

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem dan Informasi

2.1.1 Sistem

Menurut Sutabri (2004), bahwa sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lainnya berfungsi untuk mencapai tujuan tertentu. Selain itu Jogiyanto (2005), bahwa sistem adalah kumpulan dari komponen atau elemen yang saling berhubungan satu dengan yang lain membentuk satu kesatuan untuk mencapai tujuan tertentu.

2.1.2 Informasi

Menurut Davis dalam Ladjamudin (2005) informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimannya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. Sedangkan menurut McLeod dalam Ladjamudin (2005) informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti bagi penerimanya. Sumber informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian dan kesatuan nyata. Kejadian adalah sesuatu yang terjadi pada saat tertentu. Informasi diperoleh setelah data mentah diproses atau diolah.

2.2 Sistem Pendukung Keputusan

Pada dasarnya pengambilan keputusan adalah suatu pendekatan sistematis pada hakekat suatu masalah, pengumpulan fakta-fakta, dan pengambilan tindakan yang menurut perhitungan merupakan tindakan yang paling tepat Kadarsah

(2000). Namun seiring perkembangannya sistem pengambilan keputusan juga diterapkan dalam pemecahan masalah yang tidak terstruktur. Hal ini membuat sistem pendukung keputusan membutuhkan dukungan seperangkat sistem yang mampu memecahkan persoalan secara efektif dan efisien. Pengambilan keputusan didasarkan pada pemilihan dari berbagai alternatif tindakan yang mungkin dipilih yang prosesnya melalui mekanisme tertentu, dengan harapan akan menghasilkan keputusan yang terbaik.

Keputusan yang diambil merupakan sebuah kesimpulan yang dicapai setelah dilakukan pertimbangan yang terjadi setelah satu kemungkinan dipilih, sementara hal lain dikesampingkan. Pada akhirnya dapat dikatakan bahwa setiap kemungkinan itu bertoak dari beberapa kemungkinan atau alternatif untuk dipilih. Setiap alternatif tersebut berbeda satu dengan yang lainnya mengacu pada perbedaan konsekuensi yang ditimbulkan setelah dipilihnya alternatif tersebut.

Menurut Simon (1960) yang dikutip dari Kadarsah (2000), dalam sistem pendukung keputusan, proses pengambilan keputusan terdiri dari 3 fase, yaitu:

1. Intelligence

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkungan problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

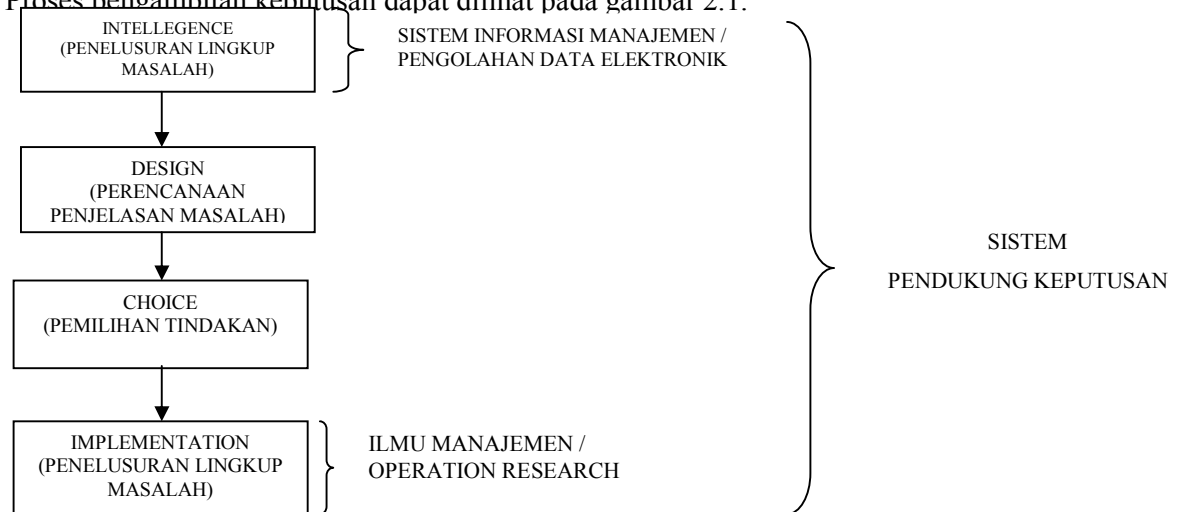
2. Design

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi proses untuk memahami masalah, menurunkan solusi dan menguji kelayakan solusi.

