

## **BAB II**

### **KAJIAN PUSTAKA**

Secara umum, pada bab ini akan dibahas mengenai kajian teori yang digunakan dalam penelitian ini yaitu graf, *vehicle routing problem* (VRP), *capacitated vehicle routing problem with time windows* (CVRPTW), dan algoritma genetika.

#### **A. Graf**

##### **1. Definisi Graf**

Menurut Munir (2009: 291), graf  $G$  didefinisikan sebagai pasangan himpunan  $(V, E)$ , ditulis dengan notasi  $G = (V, E)$ . Dalam hal ini,  $V$  merupakan himpunan tak kosong dari simpul-simpul (*vertex* atau *node*) digambarkan dalam titik-titik, dan  $E$  adalah himpunan sisi-sisi (*edges* atau *arcs*) digambarkan dalam garis-garis yang menghubungkan sepasang simpul. Dapat disimpulkan bahwa graf merupakan sekumpulan titik-titik yang dihubungkan dengan garis-garis.

##### **2. Jenis-Jenis Graf**

Graf dapat dikelompokkan menjadi beberapa kategori, bergantung pada sudut pandang pengelompokannya. Pengelompokan graf dapat dipandang berdasarkan ada tidaknya sisi ganda atau sisi gelang (*loop*), berdasarkan jumlah simpul atau berdasarkan orientasi arah pada sisi (Munir, 2009: 357).

Berdasarkan ada tidaknya sisi ganda atau sisi gelang (*loop*) pada suatu graf, maka graf dapat digolongkan menjadi dua jenis, yaitu :

1. Graf sederhana (*simple graph*)

Graf sederhana yaitu graf yang tidak mengandung gelang maupun sisi ganda.

2. Graf tak sederhana (*unsimple graph*)

Graf tak sederhana yaitu graf yang mengandung ruas ganda atau gelang.

Berdasarkan jumlah simpul pada suatu graf, maka graf dapat digolongkan menjadi dua jenis, yaitu :

1. Graf berhingga (*limited graph*)

Graf berhingga yaitu graf yang jumlah simpulnya  $n$ , berhingga.

2. Graf tak berhingga (*unlimited graph*)

Graf tak berhingga yaitu graf yang jumlah simpulnya tidak berhingga.

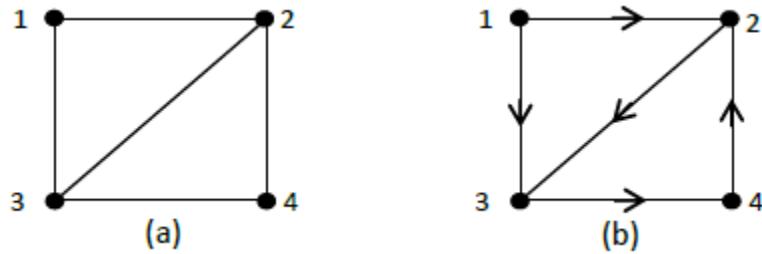
Berdasarkan orientasi arah pada sisi, maka graf dapat dibedakan menjadi dua jenis, yaitu :

1. Graf tak berarah (*undirected graph*)

Graf tak berarah yaitu graf yang sisinya tidak mempunyai orientasi arah.

2. Graf berarah (*directed graph*)

Graf berarah yaitu graf yang setiap sisinya diberikan orientasi arah.



Gambar 2.1 (a). Graf tak berarah (b). Graf berarah

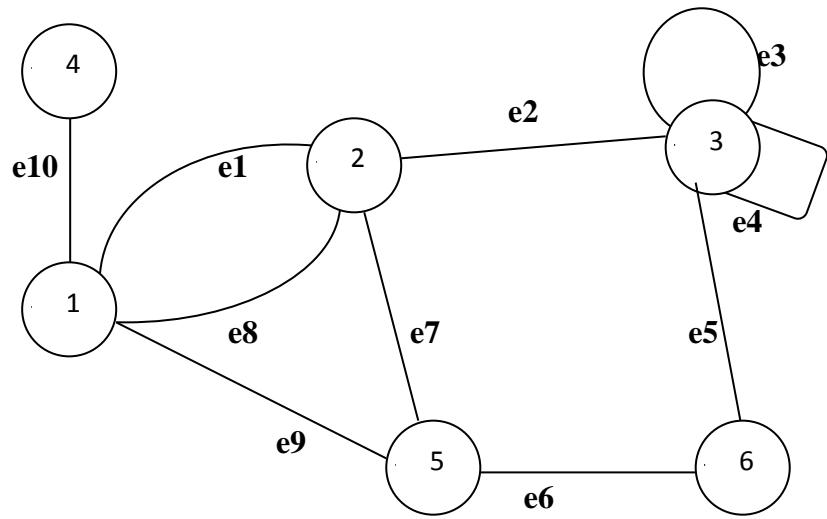
### 3. Keterhubungan

Sebuah graf  $G$  dikatakan terhubung jika untuk setiap dua simpul di  $G$  selalu terdapat lintasan (*path*) yang menghubungkan kedua simpul tersebut (Rossen, 1999: 468). Sebaliknya graf  $G$  disebut graf tidak terhubung jika tidak ada lintasan (*path*) yang menghubungkan  $u$  dan  $v$  di  $G$ . Menurut Tenia & Elisa (2016: 117), keterhubungan dapat dibagi menjadi 4 bagian, yaitu:

#### 1) Perjalanan (*Walks*)

Perjalanan dalam sebuah graf  $G = (V, E)$  adalah barisan terhingga dengan bentuk  $v_1, e_1, v_2, e_2, v_3, e_3, \dots, e_{n-1}, v_n$  dengan sisi  $e_i$  menghubungkan  $v_i$  dengan  $v_{i+1}$ .

Berikut adalah contoh sebuah graf.



