BAB II

KAJIAN PUSTAKA

Pengertian-pengertian dasar yang digunakan sebagai landasan pembahasan pada Bab II yaitu masalah ditribusi, graf, *Travelling Salesman Problem* (TSP), *Vehicle Routing Problem* (VRP), *Capacitated Vehicle Routing Problem* (CVRP), Algoritma *Clarke and Wright Savings*, dan Algoritma Genetika.

2.1 Masalah Distribusi

Distribusi menurut Tjiptono (2008) dapat diartikan sebagai kegiatan pemasaran yang berusaha memperlancar dan mempermudah penyampaian barang dan jasa dari produsen kepada konsumen, sehingga penggunaannya sesuai dengan yang diperlukan (jenis, jumlah, harga, dan tempat). Setelah produk dihasilkan oleh produsen, produk tersebut dikirimkan ke suatu distributor. Ditributor kemudian menjual produk tersebut ke pengecer atau konsumen. Kendala yang sering dihadapi oleh suatu perusahaan yaitu cara penditribusian ke konsumen.

Distribusi sangat dibutuhkan oleh konsumen untuk memperoleh barang-barang yang dihasilkan oleh produsen. Kegiatan distribusi ini sangat diperlukan untuk mencukupi ketergantungan antara produsen yang membutuhkan konsumen dan konsumen yang membutuhkan produsen. Kegiatan distribusi diperlukan untuk mempermudah pemenuhan kebutuhan antara produsen dan konsumen.

2.2 Teori Graf

Dalam teori graf, suatu graf dapat didefinisikan sebagai kumpulan dari simpul (verteks) dengan beberapa pasang (tidak semua harus berpasangan) dari simpul tersebut terhubung oleh edges (rusuk). Suatu graf G terdiri dari himpunan tak kosong V(G) yang merupakan himpunan simpul (*verteks*) graf G dan E(G) yang merupakan himpunan rusuk (*edges*) graf G (Lovaks,dkk, 2010:126).

Pada teori graf diberikan model matematika untuk setiap himpunan dari sejumlah obyek diskrit, dimana beberapa pasangan unsur dari himpunan tersebut terikat menurut suatu aturan tertentu. Menurut Slamet & Makaliwe (1991:162)

obyek dapat berupa suatu himpunan nama kota dengan aturan jalan yang menghubungkan antara kota satu ke kota yang lain.

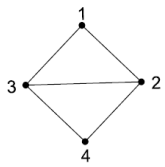
2.2.1 Macam - Macam Graf

Menurut Lovasz,dkk (2010) ada beberapa macam graf yaitu graf sederhana, graf ganda, graf kosong, graf planar, graf lengkap, dan graf berbobot.

1. Graf Sederhana

Graf sederhana adalah graf yang sepasang simpulnya dihubungkan dengan paling banyak satu ujung dan tidak ada simpul yang terhubung dengan dirinya sendiri.

Contoh 2.1:

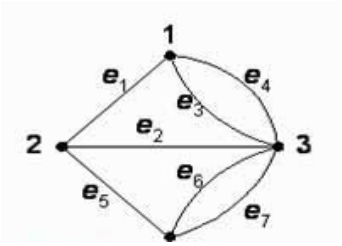


Gambar 2.1 Graf Sederhana

2. Graf Ganda (multigraph)

Graf ganda adalah graf yang mengandung gelang (loop)

Contoh 2.2:

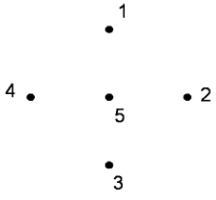


Gambar 2.2 ditunjukan dengan loop pada e_3, e_4, e_6 dan e_7

3. Graf Nol (Graf Kosong)

Graf nol adalah suatu graf yang himpunan rusuknya merupakan himpunan kosong. Dengan kata lain graf nol adalah graf yang tidak mempunyai rusuk.

Contoh 2.3:

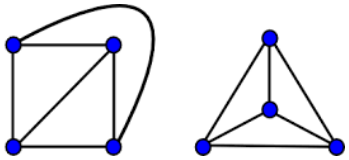


Gambar 2.3 Contoh graf nol dengan 5 titik

4. Graf Planar

Sebuah graf G disebut planar, jika G dapat digambarkan pada bidang datar sedemikian sehingga rusuk-rusuknya tidak ada yang saling berpotongan kecuali pada simpul-simpul akhir dari rusuk-rusuk tersebut.

Contoh 2.4:

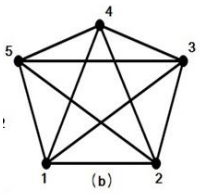


Gambar 2.4 Graf planar yang rusuk-rusuknya tidak ada yang berpotongan

5. Graf Lengkap (Graf Komplit)

Graf sederhana G disebut graf lengkap atau graf komplit jika graf tersebut tanpa loop atau rusuk ganda dan dimana setiap simpul bergabung ke setiap simpul yang lain. Graf lengkap dengan *n* simpul dinotasikan dengan K_n (Ros &Wright, 2003).