3. Choice

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

Proses pengambilan keputusan dapat dilihat pada gambar 2.1.



Gambar 2.1 Fase Proses Pengambilan Keputusan

2.3 Konsep Dasar Penjadwalan

2.3.1 Pengertian Penjadwalan

Time Tabling adalah suatu kegiatan administratif utama pada sebagian besar intitusi. Kegiatan operasi institusi akan bergantung sepenuhnya pada jadwal (*timetable*) yang dibuat. Menurut Wren (1996) penjadwalan didefinisikan sebagai berikut :

1. Penjadwalan adalah pengalokasian sumber daya pada objek-objek yang ada pada ruang waktu dan bergantung pada kendala-kendala yang sedemikian sehingga sedapat mungkin memenuhi sekumpulan sasaran yang diinginkan. Secara sederhana, penjadwalan dapat diartikan

sebagai pengalokasian sumber-sumber daya yang tersedia pada ruang waktu yang ada sehingga memenuhi kondisi-kondisi tertentu.

2. Penjadwalan ini meliputi dua hal, yaitu tentang alokasi sarana dan tentang dapat diramalkannya waktu kegiatan. Sarana kegiatan, yaitu pembicara, sarana lainnya dan waktu, harus dialokasikan secara seimbang dan efisien ke dalam program bulanan dan kegiatan harian.
3. Penjadwalan dapat didefinisikan sebagai pengalokasian sumber daya dalam jangka waktu tertentu untuk melakukan serangkaian tugas. Penjadwalan adalah proses pengorganisasian, pemilihan, dan penentuan waktu penggunaan sumber-sumber untuk mengerjakan semua aktifitas yang diperlukan yang memenuhi kendala aktifitas dan sumber daya.

2.3.2. Tujuan Penjadwalan

Menurut (Ginting, 2009) mengidentifikasi beberapa tujuan dari aktifitas penjadwalan adalah sebagai berikut :

1. Meningkatkan penggunaan sumber daya atau mengurangi waktu tunggunya, sehingga total waktu proses dapat berkurang, dan produktifitas dapat meningkat.
2. Mengurangi beberapa keterlambatan pada pekerjaan yang mempunyai batas waktu penyelesaian sehingga meminimalisasi biaya kelambatan (*penalty cost*).

2.3.3. Performasi Penjadwalan

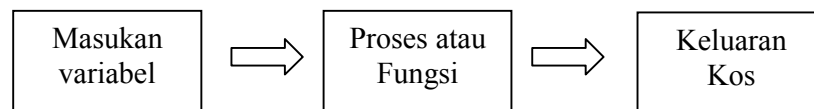
Menurut (Wahyuni, 2007) terdapat tiga tujuan pembuatan keputusan yang umum dalam penjadwalan dan ketiganya menunjukkan ukuran dasar performasi jadwal, yaitu :

1. Pemanfaatan sumber daya yang efisien : minimum maksimum saat selesai.
2. Respon yang cepat terhadap permintaan konsumen: minimum rata-rata saat selesai (*completion time*), minimum rata-rata. waktu tinggal (*flow time*), atau minimum rata-rata waktu tunggu (*waiting time*).
3. Sesuai dengan batas waktu yang ditentukan: minimum rata-rata keterlambatan (*tardiness*), minimum maksimum keterlambatan, dan minimum jumlah job yang terlambat (*the number of tardy jobs*).

2.4. Teknik Optimasi

2.4.1. Pengertian Optimasi

Menurut Haupt (2004), Optimasi adalah suatu proses yang berhubungan dengan penyesuaian masukan, pemilihan karakteristik peralatan, proses matematis dan pengujian yang dilakukan untuk menemukan keluaran optimum. Sebagaimana diperlihatkan dalam gambar 2.2, masukan terdiri atas variable-variabel, proses atau fungsi disebut sebagai fungsi kos, fungsi tujuan atau fungsi fitness, dan keluaran adalah kos atau fitness.