Gambar 2.2 Graf  $G$

Contoh suatu perjalanan di graf  $G$  adalah  $1, e_1, 2, e_2, 3, e_3, 3, e_5, 6, e_6, 5, e_9, 1$ .

#### 2) Lintasan (*Trails*)

Lintasan adalah perjalanan dengan semua sisi dalam barisan adalah berbeda.

Contoh suatu lintasan yang terdapat dalam graf  $G$  adalah  $2, e_2, 3, e_3, 3, e_5, 6, e_6, 5, e_9, 1$ .

#### 3) Jalur (*Path*)

Jalur adalah perjalanan dengan semua simpul dalam barisan adalah berbeda.

Contoh suatu jalur yang terdapat dalam graf  $G$  adalah  $1, e_1, 2, e_7, 5, e_6, 6, e_5, 3$ .

#### 4) Sirkuit (*Circuit*)

Sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama.

Contoh sirkuit yang terdapat dalam graf  $G$  adalah  $1, e_1, 2, e_7, 5, e_9, 1$ .

## **B. *Vehicle Routing Problem (VRP)***

*Vehicle routing problem* (VRP), pertama kali diperkenalkan oleh Dantzig & Ramser pada tahun 1959 dalam penelitiannya “*the Truck Dispatching Problem*”. VRP memegang peranan penting pada manajemen distribusi dan telah menjadi salah satu permasalahan dalam optimalisasi yang dipelajari secara luas. *Vehicle Routing Problem* (VRP) bertujuan untuk menemukan rute dengan ongkos yang optimum (menemukan rute terpendek, memimumkan kendaraan yang digunakan, dsb) dimulai dan diakhiri di depot, dan permintaan semua agen terpenuhi. Setiap agen hanya dikunjungi sekali oleh satu kendaraan dan setiap kendaraan pengangkut memiliki batasan kapasitas (Tonci & Hrvoje, 2008: 1). Batasan kapasitas berdasarkan daya angkut kendaraan setelah diisi logistik yang akan didistribusikan, sedangkan VRP merupakan manajemen distribusi barang yang memperhatikan pelayanan dengan periode waktu tertentu, sekelompok konsumen, dan sejumlah kendaraan. Solusi dari sebuah VRP yaitu menentukan sejumlah rute, yang masing-masing dilayani oleh satu kendaraan yang berasal dan berakhir pada depot, sehingga kebutuhan pelanggan terpenuhi, semua permasalahan operasional terselesaikan dan biaya transportasi secara umum diminimalkan.

Terdapat beberapa jenis VRP yang sangat bergantung pada jumlah faktor pembatas dan tujuan yang akan dicapai. Pembatas yang paling umum digunakan yaitu waktu dan jarak. Tujuan yang ingin dicapai biasanya adalah meminimalkan jarak tempuh, waktu maupun biaya. Menurut Toth & Vigo (2002: 4), terdapat empat tujuan umum dari permasalahan VRP, yaitu :

- a. Meminimumkan biaya transportasi terkait dengan jarak dan biaya tetap yang berhubungan dengan kendaraan
- b. Meminimumkan jumlah kendaraan yang dibutuhkan untuk melayani setiap agen
- c. Menyeimbangkan rute, untuk waktu perjalanan dan muatan kendaraan
- d. Meminimumkan pinalti akibat pelayanan yang kurang memuaskan dari agen

Menurut Suprayogi (2003: 4) beberapa contoh variasi dari VRP diantaranya:

1. VRP *with multiple trips*: satu kendaraan dapat melakukan lebih dari satu rute untuk memenuhi kebutuhan pelanggan.
2. VRP *with time window*: setiap pelanggan mempunyai rentang waktu pelayanan yaitu pelayanan harus dilakukan pada rentang *time windows* masing-masing pelanggan.
3. VRP *with split deliveries*: setiap pelanggan boleh dikunjungi lebih dari satu kendaraan.
4. VRP *with multiple products*: permintaan pelanggan lebih dari satu produk. Pada umumnya, VRP bentuk ini juga melibatkan kendaraan dengan *multi-compartments*.
5. VRP *with delivery* dan *pick-up*: terdapat sejumlah barang yang perlu dipindahkan dari lokasi penjemputan tertentu ke lokasi pengiriman lainnya.
6. VRP *with multiple depots*: depot awal untuk melayani pelanggan lebih dari satu.

### C. *Capacitated Vehicle Routing Problem with Time Window (CVRPTW)*

*Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)*

merupakan gabungan dari permasalahan *capacitated vehicle routing problem* (CVRP) dan *vehicle routing problem with time windows* (VRPTW). Penelitian VRPTW pertama kali dilakukan oleh Pullen & Webb pada tahun 1967 dan terus berkembang hingga sekarang. Jika suatu masalah VRPTW menambahkan kapasitas sebagai kendala tambahan, maka permasalahan tersebut berubah menjadi kasus *Capacited Vehicle Routing Problem with Time Window* (CVRPTW).

Permasalahan dalam CVRPTW adalah sebagai berikut: sebanyak  $N$  konsumen akan dilayani dari sebuah depot, dengan sejumlah kendaraan yang memiliki kapasitas yang sama ( $Q$ ). Untuk setiap konsumen  $i$ ,  $i = 1, 2, \dots, N$  terdapat permintaan sebanyak  $q_i$ , waktu pelayanan  $s_i$ , dan pelayanan *time window*  $z_i = [e_i, l_i]$  dengan  $e_i$  adalah waktu paling awal untuk melakukan pelayanan (*lower bound*) dan  $l_i$  adalah waktu paling lambat untuk melakukan pelayanan (*upper bound*). Sedangkan permintaan  $q_i$  dari konsumen harus dipenuhi dengan sekali pelayanan saja dalam batas *time windows* ( $z_i$ ).

