

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Distribusi**

Distribusi merupakan suatu proses yang dapat mengirimkan barang dari tempat asal atau perusahaan menuju ke tempat tujuan misalnya agen atau toko-toko. Dengan adanya proses pendistribusian, barang dan jasa dapat digunakan oleh konsumen. Sehingga kegunaan dari barang dan jasa yang telah melalui proses distribusi akan lebih meningkat jika sudah dikonsumsi.

Menurut (Kotler, 2007) saluran distribusi terbagi atas tiga bagian, yaitu :

- a) Aliran distribusi secara langsung  
Aliran distribusi dari produsen dapat langsung digunakan oleh agen atau toko-toko.
- b) Aliran distribusi secara semi langsung  
Aliran distribusi dari produsen harus melalui perantara dahulu, berupa distributor yang kemudian akan sampai ke konsumen.
- c) Aliran distribusi secara Non langsung

Distribusi menurut (Kotler, 2007) yaitu proses kegiatan pengiriman barang atau pendistribusian hasil produksi yang berupa barang dan jasa dari produsen ke konsumen untuk memenuhi keinginan agen dan toko-toko. Distribusi ini dapat dilakukan oleh pihak yang disebut sebagai distributor.

- **Tujuan Distribusi**

Distribusi bertujuan agar barang atau jasa hasil produksi dapat sampai dengan cepat kepada masyarakat, dengan tetap memperhatikan kondisi produsen dan sarana yang tersedia di pasaran, dimana sistem distribusi yang baik dapat mendukung kegiatan produksi dan konsumsi.

#### **2.1.1 Fungsi Distribusi**

Fungsi distribusi dapat dilakukan oleh badan usaha maupun perorangan mulai dari pengumpulan barang dengan cara membelinya dari produsen kemudian disalurkan kepada konsumen. Berdasarkan pernyataan ini, maka fungsi distribusi dapat dibagi menjadi :

- a. Fungsi pertukaran, kegiatan pemasaran maupun jual beli barang/jasa yang meliputi pembelian, penjualan, dan pengambilan risiko (untuk mengatasi risiko yang dapat terjadi dalam proses perdagangan maka harus menciptakan lingkungan pasar yang kondusif, serta asuransikan barang dagangan yang akan dan sedang dikelola).
- b. Fungsi penyediaan fisik, berupa penyedia bahan baku yang cukup dengan jumlah yang sangat banyak menyangkut tempat penyimpanan, pemilahan, dan pengangkutan.
- c. Fungsi penunjang, berkaitan dengan usaha penyediaan fasilitas kepada fungsi-fungsi lain supaya proses pendistribusian dapat berjalan dengan lancar, fungsi ini mencakup pelayanan, pembelanjaan, penyebaran informasi, dan koordinasi.

### 2.1.2 Sistem Distribusi

Sistem distribusi yaitu pengaturan proses pendistribusian barang dan jasa mulai dari hulu ke hilir atau mulai dari produsen kepada konsumen. Sistem distribusi dibedakan menjadi 2 yaitu sebagai berikut ini :

- a. Sistem distribusi cara pendek atau langsung  
Merupakan sistem distribusi dengan tidak menggunakan saluran distribusi. Contoh distribusi sistem berupa penyaluran hasil pertanian oleh petani ke pasar langsung.
- b. Sistem distribusi cara panjang atau tidak langsung  
Merupakan sistem distribusi dengan menggunakan saluran distribusi pada kegiatan distribusinya, biasanya melalui agen. Contoh : motor, mobil, TV

## 2.2 Program Linier

Berdasarkan (Sri, 2008), “Program Linier merupakan metode matematik dalam mengalokasikan sumber daya yang langka untuk mencapai suatu tujuan seperti memaksimalkan keuntungan atau meminimumkan biaya.” PL telah banyak diaplikasikan untuk membantu menyelesaikan permasalahan ekonomi, industri, militer, sosial, dll.

Berdasarkan (Yuwono, 2007), “Program linier (PL) adalah salah satu metode matematika yang digunakan untuk menyelesaikan masalah optimisasi, yaitu

memaksimumkan atau meminimumkan fungsi tujuan yang bergantung pada sejumlah variabel input.” Penting untuk kita lakukan yaitu mencari tahu tujuan untuk penyelesaian masalah dan penyebabnya.

Istilah “Linier” dalam *Linier Programming* memiliki arti sebagai hubungan antara factor yang bersifat linier atau konstan, serta fungsi matematik yang ditampilkan dalam bentuk haruslah fungsi linier. Hubungan linier yang apabila satu faktor berubah maka faktor lain akan berubah dalam jumlah konstan secara proporsional.

### 2.3 Pengertian *Travelling Salesman Problem* (TSP)

Masalah terkait *Travelling Salesman Problem* (TSP) berdasarkan pendapat (Madona, 2013) yang diterbitkan pada tahun 1800 oleh seorang matematikawan Irlandia, William Rowan Hamilton dan matematikawan Inggris, Thomas Penyangton, *Travelling Salesman Problem* diketahui sebagai suatu sebab permasalahan optimasi dan bersifat klasik serta *Non-Deterministik Pilynominal-time Complete* (NPC), yaitu tidak adanya penyelesaian yang dianggap paling optimal selain mencoba seluruh kemungkinan penyelesaian yang dapat dilakukan. Permasalahan yang melibatkan seorang *Travelling Salesman* mengharuskan melakukan sebuah kunjungan setidaknya sekali pada semua kota dalam suatu lintasan sebelum orang tersebut kembali ke titik awal, sehingga perjalanannya dapat dikatakan sempurna.

