pada setiap jalur yang dilewati. Algoritma *Ant Colony Optimization* disajikan dengan tahapan berikut (Mutakhiroh *et al*., 2007):

1. Inisialisasi parameter, yaitu banyak kota (n), jumlah iterasi maksimum (y), jumlah semut (m), intensitas *pheromone* awal (τ_{ij}), tetapan sisa penguapan *pheromone* (ρ), pengendali pengaruh intensitas *pheromone* (α), parameter pengendali pengaruh visibilitas (β), kuantitas *pheromone* konstan (Q), dan kota pertama setiap semut.
2. Menentukan visibilitas antarkota.
   Nilai visibilitas antarkota dapat dihitung menggunakan Persamaan (11).
   $$\eta_{ij} = \frac{1}{d_{ij}} \tag{11}$$
   dengan η_{ij} adalah visibilitas antara kota i dan kota j dan d_{ij} adalah jakar antara kota i dan kota j.
3. Menentukan kota yang akan dikunjungi selanjutnya dengan menggunakan probabilitas perjalanan semut antarkota dapat dihitung menggunakan Persamaan (12) (Dorigo *et al*., 1996).
   $$p_{ij}^{k}(t) = \begin{cases} \dfrac{(\tau_{ij}(t)^{\alpha})(\eta_{ij})^{\beta}}{\sum_{j \in N_i^k}(\tau_{ij}(t)^{\alpha})(\eta_{ij})^{\beta}}, untuk\ j\ \in N_i^k \\ 0 \qquad\qquad\qquad\quad , untuk\ lainnya \end{cases} \tag{12}$$
   dengan $p_{ij}^{k}(t)$ adalah probabilitas semut k memilih jalur dari kota i ke kota j pada iterasi ke-t, $\tau_{ij}(t)$ adalah intensitas *pheromone* pada jalur kota i ke kota j pada iterasi ke-t, α adalah parameter pengendali pengaruh intensitas *pheromoene*, β adalah parameter pengendali pengaruh visibilitas, dan N_i^k adalah kumpulan kota yang belum dikunjungi oleh semut k saat berasa di kota i.
   Probabilitas pada masing-masing kota yaang telah didapatkan akan dihitung probabilitas kumulatifnya. Probabilitas kumulatif dari perjalanan semut antarkota menggunakan Persamaan (13) (Jaelani, 2016).
   $$F(i, j_n) = \sum_{i=1}^{n} p_{ij}^{k} \tag{13}$$
   dengan $F(i, j_n)$ adalah probabilitas kumulatif dari kota i ke kota j_n dan j_n adalah kandidat kota tujuan yang belum dilewati oleh semut k.
4. Membangun satu rute lengkap.
5. Membangun rute untuk setiap semut pada satu iterasi.
6. Memperbarui *pheromone global*.
   Perhitungan perubahan *pheromone* menggunakan Persamaan (14) (Gao, 2020).

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (14)$$

dengan

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{T^k}, untuk\ semut\ k\ melalui\ sisi\ (i,j) \\ 0, \qquad\qquad lainnya \end{cases} \quad (16)$$

dengan $\tau_{ij}(t+1)$ adalah intensitas *pheromone* jalur kota i ke kota j pada iterai ke-$(t+1)$, ρ adalah tetapan sisa penguapan *pheromone* dalam setiap iterasi, $\Delta\tau_{ij}^{k}$ adalah penambahan *pheromone* jalur kota *i* ke kota *j* pada iterasi ke-*t* yang dilewati oleh semut k, Q adalah kuantitas *pheromone*, dan T^k adalah jarak yang sudah dilewati oleh semut k.

