

## **BAB II**

### **KAJIAN LITERATUR**

#### **2.1. Kajian Deduktif**

Pada bab ini berisi teori-teori yang dijadikan landasan dan metode yang digunakan dalam menyelesaikan permasalahan yang terdapat pada penelitian ini.

##### **2.1.1. Distribusi Produk**

Pada suatu sistem rantai pasok, produk yang telah dihasilkan oleh manufaktur, untuk dapat dipasarkan dan diterima oleh konsumen akan melalui proses distribusi (Chopra & Meindl, 2010). Menurut Kotler & Armstrong (2008) distribusi adalah aktivitas yang dilakukan oleh produsen untuk menyediakan produk yang telah dihasilkan kepada konsumen sasaran. Sementara itu, distribusi menurut Tjiptono (2008) adalah kegiatan pemasaran berupa penyampaian barang atau jasa yang telah dihasilkan oleh produsen agar dapat diterima konsumen sehingga produk dan jasa tersebut dapat digunakan sesuai dengan kebutuhan. Fungsi dan kegunaan dari barang dan jasa akan meningkat setelah barang dan jasa tersebut dikonsumsi atau digunakan oleh konsumen. Pada proses distribusi, sistem transportasi dibutuhkan untuk memindah satu produk dari tempat produksi ke tempat lain sesuai dengan rantai pasok. Pihak yang terlibat pada sistem transportasi dalam suatu rantai pasok adalah pengirim dan pengangkut.

Menurut Siahaya (2013) terdapat beberapa model distribusi, diantaranya adalah sebagai berikut:

1. Pengiriman langsung

Pada model distribusi ini, tidak terdapat distributor dan *retailer*. Hal ini karena transaksi dan pengiriman produk dilakukan secara langsung oleh produsen dan

peran dari distributor atau retailer digantikan oleh sarana virtual seperti internet. Tujuan dari model distribusi ini adalah untuk meminimalkan biaya distribusi.

2. Pengiriman produk melalui transit

Pada model distribusi ini, terdapat beberapa fasilitas transit yang terlibat. Fasilitas tersebut dikelola oleh distributor atau retailer yang mendistribusikan produk. Hal ini karena terdapat produk yang membutuhkan perakitan yang disebabkan komponen-komponen pembentuknya berasal dari beberapa produsen.

3. Distribusi produk melalui distributor

Pada model distribusi ini, produsen menentukan distributor yang bertugas untuk menyalurkan produk ke konsumen. Distributor melakukan fungsi penerimaan (*receiving*), penyimpanan (*storage*), hingga fungsi distribusi (*distribution*) produk sampai ke tangan konsumen.

4. Distribusi melalui desentralisasi

Pada model distribusi ini, produsen menempatkan distributor secara terpisah di setiap daerah. Penempatan distributor disesuaikan dengan segmentasi pasar sehingga dapat mendekatkan produk kepada konsumen dengan tujuan agar dapat meningkatkan layanan kepada konsumen dan untuk menghindari ketidakmampuan dalam memenuhi permintaan konsumen (*stock-out*).

5. Pengambilan langsung oleh konsumen

Pada model distribusi ini, konsumen memperoleh produk dengan cara mengambil langsung ke produsen. Produsen menyerahkan barang dengan berdasarkan sistem *cross docking*.

Mencari sebuah solusi terbaik pada proses distribusi merupakan bagian dari proses *decision-making* yang terdapat pada suatu *supply chain* (Moon et al., 2012). *Supply Chain Management* adalah gabungan dari proses produksi serta aktivitas-aktivitas produksi mulai dari *supplier* yang menyediakan bahan baku, proses pengolahan bahan baku sehingga memiliki nilai tambah, sampai penyimpanan barang jadi dan proses pendistribusian barang jadi ke konsumen (Pujawan, 2005). Oleh karena itu, distribusi merupakan salah satu kegiatan penting pada suatu rantai pasok karena apabila proses ini berlangsung secara baik, maka produk yang dihasilkan akan dapat segera dipasarkan dan diterima oleh konsumen. Selain itu, apabila proses distribusi yang dilakukan memiliki kualitas yang baik, maka kepuasan konsumen kepada produsen akan meningkat.

### 2.1.2. Optimasi

Optimasi adalah mencari nilai terbaik dengan cara memaksimalkan atau meminimalkan suatu fungsi. Menurut Tarmizi (2005) optimasi merupakan suatu cara untuk mendapatkan nilai paling maksimal atau paling minimal dari suatu fungsi yang dibatasi oleh faktor-faktor tertentu. Apabila permasalahan diselesaikan dengan mendapatkan nilai yang paling maksimal, maka keputusan tersebut adalah maksimasi.

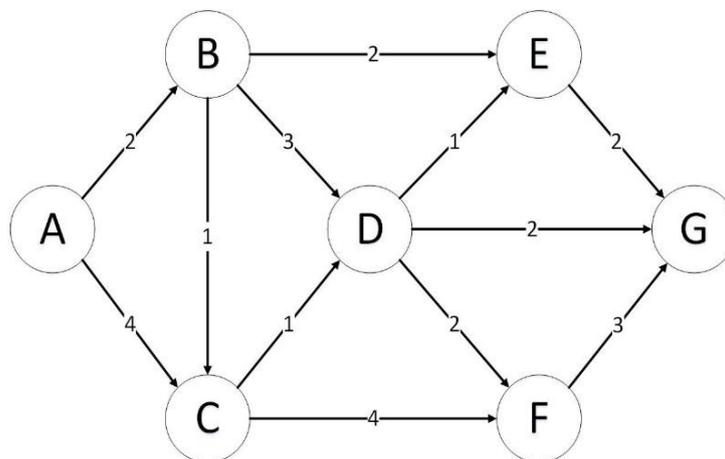
Optimasi mengarah pada proses untuk mencari nilai yang optimal. Nilai optimal merupakan suatu nilai hasil dari penyelesaian suatu masalah, yang mana nilai tersebut adalah nilai yang terbaik dari semua solusi nilai yang ada. Nilai optimal dapat berbentuk nilai maksimal atau nilai minimal dari suatu fungsi. Berikut ini adalah beberapa contoh permasalahan mengenai optimasi (Yuliani et al., 2008) :

1. Menentukan jalur terpendek dari suatu tempat ke tempat yang lain.
2. Menentukan jumlah pekerja minimal dalam mengerjakan suatu proses produksi sehingga dapat meminimalkan pengeluaran untuk biaya pekerja dengan hasil produksi yang tetap maksimal.
3. Mengatur rute kendaraan umum agar dapat menjangkau semua lokasi
4. Mengatur *routing* jaringan kabel telepon sehingga dapat meminimalkan biaya pemasangan kabel.