#### 1. Formulasi CVRPTW

Masalah CVRPTW dapat dipresentasikan sebagai suatu graf berarah  $G = (V, E)$  dengan  $V = \{v_0, v_1, v_2, \dots, v_n\}$  adalah himpunan simpul (verteks),  $v_0$  menyatakan depot yaitu tempat kendaraan memulai dan mengakhiri rute perjalanan. Dalam penelitian ini depot disebut juga sebagai gudang beras. Sedangkan  $E = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$  adalah himpunan rusuk atau garis

berarah yang menghubungkan dua simpul yaitu ruas jalan penghubung antar konsumen atau antar depot dengan konsumen.

Setiap simpul  $\{v_i \in V, i \neq 0\}$  memiliki permintaan sebesar  $q_i$ . Himpunan  $K = \{k_1, k_2, \dots, k_n\}$  merupakan kumpulan kendaraan yang homogen dengan kapasitas maksimal sama yaitu  $Q$ , sehingga panjang setiap rute dibatasi oleh kapasitas kendaraan. Setiap verteks  $(v_i, v_j)$  memiliki waktu tempuh  $t_{ij}$  yaitu waktu tempuh dari simpul  $i$  ke  $j$ .

Dari permasalahan CVRPTW tersebut, dapat diformulasikan ke dalam bentuk model matematika dengan tujuan meminimumkan total waktu pendistribusian dalam melayani semua konsumen. Jika  $z$  adalah fungsi tujuan untuk masalah CVRPTW, maka:

$$\min z = \sum_{i=1}^N \sum_{j=1}^N t_{ij} \sum_{k=1}^K x_{ijk}. \quad (2.1)$$

Dengan variabel keputusan sebagai berikut :

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

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

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

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

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

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

Dan kendala dari permasalahan CVRPTW adalah sebagai berikut :

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

$$\sum_{k=1}^K \sum_{j=1}^N x_{i,j}^k = 1. \quad (2.2)$$

2. Total jumlah permintaan konsumen dalam satu rute tidak melebihi kapasitas kendaraan yang melayani rute tersebut. Misalkan terdapat lintasan dari  $i$  ke  $j$  dengan kendaraan  $k$ , maka

$$Y_{ik} + q_j = Y_{jk}, \forall i, j \in N, \forall k \in K.$$

$$Y_{jk} \leq Q, \forall j \in N, \forall k \in K. \quad (2.3)$$

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

$$T_{ik} + s_{ik} + t_{ij} \leq T_{jk}, \forall i, j \in N, \forall k \in K. \quad (2.4)$$

4. Waktu kendaraan untuk memulai pelayanan di konsumen ke- $i$  harus

berada pada selang waktu  $[e_i, l_i]$ .

$$e_i \leq T_{ik} \leq l_i, \forall i \in N, \forall k \in K. \quad (2.5)$$

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

$$\sum_{i=1}^N x_{ijk} - \sum_{i=1}^N x_{ijk} = 0, \forall i, j \in N, \forall k \in K. \quad (2.6)$$

6. Variabel keputusan  $x_{ijk}$  merupakan integer biner.

$$x_{ijk} \in \{0,1\}, \forall i, j \in N, \forall k \in K. \quad (2.7)$$

## D. Algoritma Genetika

### 1. Pengertian Algoritma Genetika

Algoritma genetika adalah suatu proses optimasi yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah (Haupt & Haupt 2004: 22). 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. Gen biasanya merupakan simbol *string* (Gen, 1997: 1).

Proses evolusi alamiah yaitu terbentuknya populasi awal secara acak yang terdiri dari individu-individu dengan sifat yang tergantung pada gen-gen

dalam kromosomnya. Individu-individu melakukan proses reproduksi untuk melahirkan keturunan. Sifat keturunan dibentuk dari kombinasi sifat kedua induknya. Metode optimasi ini dikembangkan oleh John Holland pada tahun 1960-an dan dipopulerkan oleh mahasiswanya, David Goldberg pada tahun 1980-an. Dalam bukunya “*Adaptation in Natural and Artificial Systems*”, John Holland menjabarkan dasar-dasar algoritma genetika. Algoritma genetika didasarkan pada proses alamiah yaitu Teori Evolusi Darwin. Dalam algoritma genetika, suatu populasi dari individu bereproduksi kembali berdasarkan nilai *fitness* (Spears, 1989: 4). Kemunculan algoritma genetika diinspirasikan dari teori-teori dalam ilmu biologi, sehingga banyak terdapat istilah dalam biologi yang digunakan dalam algoritma ini. Sesuai dengan namanya, proses-proses yang terjadi dalam algoritma genetika sama dengan apa yang terjadi pada evolusi biologi yaitu seleksi, pindah silang, dan mutasi.

Menurut Zainudin (2014: 11), pencarian dimulai dengan pembangkitan sejumlah “individu” secara acak yang disebut dengan kromosom. Kromosom-kromosom ini merupakan representasi calon penyelesaian yang akan diperiksa nilai yang sebenarnya. Seperti halnya proses evolusi alamiah, kromosom-kromosom akan dinilai tingkat kebugarannya (*fitness*). Semakin besar nilai *fitness*, semakin besar pula kemungkinannya untuk dipertahankan ke dalam populasi selanjutnya. Nilai *fitness* adalah nilai yang menunjukkan nilai ketangguhan kromosom dalam beradaptasi.

Kromosom-kromosom yang dibentuk dari kromosom generasi sebelumnya disebut sebagai anak (*offspring*). Demikian juga dengan

kromosom generasi sebelumnya disebut sebagai induk (*parents*). Proses pembentukan anak dari induknya dilakukan dengan penyilangan (*crossover*). Dalam algoritma genetika, dikenal operator mutasi (*mutation*), yaitu operator yang dapat mengubah gen-gen dalam kromosom. Dalam parameter algoritma genetika, terdapat beberapa batasan. Salah satunya adalah ukuran populasi. Dalam algoritma genetika, ukuran populasi setiap generasi adalah tetap. Populasi generasi selanjutnya dibentuk dengan cara menyeleksi kromosom induk dan kromosom anak berdasarkan nilai *fitness*.

