BAB II

KAJIAN TEORI

Secara umum, pada bab ini membahas mengenai kajian teori yang digunakan dalam penelitian yaitu optimasi, graf, *traveling salesman problem* (TSP), *vehicle routing problem* (VRP), *capacitated vehicle routing problem* (CVRP), metode penyelesaian CVRP, algoritma *sweep*, algoritma genetika, dan penelitian yang relevan.

A. Masalah Optimasi

Brogan (1991 : 501) menyatakan bahwa optimasi ialah proses untuk mencapai hasil yang ideal atau optimal (nilai efektif yang dicapai). Optimasi secara intuisi berarti melakukan pekerjaan dengan cara terbaik. Sedangkan menurut Licker (2013 : 170), optimasi berasal dari kata bahasa Inggris *optimization* memiliki arti memaksimumkan atau meminimumkan sebuah fungsi yang diberikan untuk beberapa macam kendala.

Masalah optimasi digunakan untuk meyelesaikan permasalahan guna mencapai nilai minimal atau maksimal dari suatu fungsi nyata. Banyak masalah dalam dunia nyata yang dapat direpresentasikan dalam kerangka permasalahan ini, misal pendapatan yang maksimum, biaya yang minimum dan lain sebagainya. Apabila ada hal yang dioptimumkan ternyata kuantitatif, maka masalah optimum akan menjadi masalah maksimum dan minimum (Susanta, 1991). Hasil dari optimasi disebut sebagai hasil yang optimal.

Persoalan yang berkaitan dengan optimasi sangat kompleks dalam kehidupan sehari-hari. Nilai optimal yang didapatkan dalam optimasi dapat berupa besaran panjang, waktu, dan lain-lain. Optimasi mempunyai beberapa persoalan diantaranya :

1. Menentukan lintasan terpendek dari suatu tempat ke tempat lain.
2. Menentukan jumlah pekerja seminimal mungkin untuk melakukan suatu proses produksi agar pengeluaran biaya pekerja dapat diminimalkan dan hasil produksi tetap maksimal.
3. Mengatur jalur kendaraan umum agar semua lokasi dapat dijangkau.

Pada penulisan skripsi ini, optimasi yang ingin dicapai adalah optimasi rute. Optimasi rute adalah pencarian rute yang paling optimal (rute terpendek) dengan mempertimbangkan kapasitas kendaraan.

B. Graf

1. Definisi Graf

Graf G didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi $G = (V, E)$. Dalam hal ini, V merupakan himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) digambarkan dalam titik-titik, dan E adalah himpunan sisi-sisi (*edges* atau *arcs*) digambarkan dalam garis-garis yang menghubungkan sepasang simpul (Munir, 2009). Dapat dikatakan graf adalah kumpulan dari simpul-simpul yang dihubungkan oleh sisi-sisi.

2. Jenis-jenis Graf

Graf dapat dikelompokkan beberapa kategori (jenis) bergantung pada sudut pandang pengelompokkannya. Pengelompokkan graf dapat dipandang berdasarkan ada tidaknya sisi ganda atau sisi gelang (*loop*), berdasarkan jumlah simpul atau berdasarkan orientasi arah pada sisi (Munir, 2009).

1) Jenis graf berdasarkan ada tidaknya gelang dan rusuk ganda

Berdasarkan ada tidaknya gelang dan rusuk ganda graf dapat dibedakan menjadi 2 jenis yaitu graf sederhana dan tidak sederhana. Dalam sebuah graf ada kemungkinan dijumpai dua rusuk atau lebih yang menghubungkan dua simpul yang sama. Rusuk seperti ini disebut rusuk ganda. Ada pula rusuk yang menghubungkan simpul tertentu dengan dirinya sendiri yang disebut gelang (*Loop)*.Berdasar ada tidaknya gelang atau rusuk ganda, graf dapat dibedakan menjadi 2 jenis.

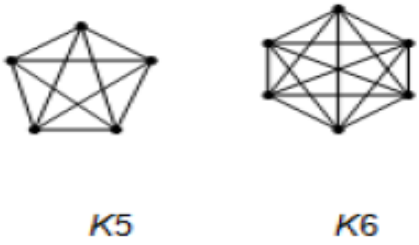
a. Graf Sederhana

Graf sederhana adalah graf yang tidak memuat rusuk ganda dan gelang. Beberapa graf sederhana dapat ditunjukkan sebagai berikut :

i. Graf nol adalah graf yang tidak memiliki rusuk atau himpunan rusuknya merupakan himpunan kosong. Gambar 2.1 menunjukkan graf nol dengan 2 buah simpul.

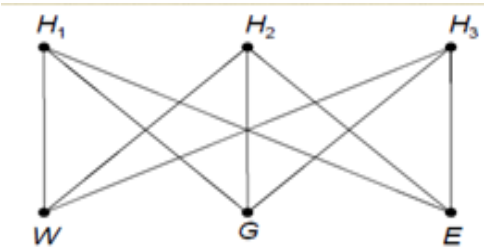
Gambar 2.1 Graf Nol dengan Banyak Simpul 2

ii. Graf lengkap adalah graf sederhana yang setiap pasang simpulnya saling berikatan. Notasi graf lengkap n simpul adalah K_n.



Gambar 2.2 Graf Lengkap

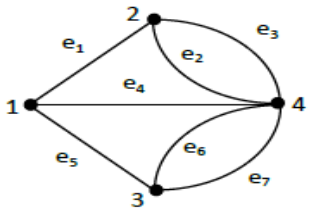
iii. Graf Bipartit adalah graf sederhana yang himpunan simpulnya dapat dipartisi menjadi 2 bagian, misal V_1 dan V_2 sehingga setiap rusuknya mempunyai titik ujung di V_1 dan titik ujung yang lain di V_2.



Gambar 2.3 Graf Bipartit

b. Graf Tidak Sederhana

Graf tidak sederahana adalah graf yang memiliki gelang atau rusuk ganda.



Gambar 2.4 Graf Tidak Sederhana

2) Jenis Graf Berdasarkan Keteraturan Derajat Simpulnya

Berdasarkan keteraturan derajat dari simpulnya graf dapat dibedakan menjadi 2 jenis yaitu :

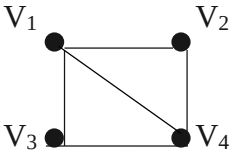
a. Graf Teratur

Graf teratur adalah graf yang setiap simpulnya berderajat sama.

Gambar 2.5. Graf Teratur

b. Graf Tidak Teratur

Graf tidak teratur adalah graf yang setiap simpulnya tidak mempunyai derajat yang sama.



Gambar 2.6 Graf Tidak Teratur

3) Jenis Graf Berdasarkan Orientasi Arah pada Sisi

a. Graf tak-berarah

Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah. Pada graf tak-berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. jadi $(u,v) = (v,u)$ adalah sisi yang sama.

b. Graf Berarah

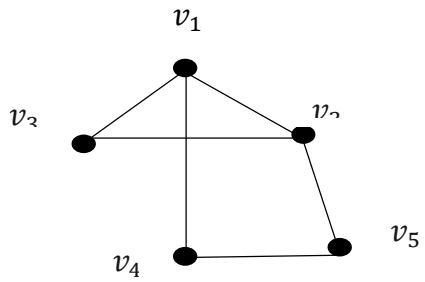
Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah. Sisi berarah dapat dinyatakan dengan sebutan busur (*arc*). Pada graf berarah, (*u,v*) dan (*v,u*) menyatakan dua buah busur yang berbeda, dengan kata lain (*u,v*) ≠ (*v,u*). Untuk busur (*u,v*), simpul *u* dinamakan simpul asal (*initial vertex*) dan simpul v dinamakan simpul terminal (*terminal vertex*) (Munir, 2009).