Contoh 2.5:

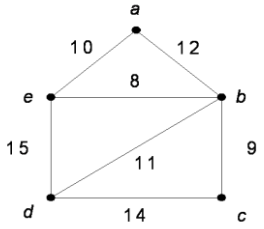


Gambar 2.5 Graf K5

6. Graf Berbobot (Weighted Graph)

Graf berbobot adalah graf yang setiap sisinya mempunyai bobot. Bobot dari tiap rusuk berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Jarak antara dua kota, biaya perjalanan antara dua buah kota, waktu tempuh pesan dari sebuah simpul komunikasi ke simpul komunikasi lain (dalam jaringan komputer), ongkos produksi dapat dinyatakan dengan bobot (Munir, 2009).

Contoh 2.6:

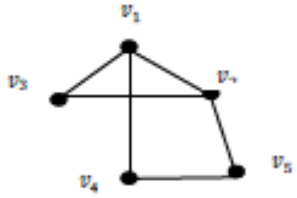


Gambar 2.6 Graf yang mempunyai bobot

Dari contoh 2.6 terlihat untuk bobot dari rusuk a-e adalah 10, rusuk a-b adalah 12, b-e adalah 8, e-d adalah 15, d-b adalah 11, d-c adalah 14, b-c adalah 9.

2.2.2 Keterhubungan

Suatu graf G dikatakan terhubung jika untuk setiap dua simpul *u* dan *v* di G, terdapat lintasan yang menghubungkan simpul itu, sebaliknya graf dikatakan tidak terhubung 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). Contoh ditunjukan pada gambar berikut (Munir, 2010:371):



Gambar 2.7 Simpul Terhubung dari Graf N

1. 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* (Wilson & Watkin, 1990 : 34). Contoh *walk* pada graf N adalah ($v2,v1,v3,v1,v4,v5,v2$,).

2. Jejak (*Trail)*

   Jejak adalah *walk* dengan semua rusuk dalam barisan adalah berbeda (Munir, 2010 :370). Contoh *trail* pada graf N adalah ($v3,v2,v1,v4,v5,v2$).

3. 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 (Wilson & Watkin, 1990 : 35). Contoh lintasan pada Graf N adalah ($v3$,$v1$,$v2$,$v5$,$v4$).

4. 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 (Wilson & Watkin, 1990 : 35). Contoh sikel pada Graf N adalah ($v2$,$v3$,$v1$,$v4$,$v5$,$v2$).

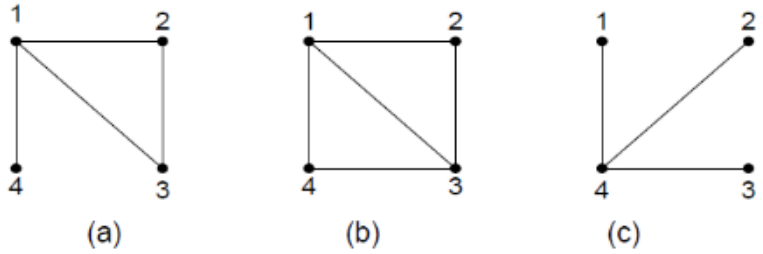
5. Sirkuit dan Lintasan Hamilton

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

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

      Contoh :

      1) Graf yang memiliki lintasan Hamilton ialah (3,2,1,4)
      2) Graf yang memiliki sirkuit Hamilton ialah (1,2,3,4,1)
      3) Graf yang tidak memiliki lintasan dan sirkuit Hamilton



Gambar 2.8 Graf Hamilton

2.3 *Travelling Salesman Problem* (TSP)

TSP adalah sebuah persoalan optimasi untuk mencari rute terpendek bagi seorang *traveling salesman*. TSP digunakan untuk menentukan urutan dari sejumlah kota yang harus dilalui oleh *salesman* dan hanya boleh dilalui satu kali dalam perjalanannya. Persoalan optimasi yang ingin dicapai adalah rute yang dilalui dan biaya yang digunakan minimum. Persoalan TSP dapat digambarkan dalam

bentuk graf perjalanan, missal dari depot menuju ke konsumen 1 kemudian ke konsumen 2 dan balik lagi ke depot.

Ada beberapa macam TSP sesuai dengan kendala, antara lain *Travelling Salesman Problem with Times Windows* (TSPTW) dan *M-Travelling Salesman Problem*. *Travelling Salesman Problem with Times Windows* (TSPTW) sebagai permasalahan untuk mencari biaya minimal dari sekumpulan konsumen, dimana tiap konsumen hanya dikunjungi sekali. Dapat dituliskan berawal dan berakhir disuatu depot, dalam batas waktu (*time window*) tertentu, dan tiap konsumen harus dikunjungi pada batas waktu (*time window)* mereka masing-masing. *M-Travelling Salesman Problem* yaitu TSP dengan kendala jumlah salesman (Dumas,dkk, 1995:367).

2.4 *Vehicle Routing Problem* (VRP)

Masalah VRP adalah permasalahan mencari rute dengan biaya minimal dari konsumen satu ke konsumen yang lain yang dimulai dari depot dan kembali ke depot. VRP adalah permasalahan untuk menentukan rute kendaraan dengan kendala seperti: 1) setiap rute dimulai dan berakhir di depot, 2) setiap konsumen dikunjungi tepat satu kali oleh suatu kendaraan, 3) total permintaan masing-masing konsumen tidak melebihi kapasitas (Appa, dkk, 2006).

