

## **BAB II**

### **KAJIAN TEORI**

Pada bab ini berisi paparan teori yang berhubungan dengan distribusi, optimisasi, graf, *vehicle routing problem* (VRP), *capacitated vehicle routing problem with time windows* (CVRPTW), metode penyelesaian CVRPTW, yakni algoritma *floyd warshall*, dan *nearest neighbour*.

#### **A. Distribusi**

Menurut Chopra dan Meindl (2010 : 86), distribusi adalah suatu kegiatan untuk memindahkan barang dari pihak *supplier* kepada pihak agen dalam suatu *supply chain*. Distribusi merupakan suatu kunci dari keuntungan yang akan diperoleh perusahaan karena distribusi secara langsung akan mempengaruhi biaya dari *supply chain* dan kebutuhan agen. Jaringan distribusi yang tepat dapat digunakan untuk mencapai berbagai macam tujuan dari *supply chain*, mulai dari biaya yang rendah sampai respon yang tinggi terhadap permintaan agen.

Salah satu fungsi distribusi adalah pengangkutan (transportasi). Pada umumnya tempat kegiatan produksi berbeda dengan tempat konsumen, maka untuk mengatasi hal ini diperlukan adanya kegiatan pengangkutan (transportasi). Seiring dengan bertambahnya jumlah penduduk, kebutuhan manusia juga semakin banyak. Hal ini mengakibatkan barang yang disalurkan juga semakin besar maka diperlukan alat untuk melakukan suatu kegiatan transportasi.

Distribusi meliputi semua aspek dalam pengiriman barang kepada agen. Beberapa permasalahan yang sering dihadapi dalam distribusi berkaitan dengan optimasi jaringan distribusi antara lain (Harry dan Syamsudin, 2011) :

## 1. Titik Depot

Dalam manajemen distribusi, selalu ada tempat untuk menyimpan dan menyediakan barang hasil produksi, tempat ini dinamakan depot. Dalam hal ini lokasi depot biasa disebut dengan titik depot. Keberadaan titik depot sangat menentukan kelancaran pada kegiatan distribusi barang, sehingga barang dapat sampai pada agen tepat pada waktunya.

## 2. Penentuan rute dan jadwal pengiriman

Salah satu hal yang menjadi keputusan terpenting dalam manajemen distribusi adalah penentuan jadwal serta rute pengiriman dari satu titik ke beberapa titik tujuan. Keputusan seperti ini sangat penting bagi suatu perusahaan yang mengirimkan barangnya dari satu titik ke berbagai titik yang tersebar di sebuah kota, karena sangat berpengaruh terhadap biaya pengiriman. Namun biaya bukanlah satu-satunya faktor yang perlu dipertimbangkan dalam proses distribusi. Selain itu dalam menentukan jadwal dan rute sering kali juga harus mempertimbangkan faktor-faktor lain seperti kapasitas kendaraan dan batas waktu pengiriman.

Secara umum, permasalahan penjadwalan dan penentuan rute pengiriman memiliki beberapa tujuan yang ingin dicapai seperti tujuan untuk meminimumkan biaya pengiriman, meminimumkan waktu atau meminimumkan jarak tempuh. Salah satu dari tujuan tersebut dapat menjadi fungsi tujuan (*objective function*) dan yang lainnya menjadi kendala (*constraint*). Misal fungsi tujuannya adalah meminimumkan biaya pendistribusian dan kendalanya adalah kapasitas kendaraan dan *time windows*.

## B. Optimisasi

Optimisasi merupakan suatu upaya sistematis untuk memilih elemen terbaik dari suatu kumpulan elemen yang ada. Didalam kontek matematika, optimisasi ini dapat dinyatakan sebagai suatu usaha sistematis untuk mencari nilai minimum atau maksimum dari suatu fungsi. Dengan kata lain, optimisasi merupakan suatu proses mencari nilai terbaik berdasarkan fungsi tujuan dengan daerah asal yang telah didefinisikan. Secara sederhana, optimisasi dapat dinyatakan dengan :

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

Sebagai contoh, kita ambil sebuah fungsi  $f(x) = x^2$ , dimana  $x$  adalah anggota bilangan real. Di dalam contoh ini,  $\min f(x)$  merupakan fungsi tujuannya karena yang dicari adalah hasil minimal dari fungsi tersebut, sedangkan  $x$  adalah daerah asal yang didefinisikan sebagai anggota bilangan real.

Optimisasi digunakan hampir di semua bidang ilmu antara lain bidang teknik, sains, ilmu sosial, ekonomi dan bisnis. Banyak permasalahan di bidang teknik, sains dan ekonomi yang dapat dinyatakan sebagai permasalahan optimisasi seperti meminimalkan biaya, mempersingkat waktu, memperkecil resiko maupun memaksimalkan keuntungan dan kualitas. Optimisasi juga seringkali menjadi fokus utama dalam pengambilan keputusan suatu tindakan. Pengambilan keputusan dalam optimisasi terdiri dari beberapa langkah (Talbi, 2009) :

### 1. Merumuskan Masalah

Pertama kali yang perlu dilakukan pada tahap ini adalah mengidentifikasi permasalahan yang akan diselesaikan. Faktor-faktor yang mempengaruhi permasalahan dan fungsi tujuan dari permasalahan tersebut perlu didata dengan cermat.

2. Memodelkan Masalah

Didalam tahap ini, permasalahanakan dimodelkan secara matematis. Permasalahan yang sesungguhnya mungkin akan sangat kompleks dan sulit jika harus diilustrasikan, sehingga perlu disederhanakan dalam pemodelan matematis.