4) Graf Berbobot

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Bobot pada tiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak antara dua buah kota, biaya perjalanan dua buah kota, waktu tempuh pesan (*message*) dari sebuah simpul komunikasi ke simpul komunikasi lain, ongkos produksi, dan sebagainya (Munir, 2009).

3. Keterhubungan

Suatu graf *G* dikatakan terhubung (*connected*) jika untuk setiap dua simpul *u* dan *v* di *G*, terdapat lintasan yang menghubungkan simpul itu, sebaliknya graf dikatakan tidak terhubung (*disconnected)* jika tidak ada lintasan yang menghubungkannya. Jika suatu graf tidak terhubung maka graf *G* akan terdiri dari beberapa subgraf yang disebut komponen graf. Banyaknya komponen graf *G* dinotasikan dengan $\omega(G)$. Graf terhubung mempunyai satu komponen dan graf tidak terhubung mempunyai lebih dari satu komponen (Mardiyono, 1996 : 44). Misalnya ditunjukan pada Gambar 2.7: (Munir, 2009:371).



Gambar 2.7 Simpul Terhubung dari Graf N

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,…,yz ;

walk tersebut dinyatakan dengan *uv, vw, wx,…,yz* atau dengan kata lain *walk* antara *u* sampai *z* (Robin J. Wilson & John M. Aldous, 1990 : 34). Contoh *walk* pada graf N adalah ($v_{2,}v_1, v_3, v_1, v_{4,}v_{5,}v_{2,}$).

b. Jejak (*Trail)*

Jejak adalah *walk* dengan semua rusuk dalam barisan adalah berbeda (Munir, 2009 :370). Contoh *trail* pada graf N adalah ($v_3, v_2, v_{1,}v_4, v_5, v_2$).

c. Lintasan (*Path)*

Jika seluruh rusuk (tidak harus seluruh simpul) pada sebuah trayek berbeda, maka trayek tersebut disebut jejak (*trail*). Sedangkan jika simpul-simpulnya berbeda jejak tersebut disebut lintasan (Robin J. Wilson & John M. Aldous, 1990 : 35). Contoh lintasan pada Graf N adalah ($v_3, v_1, v_{2,}v_5, v_4$).

d. Sikel

Trayek tertutup pada Graf G adalah sebuah rangkaian terurut rusuk-rusuk G dalam bentuk :

Kl,lm,mn,…,pq,qk;

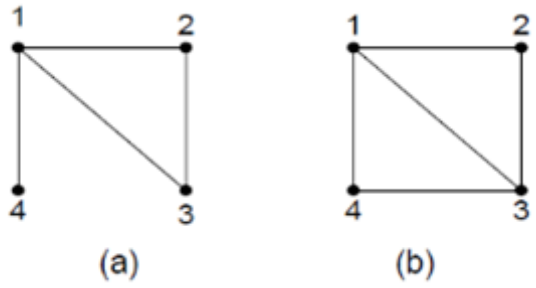
Jika seluruh rusuknya berbeda, maka trayek tersebut disebut trail tertutup. Jika rusuk *k,l,m,n,..,p,q* seluruhnya berbeda, maka trail tersebut disebut sikel (Robin J. Wilson & John M. Aldous, 1990 : 35). Contoh sikel pada Graf N adalah $(v_2, v_3, v_1, v_4, v_5, v_2)$.

e. Sirkuit dan Lintasan Hamilton

i. Sirkuit Hamilton ialah sirkuit yang melalui tiap simpul di dalam graf tepat satu kali, kecuali simpul asal (sekaligus simpul akhir) yang dilalui dua kali.

ii. Lintasan Hamilton ialah lintasan yang melalui tiap simpul di dalam graf tepat satu kali.

Contoh Sirkuit dan Lintasan Hamilton ditunjukkan pada Gambar 2.8 :



Gambar 2.8 (a). Sirkuit Hamilton, (b) Sirkuit Hamilton & Lintasan Hamilton

C. Travelling Salesman Problem (TSP)

1. Pengertian *Travelling Salesman problem* (TSP)

Travelling Salesman Problem (TSP) dikemukakan pada tahun 1800 oleh matematikawan Irlandia, Willian Rowan Hamilton dan matematikawan Inggris, Thomas Penyngton. TSP dikenal sebagai suatu permasalahan optimasi yang bersifat klasik dimana tidak ada penyelesaian yang ada. Permasalahan ini melibatkan seorang *salesman* yang harus melakukan kunjungan sekali pada semua kota dalam sebuah rute sebelum *salesman* kembali ke titik awal (depot), sehingga perjalanannya dikatakan sempurna (Era Madonna dkk, 2013).

Agus dan Wayan (2010) dalam jurnalnya menjelaskan bahwa penentuan rute perjalanan merupakan salah satu permasalahan yang sering dihadapi dalam kehidupan sehari-hari. Salah satu contoh yaitu rute manakah yang memiliki biaya paling murah untuk dilalui seorang *salesman* ketika harus mengunjungi sejumlah titik. Setiap titik tersebut harus dikunjungi tepat satu kali kemudian kembali lagi ke titik semula. Permasalahan tersebut dikenal sebagai *Travelling Salesman Problem* (TSP). Secara matematis TSP dapat diformulasikan sebagai berikut :

Didefinisikan :

$$Xij = \begin{cases} 1, \text{jika ada perjalanan } salesman \text{ dari titik i menuju titik j} \\ 0, \text{ jika tidak ada perjalanan } salesman \text{ dari titik i menuju titik j} \end{cases}$$

C_{ij} merupakan jarak dari titik *i* menuju titik *j*

Fungsi tujuan : $Z = \min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$ **(2.1)**

Dengan kendala

$\sum_{i=1}^{n} x_{ij} = 1$, untuk j=1,2,3,...,n-1 **(2.2)**

$\sum_{j=1}^{n} x_{ij} = 1$, untuk i=1,2,3,..,n-1 **(2.3)**

2. Penyelesaian *Travelling Salesman Problem* (TSP)

Permasalahan TSP dapat diselesaikan dengan beberapa cara, tergantung dengan sistem permasalahan yang dihadapi. Adapun metode atau cara yang dapat dilakukan untuk menyelesaikan berbagai macam permasalahan TSP yaitu *Djikstra, Nearest Neighbour, Insertion.* Pada penulisan skripsi ini akan digunakan metode *Nearest Neighbour* untuk menyelesaikan permasalahan distribusi gula di salah satu pabrik gula yang berada di Yogyakarta. Menurut Era Madonna dkk (2013), pada metode *Nearest Neighbour* ini, pemilihan rute akan dimulai pada rute yang memiliki nilai jarak paling minimum setiap melalui agen, kemudian akan memilih agen selanjutnya yang belum dikunjungi dan memiliki jarak yang paling minimum.