VRP pertama kali dikenalkan oleh Dantzig dan Ramser pada tahun 1959. Solusi dari sebuah VRP adalah menentukan sejumlah rute, yaitu dimulai dari depot kembali ke depot lagi. Menurut Solomon (1987), variasi dari VRP antara lain:

1. *Capacitated Vehicle Routing Problem (CVRP)*
   CVRP merupakan permasalahan VRP dengan kendala kapasitas.
2. *VRP with Time Windows* (VRPTW)
   VRPTW yaitu setiap konsumen harus mensuplai dalam jangka waktu tertentu.
3. *Multiple Depot VRP* (MDVRP)
   MDVRP yaitu distributor memiliki banyak depot untuk mensuplai konsumen.

4. *VRP with Pick-Up and Delivering* (VRPPD)

   VRPPD yaitu konsumen mungkin mengembalikan barang pada depot asal.

5. *Split Delivery VRP* (SDVRP)

   SDVRP adalah konsumen dilayani dengan kendaraan yang berbeda.

6. *Stochastic VRP* (SVRP)

   SVRP adalah munculnya '*random values'* (seperti jumlah konsumen, jumlah permintaan, waktu pelayanan atau waktu perjalanan).

7. *Periodic VRP*

   Periodic VRP adalah pengantaran hanya dilakukan dihari tertentu.

2.5 *Capacitated Vehicle Routing Problem* (CVRP)

CVRP merupakan salah satu permasalahan pada VRP. Kendala pada kasus CVRP terletak pada kapasitas setiap kendaraan. Tujuan dari CVRP yaitu untuk meminimalisasi total biaya perjalanan, dan total permintaan barang untuk setiap rute tidak melebihi kapasitas kendaraan yang melewati rute tersebut (Sungur, 2007:24).

PT Ciomas Adisatwa merupakan perusahaan yang bergerak pada pendistribusian ayam di Jawa Tengah. Dalam pendistribusian daging ayam di PT Ciomas Adisatwa pada hari Senin disediakan 2 kendaraan L300 dan dapat mengangkut maksimal 900 kg daging ayam setiap angkutnya. Pada hari senin terdapat 21 pelanggan dengan 1 depot yaitu PT Ciomas Adisatwa.

Permasalahan CVRP pada pendistribusian daging ayam dapat didefinisikan sebagai suatu graf *G=(V,E)*. Dimana V = {0,1,2,...,N} dengan 0 sampai N adalah gabungan dari konsumen C dan depot, C = {1,2,...,N} adalah konsumen 1 sampai dengan N, dengan depot dinyatakan dengan 0. Jalan yang dilalui oleh kendaraan dinyatakan sebagai himpunan rusuk berarah E yaitu penghubung antar konsumen, $E = \{(i,j)|i,j \in V, i \neq j\}$. Setiap simpul memiliki permintaan (*demand*) sebesar d_i, dengan d_i adalah integer positif. Setiap konsumen dipasok dari depot 0. Himpunan dari k kendaraan mempunyai kapasitas yang sama q ditempatkan di depot 0 dan digunakan untuk melayani konsumen. Sebuah rute didefinisikan

sebagai biaya siklus dari graf G melewati depot 0 sehingga total permintaan dari simpul yang dikunjungi tidak melebihi kapasitas kendaraan (Yeun, 2008:207). Dengan i adalah indeks untuk konsumen awal, $i = 0,1, \dots, N$, j : indeks untuk konsumen tujuan, $j = 1,2, \dots, N$, dan k : indeks untuk kendaraan, $k = 1,2, \dots, K$. Parameter c_{ij} adalah jarak antar konsumen, selanjutnya didefinisikan variabel keputusan x_{ij}^k yang memodelkan ada tidaknya perjalanan dari simpul i ke j dalam rute k.

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

Variabel keputusan yang digunakan dalam pendistribusian x_{ij}^k akan bernilai 1 jika terdapat perjalanan kendaraan k dari konsumen i langsung ke j, dan bernilai 0 jika tidak demikian. Adapun d_i menyatakan total *demand* (permintaan) setiap konsumen i. Menurut Kara, dkk (2004), VRP dapat diformulasikan dalam bentuk pemograman linear dengan meminimalkan total biaya atau total jarak tempuh dari rute perjalanan pendistribusian barang/jasa seperti berikut:

Untuk meminimumkan:

$$Z = \sum_{k=1}^{K} \sum_{i=0}^{N} \sum_{j=1}^{N} c_{ij} x_{ij}^k \quad (2.2)$$

Dengan kendala:

1. Memastikan bahwa setiap konsumen dikunjungi tepat satu kali

$$\sum_{k=1}^{K} \sum_{j=1}^{N} x_{ij}^k = 1, \qquad \forall\, j \in \{1, \dots, N\}, \quad (2.3)$$

2. Menjamin rute tetap tiap kendaraan, sehingga kendaraan yang mengunjungi suatu simpul, setelah melayani akan meninggalkan simpul tersebut

$$\sum_{i=0}^{N} x_{ij}^k - \sum_{j=1}^{N} x_{ij}^k = 0, \qquad \forall k \in \{1, \dots, K\}, \quad (2.4)$$

