

**PENJADWALAN PRODUKSI KONVEKSI DENGAN
KETERBATASAN BIAYA PRODUKSI MENGGUNAKAN
ALGORITMA GENETIK**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana dalam bidang Ilmu Komputer

Oleh:

PUTRI WAHYU SURYANINGRUM
0610963044-96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI

**PENJADWALAN PRODUKSI KONVEKSI DENGAN
KETERBATASAN BIAYA PRODUKSI MENGGUNAKAN
ALGORITMA GENETIK**

Oleh:

**PUTRI WAHYU SURYANINGRUM
0610963044-96**

**Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 10 Agustus 2011
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana dalam bidang Ilmu Komputer**

Pembimbing I

**Dian Eka R., S.Si.,M.Kom.
NIP : 197306192002122001**

**Mengetahui,
Ketua jurusan matematika
Fakultas MIPA Universitas Brawijaya**

**Dr. Abdul Rouf Alghofari, MSc
NIP : 196709071992031001**

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Putri Wahyu Suryaningrum
NIM : 0610963044
Jurusan : Matematika
Penulis Tugas Akhir Berjudul: **Penjadwalan Produksi Konveksi dengan Keterbatasan Biaya Produksi Menggunakan Algoritma Genetik.**

Dengan ini menyatakan bahwa :

1. Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.
2. Apabila dikemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 10 Agustus 2011
Yang menyatakan,

Putri Wahyu Suryaningrum
NIM. 0610963044

UNIVERSITAS BRAWIJAYA



PENJADWALAN PRODUKSI KONVEKSI DENGAN KETERBATASAN BIAYA PRODUKSI MENGGUNAKAN ALGORITMA GENETIK

ABSTRAK

Penjadwalan produksi pada konveksi digunakan untuk menghasilkan runtutan kerja yang optimal sehingga dapat menghasilkan laba (*profit*) yang maksimal. Pada skripsi ini metode yang digunakan Algoritma Genetik karena Algoritma Genetik sering digunakan sebagai salah satu metode optimasi penjadwalan dan dapat secara efektif mengatasi permasalahan *combinatorial optimization* yang berskala besar dengan cara mencari solusi terbaik yang dapat memuaskan semua kriteria dengan waktu komputasi yang relatif kecil.

Proses penjadwalan produksi pada konveksi menggunakan Algoritma Genetik ini diawali dengan memasukkan data *order*, kemudian melakukan membangkitkan populasi awal, melakukan proses *crossover* menggunakan metode *Position Based crossover*. Pada proses *crossover*, untuk sepasang kromosom yang mengalami *crossover* akan menghasilkan 1 *offsring*. Proses selanjutnya mutasi dengan metode *Reciprocal Exchange Mutation*, kemudian dilakukan proses perhitungan nilai *fitness* yang didapatkan berdasarkan nilai *profit* yang didapatkan yang kemudian digunakan untuk proses seleksi rangking yang berguna untuk menyeleksi kromosom yang akan masuk dalam proses selanjutnya. Tujuan yang diharapkan dalam proses penjadwalan ini yaitu dengan menghasilkan jadwal yang paling optimal yang dilihat berdasarkan Jumlah profit yang tertinggi.

Berdasarkan hasil pengujian dan analisa yang telah dilakukan dapat diketahui bahwa algoritma genetika dapat diterapkan untuk penjadwalan konveksi. Diketahui pula pengaruh perubahan parameter-parameter genetik berbanding lurus terhadap nilai *fitness*.

Kata kunci : Algoritma Genetika, Penjadwalan Konveksi, *Position Based crossover*, *Reciprocal Exchange Mutation*.

UNIVERSITAS BRAWIJAYA



CONVECTION PRODUCTION SCHEDULING WITH LIMITED PRODUCTION COSTS USING GENETIC ALGORITHM

ABSTRACT

Scheduling production on convection is used to generate the optimal sequence of work so that it can generate profit (profit) of the maximum. In this thesis the methods used for Genetic Algorithms Genetic algorithms are often used as one of scheduling and optimization methods can effectively solve the problems of large-scale combinatorial optimization by finding the best solution that can satisfy all the criteria with a relatively small computing time.

Productions scheduling processes on the convection using genetic algorithm begins by entering the order data, and then generates the initial population; make the process of Position Based crossover using crossover method. In the process of crossover, for a pair of chromosomes that have crossover will produce a offspring. The next process Mutation Reciprocal Exchange, and then made the process of calculating the fitness value obtained based on the value of profit earned is then used to rank a useful selection process to select chromosomes that will be in the next process. The expected goal in this scheduling process is to produce the most optimal schedule of visits by the highest amount of profit.

Based on the results of testing and analysis has been done can be seen that the genetic algorithm can be applied to scheduling convection. Note also the influence of genetic changes in these parameters proportional to fitness values.

Key words: Genetic Algorithms, Scheduling Convection, Position based crossover, Reciprocal Exchange Mutation.

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Puji syukur dipanjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayahnya, Skripsi yang berjudul “Penjadwalan Produksi Konveksi dengan Keterbatasan Biaya Produksi Menggunakan Algoritma Genetik ” ini dapat diselesaikan dengan baik. Skripsi ini disusun sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.

Dalam penyelesaian skripsi ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan penulis menyampaikan ucapan terima kasih kepada:

1. Dian Eka R., S.Si., M.Kom dan Tanzil Furqon, S.Kom, selaku pembimbing skripsi yang telah sabar memberi bimbingan, petunjuk dan masukan, sehingga skripsi ini bisa terselesaikan.
2. Dr. Abdul Rouf Alghofari, M.Sc., selaku Ketua Jurusan Matematika Fakultas Matematika Dan Ilmu Pengetahuan Alam.
3. Drs. Marji, M.T., selaku Ketua Program Studi Ilmu Komputer Jurusan Matematika.
4. Bapak ibu dosen Ilmu Komputer yang telah memberikan ilmunya kepada saya dan terima kasih bimbingannya, spesial untuk B. Dany Primanita, ST.
5. Orang tua yang senantiasa berdoa, memberi dukungan moril dan materiil serta semangat.
6. Teman baik saya Christ. Handri L yang selalu saya repotkan, terima kasih atas semua bantuan dan dukungannya.
7. Sahabat-sahabat saya Icha, Achie, Adhyn dan Endita, kalian sahabat terbaik saya.
8. Sahabat kampus saya Nafsin, Nia, Ike, Anung, Yuan dan teman kelas B 2006, makasi banyak teman-teman.
9. Dan semua pihak yang telah membantu dalam pengerjaan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Semoga penulisan laporan Skripsi ini bermanfaat bagi pembaca sekalian. Penulis menyadari bahwa Skripsi ini masih jauh dari kesempurnaan, dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca. Kepada peneliti lain mungkin masih bisa mengembangkan hasil penelitian ini. Akhirnya semoga skripsi ini bermanfaat bagi dunia pendidikan Indonesia.

Malang, 10 Agustus 2011

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN SKRIPSI.....	iii
LEMBAR PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
DAFTAR <i>SOURCE CODE</i>	xix
BAB I PENDAHULUAN BAB I PeNDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	
2.1 Sistem Produksi Konveksi.....	5
2.2 Harga Pokok Produksi (HPP).....	5
2.3 Biaya Bahan Baku	5
2.3.1 Biaya Tenaga Kerja Langsung.....	6
2.3.2 Biaya Overhead Pabrik	6
2.4 Biaya Operasional (BO)	6
2.5 Keuntungan.....	6
2.6 Penjadwalan.....	7
2.7 Algoritma Genetik	8
2.7.1 Konsep Algoritma Genetik	8
2.7.2 Alur Kerja Algoritma Genetik	9
2.7.3 Pengkodean	10
2.7.3.1 Pengkodean Biner	10
2.7.3.2 Pengkodean Permutasi	10
2.7.3.3 Pengkodean Nilai	11
2.7.3.4 Pengkodean Pohon	11
2.7.4 Fungsi Fitness	11
2.7.5 Seleksi.....	12
2.7.5.1 Daerah Sampling	12
2.7.5.2 Probabilitas Seleksi	13

2.7.5.3	Mekanisme Seleksi.....	13
2.7.6	Operasi Genetik.....	17
2.7.6.1	Crossover.....	17
2.7.6.2	Mutasi.....	22
BAB III METODELOGI DAN PERANCANGAN		
3.1	Deskripsi Umum Proses Penjadwalan	26
3.2	Perancangan Proses.....	27
3.2.1	Representasi Kromosom	28
3.2.2	Fungsi Fitness.....	30
3.2.3	Crossover.....	32
3.2.4	Mutasi.....	35
3.2.5	Seleksi Ranging.....	35
3.3	Rancangan Basis Data.....	38
3.4	Perhitungan Manual	38
3.4.1	Representasi Kromosom	39
3.4.2	Crossover.....	39
3.4.3	Mutasi.....	41
3.4.4	Fungsi Fitness.....	42
3.4.5	Seleksi Ranging.....	51
3.5	Rancangan Uji Coba	52
BAB IV IMPLEMENTASI DAN PEMBAHASAN		
4.1	Sumber Data	55
4.2	Lingkungan Implementasi	55
4.2.1	Lingkungan Perangkat Keras	55
4.2.2	Lingkungan Perangkat Lunak	55
4.3	Implementasi Antarmuka Penjadwalan Konveksi Menggunakan Algoritma Genetika	56
4.4	Implementasi Program Penjadwalan Konveksi Menggunakan Algoritma Genetika	59
4.4.1	Representasi Kromosom	59
4.4.2	Crossover.....	61
4.4.3	Mutasi.....	64
4.4.4	Fungsi Fitness.....	65
4.4.5	Seleksi Ranging.....	66
4.5	Uji Coba dan Analisa Hasil.....	67
BAB V KESIMPULAN DAN SARAN		
5.1	Kesimpulan.....	77
5.2	Saran.....	77
DAFTAR PUSTAKA.....		79

DAFTAR GAMBAR

Gambar 2.1 Siklus Algoritma Genetika oleh David Goldberg.....	9
Gambar 2.2 Contoh kromosom pada pengkodean biner.	10
Gambar 2.3 Permutation Encoding	10
Gambar 2.4 Contoh kromosom pada pengkodean nilai.	11
Gambar 2.5 Contoh kromosom dengan pengkodean pohon.	11
Gambar 2.6 Contoh populasi dengan 5 kromosom	14
Gambar 2.7 Probabilitas suatu kromosom dalam roda <i>roulette</i>	14
Gambar 2.8 Keadaan sebelum dirangking	15
Gambar 2.9 Keadaan setelah dirangking.....	15
Gambar 3.1 Diagram Alir Pembuatan Perangkat Lunak.....	24
Gambar 3.2 Flowchart Proses Penjadwalan Produksi Barang	28
Gambar 3.3 Flowchart Proses Hitung <i>Fitness</i>	30
Gambar 3.4 Ilustrasi <i>Crossover</i>	31
Gambar 3.5 Flowchart Proses <i>Crossover</i>	32
Gambar 3.6 Flowchart Proses <i>Position Based Crossover</i>	33
Gambar 3.7 Ilustrasi <i>Reciprocal Exchange Mutation</i>	34
Gambar 3.8 Flowchart Proses Mutasi	35
Gambar 3.9 Flowchart Proses Seleksi Ranking.	36
Gambar 4.1 Tampilan utama aplikasi.....	55
Gambar 4.2 Tampilan Input Parameter Genetik	55
Gambar 4.3 Tampilan Input Data Order	56
Gambar 4.4 Tampilan Pesan Inputan Tanggal Deadline Tidak sesuai	56
Gambar 4.5 Tampilan Proses Genetik.....	57
Gambar 4.6 Tampilan Tabel hasil Solusi	57
Gambar 4.7 Tampilan Tabel Kromosom CEADB	68

