

Optimasi Rute Kunjungan *Cluster Sales Officer (CSO)* Menggunakan *Ant Colony Optimization (ACO)* (Studi Kasus: Indosat Ooredoo Hutchison *Micro Cluster Mamuju*)

Irwin Yaputera*¹, Rosmalina Hanafi¹, Muhammad Rusman¹

¹Departemen Teknik Industri, Fakultas Teknik, Universitas Hasanuddin
Jl. Poros Malino km. 6, Bontomarannu, Kabupaten Gowa, Sulawesi Selatan 92171

*Email: irwinyaputera.iy@gmail.com

DOI: 10.25042/jpe.112022.04

Abstrak

Indosat Ooredoo Hutchison adalah sebuah perusahaan swasta yang bergerak di bidang telekomunikasi. Proses bisnis pada perusahaan ini dipengaruhi oleh performa dari *salesman*, dimana dalam perusahaan disebut dengan *Cluster Sales Officer (CSO)*. CSO menentukan rute kunjungan *outlets* berdasarkan pengalaman sehingga rute kunjungannya belum optimal yang menyebabkan jarak dan waktu tempuh serta biaya masih belum efisien. Penelitian ini dilakukan untuk menentukan rute kunjungan yang optimal menggunakan algoritma *Ant Colony Optimization*. Tujuannya untuk meminimalkan jarak dan waktu tempuh serta biaya. Data yang diperoleh adalah data koordinat dan urutan kunjungan CSO pada Rute I (Senin, Rabu dan Jumat) dan Rute II (Selasa, Kamis dan Sabtu). Berdasarkan data tersebut, peneliti kemudian menghitung jarak dan waktu tempuh serta biaya dikeluarkan pada rute awal CSO. Langkah selanjutnya adalah menghitung jarak dan waktu tempuh serta biaya yang dikeluarkan pada rute usulan dengan algoritma *Ant Colony Optimization*. Hasil perhitungan menunjukkan bahwa dengan menggunakan algoritma *Ant Colony Optimization*, Rute I Usulan ACO yang memiliki total jarak yang ditempuh, waktu yang diperlukan serta biaya yang harus dikeluarkan diperoleh efisiensi sekitar 8% dari rute awalnya. Sedangkan untuk Rute II Usulan ACO mengalami efisiensi sekitar 4% dari rute awalnya. Sehingga diperoleh efisiensi Rute Usulan ACO dalam sebulan sekitar 6% terhadap Rute Awal CSO.

Abstract

Optimization of Cluster Sales Officer (CSO) Visit Routes Using Ant Colony Optimization (ACO) (Case Study: Indosat Ooredoo Hutchison Micro Cluster Mamuju). Indosat Ooredoo Hutchison is a private company engaged in the telecommunications sector. Business processes in this company are influenced by the performance of salesmen, which in the company are called Cluster Sales Officer (CSO). CSO determines the route of visiting outlets based on experiences. However, the route taken is not optimal yet in terms of distance and travel time which contribute to cost inefficiency. This research was conducted to determine the optimal route using the Ant Colony Optimization algorithm to minimize the distance and travel time and costs. The data obtained are the coordinates and order of CSO visits on Route I (Monday, Wednesday, and Friday) and Route II (Tuesday, Thursday, and Saturday). Based on these data, the researchers then calculated the distance and travel time as well as the costs incurred on the initial CSO route. The next step is to calculate the distance and travel time as well as the costs incurred on the proposed route with the Ant Colony Optimization algorithm. The calculation results show that by using the Ant Colony Optimization algorithm, Proposed Route I which has the total distance traveled, time required, and costs incurred, obtained an efficiency of around 8% from initial route. As for the Proposed Route II has an efficiency of around 4% is obtained from the initial route. So that the total proposed route in a month is more efficient around 6% compared to the initial route by CSO.

Kata Kunci: *Telecommunication company, travelling salesman problem (TSP), ant colony optimization algorithm, MATLAB*

1. Pendahuluan

Setiap hari manusia akan dihadapkan pada banyak permasalahan hidup yang semakin kompleks. Penyelesaian dari setiap permasalahan harus menghasilkan solusi yang terbaik untuk setiap sumber dayanya. Salah satu permasalahan tersebut adalah pencarian rute optimal, dimana sumber daya dari waktu dan bahan bakar harus diefisienkan seminimum mungkin. Penentuan rute optimal merupakan salah satu keilmuan teknik industri dalam hal optimasi yang bertujuan

untuk menghasilkan suatu rute yang efisien baik dalam hal jarak, biaya, waktu atau lain [1]. Masalah seperti ini dikategorikan dalam suatu permasalahan kombinatorial yang dikenal dengan *Traveling Salesman Problem (TSP)*. TSP merupakan masalah klasik untuk mencari rute terpendek yang dapat dilewati salesman ke sejumlah tempat tanpa harus mendatangi tempat yang sama lebih dari satu kali [2].

Pada perkembangannya, ternyata TSP merupakan persoalan yang banyak diaplikasikan

pada berbagai persoalan dunia nyata, misalnya: efisiensi pengiriman surat dan barang, perencanaan pemasangan saluran pipa, masalah transportasi, persoalan *delivery order* (jasa pengantar makanan), dan seterusnya. Salah satu teori algoritma yang membantu dalam penyelesaian TSP adalah *Ant Colony Optimization* (ACO) [3].