3. Batas kapasitas kendaraan sehingga tidak ada kendaraan yang melebihi kapasitas

$$\sum_{j=1}^{N} d_i \, x_{ij}^{k} \leq q, \qquad \forall i \in \{0, \dots, N\} \, \forall k \in \{1, \dots, K\} \qquad (2.5)$$

4. Setiap rute perjalanan kendaraan berawal dari depot 0

$$\sum_{j=1}^{N} x_{0j}^{k} = 1, \qquad \forall k \in \{1, \dots, K\}, \qquad (2.6)$$

5. Setiap rute perjalanan kendaraan berakhir di depot 22

$$\sum_{i=0}^{N} x_{i(N+1)}^{k} = 1, \qquad \forall k \in \{1, \dots, K\}, \qquad (2.7)$$

6. x_{ij}^{k} merupakan variabel binner

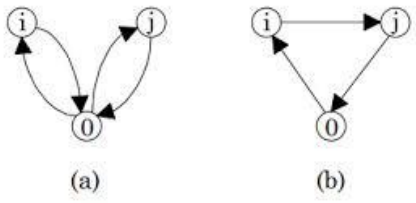
$$x_{ij}^{k} \in \{0,1\}, \qquad \forall \, i, j \in \{1, \dots, N\}, k \in \{1, \dots, K\}. \qquad (2.8)$$

2.6 Algoritma *Clarke & Wright Savings*

Algoritma *Clarke & Wright Savings* ditemukan oleh Clarke and Wright pada tahun 1964. Algoritma *Clarke & Wright Savings* melakukan perhitungan penghematan yang diukur dari seberapa banyak dapat dilakukan pengurangan jarak tempuh dan waktu yang digunakan dengan mengaitkan simpul-simpul yang ada dan menjadikannya sebuah rute bedasarkan nilai *saving* yang terbesar yaitu jarak tempuh antara simpul awal dan simpul tujuan (Octora,dkk, 2014:2). Proses perhitungannya, metode ini tidak hanya menggunakan jarak sebagai parameter, tetapi waktu untuk memperoleh nilai *saving* yang terbesar untuk kemudian disusun menjadi sebuah rute yang terbaik.

Permasalahan dari algoritma ini adalah untuk menetapkan alokasi untuk konsumen dari rute-rute yang ada, bagaimana memilih alur rute yang terpendek untuk mengunjungi semua konsumen dari rute yang telah ditetapkan. Tujuannya adalah untuk menemukan suatu solusi yang meminimalkan total pembiayaan kendaraan, dan mempunyai syarat bahwa setiap konsumen hanya dikunjungi sekali,

dan total permintaan pada suatu rute harus sesuai dengan kapasitas kendaraan. Konsep penghematan (*saving*) yaitu penghematan biaya yang diperoleh dengan menggabungkan dua rute ke satu rute seperti gambar (Lysgaard, 1997:2):

