

IMPLEMENTASI ALGORITMA ANT COLONY DALAM MENENTUKAN RUTE TERPENDEK DARI JOHOR KE UNIVERSITAS NEGERI MEDAN

Bunga Dwi Febrianti, Muhammad Naufal Musyaafa, Khildan Rifail Azis, Putri Harliana

Ilmu Komputer, Universitas Negeri Medan
Jalan William Iskandar Ps. V Medan, Indonesia
bungadwifebrianti@gmail.com

ABSTRAK

Penelitian ini bertujuan untuk mengimplementasikan algoritma Ant Colony Optimization (ACO) menggunakan python dalam menentukan rute terpendek dari Komplek Bukit Johor Mas menuju Universitas Negeri Medan (UNIMED). Representasi jalur dilakukan melalui graf berbobot, dengan pengukuran jarak antar simpul menggunakan Google Earth. Pengujian parameter ACO mencakup α (α), β (β), evaporation rate (ρ), dan jumlah semut. Hasil menunjukkan bahwa pengaturan optimal parameter, seperti $\alpha = 1$, $\beta = 2$, $\rho = 0.5$, dan jumlah semut sebanyak 100, menghasilkan rute terpendek sejauh 12,61 km. Selain itu, variasi parameter memberikan pengaruh signifikan terhadap hasil yang diperoleh, dimana eksplorasi parameter yang tepat dapat menghindari algoritma dari solusi lokal. Penggunaan Python sebagai platform implementasi terbukti efisien berkat dukungan pustaka seperti NumPy, serta kemudahan implementasi algoritmanya. Penelitian ini memberikan kontribusi dalam mempermudah analisis rute transportasi dengan menggunakan pendekatan berbasis teknologi, sekaligus menjadi acuan bagi pengembangan lebih lanjut di bidang optimasi rute dan transportasi berbasis graf.

Kata kunci : Algoritma Ant Colony, Python, Rute Terpendek, Optimasi Transportasi, Graf

1. PENDAHULUAN

Dalam era modern ini, perkembangan teknologi informasi maupun komputasi telah memberikan dampak signifikan dalam berbagai aspek kehidupan, termasuk dalam optimasi system transportasi. Salah satu tantangan utama dalam transportasi yaitu menentukan jalur terpendek yang dapat menghemat waktu, biaya, dan sumber daya. Masalah ini sering ditemui dalam aplikasi sehari-hari seperti navigasi, logistic, dan transportasi umum.

Pada penelitian sebelumnya [1] dalam jurnal berjudul "Penerapan Algoritma Ant Colony Optimization untuk Pencarian Rute Terpendek Lokasi Wisata" menggunakan PHP. Yang dimana jurnal ini telah menunjukkan potensi penggunaan Algoritma Ant Colony Optimization (ACO) dalam menyelesaikan permasalahan pencarian rute terpendek. Berdasarkan hasil tersebut, penelitian ini mengadaptasi penerapan algoritma ACO. Tetapi, penelitian ini mengharapkan adanya pengembangan penelitian lebih menggunakan lanjut proses dengan perbandingan dengan banyak algoritma menggunakan bahasa program lainnya, sehingga dapat membedakan manakah algoritma yang lebih baik dalam menentukan rute terpendek.

Dan pada penelitian dari [2] dalam jurnal berjudul "Pencarian Rute Terpendek Menggunakan Algoritma Ant Colony Optimization Pada Gui Matlab Guna Memantau Sustainable Development Goals" menjelaskan bagaimana Permasalahan *Travelling Salesman Problem* mencari rute terpendek atau terdekat menggunakan algoritma Ant Colony Optimization. Dengan mengacu pada jurnal sebelumnya yang menyelesaikan masalah TSP menggunakan ACO, penelitian ini bertujuan untuk memperluas aplikasi ACO, khususnya untuk kasus penentuan rute terdekat antara simpul awal dan simpul

tujuan. Eksperimen ini diharapkan dapat memberikan wawasan baru tentang penyesuaian parameter ACO untuk mendapatkan solusi optimal pada graf tertentu.

Sebagai alat bantu pengembangan algoritma, Python menjadi Bahasa pemrograman yang ideal karena memiliki berbagai Pustaka (library) yang mendukung penerapan ACO, seperti NumPy untuk perhitungan numerik. Python juga memiliki keunggulan dalam fleksibilitas dan efisiensi yang membuatnya cocok digunakan dalam pemrograman ilmiah dan riset optimasi. Dengan kemampuan Python, implementasi ACO dapat dilakukan secara efisien dan terstruktur, mulai dari simulasi perilaku semut hingga analisis hasil akhir.