Metode *Nearest Neighbour* merupakan metode paling sederhana untuk menyelesaikan masalah *Travelling Salesman Problem*. Pertama, memilih salah satu titik yang mewakili suatu titik awal. Selanjutnya, memilih titik tujuan yang memiliki jarak terdekat dengan titik yang sebelumnya dikunjungi. Setelah seluruh titik dikunjungi atau seluruh titik telah terhubung, maka tutup rute perjalanan dengan kembali ke titik asal. Berikut ini merupakan langkah-langkah yang harus

dilakukan dalam pengerjaan pembentukan rute dengan menggunakan metode *Nearest Neighbour* (Era Madonna, 2013):

1. Langkah 0 : Inisialisasi

   1.1 menentukan satu titik yang akan menjadi titik awal (depot) perjalanan.

   1.2 menentukan C={1,2,3,4,...,n} sebagai himpunan titik yang akan dikunjungi.

   1.3 menentukan urutan rute perjalanan saat ini(sementara)(R).

2. Langkah 1 : memilih titik yang selanjutnya akan dikunjungi

   Jika n_1 adalah titik yang berada di urutan terakhir dari rute R maka akan ditemukan titik berikutnya n_2 yang memiliki jarak paling minimum dengan n_1 , dimana n_2 merupakan anggota dari C. Apabila terdapat banyak pilihan optimal artinya terdapat lebih dari satu titik yang memiliki jarak yang sama dari titik terakhir dalam rute R dan jarak tersebut merupakan jarak yang paling minimum maka pilih secara acak.

3. Langkah 2 : menambah titik yang terpilih pada langkah 1 pada urutan rute berikutnya. Menambahkan titik n_2 di urutan terakhir dari rute sementara dan mengeluarkan yang terpilih tersebut dari daftar titik yang belum dikunjungi.

4. Langkah 3 : jika semua titik yang harus dikunjungi telah dimasukkan dalam rute atau C=Ø, maka tidak ada lagi titik yang ada di C. Selanjutnya, menutup rute dengan menambahkan titik inisialisasi atau titik awal perjalanan diakhir

rute. Dengan kata lain, rute ditutup dengan kembali lagi ke titik asal. Jika sebaliknya, kembali melakukan langkah 1.

Metode *Nearest Neeighbour* digunakan pada penulisan skripsi ini dikarenakan metode ini merupakan salah satu metode yang memiliki karakteristik pembentukan rute distribusi sesuai dengan keadaan nyata yang terdapat pada kondisi di lapangan, serta alasan penggunaan metode ini dikarenakan teknik penentuan rute yang diterapkan pada metode ini lebih mudah dilakukan dibandingkan dengan metode TSP yang lain dan metode *Nearest Neighbour* ini merupakan metode yang dapat dijadikan sebagi dasar dalam pembuatan rute distribusi dengan menggunakan metode yang lainnya.

D. *Vehicle Routing Problem (*VRP)

Yeun dkk (2008) mendefinisikan VRP sebagai masalah penentuan rute optimal kendaraan dalam pendistribusian barang atau jasa dari satu atau lebih depot ke sejumlah agen di lokasi yang berbeda dengan permintaan yang telah diketahui dan memenuhi sejumlah kendala. VRP pertama kali dipelajari oleh Dantzig dan Ramser pada tahun 1959 dalam bentuk rute dan penjadwalan truk. Solusi dari sebuah VRP yaitu sejumlah rute pengiriman kebutuhan agen dimana kendaraan berangkat dari depot lalu menuju agen dan kembali lagi ke depot (Indra dkk, 2014).

Menurut Toth dan Vigo (2002) terdapat empat tujuan umum dari VRP adalah sebagai berikut :

1. Meminimumkan biaya transportasi, terkait dengan jarak dan biaya tetap yang berhubungan dengan kendaraan.
2. Meminimumkan jumlah kendaraan yang dibutuhkan untuk melayani setiap agen.
3. Menyeimbangkan rute, untuk waktu perjalanan dan muatan kendaraan.
4. Meminimumkan penalti akibat pelayanan yang kurang memuaskan dari agen.

Menurut Indra dkk (2014) VRP terbagi menjadi beberapa jenis, antara lain :

1. *Capacitated* VRP (CVRP), yaitu setiap kendaraan mempunyai kapasitas yang terbatas.
2. VRP *with time windows* (VRPTW), yaitu setiap pelanggan harus disuplai dalam jangka waktu tertentu.
3. *Multiple Depot VRP* (MDVRP), yaitu distributor memiliki banyak depot untuk menyuplai pelanggan.
4. VRP *with pick-up and delivering* (VRPPD), yaitu pelanggan mungkin mengembalikan barang pada depot asal.
5. *Periodic* VRP, yaitu pengantar hanya dilakukan dihari tertentu.

6. *VRP with heterogeneous fleet of vehicles,* yaitu kapasitas kendaraan antar kendaraan satu dengan kendaraan lain tidak selalu sama. Jumlah dan tipe kendaraan diketahui.

Pada penulisan skripsi ini akan dibahas jenis VRP yaitu *Capacitated VRP* dengan keistimewaan yaitu dapat menentukan rute dengan jarak tempuh minimum dengan penambahan kendala kapasitas kendaraan.

E. *Capacitated Vehicle Routing Problem* (CVRP)

Menurut Wijaya dkk (2004), CVRP merupakan salah satu variasi yang paling umum dari masalah VRP, dimana terdapat penambahan kendala berupa kapasitas kendaraan yang homogen (identik) untuk mengunjungi sejumlah agen sesuai dengan permintaannya masing-masing. Pada permasalahan CVRP, total jumlah permintaan agen dalam satu rute tidak melebihi kapasitas kendaraan yang melayani rute tersebut, setiap agen dikunjungi hanya satu kali oleh satu kendaraan dan semua rute dimulai dan berakhir di depot. Tujuan dari CVRP adalah meminimumkan total jarak tempuh rute perjalanan kendaraan yang digunakan dalam mendistribusikan barang dari tempat pengiriman (depot) ke masing-masing agen.

Pemodelan untuk CVRP memiliki parameter-parameter sebagai berikut :

n : adalah jumlah agen

q : menunjukkan kapasitas setiap kendaraan

d_i : menunjukkan permintaan agen i dan

c_{ij} : jarak tempuh perjalanan dari agen *i* ke agen *j*

semua parameter dianggap nilai integer tidak negatif. Sejumlah kendaraan *homogeny* dengan kapasitas q dan sebuah depot utama, dengan indeks 0, melakukan pengiriman ke agen, dengan indeks 1 sampai n. Permasalahannya adalah menentukan rute pasti pada setiap kendaraan yang dimulai dan diakhiri di depot. Setiap agen harus dipasangkan dengan rute yang tepat, karena setiap agen hanya dapat dilayani oleh satu kendaraan. Jumlah seluruh permintaan agen yang ada pada setiap rute harus berada dalam batas kapasitas kendaraan. Tujuannya adalah untuk meminimalkan total jarak tempuh perjalanan (Wijaya dkk, 2004).