Secara singkat, Tabel 2.1 menyatakan pemetaan proses alamiah ke dalam proses komputasi algoritma genetika.

Tabel 2.1 Pemetaan Proses Alamiah ke Proses Komputasi

Proses Alamiah	Proses Komputasi
Individu	Penyelesaian masalah
Populasi	Himpunan penyelesaian
Kebugaran/ <i>fitness</i>	Kualitas penyelesaian
Kromosom	Kode/representasi penyelesaian
Gen	Bagian dari representasi penyelesaian
Pertumbuhan	Pengkodean representasi penyelesaian
Penyilangan	Operator genetika
Mutasi	Operator genetika
Seleksi Alam	Menyeleksi penyelesaian masalah (sementara) berdasarkan kualitasnya

## 2. Karakteristik Algoritma Genetika

Beberapa definisi penting dalam algoritma genetika adalah sebagai berikut :

- 1) Gen (*Genotype*) adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom.
- 2) *Allele* yaitu nilai dari sebuah gen, dapat berupa bilangan biner, float, integer, karakter, dan kombinatorial.
- 3) Kromosom adalah gabungan gen-gen yang membentuk nilai tertentu.
- 4) Individu merupakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
- 5) Populasi merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi. Populasi terdiri dari sekumpulan kromosom.
- 6) Induk adalah kromosom yang akan dikenai operasi genetika (*crossover*).
- 7) *Crossover* adalah operasi genetika yang mewakili proses perkembangbiakan antar individu.
- 8) *Offspring* adalah kromosom yang merupakan hasil dari operasi genetika (*crossover*) dikenal keturunan atau sebagai anak.
- 9) Mutasi merupakan operasi genetika 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.
- 10) Proses seleksi merupakan proses yang mewakili proses seleksi alam (*natural selection*) dari Teori Darwin. Proses ini dilakukan untuk menghasilkan keturunan (*offspring*).

- 11) Nilai *fitness* merupakan penilaian yang menentukan bagus tidaknya sebuah kromosom.
- 12) Fungsi evaluasi adalah fungsi yang digunakan untuk menentukan nilai dari nilai *fitness*. Fungsi evaluasi ini merupakan sekumpulan kriteria-kriteria tertentu dari permasalahan yang ingin diselesaikan.
- 13) 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 di bawah nilai minimal akan dihapus dari populasi.

### **3. Langkah Optimasi dalam Algoritma Genetika**

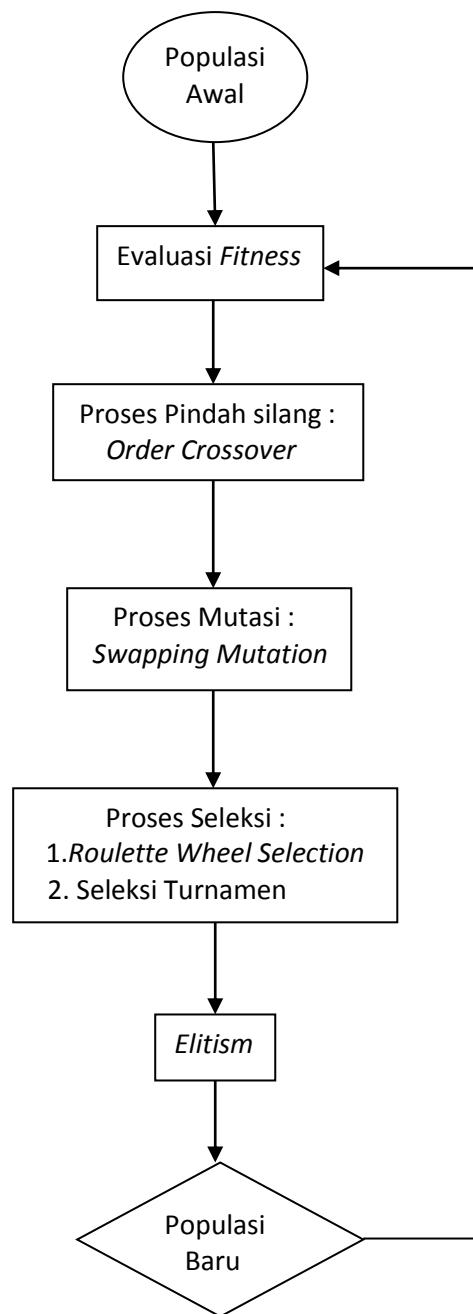
Menurut Hannawati (2002: 80), ada 5 langkah yang harus dilakukan untuk menyelesaikan masalah optimasi menggunakan algoritma genetika, yaitu :

1. Memilih dua individu sebagai orangtua (*parents*) yang kemudian dilakukan perkawinan silang (*crossover*) untuk menghasilkan individu baru sebagai anak (*offspring*).
2. Memilih satu individu sebagai orangtua (*parents*) yang kemudian akan dilakukan perubahan gen meliputi operator mutasi (*mutation*) pada tingkat tertentu.
3. Menerapkan skema pergantian untuk menghasilkan populasi baru.
4. Melakukan iterasi dengan cara mengulang seluruh proses evolusi hingga

berhenti pada kondisi tertentu. Kondisi berhenti dapat ditentukan dari jumlah iterasi yang diinginkan, berdasarkan waktu tertentu atau ketika didapatkan variasi individu terbaik yang memiliki nilai terbesar dalam suatu populasi.

5. Seleksi nilai *fitness* terbesar yang dihasilkan dari iterasi didapatkan dari populasi baru yang memiliki individu-individu yang lebih baik dari individu lama sebelumnya.

