BAB II

LANDASAN TEORI

2.1 Sistem

2.1.1 Pengertian Sistem

Definisi Sistem menurut Al-Bahra dalam bukunya yang berjudul *Analisis dan desain Sistem Informasi* menjelaskan bahwa "sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai satu tujuan"(Al-Bahra, 2005: 3). Definisi lain dari Sistem menurut Jogiyanto dalam bukunya *Analisis Desain* menjelaskan bahwa "sistem adalah suatu jaringan kerja dari prosedur–prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu"(Jogiyanto, 2005: 1).

2.1.2 Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu : tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta lingkungan. Berikut penjelasan mengenai elemen-elemen yang membentuk sebuah sistem :

a. Tujuan

   Setiap sistem memiliki tujuan (*Goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.

b. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa pelanggan).

c. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lbih bernilai, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

d. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

e. Batas

Yang disebut batas (*boundary*) sistem adalah pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh, tim sepakbola mempunyai aturan permainan dan keterbatasan kemampuan pemain.

Pertumbuhan sebuah toko kelontong dipengaruhi oleh pembelian pelanggan, gerakan pesaing dan keterbatasan dana dari bank. Tentu saja batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku sistem. Sebagai contoh, dengan menjual saham ke publik, sebuah perusahaan dapat mengurangi keterbasatan dana.

f. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

g. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan supaya tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem.

2.1.3 Karakterisitik Sistem

a. Memiliki komponen ;

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem

dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak perduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut supra sistem, misalnya suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya.

b. Batas sistem (*boundary*) ;

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan luar sistem (*environment*) ;

Adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung sistem (*interface*) ;

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

e. Masukan sistem (*input*) ;

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah maintanance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

f. Keluaran sistem *(Output*) ;

Merupakan hasil dari energi yang diolah oleh sistem.

g. Pengolah sistem (*Process*) ;

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

h. Sasaran sistem ;

Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

2.1.4 Klasifikasi Sistem

a. Sistem abstrak ;

b. sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teologia)

c. Sistem fisik ; merupakan sistem yang ada secara fisik (sistem komputer, sistem akuntansi, sistem produksi dll.)

d. Sistem alamiah ; sistem yang terjadi melalui proses alam. (sistem matahari, sistem luar angkasa, sistem reproduksi dll.
e. Sistem buatan manusia ; sistem yang dirancang oleh manusia.
f. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human-machine system* (contoh ; sistem informasi)
g. Sistem Tertentu (*deterministic system*) ; beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (contoh ; sistem komputer)
h. Sistem tak tentu (*probabilistic system*) ; sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
i. Sistem tertutup (*close system*) ; sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).
j. Sistem terbuka (*open system*) ; sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.
k. Sistem sederhana dan Sistem kompleks.

2.2 Informasi

Definisi Informasi menurut Jogiyanto dalam bukunya *Analisis dan Desain* menjelaskan bahwa "informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya" (Jogiyanto, 2005:8). Definisi lain dari informasi Menurut Aji Supriyanto dalam bukunya *Pengantar Teknologi Informasi* yang menyebutkan bahwa **"**informasi adalah data yang telah diolah menjadi bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang" (Aji Supriyanto, 2005 : 243).

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dalam kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu.

Untuk memperoleh informasi yang berguna, tindakan yang pertama adalah mengumpulkan data, kemudian mengolahnya sehingga menjadi informasi. Dari data-data tersebut informasi yang didapatkan lebih terarah dan penting karena telah dilalui berbagai tahap dalam pengolahannya diantaranya yaitu pengumpulan data, data apa yang terkumpul dan menemukan informasi yang diperlukan.

Kualitas dari suatu informasi tergantung dati tiga hal, yaitu informasi harus akurat, tepat pada waktunya dan relevan. Berguna atau tidaknya informasi tergantung pada beberapa aspek, yaitu:

1. *Tujuan si penerima*

Apabila informasi itu tujuannya untuk memberikan bantuan maka informasi itu harus membantu si penerima dalam usahanya untuk mendapatkannya.

2. *Ketelitian penyampaian dan pengolahan data*

Penyampaian dan mengolah data, inti dan pentingnya info harus dipertahankan.

