

Pengembangan Algoritma Ant Colony System Pada Heterogeneous Vehicle Routing Problem with Soft Time Window

Development of Ant Colony System Algorithm in Heterogeneous Vehicle Routing Problem with Soft Time Window

Sonna Kristina, Ricky Sianturi, Valian Janelven Wijaya

Program Studi Teknik Industri, Institut Teknologi Harapan Bangsa
E-mail: sonna@ithb.ac.id, ricky@ithb.ac.id, valianjwijaya@gmail.com

Abstrak

Setiap perusahaan umumnya memiliki sistem distribusi dan transportasi dalam menunjang pengiriman barang kepada customer. Diperlukan sistem distribusi yang efektif dan efisien sehingga biaya dari transportasi dalam perusahaan dapat diminimasi. Penelitian ini bertujuan untuk mengembangkan algoritma Ant Colony System (ACS) untuk model matematis Heterogeneous Vehicle Routing Problem with Soft Time Window (HVRPSTW) pada penentuan rute transportasi yang dapat meminimasi biaya pada perusahaan PT XYZ. HVRPSTW merupakan VRP yang mempertimbangkan kendaraan yang beragam dan jendela waktu dengan adanya biaya penalti yang dibebankan apabila kendaraan tiba di luar waktu yang telah ditentukan. Salah satu cara yang digunakan untuk menyelesaikan permasalahan VRP adalah metode metaheuristik ACS. Metode ACS diimplementasikan untuk menemukan rute kendaraan terbaik sesuai dengan kendala-kendala yang sudah ditentukan. Tahapan awal adalah mencari solusi awal menggunakan metode Nearest Neighbour yang akan digunakan sebagai pheromone awal. Proses pencarian rute pada ACS menggunakan tahapan tour construction lalu dilakukan update pheromone. Pemecahan masalah akan dilakukan dengan bantuan aplikasi Python. Hasil dari penelitian menunjukkan bahwa dihasilkan total jarak sebesar 1448,98 km dan total cost sebesar Rp. 3.582.367,86, di mana terjadi selisih jarak dengan penelitian sebelumnya menggunakan metode eksak sebesar 6,48 km (0,45%) dan selisih total biaya sebesar Rp. 42.248,86 (1,19%).

Kata kunci: ant colony optimization, vehicle routing problem, kapasitas kendaraan yang beragam, jendela waktu, biaya transportasi.

Abstract

Every enterprise commonly has distribution and transportation system to support the delivery of goods to customers. An effective and efficient distribution system is needed so that the costs of transportation can be minimized. This research is aimed to develop the Ant Colony System (ACS) algorithm for the Heterogeneous Vehicle Routing Problem with Soft Time Window (HVRPSTW) mathematical model in determining transportation routes that can minimize costs at PT XYZ. HVRPSTW is VRP which considers various vehicles and time window with a penalty fee charged if the vehicle arrives outside the specified time. One of the methods used to solve VRP problems is the metaheuristic ACS. ACS method is implemented to find the best vehicle route in accordance with predetermined constraints. The initial stages in ACS is to discover an initial solution as initial solution using the Nearest Neighbour method. The routing process in ACS begins with the tour construction stage then continue with updating the pheromone. The solution to the problem will be generated by Python. The results showed that the total distance generated using the ACS method is 1448,98 km and total cost is IDR 3.582.367,86, where the difference with previous research using the exact method is 6,68 km (0,45%) and IDR 42.248,86 (1,19%).

Keywords: ant colony optimization, vehicle routing problem, different vehicle capacity, time windows, transportation cost.

1. Pendahuluan

Proses distribusi merupakan salah satu bagian yang sangat penting dalam sistem operasional sebuah perusahaan. Secara umum distribusi memiliki fungsi untuk mengantar produk dari lokasi di mana produk itu diproduksi ke lokasi di mana produk tersebut akan digunakan (Pujawan dan Mahendrawathi, 2010). Distribusi tidak bisa lepas dari kegiatan transportasi yang sangat berperan penting dalam mendukung kegiatan distribusi dalam menyalurkan barang kepada *customer*. *Demand* yang terus bertambah akan membuat biaya transportasi akan semakin besar juga sesuai dengan rute dan moda transportasi yang digunakan. Pentingnya minimasi biaya transportasi akan membuat proses distribusi menjadi lebih optimal dan kinerja perusahaan menjadi efektif dan juga efisien, salah satunya dengan *Vehicle Routing Problem* (VRP). VRP adalah suatu model yang bertujuan untuk menemukan rute dengan biaya minimum untuk pengiriman suatu produk kepada sejumlah *customer* di beberapa lokasi yang berbeda, dengan menggunakan kendaraan, yang menggambarkan masalah transportasi sebagai model graf (G. Gunawan *et. al.*, 2012). VRP sendiri sudah banyak dikembangkan oleh berbagai peneliti untuk mendapatkan solusi yang optimum, beberapa diantaranya adalah CVRP yang mempertimbangkan kapasitas kendaraan, VRPTW yang mempertimbangkan jendela waktu, CVRPTW merupakan gabungan dari CVRP dengan VRPTW yang mempertimbangkan kapasitas kendaraan sekaligus dengan jendela waktu, HVRPSTW yang mempertimbangkan kapasitas kendaraan yang berbeda dan jendela waktu dengan biaya penalti yang dibebankan apabila kendaraan datang tidak sesuai waktu permintaan (M. Zirour, 2008).

Heterogeneous Vehicle Routing Problem with Soft Time window (HVRPSTW) merupakan pengembangan VRP yang sudah dikembangkan pada penelitian (H. Wijaya, 2019). Pengembangan model matematis ini merupakan pengembangan yang memperhatikan moda transportasi yang beragam dan memperhatikan faktor jendela waktu pada setiap lokasi untuk menyelesaikan permasalahan yang berasal dari penelitian sebelumnya dengan mempertimbangkan *fixed cost*, biaya parkir dan biaya tenaga kerja untuk penentuan rute dan moda transportasi pada PT XYZ, dimana perusahaan tersebut merupakan perusahaan distribusi elektronik dan aksesoris *handphone* di kota Bandung. Kelebihan dari penelitian (H. Wijaya, 2019) adalah dapat menemukan solusi yang optimum dikarenakan menggunakan metode solusi eksak dimana metode ini mencoba semua kemungkinan sampai didapat solusi yang optimum. Namun, sampai saat ini belum ada suatu metode eksak yang dapat menjamin keberhasilan solusi optimal dan waktu komputasi yang lama seiring dengan jumlah *customer* dan kendaraan. HVRPSTW dikarakteristikan ke dalam kelas *NP-hard* (*Nondeterministik Polynomial-hard*). Berdasarkan hal tersebut, banyak peneliti lebih memusatkan kepada pengembangan metode-metode pendekatan (*heuristic*) seperti *Simulated Annealing*, Algoritma Semut (*Ant Algorithm*), Algoritma Genetika, *Tabu Search*, dan lain sebagainya.