Berdasarkan pendapat Ferdian (2009) menjelaskan bahwa *Travelling Salesman Problem* (TSP) merupakan metode yang dapat menyelesaikan permasalahan pendistribusian untuk mencari rute terpendek dengan mempertimbangkan beberapa hal diantaranya proses pendistribusian mulai dari perusahaan dan kembali lagi ke perusahaan yang sama .

Berdasarkan pendapat Hanafi (2001) menjelaskan bahwa masalah *Travelling Salesman Problem* (TSP) dapat dinyatakan sebagai *salesman* yang akan mendistribusikan barang ke beberapa kota, dimana rangkaian kota yang orang tersebut kunjungi harus membentuk suatu rute sehingga dalam proses pendistribusian harus mengetahui rute yang tercepat dalam pendistribusian mulai dari hulu ke hilir atau pendistribusian mulai dari perusahaan dan akan kembali ke perusahaan.

Berdasarkan pendapat Glover (1997) *Travelling Salesman Problem* (TSP) adalah suatu penentuan rute perjalanan, berupa salah satu permasalahan yang sering

dihadapi dalam kehidupan sehari-hari. Salah satu contoh dari *Travelling Salesman Problem* (TSP) yaitu rute manakah yang akan memiliki biaya paling murah untuk dapat dilalui oleh seorang *salesman* ketika akan mengunjungi sejumlah daerah. Tiap daerah tersebut harus dikunjungi tepat satu kali lalu kembali lagi ke tempat semula.

### **2.3.1 Penyelesaian *Travelling Salesman Problem* (TSP)**

Permasalahan yang timbul dari *Travelling Salesman Problem* (TSP) dapat diselesaikan dengan berbagai cara, tergantung pada sistem permasalahan yang dihadapinya. Adapun metode penyelesaian dari permasalahan TSP yaitu *Vehicle Routing Problem* (VRP), *Nearest Neighbour*, *Algoritma Genetik*, *Algoritma Semut*, *Tabu Search*.

### **2.3.2 Metode *Vehicle Routing Problem* (VRP).**

Menurut Qin (2011) *Vehicle Routing Problem* (VRP) dapat diartikan sebagai masalah untuk menentukan jarak yang optimal sebagai bentuk pendistribusian barang/jasa kepada agen-agen atau toko yang mempunyai lokasi yang berbeda dan jumlah permintaan yang sudah diketahui, dari satu atau lebih agen yang memenuhi beberapa kendala.

Permasalahan *Travelling Salesman Problem* (TSP) akan berbeda dengan permasalahan pada metode *Vehicle Routing Problem* (VRP), karena metode VRP memiliki fungsi dan tujuan untuk melakukan pendistribusian yang dibatasi oleh beberapa hal, diantaranya jumlah barang yang akan di antar dengan jarak tempuh total, dan lain-lain. Sedangkan pada metode *Travelling Salesman Problem* (TSP), harus memperhatikan batasan jarak dari satu pelanggan kepada pelanggan lain tanpa batasan-batasan lainnya. Batasan-batasan tersebut berupa :

- 1) Setiap proses pendistribusian ke agen-agen akan mempunyai barang yang akan di kirim ataupun barang yang akan di ambil.
- 2) Agen kemungkinan mempunyai perhitungan kapasitas mobil yang dimilikinya. Untuk mengetahui kapasitas muat mobil dapat dilihat dari volume mobil dan berat barang yang dimuat,
- 3) Setiap agen akan mempunyai beberapa kendaraan mobil untuk proses pendistribusian barang agar dapat melayani permintaan konsumen atau agen-agen.

- 4) Agen mempunyai perhitungan dalam proses pendistribusian mulai dari perusahaan sampai agen-agen atau konsumen atau dimulai dari agen sampai kembali ke agen awal.
- 5) Setiap tempat pengangkutan maupun pengiriman dapat menerapkan aturan kunjungan, yaitu tempat pemberhentian hanya boleh dikunjungi pada waktu tertentu saja.

*Vehicle Routing Problem* (VRP) pertama kali dikemukakan oleh Dantzig dan Ramser pada tahun 1959 dalam bentuk rute dan penjadwalan truk. Clarke dan Wright pada tahun 1964 lalu melanjutkan penelitian ini dan berhasil menciptakan sebuah metode yaitu *Saving Algorithm*. Solusi dari sebuah VRP berupa sejumlah alternatif yang dapat digunakan dalam proses pendistribusian agar didapatkan rute yang terbaik untuk proses pendistribusian mulai dari perusahaan mendistribusikan barang ke setiap agen dan kembali lagi dari agen ke perusahaan (Qin, 2011). Ringkasnya karakteristik dari permasalahan *Vehicle Routing Problem* (VRP) adalah sebagai berikut :

- A. Setiap proses pendistribusian akan dimulai dari perusahaan dan kembali lagi ke perusahaan.
- B. Dalam proses pendistribusian ada beberapa agen yang harus dikunjungi dan dipenuhi permintaan dalam proses pendistribusian.
- C. Setiap kapasitas kendaraan sudah penuh maka kendaraan tidak akan melanjutkan ke agen yang lain untuk pengambilan barang, kendaraan harus pulang terlebih dahulu ke perusahaan untuk melayani titik berikutnya yang akan dituju.
- D. Tujuan dari permasalahan ini yaitu untuk meminimalkan total jarak yang di tempuh dengan cara memperhatikan semua rute yang dilewati dan memilih rute yang mempunyai waktu dan jarak yang minimum dalam setiap proses pendistribusian dari perusahaan sampai ke agen.