Gambar 2.2 Diagram optimasi fungsi atau optimasi proses

2.4.2 Klasifikasi Optimasi

Proses optimasi dapat diklarifikasi menjadi enam kategori (Haupt dan Haupt,2004) Pembagian tersebut tidak dapat dianggap sepenuhnya membagi-bagi proses optimasi persis menjadi enam kategori, karena setiap cabang saling dapat terkait. Misalnya, masalah optimasi dinamis dapat saling terkait dengan kendala (constraint) atau tanpa kendala, melibatkan variable yang diskret atau malar (continuous). Keenam kategori yang dimaksud adalah:

1. Optimasi dengan cara *trial and error* berhubungan dengan proses untuk menyesuaikan nilai variable masukan yang membuat perubahan keluaran tanpa pengetahuan yang banyak mengenai prosesnya.
2. Optimasi satu dimensi adalah optimasi yang melibatkan sebuah variable sedangkan optimasi yang melibatkan banyak variable disebut dengan optimasi multi dimensi.
3. Optimasi dinamis mempunyai keluaran yang merupakan fungsi waktu, sedangkan optimasi statis mempunyai keluaran yang independen terhadap waktu.
4. Optimasi diskret adalah optimasi yang melibatkan variable yang mempunyai sejumlah variasi nilai yang terbatas, sedangkan optimasi malar adalah optimasi yang banyaknya variasi nilai variabelnya tidak berhingga.
5. Optimasi dengan kendala menggabungkan kesamaan variable dan ketidak samamaan variable ke dalam fungsi objektif.

6. Optimasi yang penyelesaiannya dicari secara acak disebut optimasi acak, sedangkan optimasi yang penyelesaiannya dicari dari sebuah nilai variable awal tertentu disebut pencarian nilai optimum.

2.5. Algoritma genetika

2.5.1. Pengertian Algoritma genetika

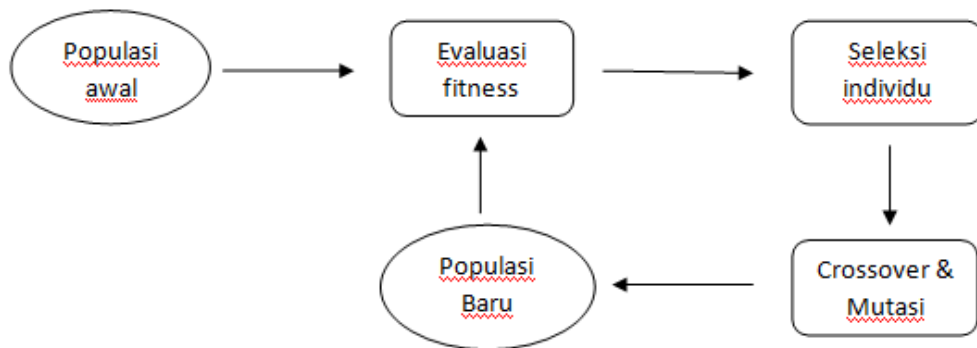
Menurut Desiani dan Arhami (2006), Algoritma genetika (AG) diperkenalkan pertama kali oleh John Holland (1975) dari Universitas Michigan, John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan ke dalam terminologi genetika. Kemudian menurut Goldberg (1989) mendefinisikan algoritma genetika ini sebagai suatu pencarian algoritma berdasarkan pada mekanisme seleksi alam dan genetika alam. Bauer (1993) mendefinisikan Algoritma genetika sebagai perangkat lunak, prosedur yang dimodelkan setelah genetika dan evolusi. Selain itu juga Algoritma genetika mempunyai karakteristik - karakteristik yang perlu diketahui sehingga dapat terbedakan dari prosedur pencarian atau optimasi yang lain, yaitu:

1. AG bekerja dengan pengkodean dari himpunan solusi permasalahan berdasarkan parameter yang telah ditetapkan dan bukan parameter itu sendiri.
2. AG melakukan pencarian pada sebuah populasi dari sejumlah individu -individu yang merupakan solusi permasalahan bukan hanya dari sebuah individu.

3. AG merupakan informasi fungsi objektif (*fitness*), sebagai cara untuk mengevaluasi individu yang mempunyai solusi terbaik, bukan turunan dari suatu fungsi.
4. AG menggunakan aturan-aturan transisi peluang, bukan aturan-aturan deterministik.

2.5.2 Struktur Umum Algoritma genetika

Menurut Charles L Karr (1999). Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasinya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Proses ini dapat direpresentasikan dalam algoritma sederhana Algoritma genetika sebagai berikut :



Gambar 2.3 Diagram alir algoritma genetika

Golberg (1989) mengemukakan bahwa algoritma genetika mempunyai 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 dan 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 genetik.

Secara umum struktur dari suatu algoritma genetik dapat mendefinisikan 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.

