BAB II

TINJAUAN PUSTAKA

2.1 PENDAHULUAN

PT.Hart.Co adalah sebuah perusahaan manufaktur yang bergerak di bidang *furniture* dengan tipe aliran proses produksi *job shop*. Seperti perusahaan *furniture* pada umumnya, dalam melaksanakan proses produksi, banyak operasi yang dikerjakan dengan menggunakan tenaga manusia yang kecepatannya tidak konstan, sehingga lama waktu proses pengerjaan sebuah produk selalu berubah ubah. Untuk itulah diperlukan sebuah jadwal produksi yang dapat mengakomodasi ketidaktepatan dan ketidakpastian waktu proses yang disebabkan oleh penggunaan tenaga manusia tersebut.

Sampai saat ini, metode yang dikembangkan untuk menyelesaikan permasalahan penjadwalan produksi baik untuk tipe aliran *flow shop* maupun *job shop* sudah cukup banyak ditemui. Metode-metode tersebut antara lain adalah (Nasution, 1999) :

- Metode *Short Processing Time* (SPT).
- Metode *Weighted Shortest Processing Time* (WSPT).
- Metode *Earliest Due Date* (EDD).

Selain metode *sequencing* di atas, terdapat juga metode yang melakukan pencarian *heuristic* untuk mendapatkan solusi optimum dengan menggunakan

algoritma genetik dengan fungsi tujuan untuk meminimasi *makespan* (Trisnawati, 2004)

Dari beberapa metode yang telah dijelaskan, terdapat beberapa kekurangan seandainya metode tersebut diterapkan ke dalam sistem nyata. Ketergantungan metode *sequencing* akan data yang bersifat pasti, akan sulit diterapkan ke dalam sistem nyata yang pada umumnya berubah ubah.

Pada metode pencarian *heuristic,* sistem pencarian yang terbentuk bersifat kaku, karena perbedaan kecil yang terdapat pada sebuah angka akan menyebabkan sistem melakukan penolakan terhadap angka tersebut bisa saja terjadi, karena sistem tidak akan mengetahui apabila angka yang ditolak tersebut memiliki potensi yang lebih besar untuk memberikan solusi yang lebih baik.

Ketidakpastian dan ketidaktepatan waktu proses yang terjadi dalam sebuah proses produksi tidak dapat diselesaikan oleh model penjadwalan deterministik yang hanya bergantung pada data yang tepat dan metode *heuristic* yang tidak mampu untuk merepresentasikan pandangan manusia mengenai beberapa kemungkinan yang dapat terjadi. Metode deterministik hanya berlaku jika semua operasi dilakukan secara otomatis dengan menggunakan mesin, karena kecepatan proses yang dilakukan dengan menggunkan mesin bersifat konstan.

Oleh karena itu, algoritma dan model penjadwalan produksi sebaiknya dikembangkan dengan menggunakan himpunan fuzzy yang berfungsi untuk memberikan ukuran mengenai ketidakpastian dan ketidaktepatan waktu proses dan memberikan keputusan mengenai pekerjaan yang terambat sesuai dengan

Pendekatan yang selalu bergantung pada data yang tepat, tidak akan sesuai apabila diterapkan dalam sistem nyata yang besifat dinamis - stokastik. Sehingga model penjadwalan deterministik dan algoritmanya harus dikembangkan dengan menggunakan *tools* dari *Artificial Intelligence* untuk mengatasi masalah tersebut.

2.3 HIMPUNAN FUZZY

Logika fuzzy adalah logika yang mendasari penalaran dengan menggunakan pendekatan (*approximate reasoning*) disamping penalaran secara akurat (*exact reasoning*). Disinilah letak kelebihan dari logika fuzzy, karena berdasarkan fakta yang ada, bahwa kebanyakan jenis penalaran manusia, terutama penalaran dengan menggunakan intuisi, menggunakan perkiraan yang paling mendekati solusi dari permasalahan yang dihadapi.