Berdasarkan langkah-langkah di atas, maka dapat dibuat bagan sebagai berikut :



Gambar 2.3 Bagan Algoritma Genetika menurut Michalewich (1996)

#### 4. Komponen dalam Algoritma Genetika

Terdapat beberapa komponen utama dalam algoritma genetika, komponen tersebut adalah sebagai berikut :

### **a. Teknik Pengkodean**

Teknik pengkodean adalah suatu cara bagaimana mengkodekan gen dari kromosom, dimana gen merupakan salah satu bagian dari kromosom. Satu gen akan mewakili satu variabel. Agar dapat diproses melalui algoritma genetika, maka alternatif solusi tersebut harus dikodekan terlebih dahulu ke dalam bentuk kromosom. Masing-masing kromosom berisi sejumlah gen yang mengkodekan informasi yang disimpan dalam kromosom (Kusumadewi, 2003: 280).

Gen dapat dipresentasikan dalam bentuk: string bit, pohon, array, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika. Pada penelitian ini, representasi gen menggunakan teknik pengkodean permutasi. Dalam teknik pengkodean permutasi, tiap gen dalam kromosom mempresentasikan suatu urutan (Anwar & Yuliani, 2005: 43).

Contoh 2.1    Kromosom 1 = 3 5 7 2 1 6 8 4

**Keterangan:** kromosom 1 berisi urutan gen secara acak dari gen kesatu sampai dengan gen kedelapan. Gen dipresentasikan dengan sebuah bilangan dan bilangan-bilangan tersebut representasi dari masing-masing titik distribusi.

### **b. Membangkitkan 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 diimplementaikan. 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: 102).

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). Di dalam evolusi alam, individu yang memiliki nilai *fitness* tinggi akan bertahan hidup, dan individu yang memiliki nilai *fitness* rendah akan mati (Goldberg, 1989: 72). Pada masalah optimasi dengan kendala kapasitas dan waktu, fungsi *fitness* yang digunakan adalah

$$f = \frac{1}{h} \quad (2.8)$$

Dengan nilai  $h$  merupakan nilai dari total waktu pendistribusian

dijumlahkan dengan waktu pinalti. Waktu pinalti yaitu waktu pelayanan yang melebihi jangka waktu yang tersedia. Semakin kecil nilai  $h$ , maka akan semakin besar nilai *fitness*nya. Tetapi jika  $h$  bernilai 0, maka akan mengakibatkan nilai *fitness*nya tak terhingga. Untuk mengatasi hal tersebut, maka nilai  $h$  perlu ditambahkan dengan bilangan yang sangat kecil sehingga fungsi *fitness*nya menjadi

$$f = \frac{1}{(h + a)}. \quad (2.9)$$

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

#### **d. Seleksi**

Seleksi yang digunakan dalam algoritma genetika merupakan adaptasi dari seleksi alam yang diteliti oleh Darwin. Seleksi dalam algoritma genetika bertujuan untuk menentukan individu-individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana keturunan (*offspring*) terbentuk dari individu-individu terpilih (Kusumadewi, 2003: 87). Proses seleksi dapat dilakukan secara proporsional berdasarkan nilai-nilai *fitness* yang dihasilkan. Dalam algoritma genetika, seleksi yang menggunakan prinsip *survival of the fittest* (bertahan karena merupakan yang terkuat) paling sering digunakan. Prinsip ini bertujuan supaya kualitas kebugaran (*fitness*) individu-individu pada setiap generasi dapat bertambah. Misalnya terdapat individu A yang lebih kuat atau nilai *fitness*nya lebih besar daripada individu B, maka individu

A berpeluang lebih besar untuk terpilih sebagai calon orangtua.

Menurut Kusumadewi (2003: 282), terdapat beberapa metode seleksi yaitu seleksi rangking (*rank-based fitness assignment*), seleksi roulette wheel (*roulette wheel selection*), *stochastic universal sampling*, seleksi lokal (*local selection*), seleksi dengan pemotongan (*truncation selection*), dan seleksi dengan turnamen (*tournament selection*). Pada penelitian ini, akan digunakan beberapa variasi seleksi yang sering digunakan untuk menyelesaikan masalah pendistribusian raskin di Kota Yogyakarta sehingga dapat diketahui metode seleksi mana yang akan memberikan solusi yang optimum. Seleksi yang akan digunakan adalah : *roulette wheel selection*, dan seleksi turnamen.

### 1) *Roulette Wheel Selection*

Pada *roulette wheel selection*, setiap kromosom dalam suatu populasi memiliki tempat yang sesuai dengan proporsinya terhadap total nilai *fitness*. Kromosom-kromosom dipetakan kedalam suatu segmen secara berurutan, hingga tiap-tiap segmen kromosom memiliki ukuran yang sesuai dengan nilai *fitness*. Langkah pertama dari seleksi ini adalah menghitung nilai *fitness* masing-masing kromosom. Setelah itu, dihitung proporsi masing-masing kromosom berdasarkan perbandingan probabilitas antara nilai *fitness* setiap kromosom dengan total nilai *fitness*. Langkah selanjutnya adalah membangkitkan bilangan real secara random antara 0 dan 1 untuk menentukan kromosom yang bertahan hidup dan menjadi induk.

Cara kerja metode ini adalah sebagai berikut :

- 1) Dihitung nilai *fitness* masing-masing individu ( $f_i$ , dimana  $i$  adalah

individu ke-1 sampai dengan ke-n).

- 2) Dihitung total nilai *fitness* semua individu.
- 3) Dihitung probabilitas masing-masing individu, dengan rumus probabilitas seleksi yaitu :

$$P_i = \frac{f_i}{\sum_{i=1}^n f_i} \times 100\%. \quad (2.10)$$

- 4) Dari perhitungan, diperoleh nilai untuk masing-masing individu pada angka 1 sampai dengan 100.
- 5) Dibangkitkan bilangan acak antara 1 sampai dengan 100.
- 6) Dari bilangan acak yang dihasilkan, dapat ditentukan individu mana yang terpilih dalam proses seleksi.