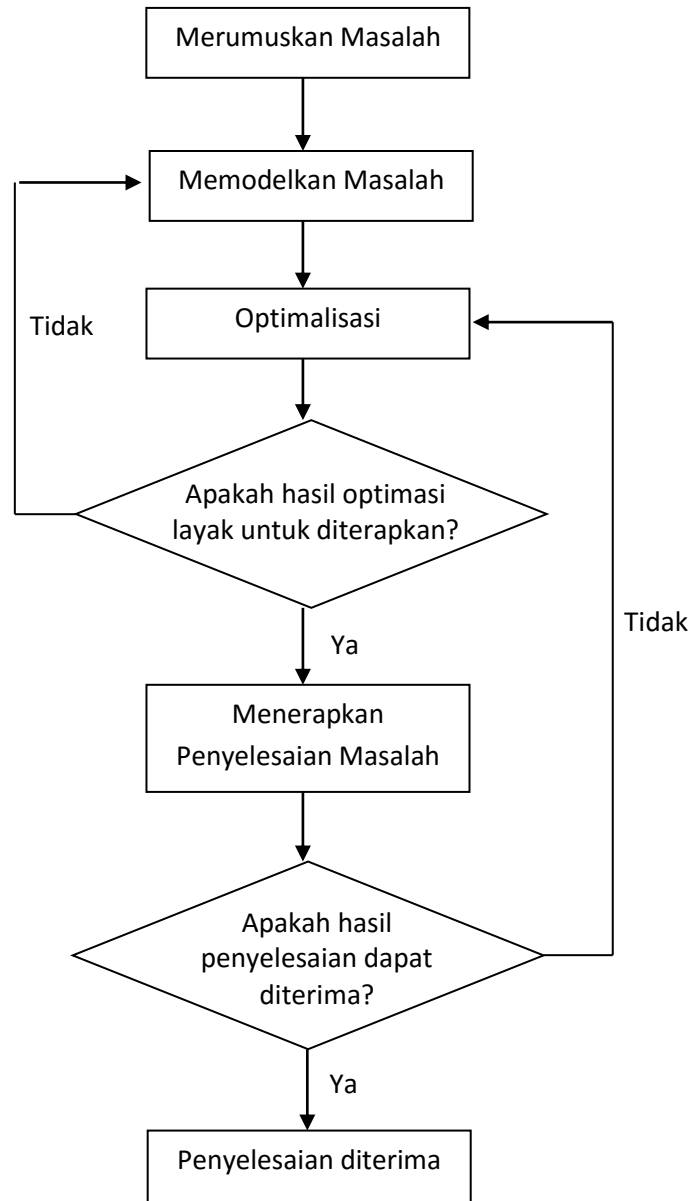
3. Optimisasi

Setelah permasalahan dimodelkan secara matematis, metode optimisasi diterapkan untuk mendapatkan penyelesaian yang dapat diterima (*acceptable solution*). Seringkali model permasalahan yang dibuat tidak dapat mewakili permasalahan yang sesungguhnya, sehingga ketepatan dalam memodelkan permasalahan sangat mempengaruhi hasil yang didapatkan. Penyelesaian yang didapatkan dapat merupakan solusi optimal maupun non-optimal.

4. Menerapkan Penyelesaian Masalah

Penyelesaian yang didapatkan kemudian diterapkan untuk mengetahui apakah penyelesaian tersebut dapat diterima atau tidak. Jika penyelesaian tersebut tidak dapat diterima, maka model permasalahan maupun metode optimisasinya perlu diperbaiki dan prosesnya diulang kembali.

Berikut penyajian langkah-langkah pengambilan keputusan dalam bentuk *flowchart* dapat dilihat pada Gambar 2.1.



**Gambar 2.1.** *Flowchart* Proses Pengambilan Keputusan

## C. Efektivitas

Efektivitas berasal dari kata efektif yang memiliki arti berhasil, tepat guna, atau sesuatu yang dilakukan berhasil dengan baik. Pius A. Partanto dan M. Dahlan Bahri dalam bukunya Kamus Ilmiah Populer (1994) mendefinisikan efektivitas sebagai ketepatangunaan, hasil guna, dan menunjang tujuan. Hidayat (1986) mendefinisikan efektivitas sebagai suatu ukuran yang menyatakan seberapa jauh target telah tercapai, dimana semakin besar persentase target yang dicapai, maka semakin tinggi efektivitasnya.

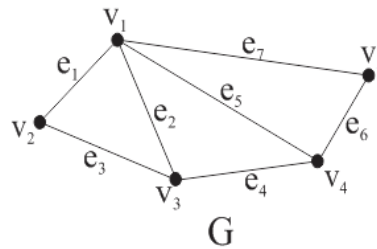
Secara umum, efektivitas dapat diartikan sebagai hal yang berhubungan dengan keberhasilan atau suatu ukuran yang menyatakan sejauh mana tujuan yang dicapai sesuai dengan tujuan yang telah ditentukan sebelumnya. Dalam hal ini, akan dibandingkan efektivitas algoritma *floyd warshall* dan *nearest neighbour* dalam menentukan rute tercepat.

## D. Graf

### 1. Definisi Graf

Definisi graf menurut Edgar G. Goodaire dan Michael M. Parmanter (2002) adalah kumpulan simpul (vertex atau nodes) yang dihubungkan satu sama lain melalui busur (edges). Secara matematis, suatu graf  $G$  didefinisikan sebagai pasangan himpunan  $G(V,E)$  dengan  $V(G)$  adalah himpunan tidak kosong dari simpul (vertex atau nodes),  $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$ , dan  $E(G)$  adalah himpunan busur (edges atau arcs)  $E(G) = \{e_1, e_2, e_3, \dots, e_n\}$ , yang menghubungkan sepasang simpul pada graf tersebut.

Sebuah graf direpresentasikan dalam bentuk gambar. Simpul pada graf digambarkan dengan lingkaran kecil, sedangkan busur yang menghubungkan dua simpul digambarkan dengan kurva sederhana atau segmen garis dengan titik akhir di kedua simpul tersebut.



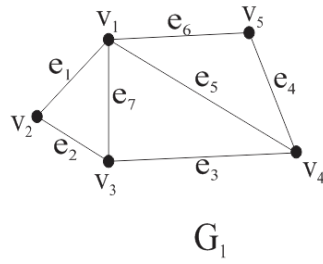
**Gambar 2.2.** Graf G

## 2. Jenis-jenis Graf

Menurut Mardiyono (1996:32) sesuai dengan kekhasan strukturnya, graf dapat diklasifikasikan dalam beberapa jenis yaitu graf sederhana, tidak sederhana, berarah, teratur, berbobot, pohon. Namun dalam hal ini hanya akan dijelaskan beberapa graf yang relevan saja dengan penelitian ini.

### a. Graf sederhana,

Graf sederhana adalah graf yang tidak memiliki rusuk ganda dan gelang. Rusuk ganda adalah dua rusuk yang menghubungkan dua simpul yang sama, sedangkan gelang adalah suatu simpul yang menghubungkan suatu simpul dengan simpul itu sendiri (*loop*). Contoh graf sederhana dapat dilihat pada Gambar 2.3.