#### **2.3.2.1 Klasifikasi Jenis-jenis *Vehicle Routing Problem* (VRP)**

Dalam metode *Vehicle Routing Problem* (VRP) merupakan metode yang sangat membutuhkan jumlah faktor pembatas dan tujuan pendistribusian yang akan dilalui. Data yang sering digunakan yaitu jarak dan waktu dalam pendistribusian. Tujuan metode *Vehicle Routing Problem* (VRP) ini yaitu ingin meminimumkan jarak

tempuh, waktu tempuh dan biaya yang dikeluarkan. Menurut Qin (2011) *Vehicle Routing Problem (VRP)* terbagi menjadi beberapa jenis, diantaranya adalah :

a) *VRP with multiple trips*

Setiap proses pendistribusian dalam satu kendaraan dapat mendistribusikan lebih dari satu rute agar dapat memenuhi kebutuhan agen.

b) *VRP with time windows*

Setiap proses pendistribusian ke agen maka ada waktu batasan penerimaan barang dari perusahaan ke agen.

c) *VRP with pickup and delivery*

Setiap agen mempunyai sejumlah barang yang harus di distribusikan kesemua agen yang terdapat dalam perusahaan misalnya pendistribusian dari agen 1 ke agen berikutnya.

d) *Capacitated VRP*

Setiap kendaraan mempunyai batas maksimal barang dalam satu mobil, sehingga diperlukan perhitungan untuk mendistribusikan barang ke agen.

e) *VRP with Multiple Products*

Setiap agen akan mempunyai banyak pesanan dari konsumen dan harus di distribusikan pesanan yang sudah masuk ke setiap konsumen.

f) *VRP with Multiple Depots*

Setiap perusahaan harus mampu melayani permintaan banyak dari setiap agen dan konsumen.

g) *Periodic VRP*

Setiap perusahaan harus mempunyai strategi atau planning agar proses pendistribusian dapat berjalan lancar.

h) *VRP with heterogeneous fleet of vehicles*

Setiap kapasitas mobil yang dimiliki oleh perusahaan pasti akan berbeda kapasitas penampungan barang antara mobil 1 dan mobil lainnya. Jenis kendaraan yang ada dalam perusahaan harus di ketahui kapasitasnya.

### 2.3.2.2 *Algoritma Sweep*

Menurut Qin (2011) *Algoritma Sweep* yaitu algoritman yang dibagi menjadi dua tahap, yaitu tahap pertama terdiri dari clustering agen yang mana *clustering* awal

dilakukan dengan menyatukan beberapa agen-agen dalam satu clustering berdasarkan kapasitas kendaraan, tahap ke dua yaitu membentuk alternatif rute-rute untuk masing-masing agen. Dalam penyelesaian masalah *Vehicle Routing Problem* (VRP) dengan menggunakan metode *Algoritma Sweep* maka ada beberapa langkah-langkah yang harus diperhatikan diantaranya sebagai berikut:

1. Tahap Pengelompokan (*Clustering*):

Langkah awal dalam *Algoritma Sweep* yaitu mengelompokkan tujuan atau titik tempat agen ke dalam sebuah cluster. Dalam pengelompokan cluster ada beberapa langkah yang harus diperhatikan seperti dibawah ini:

- a. Pada tahap ini akan menggambarkan setiap agen-agen (agen di setiap perusahaan di titik) dalam kordinat kartesius akan menetapkan titik agen sebagai pusat titik kordinat.
- b. Pada tahap selanjutnya akan menntukan semua titik kordinat setiap depo yang terdapat dalam perusahaan. Langkah untuk mengubah titik kordinat (  $x, y$  ) menjadi titik kordinat polar (  $r, \theta$  ) adalah sebagai berikut:  $r = \sqrt{x^2 + y^2}$ . Setelah didapatkan tiitk kordinat semua agen-agen maka selanjatnya akan menggambarkan titik-titik tersebut kedalam bidangn dua dimensi. Langkah untuk mengubah koordinat kartesius menjadi koordinat polar juga dapat dilakukan dengan menggunakan *Software Geogebra*.
- c. Selanjutnya akan mengelompokan setiap agen-agen menggunakan metode (*clustering*) dimulai dari agen yang mempunyai titik terdekat dari perusahaan sampai agen yang terjauh dengan memperhatikan kapasitas kendaraan.
- d. Memastikan semua agen yang terdapat dalam perusahaan masuk ke dalam metode *clustering*.
- e. Pengelompokkan setiap agen yang terdapat dalam perusahaan akan di berhentikan apabila dalam satu agen melebihi kapasitas maksimal kendaraan.
- f. Membuat rute terbaru dengan langkah yang sama dengan menggunakan langkah c dimulai dari titik yang mempunyai jarak terdekat yang belum termasuk ke dalam cluster sebelumnya (agen terakhir yang terdapat dalam perusahaan).