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

Tabel 3.1 Data <i>Order</i>	27
Tabel 3.2 Deadline <i>Order</i>	27
Tabel 3.3 Perhitungan untuk seleksi menggunakan metode Ranging	34
Tabel 3.4 Data <i>Order</i>	37
Tabel 3.5 Deadline <i>Order</i>	37
Tabel 3.6 Penjadwalan untuk kromosom A B C D E.....	42
Tabel 3.7 Penjadwalan untuk kromosom D E C B A.....	43
Tabel 3.8 Penjadwalan untuk kromosom A C D E B.....	44
Tabel 3.9 Penjadwalan untuk kromosom C E A D B.....	45
Tabel 3.10 Penjadwalan untuk kromosom E D C B A.....	46
Tabel 3.11 Penjadwalan untuk kromosom C D E B A.....	47
Tabel 3.12 Penjadwalan untuk kromosom C B E D A.....	48
Tabel 3.13 Penjadwalan untuk kromosom D C A E B.....	49
Tabel 3.14 Sebelum diurutkan	50
Tabel 3.15 Setelah diurutkan.....	50
Tabel 3.16 Tabel Percobaan Nilai <i>Fitness</i> dengan Peluang <i>Crossover</i> (Pc) yang Berbeda.	51
Tabel 3.17 Tabel Percobaan Nilai <i>Fitness</i> dengan Peluang Mutasi (Pm) yang Berbeda.....	52
Tabel 3.18 Tabel Percobaan Jumlah Generasi terhadap Nilai <i>Fitness</i>	52
Tabel 3.19 Tabel Percobaan Jumlah Populasi terhadap Nilai <i>Fitness</i>	53
Tabel 4.1 Hasil perhitungan manual di bab 3.....	65
Tabel 4.2 Perbandingan Tanggal Deadline dan Tanggal Selesai	67
Tabel 4.3 Data pengujian	69
Tabel 4.4 Nilai <i>Fitness</i> pada probabilitas <i>crossover</i>	69
Tabel 4.5 Nilai <i>Fitness</i> pada probabilitas mutasi	70
Tabel 4.6 Nilai <i>Fitness</i> terhadap perubahan jumlah generasi.....	72
Tabel 4.7 Nilai <i>Fitness</i> terhadap perubahan jumlah populasi.....	73

UNIVERSITAS BRAWIJAYA



DAFTAR SOURCE CODE

<i>Source code</i> 4.1 Representasi Kromosom dan Pembentukan Populasi Awal	60
<i>Source code</i> 4.2 Proses Crossover	62
<i>Source code</i> 4.3 Proses Mutasi	63
<i>Source code</i> 4.4 Proses Hitung Fitness	64
<i>Source code</i> 4.6 Proses Seleksi	65

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam bidang industri seperti halnya suatu perusahaan konveksi sering kali produksi suatu produk dilakukan berdasarkan pesanan. Pada proses pemesanan akan disepakati antara pemesan (konsumen) dan produsen bahwa barang pesanan harus selesai pada waktu tertentu (*deadline*). Selanjutnya konsumen memberi sejumlah uang sebagai tanda pemesanan yang disebut sebagai uang muka atau *down payment* (DP), sedangkan sisa pembayaran akan diberikan setelah barang pesanan selesai dikerjakan.

Dalam suatu perusahaan konveksi lebih dari satu pesanan (*order*) yang didapat dalam kurun waktu tertentu. *Order* yang lainnya juga mempunyai kriteria yang sama yaitu terdapat DP dan *deadline*.

Untuk memproduksi barang pesanan tersebut dibutuhkan biaya untuk membeli bahan baku produksi, menggaji pekerja serta biaya operasional yang meliputi transportasi, listrik, air dan lainnya. Akan tetapi seringkali DP yang diperoleh untuk masing-masing *order* berkisar 10-50% dari total *order*. DP tersebut tidak mencukupi untuk proses produksi, sehingga diperlukan solusi yang dapat mengatasi masalah ini.

Salah satu solusinya yaitu dengan menggunakan DP dari *order* lain ataupun dengan menggunakan pelunasan dari *order* yang telah selesai dikerjakan. *Order* yang dapat dibiayai sesuai dengan jumlah uang yang tersedia akan didahulukan dan menunda *order* lainnya sampai tersedianya dana yang diperoleh dari pelunasan *order* yang telah diselesaikan ataupun DP dari *order* baru yang lainnya. Akan tetapi, penundaan tersebut sedapat mungkin tidak melebihi dari *deadline* yang telah ditetapkan karena pengerjaan yang melebihi *deadline* akan dikenakan sejumlah denda/penalti. Penalti yang dikenakan sekitar 5% dari total omset (harga satuan barang dikali jumlah barang yang dipesan) dikalikan dengan jumlah hari keterlambatan.

Agar diketahui *order* mana yang didahulukan untuk diselesaikan atau *order* mana yang selanjutnya dikerjakan, maka diperlukan suatu penyusunan runtutan kerja yang disebut penjadwalan. Secara umum, **penjadwalan** merupakan suatu

permasalahan dalam hal melakukan *sequencing* terhadap sejumlah operasi dan mengalokasikannya ke dalam slot waktu tertentu tanpa melanggar *technical constraints* (batasan teknis) dan *capacitive constraints* (keterbatasan kapasitas yang dimiliki) (Betrianis dan Putu, 2003). Penjadwalan yang efektif bila sejumlah order dapat dikerjakan dengan waktu yang minimal sehingga mengurangi biaya operasional yang dikeluarkan.

Salah satu metode penyelesaian permasalahan penjadwalan adalah metode algoritma heuristik. Meskipun algoritma heuristik tidak menjamin tercapainya suatu solusi optimal, namun algoritma heuristik akan dapat secara efektif mengatasi permasalahan *combinatorial optimization* yang cukup sulit dan berskala besar dengan cara mencari solusi terbaik yang dapat memuaskan semua kriteria dengan waktu komputasi yang relatif kecil. Algoritma heuristik yang populer digunakan untuk permasalahan penjadwalan yaitu *Simulated Annealing*, Algoritma Genetik dan *Tabu Search*. (Betrianis dan Putu, 2003).

Algoritma Genetik dapat digunakan untuk menyelesaikan penjadwalan seperti pada “Pemakaian Algoritma Genetik untuk Penjadwalan Job Shop Dinamis Non Deterministik” (Nico, 2004), “Penjadwalan *Flowshop* dengan Algoritma Genetik” (Tessa, 1999) dan “*Scheduling with Genetic Algorithm*” (Philips).

Dengan metode tersebut, diharapkan dapat menghasilkan solusi yang efektif untuk permasalahan diatas. Berdasarkan latar belakang yang telah dipaparkan, maka penulis mengambil judul pada skripsi ini “**Penjadwalan Produksi Konveksi dengan Keterbatasan Biaya Produksi Menggunakan Algoritma Genetik**”.

1.2 Rumusan Masalah

Agar dalam pemecahan masalah tidak menyimpang dari tujuan semula dan menghindari kemungkinan meluasnya pembahasan dari seharusnya, perlu kiranya dilakukan batasan-batasan permasalahan sebagai berikut :

- a. Bagaimana membuat model genetika pada penjadwalan produksi konveksi dengan algoritma genetik ?
- b. Bagaimana pengaruh perubahan parameter genetik terhadap nilai *fitness*?

1.3 Batasan Masalah

Pada skripsi ini dibatasi pada beberapa hal, antara lain:

- a. Penjadwalan yang dilakukan untuk suatu bidang usaha konveksi TOCKE' Clothing Jl. Sumbersari 290 Malang dengan jumlah pegawai untuk potong pola 1 orang dan pegawai untuk menjahit 4 orang.
- b. Hambatan dari luar misalnya tidak tersedianya bahan baku, kerusakan mesin, pegawai berhalangan hadir diabaikan sehingga tidak berpengaruh pada waktu penyelesaian tiap proses.

1.4 Tujuan Penelitian

Tujuan penelitian dalam skripsi adalah sebagai berikut :

1. Membuat model genetika untuk masalah penjadwalan produksi konveksi dengan algoritma genetik.
2. Mengetahui pengaruh perubahan parameter genetik terhadap nilai *fitness*.

1.5 Manfaat Penelitian

Manfaat yang akan dicapai dari skripsi ini adalah untuk membantu menyelesaikan masalah penjadwalan konveksi pesanan. Sehingga pembuatan jadwal untuk produksi konveksi akan lebih cepat, efektif dan efisien, namun tetap memperhatikan aturan-aturan yang berlaku dalam aturan penjadwalan.

1.6 Sistematika Penulisan

Pembuatan skripsi ini dilakukan dengan pembagian bab sebagai berikut:

BAB I: PENDAHULUAN

Pada bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan tugas akhir.

BAB II: TINJAUAN PUSTAKA

Bab ini menjelaskan tentang penjadwalan secara umum, teori dasar algoritma genetika dan penjadwalan menggunakan algoritma genetika. Adapun literatur yang digunakan meliputi buku referensi dan dokumentasi internet.

BAB III: METODOLOGI DAN PERANCANGAN

Bab ini berisi tentang aliran proses atau alur dari sistem yang akan dibuat serta algoritma, contoh perhitungan manual dan flowchart dari program serta pembahasannya.

BAB IV: HASIL DAN PEMBAHASAN

Pada bab ini berisi implementasi program dan analisa hasil pengemplementasiannya.

BAB V: PENUTUP

Bab lima berisi kesimpulan dari pembahasan dan saran yang diharapkan bermanfaat untuk pengembangan skripsi ini selanjutnya.



BAB II TINJAUAN PUSTAKA

2.1 Sistem Produksi Konveksi

Pada masa lalu pengertian produksi hanya dikaitkan dengan unit usaha fabrikasi yaitu yang menghasilkan barang – barang nyata seperti mobil, perabot, semen dan sebagainya. Namun pengertian produksi pada saat ini menjadi semakin meluas. Produksi sering diartikan sebagai aktivitas yang ditujukan untuk meningkatkan nilai masukan (*input*) menjadi keluaran (*output*). Dengan demikian maka kegiatan usaha jasa seperti dijumpai pada perusahaan angkutan, asuransi, bank, pos, telekomunikasi, dsb menjalankan juga kegiatan produksi.

Pada sistem produksi bidang konveksi, proses produksi dilakukan secara keseluruhan oleh tiap-tiap operator jahit. Satu orang penjahit akan menjahit satu baju dari kerah, lengan, saku dan seterusnya sampai menjadi satu pakaian utuh (Andalas Clothing).

2.2 Harga Pokok Produksi (HPP)

Biaya produksi atau Harga Pokok Produksi (*Cost of Goods Manufactured*) merupakan kumpulan dari biaya-biaya yang dikeluarkan untuk memperoleh dan mengolah bahan baku sampai menjadi barang jadi (Dina Novia, 2010).