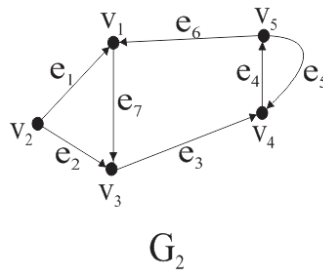


**Gambar 2.3.** Graf  $G_1$

b. Graf berarah,

Graf berarah adalah graf yang setiap rusuknya memiliki orientasi arah. Arah rusuk yang menghubungkan dua buah simpul dilambangkan dengan anak panah.

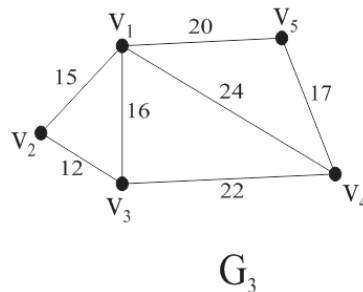
Contoh graf berarah dapat dilihat pada Gambar 2.4.



**Gambar 2.4.** Graf  $G_2$

c. Graf berbobot,

Graf yang setiap rusuknya diberi sebuah harga/bobot. Contoh graf berbobot dapat dilihat pada Gambar 2.5.

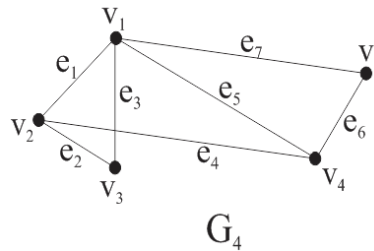


**Gambar 2.5.** Graf  $G_3$



### 3. Keterhubungan

Pada bagian ini akan dijelaskan masing-masing keterhubungan graf yaitu, pengertian jalan, lintasan, jalur/jejak, siklus dan sirkuit yang disertai dengan contoh gambar dan kalimat untuk memperjelas pengertian-pengertian tersebut.



**Gambar 2.6.** Graf  $G_4$

a. Jalan (*walk*)

Sebuah perjalanan dengan panjang  $k$  pada sebuah graf  $G$  adalah rangkaian terurut dari  $k$  rusuk pada graf  $G$  dengan bentuk :

$$uv, vw, wx, \dots yz$$

*Walk* tersebut dinyatakan dengan  $uvw\cdots yz$  atau dengan kata lain *walk* antara  $u$  sampai  $z$ . (Robin J. Wilson & John J. Watkin, 1990 : 34).

Contoh 1:

$W_1 = v_2, v_3, v_1, v_2, v_4, v_1, v_2, v_4, v_5, v_1, v_2$  berarti  $W_1$  merupakan sebuah jalan di  $G_4$ .

b. Jejak (*trail*)

Jika seluruh rusuk (tidak harus seluruh simpul) pada sebuah trayek berbeda, maka trayek tersebut disebut *trail* (jejak). (Robin J. Wilson & John J. Watkin, 1990 : 35)

Contoh 2:

$W_2 = v_2, v_3, v_1, v_4, v_5, v_1$  berarti  $W_2$  merupakan sebuah jejak di  $G_4$ .

c. Lintasan (*path*)

Jika seluruh rusuk dan simpul pada sebuah trayek berbeda, maka trayek tersebut disebut lintasan. (Robin J. Wilson & John J. Watkin, 1990 : 35).

Contoh 3:

$W_3 = v_1, v_3, v_2, v_4, v_5$  berarti  $W_3$  merupakan sebuah lintasan di  $G_4$ .

d. Sirkuit (*circuit*)

Jika seluruh rusuk (tidak harus seluruh simpul) pada sebuah trayek berbeda, serta titik awal dan titik akhirnya berhimpit, maka trayek tersebut disebut *circuit* (sirkuit).

Contoh 4:

$W_4 = v_2, v_3, v_1, v_4, v_5, v_1, v_2$  adalah merupakan sebuah sirkuit di  $G_4$ .

e. Sikel (*cycle*)

Jika seluruh rusuk dan simpul pada sebuah trayek berbeda, serta titik awal dan titik akhirnya berhimpit, maka trayek tersebut disebut *cycle* (sikel).

Contoh 5:

$W_5 = v_1, v_3, v_2, v_4, v_5, v_1$  merupakan sebuah sikel di  $G_4$ .

Suatu graf  $G$  dikatakan terhubung jika untuk setiap pasang dua simpulnya terhubung. Simpul  $u$  dan  $v$  disebut terhubung jika terdapat lintasan dari  $u$  ke  $v$ . (Nizhizeki dan Rahman, 2004: 22)

#### 4. Representasi Graf dalam Matriks

Permasalahan optimasi distribusi dapat direpresentasikan dalam sebuah graf. Simpul merepresentasikan depot/konsumen, dan rusuk merepresentasikan jalur depot ke konsumen dan jalur antar konsumen. Keterhubungan antar simpul pada graf akan disajikan dalam sebuah matriks agar dapat lebih mudah pada penyelesaiannya. Matriks tersebut dinamakan matriks ketetanggaan. Matriks ketetanggaan digunakan dalam merepresentasikan suatu graf berdasarkan banyak rusuk yang menghubungkan simpul-simpulnya. Banyaknya baris dan kolom matriks sama dengan banyak simpul yang ada dalam graf.

Jong Jek Siang (2006 : 273) mendefinisikan matriks ketetanggaan sebagai berikut. Misalkan  $G$  adalah graf tak berarah dengan simpul-simpul  $v_1, v_2, v_3, \dots, v_n$  dengan  $n$  berhingga. Matriks ketetanggaan yang sesuai dengan graf  $G$  adalah matriks  $A = [a_{ij}]$  dengan  $a_{ij}$  merupakan banyak rusuk yang menghubungkan simpul  $v_i$  dengan simpul  $v_j$  pada graf tak berarah akan selalu sama dengan banyak rusuk yang menghubungkan simpul  $v_j$  dengan simpul  $v_i$ , maka matriks ketetanggaannya merupakan matriks ketetanggaan yang simetris  $[a_{ij}] = [a_{ji}]$ , untuk semua  $i, j \in V$ , dimana  $V$  adalah himpunan simpul.

Graf  $G_4$  pada Gambar 2.7 jika dinyatakan dalam matriks ketetanggaan adalah sebagai berikut :

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

**Gambar 2.7.** Matriks Ketetanggaan

Pada matriks ketetanggaan dapat dilihat simpul yang saling berhubungan maupun tidak. Jika  $a_{ij} = 0$  maka simpul  $v_i$  tidak terhubung dengan simpul  $v_j$ , sedangkan jika  $a_{ij} = 1$  maka simpul  $v_i$  terhubung dengan simpul  $v_j$ .