3. *Waktu*

Informasi yang disajikan harus sesuai dengan perkembangan informasi itu sendiri.

4. *Ruang dan Tempat*

Informasi yang didapat harus tersedia dalam ruangan atau tempat yang tepat agar penggunaannya lebih terarah bagi si pemakai.

5. *Bentuk*

Dalam hubungannya bentuk informasi harus disadari oleh penggunaannya secara efektif, hubungan-hubungan yang diperlukan, kecenderungan-kecenderungan dan bidang-bidang yang memerlukan perhatian manajemen serta menekankan informasi tersebut ke situasi-situasi yang ada hubungannya.

6. *Semantik*

Agar informasi efektif, informasi harus ada hubungannya antara kata-kata dan arti yang cukup jelas dan menghindari kemungkinan salah tafsir. Jelaslah bahwa agar informasi itu menjadi berguna harus disampaikan kepada orang yang tepat, pada waktu yang tepat, dan dalam bentuk yang tepat pula.

2.2.1 Fungsi Informasi

Menurut Edhy Sutanta (2003:11) Suatu informasi mempunyai beberapa fungsi, antara lain:

a. Menambah pengetahuan

Adanya informasi akan menambah pengetahuan bagi penerimanya yang dapat digunakan sebagai bahan pertimbangan yang mendukung proses pengambilan keputusan.

b. Mengurangi ketidakpastian

Adanya informasi akan mengurangi ketidakpastian karena apa yang akan terjadi dapat diketahui sebelumnya, sehingga menghindari keraguan pada saat pengambilan keputusan.

c. Mengurangi resiko kegagalan

Adanya informasi akan resiko kegagalan karena apa yang akan terjadi dapat diantisipasi dengan baik, sehingga kemungkinan terjadinya kegagalan akan dapat dikurangi dengan pengambilan keputusan yang tepat.

d. Mengurangi keanekaragaman/variasi yang tidak diperlukan

Adanya informasi akan mengurangi keanekaragaman yang tidak diperlukan, karena keputusan yang diambil lebih terarah.

e. Memberi standar, aturan-aturan, ukuran-ukuran, dan keputusan-keputusan yang menentukan pencapaian sasaran dan tujuan

Adanya informasi akan memberikan standar, aturan, ukuran, dan keputusan yang lebih terarah untuk mencapai sasaran dan tujuan yang telah ditetapkan secara lebih baik berdasakan informasi yang diperoleh.

2.2.2 Nilai Informasi

Menurut Edhy Sutanta (2003:13) Nilai suatu informasi dapat ditentukan berdasarkan sifatnya. Tentang 10 sifat yang dapat menentukan nilai informasi, yaitu sebagai berikut:

1. Kemudahan dalam memperoleh

   Informasi dapat diperoleh dengan mudah jika sistem dilengkapi oleh basis data dan bagian pengolah yang mampu mengolah data dengan baik untuk memenuhi segala kebutuhan informasi secara mudah.

2. Sifat luas dan kelengkapannya

   Informasi mempunyai nilai yang lebih sempurna apabila mempunyai lingkup/cakupan yang luas dan lengkap. Informasi sepotong dan tidak lengkap menjadi tidak bernilai, karena tidak dapat digunakan secara baik. Sifat luas dan lengkap tersebut memerlukan dukungan basis data yang cukup lengkap dan terstruktur dengan baik.

3. Ketelitian (*accuracy*)

   Informasi mempunyai nilai yang lebih sempurna apabila mempunyai ketelitian yang sangat tinggi/akurat. Informasi menjadi tidak bernilai jika tidak akurat, karena akan mengakibatkan kesalahan pengambilan keputusan.

4. Kecocokan dengan pengguna (*relevance*)

   Informasi mempunyai nilai yang lebih sempurna apabila sesuai dengan kebutuhan penggunanya. Informasi berharga dan penting menjadi tidak

bernilai jika tidak sesuai dengan kebutuhan penggunanya, karena tidak dapat dimanfaatkan untuk pengambilan keputusan.

5. Ketepatan waktu

   Informasi mempunyai nilai yang lebih sempurna apabila dapat diterima oleh pengguna pada saat yang tepat. Informasi berharga dan penting menjadi tidak bernilai jika terlambat diterima/usang, karena tidak dapat dimanfaatkan pada saat pengambilan keputusan.