Biaya-biaya tersebut terdiri dari (Dina Novia, 2010):

- a. Biaya Bahan Baku (*disingkat* BBB)
- b. Biaya Tenaga Kerja Langsung (*disingkat* BTKL)
- c. Biaya Overhead Pabrik (*disingkat* BOP)

$$\text{HPP} = \text{BBB} + \text{BTKL} + \text{BOP} \quad (2.1)$$

2.3 Biaya Bahan Baku

Biaya Bahan Baku adalah harga perolehan (harga pokok) seluruh substansi / materi pokok yang terdapat pada barang jadi. Bahan baku merupakan bagian barang jadi yang dapat ditelusuri keberadaannya. Bahan baku pada sebuah pabrik dapat berasal dari barang jadi pabrik yang lain (Dina Novia, 2010).

Pada bidang usaha konveksi bahan baku meliputi pembelian bahan baku kain, benang dan keperluan lainnya. Barang tersebut harus

sudah tersedia ketika proses produksi dilakukan. Bila barang baku tidak tersedia maka proses produksi tidak akan berjalan.

2.3.1 Biaya Tenaga Kerja Langsung

Tenaga kerja langsung adalah tenaga kerja yang memiliki kinerja langsung terhadap proses pengolahan barang, baik menggunakan kemampuan fisiknya maupun dengan bantuan mesin. Tenaga kerja langsung memperoleh kontraprestasi yang dikategorikan sebagai biaya tenaga kerja langsung. Jadi, biaya tenaga kerja langsung adalah semua kontraprestasi yang diberikan kepada tenaga kerja langsung. Pada bidang usaha konveksi biaya tenaga kerja langsung meliputi biaya potong pola dan biaya jahit.

2.3.2 Biaya Overhead Pabrik

Biaya Overhead Pabrik adalah biaya-biaya yang timbul dalam proses pengolahan, yang tidak dapat digolongkan dalam biaya bahan baku dan biaya tenaga kerja langsung.

Biaya-biaya yang termasuk dalam biaya overhead pabrik yaitu :

- a. Biaya tenaga kerja tidak langsung, seperti upah pengawas, mandor, mekanik, bagian reparasi dan lain-lain.
- b. Biaya bahan penolong, yaitu macam-macam bahan yang digunakan dalam proses pengolahan, tetapi kuantitasnya sangat kecil dan tidak dapat ditelusur keberadaannya pada barang jadi.
- c. Biaya penyusutan gedung pabrik, biaya penyusutan mesin, dan lain-lain.

2.4 Biaya Operasional (BO)

Biaya operasional merupakan semua pengeluaran yg langsung digunakan untuk produksi atau pembelian barang yg diperdagangkan termasuk biaya umum, biaya penjualan, biaya administrasi, biaya transportasi, biaya sewa ruangan dan bunga pinjaman.

2.5 Keuntungan

Secara garis besar, keuntungan (*profit*) yaitu hasil (laba) yang diperoleh dari perdagangan. Dalam bidang konveksi keuntungan dapat dihitung dengan mengurangi jumlah harga barang yg dijual (omset) dengan biaya dalam pemrosesan barang tersebut.

2.6 Penjadwalan

Pengertian jadwal menurut kamus besar bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja; daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Secara umum, penjadwalan merupakan suatu permasalahan dalam hal melakukan *sequencing* terhadap sejumlah operasi dan mengalokasikannya ke dalam slot waktu tertentu tanpa melanggar *technical constraints* (batasan teknis) dan *capacitive constraints* (keterbatasan kapasitas yang dimiliki) (Betrianis dan Putu, 2003).

Secara garis besar prosedur umum yang diterapkan pada permasalahan penjadwalan dapat dibagi menjadi 2 kelompok (Betrianis dan Putu, 2003), yaitu :

1. *Constructive Procedure*

Constructive procedure ialah suatu prosedur pemecahan permasalahan penjadwalan dimana solusi penjadwalan dibuat dalam satu kali proses pencarian sampai didapat satu solusi optimal yang lengkap. Metode yang termasuk kedalamnya antara lain :

- a. *Basic Dispatching Rules*
- b. *Mathematical Programming*
- c. *Composite Dispatching Rules*
- d. *Branch and Bound*
- e. *Beam Search*

2. *Iterative Procedure*

Iterative procedure berangkat dari satu solusi penjadwalan lengkap yang ditentukan secara acak atau dengan cara lain, yang kemudian solusi tersebut dimanipulasi secara bertahap untuk mendapatkan satu solusi yang optimal atau mendekati optimal.

- a. *Classical Iterative Improvement*
- b. *Threshold Algorithms*
- c. *Tabu Search*
- d. *Simulated Annealing*
- e. *Genetic algorithms*

2.7 Algoritma Genetik

Algoritma genetika adalah suatu algoritma/prosedur penelusuran yang berdasarkan pada mekanisme dari *natural selection* dan *natural genetics* yang dapat digunakan untuk memecahkan *combinatorial optimization problems* yang sulit. Algoritma ini dikembangkan oleh **John Holland**, rekan kerja dan muridnya dari Michigan University, yang mengkombinasikan keberhasilan suatu struktur untuk bertahan hidup (*survival of the fittest*) dengan perubahan informasi dari struktur tersebut secara random untuk membentuk suatu mekanisme penelusuran seperti yang dilakukan dalam proses pembentukan manusia atau makhluk hidup (gen/sifat yang diturunkan) (Tessa,1999).

2.7.1 Konsep Algoritma Genetik

Dalam algoritma genetika dikenal adanya *chromosome* yaitu kandidat penyelesaian yang digambarkan dengan urutan *binary digits* atau *integers* sesuai kondisi yang dikehendaki. *Chromosome* terdiri dari *gen-gen* yang melambangkan ciri-ciri unik dari *chromosome* tersebut sedangkan nilai yang berkaitan dengan ciri-ciri yang dilambangkan oleh *gen* tertentu disebut *allele*. Populasi adalah sekumpulan *chromosome* yang akan diperbaharui pada setiap generasi (Tessa,1999).

Dikenal pula adanya dua operator yang memegang peranan penting dalam algoritma genetika yaitu *crossover operator* dan *mutation operator*. *Crossover operator* adalah operator yang menggabungkan dua *chromosome* sehingga menghasilkan *child chromosome* yang mewarisi ciri-ciri dasar dari *parent chromosome* sedangkan *mutation operator* dipergunakan untuk memperkenalkan transformasi (informasi) baru pada populasi yang melibatkan perubahan *chromosome* secara random (Tessa,1999).

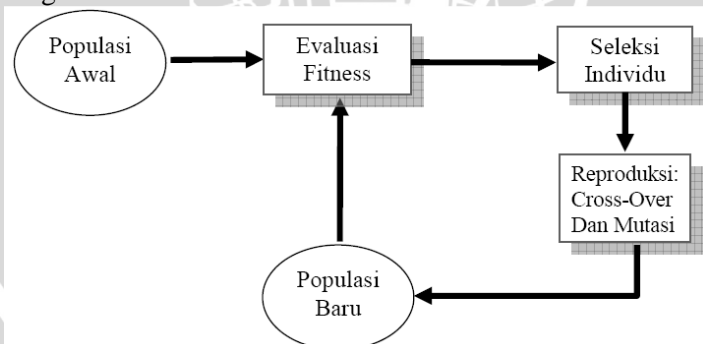
Dalam tiap generasi kromosom-kromosom tersebut di evaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan menggunakan ukuran yang disebut *fitness*. Untuk memilih kromosom yang tetap dipertahankan untuk generasi selanjutnya dilakukan proses yang disebut seleksi. Kromosom yang mempunyai nilai *fitness* tinggi akan memiliki peluang lebih besar untuk terpilih pada generasi selanjutnya (Christian, 2007).

2.7.2 Alur Kerja Algoritma Genetik

Secara garis besar, algoritma genetika dapat dijelaskan dengan algoritma berikut (Obitko, 1998):

1. **[Mulai]** Membangun populasi secara random sebanyak n kromosom (sesuai dengan masalahnya).
2. **[Fitness]** Evaluasi setiap *fitness* $f(x)$ dari setiap kromosom x pada populasi.
3. **[Populasi baru]** Membuat populasi baru dengan mengulang langkah-langkah berikut sampai populasi baru lengkap.
 - a. **[Seleksi]** Pilih dua kromosom induk dari populasi berdasarkan *fitness*-nya (semakin besar *fitness*-nya semakin besar kemungkinannya untuk terpilih).
 - b. **[Perkawinan silang/crossover]** Sesuai dengan besarnya kemungkinan perkawinan silang, induk terpilih disilangkan untuk membentuk anak. Jika tidak ada perkawinan silang, maka anak merupakan salinan dari induknya.
 - c. **[Mutasi]** Sesuai dengan besarnya kemungkinan mutasi, anak dimutasi pada setiap lokus (posisi pada kromosom).
 - d. **[Penerimaan]** tempatkan anak baru pada populasi baru.
4. **[Ganti]** Gunakan populasi yang baru dibentuk untuk proses algoritma selanjutnya.
5. **[Tes]** jika kondisi akhir terpenuhi, berhenti, dan hasilnya adalah solusi terbaik dari populasi saat itu.
6. **[Ulangi]** Ke nomer 2 .

Secara umum, proses/siklus algoritma genetika digambarkan pada gambar 2.2 yang pertama kali diperkenalkan oleh David Goldberg.



Gambar 2.2 Siklus Algoritma Genetika oleh David Goldberg

2.7.3 Pengkodean

Langkah pertama pada algoritma genetika adalah menerjemahkan/ merepresentasikan masalah riil menjadi terminologi biologi. Cara untuk merepresentasikan masalah ke dalam bentuk kromosom disebut pengkodean. Pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai kandidat solusi suatu masalah ke dalam suatu kromosom (Yohan, 2007). Terdapat beberapa cara pengkodean, dan pemilihannya berdasarkan masalah yang dihadapi. Berikut adalah beberapa jenis pengkodean yang umum digunakan (Nia, 2006) :

- a. Pengkodean biner
- b. Pengkodean permutasi
- c. Pengkodean nilai
- d. Pengkodean pohon

2.7.3.1 Pengkodean Biner

Pengkodean biner, yaitu metode pengkodean yang menggunakan bilangan biner. Metode ini banyak digunakan karena sederhana untuk diciptakan dan mudah dimanipulasi (Yohan, 2007). Pada pengkodean biner setiap kromosom terdiri atas barisan string bit 0 atau 1. Sebagai contoh:

Kromosom A	0101101100010011
Kromosom B	1011010110110101

Gambar 2.3 Contoh kromosom pada pengkodean biner.

2.7.3.2 Pengkodean Permutasi

Bentuk kromosom dari teknik pengkodean ini berupa kumpulan dari nilai integer/angka yang merepresentasikan suatu posisi dalam sebuah urutan. Teknik ini biasanya digunakan untuk masalah ordering seperti permasalahan *Travelling Salesman Problem* (TSP). Contoh dari *permutation encoding* dapat dilihat pada gambar 2.4.