Menurut Wijaya dkk (2004), model matematika dari CVRP didefinisikan sebagai suatu graf $G = (V,E)$. Himpunan V terdiri atas gabungan himpunan agen *C* dari 1 sampai n dengan penambahan depot sebagai titik 0 dan 1. Jaringan jalan yang digunakan oleh kendaraan dinyatakan sebagai himpunan rusuk berarah *E* yaitu penghubung antar agen, $E = \{(i,j) | i,j \in v, i \neq j\}$. Semua rute dimulai dari 0 dan berakhir di 0. Himpunan kendaraan *k* merupakan kumpulan kendaraan yang homogen dengan kapasitas *q*. Setiap agen i untuk setiap $i \in C$ memiliki permintaan d_i sehingga panjang rute dibatasi oleh kapasitas kendaraan. Setiap rusuk $(i,j) \in E$ memiliki jarak tempuh c_{ij} dan jarak tempuh diasumsikan simetris, contoh $c_{ij} = c_{ji}$, dan juga bahwa $c_{ii} = c_{jj} = 0$. Satu-satunya variabel keputusan adalah x_{ijk} :

$$X_{ijk} = \begin{cases} 1, \text{jika terdapat perjalanan dari } i \text{ ke } j \text{ dengan kendaraan } k \\ 0, j\text{ika tidak terdapat perjalanan dari } i \text{ ke } j \text{ dengan kendaraan } k \end{cases}$$

Fungsi tujuan dari model matematika CVRP adalah

Meminimumkan Z = $\sum_{k\in k} \sum_{i\in v} \sum_{j\in v} c_{ij} x_{ijk}$ **(2.4)**

Dengan kendala,

1. Setiap titik dikunjungi tepat satu kali oleh suatu kendaraan :

$$\sum_{k\in k} \sum_{j\in v} x_{ijk} = 1, \forall i \in \text{V} \quad \textbf{(2.5)}$$

2. Total permintaan semua titik dalam satu rute tidak melebihi kapasitas kendaraan :

$$\sum_{i\in C} d_i \sum_{j\in V} x_{ijk} \leq q, \forall k \in K \quad \textbf{(2.6)}$$

3. Setiap rute berawal dari depot 0 :

$$\sum_{j\in C} x_{ojk} = 1, \forall k \in K \quad \textbf{(2.7)}$$

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

$$\sum_{i\in C} x_{ijk} - \sum_{j\in V} x_{ijk} = 0, \forall k \in K \quad \textbf{(2.8)}$$

5. Setiap rute berakhir di depot :

$$\sum_{i\in C} x_{ijk} = 1, \forall k \in K \quad \textbf{(2.9)}$$

6. Variabel x_{ijk} merupakan variabel biner :

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

Berdasarkan definisi CVRP, diperoleh suatu kesimpulan mengenai input dari permasalahan CVRP sebagai berikut :

1. Input permasalahan CVRP adalah daftar jarak agen, daftar permintaan tiap agen dan kapasitas kendaraan.
2. Dalam terminologi graf, kumpulan agen atau titik pada permasalahan CVRP adalah sebuah graf lengkap dengan bobot rusuk adalah jarak antar agen.

F. Algoritma Sweep

Algoritma Sweep adalah algoritma dua tahap, yaitu tahap pertama terdiri dari *clustering* agen yang mana *clustering* awal dilakukan dengan menggabungkan titik-titik dalam satu *cluster* berdasarkan kapasitas maksimal kendaraan, dan tahap kedua adalah membentuk rute-rute untuk masing-masing *cluster*. Berikut ini merupakan langkah-langkah yang harus dilakukan dalam menyelesaikan permasalahan CVRP dengan menggunakan algoritma sweep (Arunya Boonkleaw etc, 2009) :

1. Tahap pengelompokan (*Clustering*)

Tahap pertama dalam algoritma sweep adalah mengelompokan masing-masing titik agen ke dalam sebuah *cluster*. Berikut ini adalah langkah-langkah dalam pengelompokan :

a. Menggambar masing-masing agen (yang selanjutnya disebut sebagai titik) dalam koordinat kartesius dan menetapkan titik depot sebagai pusat koordinat.

b. Menentukan semua koordinat polar dari masing-masing titik yang berhubungan dengan depot. Langkah untuk mengubah koordinat kartesius (x,y) menjadi koordinat polar (r,θ) adalah sebagai berikut :

$$r = \sqrt{x^2 + y^2} \qquad (2.11)$$

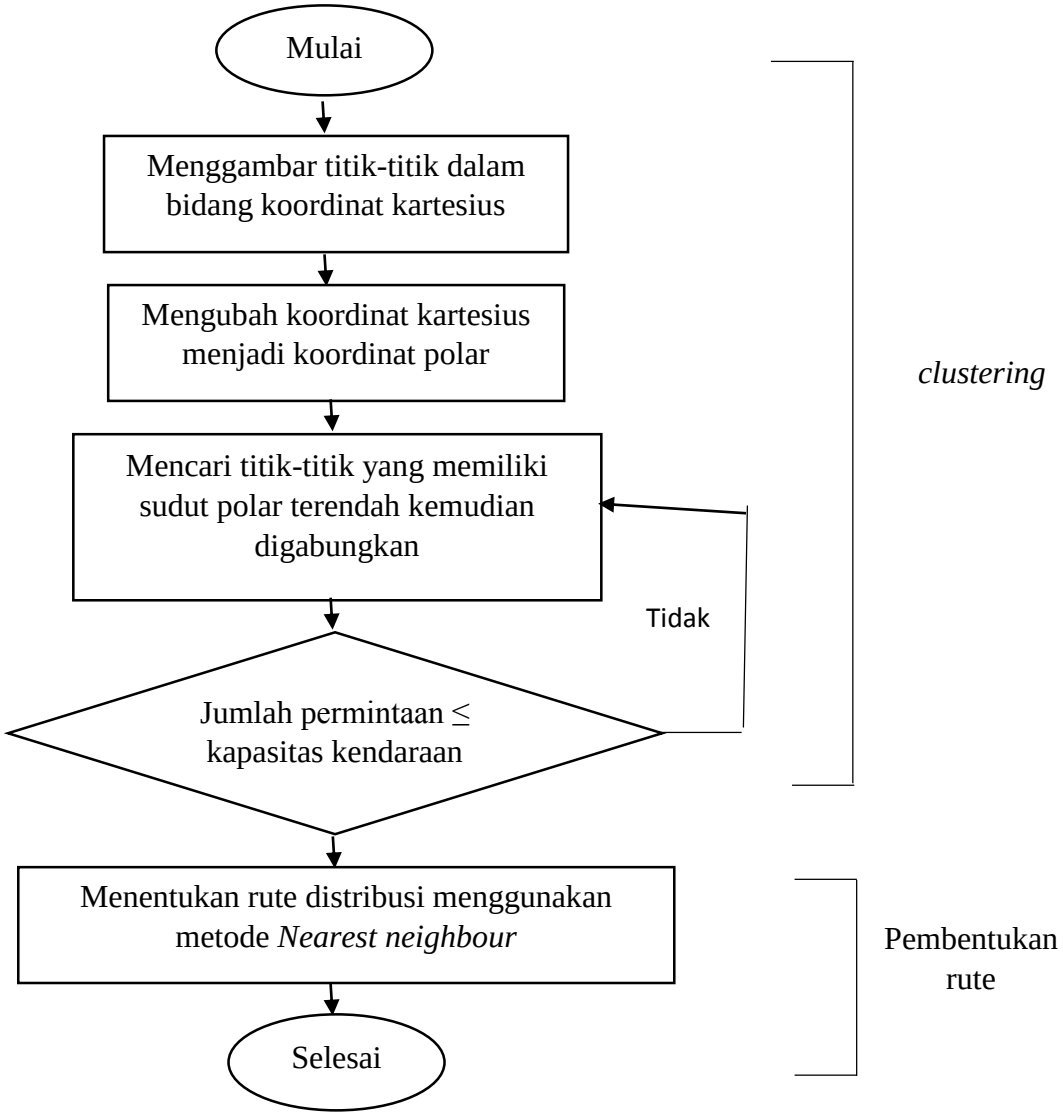
$$\theta = arc\tan\frac{y}{x} \qquad (2.12)$$