7. Mengulang langkah hingga iterasi terpenuhi.

Galat atau *error* digunakan untuk mengukur seberapa dekat solusi pendekatan dengan solusi eksak. *Absolute error* menunjukkan selisih tanpa memperhatikan tanda, sedangkan *relative error* merupakan normalisasi *absolute error* terhadap nilai eksak sehingga memudahkan perbandingan tingkat keoptimalan solusi. Secara matematis dapat dirumuskan sebagai berikut (Munir, 2015):

$$|\varepsilon| = |a - \hat{a}| \quad (17)$$

$$\varepsilon_r = \frac{|\varepsilon|}{a} \times 100\% \quad (18)$$

dengan a adalah nilai eksak, $\hat{a}$ adalah nilai pendekatan, $|\varepsilon|$ adalah *relative error*, dan ε_r adalah *relative error* dalam persen.

3. METODE PENELITIAN

Data yang dianalisis dalam penelitian ini bersumber dari data sekunder terkait destinasi wisata di Kota Semarang dan Kabupaten Semarang. Objek penelitian terdiri dari 24 destinasi wisata dan satu hotel, Hotel Horison Inn Antawirya Semarang sebagai *node* awal dan akhir rute perjalanan. Data mengenai nama destinasi diperoleh dari situs *Klook Travel* (https://www.klook.com/), sedangkan data jarak tempuh antardestinasi diperoleh menggunakan *OpenRouteService Application Programming Interface* (ORS API) dengan mode transportasi kendaraan roda empat (*driving-car*). Data koordinat destinasi wisata (*Lon dan Lat*), preferensi pengunjung (*Rating*), jumlah ulasan (*Reviews*), dan biaya masuk destinasi (*Fee*) bersumber dari *Google Maps*.

Analisis data dalam penelitian ini dilakukan melalui beberapa tahapan yang disusun secara sistematis. Proses pengolahan dan analisis data dibantu oleh *software Pyhton*. Analisis data yang diterapkan dalam penelitian ini disajikan dengan tahapan berikut:

1. Menginput data, yaitu data jarak tempuh antardestinasi berupa matriks dan data destinasi wisata.
2. Melakukan pra-pemrosesan data dengan tahapan sebagai berikut:
   a. Menyesuaikan tipe data, yaitu data *Rating* dan *Fee* dikonversi menjadi tipe data *float*, sedangkan data *Reviews* dikonversi menjadi *integer.*
   b. Mengganti nilai kosong atau nol dengan tak hingga (∞) pada elemen diagonal.
3. Melakukan analisis deskriptif.
4. Tahapan implementasi MAUT dengan pembobotan ROC sebagai berikut:
   a. Menentukan jenis kriteria.

b. Menghitung bobot setiap kriteria menggunakan metode ROC.
c. Membentuk matriks keputusan.
d. Melakukan normalisasi pada elemen matriks keputusan
e. Menghitung nilai utilitas marjinal.