Kromosom A	1 4 7 9 6 3 5 0 2 8
Kromosom B	9 3 2 5 8 1 6 0 4 7

Gambar 2.4 *Permutation Encoding*

2.7.3.3 Pengkodean Nilai

Pengkodean nilai dapat digunakan pada masalah yang sangat kompleks, dimana nilai yang dikodekan langsung merupakan representasi dari masalah. Penggunaan pengkodean biner pada tipe masalah yang kompleks akan menjadi lebih susah.

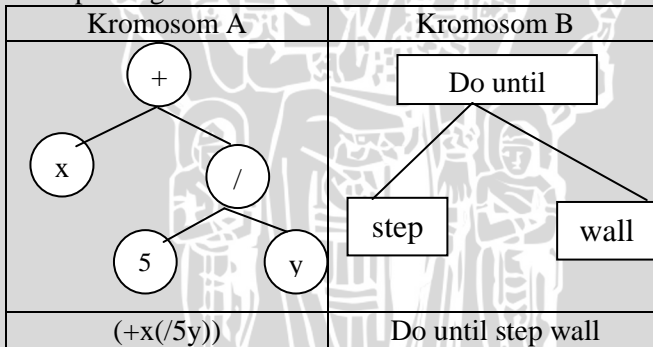
Pada pengkodean nilai, setiap kromosom adalah barisan dari beberapa nilai. Nilai dapat berupa apa saja, seperti bilangan biasa, bilangan riil, karakter sampai dengan obyek-obyek yang rumit.

Kromosom A	[red], [black], [blue], [yellow], [red]
Kromosom B	1.875, 3.9821, 9.1283, 6.8344, 4.116
Kromosom C	A, B, C, D, E, F, G, H, I, J, K, L, M, N

Gambar 2.5 Contoh kromosom pada pengkodean nilai.

2.7.3.4 Pengkodean Pohon

Pengkodean pohon lebih banyak digunakan untuk menyusun program atau ekspresi, bagi pemrograman genetika (*genetic programming*). Pada pengkodean pohon, setiap kromosom merupakan pohon dari sejumlah obyek, seperti fungsi atau perintah pada bahasa pemrograman.



Gambar 2.6 Contoh kromosom dengan pengkodean pohon.

2.7.4 Fungsi *Fitness*

Dalam algoritma genetika, fungsi evaluasi merupakan dasar untuk melakukan proses seleksi. Fungsi evaluasi memberikan penilaian kepada kromosom yang disebut nilai *fitness* (*fitness value*). Langkah-langkah untuk melakukan evaluasi ada 3, yaitu string dikonversi ke parameter fungsi, mengevaluasi fungsi objektifnya,

kemudian mengkonversi fungsi objektif tersebut ke dalam *fitness*. Jika solusi yang dicari adalah untuk maksimasi permasalahan, maka *fitness* sama dengan fungsi objektifnya. Sebaliknya, jika solusi yang dicari adalah untuk minimasi, maka nilai *fitness* sama dengan invers dari nilai objektifnya. Output yang dihasilkan dari fungsi *fitness* dipergunakan sebagai dasar untuk menyeleksi individu pada generasi berikutnya (Syamsuddin, 2004).

Dengan nilai *fitness* yang didapat lewat rumusan diatas, dapat dicari probabilitas kumulatif, yang berguna dalam pemilihan individu sebagai *parent*.

Nilai *fitness* pada masing-masing individu disarankan mempunyai rentang yang cukup lebar. Sesuai dengan prinsip evolusi alami, semakin lama kromosom beradaptasi, semakin baik kualitas yang terdapat dalam satu generasi. Sehingga, rentang nilai menjadi semakin kecil. Salah satu permasalahan yang sering terjadi jika terdapat beberapa kromosom yang mendominasi populasi sehingga dapat mengakibatkan kondisi konvergen yang terlalu dini (*premature convergence*) dan jika kromosom tersebut mempunyai nilai *fitness* yang rendah tetapi tidak optimal dapat mengakibatkan proses terjebak ke kondisi konvergen yang bersifat lokal (*local optimal*) (Kuswara, 2003).

2.7.5 Seleksi

Seleksi dilakukan untuk memilih individu yang akan ikut dalam proses reproduksi. Terdapat tiga hal penting yang harus diperhatikan dalam proses seleksi yaitu (Kuswara, 2003) :

- a. Daerah sampling
- b. Probabilitas seleksi
- c. Mekanisme seleksi

2.7.5.1 Daerah Sampling

Proses seleksi menghasilkan generasi selanjutnya. Kromosom-kromosom pada generasi selanjutnya tersebut mungkin berasal dari semua *parent* dan semua *offspring* atau sebagian dari keduanya. Persoalan asal kromosom itu disebut persoalan daerah *sampling*. Terdapat dua jenis tipe daerah *sampling* yang umum dipergunakan dalam algoritma genetic, yaitu (Kuswara, 2003) :

- a. Daerah *sampling* tetap
Ukuran populasi selalu tetap pada setiap generasi. *Parent* digantikan oleh *offspring* segera setelah kelahiran *offspring*. Pendekatan itu disebut pendekatan generasi (*generation replacement*). Karena operasi genetic bersifat tidak pandang bulu, *offspring* jelek mungkin menggantikan *parent* yang baik.
- b. Daerah *sampling* yang diperbesar
Dalam proses seleksi berdasarkan daerah *sampling* yang diperluas, populasi baru terbentuk dari semua *parent* dan semua *offspring*.

2.7.5.2 Probabilitas Seleksi

Probabilitas seleksi dari kromosom dihitung berdasarkan nilai *fitness*-nya. Langkah-langkah yang diperlukan dalam menghitung probabilitas sebagai berikut (Kuswara, 2003):

1. Menghitung nilai *fitness* dari setiap individu.
2. Menjumlahkan nilai *fitness* dari semua individu yang terdapat dalam populasi.

$$TotFit = \sum_{i=1}^m Fitness(i) \quad (2.2)$$

Keterangan :

TotFit = jumlah nilai *fitness*

m = jumlah individu dalam populasi

3. Menghitung probabilitas terpilihnya setiap individu (p_i) dengan membagi *fitness* masing-masing individu dengan total nilai *fitness* populasi.

$$p_i = \frac{Fitness}{TotFit} \quad (2.3)$$

4. Menghitung probabilitas kumulatif untuk setiap individu dalam populasi (q_i).

$$q_i = \sum_{i=1}^j p_i \quad (2.4)$$

2.7.5.3 Mekanisme Seleksi

Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Seleksi mempunyai tujuan untuk memberikan

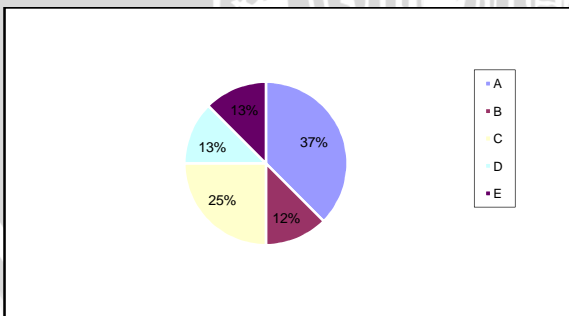
kesempatan reproduksi yang lebih besar bagi anggota populasi yang mempunyai nilai *fitness* terbaik. Beberapa jenis seleksi yang umum dipakai adalah (Yohan, 2007):

a. Seleksi Roda Roulette

Roulette wheel selection. Metode ini diajukan oleh John Holland. Ide dasarnya adalah untuk menentukan proporsi probabilitas seleksi atau probabilitas *survival* pada tiap kromosom sesuai dengan nilai *fitness*-nya. Individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap segmen individu memiliki ukuran yang sama dengan ukuran *fitness*-nya. Sebuah bilangan random dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan random tersebut akan terseleksi. Proses ini diulang hingga diperoleh sejumlah individu yang diharapkan. Metode seleksi roda *roulette* juga sering dikenal dengan nama *stochastic sampling with replacement*. Pada metode ini induk dipilih berdasarkan nilai *fitness*-nya, semakin besar nilai *fitness* maka akan semakin besar kemungkinannya untuk terpilih menjadi induk. Diandaikan semua kromosom diletakkan pada sebuah roda *roulette*, besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai *fitness*-nya seperti pada contoh berikut:

Kromosom	<i>Fitness</i>
A	15
B	5
C	10
D	5
E	5

Gambar 2.7 Contoh populasi dengan 5 kromosom



Gambar 2.8 Probabilitas suatu kromosom dalam roda *roulette*

Pada gambar 2.7 merupakan contoh dalam satu populasi terdiri dari lima kromosom. Pada tiap kromosom memiliki nilai *fitness* yang berbeda-beda. Pada gambar 2.8 dapat diketahui probabilitas terpilihnya masing-masing kromosom untuk menjadi induk. Pada kromosom A memiliki nilai *fitness* 15 dan nilai tersebut nilai *fitness* tertinggi pada populasi tersebut. Sehingga kromosom A memiliki probabilitas terbesar untuk terpilih menjadi induk.

b. Seleksi Rangking

Pada metode seleksi roda *roulette* akan bermasalah saat terdapat perbedaan *fitness* yang jauh. Sebagai contoh, jika *fitness* kromosom yang terbaik adalah 90% dari semua roda *roulette* dapat menyebabkan kromosom lain memiliki kesempatan yang kecil untuk dapat terpilih (Nia, 2006).

Proses dimulai dengan merangking atau mengurutkan kromosom di dalam populasi berdasarkan *fitness*nya kemudian memberi nilai *fitness* baru berdasarkan urutannya. Kromosom dengan nilai terburuk akan memiliki *fitness* baru nilai 1, terburuk kedua bernilai 2 dan begitu seterusnya, sehingga kromosom yang memiliki *fitness* terbaik akan memiliki nilai *fitness* N, dimana N adalah jumlah kromosom di dalam populasi. Seperti dapat dilihat pada gambar 2.9 (Nia, 2006).

Kromosom	<i>Fitness</i>
A	15
B	5
C	10
D	5
E	5

Gambar 2.9 Keadaan sebelum dirangking

Kromosom	<i>Fitness</i>	<i>Fitness</i> baru
B	5	1
D	5	2
E	5	3
C	10	4
A	15	5

Gambar 2.10 Keadaan setelah dirangking

Setelah adanya proses seleksi tersebut, maka saat ini seluruh kromosom mempunyai kesempatan untuk dipilih. Akan tetapi, metode ini dapat menyebabkan konvergensi menjadi lambat, karena kromosom terbaik tidak terlalu berbeda dengan yang lainnya (Nia, 2006).

c. Seleksi Turnamen

Seleksi turnamen merupakan variasi antara seleksi roda *roulette* dan seleksi rangking. Sejumlah k kromosom tertentu dari populasi beranggota n kromosom ($k \leq n$) dipilih secara acak dengan probabilitas yang sama. Dari k kromosom yang terpilih kemudian akan dipilih satu kromosom dengan *fitness* terbaik, yang diperoleh dari hasil pengurutan rangking *fitness* semua kromosom terpilih. Perbedaan dengan seleksi roda *roulette* adalah pemilihan kromosom yang akan digunakan untuk berkembangbiak tidak berdasarkan skala *fitness* dari populasi (Nia, 2006).