Penelitian ini mengembangkan metode *metaheuristic Ant Colony Sytem* (ACS) yang merupakan pengembangan dari ACO untuk model matematis hasil penelitian (H. Wijaya, 2019). Algoritma ACO telah banyak diaplikasikan dalam berbagai bidang yang mencakup beberapa persoalan yaitu, *Traveling Salesman Problem* (TSP) untuk mencari rute terpendek dalam sebuah *graph* menggunakan rute Hamilton, *Job-shop Scheduling Problem* (JSP) untuk mencari lintasan sejumlah n pekerjaan menggunakan sejumlah m mesin demikian sehingga seluruh pekerjaan diselesaikan dalam waktu yang seminimal mungkin dan *Vehicle Routing Problem* (VRP) untuk pengaturan rute kendaraan sehingga meminimalkan biaya (M. Dorigo dan T. Stutzle, 2004).

ACS merupakan metode yang digunakan untuk menyelesaikan masalah optimasi dimana inspirasi dalam memecahkan masalah tersebut berasal dari perilaku kumpulan atau kawanan semut dalam mencari sumber makanan dan metode *metaheuristic* ini dapat diterapkan dengan mudah pada dunia nyata dengan waktu penyelesaian yang lebih cepat dibandingkan dengan metode eksak (Karjono *et. al.*, 2016). Metode *Nearest Neighbour* akan diterapkan pada algoritma ACS sebagai solusi awal untuk pembentukan *pheromone* agar solusi dari metode ACS lebih presisi. Metode *metaheuristic* memerlukan metode *heuristic* di dalamnya agar solusi yang diperoleh tidak terjebak pada solusi *local* optimum dan melakukan pencarian di ruang solusi untuk menemukan solusi global (B. Santosa dan T. J. Ai, 2017).

Oleh karena itu rumusan masalah dari penelitian ini adalah bagaimana mengembangkan algoritma yang tepat dari ACS akan menyelesaikan permasalahan dan mendapatkan rute terpendek sehingga didapat minimasi biaya transportasi. Algoritma yang dikembangkan akan dikodekan menggunakan aplikasi *Python*.

2. Tinjauan Pustaka

2.1 Vehicle Routing Problem

VRP memiliki fungsi tujuan untuk meminimalkan biaya transportasi kendaraan dengan digambarkan sebagai suatu kasus dimana ada sejumlah kendaraan dengan kapasitas tertentu yang harus mengirim sejumlah barang dari suatu lokasi atau depot ke lokasi tujuan dengan asumsi jarak antara lokasi telah diketahui (Arvianto *et. al.*, 2018).

Formulasi VRP dapat dirumuskan sebagai berikut (S. Kandou, 2019):

Fungsi Tujuan:

$$\text{Min} \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} C_{ij} \cdot X_{ijk} \quad (1)$$

Dengan batasan:

$$\sum_{k \in V} \sum_{j \in N} X_{ijk} = 1, \forall i \in N \quad (2)$$

Setiap pelanggan hanya dapat dikunjungi oleh satu kendaraan tepat satu kali.

$$\sum_{i \in N} d_i \sum_{j \in N} X_{ijk} \leq q_i, \forall k \in V \quad (3)$$

Total jumlah permintaan dalam satu rute tidak melebihi kapasitas kendaraan.

$$\sum_{j \in N} X_{0jk} = 1, \forall k \in V \quad (4)$$

Menjamin kendaraan memulai perjalanan dari depot.

$$\sum_{i \in N} X_{ihk} - \sum_{j \in N} X_{hjk} = 0, \forall k \in V, \forall h \in V \quad (5)$$

Memastikan bahwa setiap kendaraan yang mengunjungi pelanggan setelah selesai akan meninggalkan pelanggan tersebut.

$$\sum_{i \in N} X_{i,n+1,k} = 1, \forall k \in V \quad (6)$$

Memastikan rute berakhir di depot.

$$X_{ijk} \in \{0,1\}, \forall i, j \in N, \forall k \in V \quad (7)$$

Variabel keputusan x_{ijk} merupakan bilangan biner.

Dengan Notasi:

- N : himpunan *node*
- V : himpunan kendaraan
- i, j, h : indeks *node*
- k : indeks kendaraan
- c_{ij} : biaya perjalanan dari i ke j
- d_i : jumlah permintaan pelanggan i
- q : kapasitas kendaraan

2.2 Heterogeneous Vehicle Routing Problem with Soft Time window (HVRPSTW)

HVRPSTW merupakan pengembangan dari VRP yang merupakan penggabungan dari *Heterogeneous Vehicle Routing Problem* dengan *Vehicle Routing Problem with Soft Time Window* (H. Wijaya, 2019). HVRP merupakan pengembangan VRP yang mempertimbangkan kapasitas kendaraan yang bervariasi serta *fixed cost* dan *variabel cost* dengan tujuan agar dapat mengetahui penggunaan kendaraan yang tepat sesuai dengan rute dan permintaan pelanggan dan didapat biaya transportasi yang minimal. VRPSTW merupakan pengembangan dari VRPTW yang mengizinkan kendaraan untuk datang melewati batas waktu namun dengan biaya denda yang harus dibayar.

Perumusan dari HVRPSTW dapat dirumuskan sebagai berikut:

Fungsi Tujuan:

$$\begin{aligned} \text{Min } Z = & \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} X_{ijk} * R_{ij} * BB_k + \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} X_{ijk} * ((DP_{jk} + \\ & EP_{jk}) * CP_k) + \sum_{k \in K} \sum_{i \in N, i \leq 1} \sum_{j \in N, j \geq 1} X_{ijk} * BT + \sum_{k \in K} \sum_{i \in N, i \leq 1} \sum_{j \in N, j \geq 1} X_{ijk} * \\ & WW_k + \sum_{k \in K} \sum_{i \in N, i \neq 1} \sum_{j \in N} X_{ijk} * TBP_{ik} \\ & * BP_k \end{aligned} \quad (8)$$

Di mana,

$$\text{Biaya perjalanan} = \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} X_{ijk} * R_{ij} * BB_k \quad (9)$$

$$\text{Biaya penalti} = \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} X_{ijk} * ((DP_{jk} + EP_{jk}) * CP_k) \quad (10)$$

$$\text{Biaya tenaga kerja} = \sum_{k \in K} \sum_{i \in N, i \leq 1} \sum_{j \in N, j \geq 1} X_{ijk} * BT \quad (11)$$

$$\text{Fixed cost} = \sum_{k \in K} \sum_{i \in N, i \leq 1} \sum_{j \in N, j \geq 1} X_{ijk} * WW_k \quad (12)$$

$$\text{Biaya parkir} = \sum_{k \in K} \sum_{i \in N, i \neq 1} \sum_{j \in N} X_{ijk} * TBP_{ik} * BP_k \quad (13)$$