Ant Colony Optimization (ACO) merupakan algoritma yang terinspirasi dari perilaku semut dalam menemukan jalan dari sarangnya menuju tempat makanannya. Semut memiliki zat pheromone dimana zat ini merupakan zat kimia yang berasal dari kelenjar endokrin dan digunakan oleh makhluk hidup untuk mengenali sesama jenis. Semut-semut tersebut akan memilih jalan berdasarkan kuatnya aroma pheromone. Semakin banyak semut yang menempuh suatu jalan tertentu, maka aroma pheromone pada lintasan tersebut akan semakin kuat sehingga semut-semut berikutnya akan mengikuti lintasan tersebut [4]. Algoritma *Ant Colony Optimization* (ACO) memiliki keunggulan dapat memberikan solusi dengan batasan sumber daya dapat diatur sendiri serta memberikan solusi rute optimal yang cenderung konstan untuk beberapa kali pengujian.

Penelitian ini dilaksanakan di salah satu perusahaan yang bergerak di bidang distribusi industri telekomunikasi, yakni Indosat Ooredoo Hutchison. Perusahaan swasta ini memiliki dua alur bisnis untuk memperoleh keuntungannya, yaitu *modern channel* dan *traditional channel*. Kedua alur bisnis memiliki kebermanfaatannya masing-masing, dimana *modern channel* melakukan kegiatan bisnis melalui aplikasi dan media elektronik sedangkan *traditional channel* melakukan kegiatan bisnis melalui toko konvensional (*outlets*).

Proses bisnis pada *traditional channel* dipengaruhi oleh performa dari *salesman*, dimana dalam perusahaan disebut dengan *Cluster Sales Officer* (CSO). Beban kerja CSO tersebut berorientasi pada target. Oleh sebab itu, CSO pada perusahaan ini harus bekerja cepat dalam menjalankan perjalanan kunjungan hariannya. Kunjungan harian dilakukan untuk mengecek stok dan menjual saldo, kartu perdana, maupun *voucher* kepada *outlet*, dimana pada umumnya memakan waktu sekitar 15 menit untuk setiap *outlet* yang dikunjungi. CSO memiliki tantangan untuk menyelesaikan sekitar 25 titik kunjungan ke beberapa *outlet*.

Permasalahan yang sering terjadi khususnya pada *Micro Cluster* Mamuju, yaitu durasi rute kunjungan yang masih dapat diefisienkan dikarenakan pengambilan rute yang masih belum optimal sebab penentuan rute sekarang masih menggunakan intuisi CSO itu sendiri.

Berdasarkan uraian yang dijelaskan di atas maka penulis menggunakan *Ant Colony Optimization* (ACO) untuk memecahkan *Travelling Salesman Problem* (TSP) yang dihadapi CSO *Micro Cluster* Mamuju dengan bantuan *software* MATLAB.

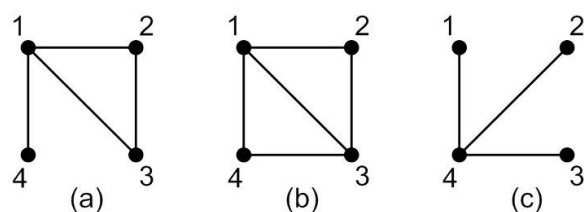
2. Tinjauan Pustaka

2.1. Teori Graph

Graph merupakan representasi dari suatu masalah yang digambarkan sebagai sekumpulan titik atau simpul (*vertex*) yang dihubungkan dengan sekumpulan garis atau sisi (*edge*). Secara singkat suatu *graph* dapat ditulis sebagai $G = (V, E)$ yang dalam hal ini $V =$ Himpunan berhingga dan titik kosong dari simpul-simpul $vertices = \{v_1, v_2, \dots, v_n\}$ dimana $E =$ Himpunan sisi yang menghubungkan sepasang simpul $= \{e_1, e_2, \dots, e_n\}$.

Simpul pada *graph* dapat dinomori dengan huruf seperti v, w, \dots , dengan bilangan asli $1, 2, 3, \dots$, atau gabungan keduanya. Sisi (*edge*) yang menghubungkan dua simpul (*vertex*) v_i dan v_j , dan dinyatakan dengan pasangan $(v_i v_j)$ atau dengan lambang e_1, e_2 , dengan kata lain, jika e adalah sisi yang menghubungkan simpul v_i dan v_j , maka e dapat ditulis sebagai $e = (v_i, v_j)$ [5].

Lintasan Hamilton adalah lintasan yang melalui tiap simpul di dalam *graph* G tepat satu kali seperti ditunjukkan pada Gambar 1 bagian (a). Bila lintasan itu kembali lagi ke simpul awal dan membentuk lintasan tertutup, maka lintasan tertutup itu dinamakan Sirkuit Hamilton [6]. Jadi, Sirkuit Hamilton adalah sirkuit yang melalui tiap simpul di dalam *graph* G tepat satu kali, kecuali simpul awal dan simpul akhir seperti ditunjukkan pada Gambar 1 bagian (b).



Gambar 1. Contoh Graph [6]

Keterangan Gambar:

- Graph* yang memiliki lintasan Hamilton (misal: 3, 2, 1, 4)
- Graph* yang memiliki sirkuit Hamilton (misal: 1, 2, 3, 4, 1)
- Graph* yang tidak memiliki lintasan maupun sirkuit Hamilton

2.2. Optimisasi

Optimisasi adalah suatu proses untuk mencapai hasil yang 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 riil. Untuk dapat mencapai nilai optimal baik minimal atau maksimal tersebut, secara sistematis dilakukan pemilihan nilai variabel integer atau riil yang akan memberikan solusi optimal [7].