- **$(\mu + \lambda)$ Selection**

Metode ini merupakan prosedur deterministik yang memilih kromosom terbaik dari orang tua dan keturunan. Metode ini biasanya digunakan pada masalah optimasi kombinatorial (Yohan, 2007).

- ***Steady-state Reproduction***

Pada metode ini sejumlah *fitness parents* yang terburuk digantikan dengan sejumlah individu baru (offspring) (Yohan, 2007).

- **Sharing**

Teknik *sharing* diperkenalkan oleh Goldberg dan Richardson untuk optimasi dengan fungsi multi model. Teknik ini digunakan untuk menjaga keanekaragaman populasi. Fungsi *sharing* adalah sebuah cara untuk menentukan degradasi *fitness* individu dikarenakan jaraknya dengan tetangga. Dengan adanya degradasi, probabilitas reproduksi individu pada puncak keramaian ditahan, individu lain akan memperoleh keturunan (Yohan, 2007).

2.7.6 Operasi Genetik

Setelah individu yang berhak melakukan operasi genetik ditentukan, operasi genetik baru dapat dilakukan. Operasi tersebut yaitu tukar silang (*crossover*) dan mutasi.

2.7.6.1 Crossover

Proses tukar silang (*crossover*) berfungsi untuk menghasilkan keturunan dari dua buah kromosom induk yang terpilih. Kromosom anak yang dihasilkan merupakan kombinasi gen-gen yang dimiliki oleh kromosom induk. Probabilitas tukar silang (pc) menentukan frekuensi dari proses tukar silang yang terjadi dalam suatu populasi. Semakin tinggi nilai pc maka semakin besar kemungkinan dilakukan tukar silang (Kuswara, 2003).

Secara umum, mekanisme tukar silang adalah sebagai berikut (Kuswara, 2003):

1. memilih dua buah kromosom sebagai induk.
2. memilih secara acak posisi dalam kromosom, biasa disebut titik perkawinan silang, sehingga masing-masing kromosom induk terpecah menjadi dua segmen.
3. lakukan pertukaran antar segmen kromosom induk untuk menghasilkan kromosom anak.

Pada pengkodean biner terdapat 4 metode penyilangan yang sering digunakan. Metode tersebut adalah (Nia, 2006):

- a. Perkawinan silang satu titik.

Proses perkawinan satu titik dimulai dengan memilih satu titik pada barisan bit kromosom secara acak sebagai titik perkawinan silang. Kromosom baru akan dibentuk dengan cara menyalin barisan bit induk pertama dari bit pertama sampai titik perkawinan silang, sedangkan sisanya disalin dari induk kedua.

Contoh:

Misalkan ada 2 kromosom dengan panjang 12:

Induk 1: 01110 | 0101110

Induk 2: 11010 | 0001101

Posisi penyilangan yang terpilih: misalkan 5

Setelah penyilangan, diperoleh kromosom baru:

Anak 1: 01110 | 0001101

Anak 2: 11010 | 0101110

b. Perkawinan silang dua titik.

Perkawinan silang dua titik dimulai dengan menentukan secara acak dua titik yang akan disilangkan. Kromosom baru akan dibentuk dengan cara menyalin barisan bit kromosom induk pertama dari bit pertama sampai dengan dengan titik perkawinan silang pertama dari titik perkawinan silang kedua sampai dengan bit terakhir, sedangkan sisanya, yaitu titik perkawinan silang pertama sampai titik perkawinan silang kedua disalin dari induk kedua.

Contoh:

Misalkan ada 2 kromosom dengan panjang 12:

Induk 1: 011100101110

Induk 2: 110100001101

Posisi penyilangan yang terpilih: misalkan 2 dan 6

Setelah penyilangan, diperoleh kromosom baru:

Anak 1: 01 | 0100 | 1011 10

Anak 2: 11 | 1100 | 0011 01

c. Perkawinan silang seragam

Sebuah *mask* penyilangan dibuat sepanjang panjang kromosom secara random yang menunjukkna bit-bit dalam *mask* yang mana induk akan mensuplay anak dengan bit-bit yang ada. Disini, *anak_1* akan dihasilkan dari *induk_1* jika bit *mask* bernilai 1, dan *anak_1* akan dihasilkan dari *induk_2* jika bit *mask* bernilai 0. Sedangkan *anak_2* dihasilkan dari kebalikan *mask*.

Contoh:

Misalkan ada 2 kromosom dengan panjang 12:

Induk 1: 011100101110

Induk 2: 110100001101

Mask bit:

sampel 1: 100111001101

sampel 2: 011000110010

Setelah penyilangan, diperoleh kromosom baru:

Anak 1: 010100001100

Anak 2: 111100101111

d. Perkawinan silang aritmatik

Pada perkawinan silang aritmatik, digunakan beberapa operator aritmatika untuk mendapatkan *offspring* yang baru.

Contoh:

Induk 1 : 11001011

Induk 2 : 11011111

Anak (AND) : 11001011

Pada pengkodean permutasi terdapat 5 metode *crossover* yaitu (Mitsuo, 1997) :

a. **Partial-Mapped Crossover (PMX)**

PMX seperti halnya pada *crossover* dua titik untuk string biner, namun pada PMX menggunakan prosedur perbaikan khusus untuk menyelesaikan kromosom yg tidak sesuai yg disebabkan oleh *crossover* dua titik sederhana. Jadi esensinya PMX yaitu *crossover* dua titik ditambah prosedur perbaikan. Langkah-langkah PMX sebagai berikut:

1. Pilih dua posisi sepanjang string secara acak.
2. Tukar dua substring antara orang tua untuk menghasilkan *proto-child*.
3. Tentukan hubungan pemetaan di antara dua bagian substring tersebut.
4. Perbaiki kromosom *proto-child* menggunakan hubungan pemetaan diatas.

Contoh :

1. Parent 1	:	1	2	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	7	8	9
Parent 2	:	5	4	<u>6</u>	<u>9</u>	<u>2</u>	<u>1</u>	7	8	3
2. Proto-child 1	:	1	2	<u>6</u>	<u>9</u>	<u>2</u>	<u>1</u>	7	8	9
Proto-child 2	:	5	4	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	7	8	3
3.		3↔6↔1, 9↔4, 2↔5								
4. Offspring 1	:	3	5	<u>6</u>	<u>9</u>	<u>2</u>	<u>1</u>	7	8	4
Offspring 2	:	2	9	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	7	8	1

b. Order Crossover (OX)

Metode ini mirip dengan PMX dengan prosedur perbaikan yang berbeda. OX bekerja sebagai berikut:

1. Pilih substring dari *parent 1* secara acak.
2. Dihasilkan *proto-child* dengan menyalin substring tersebut ke posisi yang sama dengan *parent 1*.
3. Hapus gen-gen yang sudah ada di substring dari *parent 2*.
4. Tempatkan gen yang tidak terhapus pada posisi yang kosong didalam *proto-child* dari kiri ke kanan sesuai dengan urutan pada *parent 2*.

Contoh :

1. <i>Parent 1</i>	:	1 2 <u>3 4 5 6</u> 7 8 9
<i>Parent 2</i>	:	5 4 6 9 2 1 7 8 3
2. <i>Proto-child</i>	:	- - <u>3 4 5 6</u> - - -
3. <i>Parent 2</i>	:	- - - 9 2 1 7 8 -
4. <i>Offspring</i>	:	9 2 <u>3 4 5 6</u> 1 7 8

c. Position-Based Crossover

Metode ini sama seperti pada metode OX dalam hal perbaikan kromosom. Langkah-langkah metode ini sebagai berikut:

1. Pilih beberapa posisi dari *parent 1* secara acak.
2. Menghasilkan *proto-child* dengan menyalin gen pada posisi yang dipilih tadi dan diletakkan sama seperti pada posisi awal.
3. Hapus gen-gen pada *parent 2* yang sama dengan yang terpilih pada *parent 1*.
4. Tempatkan gen yang tidak terhapus pada posisi yang kosong didalam *proto-child* dari kiri ke kanan sesuai dengan urutan pada *parent 2*.

Contoh :

1. <i>Parent 1</i>	:	1 2 <u>3</u> 4 <u>5</u> <u>6</u> 7 <u>8</u> 9
<i>Parent 2</i>	:	5 4 6 9 2 1 7 8 3
2. <i>Proto-child</i>	:	- - <u>3</u> - <u>5</u> <u>6</u> - <u>8</u> -
3. <i>Parent 2</i>	:	- 4 - 9 2 1 7 - -
4. <i>Offspring</i>	:	4 9 <u>3</u> 2 <u>5</u> <u>6</u> 1 <u>8</u> 7

d. Order-Based Crossover

Metode ini sangat mirip seperti pada metode *Position-Based Crossover*. Langkah-langkah metode ini sebagai berikut:

1. Pilih beberapa posisi dari *parent 1* secara acak.
2. Hapus gen-gen pada *parent 2* yang sama dengan yang terpilih pada *parent 1*.
3. Menghasilkan *proto-child* dengan menyalin gen-gen *parent 2* yang tersisa dan diletakkan sama seperti pada posisi awal.
4. Tempatkan gen-gen *parent 1* yang terpilih pada posisi yang kosong didalam *proto-child* dari kiri ke kanan sesuai dengan urutan pada *parent 1*.

Contoh :

1. <i>Parent 1</i>	:	1 2 <u>3</u> 4 <u>5</u> <u>6</u> 7 <u>8</u> 9
<i>Parent 2</i>	:	5 4 6 9 2 1 7 8 3
2. <i>Parent 2</i>	:	- 4 - 9 2 1 7 - -
3. <i>Proto-child</i>	:	- 4 - 9 2 1 7 - -
4. <i>Offspring</i>	:	<u>3</u> <u>4</u> <u>5</u> 9 2 1 7 <u>6</u> <u>8</u>

e. Cycle Crossover (CX)

CX mirip dengan metode dengan *Position-Based Crossover*. Perbedaannya adalah bahwa gen-gen dari *parent 1* tidak dipilih secara acak tetapi dipilih berdasarkan siklus yang dihasilkan antar *parent*. Langkah-langkah metode CX sebagai berikut:

1. Cari siklus yang dihasilkan antar *parent*.
2. Salin gen-gen *parent 1* yang termasuk dalam siklus sebagai *proto-child* dengan posisi yang sesuai dari *parent 1*.
3. Tentukan gen-gen *parent 2* yang tersisa yang tidak termasuk dalam siklus.
4. Tempatkan gen-gen *parent 2* yang tersisa pada posisi yang kosong didalam *proto-child* dari kiri ke kanan sesuai dengan urutan pada *parent 2*.

Contoh :

1. <i>Parent 1</i>	:	<u>1</u> <u>2</u> 3 <u>4</u> <u>5</u> 6 7 8 <u>9</u>
<i>Parent 2</i>	:	<u>5</u> <u>4</u> 6 <u>9</u> <u>2</u> 3 7 8 <u>1</u>
Siklus	:	1→5→2→4→9→1
2. <i>Proto-child</i>	:	<u>1</u> <u>2</u> - <u>4</u> <u>5</u> - - - <u>9</u>
3. <i>Parent 2</i>	:	- - 6 - - 3 7 8 -
4. <i>Offspring</i>	:	<u>1</u> <u>2</u> 6 <u>4</u> <u>5</u> 3 7 8 <u>9</u>

2.7.6.2 Mutasi

Mutasi merupakan cara lain untuk membuat variasi populasi. Sama seperti *crossover*, mutasi juga mempunyai probabilitas mutasi (μ) untuk menentukan tingkat mutasi yang terjadi. Proses mutasi dilakukan untuk menghindari solusi-solusi dalam populasi mempunyai nilai lokal optimum.