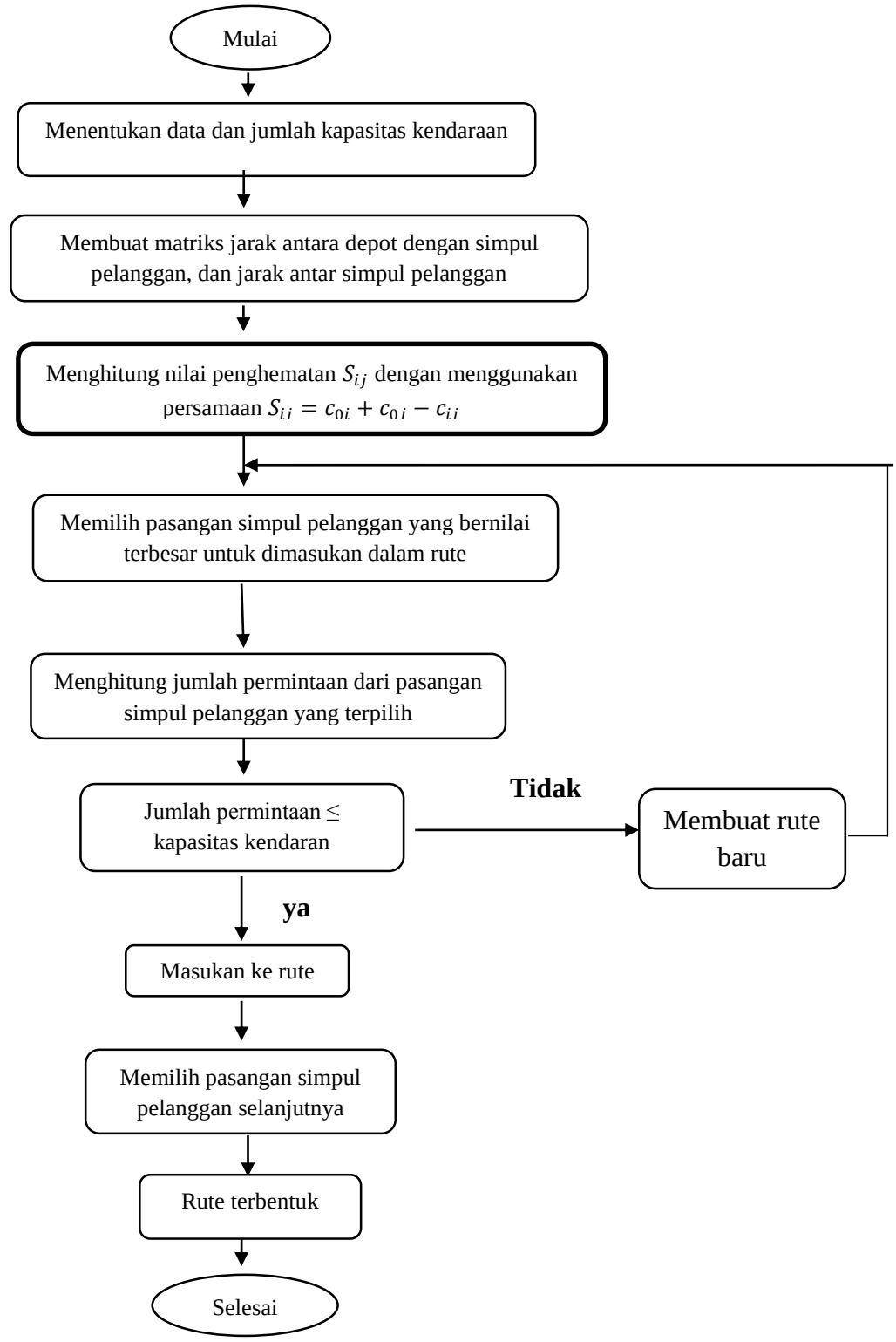


Gambar 1.9 Ilustrasi konsep saving

Awalnya pada gambar (a) seseorang akan mengunjungi konsumen *i* dan *j* melalui rute yang berbeda. Sebuah alternatif untuk mengunjungi dua konsumen pada rute yang sama, misal dalam urutan *i-j* sebagaimana di ilustrasikan pada gambar (b). Untuk masalah biaya transportasi, penghematan dihasilkan dari rute pada gambar (b).

Untuk menjelaskan langkah-langkah metode *Clarke and Wright Savings* disajikan diagram alirnya pada gambar 2.10.

Berikut ini disajikan diagram alir metode *Clarke and Wright Savings*



Gambar 2.10 Diagram alir metode *Clarke and Wright Savngs*

Menurut Purnomo (2010:3) langkah-langkah penyelesaian pada metode ini sebagai berikut:

1. Menentukan jumlah kapasitas maksimum kendaraan. Dimana setiap simpul membentuk rute tersendiri yang dilayani oleh kendaraan yang berbeda.
2. Membuat matriks jarak yaitu matriks jarak antara *depot* dengan *simpul* dan jarak antar *simpul*. Misalkan jarak dari simpul *A* ke *B* sama dengan jarak dari *B* ke *A* sehingga matriks jarak ini termasuk matriks simetrik.
3. Menghitung nilai penghematan (S_{ij}) berupa jarak tempuh dari satu simpul ke simpul yang lain, sedangkan c_{ij} adalah jarak antara titik i ke j, c_{0i} adalah jarak dari depot ke titik i, dan c_{0j} adalah jarak dari depot ke titik j. Dapat dituliskan dengan persamaan

$$S_{ij} = c_{0i} + c_{0j} - c_{ij} \qquad (2.9)$$

4. Membuat matriks penghematan, selanjutnya dicari matriks yang bernilai terbesar terlebih dahulu, dan dilakukan dengan proses berulang dari yang matrik terbesar ke matriks yang bernilai kecil, hingga dihasilkan rute yang diinginkan.

2.7 Algoritma Genetika

2.7.1 Definisi Algoritma Genetika

Algoritma Genetika adalah algoritma optimisasi yang terinsipirasi dari seleksi alam dan gen. Metode ini dapat diterapkan untuk jangkauan masalah yang sangat luas. Algoritma ini mengodekan solusi-solusi yang mungkin ke dalam struktur data alam bentuk kromosom-kromosom dan mengaplikasikan operasi rekombinasi genetika data tersebut. Algoritma genetika pertama kali diperkenalkan oleh John Holland pada tahun 1970 yang merupakan dasar dari algoritma genetika (Coley, 1999:1).

Algoritma ini bekerja dengan sebuah populasi yang terdiri dari individu-indvidu yang masing-masing individu mempresentasikan sebuah solusi yang mungkin bagi persoalan yang ada. Dalam kaitan ini, individu dilambangkan dengan sebuah nilai *fitness* yang akan digunakan untuk mencari solusi terbaik dari persoalan yang ada. Sejak Algoritma Genetika (AG) pertama kali dirintis oleh John

Holland, AG sudah diterapkan begitu luas di berbagai bidang (Sanjoyo,2006:4). AG tidak hanya bisa digunakan untuk masalah optimasi, namun AG dapat digunakan juga untuk menyelesaikan masalahan praktis yang berfokus pada parameter-parameter optimal.

2.7.2 Karakteristik Algoritma Genetika

Berikut diberikan beberapa definisi dari istilah penting yang perlu diperhatikan dalam menyelesaikan masalah menggunakan Algoritma Genetika (Gen & Cheng, 2000:25):

1. Genotype (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika ini, gen dapat berupa nilai biner, string, bit, pohon, array bilangan real, elemen permutasi, elemen program dan lain-lain.
2. Kromosom, yaitu gabungan gen-gen yang membentuk nilai tertentu.
3. Individu, merupakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
4. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
5. Induk adalah kromosom yang akan dikenai operasi genetika.
6. Genetika, menyatakan satu siklus proses evolusi atau satu interasi di dalam algoritma genetika.
7. *Fitness*, alat ukur yang digunakan untuk proses evolusi kromosom. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut.
8. Generasi berikutnya dikenal dengan anak (*offspring*) yang terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*).
9. Mutasi, operator untuk memodifikasi kromosom.