Di dalam konteks matematika, optimisasi ini bisa dinyatakan sebagai suatu usaha sistematis untuk mencari nilai minimum atau maksimum dari suatu fungsi. Fungsi ini secara sederhana dapat dinyatakan dengan:

$$\min / \max f(x) \quad (1)$$

sebagai contoh adalah fungsi kuadrat $f(x) = x^2$, dimana x anggota bilangan riil ($x \in R$), di dalam contoh ini, $f(x) = x^2$ merupakan fungsi tujuannya, sedangkan x adalah daerah asal yang didefinisikan sebagai anggota bilangan riil.

Permasalahan yang berkaitan dengan optimisasi sangat kompleks dalam kehidupan sehari-hari. Nilai optimal yang didapat dalam optimisasi dapat berupa besaran panjang, waktu, jarak, dan lain-lain. Salah satu permasalahan optimasi yang sering ditemui adalah menentukan lintasan terpendek dari suatu tempat ke tempat yang lain.

Secara umum, penyelesaian masalah pencarian rute terpendek dapat dilakukan dengan menggunakan dua metode, yaitu metode konvensional dan metode heuristik. Metode konvensional dihitung dengan perhitungan matematis biasa, sedangkan metode heuristik dihitung dengan menggunakan sistem pendekatan

2.3. Traveling Salesman Problem (TSP)

Menurut Hoffman dan Wolfe, *Traveling salesman problem* merupakan suatu masalah yang mudah dideskripsikan namun sulit untuk

diselesaikan, yaitu masalah bagaimana menentukan jarak terpendek dalam perjalanan melewati titik-titik tertentu di mana satu titik harus dilalui satu kali dan hanya boleh dilalui satu kali saja dan perjalanan harus berakhir dengan kembali ke titik pertama. Dalam hal ini titik-titik tersebut merupakan kota-kota dalam suatu wilayah tertentu [8].

Travelling Salesman Problem dikatakan ada dua jenis, yaitu [9]:

1. *Travelling Salesman Problem* Asimetri

Pada *Travelling Salesman Problem* jenis ini, biaya dari kota 1 ke kota 2 tidak sama dengan biaya dari kota 2 ke kota 1. Dengan n kota, besarnya ruang pencarian adalah $\frac{n!}{n} = (n - 1)!$ jalur yang mungkin.

2. *Travelling Salesman Problem* Simetris

Sedangkan pada *Travelling Salesman Problem* jenis simetris, biaya dari kota 1 ke kota 2 adalah sama dengan biaya dari kota 2 ke kota 1. Apabila dengan n kota, jumlah jalur yang mungkin adalah $\frac{n!}{2n} = \frac{(n-1)!}{2}$ jalur yang mungkin.

Secara singkat, karakteristik dari permasalahan TSP adalah:

- Perjalanan berawal dan berakhir dari dan ke kota awal
- Ada sejumlah kota yang semuanya harus dikunjungi tepat satu kali
- Perjalanan tidak boleh kembali ke kota awal sebelum semua kota tujuan dikunjungi
- Tujuan dari permasalahan ini adalah meminimumkan total jarak yang ditempuh salesman dengan mengatur urutan kota yang harus dikunjungi

Menurut David Bolton, algoritma adalah gambaran dari suatu langkah-langkah yang memperoleh suatu keberhasilan dari sebuah hasil. Menurut Donald E. Knuth, ada 5 ciri-ciri yang sangat penting, yaitu [10]:

1. Keterbatasan (*Finiteness*)

Keterbatasan yaitu suatu algoritma yang akan selesai jika sudah membuat beberapa proses.

2. Kepastian (*Definiteness*)

Kepastian yaitu semua cara algoritma harus di jelaskan dengan tepat dan bukan berarti ganda.

3. Masukan (*Input*)

Algoriitma mempunyai nilai nol atau lebih beberapa data masukkan (*input*)

4. Keluaran (*Output*)
Algoritma mempunyai nilai nol atau lebih beberapa data hasil keluaran (*output*).
5. Efektivitas (*Effectiveness*)
Langkah-langkah algoritma harus efektif dan dikerjakan dalam waktu yang wajar.

Untuk menyelesaikan permasalahan *Travelling Salesman Problem* (TSP) diperlu pendekatan algoritma untuk memperoleh hasil yang efisien dan efektif, diantaranya algoritma *Greedy*, *Artificial Bee Colony* (ABC), *Cheapest Insertion Heuristics* (CIH), *Ant Colony Optimization* (ACO), dan lain-lain.

2.4. *Ant Colony Optimization* (ACO)

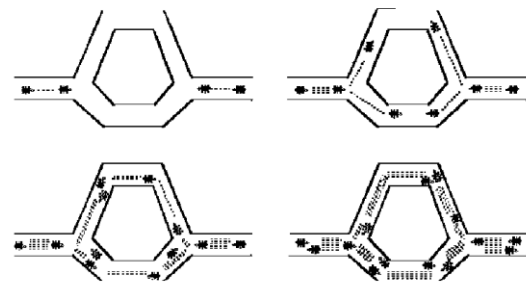
Pada tahun 1996, dunia AI diperkenalkan dengan algoritma Semut oleh Moyson dan Manderick dan secara meluas dikembangkan oleh Marco Dorigo, merupakan algoritma yang terinspirasi oleh perilaku semut dalam menemukan jalur dari sarangnya menuju makanan [11].