Setelah diperoleh masing-masing titik dalam koordinat polar maka selanjutnya menggambarkan titik-titik tersebut dalam bidang dua dimensi. Langkah untuk mengubah koordinat kartesius menjadi koordinat polar juga dapat dilakukan dengan menggunakan S*oftware Geogebra.*

c. Melakukan pengelompokan (*clustering)* dimulai dari titik yang memiliki sudut polar terkecil dan seterusnya berurutan sampai titik yang memiliki sudut polar terbesar dengan memperhatikan kapasitas kendaraan.
d. Memastikan semua titik tersapu dalam *cluster* saat ini.
e. Pengelompokan dihentikan ketika dalam satu *cluster* akan melebihi kapasitas maksimal kendaraan.
f. Membuat *cluster* baru dengan langkah yang sama seperti langkah c dimulai dari titik yang memiliki sudut polar terkecil yang belum termasuk dalam *cluster* sebelumnya (titik yang terakhir ditinggalkan).
g. Mengulangi langkah c-f, sampai semua titik telah dimasukkan dalam sebuah *cluster*.

2. Tahap pembentukan rute

Tahap kedua dalam algoritma sweep yaitu membentuk rute-rute berdasarkan *cluster* yang telah diperoleh pada tahapan *clustering*. Setiap *cluster* akan menjadi sebuah permasalahan *Travelling Salesman Problem* (TSP). Oleh karena itu, dalam menyelesaikan tahapan pembentukan rute, langkah yang digunakan sama dengan ketika menyelesaikan permasalahan TSP. Pada penulisan skripsi ini, metode yang digunakan untuk menyelesaikan permasalahan TSP adalah metode *Nearest Neighbour*. Diagram alir dari Algoritma *Sweep* ditunjukkan pada Gambar 2.9



Gambar 2.9 Diagram Alir Algoritma *Sweep*

G. Algoritma Genetika

1. Definisi Algoritma Genetika (Kusumadewi, 2003 : 87)

Algoritma Genetika merupakan suatu metode algoritma pencarian berdasarkan pada mekanisme seleksi alam dan genetik alam. Algoritma genetika

pertama kali diperkenalkan oleh John Holland dalam bukunya yang berjudul "*Adaption in natural and artificial systems*", dan oleh De Jong dalam bukunya "*Adaption of the behavior of a class of genetic adaptive systems*", yang keduanya diterbitkan pada tahun 1975 yang merupakan dasar dari algoritma genetika (Davis, 1991). Tujuan Holland mengembangkan algoritma saat itu bukan untuk mendesain suatu algoritma yang dapat memecahkan suatu masalah, namun lebih mengarah ke studi mengenai fenomena adaptasi di alam dan mencoba menerapkan mekanisme adaptasi alam tersebut ke dalam sistem komputer.

Kemunculan Algoritma Genetika diinspirasikan dari teori-teori dalam ilmu biologi, sehingga banyak istilah dan konsep biologi yang digunakan dalam Algoritma Genetika. Sesuai dengan namanya, proses-proses yang terjadi dalam Algoritma Genetika sama dengan apa yang terjadi pada evolusi biologi yaitu seleksi, pindah silang, dan mutasi.

Algoritma Genetika merupakan teknik pencarian yang didasarkan atas mekanisme seleksi dan genetik natural. Algoritma genetika berbeda dengan teknik pencarian konvensional, algoritma genetika dimulai dari himpunan solusi yang pada umumnya dihasilkan secara acak. Himpunan ini disebut populasi, sedangkan setiap individu dalam populasi disebut kromosom (merupakan representasi dari solusi) dan yang menempati kromosom disebut gen dan nilainya dapat berupa bilangan numerik, bilangan biner, symbol ataupun sebuah karakter dari permasalahan yang ingin diselesaikan (Gen, 2000 : 1).

Kromosom-kromosom berevolusi dalam suatu proses iterasi yang berkelanjutan yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan nilai fungsi *fitness* (Gen dan Cheng, 2000). Nilai *fitness* adalah nilai yang menunjukkan nilai ketangguhan kromosom dalam beradaptasi terhadap masalah. Untuk menghasilkan generasi berikutnya (*offspring*) didapatkan dari perkawinan silang (*crossover*) atau memodifikasi kromosom menggunakan operator mutasi (*mutation*) dengan harapan akan menghasilkan kromosom baru dengan tingkat *fitness* yang lebih tinggi sebagai generasi baru atau keturunan (*offspring*) berikutnya. setelah beberapa generasi maka algoritma genetika akan konvergen pada kromosom terbaik, yang diharapkan merupakan solusi optimal (Goldberg, 1989 : 71).

Hal-hal yang terdapat dalam algoritma genetika adalah sebagai berikut (Satriyanto, 2009).

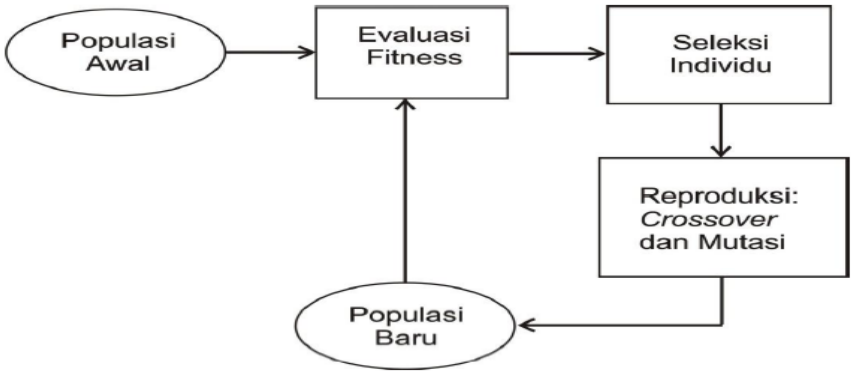
a. Gen (*Genotype*) adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom.
b. *Allele* yaitu nilai dari sebuah gen, dapat berupa bilangan biner, float, integer, karakter dan kombinatorial.
c. Kromosom adalah gabungan gen – gen yang membentuk nilai tertentu.
d. Individu merupakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.