Studi ini bertujuan untuk menghadirkan pendekatan berbasis algoritma ACO yang mampu menganalisis jaringan jalan menuju kampus secara efektif. Dengan menggunakan Bahasa pemrograman python, penelitian ini tidak hanya mempermudah proses analisis data tetapi juga menyediakan Solusi berbasis teknologi yang dapat membantu meningkatkan aksesibilitas ke kampus. Hasil penelitian ini diharapkan dapat memberikan kontribusi praktis dalam mendukung mobilitas di lingkungan Pendidikan tinggi serta menjadi referensi bagi pengembangan system transportasi berbasis teknologi dimasa depan.

2. TINJAUAN PUSTAKA

Pada penelitian [3] Pencarian lintasan terpendek pada umumnya dibagi menjadi dua metode, yaitu metode konvensional dan metode heuristik. Pada metode heuristik terdapat beberapa algoritma, salah satunya adalah algoritma Ant Colony Optimization. Algoritma ini didasarkan pada perilaku semut. Secara alamiah, koloni semut dapat menemukan rute terpendek dalam perjalanan dari sarang ke tempat

sumber makanan. Koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada jalur yang dilalui.

2.1. Optimisasi

Optimisasi merupakan suatu proses untuk mencapai hasil yang ideal atau optimal (nilai efektif yang dapat dicapai). Dalam disiplin matematika optimisasi merujuk pada studi permasalahan yang mencoba untuk mencari nilai minimal atau maksimal dari suatu fungsi nyata. Untuk dapat mencapai nilai optimal baik minimal maupun maksimal tersebut, secara sistematis dilakukan pemilihan nilai variabel integer atau nyata yang akan memberikan solusi optimal [4].

2.2. Ant Colony Optimization (ACO)

Pada penelitian [4] dijelaskan bahwa Ant colony optimization (ACO) adalah pendekatan metaheuristik yang terinspirasi dari tingkah laku semut dalam mencari sumber makanan. ACO diusulkan oleh Marco Dorigo sebagai metode untuk menyelesaikan kombinasi dari permasalahan optimisasi [5] dan untuk menyelesaikan masalah komputasi dengan menemukan jalur atau rute terbaik melalui graf. Solusi dinyatakan sebagai jalur biaya minimum (terpendek) melalui kondisi masalah sesuai dengan kendala masalah. Kompleksitas setiap semut berbeda bahkan seekor semut pun dapat menemukan solusi (kemungkinan berkualitas buruk. Solusi berkualitas tinggi hanya ditemukan sebagai hasil yang muncul dari kerja sama global antara semua koloni secara bersamaan membangun solusi yang berbeda.

Algoritma semut diperkenalkan oleh *Moyson* dan *Manderick* dan secara meluas dikembangkan oleh *Marco Dorigo*. Algoritma semut adalah bioinspired metaheuristic, mempunyai sekelompok khusus yang berusaha karakteristik kelakuan dari serangga sosial, yaitu koloni semut. Kelakuan dari tiap pelaku dalam meniru kelakuan dari semut hidup dan bagaimana mereka berinteraksi satu dengan lainnya agar dapat menemukan sumber makanan dan membawanya ke koloni mereka dengan efisien. Selama berjalan tiap semut mengeluarkan *feromon*, dimana semut lainnya sensitif dengan *feromon* tersebut sehingga memberikan harapan untuk mengikuti jejaknya. Lebih atau kurang intensitasnya tergantung pada konsentrasi dari *feromon*. Setelah beberapa waktu, jalur terpendek akan lebih sering diikuti dan feromonnya menjadi jenuh. [1].

2.3. Rute Jalur Terpendek

Rute jalur terpendek bisa dikatakan sebagai nilai minimum dari suatu lintasan. Permasalahan yang dihadapi dalam menentukan jalur terpendek yakni banyaknya rute yang mungkin dipilih dari tempat keberangkatan menuju tempat tujuan akhir. Dalam ilmu matematika diskrit, graf dapat digunakan untuk menunjukkan jarak antar kota atau tempat [5].

Secara umum, ada dua metode yang dapat digunakan untuk menyelesaikan masalah pencarian rute jalur terpendek yaitu metode algoritma konvensional dan metode heuristik. Metode konvensional adalah algoritma yang menggunakan perhitungan matematika biasa dan cenderung menghasilkan solusi optimal dengan mengikuti langkah yang telah terdefinisi. Algoritma seperti Dijkstra, Floyd Warshall, dan Bellman-Ford adalah beberapa metode konvensional yang biasa digunakan untuk mencari jalur terpendek. Metode heuristik adalah perhitungan yang menghasilkan solusi alternatif untuk menemukan jalur terpendek dan berfokus dalam menemukan solusi yang memadai atau mendekati solusi optimal. Ada beberapa algoritma dalam heuristik yang biasa digunakan dalam mencari jalur terpendek. Salah satunya adalah algoritma ant colony optimization.