Semut adalah serangga sosial yang hidupnya berkoloni, dapat bekerja sama dengan sesamanya dalam melakukan pekerjaan secara efektif. Perilaku semut dalam menemukan makanan dari sarangnya menghasilkan jalur yang optimal dengan menemukan jalur terpendek [6]. Semut juga mampu mengindra lingkungannya yang kompleks untuk mencari makanan dan kemudian kembali ke sarangnya dengan meninggalkan zat feromon pada jalur-jalur yang mereka lalui [12]. Feromon adalah zat kimia yang berasal dari kelenjar endokrin dan digunakan oleh makhluk hidup untuk mengenali sesama jenis, individu lain, kelompok, dan untuk membantu proses reproduksi. Proses peninggalan feromon ini dikenal sebagai *stigmergy*, yaitu sebuah proses memodifikasi lingkungan yang tidak hanya bertujuan untuk mengingat jalan pulang ke sarang, tetapi juga memungkinkan para semut berkomunikasi dengansesamanya [11]. Seiring waktu, bagaimanapun juga jejak feromon akan menguap dan akan mengurangi kekuatan daya tariknya. Lebih lama seekor semut pulang pergi melalui jalur tersebut, lebih lama jugalah feromon menguap [13].

Agar semut mendapatkan jalur optimal dalam perjalanannya, diperlukan beberapa proses:

1. Pada awalnya, semut berkeliling secara acak, hingga menemukan makanan.

2. Ketika menemukan makanan mereka kembali ke sarangnya sambil memberikan tanda dengan jejak feromon.
3. Jika semut-semut lain menemukan jalur tersebut, maka mereka akan bepergian dengan acak lagi, melainkan akan mengikuti jejak tersebut.
4. Jika pada akhirnya mereka pun menemukan makanan, maka mereka kembali dan menguatkan jejaknya.
5. Feromon yang berkonsentrasi tinggi pada akhirnya akan menarik semut-semut lain untuk berpindah jalur, menuju jalur paling optimal, sedangkan jalur lainnya akan ditinggalkan. Gambar 2 menunjukkan perjalanan semut dalam menemukan jalur terpendek dari sarang ke sumber makanan.



Gambar 2. Perjalanan semut menentukan makanan [13]

Pengolahan data menggunakan algoritma ACO akan dilakukan dengan tahap-tahap sebagai berikut [14].

1. Inisialisasi Parameter

Dalam algoritma ACO terdapat beberapa parameter yang akan mempengaruhi hasil rute optimal:

- a. Parameter α adalah tingkat kepentingan dari intensitas feromon.
- b. Parameter β adalah tingkat kepentingan dari *visibility* titik kunjungan.
- c. Parameter ρ adalah tingkat penguapan dari feromon.
- d. Parameter m adalah banyaknya semut
- e. Parameter τ_0 adalah intensitas feromon awal

Perhitungan parameter α , β , dan γ akan dilakukan masing-masing tergantung studi kasusnya untuk memperoleh hasil yang terbaik. Sedangkan banyaknya semut sama dengan jumlah titik pada masalah ($m = n$). Hal ini untuk menghindari jumlah semut yang berlebih sehingga akan menimbulkan ketidakefektifan dalam penyelesaian. Oleh

sebab itu, jumlah semut (m) yang digunakan adalah 24 semut. Kemudian, untuk nilai τ_0 akan digunakan nilai 1000. Hal ini diperuntukan agar koloni semut dapat mengeksplor lebih banyak peluang titik sampai ke penentuan rute stabilnya (*converge*) [15].

2. Menentukan titik kunjungan selanjutnya

Setelah menentukan kantor sebagai titik awal dan akhir dari rute maka akan ditentukan titik tujuan selanjutnya. Penentuan titik selanjutnya ini berdasarkan feromon. Semut disuruh untuk memutuskan apakah mereka harus ke arah mana, dan pilihan yang dibuat adalah keputusan acak. Akumulasi feromon akan lebih banyak pada jalur yang lebih pendek. Perbedaan kandungan feromon antara kedua jalur dari waktu ke waktu membuat semut memilih jalur yang lebih pendek. Semakin banyak semut melewati rute tersebut, semakin menarik rute itu untuk diikuti.

Pertama, tentukan *visibility* dari semua titik kunjungan dengan rumus

$$\eta = \frac{1}{matrix\ jarak} \quad (2)$$

Semut k pada titik r akan memilih titik tujuan s pada tahap selanjutnya dengan probabilitas.

$$p_k(r, s) = \begin{cases} \tau(r, s)^\alpha \eta(r, s)^\beta / \sum_{\mu \in M_k} \tau(r, \mu)^\alpha \eta(r, \mu)^\beta, & \text{for } s \in M_k \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Setelah memperoleh nilai masing-masing probabilitas setiap titik kunjungan maka akan dikumulatitkan sehingga diperoleh nilai kumulatif probabilitas masing-masing titik kunjungan.

Kemudian, akan dibangkitkan nilai *random*, lalu dibandingkan dengan nilai kumulatif probabilitas. Nilai kumulatif probabilitas yang tepat di atas nilai *random*, akan dipilih sebagai titik selanjutnya. Lalu, dilakukan pencarian titik kunjungan selanjutnya, seperti cara di atas sampai tidak ada lagi titik kunjungan.

3. Perbarui feromon

Setelah semua semut telah melakukan rute lengkap, maka akan dilakukan pembaruan feromon untuk setiap jalur titik satu ke titik lainnya. Semut k ketika melewati segmen akan meninggalkan feromon. Jumlah feromon yang terkandung dalam segmen i, j setelah melewati semut k diberikan oleh rumus

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \sum_{k=1}^m \Delta\tau_{i,j}^{(k)} \quad (4)$$

Dimana ρ adalah parameter laju evaporasi dari feromon yang dilalui semut dari titik pertama sampai kembali ke titik awal. Penurunan jumlah feromon memungkinkan semut untuk mengeksplorasi jalur yang berbeda selama proses pencarian, juga akan menghilangkan kemungkinan pemilihan jalur yang kurang baik. Jumlah feromon yang ditambahkan ke segmen $i - j$ oleh semut k diberikan sebagai

$$\Delta\tau_{i,j}^{(k)} = \frac{Q}{L_k} \quad (5)$$

di mana Q adalah konstanta dan L_k adalah total jarak yang ditempuh semut k setelah kembali ke sarang. Nilai Q biasanya ditentukan oleh pengguna. Dalam banyak kasus nilai $Q = 1$ memberikan hasil yang baik.