### 2.1.3. Jalur Terpendek

Permasalahan jalur terpendek adalah suatu persoalan optimasi yang bertujuan untuk menentukan jalur minimum antara dua simpul atau lebih yang saling berhubungan (Alamsyah, 2010). Menurut Saptono & Hidayat (2007) jalur terpendek adalah suatu jaringan pengarah yang menunjukkan perjalanan seseorang dari satu kota ke kota lain dimana jalur yang dicari adalah jalur terpendek dibandingkan jalur-jalur lain dan perjalanan tersebut mengarah pada satu titik. Pada umumnya, permasalahan mengenai penentuan jalur terpendek direpresentasikan dalam bentuk graf. Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges*). Suatu Graf  $G$  terdiri dari dua himpunan yaitu himpunan  $V$  dan himpunan  $E$ . Himpunan  $V$  merupakan himpunan simpul yang terbatas dan tidak kosong. Himpunan  $E$  adalah himpunan busur

yang menghubungkan sepasang simpul. Suatu graf ABCDEFG dapat dilihat pada Gambar 2.1 berikut.



Gambar 2.1 Graf Berbobot ABCDEFG

Sumber : Saptono & Hidayat (2007)

Pada gambar diatas, apabila seseorang berasal dari kota A dan akan melakukan perjalanan dengan tujuan kota G, maka terdapat beberapa kombinasi perjalanan yang tersedia. Berikut ini adalah kombinasi perjalanan dari Kota A dengan kota tujuan G yang dapat dilalui:

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow G$

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow F \rightarrow G$

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow G$

$A \rightarrow B \rightarrow C \rightarrow F \rightarrow G$

$A \rightarrow B \rightarrow D \rightarrow E \rightarrow G$

$A \rightarrow B \rightarrow D \rightarrow F \rightarrow G$

$A \rightarrow B \rightarrow D \rightarrow G$

$A \rightarrow B \rightarrow E \rightarrow G$

$A \rightarrow C \rightarrow D \rightarrow E \rightarrow G$

$A \rightarrow C \rightarrow D \rightarrow F \rightarrow G$

$A \rightarrow C \rightarrow D \rightarrow G$

$A \rightarrow C \rightarrow F \rightarrow G$

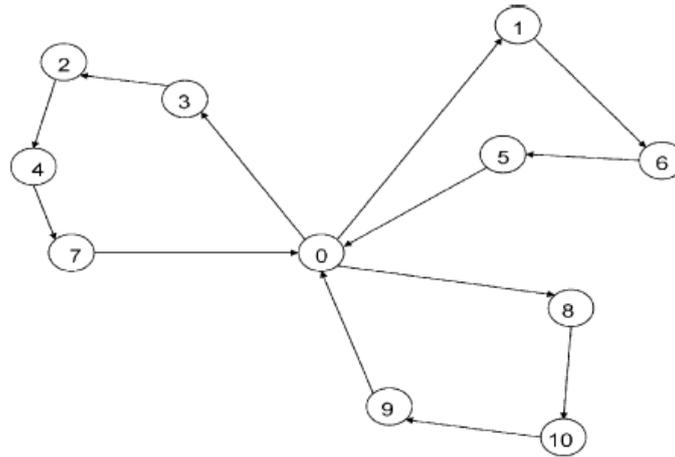
Berdasarkan beberapa alternatif rute perjalanan diatas, dapat dihitung jalur terpendek yaitu dengan mencari jarak antar jalur perjalanan tersebut. Jarak antar jalur perjalanan

dapat dihitung dengan menggunakan koordinat kota-kota tersebut. Pada umumnya terdapat dua metode yang dapat digunakan untuk menyelesaikan permasalahan jalur terpendek, yaitu metode algoritma konvensional dan metode heuristik (Mutakhiroh et al., 2007). Pada metode konvensional, penyelesaian masalah dilakukan dengan menggunakan perhitungan matematika eksak. Contoh metode konvensional yang dapat digunakan untuk menyelesaikan masalah jalur terpendek adalah algoritma Dijkstra, algoritma Floyd-Warshall, dan algoritma Bellman-Ford. Sementara itu, pada metode heuristik, penyelesaian masalah jalur terpendek dilakukan dengan kecerdasan buatan. Beberapa contoh algoritma pada metode heuristik yang dapat digunakan untuk menyelesaikan jalur terpendek adalah *Tabu Search*, Algoritma Genetika, *Ant Colony Optimization*, Algoritma *Sweep*, dan lain-lain. Pencarian jalur terpendek sangat berhubungan dengan sistem transportasi. Pada sistem transportasi, analisis keputusan dilakukan untuk mendapatkan rute terpendek pada suatu jalur transportasi. Analisis tersebut meliputi kombinasi dari rute dan penjadwalan dari suatu armada transportasi. Tujuannya adalah untuk meminimalkan jumlah kendaraan yang dipakai, waktu tempuh, maupun jarak yang ditempuh dalam proses distribusi (Bowersox, 2002).

#### **2.1.4. Vehicle Routing Problem**

*Vehicle Routing Problem* (VRP) adalah suatu masalah dalam menentukan rute optimal perjalanan kendaraan pada saat mendistribusikan barang dari satu depot atau lebih ke beberapa konsumen yang memiliki lokasi berbeda-beda dengan permintaan barang yang telah diketahui dan memenuhi sejumlah kendala (Yeun et al., 2008). Tujuan dari VRP adalah untuk meminimalkan jarak atau biaya melalui kombinasi rute (Bell & McMullen, 2004). VRP pertama kali diperkenalkan oleh Dantzig dan Ramser pada tahun 1959. Pada dunia industri, VRP merupakan masalah penentuan rute yang memegang peran penting yaitu pada bagian manajemen distribusi dan transportasi. VRP merupakan perkembangan atau perluasan dari *Travelling Salesman Problem* (TSP). *Travelling Salesman Problem* (TSP) adalah suatu permasalahan yang bertujuan untuk menentukan jalur perjalanan dari satu lokasi asal, kemudian mengunjungi beberapa lokasi lain dan kembali lagi ke lokasi asal, dimana lokasi-lokasi tersebut hanya dilewati sebanyak satu kali dan total jarak yang ditempuh adalah minimum (Guttin & Punnen, 2002).

Secara sederhana, hasil penyelesaian permasalahan VRP dapat dilihat pada Gambar 2.2 berikut.



Gambar 2.2 Contoh Solusi VRP

Sumber : Bell & McMullen (2004)

Menurut Toth & Vigo (2002), terdapat beberapa karakteristik dalam VRP yang perlu diperhatikan, yaitu mengenai komponen-komponen yang berkaitan pada VRP. Komponen-komponen tersebut adalah:

1. Pelanggan (konsumen)
2. Depot
3. Pengemudi
4. Rute kendaraan.

Tujuan yang ingin dicapai pada *Vehicle Routing Problem* menurut Toth & Vigo (2002) adalah sebagai berikut:

1. Meminimalkan biaya transportasi pada pendistribusian barang yang dipengaruhi oleh total jarak yang ditempuh dan jumlah kendaraan yang digunakan.
2. Meminimalkan jumlah kendaraan yang digunakan untuk mendistribusikan barang ke seluruh pelanggan.
3. Menyeimbangkan rute, untuk waktu perjalanan dan muatan kendaraan.
4. Meminimalkan keluhan pelanggan akibat pelayanan yang kurang maksimal.