## **E. *Vehicle Routing Problem***

### **1. *Pengertian Vehicle Routing Problem***

Menurut Rahmi dan Murti (2013): *Vehicle Routing Problem* (VRP) merupakan permasalahan dalam sistem distribusi yang bertujuan untuk membuat suatu rute yang optimal, dengan sekelompok kendaraan yang sudah diketahui kapasitasnya, agar dapat memenuhi permintaan konsumen dengan lokasi dan

jumlah permintaan yang telah diketahui. Suatu rute yang optimal adalah rute yang memenuhi berbagai kendala operasional, yaitu memiliki total jarak dan waktu perjalanan yang ditempuh terpendek dalam memenuhi permintaan konsumen serta menggunakan kendaraan dalam jumlah yang terbatas. Berikut ini adalah beberapa kendala atau batasan yang harus dipenuhi dalam VRP yaitu:

1. Rute kendaraan dimulai dari depot dan berakhir di depot,
2. Masing-masing konsumen harus dikunjungi sekali dengan satu kendaraan,
3. Kendaraan yang digunakan adalah homogen dengan kapasitas tertentu, sehingga permintaan konsumen pada setiap rute yang dilalui tidak boleh melebihi kapasitas kendaraan.
4. Jika kapasitas kendaraan sudah mencapai batas, maka konsumen berikutnya akan dilayani oleh shift berikutnya.

Tujuan umum VRP menurut Toth dan Vigo (2002) adalah

1. Meminimalkan jarak dan biaya tetap yang berhubungan dengan penggunaan kendaraan,
2. Meminimalkan banyaknya kendaraan yang dibutuhkan untuk melayani permintaan seluruh konsumen,
3. Menyeimbangkan rute-rute dalam hal waktu perjalanan dan muatan kendaraan, dan
4. Meminimalkan pinalti sebagai akibat dari pelayanan yang kurang memuaskan terhadap konsumen, seperti keterlambatan pengiriman dan lain sebagainya.

## 2. Klasifikasi Jenis-jenis VRP

Terdapat beberapa jenis VRP yang tergantung pada jumlah faktor pembatas dan tujuan yang akan dicapai. Pembatas yang paling umum digunakan yaitu jarak dan waktu. Tujuan yang ingin dicapai biasanya meminimalkan jarak tempuh, waktu maupun biaya (Indra dkk, 2014).

Berdasarkan faktor pembatasnya, VRP terbagi menjadi beberapa jenis, diantaranya :

a. *MTVRP (Multiple Trips Vehicle Routing Problem)*

Setiap kendaraan dapat melayani lebih dari satu rute untuk memenuhi kebutuhan konsumen.

b. *VRPTW (Vehicle Routing Problem with Time Windows)*

Setiap konsumen yang dilayani oleh kendaraan memiliki batas waktu menerima pelayanan.

c. *Pickup and Delivery Vehicle Routing Problem*

Terdapat sejumlah barang yang perlu dipindahkan dari lokasi penjemputan tertentu ke lokasi pengiriman lainnya.

d. *CVRP (Capacited Vehicle Routing Problem)*

Kendaraan yang memiliki keterbatasan daya angkut (kapasitas) barang yang harus diantarkan ke suatu tempat.

e. *VRP with Multiple Products*

Konsumen memiliki pesanan lebih dari satu jenis produk yang harus diantarkan.

f. *MDVRP (Multiple Depot Vehicle Routing Problem)*

Depot awal untuk melayani konsumen lebih dari satu.

g. *Periodic Vehicle Routing Problem*

Adanya perencanaan yang berlaku untuk satuan waktu tertentu.

h. *VRP with Heterogeneous Fleet of Vehicles*

Kapasitas masing-masing kendaraan tidak selalu sama. Jumlah dan tipe kendaraan diketahui.

**F. *Capacitated Vehicle Routing Problem with Time Windows***

**1. Pengertian**

*Capacitated vehicle routing problem with time windows* (CVRPTW) adalah salah satu jenis VRP yang merupakan kombinasi dari bentuk umum *capacitated vehicle routing problem* (CVRP) dan *vehicle routing problem with time windows* (VRPTW). CVRPTW bertujuan untuk membentuk rute optimal untuk memenuhi permintaan konsumen yang dilakukan secara *delivery* dengan kendala kapasitas dan *time windows*.

Kendala pertama pada CVRPTW adalah kendala kapasitas. Kendala kapasitas yang dimaksud adalah bahwa setiap kendaraan memiliki kapasitas tertentu dan jika kapasitas kendaraan sudah penuh, maka kendaraan tersebut tidak dapat melayani konsumen selanjutnya. Kendala berikutnya adalah kendala *time windows* pada masing-masing konsumen dan *time windows* pada depot. *Time windows* pada masing-masing konsumen  $[a_i, b_i]$  adalah interval waktu yang ditentukan oleh masing-masing konsumen bagi setiap kendaraan untuk dapat melakukan pelayanan pada konsumen tersebut. Kendaraan dapat memulai pelayanan di antara waktu awal konsumen ( $a_i$ ) dan waktu akhir konsumen ( $b_i$ ). Namun kendaraan juga harus menunggu sampai waktu awal konsumen dapat

dilayani apabila kendaraan tersebut datang sebelum waktu awal konsumen, sedangkan *time windows* pada depot  $[a_0, b_0]$  didefinisikan sebagai interval waktu yang menunjukkan waktu awal keberangkatan kendaraan dari depot dan waktu kembalinya kendaraan ke depot, itu artinya kendaraan tidak boleh meninggalkan depot sebelum waktu awal depot ( $a_0$ ) dan harus kembali ke depot sebelum waktu akhir depot ( $b_0$ ).

## 2. Formulasi CVRPTW

Berikut akan diberikan pendefinisian variabel dan pemodelan matematika untuk CVRPTW.

Masalah CVRPTW dapat direpresentasikan sebagai suatu graf berarah  $G = (V, A)$  dengan  $V = \{v_0, v_1, v_2, \dots, v_n\}$  adalah himpunan simpul (verteks),  $v_0$  menyatakan depot dengan  $v_{n+1}$  merupakan depot semu dari  $v_0$  yaitu tempat kendaraan memulai dan mengakhiri rute perjalanan. Sedangkan  $A = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$  adalah himpunan rusuk atau garis berarah (arc) yang menghubungkan dua simpul yaitu ruas jalan penghubung antar konsumen ataupun antara depot dengan konsumen.