2.4. Python

Python merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaan besar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis desktop, web dan mobile. Python diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televisi kesukaan Guido Monty Python's Flying Circus. Van Rossum Kepopuleran penggunaan Python menempatkannya menjadi bahasa pemrograman yang mulai banyak dipelajari oleh mahasiswa terutama mahasiswa di kampus yang berbasis IT, guna menyelesaikan tugas kuliah, tugas akhir maupun tugas penelitian, untuk dapat menyelesaikan berbagai tugas pemrograman, seseorang perlu memahami algoritma, karena pada dasarnya program computer adalah implementasi dari algoritma. mengembangkan Python sebagai hobi, kemudian Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak intuitif dan memiliki pustaka yang luas [6].

2.5. Graf

Pada penelitian [8] Sejarah teori graf bermula saat ahli matematika Swiss Leonhard Euler memecahkan masalah jembatan Königsberg. Masalah jembatan Königsberg adalah teka teki lama mengenai kemungkinan menemukan jalan setapak di tujuh jembatan yang membentang di sepanjang sebuah sungai bercabang yang melewati sebuah pulau tanpa melewati jembatan yang sama dua kali. Euler berpendapat bahwa jawabannya adalah tidak ada. Buktinya hanya mengacu pada susunan fisik jembatan, namun intinya dia membuktikan teorema pertama dalam teori graf. Secara matematis graf didefinisikan sebagai pasangan himpunan simpul dan sisi. Jika graf disimbolkan dengan G , himpunan simpul-simpul yang terdapat dalam graf disimbolkan dengan $V = \{v_1, v_2, v_3, \dots, v_n\}$, dan himpunan sisi yang menghubungkan sepasang simpul dalam graf disimbolkan dengan $E = \{$

$e_1, e_2, e_3, \dots, e_n$, maka graf dapat ditulis dengan notasi matematis = (V, E) .

3. METODE PENELITIAN

Metode penelitian ini berbasis algoritma *metaheuristic*, yaitu *Ant Colony Optimization* (ACO), untuk menyelesaikan masalah optimasi rute terpendek. Terdapat x tahapan pada penelitian ini, diantaranya adalah representasi masalah, implementasi algoritma ACO, dan pengujian parameter

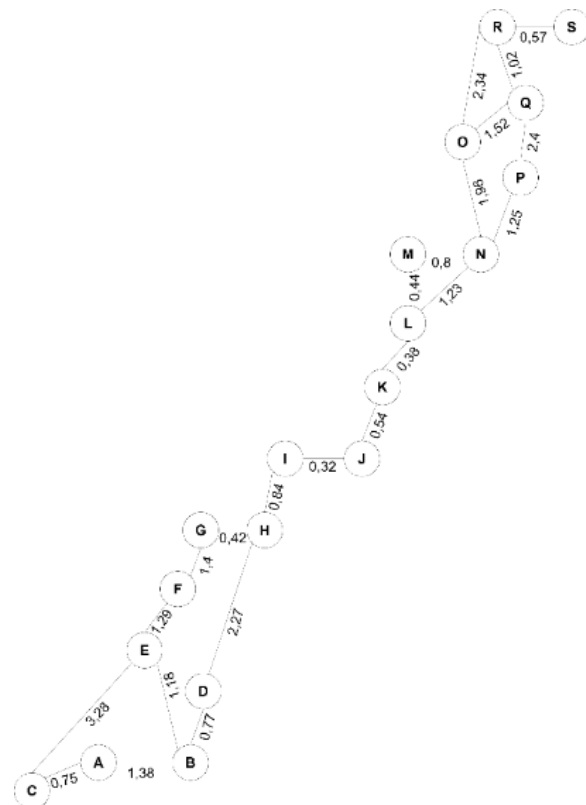
3.1. Representasi Masalah

Masalah yang dihadapi dalam penelitian ini adalah menemukan rute tercepat dari Komplek Bukit Johor Mas menuju Universitas Negeri Medan (UNIMED). Pengukuran jarak antar jalan dilakukan menggunakan alat bantu Google Earth, dengan cara mengobservasi panjang jalan antara dua titik tertentu yang merepresentasikan hubungan langsung antar simpul sebagaimana yang ditunjukkan pada Gambar 2 di bawah ini.