Urutan langkah proses seleksi menggunakan *roulette wheel selection* dapat digambarkan sebagai berikut :

Misalkan diberikan sebuah contoh populasi beserta nilai *fitnessnya* yang ditunjukkan pada Tabel 2.2.

Tabel 2.2 Contoh Populasi Beserta Nilai *Fitness*nya

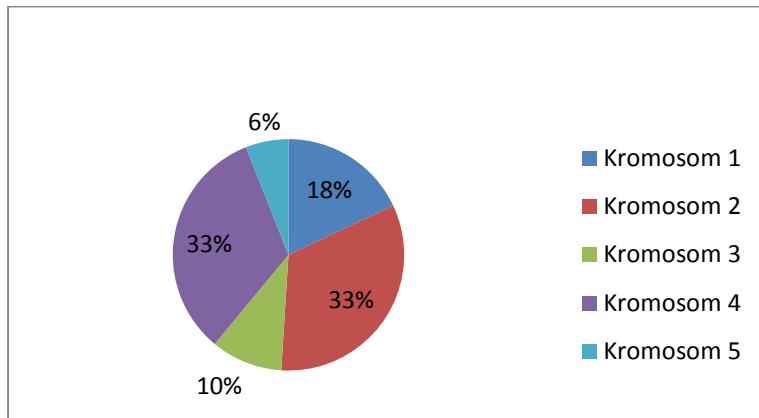
<b>Kromosom</b>	<b><i>Fitness</i></b>
Kromosom 1	0,1111
Kromosom 2	0,2000
Kromosom 3	0,0588
Kromosom 4	0,2000
Kromosom 5	0,0400
<b>Total <i>Fitness</i></b>	<b>0,6099</b>

Langkah selanjutnya adalah menentukan nilai segmen untuk masing-masing kromosom yang ditunjukkan pada Tabel 2.3.

Tabel 2.3 Nilai Probabilitas dan Segmen untuk Masing-masing Kromosom

<b>Kromosom</b>	<b><i>Fitness</i></b>	<b>Probabilitas</b>	<b>Segmen</b>
Kromosom 1	0,1111	18%	1-17
Kromosom 2	0,2000	33%	18-50
Kromosom 3	0,0588	10%	51-60
Kromosom 4	0,2000	33%	61-93
Kromosom 5	0,0400	6%	94-100

Berdasarkan data probabilitas tabel di atas maka dapat dibuat diagram lingkaran seperti pada Gambar 2.4.



Gambar 2.4 Segmen masing-masing kromosom

Selanjutnya, memutar roda roulette sebanyak kromosom pada populasi yaitu sebanyak 5 kali. Setiap kali putaran menghasilkan suatu bilangan *random* dengan rentang 1-100 yang menunjukkan daerah atau segmen dari kromosom. Tabel 2.4. adalah contoh daerah yang terpilih setelah 5 kali putaran.

Tabel 2.4 Hasil Kromosom yang Terpilih

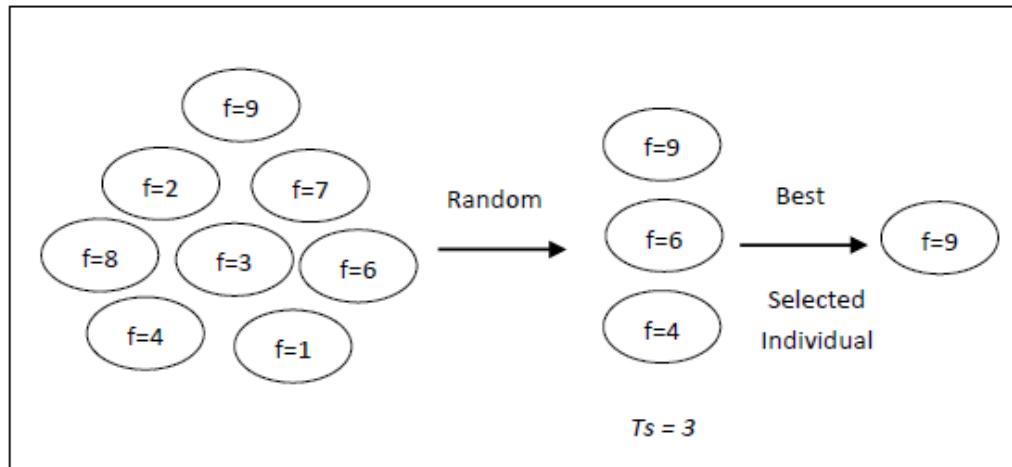
Putaran ke-	Daerah Terpilih	Kromosom Terpilih
1	40	Kromosom 2
2	76	Kromosom 4
3	84	Kromosom 4
4	15	Kromosom 1
5	35	Kromosom 2

## 2) Seleksi Turnamen

Seleksi turnamen merupakan salah satu metode seleksi yang paling popular dalam algoritma genetika karena efisiensi dan implementasi yang sederhana. Seleksi ini merupakan jenis seleksi yang divariasi berdasarkan *roulette wheel selection* dan seleksi rangking. Dalam seleksi turnamen,  $n$

individu dipilih secara acak. Banyaknya perbandingan dalam turnamen terhadap individu biasanya disebut dengan *tournament size*. Satu individu akan bersaing dengan individu lain untuk menentukan nilai *fitness* tertinggi yang nantinya akan menjadi pemenang dan individu sebagai pemenang akan terpilih dalam populasi generasi berikutnya. Seleksi turnamen juga memberikan kesempatan pada semua individu terpilih untuk mempertahankan keragamannya. Pemilihan turnamen memiliki beberapa keunggulan yang meliputi efisien kompleksitas waktu, terutama jika dilaksanakan secara paralel dan tidak ada persyaratan untuk skala nilai *fitness*.