Menggunakan logika klasik, hanya dapat dilakukan pada informasi yang bersifat benar atau salah. Logika ini tidak dapat digunakan untuk mengatasi masalah yang berhubungan dengan informasi yang bersifat tidak lengkap dan tidak tepat, namun, dalam informasi ini terdapat data yang dapat menghasilkan pemecahan yang lebih baik untuk mengatasi masalah.

Pada logika himpunan tegas. nilai keanggotaan dari elemen yang dimiliki pada himpunannya bernilai 0 jika elemen itu berada di luar himpunan, dan bernilai 1 jika elemen itu berada di dalam himpunan. Sedangkan pada himpunan fuzzy, nilai keanggotaan dari elemennya diperluas, yaitu berada dalam *range* [0 , 1]. Oleh karena itu, dapat dikatakan bahwa logika fuzzy adalah perluasan dari himpunan klasik.

Himpunan fuzzy dapat didefinisikan secara formal sebagai berikut (Galindo et.al, 2006) :

$$A = \{\mu_A(x) \mid x : x \in X, \mu_A(x) \in [0,1] \in R\}$$

$\mu_A(x) = 0$; mengindikasikan bahwa x bukan merupakan anggota himpunan A

$\mu_A(x) = 1$; mengindikasikan bahwa x merupakan anggota himpunan A secara penuh

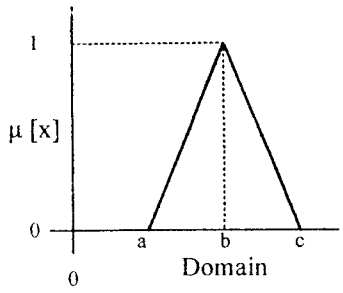
$\mu_A(x)$ berfungsi untuk memberikan keterangan khusus dari semua elemen yang ada pada sebuah himpunan dengan menetukan fungsi keanggotaan (*membership function*) setiap elemen pada himpunan tersebut.

2.3.1 FUNGSI KEANGGOTAAN PADA HIMPUNAN FUZZY

Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan pada himpunan fuzzy adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan seperti *triangular, trapezodial, bell-curves* dan *s-curves.* Pada umumnya, pemilihan bentuk fungsi merupakan hal yang subyektif, tergantung pada tujuan dari model yang akan dibangun.

2.3.1.a Himpunan Fuzzy Segitiga

Kurva Segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) yang berujung pada sebuah titik dengan nilai keanggotaan 1. Kurva ini memiliki batasan bawah (a), batasan atas (c) dan *modal point* (b) (Galindo et.al, 2006).



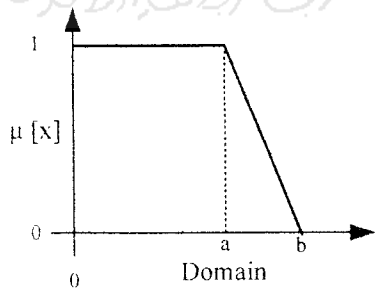
Gambar 2.1 *Triangular Fuzzy Set*

Dengan fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & \text{if } x \le a \\ (x-a)/(b-a); & \text{if } x \in (a,b) \\ (c-x)/(c-b); & \text{if } x \in (b,c) \\ 1; & \text{if } x \ge c \end{cases}$$

2.3.1.b Himpunan Fuzzy *Trapezodial* dengan L Fuzzy *Set* (*Right*)

Kurva Trapesium pada dasarnya hanya berbentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1. Pada himpunan ini, fungsi keanggotaan didefinisikan oleh dua parameter, yaitu a dan b (Galindo et.al, 2006).