Gambar 1. Pengukuran menggunakan Google Earth

Jalur antara kedua lokasi tersebut direpresentasikan dalam bentuk graf berbobot. Graf ini terdiri atas simpul-simpul (*nodes*) yang merepresentasikan jalan, serta sisi-sisi (*edges*) yang menghubungkan simpul-simpul tersebut, yang merepresentasikan jarak antar jalan. Seperti yang dapat dilihat pada gambar ini.



Gambar 2. Representasi jalur dalam bentuk graf

Keterangan:

- A. Komplek Bukit Johor Mas
- B. Jl. Karya Jaya
- C. Jl. Karya Wisata
- D. Jl. M. Basir
- E. Jl. Jenderal Besar A.H. Nasution
- F. Jl. Adi Sucipto
- G. Jl. Avros
- H. Jl. Brigjend Katamso
- I. Jl. Pelangi
- J. Jl. Sisingamangaraja
- K. Jl. Stadion
- L. Jl. HM. Joni
- M. Jl. Halat
- N. Jl. Arief Rahman Hakim
- O. Jl. Aksara
- P. Jl. Denai
- Q. Jl. Jl. Mandala By Pass
- R. Jl. William Iskandar Ps. V
- S. Gerbang 2 UNIMED

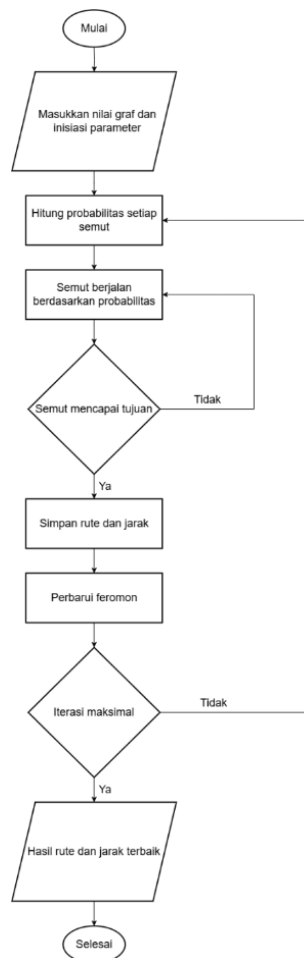
Metode ini dipilih karena memungkinkan pengukuran jarak dengan tingkat akurasi yang cukup tinggi dan kemudahan akses untuk memetakan jalur. Setiap simpul dan sisi dalam graf dirancang untuk mencerminkan kondisi sebenarnya dari rute transportasi antara Komplek Bukit Johor Mas yang terletak di kecamatan Medan Johor dan UNIMED. Tabel matriks ketetanggaan berikut menunjukkan daftar jalur yang digunakan beserta bobot (jarak) setiap sisi:

Gambar 3. Matriks Ketetanggan

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
A	0	1.38	0.75	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
B	1.38	0	∞	0.77	1.18	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
C	0.75	∞	0	∞	3.28	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
D	∞	0.77	∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
E	∞	1.18	3.28	∞	0	1.29	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
F	∞	∞	∞	∞	1.29	0	1.4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
G	∞	∞	∞	∞	∞	1.4	0	0.42	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
H	∞	∞	∞	∞	∞	∞	0.42	0	0.84	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
I	∞	∞	∞	∞	∞	∞	∞	0.84	0	0.32	∞	∞	∞	∞	∞	∞	∞	∞	∞
J	∞	∞	∞	∞	∞	∞	∞	∞	0.32	0	0.54	∞	∞	∞	∞	∞	∞	∞	∞
K	∞	∞	∞	∞	∞	∞	∞	∞	∞	0.54	0	0.38	∞	∞	∞	∞	∞	∞	∞
L	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0.38	0	0.44	1.23	∞	∞	∞	∞	∞
M	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0.44	0	0.8	∞	∞	∞	∞	∞
N	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1.23	0.8	0	1.96	1.25	∞	∞	∞
O	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1.96	0	∞	1.52	2.34	∞
P	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1.25	∞	0	2.4	∞	∞
Q	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1.52	2.4	0	1.02	∞
R	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	2.34	∞	1.02	0	0.57
S	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0.57	0

3.2. Implementasi Algoritma ACO

Setelah inisiasi rute dalam bentuk graf, Graf direpresentasikan sebagai struktur data berbentuk dictionary, di mana setiap simpul memiliki tetangga dengan bobot jalur tertentu yang menunjukkan jarak. Berikut merupakan flowchart dari algoritma ACO.