VRP adalah suatu masalah optimasi kombinatorial dalam *NP-Hard Problem* karena secara teori ataupun praktik pada dunia nyata memiliki permasalahan yang sangat banyak dan kompleks sehingga sulit untuk dipecahkan. Metode konvensional (*brute force*) dan metode metaheuristik dapat digunakan untuk menyelesaikan permasalahan *NP-Hard* (Pop et al., 2011). Menurut Glover & Kochenberger (2003), pada metode *brute force* semua kemungkinan solusi yang ada akan dibangkitkan, kemudian dari semua solusi tersebut dipilih yang terbaik. Sementara itu, pada metode metaheuristik proses perhitungan akan lebih cepat dibandingkan metode konvensional dan solusi yang diberikan mendekati optimal. Terdapat beberapa jenis VRP menurut Toth & Vigo (2002), yaitu :

1. *Capacitated VRP (CVRP)*

Pada VRP jenis ini, kendaraan yang bertugas untuk mengangkut barang memiliki batasan kapasitas tertentu.

2. *VRP with time windows (VRPTW)*

Pada VRP jenis ini, setiap pelanggan memiliki waktu tertentu untuk dapat dilayani oleh kendaraan.

3. *Multiple Depot VRP (MDVRP)*

Pada VRP jenis ini, terdapat banyak depot yang melayani konsumen.

4. *VRP with pickup and delivery (VRPPD)*

Pada VRP jenis ini, terdapat beberapa barang yang harus dipindahkan dari lokasi penjemputan tertentu ke tempat pengiriman lainnya.

5. *Split Delivery VRP (SDVRP)*

Pelanggan dilayani dengan menggunakan kendaraan yang berbeda-beda.

6. *Periodic VRP*

Pada VRP jenis ini, adanya perencanaan yang berlaku untuk satuan waktu tertentu.

Permasalahan *Vehicle Routing Problem (VRP)* dapat dijelaskan sebagai graf berarah dan berbobot  $G=(V,E)$ . Dimana  $V$  adalah sebuah simpul atau *node set*  $V=\{0, 1, \dots, n\}$ , node 0 adalah untuk sebuah depot yang merupakan tempat kendaraan memulai dan mengakhiri perjalanan, sedangkan node  $i > 0$  adalah pelanggan yang akan dilayani.  $E$  adalah jaringan jalan yang menghubungkan depot dengan pelanggan dan juga penghubung antar pelanggan,  $E = \{(i,j)/i,j \in V, i \neq j\}$ . Model VRP telah dikembangkan oleh Kallehauge et al. (2001) dengan bentuk umum yang memiliki fungsi tujuan untuk

meminimalkan total *cost*, dengan batasan setiap pelanggan hanya dilayani satu kali, setiap rute perjalanan berakhir di depot, batasan *time windows* dan batasan kapasitas. Persamaan yang dibuat memiliki beberapa notasi, diantaranya adalah sebagai berikut:

$V$	= kumpulan kendaraan dengan kapasitas yang sama
$C$	= kumpulan pelanggan
$G$	= grafik berarah
$N$	= kumpulan titik yang terdiri dari depot dan pelanggan
$A$	= kumpulan jalur
$0$	= titik awal (depot)
$n+1$	= titik akhir (depot)
$q$	= kapasitas kendaraan
$d_i$	= <i>demand</i> pelanggan
$c_{ij}$	= biaya
$t_{ij}$	= waktu perjalanan + waktu pelayanan
$a_i, b_i$	= <i>time windows</i>

Pada model Kallehauge et al. (2001) terdapat dua variabel keputusan yaitu  $x$  dan  $s$ . Untuk setiap  $(i, j)$ , dimana  $i \neq j$ ,  $i \neq n+1$ ,  $j \neq 0$ , dan setiap kendaraan  $k$ , variabel keputusan pada permasalahan tersebut dapat dilihat pada Persamaan (2.1) berikut:

$$x_{ijk} = \begin{cases} 1, & \text{jika kendaraan } k \text{ melakukan perjalanan dari } i \text{ ke } j \\ 0, & \text{jika tidak ada kendaraan } k \text{ yang melakukan perjalanan dari } i \text{ ke } j \end{cases} \quad (2.1)$$

Variabel keputusan  $s_{ik}$  menunjukkan waktu dimulai pelayanan pada *customer*  $i$  oleh kendaraan  $k$ . Apabila kendaraan  $k$  tidak melakukan pelayanan pada pelanggan  $i$ , maka  $s_{ik}$  tidak berarti apapun. Persamaan 2.2 berikut model matematis dengan tujuan untuk meminimalkan total *cost* pada perjalanan.

$$\text{Min } Z_{VRP} = \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} \quad (2.2)$$

dengan kendala:

1. Setiap simpul (pelanggan) hanya dikunjungi tepat satu kali oleh satu kendaraan.

$$\sum_{k \in V} \sum_{i \in N} x_{ijk} = 1 \quad \forall i \in C \quad (2.3)$$

2. Total kuantitas permintaan seluruh pelanggan dalam satu rute tidak melebihi kapasitas kendaraan

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijk} \leq q \quad \forall k \in V \quad (2.4)$$

3. Setiap rute perjalanan suatu kendaraan berawal dari depot 0

$$\sum_{j \in N} x_{0jk} = 1 \quad \forall k \in V \quad (2.5)$$

4. Setiap kendaraan yang mengunjungi satu titik, pasti akan meninggalkan titik tersebut

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0 \quad \forall h \in C, \forall k \in V \quad (2.6)$$

5. Setiap kendaraan setelah menyelesaikan rute, kembali ke depot

$$\sum_{i \in N} x_{i,n+1,k} = 1 \quad \forall k \in K \quad (2.7)$$

6. Kendaraan  $k$  yang akan melayani pelanggan tidak diperbolehkan sampai di pelanggan  $j$  sebelum  $s_{ik} + t_{ij}$  (sebelum waktu untuk pelayanan dan waktu perjalanan dari  $i$  ke  $j$ , dimana  $K$  merupakan bilangan riil bernilai besar.

$$s_{ik} + t_{ij} - K(1 - x_{ijk}) \leq s_{jk} \quad \forall i, j \in N, \forall k \in V \quad (2.8)$$