6. Kejelasan (*clarity*)

   Informasi yang jelas akan meningkatkan kesempurnaan nilai informasi. Kejelasan informasi dipengaruhi oleh bentuk dan format informasi. Dibandingkan dengan bentuk teks atau deskriptif, informasi dalam bentuk tabel atau grafik banyak menjadi pilihan, karena dapat dibaca dan dipahami dengan lebih mudah.

7. Fleksibilitas / keluwesannya

   Nilai informasi semakin sempurna apabila memiliki fleksibilitas tinggi. Fleksibilitas informasi diperlukan oleh para manajer/pimpinan pada saat pengambilan keputusan

8. Dapat dibuktikan

   Nilai informasi semakin sempurna apabila informasi tersebut dapat dibuktikan kebenarannya. Kebenaran informasi bergantung pada validitas data sumber yang diolah.

9. Tidak ada prasangka

   Nilai informasi semakin sempurna apabila informasi tersebut tidak menimbulkan prasangka dan keraguan adanya kesalahan informasi.

10. Dapat diukur

    Informasi untuk pengambilan keputusan seharusnya dapat diukur agar dapat mencapai nilai yang sempurna. Pengukuran informasi pada umumnya dimaksudkan untuk mengukur dan melacak kembali validitas data sumber yang digunakan.

2.3 Sistem Informasi

Definisi Sistem Informasi menurut Al-Bahra dalam bukunya yang berjudul *Analisis dan desain Sistem Informasi* menjelaskan bahwa "sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi" (Al-Bahra 2005 :13). Adapun definisi lain dari sistem informasi menurut Jogiyanto dalam bukunya *Analisis dan Desain* adalah sebagi berikut:

> "Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan stategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan–laporan yang di perlukan" (jogiyanto 2005:11).

Berdasarkan kedua definisi diatas penulis dapat menyimpulkan bahwa sistem informasi adalah sebuah kumpulan komponen sistem yang saling berhubungan untuk mencapai suatu tujuan untuk mendukung operasi manajerial dan kegiatan suatu stategi dalam suatu organisasi.

2.4 Algoritma

Definisi algoritma menurut Rinaldi Munir dalam bukunya yang berjudul algoritma dan pemrograman dalam bahasa Pascal dan C dijelaskan bahwa" Algoritma adalah urutan langkah-langkah untuk memecahkan masalah." Pada tugas akhir ini penulis menggunakan salah satu metode algoritma juga dalam proses penjadwalan, metode yang di pakai oleh penulis adalah metode algoritma genetika.

2.4.1 Pengertian Algoritma Genetika

Algoritma ini ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, David Goldberg (1989). Dimana mendefenisikan algoritma genetika ini sebagai metode algoritma pencarian berdasarkan pada mekanisme seleksi alam dan genetika alam.

Algoritma genetika adalah algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan masalah (problem solving). Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu

kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness*. Dan menurut Suyanto dalam bukunya yg berjudul *Artificial Intelligence Searching, Reasoning, Planning and Learning*, disebutkan bahwa definisi dari algoritma genetika adalah sebagai berikut: "Algoritma Genetika adalah algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah."(Suyanto 2007: 205)

Generasi ini akan merepresentasikan perbaikan-perbaikan pada populasi awalnya. Dengan melakukan proses ini secara berulang, algoritma ini diharapkan dapat mensimulasikan proses evolusioner.

Pada akhirnya, akan didapatkan solusi-solusi yang paling tepat bagi permasalahan yang dihadapi. Untuk menggunakan algoritma genetika, solusi permasalahan direpresentasikan sebagai khromosom. Tiga aspek yang penting untuk penggunaan algoritma genetika:

1. Defenisi fungsi *fitness*
2. Defenisi dan implementasi representasi genetika
3. Defenisi dan implementasi operasi genetika.