Setiap simpul  $\{v_i \in V, i \neq 0\}$  memiliki permintaan (*demand*) sebesar  $q_i$  dengan  $q_i$  adalah *integer positif*. Himpunan  $K = \{k_1, k_2, \dots, k_n\}$  merupakan kumpulan kendaran yang homogen dengan kapasitas maksimal yang identik yaitu  $Q$ , sehingga panjang setiap rute dibatasi oleh kapasitas kendaraan. Setiap verteks  $(v_i, v_j)$  memiliki waktu tempuh  $W_{t_{ij}}$  yaitu waktu tempuh dari simpul  $i$  ke  $j$ . Waktu tempuh perjalanan ini diasumsikan asimetrik yaitu  $W_{t_{ij}} \neq W_{t_{ji}}$  dan  $W_{t_{ii}} = 0$ .

Dari permasalahan CVRPTW tersebut kemudian diformulasikan ke dalam bentuk model matematika dengan tujuan meminimumkan total waktu tempuh kendaraan untuk melayani semua konsumen, jika  $z$  adalah fungsi tujuan maka

$$\min \quad z = \sum_{i=1}^n \sum_{j=1}^n (W t_{ij} \sum_{k=1}^m X_{ijk}) \quad \dots (2)$$

dengan variabel keputusan sebagai berikut.

1. Variabel  $X_{ijk}$ ,  $\forall i, j \in N, \forall k \in K, i \neq j$ .

Variabel  $X_{ijk}$  mempresentasikan ada atau tidaknya perjalanan dari konsumen ke- $i$  ke konsumen ke- $j$  oleh kendaraan ke- $k$ .

$$X_{ijk} \begin{cases} 1, & \text{jika ada perjalanan dari konsumen } i \text{ ke konsumen } j \text{ oleh kendaraan } k. \\ 0, & \text{jika tidak ada perjalanan dari konsumen } i \text{ ke konsumen } j \text{ oleh kendaraan } k. \end{cases}$$

2. Variabel  $T_{ik}$ ,  $T_{0k}$ , dan  $s_{ik}$   $\forall i \in N, \forall k \in K$ .

Variabel  $T_{ik}$  menyatakan waktu dimulainya pelayanan pada konsumen ke- $i$  oleh kendaraan ke- $k$ ,  $T_{0k}$  menyatakan waktu saat kendaraan ke- $k$  meninggalkan depot dan kembali ke depot, dan  $s_{ik}$  menyatakan lamanya pelayanan di konsumen ke- $i$ .

3. Variabel  $Y_{ik}$  dan  $q_j$ ,  $\forall i, j \in N, \forall k \in K$ .

Variabel  $Y_{ik}$  menyatakan kapasitas total dalam kendaraan ke- $k$  setelah melayani konsumen ke- $i$ , sedangkan  $q_j$  menyatakan banyaknya permintaan konsumen ke- $j$ .

Adapun kendala dari permasalahan CVRPTW adalah sebagai berikut.

1. Setiap konsumen hanya dikunjungi tepat satu kali oleh kendaraan yang sama.

$$\sum_{k=1}^m \sum_{j=1}^n \sum_{i=1}^n X_{ijk} = 1, \quad \forall i, j \in N, \forall k \in K \quad \dots (3)$$



2. Total jumlah permintaan konsumen dalam satu rute tidak melebihi kapasitas kendaraan yang melayani rute tersebut.

Jika ada lintasan dari  $i$  ke  $j$  dengan kendaraan  $k$ , maka

$$Y_{ik} + q_j = Y_{jk}, \forall i, k \in N, \forall k \in K \quad \dots (4)$$

$$Y_{jk} \leq Q, \forall j \in N, \forall k \in K \quad \dots (5)$$

3. Jika ada perjalanan dari konsumen ke- $i$  ke konsumen ke- $j$ , maka waktu memulai pelayanan di konsumen ke- $j$  lebih dari atau sama dengan waktu kendaraan ke- $k$  untuk memulai pelayanan di konsumen ke- $i$  ditambah waktu pelayanan konsumen ke- $i$  dan ditambah waktu tempuh perjalanan dari konsumen ke- $i$  ke konsumen ke- $j$ .

Jika ada lintasan dari  $i$  ke  $j$  dengan kendaraan  $k$ , maka

$$T_{ik} + s_{ik} + Wt_{ij} \leq T_{jk}, \forall i, j \in N, \forall k \in K \quad \dots (6)$$

4. Waktu kendaraan untuk memulai pelayanan di konsumen ke- $i$  harus berada pada selang waktu  $[a_i, b_i]$ .

$$a_i \leq T_{ik} \leq b_i, \forall i \in N, \forall k \in K \quad \dots (7)$$

5. Setiap rute perjalan berawal dari depot dan berakhir di depot.

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^m X_{ijk} = 1, \forall i, j \in N, \forall k \in K \quad \dots (8)$$

6. Kekontinuan rute, artinya kendaraan yang mengunjungi setiap konsumen, setelah selesai melayani akan meninggalkan konsumen tersebut.

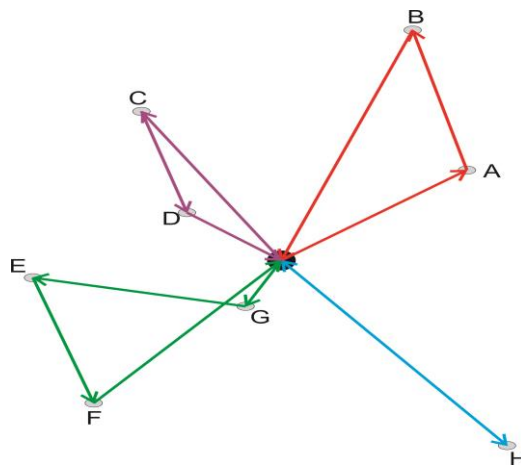
$$\sum_{i=1}^n X_{ijk} - \sum_{i=1}^n X_{ijk} = 0, \forall i, j \in N, \forall k \in K \quad \dots (9)$$

7. Variabel keputusan  $X_{ijk}$  merupakan integer biner

$$X_{ijk} \in \{0,1\}, \forall i, j \in N, \forall k \in K \quad \dots (10)$$

### 3. Representasi Solusi

Solusi layak dari formulasi CVRPTW yang dihasilkan adalah himpunan rute kendaraan yang memiliki total waktu tempuh minimal dengan memenuhi semua kendala yang ada. Himpunan rute tersebut dapat dituliskan sebagai berikut,  $rute = \{rute1, rute2, \dots, rute n\}$ . Solusi CVRPTW dapat digambarkan dalam bentuk graf yang setiap rute perjalanannya merupakan lintasan tertutup dengan depot sebagai simpul awal dan simpul akhir, sedangkan simpul lainnya adalah konsumen. Ilustrasi mengenai solusi layak dari CVRPTW seperti pada contoh berikut (Gambar 2.8).