5. Menghitung nilai utilitas total setiap destinasi.
6. Menentukan 10 destinasi prioritas berdasarkan urutan nilai utilitas total tertinggi.
7. Membangun submatriks jarak antardestinasi.
8. Tahapan implementasi algoritma B&B untuk optimasi rute sebagai berikut:
   a. Inisialisasi: Horison Inn Antawirya Semarang sebagai *node* awal dan akhir, submatriks sebagai jarak antardestinasi, antrean Q, dan *minimal lenght* = ∞
   b. Menghitung nilai *lower bound* pada *root node*
   c. Membentuk rute awal dari hotel menuju masing-masing destinasi.
   d. Menghitung $\hat{c}(i)$ untuk setiap rute awal.
   e. Memasukkan *node* ke dalam antrean Q.
   f. Memeriksa antrean Q kosong atau tidak. Jika tidak, memilih *node* dengan $\hat{c}(i)$terkecil dari antrean Q untuk dieksplorasi. Jika iya, proses berakhir.
   g. Memeriksa semua destinasi telah dikunjungi atau tidak. Jika iya, menghitung total *cost*. Jika tidak, kembali branching dan kembali ke langkah f.
   h. Memeriksa total *cost* ≥ *minimal length* atau tidak (total *cost* < *minimal length*). Jika iya, melakukan *pruning*. Jika tidak, memperbarui *minimal lenght*
   i. Proses ini berlanjut hingga seluruh jalur telah dieksplorasi atau dipangkas.
   j. Menghitung total jarak tempuh.
9. Tahapan implementasi NN untuk optimasi rute sebagai berikut:
   a. Inisialisasi: Horison Inn Antawirya Semarang sebagai *node* awal dan akhir, serta submatriks sebagai jarak antardestinasi.
   b. Menentukan destinasi-destinasi yang terhubung dengan hotel dan belum dikunjungi.
   c. Memilih destinasi terdekat dari hotel, kemudian tandai sebagai telah dikunjungi.
   d. Destinasi yang terpilih menjadi *node* awal selanjutnya.
   e. Memeriksa semua destinasi telah dikunjungi atau tidak. Jika iya, kembali ke *node* awal. Jika tidak, kembali memilih destinasi terdekat dari *node* awal saat ini.
   f. Menghitung total jarak tempuh rute.
10. Tahapan implementasi algoritma ACO untuk optimasi rute sebagai berikut:
    a. Inisialisasi: submatriks sebagai jarak antardestinasi dan parameter-parameter yang digunakan, yaitu: $y = 5, m = 10, \alpha = 1, \beta = 2, \rho = 0{,}5\ , \tau_{ij}(1) = 1, Q = 1$
    Penentuan parameter dilakukan dengan mengacu pada penelitian Dorigo *et al*. (1996) serta Dorigo dan Stützle (2004).
    b. Setiap semut ditempatkan dan memulai perjalanan dari *node* awal yaitu Hotel Horison Inn Antawirya Semarang.
    c. Menentukan visibilitas antardestinasi (η_{ij}).
    d. Menghitung probabilitas maisng-masing destinasi.
    e. Membangkitkan bilangan *random* (r).
    f. Setiap semut mengulangi langkah d. hingga seluruh destinasi dikunjungi satu kali dan kembali ke hotel.
    g. Jika semua semut sudah menyelesaikan satu rute perjalanan, hitung panjang rute perjalanan masing-masing semut.
    h. Memperbarui intensitas *pheromone global*.
    i. Jika iterasi maksimum belum tercapai, lakukan proses dari langkah d. menggunakan *intensitas pheromone* yang telah diperbarui.

j. Menghitung total jarak tempuh.

11. Membandingkan dan menginterpretasi kinerja algoritma.

4. HASIL DAN PEMBAHASAN

Analisis Deskriptif menyajikan gambaran umum data penelitian meliputi *mean*, standar deviasi, nilai minimum, dan maksimum. Tabel 1 menyajikan analisis deskriptif data penelitian yang berkaitan dengan *Rating, Reviews,* dan *Fee.*

Tabel 1. Analisis Deskriptif

	Rating	*Reviews*	*Fee*
Mean	4,46	18.613,75	16.500,00
Standar Deviasi	0,18	19.102,47	25.411,40
Nilai Minimum	4,00	949,00	0,00
Nilai Maksimum	4,80	69.718,00	120.000,00

Tabel 1 menunjukkan secara keseluruhan, kriteria *Rating* cenderung bersifat homogen, sedangkan kriteria *Reviews* dan *Fee* menunjukkan adanya variasi data antardestinasi wisata.