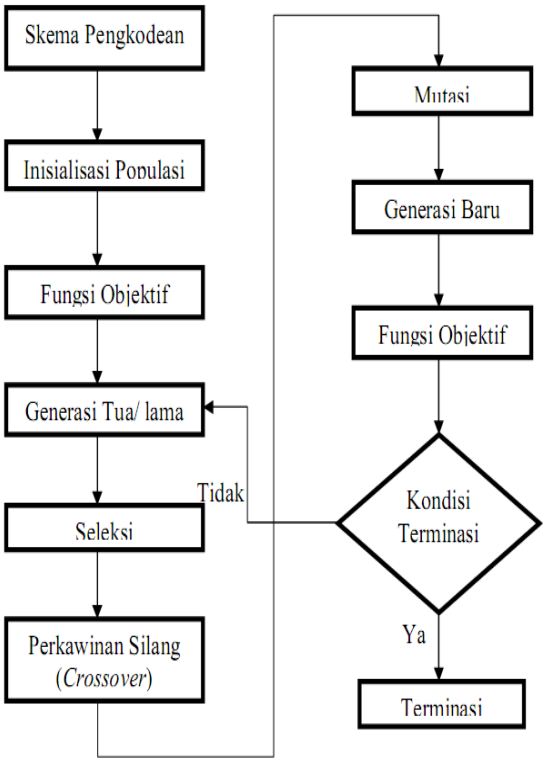
Jika ketiga aspek di atas telah didefinisikan, algoritma genetika akan bekerja dengan baik. Tentu saja, algoritma genetika bukanlah solusi terbaik untuk memecahkan segala masalah. Sebagai contoh, metode tradisional telah diatur untuk untuk mencari penyelesaian dari fungsi analitis convex yang "berperilaku baik" yang variabelnya sedikit. Pada kasus-kasus ini, metode berbasis kalkulus lebih unggul dari algoritma genetika karena metode ini dengan cepat menemukan

solusi minimum ketika algoritma genetika masih menganalisa bobot dari populasi awal.

Untuk masalah-masalah ini pengguna harus mengakui fakta dari pengalaman ini dan memakai metode tradisional yang lebih cepat tersebut. Akan tetapi, banyak persoalan realistis yang berada di luar golongan ini. Selain itu, untuk persoalan yang tidak terlalu rumit, banyak cara yang lebih cepat dari algoritma genetika. Jumlah besar dari populasi solusi, yang merupakan keunggulan dari algoritma genetik, juga harus mengakui kekurangannya dalam dalam kecepatan pada sekumpulan komputer yang dipasang secara seri-*fitness* function dari tiap solusi harus dievaluasi. Namun, bila tersedia komputer- komputer yang paralel, tiap prosesor dapat mengevaluasi fungsi yang terpisah pada saat yang bersamaan. Karena itulah, algoritma genetika sangat cocok untuk perhitungan yang paralel.

2.4.2 Struktur Umum Algoritma Genetika

Algoritma genetika memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetik, pembentukan kromosom baru serta seleksi alami seperti yang terjadi pada makhluk hidup. Algoritma Genetika secara umum dapat diilustrasikan dalam diagram alir berikut ini:



Gambar 2.1 Diagram Alir Algoritma Genetika

Golberg (1989) mengemukakan bahwa algoritma genetika mempunyai karakteristik-karakteristik yang perlu diketahui sehingga dapat terbedakan dari prosedur pencarian atau optimasi yang lain, yaitu:

1. Algoritma genetika dengan pengkodean dari himpunan solusi permasalahan berdasarkan parameter yang telah ditetapkan dean bukan parameter itu sendiri.
2. Algoritma genetika pencarian pada sebuah solusi dari sejumlah individu-individu yang merupakan solusi permasalahan bukan hanya dari sebuah individu.
3. Algoritma genetika informasi fungsi objektif (*fitness*), sebagai cara untuk mengevaluasi individu yang mempunyai solusi terbaik, bukan turunan dari suatu fungsi.

4. Algoritma genetika menggunakan aturan-aturan transisi peluang, bukan aturan-aturan deterministik.

Variabel dan parameter yang digunakan pada algoritma genetika adalah:

1. Fungsi *fitness* (fungsi tujuan) yang dimiliki oleh masing-masing individu untuk menentukan tingkat kesesuaian individu tersebut dengan kriteria yang ingin dicapai.
2. Populasi jumlah individu yang dilibatkan pada setiap generasi.
3. Probabilitas terjadinya persilangan (*crossover*) pada suatu generasi.
4. Probabilitas terjadinya mutasi pada setiap individu.
5. Jumlah generasi yang akan dibentuk yang menentukan lama penerapan algoritma genetika.