Dengan pembatas:

$$\sum_{i \in N} \sum_{k \in K} X_{ijk} = 1, \forall j \in N, j \neq 1 \quad (14)$$

$$\sum_{j \in N} \sum_{k \in K} X_{ijk} = 1, \forall i \in N, i \neq 1 \quad (15)$$

$$\sum_{i \in N, i \neq p} X_{ipk} - \sum_{j \in N, j \neq p} X_{pjk} = 0, \forall k \in K, p \in N, p \neq 1 \quad (16)$$

$$U_{ik} - U_{jk} + N * X_{ijk} \leq N - 1, \forall k \in K, i \in N, i \neq 1, j \in N, j \neq 1 \quad (17)$$

$$\sum_{i \in N, i \neq 1} \sum_{j \in N} X_{ijk} * D_i \leq Q_k, \forall k \in K \quad (18)$$

$$X_{iik} = 0, \forall k \in K, i \in N \quad (19)$$

$$X_{ijk} = \{0,1\}, \forall k \in K, i \in N, j \in N \quad (20)$$

$$T_{ijk} = R_{ij} * 1/L_k, \forall k \in K, i \in N, j \in N \quad (21)$$

$$A_{jk} \geq B_{ik} + T_{ijk} - M(1 - X_{ijk}), \forall j \in N, j > 1, i \in N, k \in K \quad (22)$$

$$A_{jk} \leq B_{ik} + T_{ijk} + M(1 - X_{ijk}), \forall j \in N, j > 1, i \in N, k \in K \quad (23)$$

$$E_k \geq G_{jk}, \forall k \in K, i \in N, j \in N \quad (24 a)$$

$$G_{jk} \geq B_{ik} + T_{ijk} + M(X_{ijk} - 1), \forall k \in K, i \in N, j \in N \quad (24 b)$$

$$A_{ik} \leq B_{ik} - S_i, \forall i \in N, k \in K \quad (25)$$

$$\sum_{i \in N} X_{ijk} \leq 1, \forall k \in K, k \in K, j \in N, j = 1 \quad (26)$$

$$EP_{jk} = \begin{cases} F_j - A_{jk}, & F_j > A_{jk} \\ 0, & F_j < A_{jk} \end{cases}, \forall j \in N, k \in K \quad (27)$$

$$DP_{jk} = \begin{cases} B_{jk} - A_{jk} - S_j, & A_{jk} > F_j \\ B_{jk} - F_j - S_j, & A_{jk} < F_j \end{cases}, \forall j \in N, k \in K \quad (28)$$

$$TBP_{jk} = ((B_{jk} - A_{jk}) \text{ div } 60) + 1 \quad (29)$$

Persamaan (14) dan (15) merupakan pembatas untuk jumlah *maximal* kendaraan yang dapat melayani satu lokasi. Persamaan (16) merupakan pembatas kekontinuan rute. Persamaan (17) merupakan pembatas eliminasi subrute. Persamaan (18) merupakan pembatas jumlah muatan kendaraan. Pembatas (19) dan (20) merupakan pembatas bilangan biner. Persamaan (21)

merupakan pembatas waktu perjalanan. Persamaan (22) dan (23) merupakan pembatas waktu tiba dan keberangkatan kendaraan. Persamaan (24 a) dan (24 b) merupakan pembatas waktu penggunaan kendaraan. Persamaan (25) merupakan pembatas waktu keberangkatan. Persamaan (26) merupakan pembatas eliminasi multi trip. Persamaan (27) dan (28) merupakan pembatas besar biaya penalti. Persamaan (29) merupakan pembatas lama parkir.

Dengan notasi:

- D_i : Demand permintaan pengiriman pada lokasi i
- S_i : Waktu *unloading* pada lokasi i
- Q_k : Kapasitas maksimum kendaraan k
- BB_k : Biaya bensin untuk kendaraan k
- L_k : Kecepatan rata-rata kendaraan k
- E_k : Batas waktu penggunaan k
- CP_k : Biaya penalti kendaraan k
- WW_k : *fixecost* untuk kendaraan k
- BP_k : Biaya Parkir
- BT : Biaya Tenaga Kerja
- EP_{ik} : Lama waktu *early* penalti pada lokasi i dengan kendaraan k
- DP_{ik} : Lama waktu *delay* penalti pada lokasi i dengan kendaraan k
- G_{jk} : Waktu kendaraan k tiba di depot
- R_{ij} : Jarak antara lokasi i ke lokasi j
- C_{ijk} : Biaya kendaraan dari lokasi i ke lokasi j dengan kendaraan k
- T_{ijk} : Waktu perjalanan dari lokasi i ke lokasi j dengan menggunakan kendaraan k
- A_{ik} : Waktu kendaraan k datang ke lokasi i
- B_{ik} : Waktu keberangkatan kendaraan k dari lokasi i
- TBP_{jk} : Lama kendaraan k parkir di lokasi j
- U_{ij} : Variabel bantu untuk eliminasi sub tur
- X_{ijk} : Rute terpilih dari lokasi i ke lokasi j dengan kendaraan pengiriman k

2.3 Ant Colony Optimization (ACO)

Algoritma semut pertama kali diperkenalkan oleh Moysos dan Manderick, lalu dikembangkan secara meluas oleh Marco Dorigo. Algoritma semut merupakan salah satu pendekatan *metaheuristic* pada VRP dengan mengikuti perilaku semut ketika mencari makanannya (M. Dorigo and T. Stützle, 2004). Semut akan memilih jalan dan menyimpan *pheromone* sebagai panduan bagi semut yang lain untuk mengikuti jalan mereka. *Pheromone* akan membusuk ketika jalur terlalu panjang sehingga probabilitas dari koloni untuk mengikuti jalan dari semut sebelumnya akan menjadi kecil dibanding dengan jalur yang pendek. ACO memiliki beberapa jenis, yaitu *Ant System* (AS), *Max-Min Ant System* (MMAS) dan *Ant Colony System* (ACS) makanannya (M. Dorigo and T. Stützle, 2004).

2.3.1 Ant System

Ant System merupakan jenis ACO yang pertama kali ditemukan. Karakteristik utamanya adalah pada setiap iterasi, nilai *pheromone* diperbaharui oleh semua semut yang telah dibangun makanannya (M. Dorigo and T. Stützle, 2004). *Ant System* memiliki beberapa tahapan, yaitu:

1. Tour Construction

Dalam menentukan rutennya, semut akan memilih secara acak lokasi tujuan mereka dengan probabilitas. Berikut adalah formulasi probabilitas penentuan lokasi:

$$P_{ij}^m = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in N(s^P)} \tau_{il}^\alpha \cdot \eta_{il}^\beta}, & \text{if } l \in N(s^P) \\ 0, & \text{otherwise} \end{cases} \quad (30)$$

Di mana,

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (31)$$

Dengan notasi:

- i, j : indeks lokasi
- l : *node* yang belum dikunjungi oleh semut k
- m : semut
- P_{ij}^m : probabilitas semut k menuju *node* selanjutnya
- α : parameter intensitas jejak semut
- β : parameter visibilitas jarak antar *node*
- τ_{ij} : intensitas *pheromone* pada *node* i, j
- $N(s^P)$: kumpulan dari *feasible component*
- η_{ij} : invers dari jarak antar *node* i, j
- d_{ij} : jarak antar *node* i, j
- τ_{il} : intensitas *pheromone* pada *node* i, l
- η_{il} : invers dari jarak antar *node* i, l

2. Pheromone Update

Setelah semut selesai dalam memilih rute, intensitas dari *pheromone* akan diperbaharui dengan formula:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{m=1}^M \Delta\tau_{ij}^m \quad (32)$$

Di mana,

$$\Delta\tau_{ij}^m = \begin{cases} C/L_m, & \text{if ant } k \text{ used edge } (i, j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases} \quad (33)$$

Dengan notasi:

- i, j : indeks lokasi
- ρ : intensitas penguapan *pheromone*
- m : semut
- M : jumlah maksimal semut
- $\Delta\tau_{ij}^m$: jumlah *pheromone* pada *node* i, j oleh semut m
- C : konstanta
- L_m : panjang *tour* yang dibuat oleh semut m

2.3.2 Max-Min Ant System (MMAS)

Max-Min Ant System merupakan pengembangan dari ACO yang sebelumnya. Pada MMAS, hanya semut terbaik saja yang dapat mempengaruhi jejak *pheromone* (M. Dorigo and T. Stutzle, 2004). Berikut adalah tahapan dalam MMAS:

1. Pheromone Update

$$\tau_{ij} \leftarrow [(1 - \rho) \cdot \tau_{ij} + \sum_{m=1}^M \Delta\tau_{ij}^{best}]_{\tau_{min}}^{\tau_{max}} \quad (34)$$

Di mana nilai τ_{max} dan τ_{min} merupakan batas atas dan batas bawah pada *pheromone*, dengan operator $[x]_b^a$ adalah:

$$[x]_b^a = \begin{cases} a & \text{if } x > a \\ b & \text{if } x < b \\ x & \text{otherwise} \end{cases} \quad (35)$$

Dan $\Delta\tau_{ij}^{best}$ adalah:

$$\Delta\tau_{ij}^{best} = \begin{cases} 1/L_{best}, & \text{if } (i,j) \text{ belongs to best tour} \\ 0, & \text{otherwise} \end{cases} \quad (36)$$

Dengan notasi:

- ρ : intensitas penguapan *pheromone*
- m : jumlah semut
- $\Delta\tau_{ij}^{best}$: jumlah *pheromone* pada *node* i,j oleh semut terbaik
- L_{best} : panjang *tour* yang dibuat oleh semut terbaik

2.3.3 Ant Colony System (ACS)

Penelitian ini menggunakan algoritma dari *Ant Colony System* (ACS) karena *Ant Colony System* (ACS) merupakan pengembangan dari ACO yang lebih baik dibandingkan dengan *Ant System* (AS), di mana yang membedakan metode ACS dengan AS adalah adanya penambahan *local pheromone update* ketika proses *update pheromone* pada akhir konstruksi rute (T.T.H. Diep dan N. Van Hop, 2018). Pada *update pheromone* metode AS, dilakukan pada semua rute yang terbentuk, sedangkan pada metode ACS ada tahapan *global update pheromone* sehingga semut akan lebih mudah menentukan titik selanjutnya dan tahapan *local update pheromone* sehingga *pheromone* pada rute akan berkurang dan tidak akan dilewati lagi oleh semut yang lainnya.

Kebiasaan dari metode ACS dibandingkan metode ACO yang lain adalah pada *tour construction* dimana ACS memilih jalur dengan *pseudorandom proportional rule*, sedangkan metode ACO lainnya berdasarkan *random proportional rule* (A. Leksono, 2009). Berikut adalah tahapan dari ACS:

1. Tour Construction

Ketika pada *node* i , semut k akan memilih *node* j dengan formulasi:

$$j = \begin{cases} \operatorname{argmax}_{l \in N(s^k)} [\tau_{il} (\eta_{il})^\beta], & \text{if } q \leq q_0 \\ J, & \text{otherwise} \end{cases} \quad (37)$$

Di mana, J merupakan persamaan (30) dengan nilai $\alpha = 1$. Dengan notasi:

- i,j : indeks lokasi
- l : *node* yang belum dikunjungi oleh semut k
- β : parameter visibilitas jarak antar *node*
- τ_{il} : intensitas *pheromone* pada *node* i,l
- $N(s^k)$: kumpulan dari *feasible component*
- η_{il} : invers dari jarak antar *node* i,l
- q : bilangan random yang terdistribusi *uniform* [0,1]
- q_0 : parameter ($0 \leq q_0 \leq 1$)

2. Global Pheromone Update

Pembaharuan *pheromone* hanya dapat dilakukan oleh semut terbaik dengan formulasi:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij}^{best} \quad (38)$$

Di mana,

$$\Delta\tau_{ij}^{best} = \begin{cases} 1/TC_{best}, & \text{if } (i,j) \text{ belongs to best tour} \\ 0, & \text{otherwise} \end{cases} \quad (39)$$

Dengan notasi:

- i, j : indeks lokasi
- ρ : intensitas penguapan *pheromone*
- τ_{ij} : intensitas *pheromone* pada *node* i, j
- ρ : intensitas penguapan *pheromone*
- $\Delta\tau_{ij}^{best}$: jumlah *pheromone* pada *node* i, j oleh semut terbaik
- TC_{best} : Total *cost* yang dibuat oleh semut terbaik

3. Local Pheromone Update

Semut menggunakan aturan *local pheromone update* melewati (i, i) dengan formulasi:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \quad (40)$$

Di mana,

$$\tau_0 = \frac{1}{n \cdot TC_{initial}} \quad (41)$$

Dengan notasi:

- i, j : indeks lokasi
- ρ : intensitas penguapan *pheromone*
- n : total dari jumlah *customer*
- τ_0 : inisialisasi *pheromone*
- $TC_{initial}$: total initial *cost*

2.4 Nearest Neighbour

Metode *Nearest Neighbour* merupakan metode *heuristic* yang dapat memecahkan permasalahan penentuan rute kendaraan atau VRP (N. Mardiani *et. al.*, 2014). *Nearest Neighbour* adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data yang jaraknya paling dekat dengan objek tersebut. Metode *Nearest Neighbour* sering disebut juga sebagai pencarian jarak atau pencarian titik terdekat, dimana sebuah kendaraan k akan menuju lokasi selanjutnya yang belum dikunjungi dengan permintaan dari lokasi tersebut tidak melebihi kapasitas kendaraan k berdasarkan jarak yang terdekat dengan lokasi asal. Apabila *demand* melebihi kapasitas kendaraan k maka pengiriman dilakukan lebih dari satu kali, di mana kendaraan menuju depot terlebih dahulu untuk *loading* lalu menuju ke tempat terdekat selanjutnya.

Langkah-langkah secara umum dalam menyelesaikan permasalahan dengan menggunakan Algoritma *Nearest Neighbour* adalah sebagai berikut:

1. Untuk tur pertama ($t=1$) dan rute pertama ($r=1$), lokasi awal berada pada depot, lanjutkan ke langkah 2.
2. Cari lokasi tujuan yang paling dekat dengan lokasi awal, lalu hubungkan kedua titik tersebut, lanjutkan ke langkah 3.
3. Set lokasi terakhir sebagai titik awal, ulangi langkah 2 hingga semua titik dikunjungi. Jika semua titik sudah dikunjungi lanjutkan ke langkah 4.
4. Penentuan rute berhenti ketika semua lokasi sudah selesai dilayani.