Tabel 2. Bobot dan Kriteria

Kriteria	Keterangan	Bobot
C_1	*Rating*	0,6111
C_2	*Reviews*	0,2778
C_3	*Fee*	0,1111

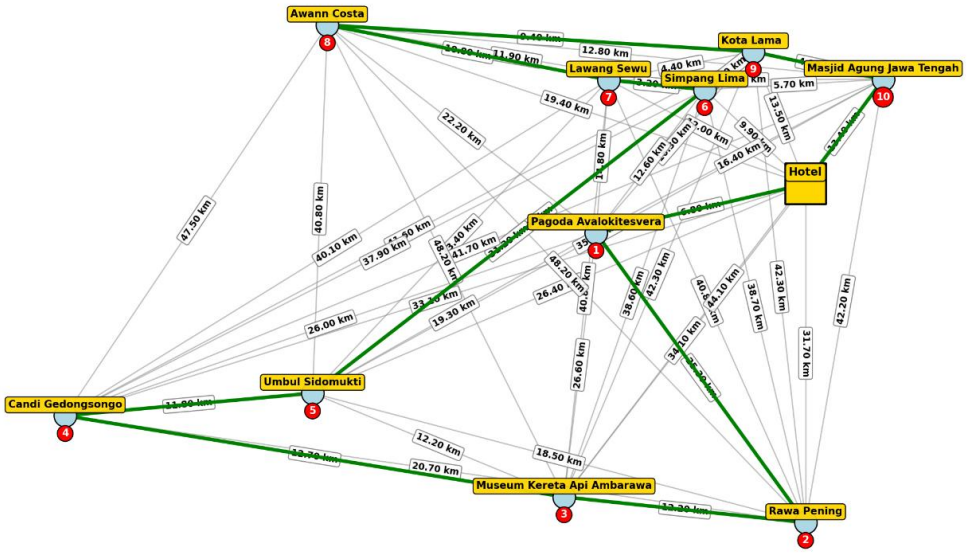
Tabel 2 menunjukan pembobotan kriteria dalam metode MAUT yang dilakukan menggunakan perhitungan metode ROC. Bobot yang dihasilkan digunakan dalam perhitungan nilai utilitas pada metode MAUT untuk memperoleh nilai utilitas total setiap destinasi wisata Semarang. Hasil perhitungan MAUT diurutkan sehingga diperoleh sepuluh destinasi wisata prioritas dengan nilai utilitas total tertinggi.

Tabel 3. Hasil Nilai Utilitas Total

Nama	u_i	*Rank*
Kota Lama (A_4)	0,7302	1
Simpang Lima (A_{20})	0,6530	2
Lawang Sewu (A_1)	0,6141	3
Masjid Agung Jawa Tengah (A_{19})	0,5453	4
Pagoda Avalokitesvera (A_5)	0,5229	5
Umbul Sidomukti (A_{10})	0,3574	6
Candi Gedongsongo (A_{14})	0,3533	7
Museum Kereta Api Ambarawa(A_{15})	0,3424	8
Awann Costa (A_{13})	0,2825	9
Rawa Pening (A_{24})	0,2824	10

Tabel 3 menunjukkan sepuluh destinasi wisata prioritas dengan nilai utilitas total tertinggi dari hasil perhitungan MAUT dengan pembobotan ROC. Selanjutnya, kesepuluh destinasi tersebut digunakan dalam pencarian rute wisata yang optimal, yaitu dengan jarak tempuh minimal menggunakanjalgoritma *Branch and Bound* (B&B), *Nearest Neighbor* (NN), dan *Ant Colony Optimization* (ACO).