Secara umum struktur dari suatu algoritma genetika dapat mendefenisikan dengan langkah-langkah sebagai berikut:

1. Membangkitkan populasi awal

Populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal. Populasi itu sendiri terdiri atas sejumlah kromosom yang merepresentasikan solusi yang diinginkan.

2. Membentuk generasi baru

Untuk membentuk generasi baru, digunakan operator reproduksi seleksi, *crossover* dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru dimana generasi

baru ini merupakan representasi dari solusi baru. Generasi baru ini dikenal dengan istilah anak (*offspring*).

3. Evaluasi solusi

Pada tiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dinamakan *fitness*. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut.

Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. Beberapa kriteria berhenti sering digunakan antara lain: berhenti pada generasi tertentu, berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi tidak berubah, berhenti dalam n generasi tidak didapatkan nilai *fitness* yang lebih tinggi.

Fungsi *fitness* tersebut sebagai berikut:

$$Fitness = \frac{1}{1 + Penalti}$$

dimana:

$$penalty = \sum \text{Bp} \sum \text{Np}$$

Dari persamaan diatas nilai *fitness* ditentukan oleh nilai *penalty* . *Penalty* tersebut menunjukkan jumlah pelanggaran kendala pada suatu kromosom. Semakin tinggi nilai *fitness* akan semakin besar kemungkinan kromosom tersebut terpilih ke

generasi berikutnya. Jadi nilai *penalty* berbanding terbalik dengan nilai *fitness*, semakin kecil nilai penalty (jumlah pelanggaran) semakin besar nilai *fitness* nya.

Jadi fungsi *fitness* :

$$Fitness = \frac{1}{1 + \sum Bp \sum Np}$$

Keterangan:

Bp : Bobot Pelanggaran

Np : Indikator Pelanggaran

2.4.3 Pengkodean

Pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatu kunci pokok persoalan ketika menggunakan algoritma genetika.

Berdasarkan jenis symbol yang digunakan sebagai nilai suatu gen, metode pengkodean dapat diklasifikasikan sebagai berikut:

1. Pengkodean biner merupakan cara pengkodean yang paling umum digunakan karena adalah yang pertama kali digunakan dalam algoritma genetika oleh Holland. Keuntungan pengkodean ini adalah sederhana untuk diciptakan dan mudah dimanipulasi. Pengkodean biner memberikan banyak kemungkinan untuk kromosom walaupun dengan jumlah nilai-nilai yang mungkin terjadi pada suatu gen yang sedikit (0 dan 1). Di pihak lain, pengkodean biner sering tidak sesuai untuk banyak masalah dan kadang pengoreksian harus dilakukan setelah operasi *crossover* dan mutasi.

2. Pengkodean bilang riil adalah suatu pengkodean bilangan dalam bentuk riil. Masalah optimalisasi fungsi dan optimalisasi kendala lebih tepat jika diselesaikan dengan pengkodean bilangan riil karena struktur topologi ruang genotif untuk pengkodean bilangan riil identik dengan ruang fenotifnya, sehingga mudah membentuk operator genetika yang efektif dengan cara memakai teknik yang dapat digunakan yang berasal dari metode konvensional.

3. Pengkodean bilangan bulat adalah metode yang mengkodekan bilangan dalam bentuk bilangan bulat. Pengkodean ini baik digunakan untuk masalah optimisasi kombinatorial.

4. Pengkodean struktur data adalah model pengkodean yang menggunakan struktur data. Pengkodean ini digunakan untuk masalah kehidupan yang lebih kompleks seperti perencanaan jalur robot, dan masalah pewarnaan Graph.

2.4.4 Operator Genetika

Algoritma genetika merupakan proses pencarian yang heuristik dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetik dalam menemukan solusi optimum suatu masalah yang diberikan. Hal yang harus diperhatikan adalah menghindari terjadinya konvergensi premature, yaitu mencapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalah hasil optimum lokal.