Gambar 3. Flowchart

Setelah inisialisasi dalam bentuk dictionary, algoritma memasuki iterasi utama yang dijalankan sebanyak jumlah iterasi yang telah ditentukan. Pada setiap iterasi, seluruh semut melakukan perjalanan dari simpul awal menuju simpul tujuan. Perjalanan setiap semut dimulai dengan pemilihan simpul tetangga berdasarkan probabilitas yang dihitung menggunakan kombinasi nilai feromon dan heuristik. Probabilitas dihitung dengan rumus:

$$P_{(i,j)} = \frac{\tau_{(i,j)}^{\alpha} \cdot h_{(i,j)}^{\beta}}{\sum \tau_{(i,u)}^{\alpha} \cdot h_{(i,u)}^{\beta}}$$

dimana:

$p_{(i,j)}$ = nilai probabilitas untuk pemilihan rute

α = derajat kepentingan feromon

β = derajat kepentingan visibility

$h_{(i,j)}$ = nilai visibility (1/jarak ruas ij)

Simpul tetangga dipilih secara probabilistik berdasarkan nilai $p_{(i,j)}$, sehingga jalur yang memiliki probabilitas lebih tinggi lebih cenderung dipilih, tetapi semut tetap memiliki peluang untuk mengeksplorasi jalur lain.

Setiap semut mencatat rute yang dilaluinya beserta jarak total yang ditempuh. Jika semut berhasil mencapai simpul tujuan, rutenya disimpan dan dibandingkan dengan rute terbaik yang ditemukan sejauh ini. Setelah semua semut selesai melakukan perjalanan pada iterasi tersebut, nilai feromon pada jalur diperbarui. Proses pembaruan feromon melibatkan dua tahap: penguapan feromon untuk mencegah dominasi jalur tertentu dan penambahan feromon baru pada jalur yang dilewati oleh semut. Penguapan dilakukan untuk mencegah dominasi jalur tertentu dan dihitung menggunakan rumus:

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j}$$

dimana:

ρ = parameter tingkat penguapan pada ruas yang telah dilalui oleh semut k.

Penambahan feromon dilakukan pada jalur yang dilewati oleh semut, proporsional dengan kebalikan dari jarak rute (L_k), menggunakan rumus:

$$\Delta\tau_{i,j}^{(k)} = \begin{cases} Q / L_k \\ 0 \end{cases}$$

dimana:

Q = konstanta yang ditentukan oleh peneliti

L^k = total jarak yang dilalui semut k hingga kembali ke tempat asal

$\Delta\tau_{i,j}^{(k)} = Q / L_k$ apabila (i,j) merupakan lintasan yang terbaik

$\Delta\tau_{i,j}^{(k)} = 0$ untuk lintasan lainnya

Penambahan feromon ini proporsional terhadap kebalikan dari jarak rute, sehingga jalur dengan jarak lebih pendek mendapatkan penambahan feromon yang lebih besar. Proses iteratif ini berlangsung hingga semua iterasi selesai. Pada akhir iterasi, algoritma menghasilkan rute terbaik dan jarak terpendek yang ditemukan oleh semut selama proses pencarian. Hasil akhir ini digunakan untuk menyelesaikan permasalahan rute terpendek dengan pendekatan ACO.

3.3. Pengujian Parameter

Dalam penelitian ini, pengujian parameter dilakukan untuk memahami pengaruh setiap parameter algoritma Ant Colony Optimization (ACO) terhadap hasil pencarian rute terpendek. Parameter yang diuji meliputi α (α), β (β), evaporation rate (ρ), jumlah semut, dan jumlah iterasi. Parameter α menentukan seberapa besar pengaruh feromon terhadap pemilihan jalur oleh semut, sementara β mengontrol pengaruh heuristik (jarak). Pengujian dilakukan dengan mengubah satu parameter pada satu waktu (one-variable-at-a-time), sementara parameter lainnya tetap konstan pada nilai awal (default). Nilai default yang digunakan adalah sebagai berikut:

- Alpha (α): 1
- Beta (β): 2
- Evaporation Rate (ρ): 0.5
- Jumlah Semut: 10

Pengujian dilakukan dengan tiga variasi nilai untuk setiap parameter, dengan variasi seperti yang ditunjukkan pada tabel di bawah ini:

Tabel 2. Parameter Pengujian

Parameter yang diubah	Nilai yang diuji
Alpha (α)	0.5, 1.5, 2.5
Beta (β)	1, 3, 5
Evaporation Rate (ρ)	0.3, 0.7, 0.9
Jumlah Semut	5, 50, 100

4. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan variasi parameter α (α), β (β), evaporation rate (ρ), jumlah semut, dan jumlah iterasi. Setiap parameter diuji dengan nilai yang berbeda, sedangkan parameter lainnya dijaga tetap. Berikut adalah hasil pengujian untuk variasi tiap parameter.