- g. Mengulangi lagi tahap c - f, sampai semua titik telah dimasukkan dalam sebuah cluster.

### 2.3.3 Metode *Nearest Neighbour*

Pada metode *Travelling Salesman Problem* (TSP) mempunyai beberapa cara untuk menyelesaikan masalah, tergantung permasalahan yang sedang terjadi didalam perusahaan. Adapun beberapa metode yang dapat menyelesaikan permasalahan tentang pendistribusian yaitu menggunakan metode *Nearest Neighbour*, *Insertion*, dan *Sweep*.

Dijelaskan oleh Ferdian (2009) pada metode *nearest neighbour*, rute proses pendistribusian yang dipilih mempunyai waktu dan jarak yang terdekat dari perusahaan, kemudian akan dipilih daerah selanjutnya yang belum dikunjungi dan mempunyai rute dan jarak yang minimum.

Metode *Nearest Neighbour* berupa metode yang sederhana untuk menyelesaikan permasalahan *Travelling Salesman Problem* (TSP) yakni dapat dipilih melalui salah satu rute awal yang memiliki waktu dan jarak paling terdekat dengan perusahaan. Lalu pemilihan rute alternatif yang hendak dikunjungi selanjutnya harus memperhatikan waktu dan jarak ditempuh pada proses pendistribusian. Kemudian, ketika semua agen sudah dikunjungi atau semua agen sudah terhubung, maka rute perjalanan harus ditutup dengan cara kembali ke kota asal (node asal). Langkah yang harus dipilih pada metode ini secara umum yaitu :

1. Langkah 0 : Inisialisasi  
Penentuan  $N = \{1,2,3,4,\dots,n\}$  yang menyatakan jumlah agen atau tempat pada perusahaan yang akan dikunjungi. Tentukan sembarang salah satu agen yang dipilih sebagai titik awal dari perjalanan ( $i_0$ ), dan  $V$  menyatakan jumlah kota lain yang harus dikunjungi, serta  $S$  menyatakan urutan rute perjalanan pada saat ini. Pada langkah 1,  $S = (i_0)$ , karena ada kota lain yang belum dikunjungi.
2. Langkah 1 : kunjungi agen di kota berikutnya yang sudah dipilih  
Apabila  $i_1$  merupakan urutan terakhir dari agen di kota tersebut dalam pendistribusian  $S$ . Maka, agen berikutnya harus ditemukan ( $j^*$ ) yang mempunyai jarak paling minimal dengan  $i_1$ , dimana  $j^*$  adalah anggota dari  $V$ . Ketika ada banyak pilihan optimal maka harus dipilih secara acak.



3. Langkah 2 : kemudian pilih agen yang memiliki rute paling terdekat Masukan kota  $j$  \* pada urutan akhir, lalu pilih alternatif terhadap agen dengan jarak yang terdekat dan waktu yang terkecil pada proses pendistribusian.
4. Langkah 3 : Apabila seluruh agen yang harus dikunjungi sudah ditambahkan pada rute atau  $V=0$ , maka tertinggalnya kota tidak akan terjadi. Kemudian, tutup rute dengan cara menambahkan kota inisialisasi atau  $i_0$  pada akhir rute. Maka dari itu, rute akan ditutup dan kembali lagi ke kota awal. Apabila terjadi berikutnya lakukan dari langkah 1 lagi.

Lokasi pada koordinat cartesius didapatkan dengan proses pemetaan pada lokasi, maka digunakan Persamaan 2.1 dan 2.2 sebagai berikut :

$$\text{Koordinat-}X_i = \text{Koordinat } BT_i - \text{Koordinat } BT_0 \dots\dots\dots (2.1)$$

dengan: koordinat- $X_i$  = koordinat- $X$  lokasi  $i$   
koordinat- $BT_i$  = koordinat- $BT$  lokasi  $i$   
koordinat- $BT_0$  = koordinat- $BT$  lokasi depot

$$\text{Koordinat-}Y_i = \text{Koordinat } LS_0 - \text{Koordinat } LS_i \dots\dots\dots (2.2)$$

dengan: koordinat- $Y_i$  = koordinat- $Y$  lokasi  $i$   
koordinat- $LS_i$  = koordinat- $LS$  lokasi  $i$   
koordinat- $LS_0$  = koordinat- $LS$  lokasi depot

Saat pemilihan alternatif pada rute tersebut dapat ditentukan bahwa alternatif pendistribusian ke agen ( $BT;LS$ ), pada koordinat cartesius ditentukan titik (0:0) sebagai titik pusatnya.

Metode *Nearest Neighbour* adalah salah satu metode rute distribusi yang mempunyai karakteristik distribusi sesuai dengan kondisi di lapangan, metode ini digunakan karena teknik penentuan rute yang diterapkan lebih mudah untuk dilakukan jika dibandingkan dengan metode *Travelling Salesman Problem* (TSP) lainnya dan metode *Nearest Neighbour* ini merupakan metode acuan dasar dalam pembuatan rute distribusi yang dapat pula digunakan pada metode lainnya.

Menurut Qin (2011) beberapa langkah yang dapat digunakan untuk mengerjakan pembentukan rute dengan menggunakan Metode *Nearest Neighbour* :

- 1) Langkah 0 : Inisialisasi
  - 1.1 Tentukan satu titik awal (agen) perjalanan.
  - 1.2 Tentukan  $C = \{1, 2, 3, 4, \dots, N\}$  untuk jumlah agen yang akan dikunjungi.