7. Batasan *time window* untuk setiap pelanggan terpenuhi

$$a_i \leq s_{ik} \leq b_i \quad \forall i \in N, \forall k \in V \quad (2.9)$$

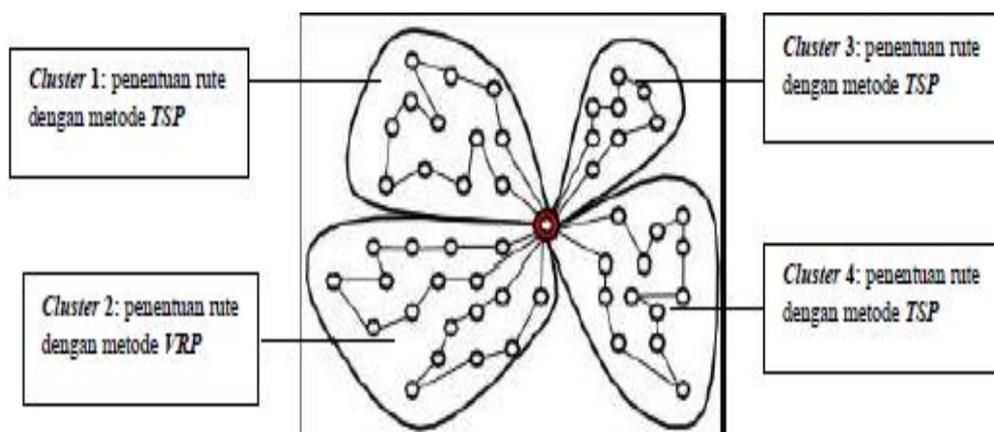
8. Variabel keputusan  $x_{ijk}$  adalah bilangan biner.

$$x_{ijk} \in \{0,1\} \quad \forall i, j \in V, \forall k \in V \quad (2.10)$$

### 2.1.5. Clustering Nearest Neighbor

Algoritma *nearest neighbor* adalah salah satu teknik yang sederhana dan terbuka untuk berbagai macam variasi masalah (Pop et al., 2011). Algoritma ini digunakan pada metode *clustering* untuk mengelompokkan beberapa data atau anggota yang didasarkan pada kesamaan atau kedekatan. Menurut Tan et al. (2006), *clustering* adalah proses untuk mengelompokkan objek yang memiliki kemiripan ke dalam satu himpunan sama yang disebut dengan *cluster*. Pada proses distribusi, pembentukan *cluster* dilakukan untuk membuat suatu batas wilayah antar kelompok. *Cluster* pada proses distribusi beranggotakan konsumen-konsumen pada suatu area yang memiliki karakteristik sama yaitu dilayani oleh *salesman* yang sama dan *salesman* tersebut tidak dapat melayani konsumen yang berada di *cluster* lain (Cao & Glover, 2010). Anggota yang terdapat dalam satu *cluster*, memiliki karakteristik yang sama dimana dalam hal ini adalah yang memiliki jarak berdekatan. Proses *clustering* sangat mungkin dilakukan oleh perusahaan

yang melakukan distribusi karena adanya keterbatasan sumber daya, jumlah *salesman*, jumlah kendaraan, dan panjang waktu produktif untuk melakukan proses pengiriman sehingga harus membagi wilayah pelayanan menjadi beberapa *cluster* (Cao & Glover, 2010). Pembentukan *cluster* pada proses distribusi untuk kasus TSP dan VRP dapat dilihat pada Gambar 2.3 berikut



Gambar 2.3 Clustering pada TSP & VRP

Sumber: Lian dan Castelain (2010)

Pada kasus VRP, metode *nearest neighbor* dapat digunakan untuk mengelompokkan beberapa agen yang memiliki jarak berdekatan yang kemudian akan berada pada satu *cluster* yang sama. Menurut Pujawan (2005), *nearest neighbour* dapat digunakan secara efektif untuk mencari konsumen yang dilayani pada proses distribusi berdasarkan jarak terdekat dari lokasi terakhir kendaraan. Pengelompokan dilakukan dengan mencari pelanggan yang berdekatan dengan pelanggan yang terakhir untuk kemudian ditambahkan pada satu kelompok jalur yang sama tersebut. Rute baru dimulai dengan cara yang sama jika tidak terdapat posisi yang fisibel untuk menempatkan pelanggan baru karena kendala kapasitas atau *time windows* (Braysy & B. Gendreau, 2005).

Menurut Fauzi & Susanty (2015), langkah-langkah pengelompokan dengan metode *nearest neighbor* pada proses distribusi adalah sebagai berikut.

Langkah 1:

Menentukan titik awal pengiriman (depot)

Langkah 2:

Menentukan konsumen yang akan dikunjungi berdasarkan jarak terdekat dari titik terakhir atau dari titik awal (depot)

Langkah 3:

Titik terakhir konsumen yang dikunjungi menjadi titik awal untuk selanjutnya mencari konsumen lain yang akan dikunjungi berdasarkan jarak terdekat dari titik tersebut

Langkah 4:

Lakukan langkah 3 hingga semua konsumen terlayani

Langkah 5:

Kendaraan kembali ke titik awal/ depot jika semua konsumen telah terlayani semua dan tidak melebihi batasan.

### 2.1.6. Algoritma Semut

#### A. Pengertian Algoritma Semut

Algoritma Semut merupakan teknik probabilistik untuk menyelesaikan masalah komputasi dengan menemukan jalur terbaik. Algoritma ini diambil dari analogi perilaku hewan yaitu semut dalam menemukan jalur dari koloninya menuju makanan. *Ant Colony Optimization* (ACO) atau Algoritma Koloni Semut adalah metode yang termasuk dalam algoritma metaheuristik untuk menyelesaikan masalah optimasi yang bersifat kombinasi (Utama, 2010). Algoritma Semut adalah algoritma yang mengadopsi perilaku koloni semut yang dikenal sebagai sistem semut.

Pada *Ant Colony Optimization* (ACO), semut merupakan representasi dari agen. Agen inilah yang akan mencari solusi dengan menelusuri berbagai kemungkinan jalur yang ada dengan menggunakan aturan tertentu pada algoritma ACO. Penerapan *Ant Colony Optimization* (ACO) pada awalnya digunakan untuk menyelesaikan permasalahan *Travelling Salesman Problem* (TSP). Seiring dengan perkembangan penelitian, kasus *Vehicle Routing Problem* juga dapat diselesaikan dengan metode Algoritma Semut (Donati et al., 2008). Saat ini, Algoritma Semut dapat digunakan menyelesaikan permasalahan optimasi yang lain. Menurut Karjono et al. (2016), banyak penyelesaian

masalah optimasi yang dapat diselesaikan dengan Algoritma Semut dan hasilnya mendekati optimal. Beberapa contoh penerapan Algoritma Semut adalah:

1. *Traveling Salesman Problem* (TSP)
2. *Vehicle Routing Problem* (VRP)
3. *Quadratic Assignment Problem* (QAP)
4. *Job-shop Scheduling Problem* (JSP)
5. *Graph coloring*
6. *Network Routing*