Pada Gambar 2.5 akan ditunjukkan mekanisme seleksi turnamen menurut Razali (2011: 1135).



Gambar 2.5 Mekanisme Seleksi Turnamen

Pada gambar diatas menyatakan bahwa *tournament size* (Ts) sebanyak tiga perbandingan, yang berarti bahwa tiga kromosom bersaing satu sama lain. Hanya kromosom yang memiliki nilai terbaik akan dipilih untuk reproduksi pada generasi selanjutnya. Dalam seleksi turnamen, nilai yang lebih besar dari

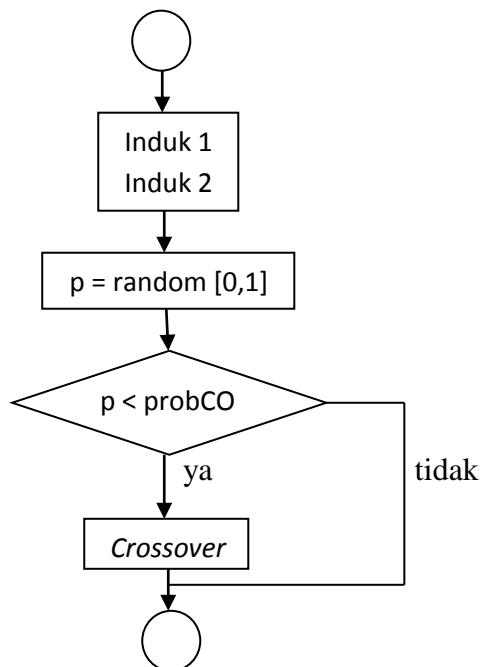
*tournament size* menyebabkan hilangnya keanekaragaman individu yang diharapkan. Semakin besar *tournament size* maka populasi yang terbentuk semakin kecil untuk memberikan kontribusi terhadap keanekaragaman genetik. Terdapat dua faktor yang menyebabkan hilangnya keanekaragaman dalam seleksi turnamen regular yaitu beberapa individu mungkin tidak mendapatkan sampel untuk berpartisipasi dalam turnamen sama sekali, sementara individu lain mungkin tidak dipilih untuk populasi baru karena individu tersebut telah kehilangan turnamen (Razali, 2011: 1135).

#### e. Pindah Silang (*Crossover*)

*Crossover* adalah operator dalam algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. *Crossover* 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 *crossing* dengan *Pc* (*Probability Crossover*) antara 0,6 sampai dengan 0,95. Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan pada keturunan (Michalewicz, 1996: 78). Penyilangan merupakan operator dalam algoritma genetika yang bertujuan untuk melahirkan kromosom baru yang mewarisi sifat-sifat induknya sebagaimana proses reproduksi yang terjadi dalam kehidupan alam. Dengan adanya operator ini proses pencarian yang dilakukan oleh algoritma genetika akan bergerak menuju titik-titik pencarian yang berbeda (Zainudin, 2014: 43).

Prinsip dari pindah silang ini adalah melakukan operasi pertukaran pada gen yang bersesuaian dari induk untuk menghasilkan individu baru.

Proses *crossover* dilakukan pada setiap individu dengan nilai probabilitas *crossover* yang telah ditentukan. Secara skematis, proses *crossover* seperti yang ditunjukkan Gambar 2.6. berikut :



Gambar 2.6. Sistematika proses *crossover*

Dari Gambar 2.6, 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.

Dalam penelitian ini dipilih penyilangan *order crossover* (OX) sebagai acuan dalam proses penyilangan. Teknik OX diawali dengan membangkitkan dua bilangan acak. Kemudian gen yang berada diantara kedua bilangan acak akan disalin ke keturunan (*offspring*) dengan posisi yang sama. Langkah berikutnya untuk mendapatkan keturunan pertama adalah

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. Selanjutnya, gen yang telah diurutkan dibandingkan dengan keturunan pertama. Apabila gen tersebut ada pada keturunan kedua maka gen tersebut diabaikan dari urutan. Kemudian masukkan urutan yang telah 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. Langkah tersebut juga berlaku untuk mencari keturunan kedua.

Contoh 2.2. *Order Crossover.*

Misalkan diketahui 2 induk yaitu :

Langkah 1 : Pilih *substring* dari sebuah induk secara acak.

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

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

Langkah 2 : Mambangkitkan sebuah *proto-child* dengan mengosongkan tempat *substring* Induk 2 pada Induk 1.

Protochild 1 = (x x 3 4 5 x 7 8 x)

Protochild 2 = (x x x 9 2 1 7 8 x)

Langkah 3 : Pindah gen dari *substring* pada tempat yang bersesuaian.

Protochild 1 = (7 8 x x x x 3 4 5)

Protochild 2 = (7 8 x x x x 9 2 1)

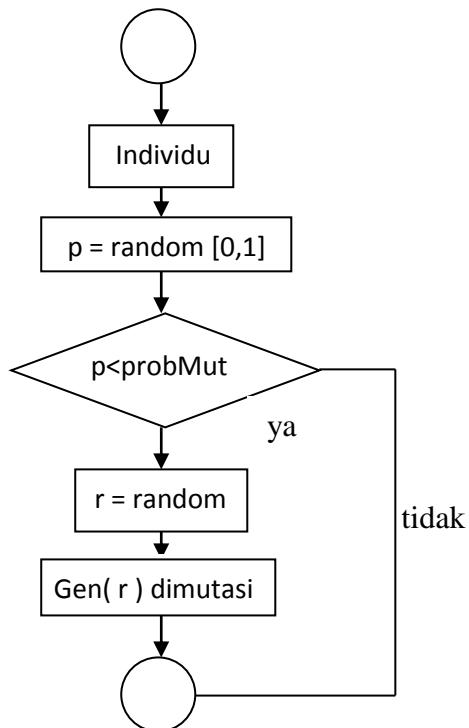
Langkah 4 : Tukar *substring* antara 2 induk.