e. Populasi merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi. Populasi terdiri dari sekumpulan kromosom.
f. Induk, adalah kromosom yang akan dikenai operasi genetik pindah silang
g. Pindah silang merupakan operasi genetik yang mewakili proses perkembangbiakan antar individu.
h. *Offspring* adalah kromosom yang merupakan hasil dari operasi genetik pindah silang dikenal keturunan atau sebagai anak.
i. Mutasi merupakan operasi genetik yang mewakili proses mutasi dalam perjalanan hidup individu. Mutasi berperan menghasilkan perubahan acak dalam populasi, yang berguna untuk menambah variasi dari kromosom – kromosom dalam sebuah populasi.
j. Proses Seleksi merupakan proses yang mewakili proses seleksi alam (*natural selection*) dari teori Darwin. Proses ini dilakukan untuk menentukan induk dari operasi genetik pindah silang yang akan dilakukan untuk menghasilkan keturunan (*offspring*).
k. Nilai *fitness* merupakan penilaian yang menentukan bagus tidaknya sebuah kromosom.
l. Fungsi Evaluasi adalah fungsi yang digunakan untuk menentukan nilai *fitness*. Fungsi evaluasi ini merupakan sekumpulan kriteria-kriteria tertentu dari permasalahan yang ingin diselesaikan.
m. Generasi merupakan satuan dari populasi setelah mengalami operasi-operasi genetika, berkembang biak, dan menghasilkan keturunan. Pada akhir dari

setiap generasi, untuk menjaga agar jumlah kromosom dalam populasi tetap konstan, kromosom–kromosom yang mempunyai Nilai *fitness* yang rendah dan memiliki peringkat dibawah nilai minimal akan dihapus dari populasi.

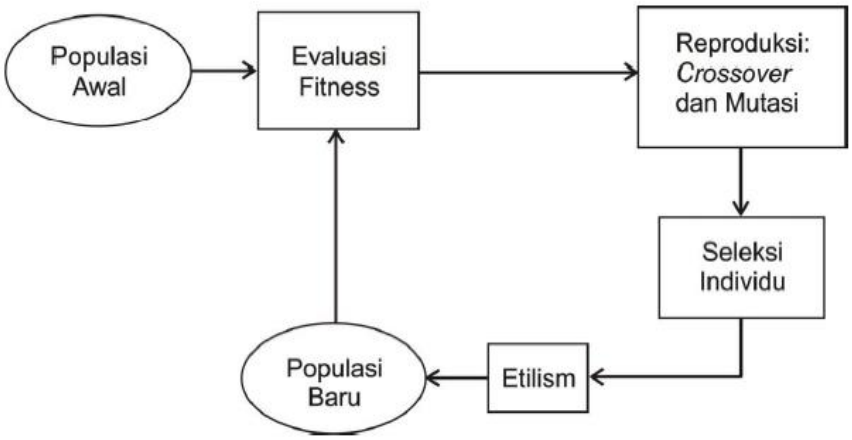
2. Skema Alur Algoritma Genetika

Skema algoritma genetika pertama kali dikemukakan oleh David Goldberg (1989), dengan skema tersebut dapat dilihat pada Gambar 2.10 :



Gambar 2.10 Skema Algoritma Genetika oleh David Goldberg (1989)

Siklus ini kemudian diperbaiki oleh beberapa ilmuwan yang mengembangkan algoritma genetika, yaitu Zbignew Michalewichz dengan menambahkan operator *elitism* dan membalik proses seleksi setelah proses reproduksi.



Gambar 2.11 Skema Algoritma Genetika oleh Michalewichz (1996)

3. Komponen Algoritma Genetika

a. Penyandian Permasalahan (Pengkodean)

Teknik penyandian adalah proses penyandian gen dari kromosom. Gen merupakan bagian dari kromosom, satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk bit, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan dalam operator genetika (Satriyanto, 2009).

Terdapat beberapa teknik pengkodean dalam algoritma genetika diantaranya pengkodean biner, pengkodean permutasi, pengkodean nilai dan pengkodean pohon (Lukas, Anwar dan Yuliani, 2005). Pada penelitian ini, representasi gen menggunakan teknik pengkodean permutasi. Dalam pengkodean ini, tiap gen dalam kromosom merepresentasikan suatu urutan (Anwar dan Yuliani, 2005).

Contoh 2.3 kromosom 1 = 2 3 4 5 1 6 7

Keterangan: kromosom 1 berisi urutan secara acak gen kesatu sampai ke tujuh. Gen direpresentasikan dengan sebuah bilangan dan bilangan-bilangan tersebut representasi dari masing-masing kota.

b. Membangitkan Populasi Awal (*Spanning*)

Membangkitkan populasi awal adalah membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada (Kusumadewi, 2003: 282).

Terdapat berbagai teknik dalam pembangkitan populasi awal diantaranya adalah random generator, pendekatan tertentu, dan permutasi gen. Pada penelitian ini pembangkitan populasi awal dilakukan dengan menggunakan random generator. Inti dari cara ini adalah melibatkan pembangkitan bilangan random dalam interval (0,1) untuk setiap nilai gen sesuai dengan representasi kromosom yang digunakan.

c. Evaluasi Nilai *Fitness (Fitness Value)*

Evaluasi nilai *fitness* berfungsi untuk mengukur kualitas dari sebuah solusi dan memungkinkan tiap solusi untuk dibandingkan (Michalewicz, 1996: 72). Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran baik

tidaknya individu tersebut. Di dalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup, sedangkan individu yang bernilai *fitness* rendah akan mati (Goldberg, 1989). Pada masalah optimasi, fungsi *fitness* yang yang digunakan adalah

$$f = \frac{1}{x}, \qquad \textbf{(2.13)}$$

dengan *x* merupakan nilai dari individu, yang artinya semakin kecil nilai *x*, maka semakin besar nilai *fitness*nya. Tetapi hal ini akan menjadi masalah jika *x* bernilai 0, yang mengakibatkan *f* bisa bernilai tak hingga. Untuk mengatasinya, *x* perlu ditambah sebuah bilangan sangat kecil sehingga nilai *fitness*nya menjadi

$$f = \frac{1}{(x+a)}, \qquad \textbf{(2.14)}$$

dengan *a* adalah bilangan yang dianggap sangat kecil (konstanta) dan bervariasi sesuai dengan masalah yang akan diselesaikan (Suyanto, 2005:10).

d. Seleksi (*Selection)*

Seleksi merupakan pemilihan dua buah kromosom untuk dijadikan sebagai induk yang dilakukan secara proporsional sesuai dengan dengan nilai *fitness*-nya (Michalewicz, 1996: 75). Masing-masing individu yang diseleksi akan diberikan probabilitas reproduksi tergantung dari nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya.

Terdapat beberapa metode seleksi menurut Kusumadewi (2003:105), yaitu *rank-based fitness assignment, roulette wheel selection*, s*tochastic universal sampling*, seleksi lokal (l*ocal selection*), seleksi dengan pemotongan (*truncation selection*) dan seleksi dengan turnamen (*tournament selection*).Skripsi ini menggunakan metode *Roulette Wheel Selection*.