**Gambar 2.8.** Ilustrasi Solusi Layak CVRPTW

Pada Gambar 2.8 terdapat 8 konsumen yaitu  $A, B, C, D, E, F, G, H$ . Dengan penggunaan algoritma eksak ataupun heuristik didapat sebuah solusi yang terdiri dari empat rute perjalanan kendaraan yang diawali dan diakhiri di depot serta memenuhi kendala yang ada pada CVRPTW. Rute pertama terdiri dari konsumen  $A$  dan  $B$ , rute kedua terdiri dari konsumen  $C$  dan  $D$ , rute ketiga terdiri dari konsumen  $E, F$  dan  $G$ , dan rute yang terakhir hanya terdiri dari konsumen  $H$  saja.

Dengan demikian, representasi solusi dari masalah CVRPTW tersebut adalah  $rute = \{0 - A - B - 0, 0 - C - D - 0, 0 - E - F - G - 0, 0 - H - 0\}$

dengan total waktu tempuh kendaraan =  $Wt_{0A,t} + Wt_{AB,t} + Wt_{B0,t} + Wt_{0C,t} + Wt_{CD,t} + Wt_{D0,t} + Wt_{0E,t} + Wt_{EF,t} + Wt_{FG,t} + Wt_{G0,t} + Wt_{0H,t} + Wt_{H0,t}$ .

## G. Algoritma

Menurut Al-Khawarizmi (dalam Sutejo, 1997 : 5) algoritma di dalam bidang pemrograman didefinisikan sebagai suatu metode khusus yang tepat dan terdiri dari serangkaian langkah-langkah yang terstruktur dan dituliskan secara sistematis yang dikerjakan untuk menyelesaikan suatu masalah dengan bantuan komputer, sedangkan menurut Antony Pranata (2000 : 8) definisi algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika.

Dari kedua definisi di atas, dapat disimpulkan bahwa algoritma adalah suatu metode khusus yang tepat dan terdiri dari langkah-langkah atau serangkaian langkah-langkah berhingga yang disusun secara logis, terstruktur, dan dituliskan secara sistematis untuk menyelesaikan suatu masalah. Dalam kehidupan sehari-hari, manusia juga menyelesaikan suatu masalah dalam hidupnya dengan menggunakan algoritma baik secara sadar maupun tidak.

Contoh 1:

Jika kita ingin mengambil uang tabungan di bank, pastinya akan melakukan suatu prosedur-prosedur dengan urutan sebagai berikut.

1. Mengambil no.antrian di bank
2. Mengambil blanko transaksi
3. Mengisi blanko transaksi
4. Menyerahkan blanko kepada petugas Bank
5. Menunggu antrian untuk dipanggil
6. Menerima uang tabungan tersebut dari petugas Bank

**Gambar 2.9.** Contoh Algoritma Mengambil Uang Tabungan di Bank

Contoh tersebut merupakan salah satu bukti bahwa apa yang dilakukan dalam kehidupan sehari-hari menggunakan suatu algoritma, karena prosedur-prosedur yang dilakukan merupakan suatu algoritma.

Algoritma merupakan sebuah himpunan berhingga yang berisi instruksi-instruksi yang mempunyai karakteristik tertentu. Menurut Richard Johnsonbaugh (2001 : 121) karakteristik algoritma adalah sebagai berikut :

1. Presisi (*precision*)

Presisi (*precision*) adalah langkah-langkah dari suatu algoritma yang harus dinyatakan dengan jelas sehingga algoritma tersebut dapat ditulis dalam bahasa pemrograman.

2. Unik (*uniueness*)

Unik (*uniueness*) adalah hasil lanjutan dari setiap langkah dari pelaksanaan didefinisikan secara tunggal dan semata-mata bergantung pada masukan dan hasil dari langkah sebelumnya.

3. Terhingga (*finiteness*)

Terhingga (*finiteness*) adalah algoritma berhenti setelah beberapa instruksi terhingga dilaksanakan. Jadi tidak mungkin jika algoritma berjalan terus tanpa henti.

4. Masukan (*input*)

Masukan (*input*) adalah algoritma yang memerlukan suatu masukan. Masukan adalah besaran yang diberikan kepada algoritma sebelum algoritma mulai bekerja.

5. Keluaran (*output*)

Keluaran (*output*) adalah sebuah algoritma yang menghasilkan suatu keluaran dimana keluaran tersebut adalah hasil dari proses yang nilainya tergantung pada masukan.

6. Umum (*generality*)

Umum (*generality*) adalah algoritma yang berlaku pada semua anggota himpunan masukan.

Terdapat dua macam metode untuk penyelesaian permasalahan VRP, yaitu metode eksak dan metode heuristik :

1. Algoritma Eksak

Penyelesaian *vehicle routing problem* (VRP) menggunakan algoritma eksak dilakukan dengan cara menghitung setiap solusi sampai ditemukan solusi terbaik. Kelemahan dari algoritma ini adalah menghabiskan waktu yang sangat lama karena waktu yang dibutuhkan untuk mencari solusi permasalahan akan bergerak secara eksponensial dengan semakin rumitnya permasalahan. Kelebihan dari algoritma eksak dibandingkan dengan solusi heuristik adalah selalu mendapat solusi yang optimum.

2. Algoritma Heuristik

Pada kasus-kasus tertentu mungkin solusi optimum dapat diberikan namun memerlukan proses perhitungan yang sangat panjang dan tidak praktis. Bahkan beberapa masalah hanya dapat ditentukan solusi optimumnya dengan cara mencoba satu persatu semua solusi yang mungkin terjadi.

Ketika dihadapkan dengan masalah dalam lingkup yang besar, pencarian solusi dengan cara mencoba solusi satu persatu akan memakan waktu yang sangat lama dan tidak efektif. Oleh karena itu dibutuhkan suatu metode praktis yang mampu melakukan pendekatan agar dapat meminimalkan waktu sesingkat mungkin dalam mencari solusi. Algoritma ini disebut heuristik.