Anak 1 = (7 8 6 9 2 1 3 4 5)

Anak 2 = (7 8 3 4 5 6 9 2 1)

#### f. Mutasi (*Mutation*)

Mutasi merupakan operator dalam algoritma genetika yang bertujuan untuk mengubah gen-gen tertentu dari sebuah kromosom. Proses ini dimodelkan sebagaimana yang terjadi dalam kehidupan alam. Probabilitas mutasi dari suatu gen biasanya dipilih sangat kecil, persis seperti kejadian sebenarnya dalam kehidupan alamiah yang memungkinkan terjadinya mutasi genetis tetapi dalam persentase yang sangat kecil. (Zainudin, 2014: 46). 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. Penekanan selektif memegang peranan penting. Jika dalam proses pemilihan kromosom-kromosom cenderung terus pada kromosom yang memiliki *fitness* yang tinggi saja, konvergensi akan sangat mudah terjadi (Murniati, 2009: 24). Secara skematis, proses mutasi ditunjukkan seperti Gambar 2.7. berikut :



Gambar 2.7. Sistematika Proses Mutasi

Dari Gambar 2.7. 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.

Dalam penelitian ini dipilih teknik *swapping mutation* sebagai acuan dalam proses mutasi. Teknik mutasi menggunakan *swapping mutation* diawali dengan memilih dua bilangan acak pertama ditukar dengan gen yang berada pada bilangan acak kedua dalam probabilitas tertentu (Suyanto, 2005: 67).

Contoh 2.3 *swapping mutation*.

Individu = (1 2 3 4 5 6 7 8 9)

Memindahkan 5 ke 2, sehingga diperoleh individu baru.

Individu = (1 5 3 4 2 6 7 8 9)

#### **g. *Elitism***

*Elitism* adalah proses untuk mempertahankan supaya individu yang mempunyai nilai *fitness* terbesar tetap ada selama proses evolusi (Kusumadewi, 2003: 112). Pemilihan individu yang akan diseleksi akan dilakukan secara *random* sehingga individu dengan nilai *fitness* tertinggi tidak selalu terpilih. Jika individu bernilai *fitness* tertinggi terpilih, ada kemungkinan individu tersebut akan rusak (nilai *fitness* menurun) karena proses pindah silang.

Komponen-komponen di atas akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Proses optimasi yang dilakukan dengan algoritma genetika akan berhenti setelah suatu syarat berhenti terpenuhi. Beberapa syarat berhenti yang biasa digunakan adalah batas nilai fungsi *fitness*, batas nilai fungsi objektif, batas waktu komputasi, banyak generasi dan terjadinya konvergensi (Zainudin, 2014: 48). Pada penelitian ini, proses optimasi akan berhenti jika banyak generasi yang diinginkan telah terpenuhi.

#### **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, proses pindah silang, proses mutasi pada populasi baru untuk membentuk populasi baru selanjutnya.

#### **E. Teknik Penarikan Kesimpulan**

Percobaan yang dilakukan dengan menggunakan *software* Matlab memberikan hasil total waktu yang berbeda-beda. Penelitian ini akan membandingkan total waktu yang dihasilkan antara *roulette wheel selection* dan seleksi turnamen. Selanjutnya akan diuji perbedaan antara rata-rata total waktu yang dibutuhkan untuk proses pendistribusian pada metode *roulette wheel selection* dan metode seleksi turnamen. Sebelum dilakukan uji beda rata-rata, perlu dilakukan uji normalitas dan homogenitas. Jika data berdistribusi normal dan memiliki variansi yang homogen, maka data tersebut termasuk statistika parametrik sehingga uji beda rata-rata dapat dilakukan dengan uji T, sedangkan jika penaksiran dan pengujian hipotesis data berdistribusi tidak normal maka data tersebut termasuk ke dalam statistika non parametrik (Asni dkk, 2012 :87).

#### **F. Penelitian yang Relevan**

Seiring berkembangnya jaman dan teknologi, banyak penelitian tentang algoritma genetika yang telah dilakukan, antara lain “*Algoritma Genetika pada Penyelesaian Capacitated Vehicle Routing Problem (Optimasi Rute Pendistribusian Aqua Galon PT. Tirta Investama)*” oleh Adam Arif

Dirgantara. Hasil dari penelitian ini diperoleh 7 rute pendistribusian dengan nilai *fitness*nya 0,006628 dan total jarak yang ditempuh yaitu 150,9 Km.

Selain itu, penelitian dari Ikhsan Hidayat yang berjudul “*Penerapan Algoritma Genetika pada Penyelesaian Capacitated Vehicle Routing Problem (CVRP) untuk Distribusi Surat Kabar Kedaulatan Rakyat di Kabupaten Sleman*”. Hasil dari penelitian ini diperoleh 2 rute pendistribusian. Pada rute pertama kapasitas kendaraan pengangkut adalah 336,2 kg dengan total jarak tempuhnya 89,7 Km dan pada rute kedua kapasitas kendaraan pengangkut adalah 319,1 kg dengan total jarak tempuhnya 44 Km.

Terdapat persamaan dan perbedaan penelitian ini dengan penelitian-penelitian sebelumnya. Persamaan dalam penelitian ini dengan penelitian dari Ikhsan Hidayat dan Adam Arif Dirgantara yaitu menggunakan algoritma genetika dengan metode *order crossover*, dan mutasi *sweep*. Perbedaan dari penelitian ini dengan penelitian sebelumnya yaitu dalam penggunaan seleksi serta kendala. Dalam penelitian ini, digunakan dua metode seleksi yaitu : *roulette wheel selection*, dan seleksi turnamen. Serta kendala yang digunakan dalam penelitian ini yaitu kapasitas dan waktu.