1.3 Tentukan urutan rute perjalanan pada saat ini.

- 2) Langkah 1 : Pemilihan agen selanjutnya yang hendak dikunjungi  
Apabila  $n_1$  merupakan urutan terakhir dari titik pada rute R maka titik berikut yang didapat berupa titik  $n_2$  yang mempunyai jarak paling minimum dengan  $n_1$ , ketika  $n_2$  merupakan anggota dari rute R. Jika memiliki banyak pilihan optimal berarti terdapat lebih dari satu titik yang mempunyai jarak sama dari titik terakhir pada rute R dan jarak tersebut adalah jarak yang paling minimum hendaklah pilih secara acak.
- 3) Langkah 2 : Menambahkan agen yang dipilih pada langkah 1 pada urutan rute selanjutnya.  
Menambahkan titik  $n_2$  di urutan akhir dari rute sementara dan mengeluarkan yang terpilih tersebut dari daftar titik yang belum dikunjungi.
- 4) Langkah 3: Jika semua agen yang terdapat dalam perusahaan sudah mempunyai alternatif rute  $C = 0$  atau, maka tidak ada lagi titik yang ada di node, selanjutnya akan menutup rute dengan menambahkan alternatif rute lain perjalanan pendistribusian. Dengan kata lain, rute ditutup dengan kembali lagi ke titik asal. Jika sebaliknya, kembali melakukan langkah 1.

#### 2.3.4 Metode Algoritma Genetik

Definisi *Algoritma Genetik* didasarkan pada suatu prinsip seleksi alam yaitu “siapa yang kuat, dia yang bertahan”. *Algoritma Genetik* pertama kali ditemukan oleh John Holland pada tahun 1960. John Holland dengan rekan-rekan dan temennya, John Holland menerbitkan *Algoritma Genetik* dalam buku yang berjudul *Adaption of Natural and Artificial Systems* pada tahun 1975 (Coley, 1999). Algoritma Genetik (AG) merupakan algoritma optimasi ini terinspirasi dari gen dan alam. Algoritma ini dapat memberikan solusi-solusi data dalam bentuk kromosom-kromosom dan mengimplementasikan operasi rekombinasi genetik ke struktur data tersebut (Whitley, 2002).

Menurut Hanafi (2001) algoritma genetic mempunyai beberapa langkah diantaranya sebagai berikut ini:

- a. *Gen (Genotype)* merupakan suatu angka yang memperlihatkan satuan dasar dalam membuat arti tertentu pada satu kesatuan gen yang disebut kromosom.
- b. *Allele* merupakan hal utama yang ada di dalam gen, dapat berbentuk bilangan biner, *float*, *integer*, karakter dan kombinatorial.
- c. Kromosom merupakan kumpulan beberapa gen yang membuat sejumlah nilai dengan pasti.
- d. Individu adalah keadaan atau kondisi yang menunjukkan salah satu solusi yang menunjukkan permasalahan yang akan terjadi atau yang akan diangkat.
- e. Populasi adalah sekelompok individu yang hendak melakukan proses bersamaan dalam sebuah siklus proses evolusi. Populasi ini merupakan sejumlah kromosom.
- f. Induk, adalah kromosom yang ditandai pada operasi genetic (*crossover*)
- g. *Crossover* adalah salah satu operasi genetik untuk mengamati proses perkembangbiakan antara individu.
- h. *Offspring* adalah kromosom yang mempunyai hasil dari operasi genetik (*crossover*) disebut dengan keturunan atau anak.
- i. Mutasi adalah operasi genetik untuk dapat mewakili proses mutasi dalam berlangsungnya kehidupan suatu individu. Mutasi ini dapat memberikan suatu perbedaan yang berubah secara acak dalam suatu kelompok, yang dapat digunakan untuk menambah variasi dari sejumlah kromosom pada sebuah kelompok.
- j. Proses Seleksi yaitu suatu proses yang dapat memimpin proses seleksi yang berlangsung secara alami (*natural selection*) dari teori Darwin. Proses ini digunakan untuk dapat menjadikan induk dari operasi genetic (*crossover*) yang akan dilakukan untuk dapat menghasilkan keturunan (*offspring*).
- k. Nilai *fitness* yaitu penilaian sebuah *kromosom* yang dianggap bagus.
- l. Fungsi Evaluasi yaitu fungsi ini dapat dimanfaatkan untuk menentukan nilai *fitness*. Fungsi evaluasi adalah sejumlah ide-ide tertentu yang dapat digunakan untuk menyelesaikan suatu permasalahan.
- m. Generasi yaitu bagian dari populasi yang dapat mengalami beberapa operasi genetika, berkembang biak, dan menghasilkan keturunan atau

anak. Pada akhir dari setiap generasi, agar jumlah kromosom tetap terjaga dalam sebuah populasi untuk tetap konstan, kromosom-kromosom dengan nilai fitness yang rendah dan mempunyai nilai dibawah nilai minimal akan dihapus dari populasi.