Operator genetika yang digunakan setelah proses evaluasi tahap pertama membentuk populasi baru dari generasi sekarang. Operator-operator tersebut adalah operator seleksi, *crossover* dan mutasi.

2.4.4.1 Seleksi

Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit. Langkah pertama dalam seleksi ini adalah pencarian nilai *fitness* . Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya (Kusumadewi, 2003).

Kemampuan algoritma genetika untuk memproduksi kromosom yang lebih baik secara progresif tergantung pada penekanan selektif (*selective pressure*) yang diterapkan ke populasi. Penekanan selektif dapat diterapkan dalam dua cara. Cara pertama adalah membuat lebih banyak kromosom anak yang dipelihara dalam populasi dan memilih hanya kromosom-kromosom terbaik bagi generasi berikut. Walaupun orang tua dipilih secara acak, metode ini akan terus menghasilkan kromosom yang lebih baik berhubungan dengan penekanan selektif yang diterapkan pada individu anak tersebut. Cara lain menerapkan penekanan selektif adalah memilih orang tua yang lebih baik ketika membuat keturunan baru. Dengan metode ini, hanya kromosom sebanyak yang dipelihara dalam populasi yang perlu dibuat bagi generasi berikutnya. Walaupun penekanan selektif tidak diterapkan ke level keturunan, metode ini akan terus menghasilkan kromosom yang lebih baik, karena adanya penekanan selektif yang diterapkan ke orangtua.

Ada beberapa metode untuk memilih kromosom yang sering digunakan antara lain adalah seleksi roda rolet (*roulette wheel selection*), seleksi ranking (*rank selection*) dam seleksi turnamen (*tournament selection*).

Dalam penelitian ini, metode yang digunakan adalah seleksi roda rolet (*roulette wheel selection*). Pada seleksi ini, orang tua dipilih berdasarkan *fitness* mereka. Lebih baik kualitas suatu kromosom, lebih besar peluangnya untuk terpilih. Probabilitas suatu individu terpilih untuk *crossover* sebanding dengan *fitness* -nya. Cara penyeleksian ini merupakan peniruan dari permainan roda rolet.

2.4.4.2 *Crossover*

Crossover (perkawinan silang) bertujuan menambah keanekaragaman string dalam populasi dengan penyilangan antar-string yang diperoleh dari sebelumnya. Beberapa jenis *crossover* tersebut adalah:

1. *Crossover* 1-titik

Pada *crossover* dilakukan dengan memisahkan suatu string menjadi dua bagian dan selanjutnya salah satu bagian dipertukarkan dengan salah satu bagian dari string yang lain yang telah dipisahkan dengan cara yang sama. Proses yang demikian dinamakan operator *crossover* satu titik seperti diperlihatkan pada gambar berikut:

Tabel 2.1 Contoh *Crossover* 1-titik

Kromosom Orangtua 1	**11001**011
Kromosom Orangtua 2	11011**111**
Keturunan	**11001111**

2. *Crossover* 2-titik

Proses *crossover* ini dilakukan dengan memilih dua titik *crossover*. Kromosom keturunan kemudian dibentuk dengan barisan bit dari awal kromosom sampai titik *crossover* pertama disalin dari orangtua pertama, bagian dari titik *crossover* pertama dan kedua disalin dari orangtua kedua, kemudian selebihnya disalin dari orangtua pertama lagi.

Tabel 2.2 Contoh *Crossover* 2-titik

Kromosom Orangtua 1	**11**001**11**
Kromosom Orangtua 2	11**011**11
Keturunan	**11011111**

3. *Crossover* seragam

Crossover seragam manghasilkan kromosom keturunan dengan menyalin bit-bit secara acak dari kedua orangtuanya.

Tabel 2.3 Contoh *Crossover* seragam

Kromosom Orangtua 1	**110**01**11**
Kromosom Orangtua 2	110**11**11
Keturunan	**11011111**

2.4.4.3 Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi *crossover* 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 pada kromosom yang memiliki *fitness* yang tinggi saja, konvergensi *premature* , yaitu mencapai solusi yang optimal lokal sangat mudah terjadi. Untuk menghindari konvergensi premature tersebut dan tetap menjaga perbedaan (*diversity*) kromosom-kromosom dalam populasi, selain melakukan penekanan selektif yang lebih efisien, operator mutasi juga dapat digunakan.