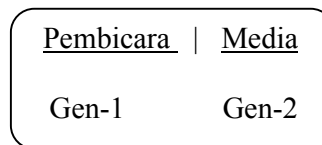
2.5.3. Komponen utama AG

Ada 6 komponen utama dalam AG, yaitu:

a) Teknik pengkodean

Menurut Desiani (2006), 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. Teknik pengkodean meliputi pengkodean gen dari kromosom. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variable.

Gen dapat direpresentasikan dalam bentuk *code angka*, yang terdiri dari Gen-1 sebagai Pembicara, Gen-2 sebagai media



Gambar 2.4 Kromosom

b) Fungsi Evaluasi

Ada 2 hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif ke dalam fungsi *fitness*. Secara umum, fungsi fitness diturunkan dari fungsi objektif dengan nilai yang tidak negatif. Dalam AG, individu yang memiliki nilai fitness tinggi pada kromosomnya yang akan dipertahankan. Sedangkan individu yang pada kromosomnya bernilai fitness rendah akan diganti. Fungsi

fitness tergantung pada permasalahan tertentu dari representasi yang digunakan.

Fungsi *fitness* tersebut sebagai berikut:

$$\text{fitness} = \frac{1}{1 + \sum \text{Penalti}} \quad (\text{sam'ani,2012}) \quad (2.1)$$

Keterangan:

Penalty= nilai kesesuaian dan tidak kesesuaian jadwal.

c) Seleksi Roda Roulette

Seleksi ini bertujuan untuk memilih individu yang akan dipilih untuk proses persilangan dan mutasi, dimana diharapkan diperoleh calon induk yang baik. Induk yang baik akan menghasilkan keturunan yang baik. Langkah pertama dalam seleksi yaitu 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.

d) Persilangan (*crossover*)

Salah satu komponen yang paling penting dalam algoritma genetika adalah persilangan (*crossover*) persilangan merupakan proses pada algoritma genetika yang bekerja untuk menggabungkan dua kromosom orang tua (*parent*) menjadi kromosom baru (*offspring*) pada suatu waktu. Sebuah kromosom yang mengarah pada solusi yang baik dapat diperoleh melalui proses *crossover* pada dua buah kromosom. Tidak semua kromosom mengalami persilangan. Cara sederhana pada proses persilangan yaitu dengan memilih suatu titik yang dipisahkan secara random dan kemudian membentuk *offspring*

dengan cara mengkombinasikan segmen dari satu induk ke sebelah kiri dari titik yang dipisahkan dengan segmen dari induk yang lain ke sebelah kanan dari titik yang dipisahkan.

Terdapat beberapa jenis *crossover* :

A. Crossover satu titik

Proses crossover dilakukan dengan memisahkan satu string menjadi dua bagian, kemudian salah satu bagian dipertukarkan dengan salah satu bagian dari string yang lain yang telah dipisahkan.

Contoh :

misal ada 2 kromosom dengan panjang 12 :

Induk 1 : 0 1 1 1 0 | **0 1 0 1 1 1 0**

Induk 2 : **1 1 0 1 0** | 0 0 0 1 1 0 1

kromosom yang dipilih : 5

Anak 1 : 0 1 1 1 0 | 0 0 0 1 1 0 1

Anak 2 : **1 1 0 1 0** | **0 1 0 1 1 1 0**

B. Crossover banyak titik

Proses crossover ini dilakukan dengan memisahkan suatu string menjadi beberapa bagian dan selanjutnya dipertukarkan dengan bagian dari string yang lain yang telah dipisahkan .

Contoh :

misal ada 2 kromosom dengan panjang 12 :

Induk 1 : **0 1** | **1 1 0 0** | **1 0 1 1** | **1 0**

Induk 2 : 1 1 | 0 1 0 0 | 0 0 1 1 | 0 1

Posisi yang dipilih : kromosom setelah 2,6, dan 10

Kromosom baru terbentuk:

Anak 1 : **0 1** | 0 1 0 0 | **1 0 1 1** | 0 1

Anak 2 : 1 1 | **1 1 0 0** | 0 0 1 1 | **1 0**

C. Crossover Partial Mapped Crossover (PMX)

Proses croosover ini dilakukan dengan menukarkan substring antar induk untuk menghasilkan *proto-child*. Menghubungkan pemetaan diantara dua daerah pemetaan lalu kromosom keturunan mengacu pada hubungan pemetaan, contoh :