Menurut Glover (1997) Secara umum, langkah-langkah dalam algoritma genetic sebagai berikut:

- 1) Memberikan nilai secara acak kepada populasi awal yang ada.
- 2) Membuat suatu generasi yang baru dengan mengaplikasikan operasi seleksi, operasi crossover dan operasi mutasi melalui tahapan diperoleh dari kromosom yang cukup agar dapat terbentuk generasi baru yang dapat menjadi representasi dari solusi baru.
- 3) Melakukan evaluasi pada setiap kelompok dengan memperhatikan nilai fitness pada tiap kromosom agar dapat memenuhi kriteria yang cukup untuk berhenti. apabila kriteria tersebut terhenti sebelum dapat terpenuhi, maka kromosom akan membentuk lagi generasi baru dengan cara mengulangi langkah 2.

### 2.3.5 Metode *Tabu Search*

*Algoritme tabu search* adalah sebuah metode optimisasi matematik yang dapat mencari solusi ke arah penentuan solusi lokal atau pencarian solusi *neighborhood* secara iteratif dapat memulai melalui agen selanjutnya melakukan pencarian ke arah solusi lain dengan menambahkan status tabu pada solusi yang sudah terdapat pada pencarian sebelumnya. *Algoritme tabu search* yang bermula ditemukan oleh *Fred Glover* pada tahun 1986. *Tabu search* mempunyai dua struktur memori yang berbentuk *short-term memory* dan *long term memory* dengan fungsi agar dapat melakukan evaluasi solusi yang sudah ditemukan. *Short-term memory* terdiri dari *tabu list*, *tabu tenure* dan kriteria aspirasi. *Short-term memory* sebagai penghilang *cycle* terhadap solusi yang sudah ditemukan pada iterasi-iterasi sebelumnya. *Long-term memory* mulanya dimanfaatkan sebagai penyimpanan informasi secara lebih detail dari solusi yang sudah terekam selama dalam proses pencarian solusi. *Longterm memory* dapat dimanfaatkan sebagai fase intensifikasi dan diversifikasi.

Menurut Glover (1997) Algoritma *tabu search* merupakan metode *metaheuristic* yang mengarahkan cara pencarian *local heuristic* agar dapat

memeriksa ruang solusi melebihi local optimum. Salah satu komponen utama *tabu search* adalah penggunaan *adaptive memory* yang membuat pencarian bersifat lebih fleksibel. Prinsip pada *tabu search* terkadang cukup berpotensi untuk menghasilkan penyelesaian masalah yang efektif dengan ketentuannya dan mengabaikan ketergantungannya dengan memori.

Menurut Ferdian (2009) *Algoritme tabu search* mempunyai beberapa langkah penting yang menjadi dasar setiap langkah pada *algoritme*. Langkah-langkah penting ini dapat dilihat sebagai berikut:

1. Menentukan solusi awal,
2. Mencari solusi *neighborhood*,
3. Menggunakan beberapa memori yaitu *short-term memory* dan *long-term memory*, 4 penghentian algoritme.

Menurut Glover (1997) dasar dalam penggunaan *tabu search* memiliki 3 hal diantara sebagai berikut:

1. Menggunakan atribut berbasis struktur memori yang fleksibel dan didesain untuk memperbolehkan kriteria evaluasi dan pencarian historis informasi untuk dieksploitasi sepenuhnya dengan struktur memori yang lebih rigid atau dengan pendekatan secara random.
2. Sebuah *control* mekanisme yang terasosiasi untuk menggunakan struktur memori berdasarkan kondisi dan kendala yang saling mempengaruhi serta bebas dalam proses pencarian (diwujudkan dalam *tabu restriction* dan *aspiration criteria*)
3. Menggabungkan suatu fungsi memori dengan rentang waktu yang berbeda, mulai dari jangka pendek sampai pada jangka panjang agar dapat memiliki strategi dengan mengintensikan dan mendiversikan pencarian.

Menurut Glover (1997) Strategi utama dalam *tabu search* terdapat 3 sebagai berikut:

1. *Forbidding strategy*  
Mengamati apa yang dapat masuk ke dalam *tabu list*
2. *Freeing strategy*  
Mengamati apa yang dapat keluar dari *tabu list* dan catatan waktunya.
3. *Short-tern strategy*  
Mengatur saling berpengaruhnya antara *forbidding strategy* dan *freeing strategy* untuk memilih solusi trial.

### 2.3.5.1 Menentuka solusi awal

Menentukan permulaan solusi *tabu search* dengan melakukan beberapa strategi pengelompokan pelanggan (*smallest maximum load* atau *largest maximum load*) dan standar penentuan rute (*independent grouping and routing* (IGR) atau *simultaneous grouping and routing* (SGR)).

Contohnya *net demand* (permintaan bersih) dari konsumen adalah  $nd(i)$  berupa selisih jumlah barang yang tersedia dan dikirim dengan jumlah barang yang dikirim sesuai dengan permintaan pelanggan  $i$ , yaitu

$$nd(i) = p_i - d_i$$

Apabila  $nd(i) < 0$  maka berkurangnya beban kendaraan sebesar  $nd(i)$  setelah mendatangi pelanggan. Apabila  $nd(i) > 0$  maka bertambahnya beban kendaraan sebesar  $nd(i)$  setelah mendatangi pelanggan.