Cara kerja metode *Roulette Wheel Selection* adalah sebagai berikut :

1. Dihitung nilai *fitness* dari masing-masing individu (f_i, dimana *i* adalah individu ke-1 s/d ke-n).
2. Dihitung total *fitness* semua individu.
3. Dihitung probabilitas masing-masing individu.
4. Dari probabilitas tersebut, dihitung jatah masing-masing individu pada angka 1 sampai 100.
5. Dibangkitkan bilangan random antara 1 sampai 100.
6. Dari bilangan random yang dihasilkan, ditentukan individu mana yang terpilih dalam proses seleksi.

Urutan langkah proses seleksi menggunakan metode *Roulette Wheel* adalah sebagai berikut

a. Misalkan diberikan contoh populasi beserta nilai *fitness*nya.
b. Menentukan nilai segmen unutk masing-masing individu.
c. Dibangkitkan bilangan random antara 1-100 sebanyak 5 kali.

d. Memutar roda *Roulette* sebanyak individu tersebut. Setiap kali putaran akan menghasilkan suatu bilangan random dengan rentang antara 1-100 yang menunjukkan daerah atau segmen dari individu.

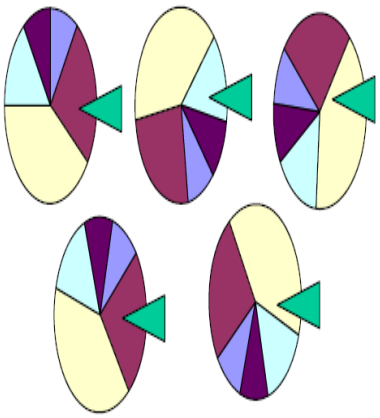
e. Terpilih hasil individu yang setelah roda *Roulette* diputar sebanyak 5 kali.

Individu 1: fitness =10%
Individu 2: fitness =25%
Individu 3: fitness =40%
Individu 4: fitness =15%
Individu 5: fitness =10%

Jatah untuk individu 1=1-10
Jatah untuk individu 2=11-35
Jatah untuk individu 3=36-75
Jatah untuk individu 4=76-90
Jatah untuk individu 5=91=100

Individu Terpilih

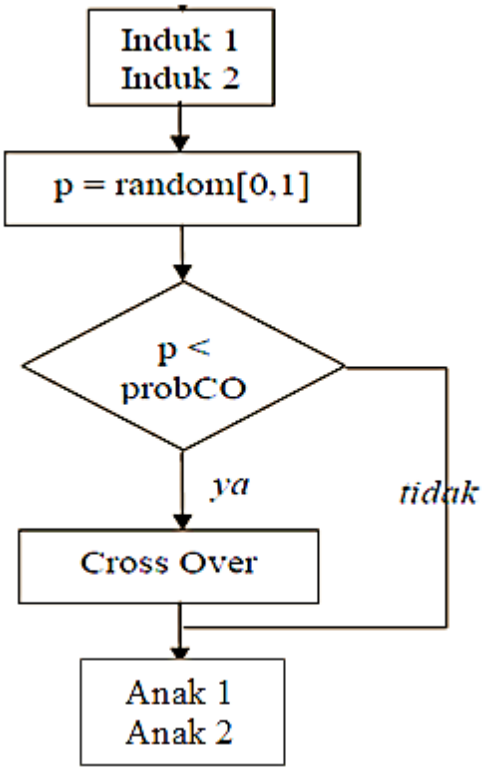
Random 30= Individu 2
Random 88= Individu 4
Random 64= Individu 3
Random 18= Individu 2
Random 44= Individu 3



e. Pindah Silang (*Crossover)*

Pindah Silang (*crossover*) adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan keturunan baru dalam ruang pencarian yang siap diuji. Operasi ini tidak selalu dilakukan pada setiap individu yang ada. Individu dipilih secara acak untuk dilakukan *crossover* dengan Pc (*Probabilitas Crossover*) antara 0,6 s/d 0,95. Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunan (Michalewicz, 1996: 78).

Prinsip dari pindah silang ini adalah melakukan operasi (pertukaran aritmatika) pada gen yang bersesuaian dari dua induk untuk menghasilkan individu baru. Proses *crossover* dilakukan pada setiap individu dengan probabilitas *crossover* yang ditentukan. Secara skematis proses *crossover* dapat digambarkan seperti Gambar 2.12



Gambar 2.12 Sistematika Proses *Cross Over*

Dari Gambar 2.12, jika bilangan p yang dibangkitkan secara acak kurang dari probabilitas *crossover* (probCO), maka kedua induk dilakukan operasi pindah silang (*crossover*). Tetapi jika bilangan p yang dibangkitkan lebih dari atau sama dengan probCO, maka tidak dilakukan operasi pindah silang.

Teknik *crossover* yang digunakan adalah teknik *Order Crossover* (OX) yang diperkenalkan oleh Davis (Tanjung, 2010).

Prosedur O*X*

Teknik ini diawali dengan membangkitkan dua bilangan acak. Kemudian gen yang berada diantara kedua bilangan acak tersebut (*substring*) disalin ke keketurunan (*offspring*) dengan posisi yang sama pada masing-masing kromosom orang tua. Mengurutkan gen yang berada dalam kromosom orang tua kedua dengan urutan gen yang pertama adalah dari gen yang berada pada posisi setelah bilangan acak kedua yang telah dibangkitkan sebelumnya lalu diikuti oleh gen-gen yang berada pada posisi sebelum bilangan acak pertama dan diakhiri dengan gen-gen yang berada pada posisi diantara kedua bilangan acak.

Gen yang telah diurutkan tadi dibandingkan dengan *offspring* pertama. Apabila gen yang terurut tersebut mengandung gen yang berada pada *offspring* pertama, maka abaikan gen tersebut dari urutan. Masukkan urutan yang baru saja didapat ke *offspring* pertama dengan cara memasukkan urutan gen pada posisi setelah bilangan acak kedua terlebih dahulu pada *offspring* pertama dan sisanya dimasukkan pada posisi sebelum bilangan acak pertama. Untuk menghasilkan *offspring* kedua dilakukan cara yang sama untuk kromosom orang tua pertama.

Gambar 2.13 adalah ilustrasi dari OX

Order Crossover dapat diselesaikan dengan langkah-langkah berikut :

1. Membangkitkan dua bilangan acak.

Induk 1	1	2	3	4	5	6	7	8	9
Induk 2	4	5	2	1	8	7	6	9	3

2. Dibangkitkan 2 bilangan acak sebelum gen induk-1 dan setelah gen induk 1. Hal yang sama juga dilakukan untuk induk-2. Didapatkan keturunan dengan gen yang sama.

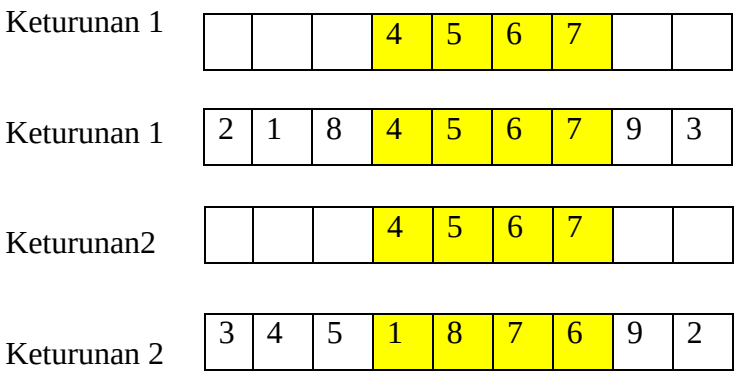
Keturunan 1				4	5	6	7		
Keturunan 2				1	8	7	6		

3. Mengurutkan gen yang berada pada induk kedua dengan urutan gen yang berada pada posisi setelah bilangan acak kedua diikuti dengan gen yang berada pada posisi sebelum bilangan acak pertama dan diakhiri dengan gen yang berada pada posisi diantara kedua bilangan acak.