Gambar 2.2 *Trapezodial Fuzzy Set (Right)*

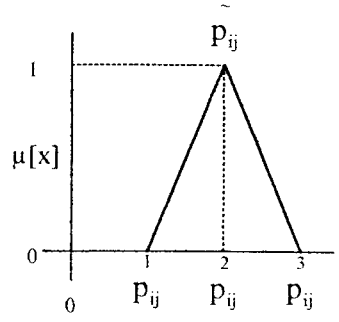
Dengan Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & \text{if } x > b \\ (x-a)/(b-a); & \text{if } a \le x \le b \\ 1; & \text{if } a \le x \end{cases}$$

2.3.2 APLIKASI LOGIKA FUZZY PADA PENJADWALAN PRODUKSI

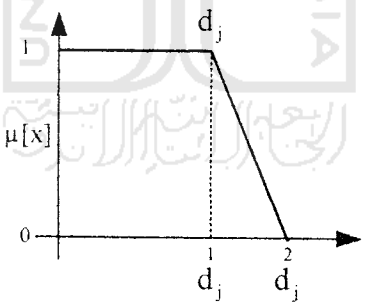
Estimasi waktu proses pada setiap operasi disesuaikan dengan kondisi penggunaan mesin. Beberapa mesin berjalan secara otomatis dan dapat dioperasikan dengan kecepatan yang berbeda dan konstan. Beberapa mesin lainnya dioperasikan dengan menggunakan tenaga manusia yang menyebabkan lamanya waktu proses bergantung pada kecepatan manusia yang bersifat tidak tetap.

Ketidak pastian waktu proses $\tilde{p}_j^i$ dimodelkan dengan menggunakan fungsi keanggotan dengan kurva segitiga. Pada kurva tersebut, fungsi keanggotaan digambarkan dalam tiga lapis (*triplet*) (p_j^1, p_j^2, p_j^3), dimana p_j^1 dan p_j^3 adalah batasan atas dan batasan bawah dari toleransi yang diberikan pada waktu proses, dan p_j^2 atau *modal point* digunakan untuk merpresentasikan waktu proses yang diharapkan (Fayad dan Petrovic, 2005).



Gambar 2.3 Waktu proses fuzzy pada kurva segitiga

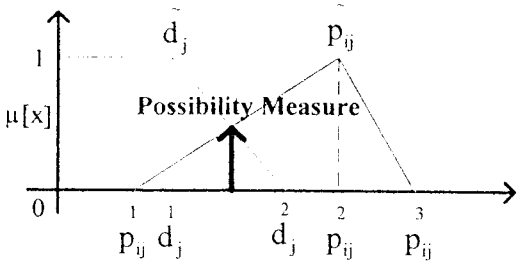
Himpunan Fuzzy *Trapezodial* dengan L Fuzzy *Set* (*Right*) digunakan untuk memodelkan *due date* $\tilde{d}_j^i$ pada setiap *job*, yang digambarkan dalam dua lapis fungsi keanggotaan (*doublet*) (d_j^1, d_j^2). Dimana d_j^1 adalah *due date* awal dengan menggunakan himpunan tegas, dan d_j^2 dari *trapezoid* sesuai dengan besarnya prosentase toleransi dari *due date* awal yang digunakan sebagai nilai toleransi (Fayad dan Petrovic, 2005).



Gambar 2.4 *Due date* fuzzy

Ketidak pastian waktu proses akan mempengaruhi waktu selesainya sebuah produk. Dan ketidak pastian waktu selesainya sebuah produk akan mempengaruhi *due date* dari produk tersebut. Maka, untuk membandingkan waktu selesai produk

dengan nilai fuzzy dengan *due date* produk yang juga bernilai fuzzy, digunakan *possibility measure* untuk mengetahui kemungkinan terjadinya waktu selesai produk fuzzy ($\tilde{c}_j$) dalam himpunan fuzzy *due date* ($\tilde{d}_j^i$) sebagai berikut (Fayad dan Petrovic, 2005) :



Gambar 2.5 *Possibility Measure* untuk mencari nilai keanggotaan fuzzy C_j pada himpunan fuzzy d_j

$$SG_T(C_j) = \sup_{x \in X} \min\{\mu_{\tilde{C}_j}(t), \mu_{\tilde{D}_j}(t)\}$$

dimana $\mu_{c_j}(t)$ dan $\mu_{d_j}(t)$ adalah fungsi keanggotaan dari himpunan fuzzy $\tilde{c}_j$ dan $\tilde{d}_j$.