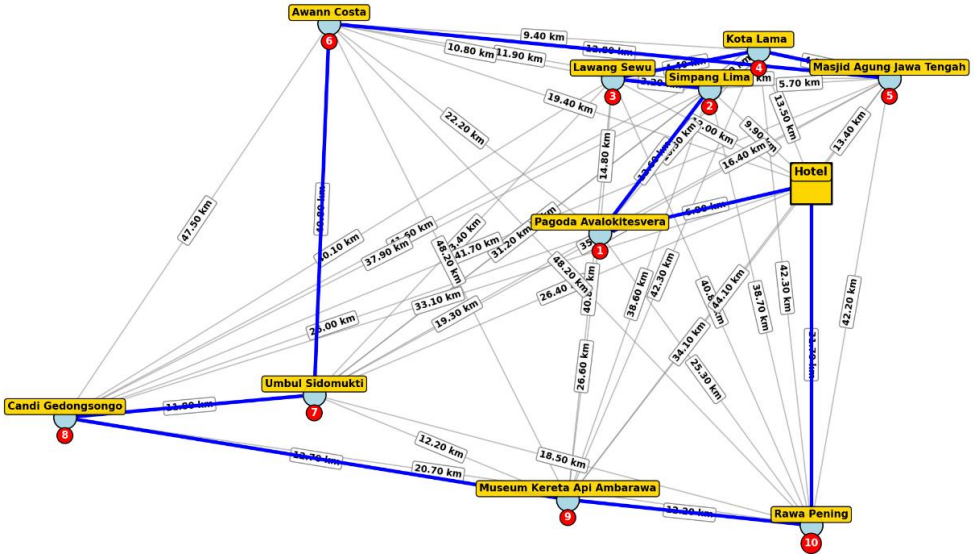
Algoritma B&B digunakan sebagai metode optimasi eksak untuk menentukan rute wisata dengan total jarak tempuh minimum. Algoritma ini mengevaluasi seluruh kemungkinan secara sistematis dengan menerapkan mekanisme percabangan dan pemangkasan berdasarkan nilai batas.



Gambar 1. Rute Optimal Algoritma *Branch and Bound*

Gambar 1 menampilkan visualisasi rute wisata hasil optimasi dengan algoritma *Branch and Bound*. Rute dimulai dan diakhiri di Hotel Horison Inn Antawirya Semarang. Rute yang dihasilkan, yaitu Hotel → Pagoda Avalokitesvera → Rawa Pening → Museum Kereta Api Ambarawa → Candi Gedongsongo→ Umbul Sidomukti → Simpang Lima→ Lawang Sewu → Awann Costa → Kota Lama → Masjid Agung Jawa Tengah → Hotel. Rute wisata hasil algoritma B&B menghasilkan total jarak tempuh sebesar 140,9 km dengan waktu komputasi 49,6643 detik.

Algoritma NN diterapkan sebagai metode heuristik yang menentukan rute wisata dengan total jarah tempuh minimum berdasarkan pemilihan destinasi terdekat yang belum dikunjungi dari lokasi saat ini.