2.7.3 Komponen dalam Algoritma Genetika

Ada enam komponen dalam Algoritma Genetika. Masing-masing akan dibahas sebagai berikut:

1. **Teknik Penyandian**

Teknik penyandian meliputi penyandian gen dari kromosom. Gen merupakan bagian dari kromosom. Masing-masing kromosom berisi sejumlah gen yang menyandikan informasi yang disimpan di dalam kromosom. Gen dapat dipresentasikan dalam bentuk : string, bit, pohon, array bilangan real, elemen permutasi, elemen program dan lain-lain (Sanjoyo, 2006:9).

Contoh dari representasi kromosom antara lain sebagai berikut :

a) String bit : 10011, 11101, dst
b) Bilangan Real : 65.65, 562.88, dst
c) Elemen Permutasi : E2, E10, dst
d) Elemen Program : pemrograman genetika, dst
e) Struktur lainnya

2. **Prosedur *Inisialisasi* (Membangkitkan Populasi Awal)**

Membangkitkan populasi awal dilakukan dengan membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diterapkan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal menggunakan teknik tertentu (Kusumadewi, 2003: 281). Teknik dalam membangkitkan populasi awal ini ada beberapa macam, diantaranya adalah *random generator*, pendekatan tertentu, dan permutasi gen. Teknik membangkitkan populasi yang digunakan pada skripsi ini adalah teknik *random generator* yaitu membangkitkan bilangan *random* untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan.

3. **Nilai *fitness***

Di dalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup, sedangkan yang rendah akan mati. Pada masalah nilai optimasi, fungsi *fitness* yang digunakan adalah (Michalewicz, 1996:72) :

$$f = \frac{1}{x} \qquad (2.10)$$

dengan x merupakan nilai dari individu yaitu total jarak dari jalur yang didapatkan.

4. Seleksi (*Selection*)

Seleksi memiliki tujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang terpilih. Seleksi akan menentukan individu-individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana *offspring* terbentuk dari individu-individu terpilih tersebut. Menurut Kusumadewi (2003:43) terdapat beberapa metode seleksi yaitu:

1) Seleksi *roulette wheel*

   Menurut Sanjoyo (2006:6) seleksi *roulette wheel* adalah metode yang menirukan permainan *roulette wheel* dimana masing-masing kromosom menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitnessnya*. Kromosom yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai *fitness* rendah.

2) Seleksi rangking (*rank-based fitness)*

   Seleksi rangking memperbaiki proses seleksi *roulette wheel*. Populasi diurutkan menurut nilai *fitnessnya*.

3) *Stochastic universal sampling*

   Stochastic universal sampling memiliki nilai bias nol dan penyebaran yang minimum. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurut sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran *fitnessnya*.

4) Seleksi lokal

   Setiap individu yang berada di dalam konstrain tertentu disebut dengan nama lingkungan lokal. Langkah pertama yang dilakukan dalam seleksi lokal adalah menyeleksi separuh pertama dari populasi yang

berpasangan secara random. Kemudian lingkungan baru tersebut diberikan pada setiap individu yang terseleksi.

5) Seleksi dengan turnamen

   Ditetapkan suatu nilai *tour* untuk individu-individu yang terpilih secara acak dari suatu populasi.

6) *Truncation selection*.

   Merupakan seleksi buatan yang digunakan oleh populasi yang jumlahnya sangat besar.

4.1 Seleksi Rangking (*Rank-Based Fitness Assignment*)

Metode *rank-based* (seleksi rangking) muncul untuk mengatasi permasalahan yang ada pada *roullete wheel* yaitu memungkinan bagi individu dengan probabilitas kecil dalam hal ini individu yang kurang baik untuk berpeluang ikut terpilih dalam proses seleksi dengan meningkatkan probabilitas menggunakan rangking berdasarkan individu yang kurang baik ke individu yang paling baik (Kusumadewi: 2003). Untuk individu yang kurang baik akan mendapatkan rangking 1 sedangkan yang paling baik mendapatkan rangking N. Pada *roullet wheel* untuk menentukan probablitas seleksi pada setiap kromosom, sesuai dengan nilai *fitnessnya* sedangkan pada seleksi rangking mengurutkan bedasarkan rangking *fitnessnya*, kemudian menetapkan probablitias seleksi tiap kromosom bedasarkan urutan rangking. Menurut cara kerja metode seleksi rangking sebagai berikut:

a) Membangkitkan populasi awal, sehingga mendapatkan populasi awal.
b) Menghitung nilai fitness dari populasi awal yang sudah terbentuk.
c) Setelah diurutkan, kromosom terburuk diberi nilai fitness baru sebesar 1, kromosom kedua terburuk diberi nilai 2, dan seterusnya. Kromosom terbaik diberi nilai *fitness* baru sebesar N dimana N adalah banyak kromosom dalam suatu populasi.
d) Mencari jarak tempuh tiap rute (z_i)
e) Mencari total jarak dari seluruh rute $(\sum_{i=1}^{N} z_i)$
f) Mencari total fitness $(\sum_{i=1}^{N} f_i)$

g) Mencari probabilitas fitness tiap rute (p_i), dengan rumus

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i}$$

h) Mencari probablitas kumulatif tiap rute (q_i), dengan rumus

$$q_i = \sum_{i=1}^{N} p_i$$

i) Dari probabilitas tersebut, dihitung jatah masing-masing individu pada angka 0 sampai 1

j) Dibangkitkan bilangan random antara 0 sampai 1

k) Dari bilangan random yang dihasilkan, ditentukan individu mana yang terpilih dalam proses seleksi.