Algoritma heuristik bersifat memberikan pendekatan terhadap solusi yang ingin dicari. Algoritma heuristik yang baik selalu dapat memberikan solusi yang mendekati optimum. Algoritma heuristik mempunyai ciri-ciri menemukan solusi yang baik tetapi tidak harus terbaik dan lebih cepat atau mudah diterapkan dibandingkan dengan algoritma terperinci yang diketahui.

Algoritma heuristik merupakan algoritma yang dapat memberikan solusi yang mendekati optimal lebih cepat dibandingkan dengan algoritma eksak. Kelebihan algoritma heuristik adalah lebih cepat dan fleksibel untuk diaplikasikan dibandingkan dengan algoritma eksak.

## **H. Algoritma *Floyd Warshall* dan *Nearest Neighbour***

### **1. Algoritma *Floyd Warshall***

Menurut Jong Jek Siang (2011 : 297) : algoritma *floyd warshall* untuk mencari lintasan terpendek merupakan algoritma yang sederhana dan mudah implementasinya. Prinsip algoritma *floyd warshall* yaitu pada iterasi ke-1, dihitung jarak terpendek dari semua titik ke semua titik. Misalkan  $W^{(0)}$  adalah matriks keterhubungan awal graf berarah berbobot,  $W^*$  adalah matriks ketetanggaan berbobot terpendek dengan  $W_{ij}^*$  sama dengan lintasan terpendek dari titik  $v_i$  ke  $v_j$ ,  $i$  adalah baris pada matriks,  $j$  adalah kolom pada matriks dan  $k$  adalah

iterasi ke 1 hingga  $n$  yang dilakukan pada matriks. Namun dalam hal ini kita tidak mencari lintasan terpendek melainkan waktu tempuh terpendek pada masalah CVRPTW.

Adapun langkah-langkah pemecahan masalah CVRPTW dengan metode *floyd warshall* sebagai waktu tempuh terpendek antar node.

a. Langkah 1

Buat waktu tempuh antar node-node ( $W_t^{(0)}$ ) pada saat  $t$  dengan rumus :  
 $W_{ij,t} = \text{jarak tempuh} \times 60 / \text{kecepatan pada saat } t \text{ di jalur } i-j$

b. Langkah 2

$$W_t = W_t^{(0)}$$

c. Langkah 3

Untuk  $h = 1$  hingga  $n$

Untuk  $i = 1$  hingga  $n$

Untuk  $j = 1$  hingga  $n$

Jika  $W_{ij,t} > W_{ih,t} + W_{hj,s}$ ,  $s = W_{ih,t}$

maka tukar  $W_{ij,t}$  dengan  $W_{ih,t} + W_{hj,s}$ ,  $s = W_{ih,t}$

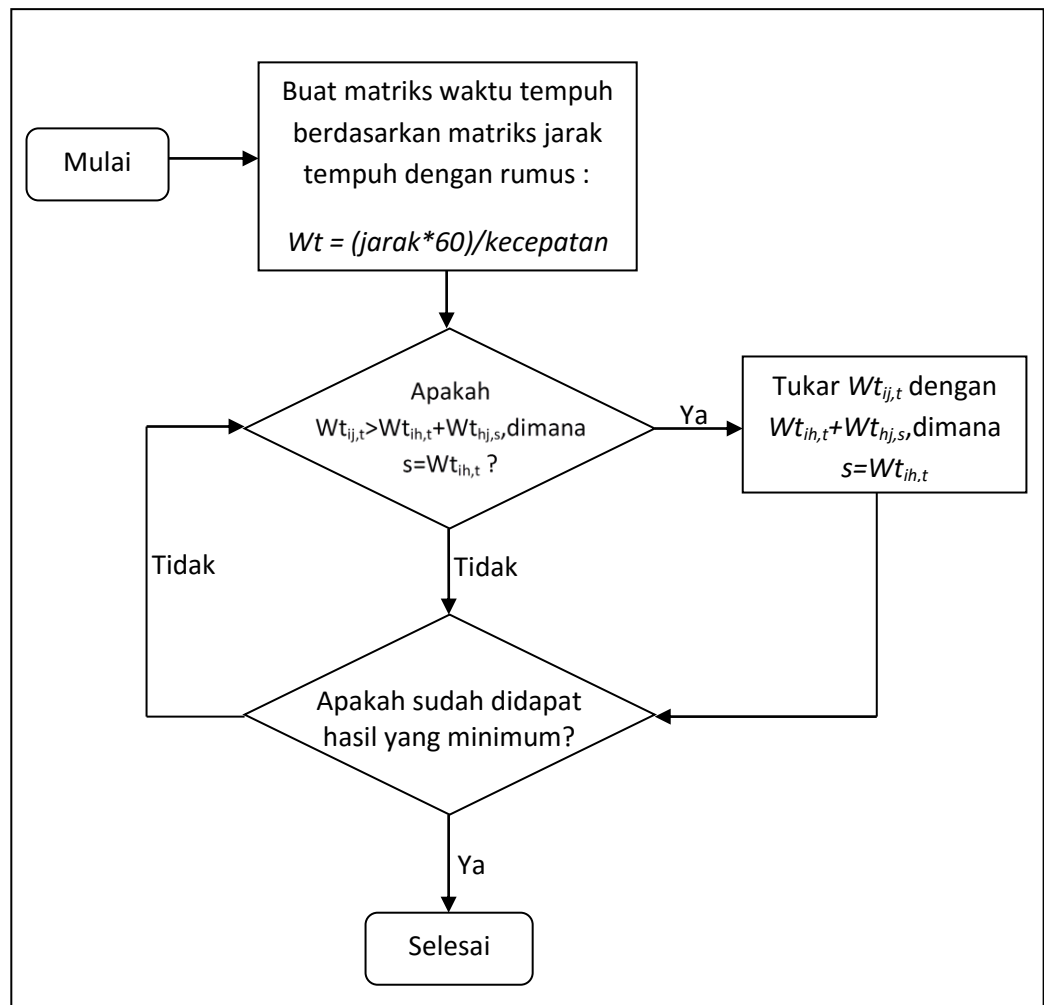
d. Langkah 4

$$W_t^* = W_t$$

e. Langkah 5

Ulangi langkah 4 sampai didapatkan hasil yang minimum.