2.3.2.a Tingkat Kepuasan dari Rata-Rata Keterlambatan

Tingkat kepuasan dari rata-rata keterlambatan (S_{AT}) dapat diperoleh dengan mencari nilai rata-rata tingkat kepuasan *completion time* dari masing-masing produk yang akan dijadwalkan.

$$S_{AT} = \frac{1}{n}\sum_{j=1}^{n} SG_T\left(\tilde{C}_j\right), j = 1, 2, 3, \ldots n$$

$SG_T\left(\tilde{C}_j\right)$ adalah tingkat kepuasan fuzzy yang dimiliki oleh *completion time* setiap material yang dijadwalkan. Ukuran $SG_T\left(\tilde{C}_j\right)$ ini terbagi lagi menjadi 3 bagian, yaitu :

1. Kondisi pertama :

   Jika $p_j^2 < d_j^1$, maka SG = 1

2. Kondisi kedua :

   Jika $d_j^1 \leq p_j^1 < d_j^2$ atau $(p_j^1 \leq d_j^1$ dan $p_j^2 > d_j^2)$ atau

   $(p_j^1 \leq d_j^1$ dan $d_j^1 \leq p_j^2 < d_j^2)$,

   maka $SG = \sup_{x \in X} \min\{\mu_{dj}(t); \mu_{pj}(t)\}$

3. Kondisi ketiga :

   Jika $(p_j^1 \geq d_j^2$ dan $p_j^2 \geq d_j^2)$, maka SG = 0

Kemudian untuk menentukan apakah material yang dijadwalkan terlambat atau tidak, digunakan variabel λ, dengan $0 \leq \lambda \leq 1$. Jika nilai fuzzy $SG_T\left(\tilde{C}_j\right)$ dari sebuah $job_{(j)}$ lebih kecil dari nilai λ yang ditentukan di awal, maka *job* tersebut dinyatakan terlambat.

2.3.2.b Tingkat Kepuasan Jumlah Pekerjaan yang Terlambat

Setelah menghitung rata-rata jumlah *job* yang terlambat, untuk mencari tingkat kepuasan dari jumlah *job* yang terlambat (S_{NT}) ditentukan dengan menggunakan model sebagai berikut (Fayad dan Petrovic, 2005) :

$$S_{NT} = \begin{cases} 1 & \text{jika nTardy} = 0 \\ (n'' - nTardy)/n'' & \text{jika } 0 < \text{nTardy} < n'' \\ 0 & \text{jika nTardy} > n'' \end{cases}$$

Dengan nilai n" adalah batasan jumlah maksimal untuk jumlah produk yang terlambat yang tetap akan dimasukkan ke dalam jadwal produksi yang akan dibentuk.

2.3.2.c Model Optimasi untuk Mencari Solusi Jadwal Optimum

Dari dua fungsi tujuan yang telah ditetapkan, dapat dibentuk formulasi matematis untuk mendapatkan solusi optimum.

Fungsi Tujuan :

$$\max_{1 \le g \le n} \left\{ \max_{1 \le p \le n} \left(\frac{S_{AT(gp)} + S_{NT(gp)}}{2} \right) \right\}$$

Dimana g = generasi, p = populasi

Batasan *Presedence* :

$c_{jom} - c_{j(o-1)(m-1)} \geq t_{jom}$

Dimana c_{jom} adalah waktu selesai proses *job* j, operasi o, pada mesin m, $c_{j(o-1)(m-1)}$ adalah waktu selesai *job* j pada mesin sebelum mesin k. t_{jom} adalah waktu proses *job* j, operasi o, pada mesin m.

2.4 ALGORITMA GENETIK

Algoritma Genetik (AG) adalah teknik pencarian stokastik yang berdasarkan pada mekanisme dari seleksi alami dan genetika alami atau secara umum disebut dengan evolusi biologis (Gen dan Cheng, 1997).

2.4.1 STRUKTUR UMUM ALGORITMA GENETIKA

Berbeda dengan metode pencarian yang lain yang hanya melakukan pencarian pada lingkup solusi, AG dimulai dari pembentukan himpunan dari solusi awal secara acak yang disebut dengan populasi. Setiap organisme atau individu dalam sebuah populasi dinamakan kromosom, dimana setiap kromosomnya menggambarkan solusi dari sebuah permasalahan. Kromosom ini terdiri dari beberapa bagian yang disebut dengan *gen* yang berbentuk simbol.