Proses mutasi dalam system biologi berlangsung dengan mengubah isi allele gen pada suatu locus dengan allele yang lain. Proses mutasi ini bersifat acak sehingga tidak selalu menjamin bahwa setelah proses mutasi akan diperoleh kromosom dengan*fitness* yang lebih baik.

Operator mutasi merupakan operasi yang menyangkut satu kromosom tertentu. Beberapa cara operasi mutasi diterapkan dalam algoritma genetika menurut jenis pengkodean terhadap *phenotype* , antara lain:

1. Mutasi dalam Pengkodean Biner

Mutasi pada pengkodean biner merupakan operasi yang sangat sederhana. Proses yang dilakukan adalah menginversi nilai bit pada posisi tertentu yang terpilih secara acak (atau menggunakan skema tertentu) pada kromosom, yang disebut *inverse bit*.

Tabel 2.4 Contoh Mutasi pada pengkodean biner

Kromosom sebelum mutasi	10010**1**11
Kromosom setelah mutasi	10010**0**11

2. Mutasi dalam Pengkodean Permutasi

Proses mutasi yang dilakukan dalam pengkodean biner dengan mengubah langsung bit-bit pada kromosom tidak dapat dilakukan pada pengkodean permutasi karena konsistensi urutan permutasi harus diperhatikan. Salah satu cara yang dapat dilakukan adalah dengan memilih dua posisi (*locus*) dari kromosom dan kemudian nilainya saling dipertukarkan.

Tabel 2.5 Contoh Mutasi pada pengkodean permutasi

Kromosom sebelum mutasi	123465879
Kromosom setelah mutasi	127465839

3. Mutasi dalam Pengkodean Nilai

Mutasi pada pengkodean nilai hampir sama dengan yang dilakukan pada pengkodean biner, tetapi yang dilakukan bukan menginversi nilai bit. Penerapannya bergantung pada jenis nilai yang digunakan. Sebagai contoh untuk nilai riil, proses mutasi dapat dilakukan seperti yang dilakukan pada pengkodean permutasi, dengan saling mempertukarkan nilai dua gen pada kromosom.

4. Mutasi dalam Pengkodean Pohon

Mutasi dalam pengkodean pohon dapat dilakukan antara lain dengan cara mengubah operator (+, -, *, /) atau nilai yang terkandung pada suatu verteks pohon yang dipilih. Atau, dapat juga dilakukan dengan memilih dua verteks dari

pohon dan saling mempertukarkan operator atau nilainya. Tidak setiap gen selalu dimutasi tetapi mutasi dikontrol dengan probabilitas tertentu yang disebut dengan *mutation rate* (probabilitas mutasi) dengan notasi Pm. Jenis operator mutasi antara lain:

1. Mutasi Terarah

Mutasi terarah tergantung dari informasi gen. Informasi gen tersebut berupa nilai pelanggaran gen (*violation gen*). Ini berarti bahwa setiap gen mempunyai peluang yang berbeda untuk terjadi mutasi. Gen yang mempunyai nilai pelanggaran yang lebih besar maka gen tersebut mempunyai peluang untuk terjadi mutasi. Mutasi ini menghubungkan nilai pelanggaran relatif (nilai pelanggaran suatu gen dibagi dengan nilai pelanggaran total suatu kromosom) dengan probabilitas terjadinya mutasi dari suatu gen pada kromosom. Hubungan tersebut dinyatakan secara matematis sebagai berikut:

$$nr(i) = \frac{n(i)}{1 + n_{total}}$$

$$pm(i) = (1 + nr(i))^2 \, pm$$

Keterangan persamaan:

nr(i) : nilai pelanggaran relative gen ke-i

n_{total} : nilai pelanggaran total kromosom

pm(i) : probabilitas mutasi gen ke-i

pm : probabilitas mutasi

2. Mutasi Biasa

Mutasi biasa tidak tergantung dari informasi gen. Setiap gen mempunyai peluang yang sama untu terjadi mutasi.