4. Pengulangan / Iterasi

Setelah dilakukan pembaruan feromon maka akan dilakukan iterasi dari tahap awal dengan feromon terbaru. Iterasi dilakukan sampai hasil perhitungan dinilai cukup ataupun iterasi sebelumnya ke selanjutnya tidak mengalami perubahan yang signifikan atau dengan kata lain telah mendekati titik stabilnya (*converge*).

2.5. Pemrograman

Menurut Sunarto, mengatakan bahwa program beberapa kumpulan arahan yang akan dicapai baik itu dalam kode, bagan dan bahasa. Jika beberapa arahan tersebut dikumpulkan atau di gabungkan menjadi satu menggunakan media dan dapat di baca oleh sistem komputer maka itu akan membuat komputer dapat bekerja untuk melakukan suatu fungsi dalam sebuah persiapan merancang instruksi-instruksi tersebut [5].

Untuk membuat program penyelesaian TSP dibutuhkan algoritma dan bahasa pemrogramannya. Program TSP ini dapat berjalan dalam perangkat lunak (*software programming*). *Software programming* yang dapat digunakan untuk penyelesaian *Travelling Salesman Problem* adalah sebagai berikut:

1. Visual Studio Code

Visual Studio Code adalah *software* yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan *built-in* dukungan untuk

JavaScript, naskah dan Node.js dan memiliki array beragam ekstensi yang tersedia untuk bahasa lain, termasuk C++, C#, Python, dan PHP [16]. Banyak sekali fitur-fitur yang disediakan oleh Visual Studio Code, diantaranya Intellisense, Git Integration, Debugging, dan fitur ekstensi yang menambah kemampuan teks editor. [17].

2. Matrix Laboratory (MATLAB)

Matlab adalah singkatan dari Matrix Laboratory (Laboratorium Matriks) dan merupakan bahasa pemrograman yang dibuat dengan tujuan sebagai alat bantu perhitungan yang rumit atau simulasi dari suatu sistem yang ingin di simulasikan, dalam matlab mutlak dibutuhkan pengetahuan tentang matriks yang dapat dipelajari dalam ilmu matematika [18].

2.6. MATLAB

Matlab adalah suatu software pemrograman perhitungan dan analisis yang banyak digunakan dalam semua area penerapan matematika baik bidang pendidikan maupun penelitian pada universitas dan industri. Dengan matlab, maka perhitungan matematis yang rumit dapat diimplementasikan dalam program dengan lebih mudah [19].

MATLAB (Matrix Laboratory) adalah suatu program untuk analisis dan komputasi numerik dan merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk matriks. Pada awalnya, program ini merupakan interface untuk koleksi rutin-rutin numeric dari proyek LINPACK dan EISPACK, dan dikembangkan menggunakan bahasa FORTRAN namun sekarang merupakan produk komersial dari perusahaan Mathworks, Inc. yang dalam perkembangan selanjutnya dikembangkan menggunakan bahasa C++ dan assembler (utamanya untuk fungsi-fungsi dasar MATLAB) [20].

3. Metode Penelitian

Objek penelitian ini adalah rute kunjungan harian Cluster Sales Officer (CSO) Indosat Ooredoo Hutchison Micro Cluster Mamuju. Jenis data yang diambil dari penelitian adalah data primer internal perusahaan dengan mengumpulkan data koordinat kunjungan harian

outlet dengan bantuan Google Maps. Metode pengambilan data yang diambil dalam penelitian ini dilakukan dengan dua cara sebagai berikut:

1. Observasi
Melakukan observasi terkait titik kunjungan dan pemilihan rute CSO
2. Wawancara
Melakukan wawancara kepada CSO terkait urutan kunjungan harian yang dilakukan CSO

Data dari hasil observasi lapangan dan wawancara dari pihak CSO akan dikumpulkan, lalu dengan bantuan Google Maps akan dicari jarak dari titik kunjungan satu ke titik kunjungan lainnya. Sehingga dapat digambarkan dalam bentuk matriks.

Pengumpulan literatur tentang TSP dan Ant Colony Optimization (ACO) sebagai pedoman dalam penyelesaian penelitian ini. Dimana literatur tersebut sebagian besar berisi tentang hal yang berhubungan dengan TSP misalnya teori graph dan program dinamik dan berhubungan dengan menentukan rute tercepat dari suatu graph dengan ACO.

Setelah pengumpulan literatur, dilakukan pembuatan program berdasarkan matriks yang diperoleh dari data titik koordinat yang telah dikumpulkan dengan bantuan software MATLAB.

Terakhir, hasil rute usulan dengan menggunakan ACO akan dibandingkan dengan rute awal yang CSO lakukan.

4. Hasil dan Pembahasan

Berdasarkan hasil wawancara dan observasi pada Cluster Sales Officer (CSO) Micro Cluster Indosat Ooredoo Hutchison diambil satu orang sebagai sampel penelitian karena memenuhi syarat TSP, yakni membentuk sirkuit Hamilton. Selanjutnya dilakukan wawancara untuk mengetahui 2 rute perjalanan yang dilakukan selama ini, dimana rute itu digunakan secara berulang-ulang. Rute I digunakan untuk hari kerja Senin, Rabu dan Jumat sedangkan Rute II digunakan untuk hari kerja Selasa, Kamis dan Sabtu. Data koordinat dari semua titik kunjungan CSO dapat dilihat pada Tabel 1.