Seperti pada evolusi di sebuah organisme, dalam AG, kromosom pada sebuah populasi juga dapat berevolusi melalui sebuah iterasi yang dinamakan generasi. Pada setiap proses generasi, dilakukan evaluasi pada setiap kromosom berdasarkan dari nilai *fitness* yang dimilikinya. Untuk membentuk sebuah generasi berikutnya apabila kromosom awal bukan merupakan kondisi akhir, terlebih dahulu dibentuk sebuah kromosom baru yang disebut dengan kromosom anak atau *offspring*, yang dapat dilakukan dengan mengunakan 2 operator genetika, kemudian setelah didapatkan kromosom anak, sebuah generasi baru dapat terbentuk dengan melakukan seleksi untuk mendapatkan nilai *fitness* tertinggi.

Semakin tinggi nilai *fitness* dari sebuah kromosom, semakin tinggi pula kemungkinannya untuk dipilih menjadi bagian dari populasi baru. Generasi biasanya dilakukan berulang-ulang hingga menemukan solusi optimal dari lingkup pencarian yang dilakukan atau hanya melakukan sekali generasi saja

untuk mendapatkan solusi dengan konsekuensi solusi yang lebih optimal mungkin tidak akan tercapai.

Setelah beberapa generasi dilakukan, AG akan menemukan kromosom-kromosom terbaik yang menggambarkan solusi optimal atau sub-optimal dari sebuah permasalahan.

2.4.2 ALGORITMA GENETIK UNTUK PENJADWALAN PRODUKSI FUZZY

Karakteristik utama dari algorima genetik untuk menyelesaikan masalah penjadwalan produksi dengan *due date* dan waktu penyelesaian produk fuzzy pada tipe manufaktur *job shop* digambarkan sebagai berikut :

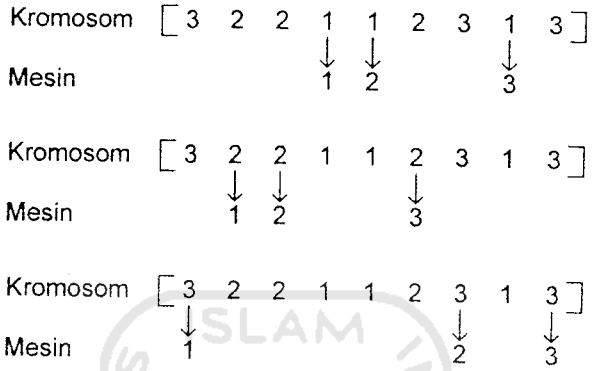
2.4.2.a Kromosom

Representasi kromosom dilakukan secara langsung berdasarkan operasi yang dilakukan atau *operation based chromosome*. Representasi ini meng-*encode*-kan sebuah jadwal sesuai dengan urutan operasinya dengan panjang *m* (mesin) x *n* (*job*), dan setiap *gen* mewakili sebuah operasi. *Gen* ini berupa simbol yang sama dari operasi sebuah *job* dan menggambarkan kejadian urutan penjadwalan pada kromosom yang diberikan (Gen dan Cheng, 1997).

Jika terdapat kromosom sebagai berikut [3 2 2 1 1 2 3 1 3] dimana angka 1 mewakili *job* 1, 2 untuk *job* 2, dan 3 untuk *job* 3, sesuai pada tabel 2.1 dapat dijelaskan sebagai berikut :

Tabel 2.1 Permasalahan 3 *job* 3 mesin

	Urutan Mesin		
	Operasi		
Job	1	2	3
J_1	m_1	m_2	m_3
J_2	m_1	m_3	m_2
J_3	m_2	m_1	m_3