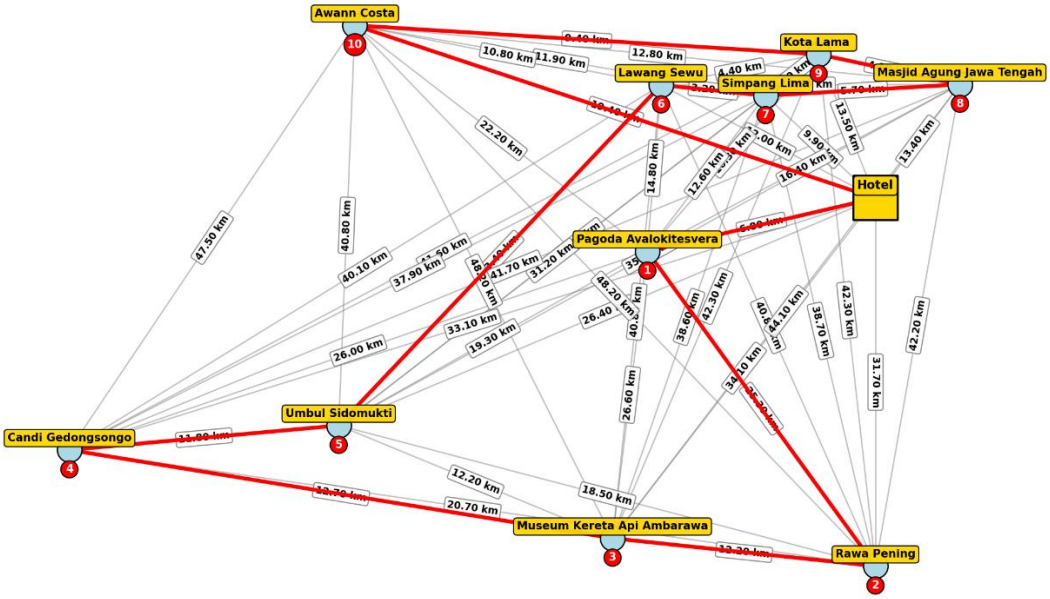


Gambar 2. Rute Optimal Algoritma *Nearest Neighbor*

Gambar 2 menampilkan visualisasi rute wisata hasil optimasi dengan algoritma *Nearest Neighbor*. Rute dimulai dan diakhiri di Hotel Horison Inn Antawirya Semarang. Rute wisata yang dihasilkan, yaitu Hotel → Pagoda Avalokitesvera → Simpang Lima → Lawang Sewu → Kota Lama → Masjid Agung Jawa Tengah → Awann Costa → Umbul Sidomukti →

Candi Gedongsongo → Museum Kereta Api Ambarawa → Rawa Pening → Hotel. Rute wisata hasil algoritma NN menghasilkan total jarak tempuh sebesar 153,1 km dengan waktu komputasi 0,0014 detik.

Algoritma ACO diterapkan sebagai metode optimasi metaheuristik yang mengadaptasi cara koloni semut menemukan jalur terpendek antara sarang dan sumber makanan. Algorima ini menerapkan mekanisme probabilistik dan pembaruan *pheromone* untuk menentukan rute wisata dengan total jarak tempuh minimum.



Gambar 3. Rute Optimal Algoritma *Ant Colony Optimization*

Gambar 3 menampilkan visualisasi rute wisata hasil optimasi dengan algoritma *Ant Colony Optimization*. Rute dimulai dan diakhiri di Hotel Horison Inn Antawirya Semarang. Rute wisata yang dihasilkan, yaitu Hotel → Pagoda Avalokitesvera → Rawa Pening → Museum Kereta Api Ambarawa → Candi Gedongsongo → Umbul Sidomukti → Lawang Sewu → Simpang Lima → Masjid Agung Jawa Tengah → Kota Lama → Awann Costa → Hotel. Rute wisata hasil algoritma ACO menghasilkan total jarak tempuh sebesar 144 km dengan waktu komputasi 0,0331 detik.

Perbandingan kinerja algoritma dilakukan berdasarkan tiga indikator utama, yaitu total jarak tempuh, waktu komputasi, dan *Relative Error* (RE) terhadap solusi optimal yang dihasilkan oleh algoritma *Branch and Bound*.

Tabel 4. Perbandingan Kinerja Algoritma

Algoritma	Total Jarak Tempuh	Waktu Komputasi	*Relative Error* (RE)
Branch and Bound	140,9 km	49,6643 detik	0
Nearest Neighbor	153,1 km	0,0014 detik	8,6586%
Ant Colony Optimization	144,0 km	0,0331 detik	2,2001%

Hasil perbandingan pada Tabel 4 menunjukkan perbedaan kinerja dalam optimasi rute wisata Semarang. Pada penelitian ini, algoritma B&B merupakan metode eksak yang dapat dijamin menghasilkan solusi optimal, tetapi memerlukan waktu komputasi yang lebih lama. Algoritma B&B menghasilkan rute wisata optimal dengan total jarak tempuh minimum sebagai acuan pembanding untuk perhitungan *Relative Error*. Algoritma NN unggul dari sisi kecepatan komputasi, tetapi menghasilkan *Relative Error* yaitu sebesar 8,6586% terhadap solusi optimal. Algorima ACO mampu menghasilkan solusi yang lebih mendekati solusi

optimal ditunjukkan dengan *Relative Error* yang lebih kecil yaitu sebesar 2,2001%, tetapi membutuhkan waktu komputasi yang lebih lama dibandingkan dengan algoritma NN.