Fungsi *rank-based* adalah menyeragamkan skala untuk seluruh individu dalam populasi agar dapat ikut berpeluang terpilih dalam proses seleksi. *Rank-based* pemilihan individu sama sekali tidak bedasarkan pada nilai *fitness* melainkan pada rangking. Hal ini dapat menguntungkan bagi individu yang kurang baik namun kekurangannya adalah semua individu dianggap sama tidak ada lagi individu yang terbaik dan yang paling buruk.

5. Pindah Silang (*Crossover*)

Crossover dalam algoritma genetika operator paling utama karena beroperasi pada dua kromosom pada suatu waktu dan membentuk *offspring* dengan mengombinasikan dua bentuk kromosom. Setelah terpilih induk-induk dari proses seleksi, selanjutnya induk-induk tersebut akan dilakukan proses pindah silang. Pindah silang akan menghasilkan sepasang anak baru hasil dari dua induk. Setiap pasang induk akan dibangkitkan sebuah bilangan acak. Jika bilangan acak tersebut bernilai kurang dari *Probabilitas crossover* (Pc) antara 0,6 s/d 0,95 maka induk tersebut akan dikenai pindah silang. Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan sepenuhnya kepada anak (Michalewicz, 1996:35).

Teknik *crossover* yang digunakan dalam skripsi ini adalah teknik *ordercrossover*. *Ordercrossover* (OX) diperkenalkan oleh Davis (Wira, 2010: 47). Teknik ini dijelaskan dalam contoh sebagai berikut :

Contoh 5 :

Dari 2 induk diketahui:

Induk 1 = 1 2 3 | **4 5 6 7** |8 9

Induk 2 = 4 5 2 | **1 8 7 6** |9 3

Dibangkitkan 2 bilangan acak sebelum gen Induk 1 dan setelah gen Induk 1. Hal yang sama juga dilakukan untuk Induk 2, didapatkan anak dengan gen yang sama:

Anak 1 = x x x | **4 5 6 7** |x x

Anak 2 = x x x | **1 8 7 6** |x x

Langkah berikutnya untuk mendapatkan Anak 1 adalah mengurutkan gen yang berada pada Induk 2 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

Kemudian gen yang telah diurutkan tersebut dibandingkan dengan Anak 1. Apabila gen tersebut ada pada Anak 2 maka abaikan gen tersebut dari urutan itu. Kemudian urutan yang baru saja didapat dimasukkan pada anak 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 menghasikan Anak 2. Anak 1 diperoleh:

Anak 1 = x x x | 4 5 6 7 |x x

Anak 1 = 2 1 8 | 4 5 6 7| 9 3

dengan jalan yang sama dibuat Anak 2 sehingga :

Anak 2 = x x x | 1 8 7 6 |x x

Anak 2 = 3 4 5 | 1 8 7 6 |9 2

6. Mutasi

Mutasi dapat dikatakan sebagai operasi pendukung yang menghasilkan perubahan secara acak dan seketika pada berbagai jenis kromosom. Cara mudah untuk mendapatkan nilai mutasi dengan mengubah satu atau lebih *genes.* Mutasi memiliki peran penting dalam algoritma genetika, yaitu menggantikan gen yang hilang dari populasi selama proses seleksi, sehingga dapat diperoleh gen baru sebagai kandidat solusi pada generasi mendatang dengan fitness yang lebih baik.

Mutasi melakukan kontrol dimana gen baru dalam populasi dapat diuji seleksi. Probabilitas mutasi (Pm) didefinisikan sebagai persentasi dari jumlah total gen pada populasi yang mengalami mutasi. Pm mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika Pm terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi jika Pm terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya, dan juga algoritma akan kehilangan kemampuan untuk belajar dari pencarian sebelumnya (Kusumadewi, 2003:296).

Teknik mutasi yang digunakan dalam skripsi ini adalah teknik *swapping mutation.* Teknik ini 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, 2005:65).

7. Elitisme

Proses ini dilakukan secara *random* sehingga tidak ada jaminan bahwa suatu individu yang mempunyai nilai *fitness* tertinggi akan selalu terpilih. Walaupun individu bernilai *fitness* tertinggi terpilih, ada kemungkinan individu tersebut akan rusak (nilai *fitnesnya* 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 beberapa kopinya. Prosedur ini dikenal sebagai elitisme (Sanjoyo, 2006:10).