Setelah itu, data koordinat dimasukkan dalam maps dan dicari jarak lintasan dari satu ke titik lainnya, guna mencari matriks jarak dan waktunya. Dengan asumsi konsumsi bahan bakar jenis Pertalite sebanyak 1-liter habis

terpakai untuk perjalanan sejauh 30 km, dan 1-liter Peralite seharga Rp7.650, maka dapat diketahui jumlah komsumsinya dan biayanya.

Tabel 1. Titik kunjungan Rute I dan Rute II CSO

Rute	Kode	Nama	Longitude	Latitude
I	Kantor	Kantor Distribusi	119,2291565	-3,4001499
	A	Maestro cell	119,2293776	-3,4037714
	B	nurmada cell	119,2117967	-3,40167
	C	MamanOnepiece Cell	119,211848	-3,4019797
	D	Akbar cell	119,2107685	-3,4018182
	E	Ich One Cell	119,2102268	-3,401488
	F	FARIS CELL 02	119,2115459	-3,4035562
	G	Harmonis cell	119,2115068	-3,4036639
	H	Alfhy Cell	119,211264	-3,4058316
	I	Zhafran cell	119,2108938	-3,412605
	J	Fariz Bumiayu cell	119,2104848	-3,4229177
	K	Tugu cell	119,2099567	-3,423095
	L	Fortuna cell	119,2186879	-3,4276195
	M	Kanita cell	119,2195013	-3,4273898
	N	Avika cell	119,220683	-3,4280889
	O	KEMBAR CELL	119,2179475	-3,4340474
	P	TokoRahman cell	119,2188057	-3,4410952
	Q	Radinal cell	119,2162717	-3,4453666
	R	Wiwin cell	119,2150094	-3,4468107
	S	Naya cell	119,2349743	-3,444964
	T	ATK NIA CELL	119,242885	-3,4295433
	U	Net1 cell	119,2365358	-3,4260139
	V	fitra dj cell	119,2354267	-3,4228367
	W	FTH cell	119,2364508	-3,4260024
X	ice cell	119,2349817	-3,4138667	
II	a	BB cell	119,2291269	-3,4001439
	b	4G cell	119,2227674	-3,3973401
	c	RR cell	119,2208323	-3,3913528
	d	WNK JAYA CELL	119,2211239	-3,3912891
	e	Arwin cell	119,2175723	-3,3883993
	f	Fira Cell	119,2192736	-3,3875878
	g	Cahaya Belawa	119,2352724	-3,3492122
	h	Sinar Tapango Cell	119,2353797	-3,3490762
	i	Annisa cell	119,2361026	-3,3489393
	j	Rahmat Hidayat	119,2360852	-3,3490499
	k	November cell	119,2358783	-3,3491683
	l	Aisyah cell	119,2406599	-3,3471889
	m	Putri cell	119,2412904	-3,3445083
	n	77 cell	119,2432376	-3,3413709
	o	Bleeksweet Cell	119,2434767	-3,3410034
	p	Dhita cell	119,2444681	-3,339089
	q	NyoMan Store	119,2491563	-3,3485978
	r	Ayunhi Cell	119,2516642	-3,3486863
	s	Tiara Cell	119,2171272	-3,3930448
	t	FARIZ PERDANA	119,2086062	-3,3992533
	u	Jomblo cell 02	119,2086802	-3,3992026
	v	Itha cell	119,2085517	-3,3992691
	w	Wono store	119,208801	-3,3995664
	x	Aqila cell	119,2050133	-3,4003317

(Sumber: Hasil Observasi Titik Outlet CSO Indosat Ooredoo Hutchison Micro Cluster Mamuju)

Urutan Rute I yang dilakukan oleh CSO adalah sebagai berikut: Kantor – A – B – C – D – E – F – G – H – I – J – K – L – M – N – O – P – Q – R – S – T – U – V – W – X – Kantor, dengan total jarak 22,921 km dan total waktu tempuhnya 34,387 menit., serta komsumsi bahan bakar untuk menyelesaikan Rute I Awal CSO adalah 0,764 L dan biaya bahan bakar yang harus dikeluarkan CSO untuk menyelesaikan satu kali Rute I Awal CSO adalah sebesar Rp5.844,-.

Urutan Rute II yang dilakukan oleh CSO adalah sebagai berikut: Kantor – a – b – c – d – e – f – g – h – i – j – k – l – m – n – o – p – q – r – s – t – u – v – w – x – Kantor, dengan total jarak 26,772 km dan total waktu tempuhnya 40,168 menit, serta komsumsi bahan bakar untuk menyelesaikan Rute II Awal CSO adalah 0,892 L dan biaya bahan bakar yang harus dikeluarkan CSO untuk menyelesaikan satu kali Rute II Awal CSO adalah sebesar Rp6.823,-.

Pengolahan data diawali dengan penentuan parameter awal yang terdiri dari parameter m atau jumlah semut yang digunakan adalah sama dengan titik yang akan dikunjungi, yakni 24 dan parameter τ_0 atau intensitas feromon awal sebesar 1000. Sedangkan untuk parameter α (tingkat kepentingan dari intensitas feromon), β (tingkat kepentingan dari visibility), γ (tingkat penguapan dari feromon) akan dilakukan uji pada masing-masing kasusnya dengan iterasi 300. Pengujian dilakukan tiga kali untuk mencari nilai minimum dan rata-rata terkecil. Hasil parameter yang digunakan ditunjukkan Tabel 2 di bawah.