Pada pengkodean permutasi terdiri dari 4 metode sebagai berikut (Mitsuo, 1997) :

1. **Inversion Mutation**

Metode bekerja dengan memilih dua posisi dalam kromosom secara acak dan kemudian membalikkan substring di antara dua posisi tersebut.

Contoh :

- a. Memilih dua posisi secara acak

1 2 3 4 5 6

- b. Membalikkan substring tersebut

1 2 5 4 3 6

2. **Insertion Mutation**

Metode bekerja dengan memilih sebuah gen secara acak dan kemudian gen tersebut disisipkan dalam posisi yang acak.

Contoh :

- a. Memilih sebuah gen secara acak

1 2 3 4 5 6

- b. Mensisipkan dalam posisi acak

1 2 5 3 4 6

3. **Displacement Mutation**

Pada metode ini dilakukan dengan memilih beberapa gen secara acak dan kemudian disisipkan secara acak pula.

Contoh :

- a. Memilih beberapa gen secara acak

2 3 4 5 6

- b. Mensisipkan dalam posisi acak

1 5 3 4 6 2

4. **Reciprocal Exchange Mutation**

Metode ini dilakukan dengan memilih dua posisi gen secara acak dan saling menukar posisi untuk masing-masing gen tersebut.

Contoh :

- a. Memilih dua posisi gen secara acak

2 3 4 5 6

- b. Menukar menukar posisi untuk masing-masing gen

1 2 5 4 3 6

UNIVERSITAS BRAWIJAYA



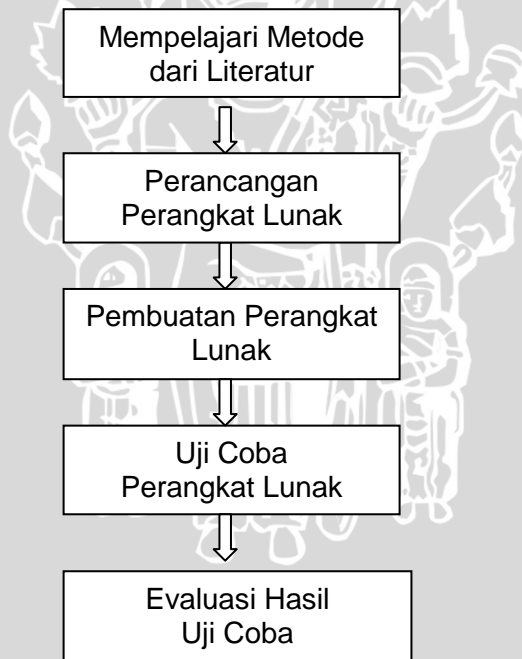
UNIVERSITAS BRAWIJAYA



BAB III METODOLOGI DAN PERANCANGAN

Pada bab metode dan perancangan ini akan dibahas langkah-langkah yang digunakan dalam pembuatan perangkat lunak dan metode percobaan. Tahapan pembuatannya adalah sebagai berikut.

1. Mempelajari metode yang digunakan dari literatur yang sudah ada dan yang telah diringkas dalam bab 2.
 2. Merancang perangkat lunak.
 3. Membuat perangkat lunak berdasarkan analisis dan perancangan yang sudah dilakukan.
 4. Uji coba perangkat lunak
 5. Evaluasi hasil uji coba yang sudah dilakukan oleh sistem.
- Langkah- langkah pembuatan perangkat lunak digambarkan pada gambar 3.1.



Gambar 3.1 Diagram Alir Pembuatan Perangkat Lunak

3.1 Deskripsi Umum Proses Penjadwalan

Penjadwalan produksi pada bidang usaha konveksi yang dilakukan dalam skripsi ini mengacu pada kendala keterbatasan modal. Dimana dengan modal yang ada, semua *order* dapat terselesaikan tepat waktu. Untuk melakukan proses penjadwalan dibutuhkan data-data *order* sebagai berikut :

- a. Banyaknya pesanan (Q)
Banyaknya pesanan merupakan jumlah pesanan yang dipesan dalam 1 *order*.
- b. Jenis pesanan
Jenis pesanan yaitu jenis barang yang dipesan meliputi kaos dan jaket. Jenis pesanan ini akan berpengaruh pada waktu penyelesaian *order*. Untuk jenis pesanan jaket, proses potong pola setiap harinya dapat terselesaikan 20 jaket atau 100 kaos dan proses jahit 20 jaket atau 200 kaos. Untuk proses sablon dilakukan untuk jenis *order* kaos dengan 50 kaos perhari untuk waktu penyelesaiannya. Sedang untuk proses bordir dilakukan untuk jenis *order* jaket dengan 25 jaket perhari untuk waktu penyelesaiannya.
- c. Harga per satuan
Harga satuan merupakan harga barang pesanan yang dijual pada konsumen per satu buah barang. Harga satuan digunakan untuk menghitung omset per *order*. Omset tersebut digunakan untuk menghitung profit per *order*.
- d. Jumlah uang muka (DP)
DP ini digunakan sebagai modal awal untuk proses produksi.
- e. Harga pokok produksi (HPP)
HPP merupakan jumlah biaya yang digunakan untuk memproses barang pesanan meliputi bahan baku, ongkos potong, jahit dan bordir/sablon.
- f. Biaya operasional (BO)
BO merupakan jumlah biaya yang dikeluarkan per hari diluar biaya pokok produksi yang meliputi biaya transportasi, listrik, air, listrik dan semua yang tidak termasuk dalam HPP.

3.2 Perancangan Proses

Proses yang akan dilakukan dalam menyelesaikan permasalahan penjadwalan ini menggunakan algoritma genetik. Algoritma genetik terdiri dari beberapa proses yaitu representasi kromosom, perhitungan fungsi *fitness*, proses seleksi, *crossover* dan mutasi.

Langkah-langkah implementasi algoritma genetika untuk menyelesaikan permasalahan penjadwalan ini adalah sebagai berikut:

1. Membaca data dari database yang terdiri dari:
 - a. Jenis pesanan (Jenis).
 - b. Banyaknya pesanan (Q).
 - c. Harga per satuan (Harga).
 - d. Jumlah uang muka (DP).
 - e. Harga pokok produksi (HPP).
 - f. Biaya operasional per hari (BO).
 - g. Tanggal Deadline order (Deadline).
 - h. Tanggal awal mulai proses produksi.
2. Menentukan semua parameter genetika yang diperlukan :
 - a. Jumlah individu awal pada populasi awal.
 - b. Jumlah generasi.
 - c. Probabilitas *crossover*.
 - d. Probabilitas mutasi.
3. Membangkitkan populasi awal secara acak dengan jumlah gen sejumlah order yang dijadwalkan.
4. Membentuk populasi baru dengan melakukan proses-proses genetika. Langkah-langkahnya sebagai berikut :
 - a. Melakukan proses *crossover* dengan metode *Position-Based crossover* dua titik dan mengacu pada probabilitas *crossover* (P_c) yang sudah ditentukan, bila jumlah kromosom yang terpilih lebih dari 2 kromosom maka dikombinasikan kromosom-kromosom tersebut.
 - b. Melakukan proses mutasi menggunakan metode *Reciprocal Exchange Mutation* dengan titik mutasi ditentukan berdasarkan probabilitas mutasi (P_m).
 - c. Mengevaluasi semua individu yang ada dengan menghitung nilai *fitness*.
 - d. Melakukan seleksi menggunakan metode Ranking dengan membuang kromosom yang mempunyai nilai

fitness terendah dan menyisakan individu sejumlah populasi awal.

e. Menempatkan anak baru pada populasi baru.

5. Menggunakan populasi baru untuk proses selanjutnya
6. Kembali ke proses 4 sampai sejumlah generasi yang ditentukan.
7. Mengevaluasi individu dari populasi akhir yang dihasilkan dengan menghitung nilai *fitness*.
8. Nilai *fitness* tertinggi merupakan solusi.

Proses penjadwalan produksi barang pesanan ditunjukkan pada gambar 3.2

3.2.1 Representasi Kromosom

Pada penelitian ini representasi kromosom dilakukan berdasarkan *order* yang ada. Gen yang menyusun kromosom melambangkan nama *order* yang akan dijadwalkan. Jumlah gen yang terbentuk dalam satu kromosom sama dengan jumlah *order* yang akan dijadwalkan. Pengkodean yang digunakan yaitu pengkodean permutasi. Contoh permasalahan yang digunakan pada penjadwalan produksi barang pesanan ini adalah 5 *order* yang digambarkan pada tabel 3.1.

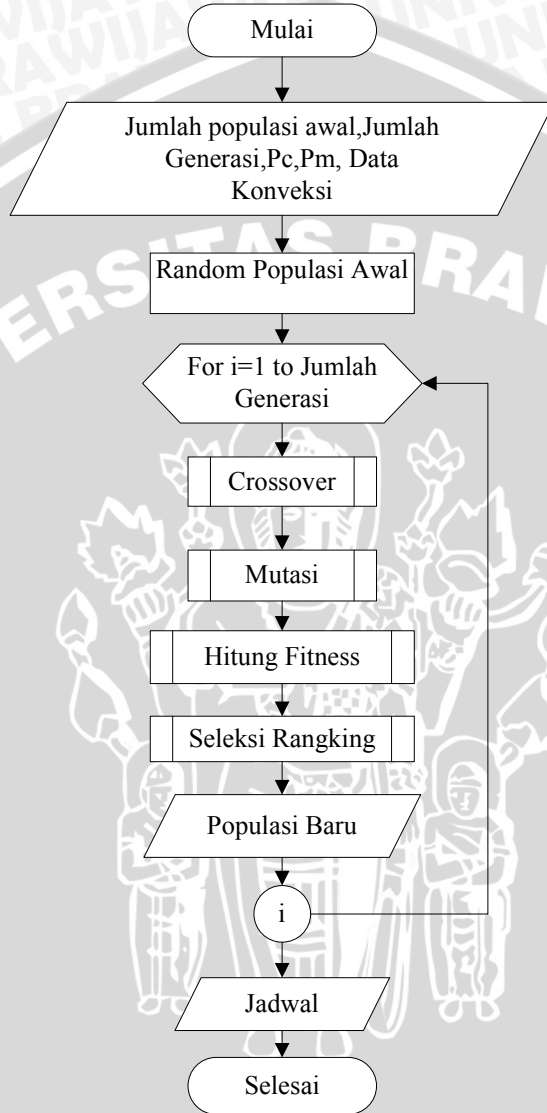
Tabel 3.1 Data *Order*

Order	Jenis	Q	Harga (Ribuan)	Dp (Ribuan)	Hpp		
					BBB (Ribuan)	BTKL (Ribuan)	BOP (Ribuan)
A	Jaket	40	80	1600	1100	400	500
B	Jaket	40	90	1890	1300	500	150
C	Kaos	80	40	1600	1800	290	280
D	Jaket	50	92	2300	2000	400	200
E	Kaos	100	50	2500	1900	420	120

Tabel 3.2 *Deadline Order*

Order	Deadline
A	12 April 2011
B	16 April 2011
C	10 April 2011
D	17 April 2011
E	15 April 2011

contoh representasi kromosom : B A C D E



Gambar 3.2 Flowchart Proses Penjadwalan Produksi Barang

3.2.2 Fungsi *Fitness*

Solusi optimal yang dicari pada penjadwalan ini termasuk ke dalam kasus maximal, artinya jika digunakan notasi *profit* untuk menghitung laba maka nilai *fitness*nya adalah *profit*. Sedangkan *profit* itu sendiri dirumuskan sebagai berikut :

$$profitNetto = \sum_{i=1}^n profitBruto(i) - (BO * hari) \quad (3.1)$$

Keterangan :

- profitNetto* = total laba setelah dikurangi biaya operasional.
- profitBruto* = laba per order.
- BO* = biaya operasional per hari.
- Hari* = hari penyelesaian semua order.