*Smallest maximum load* dapat terjadi saat kendaraan mendistribusikan barang kepada sejumlah pelanggan yang memiliki  $nd(i) < 0$  terlebih dahulu lalu selanjutnya mendatangi pelanggan lain yang memiliki  $nd(i) > 0$ . *Largest maximum load* terjadi saat kendaraan mendatangi sejumlah pelanggan dengan  $nd(i) > 0$  terlebih dahulu lalu selanjutnya mendatangi pelanggan dengan  $nd(i) < 0$ . Dapat dilakukan penyisipan pelanggan baru pada suatu grup apabila tidak melanggar batas daya angkut kendaraan.

Penentuan rute didasarkan atas *independent grouping and routing* (IGR) dapat digunakan cara menentukan semua pelanggan kedalam satu grup terlebih dahulu, selanjutnya akan menentukan rute setiap kendaraan. Prosedur ini akan digunakan untuk memecahkan masalah tanpa adanya kendala pada jarak maksimum.

Penentuan rute didasarkan atas *simultaneous grouping and routing* (SGR) dapat digunakan untuk memecahkan masalah pada jarak maksimum. Dengan standar prosedur ini, dapat menambahkan suatu agen baru pada suatu perusahaan harus memiliki dua hal, yaitu dimasukkannya pelanggan ke dalam sebuah grup apabila ada tambahan *pick-up* dan *delivery* kepada pelanggan yang tidak melebihi batas maksimum kapasitas kendaraan serta jarak ditempuh pengantaran barang tidak melebihi jarak tempuh maksimum yang sudah ditentukan untuk kendaraan tersebut.

Menurut Qin (2011) ada empat standar prosedur heuristik yang dapat diaplikasikan sebagai penentu solusi awal *tabu search* ialah :

- a. IGR1 : penggolongan pelanggan dengan berdasarkan *smallest maximum load*, dan metode heuristik 2-opt dimanfaatkan sebagai perbaikan rute awal yang didapat dari pengurutan kunjungan pelanggan,
- b. IGR2 : penggolongan pelanggan dengan berdasarkan *largest maximum load*, dan metode *heuristik 2-opt* dimanfaatkan sebagai perbaikan rute awal yang didapat dari pengurutan kunjungan pelanggan,
- c. SGR1 : penggolongan pelanggan dengan berdasarkan *smallest maximum load*, dan pelanggan lainnya dimasukan tanpa melanggar batas daya angkut serta jarak tempuh maksimum kendaraan. Metode heuristik 2-opt dimanfaatkan sebagai perbaikan rute awal yang didapat secara berurutan terhadap kunjungan pelanggan,
- d. SGR2: penggolongan pelanggan dengan berdasarkan *largest maximum load*, dan pelanggan lainnya dimasukan tanpa tidak melanggar batas daya angkut dan jarak tempuh maksimum kendaraan. Metode *heuristik 2-opt* dimanfaatkan sebagai perbaikan rute awal yang didapat secara berurutan terhadap kunjungan pelanggan.

Menurut Glover (1997) dalam *Tabu Search* spesifikasi pemberhentian yang sering dilakukan adalah:

1. Sesudah seluruh iterasi yang sudah ditentukan sebelumnya tercapai.
2. Sesudah beberapa iterasi tanpa adanya perbaikan pada nilai fungsi objektif.
3. Saat fungsi objektif mencapai nilai batas maksimum atau nilai batas minimum yang sudah ditentukan sebelumnya
4. Saat solusi baru tidak ada yang bisa dibangkitkan dari *Nearest Neighbour* dimana seluruh move yang ada dalam *tabu list*.

Menurut Glover (1997) Langkah-langkah Algoritma *Tabu Search* adalah sebagai berikut:

- **Langkah 1**  
Tentukan standar solusi awal.
- **Langkah 2**  
Tentukan standar solusi alternatif dengan melakukan *move* (menukarkan) dua titik di dalam solusi.
- **Langkah 3**  
Lakukan evaluasi terhadap solusi-solusi alternatif dengan *tabu list* untuk menentukan apabila kandidat solusi (solusi alternatif) tersebut telah terdapat

pada *tabu list*. Jika solusi alternatif telah terdapat pada *tabu list*, maka solusi alternatif yang sudah dilakukan tidak akan dievaluasi kembali. Jika solusi alternatif belum terdapat pada *tabu list*, maka solusi alternatif yang sudah dilakukan disimpan pada *tabu list* sebagai solusi alternatif terbaik.

- **Langkah 4**

Pilihlah solusi terbaik dari seluruh daftar kandidat solusi alternatif dan menetapkannya sebagai solusi optimum yang baru.

- **Langkah 5**

memperbaharui *tabu list* dengan menambahkan solusi optimum yang baru.

- **Langkah 6**

Jika spesifikasi pemberhentian tercapai maka proses berhenti dan didapat solusi optimum. apabila tidak, proses tersebut kembali berulang mulai dari langkah kedua.