5. KESIMPULAN

Metode MAUT mampu menentukan destinasi wisata prioritas di Kota Semarang dan Kabupaten Semarang secara objektif dengan mempertimbangkan beberapa kriteria yang ditentukan sehingga destinasi yang terpilih untuk dilanjutkan ke pencarian rute wisata optimal terjamin kualitas destinasinya. Berdasarkan kinerja algoritma NN dan ACO menunjukkan bahwa keduanya mampu menghasilkan rute yang *feasible*, dengan algoritma NN lebih unggul pada sisi waktu komputasi, tetapi menghasilkan jarak tempuh lebih panjang dan RE lebih besar. Algoritma ACO menghasilkan rute lebih pendek dengan RE lebih kecil, tetapi membutuhkan waktu komputasi yang lebih lama. Hasil perbandingan ini menunjukkan adanya *trade-off* antara kualitas solusi dan waktu komputasi. Jika kualitas solusi lebih diprioritaskan dibandingkan waktu komputasi, algoritma ACO memberikan kinerja paling mendekati solusi optimal karena menghasilkan RE hanya 2,2001% terhadap solusi optimal algoritma B&B.

DAFTAR PUSTAKA

Abbaspour, S., Aghsami, A., Jolai, F., dan Yazdani, M. 2022. An Integrated Queueing-Inventory-Routing Problem in a Green Dual-Channel Supply Chain Considering Pricing and Delivery period : a Case Study of Construction. *Journal of Computational Design and Engineering* Vol. 9, No.5, Hal: 1917–1951. DOI: 10.1093/jcde/qwac089.

Apriyani, D. C. N. 2024. *Matematika Diskret (Graf)*. Pacitan: LPPM Press STKIP PGRI.

Aqila, Rangkuti, Y. M., Karo, I. M. K., dan Al-Idrus, S. I. 2024. Determining Tourist Destination Priorities Using Website-Based Particle Swarm Optimization Methods (Case Study : North Padang Lawas Regency). *Journal of Mathematics, Computations and Statistics* Vol. 7, No. 2, Hal: 371–379. DOI: 10.35580/jmathcos.v7i2.4233.

Dorigo, M., Maniezzo, V., dan Colorni, A. 1996. The Ant System : Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetic* Vol. 26, No. 1, Hal: 1–13.

Gao, W. 2020. New Ant Colony Optimization Algorithm for the Traveling. *International Journal of Computational Intelligence Systems* Vol. 13, No. 1, Hal: 44-55. DOI: 10.2991/ijcis.d.200117.001.

Gunawan, I., Tambunan, H. S., dan Hartama, D. 2022. *Monograf Algoritma Tabu Search dalam Kasus Traveling Salesman Problem*. Indramayu: Penerbit Adab.

Ilhami, L. T. A., Ashari, M., dan Fadli, S. 2024. Sistem Pendukung Keputusan Seleksi Pemain Basket Menggunakan Metode SAW dengan Perbobotan ROC. *Journal of Social Science Research* Vol. 4, No. 3, Hal: 3073–3088.

Isnaeny, S. N., Puspita, D. D., dan Euigie, A. N. 2025. Analisis Algoritma Nearest Neighbor , Branch and Bound dan Dijkstra dalam Penyelesaian Travelling Salesman Problem (TSP). *Universitas Negeri Malang*

Jaelani, I. 2016. Penerapan Algoritma Semut Untuk Optimasi Rute Penjemputan Barang Pada Tempat Jasa Penitipan Sementara Lion Express. *Jurnal Ilmiah Sarjana Mahasiswa UMRAH* Vol. *1*, Hal: 1–14.

Karjono, Moedjiono, dan Kurniawan, D. 2016. Ant Colony Optimization. *Jurnal TICOM* Vol. 4, No. 3, Hal: 119-125.