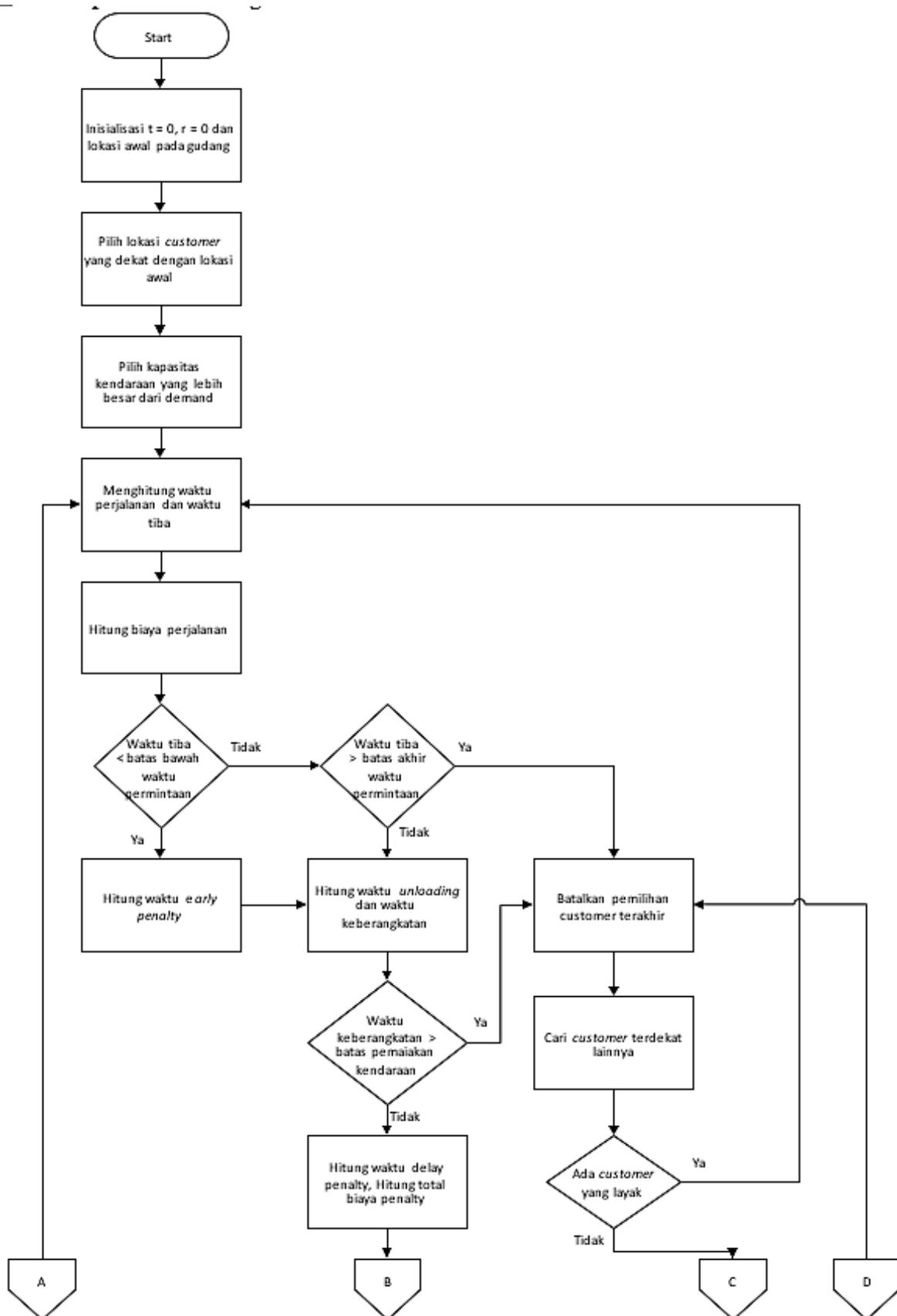
2.5 Python

Python merupakan salah satu bahasa pemrograman yang mudah untuk dipelajari dibandingkan dengan bahasa pemrograman lainnya (J. Enterprise, 2016). *Python* merupakan bahasa pemrograman yang sifatnya interpretatif. *Python* pertama kali dikembangkan oleh Guido van Rossum yang merupakan kelahiran Belanda pada tahun 1990 di Amsterdam. Nama *Python* sendiri berasal dari acara televisi Monty Python's Flying Circus yang merupakan acara televisi kesukaan dari Guido.

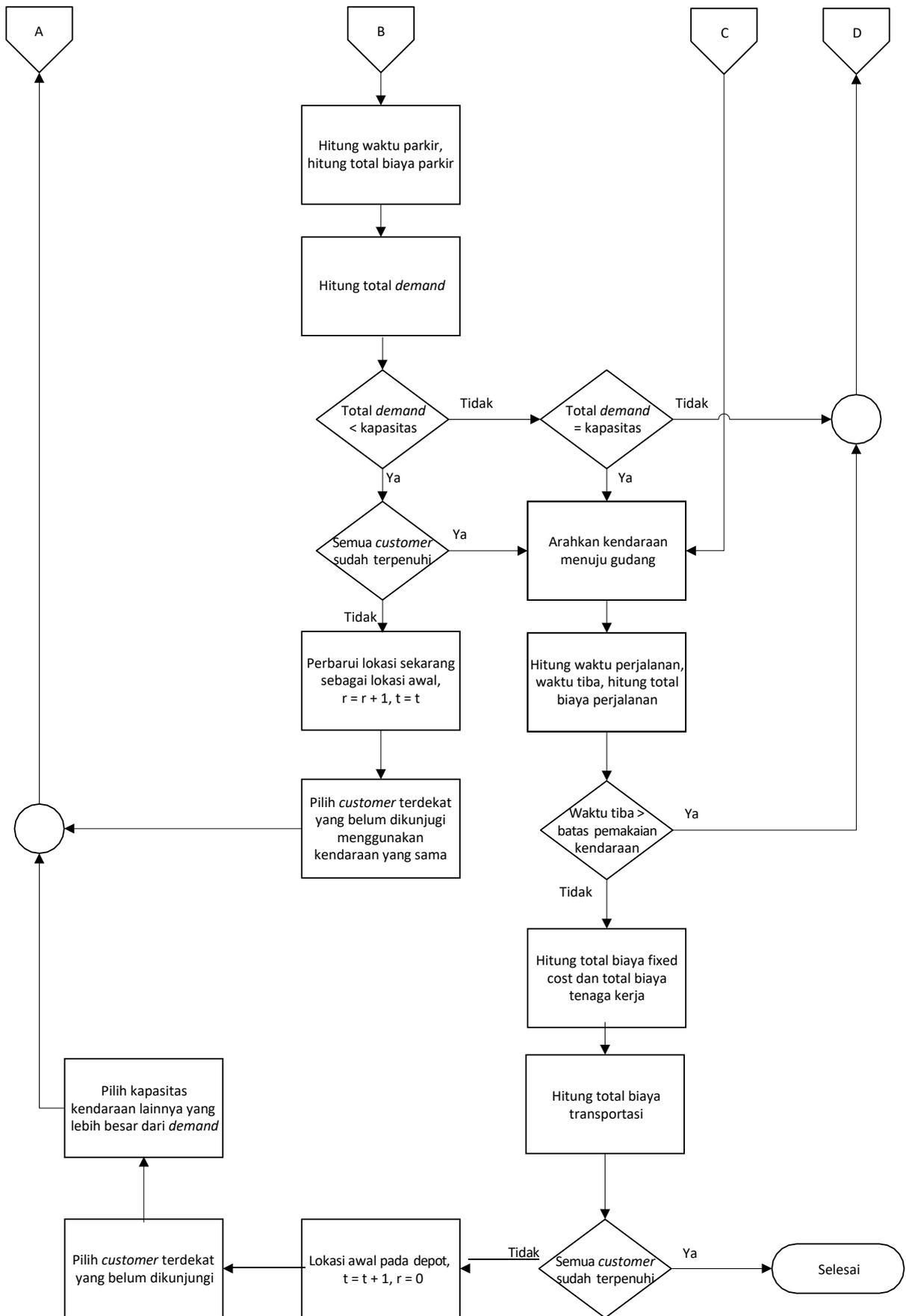
Python merupakan pemrograman *multiplatform* yang berarti dapat dijalankan di berbagai *platform* sistem operasi. Fitur-fitur pada *Python* adalah:

- Memiliki modul-modul yang 'siap pakai' untuk berbagai keperluan.
- Memiliki struktur bahasa yang jelas, sederhana, dan mudah dipelajari.
- Memiliki system pengelolaan memori otomatis
- Bersifat modular sehingga mudah dikembangkan dengan menciptakan modul-modul baru.

3. Pengembangan Algoritma
 3.1 Tahapan *Nearest Neighbour*

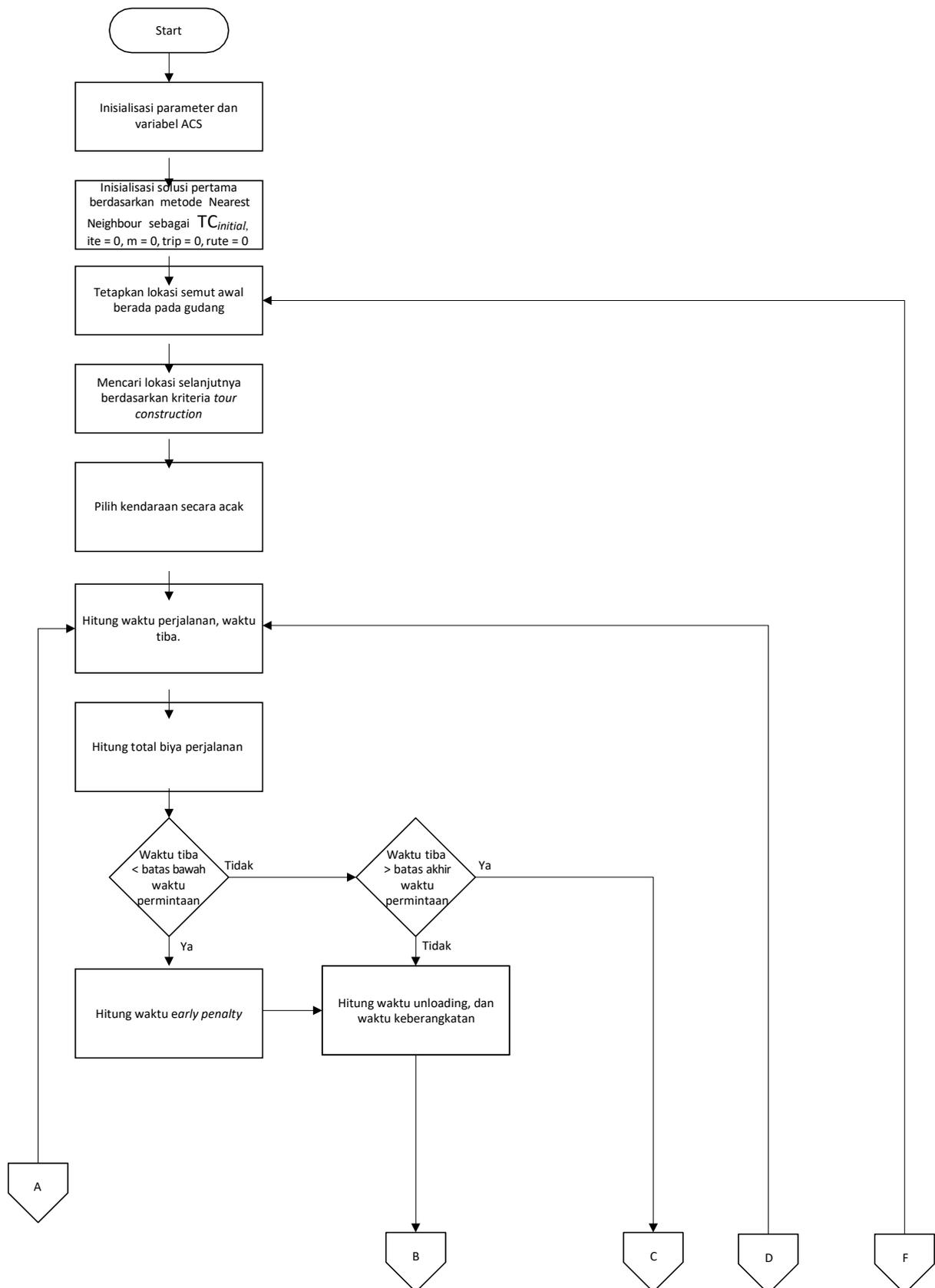


Gambar 1. Bagan Alir *Nearest Neighbour*



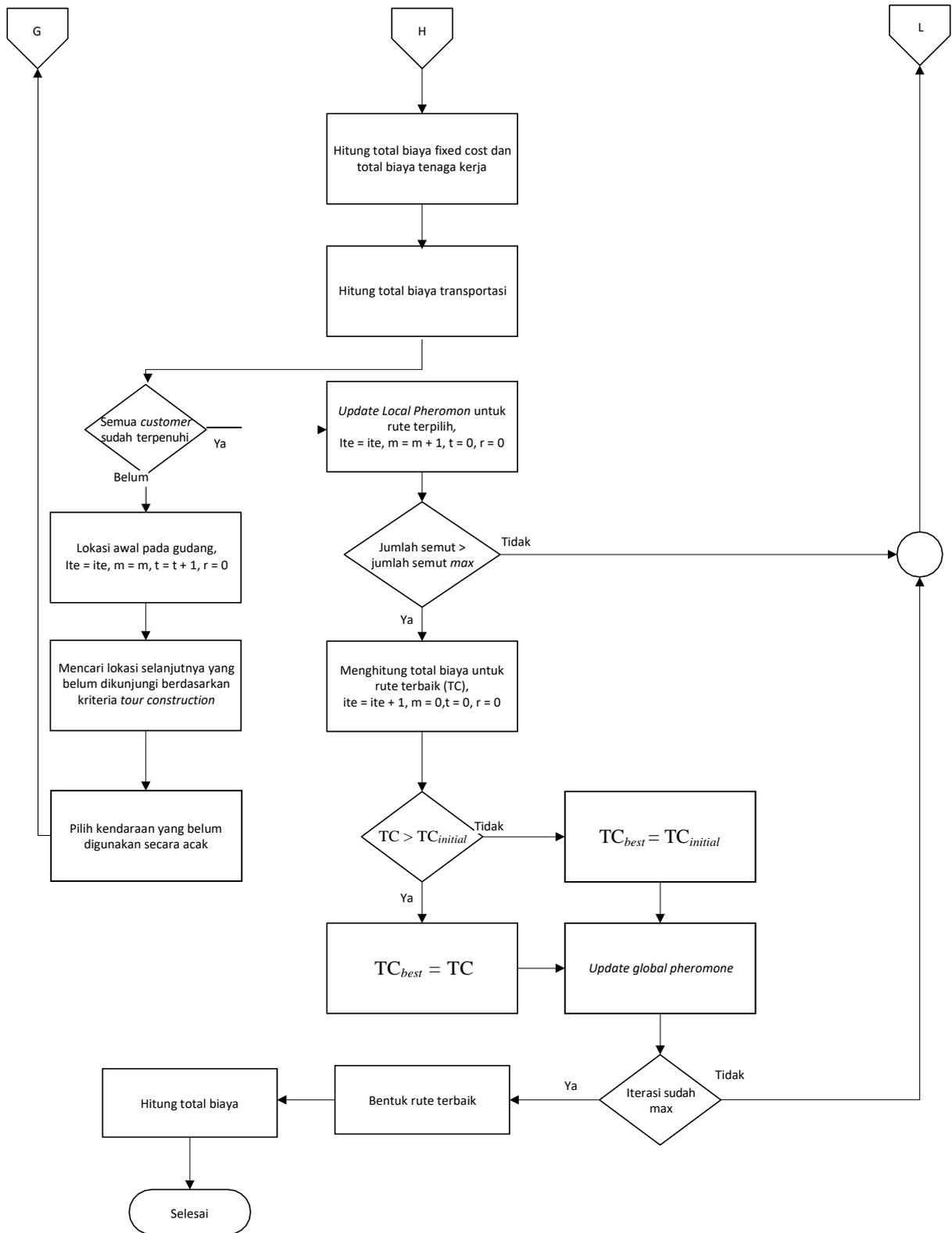
Gambar 1. Bagan Alir Nearest Neighbour (lanjutan)