9	3	4	5	2	1	8	7	6

4. Kemudian gen yang telah diurutkan tersebut dibandingkan dengan keturunan pertama. Apabila gen tersebut ada pada keturunan kedua maka abaikan gen tersebut dari urutan itu.
5. Kemudian masukkan urutan yang baru saja didapat pada keturunan dengan cara memasukkan urutan gen pada posisi setelah bilangan acak kedua terlebih dahulu

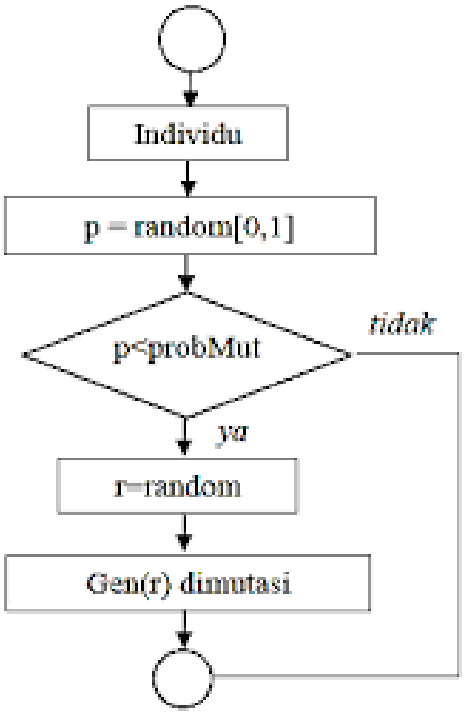
dan sisanya dimasukkan pada posisi sebelum bilangan acak pertama. Begitu juga untuk menghasilkan keturunan kedua.



Gambar 2.13 Ilustrasi OX Operator

f. Mutasi *(Mutation)*

Mutasi merupakan proses untuk mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi mutasi yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan. Akan tetapi, untuk mencapai hal ini, penekanan *selektif* juga memegang peranan yang penting. Jika dalam proses pemilihan kromosom-kromosom cenderung terus pada kromosom yang memiliki *fitness* yang tinggi saja, konvergensi prematur akan sangat mudah terjadi (Murniati, 2009: 24). Gambar 2.14 adalah alur skematis proses mutasi.



Gambar 2.14 Sistematika Proses Mutasi

Dari Gambar 2.14, jika p merupakan bilangan random yang dibangkitkan kurang dari probabilitas mutasi (ProbMut) maka individu hasil *crossover* dilakukan proses mutasi. Sedangkan jika bilangan p yang dibangkitkan lebih dari atau sama dengan probMut, maka individu hasil *crossover* tidak dilakukan proses mutasi. Teknik *swapping mutation* diawali dengan memilih dua bilangan acak kemudian gen yang berada pada posisi bilangan acak pertama ditukar dengan gen yang berada pada bilangan acak kedua dalam probabilitas tertentu (Suyanto, 2005: 57).

Contoh

Sebelum

1	2	3	4	5	6	8	9	7

Sesudah

1	8	3	4	5	6	2	9	7

g. Elitism

Elitism merupakan proses untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi (Kusumadewi, 2003: 112). Proses seleksi dilakukan secara random sehingga tidak ada jaminan bahwa suatu indvidu yang bernilai fitness tertinggi akan selalu terpilih. Walaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness*-nya menurun) karena proses pindah silang. Oleh karena itu, untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi, maka perlu dibuat satu atau lebih. Proses *Elitism* dilakukan dengan menduplikat individu dengan nilai fitness terbaik untuk dijadikan individu pertama pada generasi berikutnya.

h. Pembentukan Populasi Baru

Proses membangkitkan populasi baru bertujuan untuk membentuk populasi baru yang berbeda dengan populasi awal. Pembentukan populasi baru ini didasarkan pada keturunan-keturunan baru hasil mutasi ditambah dengan individu terbaik setelah dipertahankan dengan proses *elitism*. Setelah populasi baru

terbentuk, kemudian mengulangi langkah-langkah evaluasi nilai fitness, proses seleksi dengan *truncation selection*, proses pindah silang, proses mutasi pada populasi baru untuk membentuk populasi baru selanjutnya.

H. Penelitian yang Relevan

Beberapa penelitian tentang CVRP dan algoritma genetika telah banyak dilakukan antara lain "Algoritma Genetika pada Penyelesaian *Capacitated Vehicle Routing Problem* (Optimasi Rute Pendistribusian Aqua Galon PT Tirta Investama" oleh Adam Arif, penelitian ini menggunakan algoritma genetika dalam menyelesaikan CVRP dengan menggunakan data dari peneliti sebelumnya yang menggunakan algoritma *Branch and Bound.* Selain penelitian yang menggunakan algoritma genetika sebagai metode penyelesaiannya terdapat juga penelitian tentang CVRP dengan metode heuristik yaitu Algoritma *Sweep* sebagai metode penyelesaiannya oleh Wahyu Kartika Cahyaningsih yang berjudul "Penyelesaian *Capacitated Vehicle Routing Problem* (CVRP) menggunakan algoritma *sweep* untuk optimasi rute distribusi surat kabar Kedaulatan Rakyat". Hasil dari penelitian ini adalah bahwa algoritma *sweep* menghasilkan total jarak tempuh kendaraan dan waktu tempuh yang lebih baik daripada total jarak tempuh kendaraan dan waktu tempuh perusahaan saat ini yaitu 52 dengan jarak 142,9 km dan waktu tempuh 210 menit sedangkan milik perusahaan yaitu dengan jarak 174,9 km dan waktu tempuh 233 menit. Menghasilkan penghematan jarak tempuh sebesar 18,29 %.

Ada persamaan dan perbedaan penelitian ini dengan penelitian sebelumnya. Persamaan pada skripsi Adam Arif dan Wahyu Kartika Cahyaningsih yaitu menggunakan algoritma genetik dan algoritma *Sweep*.Perbedaannya yaitu dalam metode seleksi yang digunakan, dalam penelitian Adam Arif menggunakan metode seleksi *Rangking Selection* sedangkan pada penelitian ini menggunakan Seleksi *Roulette Whell Selection* dan perbedaan dalam menentukan nilai paramater yang digunakan dalam algoritma genetika seperti Ukuran Populasi, Jumlah Generasi, Probabilitas Mutasi dan Probabilitas *Crossover*. Dalam skripsi ini akan dibandingkan antara Algoritma Genetika dan Algoritma *Sweep*. Data yang digunakan adalah data primer dari hasil penelitian yang dilakukan di salah satu Pabrik Gula di Yogyakarta.