2.5 Sekilas Tentang Java

Java merupakan sebuah bahasa pemrograman berorientasi objek yang dikembangkan oleh *Sun Microsystem*. Dalam mendiskusikan Java, kiranya penting sekali untuk membedakan bahasa pemrograman Java, *Java Virtual Machine*, dan platform Java.

Bahasa pemrograman Java adalah bahasa bahasa yang digunakan untuk menghasilkan aplikasi-aplikasi Java. Pada saat program Java dikompilasi, ia akan dikonversi ke bentuk *bytecode*, yang merupakan bahasa mesin yang *portable*. Selanjutnya, *bytecode* tersebut dijalankan di *Java Virtual Machine* (JVM). Platform dapat didefinisikan sebagai perangkat lunak pendukung untuk aktivitas-aktivitas tertentu yang merupakan himpunan kelas-kelas Java yang sudah didefinisikan sebelumnya dan eksis sejak instalasi Java. Platform Java juga mengacu pada lingkungan *runtime* atau API (*Application Programming Interface*) Java.

Untuk membuat aplikasi berbasis Java, diperlukan Java Development Kit (JDK). Saat ini JDK telah mencapai versi 2 yang dikenal dengan Java 2 Platform. Java 2 Platform, dibagi menjadi tiga kategori, yaitu :

1. *Java 2 Standard Edition* (J2SE)

Kategori ini digunakan untuk mengembangkan dan menjalankan aplikasi Java berbasis PC.

2. *Java 2 Enterprise Edition* (J2EE)

   Kategori ini digunakan untuk mengembangkan dan menjalankan aplikasi Java pada lingkungan enterprise, dengan fungsi-fungsi seperti Servlet dan Java Server JSP).

3. *Java 2 Micro Edition* (J2ME)

   Kategori ini digunakan untuk mengembangkan dan menjalankan aplikasi Java berbasis *handheld devices*, seperti Personal Data Assistant (PDA), HandPhone dan PocketPC.

4. PROGRAM JAVA

   Ada dua macam pemrograman yang dapat dibuat menggunakan bahasa pemrograman Java, yaitu :

   a. Applet Java – program yang disisipkan dalam halaman Web.
   b. Aplikasi Java – program stand alone pada umumnya dibagi menjadi dua macam, yaitu aplikasi konsol dan aplikasi window. Aplikasi konsol hanya mendukung output karakter pada screen komputer. Sedangkan aplikasi window dapat membuat dan mengatur banyak window, seperti penggunaan menu, toolbar, dialog, dan lain-lain.

(Sumber : *http://ffaaddiillaahh.wordpress.com/2010/01/31/sekilas-tentang-java/*)

2.6 Sekilas Tentang *MySQL*

MySQL adalah sebuah *system manajemen database*. Database adalah sekumpulan data yang terstruktur. Data-data itu dapat suatu daftar belanja yang sangat sederhana sampai ke galeri lukisan atau banyaknya jumlah informasi pada jaringan perusahaan. Untuk menambah, mengakses dan memproses data yang tersimpan pada database komputer, kita membutuhkan manajemen database seperti *MySQL*.

MySQL adalah sebuah *system manajemen database* yang saling berhubungan. Sebuah hubungan databse dari data yang tersimpan pada table yang terpisah daripada menyimpan semua data pada ruang yang sangat besar. Hal ini menambah kecepatan dan fleksibilitas. Tabel-tabel tersebut dihubungkan oleh hubungan yang sudah didefinisikan mengakibatkan akan memungkinkan untuk mengkombinasikan data dari beberapa tabel sesuai dengan keperluan.

Awalnya *MYSQL* dijalankan pada sistem operasi Unix dan Linux. Tapi, para penggemar *Windows* pun sekarang sudah tersedia *MYSQL* versi windows. Mereka yang menggunakan *Linux* biasanya *MYSQL* sudah terinstall secara *default*. Sedikit menjelaskan sejarah tentang *MYSQL*, pada tahun 1995 pengembangan mulai dilakukan, kemudian *direlease* sendiri pada tahun 1996. Antara tahun 1998-2000 *MYSQL* memperoleh penghargaan dan mendapatkan GNU Publik License.

(Sumber : *http://landfray.blogspot.com/2011/06/sekilas-tentang-mysql.html*)