Tabel 2. Parameter untuk perhitungan Rute I dan II

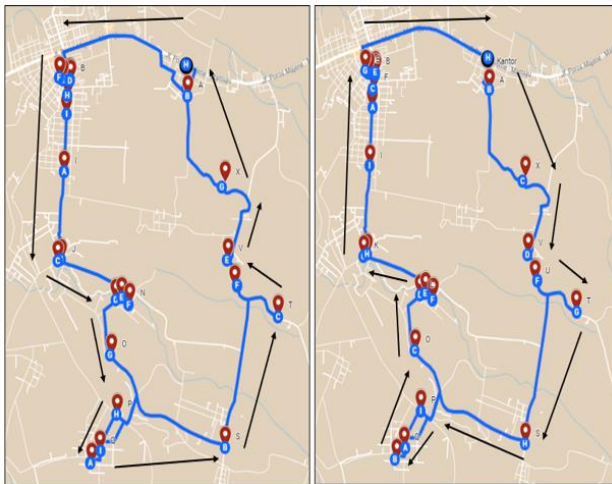
Parameter	Rute I	Rute II
α	2	2
β	1	0,5
γ	0,1	0,1
m	24	24
τ_0	1000	1000
Iterasi	300	300

Hasil terbaik diperoleh untuk Rute II adalah sebagai berikut. Kantor – A – X – V – W – U – T – S – P – Q – R – O – L – M – N – K – J – I – H – G – F – C – B – D – E – Kantor, dengan total jarak 20,982 km dan total waktu tempuhnya 31,479 menit. Komsumsi bahan bakar untuk Rute I usulan adalah 0,699 L dan biayanya sebesar Rp5.347,-.

Hasil terbaik diperoleh untuk Rute II adalah sebagai berikut: Kantor – a – b – u – t – v – w – x – s – c – d – e – f – g – h – k – j – i – m – n – o – p – l – r – q – Kantor, dengan total jarak 25,785 km dan total waktu tempuhnya 38,687 menit. Komsumsi bahan bakar untuk Rute II usulan adalah 0,859 L dan biayanya sebesar Rp6.571,-.

Perbedaan tampilan Rute I awal dan usulan dapat dilihat pada Gambar 3.

Dari Gambar 3 di atas dapat dilihat adanya perubahan dari urutan kunjungannya yang cukup signifikan dari rute awal CSO dan rute usulan menggunakan ACO.



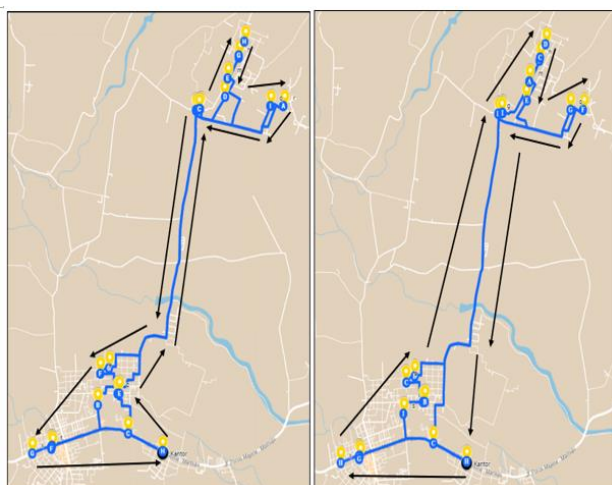
Gambar 3. (a) Rute I awal CSO (b) Rute I usulan ACO

Tabel 3 menunjukkan perbandingan total jarak dan waktu tempuh serta biaya yang harus dikeluarkan sekali menyelesaikan Rute I.

Tabel 3. Perbandingan Rute I awal CSO dan usulan ACO

	Rute Awal	Rute Usulan	Selisih	Efisiensi
Jarak Tempuh (km)	22,921	20,982	1,939	8,459%
Waktu Tempuh (menit)	34,387	31,479	2,908	8,457%
Biaya (Rupiah)	5.844	5.347	497	8,504%

Sedangkan, perbedaan tampilan Rute I awal dan usulan dapat dilihat pada Gambar 4 .



Gambar 4. (a) Rute I awal CSO (b) Rute I usulan ACO

Dari Gambar 4 dapat dilihat adanya perubahan dari urutan kunjungannya yang cukup signifikan dari rute awal CSO dan rute usulan menggunakan ACO. Tabel 4 menyajikan perbandingan total jarak dan waktu tempuh serta

biaya yang harus dikeluarkan sekali menyelesaikan Rute II.

Tabel 4. Perbandingan Rute II awal CSO dan usulan ACO

	Rute Awal	Rute Usulan	Selisih	Efisiensi
Jarak Tempuh (km)	26,772	25,785	0,987	3,687%
Waktu Tempuh (menit)	40,168	38,687	1,481	3,687%
Biaya (Rupiah)	6.823	6.571	252	3,693%

Dengan asumsi 1 bulan terdiri dari 30 hari dengan 4 hari libur (Minggu) maka terdapat 26 hari kerja yang berarti masing-masing 13 hari kerja untuk Rute I dan II. Oleh sebab itu, dapat diperoleh perbandingan total jarak dan waktu tempuh serta biaya yang harus dikeluarkan dalam 1 bulan. Dengan perhitungan masing-masing rute, yaitu Rute I dan II dikalikan 13 hari dapat kita ketahui perbandingan total jarak tempuh, waktu tempuh dan biaya yang harus dikeluarkan CSO dalam 1 bulan ditunjukkan Tabel 5.