### 2.3.5.2 Mekanisme Pembentukan Solusi

Menurut Glover (1997) cara terbentuknya solusi tetangga adalah dengan ditentukannya suatu set operator yang diterapkan pada  $S$  untuk dapat hasil *move* ke solusi  $S'$  lainnya sebagai tetangga  $S$ ,  $N(S)$ . sebagai penerapannya, diambil cara terbentuknya *interchange* oleh Osman sebagai masalah *routing* dan *grouping*. Contohnya sepasang distributor dengan *retailer* ( $R_p$ ,  $R_q$ ) dalam  $S$ , *interchange* yang dapat dimanfaatkan adalah:

1. ***Interchange mechanism***

Tahapan pada mekanisme ini ada dua yaitu proses pindah (*shift*) dan proses tukar (*exchange*), tahapan perpindahan yang didasarkan oleh operator  $(1,0)$  dan  $(0,1)$ , sedangkan tahapan tukar yang didasarkan oleh operator  $(1,1)$ . Operator pindah  $(1,0)$  mengalihkan satu produk dari distributor  $R_p$  ke *retailer*  $R_q$  sedangkan operator pindah  $(0,1)$  memindahkan satu produk dari distributor  $R_q$  ke *retailer*  $R_p$  operator  $(1,1)$  pertukaran pada tiap-tiap satu produk dari rute distributor  $R_p$  ke *retailer*  $R_q$  secara serentak.



Sequence A      0 - 1 - 3 - 4 - 5 - 0 - 2 - 7 - 8 - 6 - 0 - 10 - 9 - 11 -  
0 - 0

Sequence B      0 - 2 - 7 - 8 - 6 - 0 - 10 - 9 - 11 - 0 - 0 - 1 - 3 - 4 -  
5 - 0

## 2. *Consecutive node interchange mechanism*

Mekanisme ini menggunakan semua operator pada *interchange mechanism* ditambah operator pindah (2,0) dan (0,2) dan operator tukar (2,1), (1,2) dan (2,2). Cara kerja operator-operator tersebut sama dengan pada *interchange mechanism* hanya saja jumlah produk yang dipindahkan atau dipertukarkan berbeda. Pada operator tambahan tadi yang berpindah atau dipertukarkan sebanyak dua konsumen, yang mana dua konsumen tersebut berurutan atau tidak berurutan.

Sequence A      0 - 1 - 3 - 4 - 5 - 0 - 2 - 7 - 8 - 6 - 0 - 10 - 9 - 11 -  
0 - 0

Sequence B      0 - 7 - 4 - 5 - 0 - 2 - 1 - 3 - 8 - 6 - 0 - 10 - 9 - 11 -  
0 - 0

### 2.3.5.3 *Komponen Tabu Search*

Struktur memori fundamental dalam *tabu search*, *tabu list*, menyimpan atribut dari sebagai *move* (transisi solusi) yang disimpan dalam iterasi-iterasi sebelumnya. *Tabu search* menggunakan *tabu-list* untuk mencari rute pendistribusian yang terdaik dengan cara memilih semua alternatif yang mempunyai jarak dan waktu yang paling minimum dalam proses semua pendistribusian ke setiap agen-agen atau konsumen. Dengan menggunakan strategi ini, *local search* yaitu metode yang telah digunakan dalam proses pendistribusian dengan menentukan rute yang terbaik atau yang memiliki waktu dan jarak yang minimum

Jadi *tabu list* hanya menyimpan langkah *transisi (move)* yang merupakan proses pendistribusi yang memiliki langkah-langkah yang kebalikan dari *local search*. Untuk *tabu list* berisi langkah-langkah yang kebalikan dari metode *local search*. Dengan kata lain *tabu list* berisi langkah-langkah yang membalikan solusi yang baru ke solusi yang lama.

Pada tiap iterasi, dipilih maka akan dipilih rute yang terbaik dalam pendistribusian menggunakan metode *neighborhood* dan tidak tergolong sebagai *tabu list*. Kualitas usulan dari rute yang terbaru ini tidak harus lebih baik dari rute

solusi sekarang. Apabila rute yang terbaru memiliki nilai dan waktu yang paling minimum dibandingkan dengan aliran rute yang telah digunakan sebelumnya, maka rute yang terbaik ini akan dijadikan rute yang paling optimal. Sebagai tambahan dari *tabu-list*, dikenal mempunyai kriteria aspirasi, merupakan suatu penanganan khusus terhadap *move* yang dinilai dapat menghasilkan solusi yang baik namun *move* tersebut bersatu tabu. Dalam hal ini, jika *move* tersebut memenuhi kriteria aspirasi yang telah ditetapkan sebelumnya, maka *move* tersebut dapat digunakan untuk membentuk solusi berikutnya.

#### 2.3.5.4 Prosedur umum *Tabu Search*

Menurut Glover (1997) Prosedur dalam pengoptimalan dengan *tabu search* adalah sebagai berikut:

1. Menentukan solusi awal
  - Solusi awal dapat diperoleh dari hasil pengolahan data dengan *sweeping method* atau secara random.
2. Inisialisasi *tabu search*
  - Pilihlah skema dari metode *tabu search* dengan memilih nilai untuk tiap parameternya
  - Pilihlah rute pendistribusian.
  - Matriks tabu list diset nol.
3. Lakukan iterasi
  - Pilihlah *move* untuk membuat solusi terbaik, dan pilih solusi  $S'$  terbaik yang diizinkan dari daftar kandidat.
  - Tentukan solusi saat ini (current solution)  $S$  menjadi  $S'$
  - Lakukan heuristic 2-opt untuk memperbaiki urutan dalam rute tersebut.
4. Perbarui skema *Tabu Search*
  - Perbaharui daftar dalam *tabu list*
  - Perbaharui komponen skema tabu tenure jika diperlukan.
5. Perbaharui solusi Baru
  - Jika  $C(S) < C(S_{best})$ , maka tentukan  $S_{best}$