4.1. Pengaruh Veromon (Alpha, α)

Pengujian dilakukan dengan variasi α (0.5, 1.5, 2.5), sementara parameter lainnya tetap: $\beta = 2.0$, $\rho = 0.5$, dan jumlah semut = 10. Tabel dibawah menunjukkan hasil dari pengujian yang dilakukan.

Tabel 3. Pengaruh pada Veromon

Pengaruh Veromon (α)	Rute yang Dihasilkan	Jarak yang Dihasilkan
1 (default)	A-C-E-F-G-H-I-J-K-L-M-N-P-Q-O-R-S	18.54
0.5	A-B-E-F-G-H-I-J-K-L-M-N-O-Q-R-S	14.06
1.5	A-C-E-F-G-H-I-J-K-L-M-N-P-Q-O-R-S	18.54
2.5	A-C-E-F-G-H-I-J-K-L-M-N-P-Q-O-R-S	18.54

Berdasarkan hasil pengujian, terlihat bahwa variasi nilai α (α) memengaruhi pemilihan jalur oleh semut. Pada nilai default $\alpha = 1$, semut menghasilkan rute dengan jarak 18.54 km. Ketika nilai α diturunkan menjadi 0.5, jarak rute berkurang menjadi 14.06 km. Hal ini menunjukkan bahwa pengaruh feromon yang lebih rendah memungkinkan eksplorasi jalur yang lebih baik, sehingga menghasilkan solusi yang lebih optimal. Namun, ketika nilai α dinaikkan menjadi 1.5 atau 2.5, jarak rute kembali ke 18.54 km, yang menunjukkan bahwa pengaruh feromon yang terlalu besar membuat semut lebih terfokus pada jalur dengan feromon tinggi, sehingga eksplorasi jalur baru berkurang. Ini menandakan bahwa nilai α yang terlalu besar dapat menyebabkan algoritma terjebak pada solusi lokal.

4.2. Pengaruh Heuristik (Beta, β)

Pengujian dilakukan dengan variasi β (1, 3, 5), sementara parameter lainnya tetap: $\alpha = 1$, $\rho = 0.5$, dan jumlah semut = 10. Tabel dibawah menunjukkan hasil dari pengujian yang dilakukan.

Tabel 4. Pengaruh pada Heuristik

Pengaruh Heuristik	Rute yang Dihasilkan	Jarak yang Dihasilkan
2 (default)	A-C-E-F-G-H-I-J-K-L-M-N-P-Q-O-R-S	18.54
1	A-B-E-F-G-H-I-J-K-L-M-N-O-Q-R-S	14.06
3	A-B-D-H-I-J-K-L-M-N-O-Q-R-S	12.81
5	A-B-E-F-G-H-I-J-K-L-M-N-O-Q-R-S	14.06

Hasil pengujian menunjukkan bahwa variasi β (β) memiliki dampak signifikan terhadap hasil rute. Pada nilai default $\beta = 2$, jarak rute adalah 18.54 km. Ketika β diturunkan menjadi 1, jarak rute menjadi lebih pendek, yaitu 14.06 km, menunjukkan bahwa semut lebih mengandalkan informasi feromon dan eksplorasi jalur baru yang lebih terbatas. Sebaliknya, pada $\beta = 3$, jarak rute menjadi 12.81 km, yang merupakan hasil terbaik dalam pengujian ini. Namun,

ketika β dinaikkan menjadi 5, jarak rute kembali menjadi 14.06 km, menandakan bahwa pengaruh heuristik yang terlalu besar dapat mengurangi optimalisasi hasil rute terdekat. Dengan demikian, nilai β yang optimal ditemukan pada 3.

4.3. Pengaruh Evaporation Rate (Rho, ρ)

Pengujian dilakukan dengan variasi evaporation rate (0.3, 0.7, 0.9), sementara parameter lainnya tetap: $\alpha = 1$, $\beta = 2.0$, dan jumlah semut = 10. Tabel dibawah menunjukkan hasil dari pengujian yang dilakukan.