3.2 Tahapan Ant Colony System



Gambar 2. Diagram Alir Ant Colony System

PENGEMBANGAN ALGORITMA ANT COLONY SYSTEM (Sonna K., dkk.)



Gambar 2. Diagram Alir Ant Colony System (lanjutan)

3.2 Analisis Sensitifitas

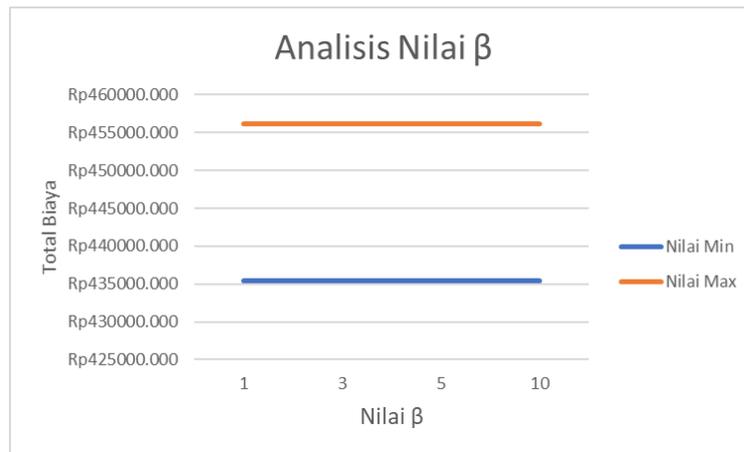
Parameter memiliki dampak yang signifikan pada kinerja algoritma koloni semut, oleh karena itu, uji sensitifitas dibuat untuk mengetahui hubungan antara parameter dengan hasil dari ACS. Parameter yang akan diuji adalah:

1. Visibilitas jarak antar *node* (β)
2. Intensitas penguapan *pheromone* (ρ)
3. Probabilitas aturan transisi (q_0)

Input dari analisis sensitifitas merupakan data *real* dari perusahaan pada tanggal 2 Januari 2017 dengan jumlah 8 *customer* dan 6 kendaraan. Setiap pengujian dilakukan 50 replikasi.

3.3.1 Visibilitas Jarak Antar Node (β)

Nilai β memutuskan seberapa besar nilai visibilitas jarak antar *node* pada *heuristic* dapat dipertimbangkan dalam mekanisme pemilihan rute. Ketika nilai β meningkat maka peluang semut memilih jalur yang terpendek pun akan meningkat. Dalam penelitian ini, nilai β diuji dengan nilai 1, 3, 5 dan 10 (T.T.H. Diep dan N. Van Hop, 2018). Masing- masing dengan nilai $\rho = 0,1$, $q_0 = 0,9$, iterasi = 100, dan jumlah semut = 10. Hasil dari pengujian dapat dilihat pada gambar 6.

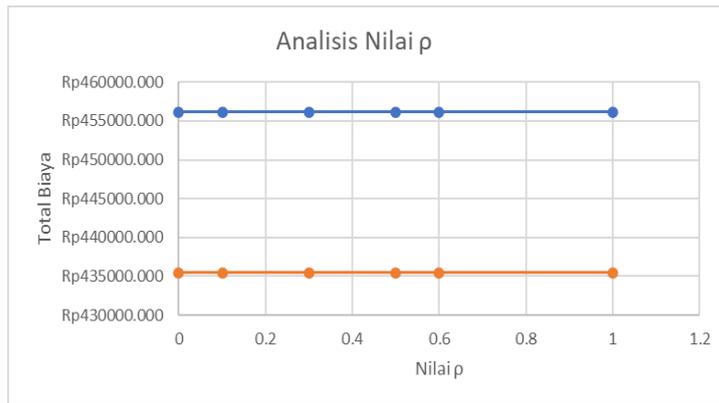


Gambar 3. Hasil Pengujian β

Berdasarkan gambar 3 menunjukkan hubungan antara β dengan total biaya transportasi, didapatkan nilai maksimum dan minimum dari hasil pengujian adalah konstan yang berarti nilai dari total biaya transportasi tidak sensitif terhadap perubahan nilai β . Berdasarkan 50 iterasi, didapatkan rata-rata nilai β optimum berada pada nilai $\beta = 0,3$.

3.3.2 Intensitas Penguapan Pheromone (ρ)

Nilai ρ ditetapkan sebagai penunda konvergensi semut yang lebih cepat menuju suboptimal. Nilai ρ digunakan sebagai parameter dalam prosedur pembaruan *pheromone*. Dalam penelitian ini, nilai ρ diuji dengan nilai 0; 0,1; 0,3; 0,5 dan 1 (T.T.H. Diep dan N. Van Hop, 2018). Masing-masing dengan nilai $\beta = 1$, $q_0 = 0,9$, iterasi = 100, dan jumlah semut = 10. Hasil dari pengujian dapat dilihat pada gambar 4.