Hari penyelesaian tergantung dengan jumlah modal (uang yang tersedia) yang tersedia untuk membeli bahan baku (belanja). Dalam 1 hari hanya diperbolehkan belanja max. 2 order, proses potong : jaket sebanyak 20 buah atau kaos sebanyak 100 buah, proses sablon digunakan hanya untuk jenis kaos sebanyak 50 buah, proses bordir digunakan hanya untuk jenis jaket sebanyak 25 buah dan proses jahit : sebanyak 20 buah atau kaos sebanyak 200 buah. Proses dilakukn secara berurutan: belanja → potong → bordir / sablon → jahit. Proses selanjutnya bisa dilakukan bila proses sebelumnya telah selesai dan dalam 1 hari 1 order hanya diperbolehkan melakukan 1 proses saja.

$$profitBruto(i) = (Q(i) * harga(i)) - hpp(i) \quad (3.2)$$

Keterangan :

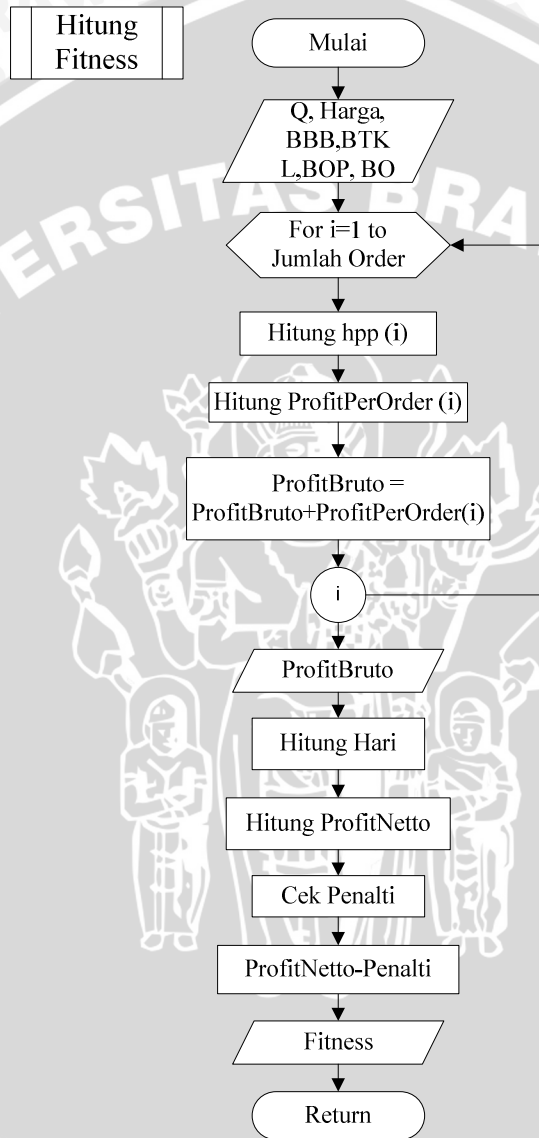
- Q(i)* = jumlah pesanan untuk *order* ke-i.
- Harga(i)* = harga persatuan untuk *order* ke-i.
- Hpp(i)* = harga pokok produksi untuk *order* ke-i.

Kemudian dilakukan pengecekan, apakah penyelesaian *order* melebihi *deadline* yang ditentukan? Jika iya, maka pemesan akan dikenakan biaya penalti sebesar 5% perhari dari omset *order* yang melebihi *deadline*.

$$penalti = omset * 5\% \quad (3.3)$$

$$fitness = profitNetto - penalti \quad (3.4)$$

Proses menghitung nilai *fitness* dari tiap kromosom digambarkan pada gambar 3.3.



Gambar 3.3 Flowchart Proses Hitung *Fitness*

3.2.3 Crossover

Crossover yang digunakan pada skripsi ini yaitu metode *Position-Based Crossover*. Proses *crossover* dilakukan menggunakan kromosom yang terpilih (*parent*) berdasarkan nilai probabilitas *crossover* (P_c). Jika jumlah kromosom yang terpilih lebih dari 2 kromosom maka dikombinasikan kromosom-kromosom tersebut. Pengkombinasian kromosom dilakukan dengan memasangkan semua gen dengan menerapkan prinsip kombinasi yaitu memasangkan 2 kromosom dengan tidak memerhatikan urutan, misal : A C sama dengan C A. Jumlah perpasangan kromosom dapat dihitung dengan

$$\text{rumus : } \frac{n!}{2!(n-2)!} \quad (3.5)$$

Keterangan :

n = jumlah kromosom yang terpilih berdasarkan P_c

Setelah diperoleh *parent*, pertama yang dilakukan dalam metode *Position Based Crossover* yaitu memilih secara acak beberapa posisi dalam *parent* 1 berdasarkan nilai probabilitas *crossover*. Kemudian dihasilkan *proto-child* dengan menyalin gen pada posisi yang dipilih tadi dan diletakkan sama seperti pada posisi awal. Setelah itu, menghapus gen-gen pada *parent* 2 yang sama dengan yang terpilih pada *parent* 1. Langkah yang terakhir dengan menempatkan gen-gen *parent* 2 tidak terhapus pada posisi yang kosong didalam *proto-child* dari kiri ke kanan sesuai dengan urutan pada *parent* 2.

1. Memilih posisi pada *parent* 1.

Parent 1 : A **B** C **D** **E** F

Parent 2 : F E B A C D

2. Menyalin Gen yang sudah terpilih.

Proto-child : - **B** - **D** **E** -

3. Menghapus gen-gen yang sama dengan *proto-child*.

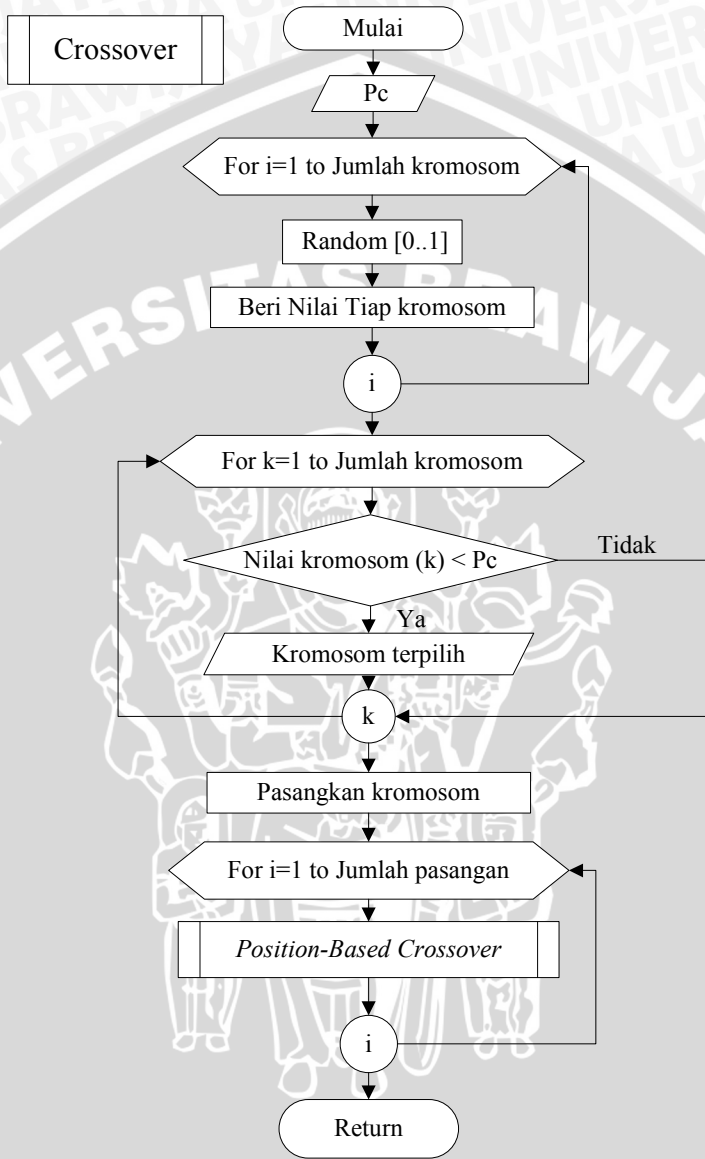
Parent 2 : **F** - - **A** **C** -

4. Menempatkan gen *parent* 2 ke *proto-child*.

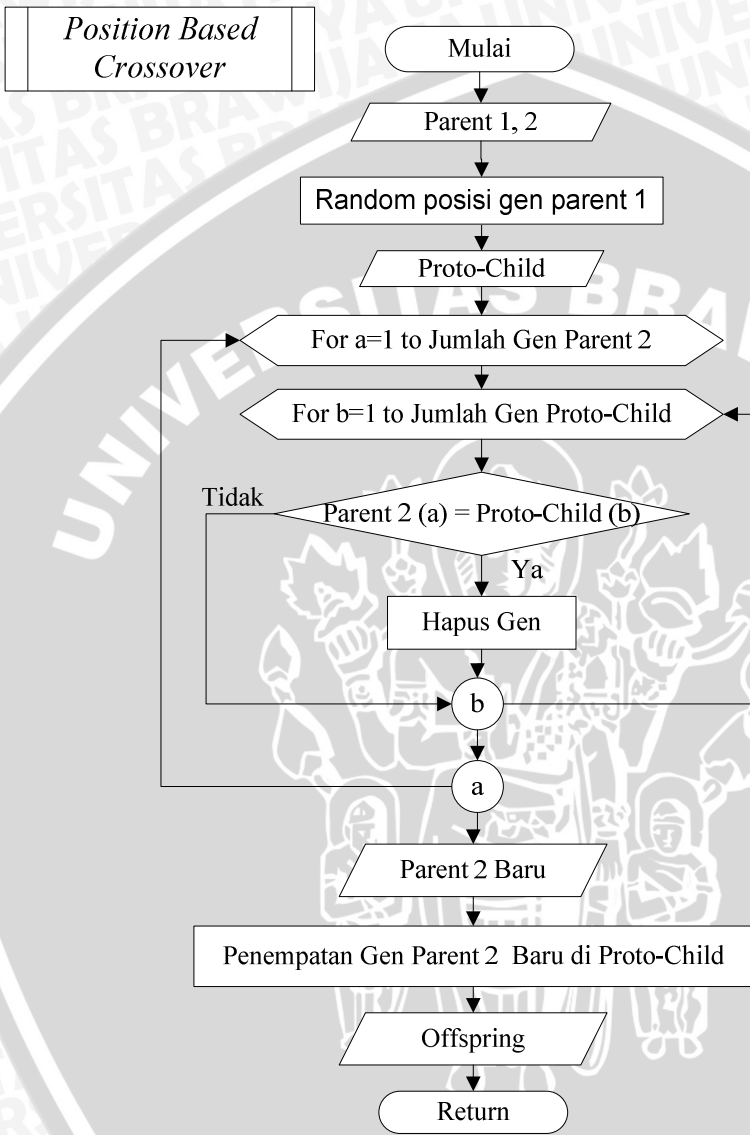
Offspring : **F** **B** **A** **D** **E** **C**

Gambar 3.4 Ilustrasi *Crossover*

Proses *Position Based Crossover* dapat dilihat pada gambar 3.6 . Proses *crossover* (perkawinan silang) secara keseluruhan sehingga mendapatkan *offspring* dapat dilihat pada gambar 3.5.