Tabel 5. Pengaruh pada Evaporation Rate

Pengaruh Tingkat Evaporasi	Rute yang Dihasilkan	Jarak yang Dihasilkan
0.5 (default)	A-C-E-F-G-H-I-J-K-L-M-N-P-Q-O-R-S	18.54
0.3	A-C-E-F-G-H-I-J-K-L-M-N-P-Q-O-R-S	18.54
0.7	A-C-E-F-G-H-I-J-K-L-M-N-P-Q-O-R-S	18.54
0.9	A-C-E-F-G-H-I-J-K-L-M-N-P-Q-O-R-S	18.54

Pengujian variasi evaporation rate (ρ) menunjukkan bahwa perubahan nilai ini tidak selalu memberikan dampak signifikan pada jarak rute. Pada nilai default $\rho = 0.5$, jarak rute yang dihasilkan adalah 18.54 km. Pada nilai $\rho = 0.3$ dan 0.7, hasil rute tetap sama, yaitu 18.54 km, yang menunjukkan bahwa laju penguapan feromon pada kisaran ini tidak cukup besar untuk memengaruhi pemilihan jalur. Namun, pada $\rho = 0.9$, jarak rute masih sama, menandakan bahwa jalur yang optimal sudah ditemukan, dan variasi laju penguapan tidak memengaruhi hasil. Hasil ini menunjukkan bahwa algoritma relatif stabil terhadap perubahan parameter ρ , tetapi dapat dieksplorasi lebih lanjut untuk kasus yang lebih kompleks.

4.4. Pengaruh Jumlah Semut

Pengujian dilakukan dengan variasi Jumlah Semut (5, 20, 50), sementara parameter lainnya tetap: $\alpha = 1$, $\beta = 2.0$, dan $\rho = 0.5$. Tabel dibawah menunjukkan hasil dari pengujian yang dilakukan.

Tabel 6. Pengaruh Jumlah Semut

Pengaruh Jumlah Semut	Rute yang Dihasilkan	Jarak yang Dihasilkan
10 (default)	A-C-E-F-G-H-I-J-K-L-M-N-P-Q-O-R-S	18.54
5	A-B-D-H-I-J-K-L-M-N-O-Q-R-S	12.81
50	A-B-D-H-I-J-K-L-M-N-P-Q-R-S	12.98
100	A-B-D-H-I-J-K-L-M-N-O-R-S	12.61

Dari pengujian, terlihat bahwa variasi jumlah semut memiliki dampak signifikan terhadap hasil rute. Pada jumlah semut default (10), jarak rute adalah

18.54 km. Ketika jumlah semut ditingkatkan menjadi 50, jarak rute berkurang menjadi 12.98 km. Penurunan jarak yang lebih signifikan terlihat pada jumlah semut 100, di mana jarak rute menjadi 12.61 km, hasil terbaik dalam pengujian ini. Hal ini menunjukkan bahwa jumlah semut yang lebih banyak meningkatkan eksplorasi jalur dan membantu algoritma menemukan solusi yang lebih optimal. Namun, jumlah semut yang terlalu besar dapat meningkatkan waktu komputasi, sehingga diperlukan keseimbangan antara jumlah semut dan efisiensi algoritma.

Dari hasil pengujian di atas didapati pengujian dengan parameter alpha (α): 1, beta (β): 2, evaporation Rate (ρ): 0.5, dan jumlah semut: 100 menghasilkan rute dengan jarak terdekat dengan rincian rute adalah Komplek Bukit Johor Mas - Jl. Karya Jaya - Jl. M. Basir - Jl. Brigjend Katamso - Jl. Pelangi - Jl. Sisingamangaraja - Jl. Stadion - Jl. HM. Joni - Jl. Halat - Jl. Arief Rahman Hakim - Jl. Aksara - Jl. William Iskandar Ps. V - Gerbang 2 UNIMED. Hal ini mengindikasikan bahwa algoritma ACO sangat bergantung pada nilai parameter yang digunakan, dan perubahan kecil pada nilai parameter tertentu dapat memengaruhi hasil yang diperoleh.

Pada beberapa pengujian, hasil rute yang ditemukan oleh algoritma tetap sama meskipun parameter seperti α (pengaruh feromon), β (pengaruh heuristik), atau ρ (evaporation rate) diubah. Hal ini dapat terjadi karena beberapa alasan. Pertama, jika graf memiliki jalur yang sangat jelas lebih pendek dibandingkan jalur lain, algoritma cenderung selalu menemukan solusi optimal tersebut terlepas dari variasi parameter. Kedua, eksplorasi semut yang terbatas juga dapat menjadi penyebab. Sebagai contoh, ketika nilai α terlalu besar, semut lebih fokus pada jalur dengan feromon tinggi sehingga eksplorasi jalur baru menjadi berkurang. Selain itu, distribusi bobot yang relatif homogen pada graf juga dapat membuat probabilitas pemilihan jalur tidak berubah secara signifikan meskipun parameter dimodifikasi. Ketiga, jumlah iterasi yang tidak cukup dapat membatasi kemampuan algoritma untuk mengeksplorasi semua jalur yang tersedia, sehingga menghasilkan solusi yang sama pada setiap pengujian. Terakhir, hasil rute yang sama juga dapat menunjukkan stabilitas algoritma, di mana algoritma berhasil mencapai keseimbangan antara eksplorasi dan eksploitasi sehingga solusi optimal tetap ditemukan meskipun parameter diubah. Fenomena ini menunjukkan bahwa karakteristik graf dan parameter memiliki peran penting dalam memengaruhi hasil algoritma Ant Colony Optimization (ACO).