## **B. Cara Kerja Algoritma Semut**

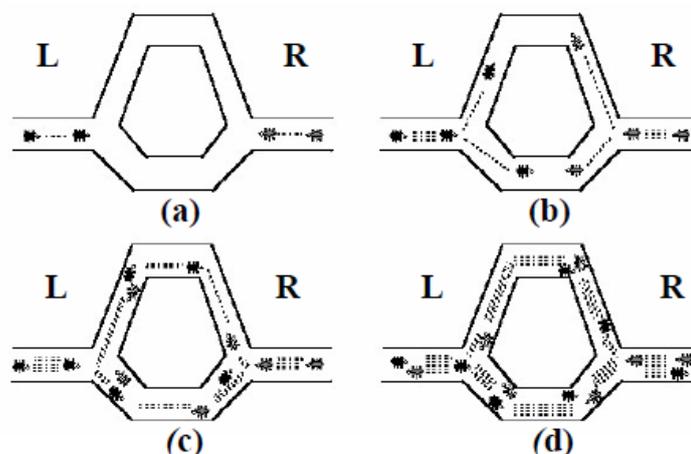
Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan. Hal ini karena pada saat mencari makanan dan kembali ke sarangnya, semut akan meninggalkan zat *Pheromone* pada rute-rute yang dilalui. *Pheromone* adalah zat kimia yang digunakan oleh makhluk hidup untuk mengenali sesama jenis, individu lain, kelompok, dan untuk membantu proses reproduksi. Zat ini berasal dari kelenjar endoktrin. Semut yang telah menemukan makanan, akan kembali ke sarangnya dan meninggalkan *pheromone* pada jalur yang dilewati saat kembali ke sarangnya. Apabila kandungan *pheromone* pada suatu jalur padat, maka semut lain akan mencium *pheromone* tersebut dan mengikuti rute tersebut. Semakin padat kandungan *pheromone* pada suatu rute, akan semakin besar kemungkinan semut lain untuk melewati rute tersebut.

Proses peninggalan *pheromone* pada jalur yang dilewati disebut dengan *stigmergy*, yaitu sebuah proses memodifikasi lingkungan yang tidak hanya bertujuan untuk mengingat jalan pulang ke sarang, tetapi juga memungkinkan para semut berkomunikasi dengan koloninya. Jejak *pheromone* yang ditinggalkan semut semakin lama akan menguap dan kekuatan daya tariknya akan berkurang. Semakin cepat semut kembali ke sarang dan melewati rute keberangkatan, maka jumlah *pheromone* yang menguap hanya sedikit. Sebaliknya, apabila semut membutuhkan waktu lama dalam melakukan perjalanan menuju ke sumber makanan dan kembali ke sarang, maka akan semakin banyak *pheromone* yang menguap.

Melalui jejak kaki yang terdapat pada lintasan yang dilewati, koloni semut akan dapat menentukan jalur terpendek dari sarang menuju sumber makanan. Jejak kaki semut akan terlihat semakin jelas pada lintasan yang banyak dilewati oleh semut. Hal ini menyebabkan lintasan yang dilalui semut dalam jumlah sedikit semakin lama semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati oleh semut sama sekali. Sebaliknya lintasan yang dilalui semut dalam jumlah banyak semakin lama akan semakin bertambah kepadatan semut yang melewatinya atau bahkan semua semut melewati lintasan tersebut (Dorigo & Stutzle, 2004)

Filosofi semut dalam mencari makanan hingga kembali ke sarangnya adalah sebagai berikut (Ahmad & Srivastava, 2008):

1. Terdapat perintah untuk mencari dan menemukan makanan yang dilakukan secara acak.
2. Saat semut telah menemukan sumber makanan, semut tersebut akan kembali ke sarang dengan melewati jalur yang sama pada saat keberangkatan menuju ke sumber makanan.
3. Pada saat melewati suatu jalur, semut akan meninggalkan *pheromone* di jalur tersebut dan seiring berjalannya waktu, *pheromone* dapat menguap.
4. Secara berulang, akan terdapat penambahan jalur apabila kadar *pheromone* pada suatu jalur berkurang.
5. Semut-semut akan memilih untuk melewati jalur yang memiliki kadar *pheromone* kuat.



Gambar 2.4 Perjalanan Semut Menuju Sumber Makanan

Sumber : Dorigo & Gambardella (1997)

Pada Gambar 2.4. (a), diketahui terdapat dua kelompok semut yang melakukan perjalanan, yaitu kelompok semut L dan kelompok semut R. Kelompok semut L berjalan dari sisi kiri ke sisi kanan dan kelompok semut R berjalan dari sisi kanan ke sisi kiri. Kedua kelompok semut berjalan dari titik keberangkatan dan dalam posisi menentukan jalan mana yang akan dipilih untuk dilewati. Kelompok semut L membagi koloninya menjadi dua kelompok, yaitu kelompok yang melewati jalan atas dan kelompok yang melewati jalan bawah. Hal ini juga berlaku pada semut kelompok R. Gambar 2.4.b dan Gambar 2.4.c menunjukkan bahwa kelompok semut berjalan pada kecepatan yang sama dengan meninggalkan *pheromone* atau jejak kaki di jalan yang telah dilalui. Karena jalur atas yang dilalui semut memiliki jarak tempuh yang lebih panjang dibanding jalur bawah, maka *pheromone* yang ditinggalkan pada jalur atas telah mengalami banyak penguapan. Hal ini karena semut yang melewati jalur atas berjumlah lebih sedikit dibandingkan jalan yang di bawah. Sedangkan *pheromon* yang berada pada bagian bawah penguapannya cenderung lebih lama. Hal ini karena semut yang melewati jalur bawah lebih banyak dibandingkan dengan semut yang melewati jalur atas.

Gambar 2.4.d menunjukkan bahwa semut-semut yang lain pada akhirnya memutuskan untuk melewati jalur bawah karena terdapat banyak *pheromone* yang ditinggalkan, sedangkan *pheromon* pada jalur atas telah banyak yang menguap sehingga semut-semut tidak memilih jalur atas. Semakin banyak semut yang melewati suatu jalan, maka semakin banyak semut yang mengikuti jalan tersebut. Sementara itu, semakin sedikit semut yang melewati suatu jalan, maka *feromon* yang ditinggalkan semakin berkurang bahkan hilang. Dari sinilah kemudian terpilihilah jalur terpendek antara sarang dan sumber makanan. Pencarian solusi yang optimal pada Algoritma Semut dilakukan oleh semua semut yang terdapat dalam satu koloni. Semut direpresentasikan sebagai *salesman* yang memiliki tugas untuk menemukan rute terpendek (masalah optimasi). Setelah semua semut dalam satu koloni selesai melakukan kombinasi solusi, maka dilakukan pemilihan semut terbaik yang akan dibandingkan dengan semut terbaik secara global sehingga menghasilkan solusi akhir (Dorigo & Stutzle, 2004).

### C. Tahapan Algoritma Semut

Menurut Mutakhirah et al. (2007) terdapat beberapa langkah untuk menentukan jalur terpendek dengan metode Algoritma Semut, diantaranya adalah sebagai berikut:

Langkah 1:

a. Inisialisasi harga parameter-parameter Algoritma Semut.