Gambar 3.5 Flowchart Proses *Crossover*



Gambar 3.6 Flowchart Proses *Position Based Crossover*

3.2.4 Mutasi

Proses mutasi dilakukan bertujuan untuk mendapatkan variasi individu dalam satu populasi. Metode mutasi yang digunakan adalah *Reciprocal Exchange Mutation*. Kromosom yang mengalami mutasi dipilih berdasarkan probabilitas mutasi (P_m). Mutasi dilakukan dengan memilih secara acak posisi yang akan ditukar berdasarkan probabilitas mutasi. Proses mutasi dapat dilihat pada gambar 3.8.

1. Memilih secara acak 2 posisi titik mutasi.
Misal titik yang dipilih 2 dan 4

Parent

B	E	C	D	A
---	---	---	---	---

2. Tukar 2 gen yang terpilih.

Offspring

B	D	C	E	A
---	---	---	---	---

Gambar 3.7 Ilustrasi *Reciprocal Exchange Mutation*

3.2.5 Seleksi Rangkings

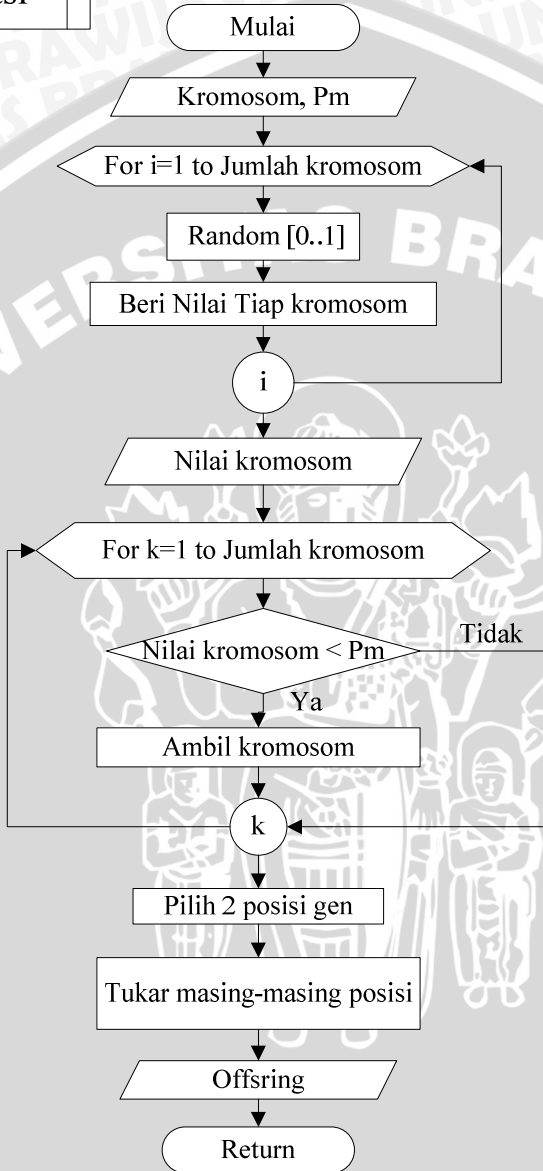
Proses seleksi dilakukan dengan tujuan untuk memilih kromosom-kromosom yang baik untuk dijadikan populasi baru. Metode yang digunakan pada proses seleksi ini yaitu metode Rangkings. Pertama-tama menghitung nilai *fitness* dari masing-masing kromosom. Kemudian mengurutkan kromosom dari *fitness* terendah hingga tertinggi. Contoh seleksi menggunakan metode Rangkings dapat dilihat pada tabel 3.3 :

Tabel 3.3 Perhitungan untuk seleksi menggunakan metode Rangkings

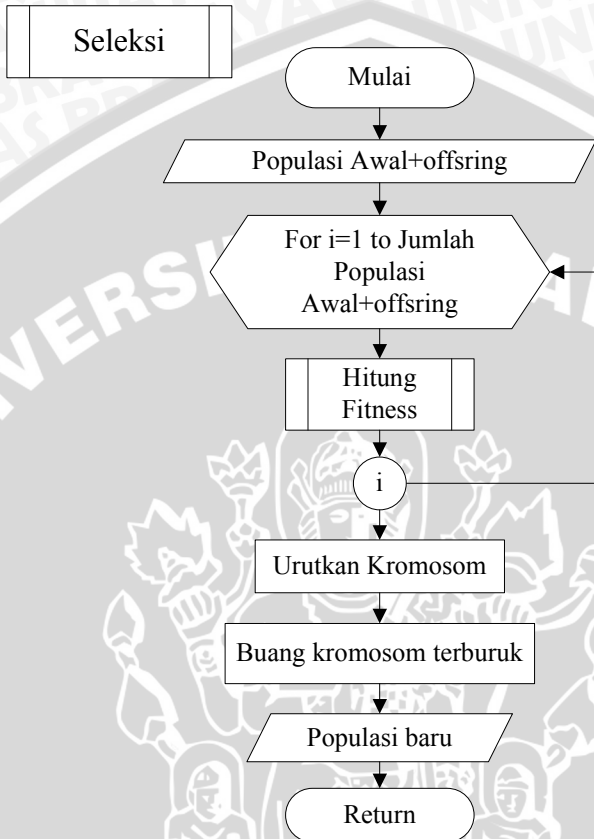
Kromosom	<i>Fitness</i>	<i>Fitness</i> baru
A B C D E	5	1
A C B D E	6	2
A B C E D	9	3
B A C D E	10	4
B A D C E	15	5

Proses seleksi Rangkings kromosom digambarkan pada gambar 3.9.

Mutasi



Gambar 3.8 Flowchart Proses Mutasi



Gambar 3.9 Flowchart Proses Seleksi Ranking.

3.3 Rancangan Basis Data

Basis data merupakan salah satu komponen utama aplikasi ini. Basis data digunakan untuk menyimpan data order. Basis data yang digunakan untuk proses penjadwalan ini terdiri dari 1 tabel. Tabel tersebut terdiri dari :

Nama	Tipe Data
NamaOrder	Text
Jenis	Text
Q	Number
DP	Number
BBB	Number
BTKL	Number
BOP	Number
Deadline	Date/Time

3.4 Perhitungan Manual

Contoh permasalahan yang digunakan pada penjadwalan produksi konveksi ini terdiri dari 5 buah *order* yang dijadwalkan dengan biaya operasional perhari (BO) sebesar Rp. 100.000,- dengan data yang tersedia sebagai berikut :

Tabel 3.4 Data Order

Order	Jenis	Q	Harga (Ribuan)	Dp (Ribuan)	Hpp		
					BBB (Ribuan)	BTKL (Ribuan)	BOP (Ribuan)
A	Jaket	40	80	1600	1600	400	500
B	Jaket	42	90	1890	1890	500	150
C	Kaos	80	40	1600	1600	290	280
D	Jaket	50	92	2300	2300	400	200
E	Kaos	100	50	2500	2500	420	120

Tabel 3.5 *Deadline Order*

Order	Deadline
A	25 April 2011
B	30 April 2011
C	24 April 2011
D	27 April 2011
E	19 April 2011

Biaya operasional per hari (BO) = Rp. 100.000,-

3.4.1 Representasi Kromosom

Representasi kromosom dilakukan berdasarkan nama *order* yang ada dengan menggunakan pengkodean permutasi. Contoh representasi kromosomnya sebagai berikut:

Pembentukan 5 kromosom secara acak (populasi awal) :

1. Kromosom 1 : A B C D E
2. Kromosom 2 : E D C B A
3. Kromosom 3 : A C D E B
4. Kromosom 4 : C E A D B
5. Kromosom 5 : C E D B A

3.4.2 Crossover

Proses selanjutnya adalah *crossover*. Metode yang digunakan untuk proses *crossover* adalah *Position Based Crossover*. Proses *crossover* tergantung pada suatu parameter yaitu probabilitas *crossover* (P_c). P_c adalah 0,4. Langkah-langkah proses *Position Based Crossover* sebagai berikut:

1. Membangkitkan nilai *random* antara 0 sampai 1 untuk setiap kromosom.
 - a. Kromosom 1 : A B C D E \rightarrow 0,98
 - b. Kromosom 2 : E D C B A \rightarrow 0,35
 - c. Kromosom 3 : A C D E B \rightarrow 0,38
 - d. Kromosom 4 : C E A D B \rightarrow 0,42
 - e. Kromosom 5 : C E D B A \rightarrow 0,25
2. Kromosom yang mempunyai nilai $< 0,4$ (P_c) maka kromosom tersebut yang mengalami *crossover*.
 - a. Kromosom 2 : E D C B A \rightarrow 0,35
 - b. Kromosom 3 : A C D E B \rightarrow 0,38
 - c. Kromosom 5 : C E D B A \rightarrow 0,25

3. Memasangkan kromosom.
 - a. E D C B A dan A C D E B
 - b. E D C B A dan C E D B A
 - c. A C D E B dan C E D B A
4. Memilih secara acak posisi gen pada *parent 1*. Jumlah posisi gen yang dipilih adalah setengah dari jumlah gen.
 - a. Jumlah posisi = jumlah gen : 2
= $5:2 = 2,5 \approx 3$ posisi (dibulatkan ke atas)
 - b. Posisi gen yang terpilih : 2, 4, 5

Pasangan I :

Parent 1 : E **D** C B A

Parent 2 : A C D E B

Pasangan II :

Parent 1 : E **D** C B A

Parent 2 : C E D B A

Pasangan III :

Parent 1 : A **C** D E B

Parent 2 : C E D B A

5. Menyalin gen yang sudah terpilih kedalam *protochild*.

Pasangan I :

Proto-child : - **D** - **B** A

Pasangan II :

Proto-child : - **D** - **B** A

Pasangan III :

Proto-child : - **C** - **E** B

6. Menghapus gen-gen pada *parent 2* yang sama dengan *proto-child*.

Pasangan I :

Parent 2 : - **C** - **E** -

Pasangan II :

Parent 2 : **C** **E** - - -

Pasangan III :

Parent 2 : - - - **D** - **A**

7. Menempatkan gen *parent 2* ke *proto-child*.

UNIVERSITAS BRAWIJAYA