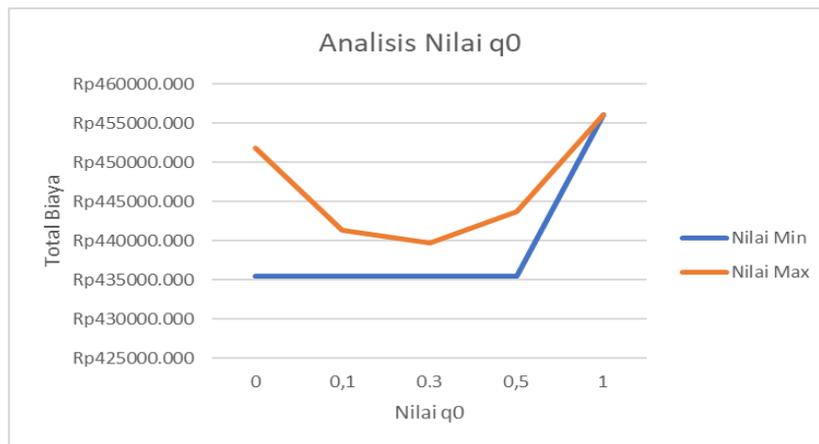


Gambar 4. Hasil Pengujian ρ

Berdasarkan gambar 4 menunjukkan hubungan antara ρ dengan total biaya transportasi, didapatkan untuk nilai maksimum dan minimum dari pengujian tersebut adalah konstan yang berarti nilai dari total biaya transportasi tidak sensitif terhadap perubahan nilai ρ . Berdasarkan 50 iterasi, didapatkan rata-rata nilai ρ optimum berada pada nilai $\rho = 0,1$.

3.3.3 Probabilitas Aturan Transisi (q_0)

Nilai q_0 ditetapkan untuk memutuskan apakah semut melakukan eksploitasi atau melakukan eksplorasi dalam pencarian rute. Semakin besar nilai q_0 maka akan semakin besar juga kemungkinan semut dalam melakukan eksploitasi untuk melakukan pencarian rute. Semakin kecil nilai q_0 maka semut akan memiliki kecenderungan memilih rute berikutnya secara acak. Dalam pengujian ini nilai q_0 diuji dengan nilai 0; 0,1; 0,3; 0,5 dan 1 (T.T.H. Diep dan N. Van Hop, 2018). Masing- masing dengan nilai $\beta = 1$, $\rho = 0,1$, iterasi = 100, dan jumlah semut = 10. Hasil dari pengujian dapat dilihat pada gambar 8.



Gambar 5. Hasil Pengujian q_0

Berdasarkan gambar 5, menunjukkan hubungan antara q_0 dengan total biaya transportasi, didapatkan pengujian nilai total biaya transportasi sensitif terhadap perubahan nilai q_0 dikarenakan total biaya maksimum dan minimum yang didapat tidak konstan. Sehingga berdasarkan gambar 5, maka nilai q_0 yang optimum berada pada nilai $q_0 = 0,3$.

4. Hasil Perhitungan Biaya Transportasi

Perhitungan biaya transportasi dilakukan dengan bantuan dari aplikasi *Python* untuk mencari rute yang optimum pada model matematis HVRPSTW menggunakan ACS. Hasil perhitungan transportasi secara lengkap dengan parameter ACS $\beta = 3$, $\rho = 0,1$, $q_0 = 0,3$, iterasi = 1000, dan jumlah semut = 10.

Berikut hasil perhitungan biaya transportasi untuk bulan Januari 2017 menggunakan metode eksak yang didapat dari penelitian sebelumnya dan metode ACS pada penelitian ini:

Tabel 1. Hasil Perbandingan Biaya Transportasi

	Total Jarak	Total Biaya Transportasi
Eksak	1442,5 Km	Rp 3.540.119,00
ACS	1448,98 Km	Rp 3.582.367,86
Selisih	6,48 Km (0,45%)	Rp 42.248,86 (1,19%)

Berdasarkan tabel 1, hasil perhitungan biaya transportasi menggunakan ACS mampu mendekati solusi optimum yang menggunakan metode eksak. Selisih dari metode ACS dan metode eksak sebesar 1,19%.

5. Kesimpulan

Berdasarkan rumusan masalah dan tujuan penelitian yang sudah dijabarkan, maka dapat diambil kesimpulan bahwa metode *metaheuristic* ACS dapat diterapkan dan berhasil dikembangkan untuk model matematis HVRPSTW yang terbukti dengan dihasilkannya total biaya dengan selisih sebesar 1,19% dan selisih jarak 0,45% saja terhadap penelitian milik Wijaya, 2019 [3], yang menggunakan metode eksak dengan hasil perhitungan total transportasi pada metode ACS sebesar Rp. 3.582.367,86 dan total jarak sebesar 1448,98 Km dengan total waktu komputasi sebesar 98,899 detik.

Daftar Pustaka

- A. Leksono (2009), *Algoritma Ant Colony Optimization (ACO) Untuk Menyelesaikan Masalah Traveling Salesman Problem (TSP)*, Undergraduate Thesis, Program Studi Matematika, Universitas Diponegoro.
- Arvianto, et al. (2018), *Penerapan Simulasi dan Reliabilitas Pada Model Vehicle Routing Problem (VRP) Dengan Permintaan Probabilistik*, Jurnal SIMETRIS, Vol. 9 No. 1, ISSN: 2252-4983.
- B. Santosa, dan T. J. Ai (2017), *Pengantar Metaheuristik Edisi Pertama*, Penerbit ITS Tekno Sains, Surabaya.
- G. Gunawan, et al. (2012), *Optimasi Penentuan Rute Kendaraan Pada Sistem Distribusi Barang Dengan Ant Colony Optimization*, Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2012, Vol. 2 No.1, ISBN: 979-26-0255-0.
- H. Wijaya (2019), *Penyusunan Model Matematis Heterogeneous Vehicle Routing Problem with Soft Time window Untuk Menentukan Rute Transportasi Yang Dapat Meminimasi Biaya Transportasi*, Program Studi Teknik Industri, Institut Teknologi Harapan Bandung.
- I.N. Pujawan dan E.R., Mahendrawathi (2010), *Supply Chain Management*, Edisi Kedua, Surabaya.
- J. Enterprise, (2016), *Trik Cepat Menguasai Pemrograman Python*, Penerbit Elex Media Komputindo, Jakarta.
- Karjono, et al. (2016), *Ant Colony Optimization*, Jurnal TICOM Vol. 2 No. 3.
- M. Dorigo dan T. Stützle, (2004), *Ant Colony Optimization*, Bradford Books, MA: MIT Press.
- M. Zirour (2008), *Vehicle Routing Problem: Models and Solution*, Journal of Quality Measurement and Analysis, JQMA 4(1) 205-218.

PENGEMBANGAN ALGORITMA ANT COLONY SYSTEM (Sonna K., dkk.)

N. Mardiani, et al. (2014), *Penentuan Rute untuk Pendistribusian BBM Menggunakan Algoritma Nearest neighbour (Studi Kasus di PT X)*, Jurnal Online Institut Teknologi Nasional, Vol.1 No. 4, ISSN:2338-5081.

S. Kandou (2019), *Penentuan Rute Transportasi Dengan Vehicle Routing Problem dan Time Window Pada CV. Cihanjuang Teknik*, Program Studi Teknik Industri, Institut Teknologi Harapan Bandung.

T.T.H. Diep dan N. Van Hop (2018), *Heterogeneous Fleet Vehicle Routing Problem with Time Window and Backlog Orders: An Ant Colony Optimization Approach*, Conference: The 13th International Congress on Logistics and SCM Systems (ICLS 2018), At HCMC, Vietnam.