Terdapat beberapa parameter masukkan yang ada pada Algoritma Semut sebagai inisialisasi awal untuk melakukan proses optimasi. Beberapa parameter tersebut adalah :

1. Intensitas jejak semut antar kota (titik) dan perubahannya ( $\tau_{ij}$ ). Nilai  $\tau_{ij}$  digunakan pada persamaan probabilitas dalam mencari kota yang akan dikunjungi. Setelah satu siklus perjalanan selesai, nilai  $\Delta\tau_{ij}$  diinisiasi untuk menentukan nilai  $\tau_{ij}$  siklus selanjutnya.
2. Banyak kota (titik) yang ada, termasuk koordinat  $(x,y)$ , dan jarak antar kota ( $d_{ij}$ )
3. Penentuan titik asal dan titik tujuan.
4. Tetapan siklus semut ( $Q$ ). Nilai  $Q$  adalah konstanta yang digunakan dalam mencari nilai  $\Delta\tau_{ij}$ . Nilai ini ditentukan oleh pengguna.
5. Tetapan pengendali intensitas jejak semut ( $\alpha$ ). Nilai  $\alpha$  digunakan pada persamaan probabilitas dalam mencari kota yang akan dikunjungi. Nilai ini berfungsi sebagai pengendali intensitas jejak semut. Nilai  $\alpha \geq 0$ , ditentukan oleh pengguna.
6. Tetapan pengendali visibilitas ( $\beta$ ). Nilai  $\beta$  digunakan pada persamaan probabilitas dalam mencari kota yang akan dikunjungi. Nilai ini berfungsi sebagai pengendali visibilitas. Nilai  $\beta \geq 0$ , ditentukan oleh pengguna.
7. Visibilitas antar titik ( $\eta_{ij}$ ). Nilai  $\eta_{ij}$  digunakan pada persamaan probabilitas dalam mencari kota yang akan dikunjungi. Nilai  $\eta_{ij}$  merupakan hasil dari  $1/d_{ij}$  (jarak antar tempat).
8. Banyak semut ( $m$ ). Nilai  $m$  adalah banyaknya semut yang akan melakukan perjalanan untuk mencari rute dalam Algoritma Semut. Nilai  $m$  ditentukan oleh pengguna.
9. Tetapan penguapan jejak semut ( $\rho$ ). Nilai  $\rho$  digunakan untuk menentukan  $\tau_{ij}$  pada siklus selanjutnya. Nilai  $\rho$  ditentukan oleh pengguna.

10. Jumlah siklus maksimum ( $NCmax$ ). Nilai  $NCmax$  adalah banyaknya siklus maksimal yang akan berlangsung. Nilai  $NCmax$  ditentukan oleh pengguna.

b. Inisialisasi kota pertama setiap semut.

Setelah inisialisasi  $\tau_{ij}$  dilakukan, kemudian  $m$  semut ditempatkan pada kota pertama yang telah ditentukan.

Langkah 2:

Pengisian kota pertama ke dalam *tabu list*. Kota asal diisikan sebagai elemen pertama pada *tabu list* ( $k,1$ ) dimana  $k=1,\dots,m$ . Hasil dari langkah ini adalah terisinya elemen pertama *tabu list* setiap semut dimana setiap tabu  $k$  (1) dapat berisi indeks antara kota 1 hingga kota  $n$  sebagaimana hasil inisialisasi pada langkah 1.

Langkah 3:

Penyusunan jalur kunjungan setiap semut ke setiap kota. Koloni semut yang sudah terdistribusi ke kota pertama akan mulai melakukan perjalanan dari kota pertama sebagai kota asal menuju ke kota-kota lainnya yang merupakan kota tujuan. Kemudian dari kota kedua, masing-masing koloni semut akan melanjutkan perjalanan menuju ke kota selanjutnya dengan cara memilih salah satu dari kota-kota yang tidak terdapat pada *tabu k*. Perjalanan koloni semut berlangsung terus menerus hingga semua kota terkunjungi. Jika  $s$  menyatakan indeks urutan kunjungan, kota asal dinyatakan sebagai *tabuk(s)* dan kota-kota lainnya dinyatakan sebagai  $\{N-tabuk\}$ . Persamaan (2.11) dan Persamaan (2.12) berikut menunjukkan probabilitas yang digunakan untuk menentukan kota selanjutnya yang akan dikunjungi.

$$P_{ij}^k = \frac{[\tau(t,v)]^\alpha \cdot [\eta(t,v)]^\beta}{\sum_{k' \in (N-Tabu_k)} [\tau(t,v)]^\alpha \cdot [\eta(t,v)]^\beta} \quad \text{Untuk } \{N-tabu_k\} \quad (2.11)$$

$$P_{ij}^k = 0 \quad \text{Untuk } j \text{ lainnya} \quad (2.12)$$

Dengan  $i$  adalah indeks kota asal dan  $j$  adalah indeks kota tujuan.

Langkah 4:

a. Perhitungan panjang jalur setiap semut.

Perhitungan panjang jalur tertutup (*length closed tour*) atau  $L_k$  setiap semut dilakukan setelah semua semut menyelesaikan satu siklus perjalanan. Perhitungan panjang rute dilakukan dengan berdasarkan  $tabu_k$  masing-masing sesuai yang ditunjukkan Persamaan (2.13) berikut.

$$L_k = d_{tabu_k(n), tabu_k(1)} + \sum_{s=1}^{n-1} d_{tabu_k(s-1)} \quad (2.13)$$

dengan  $d_{ij}$  adalah jarak antara titik  $i$  ke  $j$  yang dihitung berdasarkan Persamaan (2.14)

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2.14)$$

b. Pencarian rute terpendek.

Setelah menghitung  $L_k$  setiap semut, selanjutnya diperoleh harga minimal panjang jalur tertutup setiap siklus atau  $L_{minNC}$  dan harga minimal panjang jalur tertutup secara keseluruhan adalah atau  $L_{min}$ .

c. Perhitungan perubahan harga intensitas jejak kaki semut antar kota.

Jejak-jejak kaki semut akan tertinggal pada lintasan antar kota yang dilaluinya. Adanya penguapan dan perbedaan jumlah semut yang lewat, menyebabkan kemungkinan terjadinya perubahan harga intensitas jejak kaki semut antar kota. Persamaan (2.15) menunjukkan perubahan harga intensitas jejak kaki semut antar kota:

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2.15)$$

Dengan  $\Delta\tau_{ij}^k$  adalah perubahan harga intensitas jejak kaki semut antar titik setiap semut yang dihitung sesuai dengan Persamaan (2.16) dan Persamaan (2.17) berikut:

$$\Delta\tau_{ij}^k = \frac{Q}{L_k}, \text{ untuk } (i,j) \in \text{kota asal dan titik tujuan dalam } tabu_k \quad (2.16)$$

$$\Delta\tau_{ij}^k = 0, \text{ untuk } (i,j) \text{ lainnya} \quad (2.17)$$

Langkah 5:

a. Menghitung harga intensitas jejak kaki semut antar titik untuk siklus selanjutnya.