2.7.4 Parameter yang digunakan dalam Algoritma Genetika

Menurut Tanjaya, dkk (2011) terdapat beberapa parameter yang digunakan dalam Algoritma Genetika, yaitu:

1. Jumlah generasi

   Merupakan jumlah perulangan (iterasi) dilakukan rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan output dan lama iterasi (waktu proses algoritma genetika). Jumlah generasi semakin besar akan semakin membutuhkan waktu *running* yang lama.
2. Ukuran Populasi

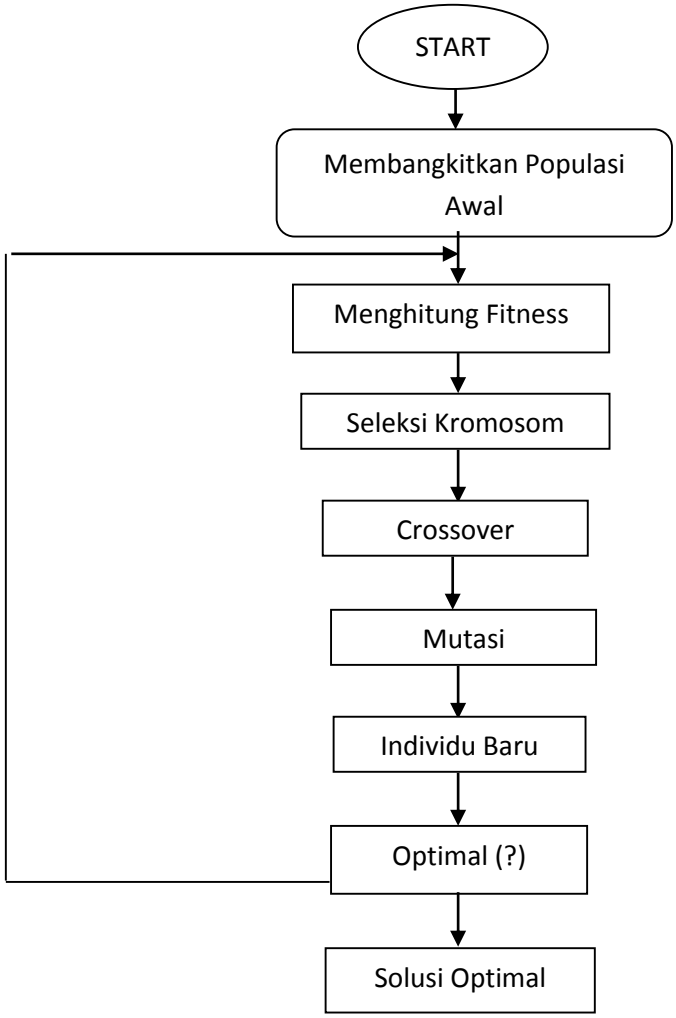
   Ukuran populasi menyatakan jumlah kromosom yang ada pada populasi. Ukuran populasi yang ideal berbeda-beda untuk setiap permasalahan.
3. Probabilitas *Crossover* (Pc)

   Probabilitas c*rossover* ini digunakan untuk mengendalikan frekuensi operator *crossover*. Pada setiap individu dibangkitkan sebuah bilangan acak *p*, jika $p < Pc$ maka individu tersebut akan dilakukan *crossover* dan sebaliknya. Menurut Michalewicz (1996:35), probabilitas *crossover* yang ideal yaitu berkisar antara 60% sampai 95%.
4. Probabilitas Mutasi (Pm)

   Probabilitas mutasi ini digunakan untuk menentukan tingkat mutasi yang terjadi. Sama halnya seperti proses *crossover*, proses mutasi juga membangkitkan sebuah bilangan acak *p*, jika $p < Pm$ maka sebuah gen acak *r* dimutasi dan sebaliknya. Menurut Suyanto (2005:14), probabilitas mutasi yang ideal adalah 1/jumlah gen.

Untuk menjelaskan langkah-langkah Algoritma Genetika disajikan diagram alirnya pada gambar 2.11.

Langkah-langkah penyelesaian Algoritma Genetika menurut Suyanto (2005) dapat dibuat diagram alirnya sebagai berikut:



Gambar 2.11 Diagram alir proses Algoritma Genetika

2.8 Penelitian yang Relevan

Beberapa penelitian tentang CVRP dan *Clarke and Wright Savings* telah banyak dilakukan antara lain penelitian dari Puji Rahmawati, penelitian ini menggunakan metode *Clarke and Wright Savings* dalam menyelesaikan CVRP dengan menggunakan data dari PT Wina Putra Jaya. Dalam penelitian ini membandingkan rute pendistribusian gas LPG 3 kg dari perusahaan dan hasil dari metode *Clarke and Wright Savings* ternyata lebih optimal menggunakan rute

pendistribusian dari hasil metode *Clarke and Wright Savings.* Selain penelitian menggunakan metode *Clarke and Wright Savings* sebagai penyelesaian untuk rute pendistribusian terdapat juga penelitian tentang CVRP dan Algoritma Genetika sebagai penyelesaian rute pendistribusian oleh Ikhsan Hidayat dengan membandingkan Algoritma Genetika dan Algoritma *Sweep* untuk optimasi rute distribusi surat kabar Kedaulatan Rakyat. Hasil dari penelitian ini adalah Algoritma Genetika menghasilkan solusi yang optimal daripada Algoritma *Sweep*.