Penyajian langkah-langkah dari algoritma *floyd warshall* dalam bentuk *flowchart* dapat dilihat pada Gambar 2.10.



**Gambar 2.10.** *Flowchart* Algoritma *Floyd Warshall*

## 2. Algoritma *Nearest Neighbour*

Menurut Chairul, dkk (2014) : Metode *nearest neighbour* merupakan metode yang digunakan untuk memecahkan masalah pemilihan rute dengan cara mencari jarak terpendek untuk menempuh lokasi pengiriman. Prinsip dasar dari metode ini adalah membentuk rute dengan memilih konsumen yang terdekat dari lokasi awal.



Adapun langkah-langkah pemecahan masalah CVRPTW dengan algoritma *nearest neighbour* adalah sebagai berikut.

a. Langkah 1

Set depot sebagai titik awal,  $t = 0$ , dan  $demand = 0$ .

b. Langkah 2

Cari konsumen ke- $j$  yang memiliki waktu tempuh terpendek dari titik awal  $i$ .

c. Langkah 3

Hitung total waktu tempuh kendaraan ( $Wt = t + \text{waktu pelayanan } i + Wt_{ij,t}$ ). Untuk  $Wt \geq a_j$  maka  $Wt = a_j$ . Jika  $Wt \leq b_j$  maka lanjut ke Langkah 4. Jika  $Wt > b_j$ , maka lanjut ke Langkah 6.

d. Langkah 4

Hitung permintaan/muatan kendaraan ( $demand = demand + q_j$ ). Jika  $demand \leq Q$ , maka lanjut ke Langkah 5. Jika  $demand > Q$ , maka lanjut ke Langkah 6.

e. Langkah 5

Set konsumen ke- $j$  sebagai titik awal, kemudian ulangi ke Langkah 3.

f. Langkah 6

Batalkan pemilihan konsumen, kemudian pilih konsumen yang belum dilayani dan yang terdekat dengan titik awal berdasarkan keterurutan dan kembali ke Langkah 3. Jika semua konsumen tidak ada yang layak, lanjutkan ke Langkah 7.

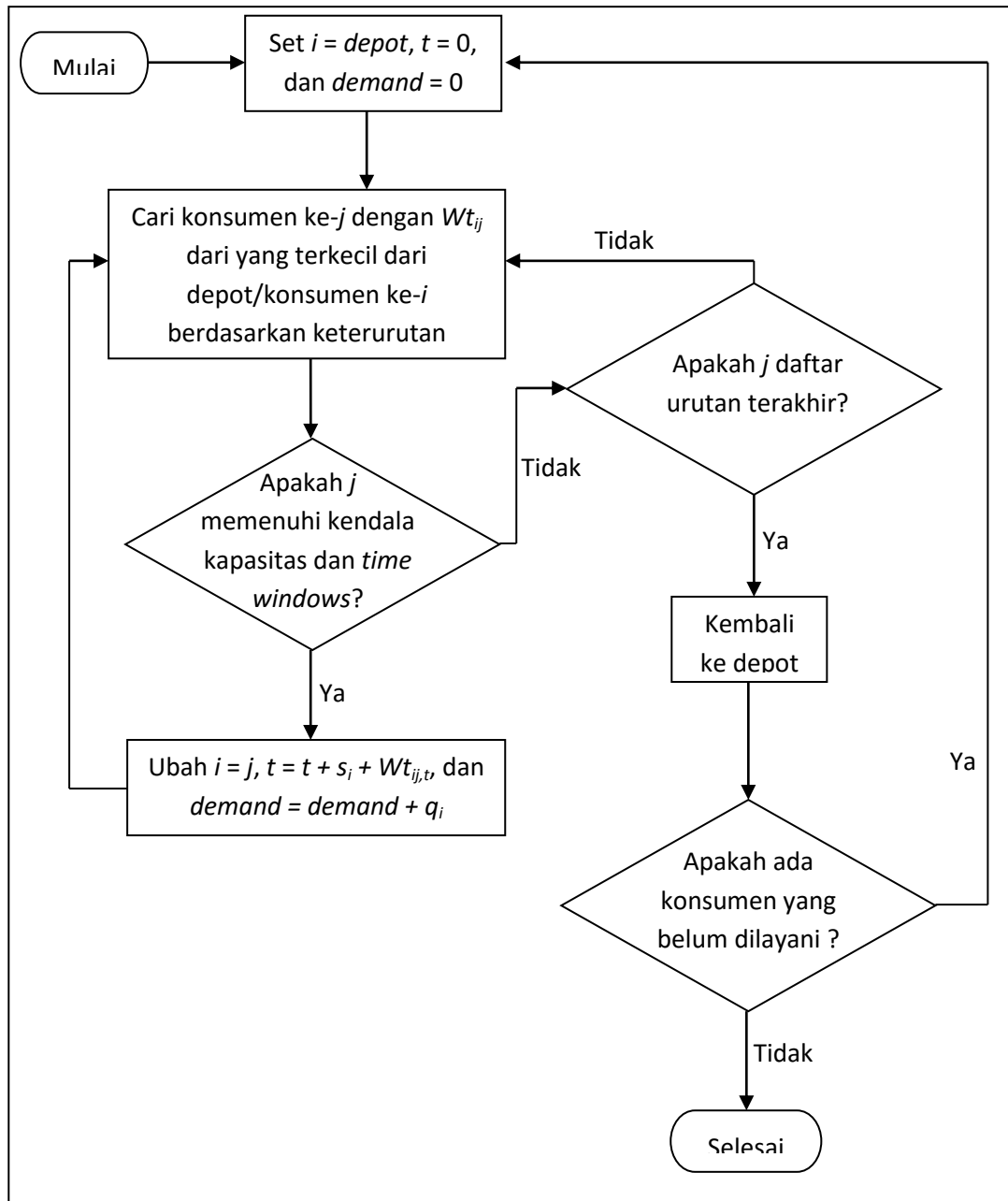
g. Langkah 7

Kembali ke depot dan lanjut ke Langkah 8.

h. Langkah 8

Jika semua konsumen telah dilayani maka algoritma dihentikan. Jika ada konsumen yang belum dilayani maka kembali ke Langkah 1.

Penyajian langkah-langkah dari algoritma *nearest neighbour* dalam bentuk *flowchart* dapat dilihat pada Gambar 2.11.



**Gambar 2.11.** *Flowchart* Algoritma *Nearest Neighbour*