1. Pilih posisi untuk menentukan substring secara acak

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

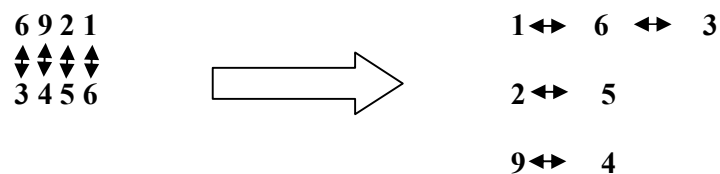
Induk 2 : 5 4 **6 9 2 1** 7 8 3

2. Tukar substring diantara induk

Proto_child 1 : 1 2 **6 9 2 1** 7 8 9

Proto_child 2 : 5 4 **3 4 5 6** 7 8 3

3. Menentukan hubungan mapping



4. Menentukan kromosom mengacu pada hubungan mapping

Turunan 1 : 3 5 **6 9 2 1** 7 8 9

Turunan 2 : 2 9 **3 4 5 6** 7 8 1

e) Mutasi (*mutation*)

Operator berikutnya mutasi (*mutation*). Proses mutasi merupakan proses untuk mengubah salah satu atau lebih beberapa gen dari suatu kromosom. Proses ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Beberapa cara operasi mutasi yang diterapkan dalam algoritma genetika menurut jenis pengkodeannya, 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 dipilih secara acak (atau dengan menggunakan skema tertentu) pada kromosom.

Contoh :

mutasi pada pengkodean biner :

Kromosom sebelum mutasi : 1 0 0 1 0 1 1 1

Kromosom sesudah mutasi : 1 0 0 1 0 0 1 1

2. Mutasi dalam pengkodean permutasi

Proses mutasi dalam pengkodean permutasi dilakukan dengan memilih dua posisi (lokasi) dari kromosom dan nilainya saling dipertukarkan

Contoh

mutasi dalam pengkodean permutasi

Kromosom sebelum mutasi : 1 2 3 4 6 5 8 7 9

Kromosom sesudah mutasi : 1 2 7 4 6 5 8 3 9

f) Penentuan Parameter

Parameter kontrol genetika diperlukan untuk mengendalikan operator – operator seleksi. Pemilihan parameter genetika menentukan penampilan kinerja algoritma genetika dalam memecahkan masalah Desiani (2006). Ada dua parameter dasar dari algoritma genetika, yaitu probabilitas *crossover* (pc) dan probabilitas mutasi (pm)

1. probabilitas crossover (pc)

probabilitas crossover akan mengendalikan operator *crossover* dalam setiap generasi dalam populasi yang mengalami *crossover*. Semakin besar nilai probabilitas *crossover*, akan semakin cepat struktur individu baru terbentuk ke dalam populasi. Sedangkan apabila nilai probabilitas *crossover* terlalu besar. Individu yang merupakan kandidat solusi terbaik mungkin akan dapat hilang lebih cepat pada generasi selanjutnya. Disarankan nilai probabilitas *crossover* berkisar 0,6%

2. Probabilitas mutasi (Pm)

Probabilitas mutasi akan mengendalikan operator mutasi pada setiap generasi dimana peluang mutasi yang digunakan biasanya lebih kecil daripada peluang *crossover*. Pada seleksi alam murni, mutasi jarang sekali muncul. Oleh karena itu, operator mutasi pada algoritma genetika juga tidak selalu terjadi. Untuk itulah nilai peluang mutasi dibuat lebih kecil untuk setiap generasi. Disarankan nilai probabilitas mutasi kecil berkisar 0,05%

Parameter lain yang juga ikut menentukan efisiensi kinerja algoritma genetika adalah ukuran populasi (*popsize*), yaitu banyaknya kromosom dalam satu populasi. Jika terlalu sedikit kromosom dalam populasi, algoritma genetika mempunyai kemungkinan sedikit untuk melakukan crossover dan hanya sebagian kecil dari ruang pencarian yang dieksplorasi. Sebaliknya, jika terlalu banyak jumlah kromosom, algoritma genetika cenderung menjadi lambat dalam menemukan solusi.

Ada beberapa rekomendasi yang bisa digunakan dalam kontrol algoritma genetika, antara lain:

A. Untuk permasalahan yang memiliki kawasan solusi cukup besar,

De Jong merekomendasikan untuk nilai parameter kontrol:

$$(popsize;pc;pm) = (50; 0,6; 0,001).$$

B. Bila rata-rata fitness setiap generasi digunakan sebagai indikator,

maka Grefenstette merekomendasikan :

$$(popsize;pc;pm) = (30; 0,95; 0,01).$$