Tabel 5. Perbandingan Rute awal CSO dan usulan ACO sebulan

	Rute Awal	Rute Usulan	Selisih	Efisiensi
Jarak Tempuh (km)	646,009	607,971	38,038	5,888%
Waktu Tempuh (menit)	969,215	912,158	57,057	5,887%
Biaya (Rupiah)	164.671	155.934	9.737	5,913%

Dapat diketahui bahwa dengan adanya rute usulan dengan ACO dapat diperoleh pengoptimalan/efisiensi sekitar 6% dibandingkan rute awal CSO dalam sebulan.

Pada perbandingan hasil rute usulan dengan algoritma *Ant Colony Optimization* terhadap rute awal CSO, mengalami pengoptimalan yang ditandai penurunan dari segi jarak, waktu dan biaya. Penurunan yang terjadi pun dapat dikatakan tidak terlalu signifikan. Hal ini disebabkan pengalaman CSO dalam membentuk rute awal sudah cukup baik, hanya saja masih dapat ditingkatkan untuk efisiensi yang lebih tinggi.

5. Kesimpulan

Rute awal yang dilakukan CSO memerlukan perbaikan demi memperoleh efisiensi waktu dan biaya. Penelitian ini bertujuan untuk mencari rute usulan terbaik menggunakan algoritma *Ant Colony Optimization*. Hasilnya rute awal CSO

dan rute usulan dengan ACO mengalami efisiensi (pengoptimalan) dalam hal jarak dan waktu tempuh serta biaya yang harus dikeluarkan. Rute I usulan ACO terhadap Rute I awal CSO mengalami efisiensi sekitar 8%. Sedangkan Rute II usulan ACO terhadap Rute II awal CSO mengalami efisiensi sekitar 4%. Lalu dibandingkan dalam sebulan Rute I dan II secara keseluruhan mengalami efisiensi sekitar 6%.

Referensi

- [1] M. Tisen, "Penentuan Rute Optimal Pengantaran Koran Menggunakan Travelling Salesman Problem," pp. 27–32, 2013.
- [2] Kusri and J. E. Istiyanto, "Penyelesaian Travelling Salesman Problem Dengan Algoritma Cheapest Insertion Heuristics Dan Basis Data," *J. Inform.*, vol. 8, no. 2, pp. 109–114–114, 2007, doi: 10.9744/informatika.8.2.pp.109-114.
- [3] U. Septima, "Implementasi Algoritma Ant Colony System untuk Penyelesaian Travelling Salesman Problem," *Jurnal Percikan*, vol. 92, 2008.
- [4] C. B. Hotang, "Implementasi Algoritma Ant Colony Optimization (ACO) Pada Masalah Pendistribusian Barang di Kantor Pos," *Universitas Sriwijaya, Palembang*, 2016.
- [5] I. K. Budayasa, *Teori Graf dan Aplikasinya*. Surabaya: Unesa University Press, 2007
- [6] I. Helene, et al., "Penyelesaian Masalah Travelling Salesman Problem Menggunakan Ant Colony System," *Jurnal Media Informatika*, volume 6, 2008.
- [7] I. S. Wardy, "Penggunaan graph dalam algoritma semut untuk melakukan optimisasi," *Program studi Teknik Informatika*, ITB, Bandung, 2007.
- [8] D. B. Paillin & J. M. Tupan, "Pemecahan Traveling Salesman Problem Menggunakan Teknik Branch and Bound dan Cheapest Insertion Heuristic (Studi Kasus : PT. Paris Jaya Mandiri – Ambon)," *Seminar Dan Konferensi Nasional IDEC, 2014*, 7–8, 2018.
- [9] Dian, "Algoritma Optimasi Untuk Penyelesaian Travelling Salesman Problem (Optimization Algorithm For Solving Travelling Salesman Problem)," *Jurnal Transformatika. Teknologi Informasi Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang*, volume 11(1), 2013
- [10] Delima, *Konsep dasar algoritma pemrograman dan bahasa pemrograman*, 2020.
- [11] M. Shahabudin Fikri, "Studi dan Impementasi Mengenal Algoritma Semut," *Jurnal Teknik Informatika dan Komputer*, volume 3, 2010.
- [12] L. N. Alfriany, "Implementasi Algoritma Ant Untuk Menentukan Jalur Terpendek Dalam Proses Pengiriman Surat Pos Dengan Menggunakan Visual Basic 6.0," *Jurnal Mitra Tahun XIV*, volume 2, 2008
- [13] R. Refianti, "Solusi Optimal Travelling Salesman Problem dengan Algoritma Ant Colony System," *Journal of Informatics and Computer*, 2005.
- [14] B. Santosa, "Tutorial on Ant Colony Optimization," 1992.
- [15] A. Leksono, "Ant Colony Optimization," Universitas Diponegoro, 2009.
- [16] Robitoh, "Perancangan Sistem Administrasi Pada Puskesmas Sarolangun Berbasis Web," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2018.
- [17] A. Y. Permana and P. Romadlon, "Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode SDLC Pada PT. Mandiri Land Prosperous Berbasis Mobile," *SIGMA – J. Teknol. Pelita Bangsa*, vol. 10, no. Volume 10 Nomor 2 Desember 2019, pp. 153–167, 2019.
- [18] M. Noviansyah, *Pengenalan Dasar Matlab*. Jakarta: Universitas Bina Sarana Informatika, 2019.
- [19] Y. Megalina, "Universitas Sumatera Utara Poliklinik Universitas Sumatera Utara," 2010.
- [20] B. Cahyono, "Penggunaan Software Matrix Laboratory (Matlab) Dalam Pembelajaran Aljabar Linier," *Phenom. J. Pendidik. MIPA*, vol. 3, no. 1, pp. 45–62, 2016, doi: 10.21580/phen.2013.3.1.174.