5. KESIMPULAN DAN SARAN

Penelitian ini telah berhasil menerapkan algoritma Ant Colony Optimization (ACO) menggunakan Python untuk mencari rute terpendek dari komplek Bukit Johor Mas ke Universitas Negeri Medan (UNIMED). Berdasarkan hasil pengujian parameter yang telah kami lakukan, diperoleh melalui

pengaturan parameter optimal, alpha (α) sebesar 1, beta (β) sebesar 2, evaporation Rate (ρ) sebesar 0.5, dan jumlah semut sebanyak 100 yang menghasilkan jarak terpendek sepanjang 12,61 km. Penggunaan Python terbukti efektif berkat pustaka pendukungnya yang mempermudah analisis dan visualisasinya. Untuk pengembangan lebih lanjut, disarankan untuk melakukan perbandingan ACO dengan algoritma lainnya seperti Dijkstra, menguji pada jaringan jalan yang lebih kompleks, serta mempertimbangkan integrasi hasil penelitian ini kedalam aplikasi navigasi berbasis teknologi untuk memberikan manfaat praktis yang lebih luas.

Nur Alfa Husna, D. H. (2023). Implementasi Algoritma Ant Colony Optimization untuk Penentuan Jalur Terpendek Klinik dari Lokasi Rawan Kecelakaan Di Kota Pekanbaru . *Institut Riset dan Publikasi Indonesia*, 112-119.

DAFTAR PUSTAKA

- [1] Daniel Udjulawa, S. O. (2022). Penerepan Algoritma Ant Colony Optimaztion Untuk Pencairan Rute Terpendek Lokasi Wisata (Studi Kasus Wisata Di Kota Palembang). *Jurnal Ilmu Komputer*, 26-33.
- [2] Via Risqiyanti, A. D. (2020). Pencarian Rute Terpendek Menggunakan Algoritma Ant Colony Optimization Guna Memantau Sustainable Development Goals. *Seminal Nasional Official Statistics*, 31-38.
- [3] Siti Nurhalisa Jumaedi, W. A. (2024). Penerapan Algoritma Ant Colony Optimization (ACO) Rute Jalur Terpendek (Studi Kasus Distribusi Barang JNE Wilayah Bumi Tamalanrea Permai (BTP)) . *Jurnal Matematika dan Statistika serta Aplikasinya*, 108-115.
- [4] Kumala Dewi Ts, M. I. (2020). Penerapan Algoritma Ant Colony Optimizaton Pada pencarian Rute optimal untuk Transportasi Umum Kota Bandung. *e-Proceeding of Engineering*, 2827-2839.
- [5] Sihotang, H. T. (2021). Determination of the Shortest Path Using the Ant Colony Optimization (ACO) Algorithm Approach. *Computer, Mathematics and Engineering Applications*, 123-133.
- [6] Muhammad Romzi, B. K. (2020). PEMBELAJARAN PEMROGRAMAN PYTHON DENGAN PENDEKATAN LOGIKA ALGORITMA. *Jurnal Teknik Informatika Mahakarya*, 37-44.
- [7] Nur Alfa Husna, D. H. (2023). Implementasi Algoritma Ant Colony Optimization untuk Penentuan Jalur Terpendek Klinik dari Lokasi Rawan Kecelakaan Di Kota Pekanbaru . *Institut Riset dan Publikasi Indonesia*, 112-119.
- [8] Said Broumi, P. K. (2023). Solving shortest path problems using an ant colony algorithm with triangular neutrosophic arc weights . *International Journal of Neutrosophic Science*, 128-137.
- [9] Aeri Racmad, M. S. (2023). Application of ant colony optimization algorithm to determine optimal value in choosing tourist attractions in Bangkalan-Madura. *AIP Conference Proceedings*, 1-7.
- [10] Anten Regita Rangkuti, F. H. (2020). Usulan Perbaikan Rute Pengambilan Sampah Menggunakan Algoritma Clarke & Wright Savings Dan Algoritma Ant Colony Optimization . *Desiminasi*, 1-10.
- [11] Arief Kelik Nugroho, I. P. (2020). IMPLEMENTASI JALUR TERPENDEK MENGGUNAKAN ANT COLONY OPTIMIZATION . *Dinamika Rekayasa*, 61-68.