Adanya penguapan dan perbedaan jumlah semut yang melewati lintasan menyebabkan berubahnya harga intensitas jejak kaki semut yang terdapat di antar titik. Sehingga pada siklus selanjutnya, nilai intensitas jejak kaki semut telah berubah. Persamaan untuk menghitung harga intensitas jejak kaki semut antar kota untuk siklus selanjutnya dapat dilihat pada Persamaan (2.18) berikut:

$$\tau_{ij} = (1-\rho) \cdot \tau_{ij} + \Delta\tau_{ij} \quad (2.18)$$

b. Atur ulang harga perubahan intensitas jejak kaki semut antar titik.

Untuk siklus selanjutnya perubahan harga intensitas jejak semut antar kota perlu diatur kembali agar memiliki nilai sama dengan nol.

Langkah 6:

Pengosongan *tabu list*, dan ulangi langkah dua jika diperlukan. Apabila jumlah siklus maksimum belum tercapai (belum terjadi konvergensi) maka *tabu list* perlu dikosongkan untuk diisi lagi dengan urutan kota yang baru pada siklus selanjutnya. Algoritma diulang lagi dari langkah dua dengan harga parameter intensitas jejak kaki semut antar kota yang sudah diperbaharui.

## 2.2 Kajian Induktif

Pada sub bab ini berisi penelitian-penelitian terdahulu yang berhubungan dengan penelitian ini, yaitu mengenai penentuan jalur optimal pada rute distribusi. Adapun penelitian-penelitian terdahulu antara lain sebagai berikut :

Liu et al. (2016) mengembangkan Algoritma Semut pada penentuan jalur evakuasi. Optimasi jalur evakuasi sangat penting untuk mengurangi bahaya yang beresiko bagi manusia dan sosial serta untuk menghemat waktu bantuan. Penentuan jalur dilakukan dengan menggunakan Algoritma Semut yang dikombinasikan dengan konsep dasar *quantum-inspired evolutionary algorithm* (QEA), yang kemudian disebut dengan *quantum ant colony algorithm* (QACA).

Algoritma *Ant Colony System* (ACS) diterapkan oleh Gomez et al. (2014) untuk menyelesaikan permasalahan *vehicle routing problem with time windows* (VRPTW) pada sistem logistik dalam pendistribusian produk botol di Mexico. Metode ACS dikombinasikan dengan metode *Extended Constrained List* (ECL) yang memberikan prioritas pada *customer* dengan *time window* pendek. Pada kasus tersebut, terdapat dua aspek yang dipertimbangkan yaitu topologi dan *time windows* masing-masing *customer*. Proses pencarian jalur diawali dengan melakukan *clustering* yang didasarkan pada kondisi topologi. Hasilnya didapatkan bahwa Algoritma Semut dapat mereduksi penggunaan kendaraan sebagai sarana transportasi hingga 36,86%.

Kheiri (2016) menggunakan Algoritma Semut untuk menganalisis jalur untuk perjalanan kaki terutama saat kondisi ramai di suatu taman yang berlokasi di Teheran, Iran. Tujuannya adalah untuk meminimalkan jarak perjalanan, mengurangi *material handling*, dan menghindari obstruksi bahan dan gerakan tanaman. Hasilnya didapatkan bahwa jalur masuk pengunjung taman dapat dimulai dari bagian utara taman. Hal ini karena dengan mendesain pintu masuk pada bagian utara, pengunjung akan dapat melalui taman dengan total jarak yang lebih minimal.

Kaabachi et al. (2017) menggunakan algoritma semut untuk menyelesaikan permasalahan *Green Travelling Salesman Problem with Time Windows (GTSPTW)*. Algoritma Semut dikembangkan dengan metode metaheuristik lain untuk meminimalkan konsumsi bahan bakar dan meminimalkan emisi dari perjalanan *salesman* sebuah perusahaan peralatan listrik yang terdapat di Tunisia. Kendaraan melakukan perjalanan dari perusahaan menuju ke 7 kota yang terdapat di selatan Tunisia. Hasilnya, Algoritma Semut dapat meminimalkan total jarak dan total bahan bakar yang digunakan.

Abdulkader et al. (2015) mengusulkan algoritma hibridasi yang menggabungkan pencarian lokal dengan *Ant Colony System (ACS)* untuk memecahkan masalah *Multi Compartment Vehicle Routing Problem (MCRVP)*. MCRVP adalah pengembangan dari VRP dimana proses distribusi dilakukan terhadap produk-produk yang berbeda untuk kemudian diantarkan kepada pelanggan. Produk yang berbeda diangkut bersama dalam satu kendaraan dengan banyak kompartemen. Produk disimpan dalam kompartemen yang berbeda karena setiap produk membutuhkan perlakuan yang berbeda. Algoritma koloni semut yang ada dan algoritma koloni semut hibrida yang diusulkan dibandingkan. Hasilnya, Algoritma Semut yang dikembangkan saat diterapkan untuk memecahkan kasus MCRVP memiliki total jarak yang lebih pendek dan waktu komputasi yang lebih singkat.

Suwansuksamran & Ongkunaruk (2013) menyelesaikan permasalahan VRPTW pada pendistribusian produk bubuk perasa dengan menggunakan metode *Mixed Integer Programming (MIP)*. Penelitian diawali dengan melakukan pengelompokan (*clustering*) dengan membentuk 6 kelompok terhadap 690 *customer* yang dimiliki perusahaan

*seasoning* di Thailand. Kemudian penentuan rute dilakukan dengan metode MIP dan menggunakan *software* IB, ILOG CPLEX. Hasil penelitian menunjukkan adanya penurunan biaya transportasi bulanan sebesar sebesar 23% atau 9,413 bath per minggu dan penurunan overtime yang disebabkan penjadwalan pengiriman yang tidak sesuai sebesar 50% atau 2 jam per hari.

Fauzi & Susanty (2015) menyelesaikan permasalahan *Vehicle Routing Problem* (VRP) pada distribusi tabung gas di Bandung Raya. Permasalahan penentuan rute pada penelitian ini diselesaikan menggunakan *Nearest Neighbour* sebagai pembentukan rute awal yaitu untuk mencari kelompo-kelompok konsumen yang dilayani berdasarkan jarak terdekat dari titik terakhir kendaraan untuk selanjutnya di distribusikan. Rute yang dihasilkan kemudian diperbaiki dengan menggunakan metode (1-0) *Insertion Intra Route*. Perhitungan dengan menggunakan (1-0) *Insertion Intra Route* menghasilkan rute yang lebih pendek dibandingkan *Nearest Neighbour*. Rute distribusi yang didapatkan memiliki jarak yang lebih pendek dan terdapat penghematan waktu dibandingkan rute awal, sehingga proses distribusi berlangsung lebih cepat.