Khan, A. A., dan Agrawal, H. 2016. A Comparative Study od Nearest Neighbour Algorithm and Genetic Algorithm in Solving Traveling Salesman Problem. *International Research Journal of Engineering and Technology (IRJET)* Vol. 3, No. 5, Hal: 234–238.

Klook Indonesia. 2025. *24 Rekomendasi Tempat Wisata di Semarang yang Wajib Kamu Kunjungi*. Tersedia: https://www.klook.com/id/blog/rekomendasi-tempat-wisata-di-semarang/ (diakses pada tangga 4 November 2025).

Margiyani, S., dan Mussafi, N. 2014. Aplikasi Algoritma Branch and Bound Untuk Optimasi Jalur Pemadam Kebakaran Kota Yogyakarta. *Jurnal Fourier* Vol. 3, No. 1, Hal: 59–66.

Munir, R. 2009. *Matematika Diskrit* (Edisi Ketiga). Bandung: Informatika Bandung

Munir, R. 2015. *Metode Numerik*. Bandung: Informatika Bandung /

Mutakhiroh, I., Saptono, F., Hasanah, N., dan Wiryadinata, R. 2007. Pemanfaatan Metode Heuristik dalam Pencarian Jalur Terpendek dengan Algoritma Semut dan Algoritma Genetika. *Prosiding Seminar Nasional Aplikasi Teknologi Informasi 2007*, Universitas Islam Indonesia: 16 Juni 2007.

Nuroji, N. 2022. Penerapan Multi-Attribute Utility Theory (MAUT) Dalam Penentuan Pegawai Terbaik. *Jurnal Ilmiah Informatika Dan Ilmu Komputer* Vol. 1, No. 2, Hal: 46-53. DOI: 10.58602/jima-ilkom.v1i2.7.

Prasetio, S. A. J., dan Suseno. 2024. Optimasi Rute dengan Metode Ant Colony Optimization dan Nearest Neighbor di Perusahaan Yellow Moon Production. *Jurnal Multidisiplin Ilmu Akademik* Vol. 2, No. 6, Hal: 619–632. DOI:10.61722/jmia.v1i3.1621.

Putra, A. G. P. H., Solikah, E. A., dan Nisak, S. A. 2023. Penerapan Algoritma Nearest Neighbor dalam Permasalahan TSP untuk Menentukan Rute Terpendek Pendistribusian Krupuk Rengginang. *Jurnal Pendidikan Matematika* Vol. 10, No. 2, Hal: 59–68. DOI: 10.18592/jpm.v10i2.9756.

Sembiring, M. T., dan Chailes, S. 2020. Ant Colony Optimization Implementation on Traveling Salesman Problem to Achieve the Shortest Logistic Route. *2nd International Conference on Industrial and Manufacturing Engineering*, Universitas Sumatera Utara: 3-4 September 2020.

Sholihaningtias, D. N. 2023. Penerapan Kombinasi Metode MAUT dan ROC Dalam Seleksi Karyawan. *Techno.Com* Vol. 22, No. 1, Hal: 145–155. DOI:10.33633/tc.v22i1.7480.

Wira, A. 2019. Sistem Pendukung Keputusan Pemilihan PimpinanDengan MetodeMulti Attribute Utility Theory (MAUT) di PT.Sagami Indonesia. *Jurnal Mantik* Vol. 3, No. 2, Hal: 10–19.

Witin, S. S. G., Permatasari, W. D., Aminah, N., Rande, P., Amijaya, F. D. T., dan Putri, D. F. 2025. Penerapan Algoritma Branch and Bound untuk Optimasi Rute Wisata Kalimantan Timur Berdasarkan Traveling Salesman Problem. *Jurnal Ilmiah Matematika* Vol. 13, No. 2, Hal: 198–199.

Zukhruf, F., dan Frazila, R. B. 2021. *Pengantar Optimasi dalam Rekayasa Transportasi*. Bandung: ITB Press.