Penelitian Dahni & Rahmiati (2017) bertujuan untuk menentukan jalur terpendek pada proses pengantaran surat di Kota Pekanbaru dengan menggunakan Algoritma Semut. Peneliti memilih Algoritma Semut sebagai metode yang digunakan untuk menyelesaikan permasalahan karena dapat mempermudah petugas dalam proses pencarian lokasi kantor dan menentukan rute terpendek pengantar surat di Pemerintah Kota Pekanbaru. Adapun kesimpulan dari hasil pengujian diketahui bahwa algoritma semut dapat menemukan jalur terpendek dalam proses pengantaran surat oleh kurir.

Karimah et al. (2017) menggunakan Algoritma Genetika (GA) untuk menentukan rute distribusi yang optimal pada produk air minum UD Tosa Malang. Pada penelitian ini didapatkan urutan rute baru dari jalur distribusi yang mana lebih baik dari sebelumnya. Dengan 180 populasi, banyak generasi 400 serta kombinasi *crossover rate* 0.6 dan *mutation rate* 0.4 didapatkan bahwa rute distribusi yang baru memiliki rute distribusi lebih pendek dengan selisih jarak sebesar 89.3km.

Rahayu & Yuliana (2017) menggunakan metode *Saving Matrix* untuk merencanakan jadwal dan menentukan rute distribusi pada pendistribusian produk otomotif di gudang PT Indoprima Gemilang Pulo Gadung. Tujuan dari penelitian ini adalah untuk membuat jadwal pengiriman produk lebih efisien dan pengalokasian produk ke setiap armada yang tersedia secara tepat serta pemilihan rute yang memiliki waktu lebih singkat. Input data yang digunakan pada penelitian ini adalah waktu tempuh perjalanan. Hal ini karena adanya kondisi lalu lintas yang menyebabkan jarak tidak linier dengan waktu. Hasilnya didapatkan bahwa rute perjalanan yang didapatkan memiliki total waktu tempuh yang lebih singkat dan adanya penghematan dalam penggunaan armada untuk proses distribusi.

Penelitian Amalia (2015) menerapkan metode *Ant Colony System (ACS)* untuk mencari jalur terpendek pada 9 obyek wisata yang akan dilalui oleh para wisatawan. Dalam penelitian ini didapatkan jalur optimal terbaik yang dapat ditempuh wisatawan. Rute yang diperoleh memiliki jarak tempuh sepanjang 27482 meter dengan rute yang dihasilkan adalah dimulai dari Kebun Raya Bogor, Museum Zoologi, Istana Bogor, Museum Etnobotani, Plaza kapten Muslihat, Museum Perjuangan, Museum Pembela Tanah Air, Situ Gede, Prasasti Batu Tulis dan kembali lagi ke kebun Raya Bogor.

Hadiatma & Purbo (2017) mengimplementasikan metode *Ant Colony* untuk menyelesaikan masalah *Vehicle Routing Problem (VRP)* dengan memanfaatkan *Google Maps* untuk mencari jarak antar agen. Permasalahan yang diselesaikan pada penelitian ini adalah penentuan jalur yang optimal pada pendistribusian produk oleh PT CircleK wilayah Yogyakarta. Hasilnya didapatkan rute distribusi yang optimal dengan proses distribusi dimulai dari 1 gudang menuju ke 34 toko yang terdapat di daerah Yogyakarta.

Penggunaan Algoritma Semut dalam menentukan rute optimal juga diterapkan oleh Lestari & Sari (2013) untuk menentukan rute pengangkutan sampah di Kota Yogyakarta. Rute pengangkutan didasarkan pada kapasitas Tempat Pembuangan Sampah (TPS) dan jarak antar TPS. Pada penelitian ini, pencari rute distribusi dilakukan pada sebanyak 41 TPS yang tersebar di Kota Yogyakarta. Hasil penerapan Algoritma Semut menghasilkan rute distribusi pengangkutan sampah yang lebih pendek dan biaya pengangkutan yang lebih minimal

Tabel 2.1 Posisi Penelitian

Tahun	Penulis	Fokus Penelitian			Metode				Input Data		
		Distribusi Produk	Rute Perjalanan	Lainnya	GA	Algoritma Semut	Algoritma Semut Kombinasi	Lainnya	Jarak tempuh	Waktu tempuh	Lainnya
2016	Liu et al		v				v		v		
2014	Gomez et al	v					v		v		v
2016	Kheiri		v			v					v
2017	Kaabachi et al	v				v			v		
2015	Abdulkader et al			v			v		v		
2013	Suwansuk-samran & Ongkunaruk	v						v	v		v
2015	Fauzi & Susanty	v					v	v			
2017	Dahni & Rahmiani		v			v			v		
2017	Karimah et al	v			v				v		
2017	Rahayu & Yuliana	v						v		v	
2015	Amalia		v			v			v		
2017	Hadiatma & Purbo	v				v			v		
2013	Lestari & Sari		v			v			v		
2018	Penelitian yang diusulkan	v				v				v	

Berdasarkan beberapa penelitian terdahulu dapat diketahui bahwa terdapat berbagai metode yang dapat digunakan untuk melakukan optimasi yang berkaitan dengan jalur distribusi, salah satunya adalah Algoritma Semut. Pada penelitian ini, optimasi pada jalur distribusi PT Ultrajaya Milk Industry Tbk distributor cabang Yogyakarta dilakukan dengan menggunakan metode Algoritma Semut dan input data yang digunakan adalah waktu tempuh antar agen dan distributor. Waktu tempuh digunakan sebagai input data karena menurut penelitian sebelumnya, jarak tempuh tidak linear dengan waktu. Hal ini disebabkan adanya kondisi lalu lintas yaitu tingkat kemacetan pada jalur tertentu sehingga penentuan rute distribusi yang optimal dapat lebih relevan apabila input data yang digunakan adalah waktu tempuh.

Berdasarkan penelitian sebelumnya, apabila terdapat lebih dari 1 rute, metode *Clustering* menggunakan *Nearest Neighbor* dapat digunakan untuk menentukan anggota pada setiap rute. Anggota-anggota yang terdapat dalam setiap tur terdiri dari beberapa agen yang berada di suatu wilayah yang berdekatan. Sehingga, penentuan rute distribusi yang optimal pada penelitian ini dilakukan dengan metode Algoritma Semut menggunakan input data waktu tempuh, dimana sebelumnya dilakukan *clustering* menggunakan *Nearest Neighbor* untuk menentukan anggota pada setiap rute. Waktu tempuh digunakan sebagai input data dalam menentukan rute distribusi pada penelitian ini karena dapat mempertimbangkan kondisi kepadatan lalu lintas Yogyakarta dan penentuan rute dapat lebih sesuai dengan tujuan perusahaan yaitu untuk mengurangi waktu distribusi *salesman*. Rute distribusi hasil penelitian ini dapat dijadikan perbandingan dan sebagai masukan untuk perbaikan dari rute distribusi yang telah diterapkan perusahaan saat ini.