Gambar 2.6 Operasi dari *job* pada mesin

2.4.2.b Inisialisasi

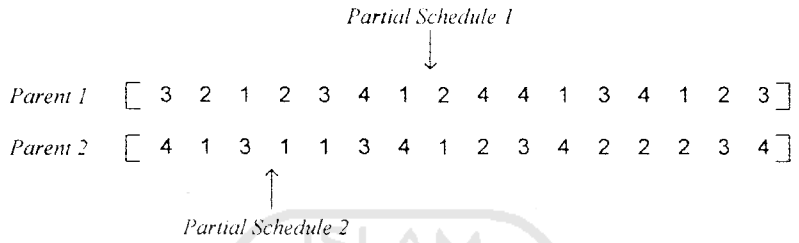
Inisialisasi dilakukan pada solusi awal yang terbentuk secara acak untuk mengetahui nilai *fitness* (pada permasalahan ini, nilai *fitness* adalah tingkat kepuasan fuzzy dari jadwal yang terbentuk) yang ada pada masing-masing kromosom dengan memperhatikan bahwa jumlah operasi sesuai dengan jumlah pekerjaan yang ada.

2.4.2.c Persilangan (*Crossover*)

Operator persilangan yang akan digunakan pada permasalahan menggunakan pendekatan Gen, Tsujimura dan Kubota. Operator ini sesuai dengan representasi

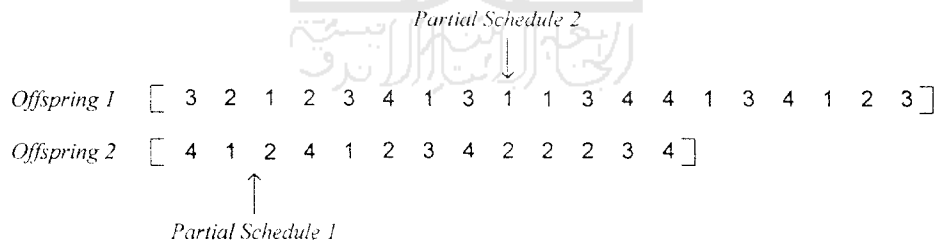
kromosom berdasarkan operasi yang dinamakan *partial schedule exchange crossover* (Gen dan Cheng, 1997).

Bagian dari kromosom yang akan disilangkan (*partial schedule*) pada setiap induk ditentukan dari pekerjaan yang sama pada *range* posisi yang pertama dan posisi selanjutnya yang terdekat.



Gambar 2.7 Pemilihan bagian kromosom yang akan disilangkan

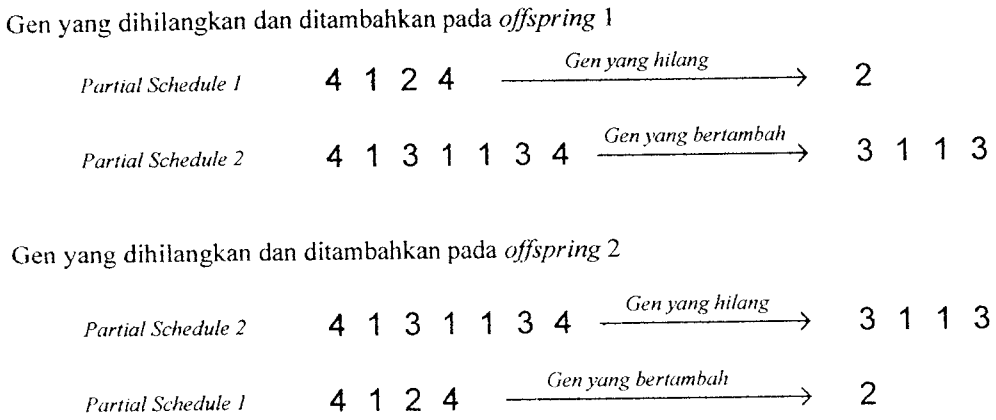
Menyilangkan *partial schedule* seperti pada gambar 2.12 akan menyebabkan perbedaan jumlah kromosom anak (*offspring*) yang terbentuk, sehingga *offspring* yang dihasilkan bersifat *illegal*.



Gambar 2.8 *Partial schedule* yang telah di silangkan

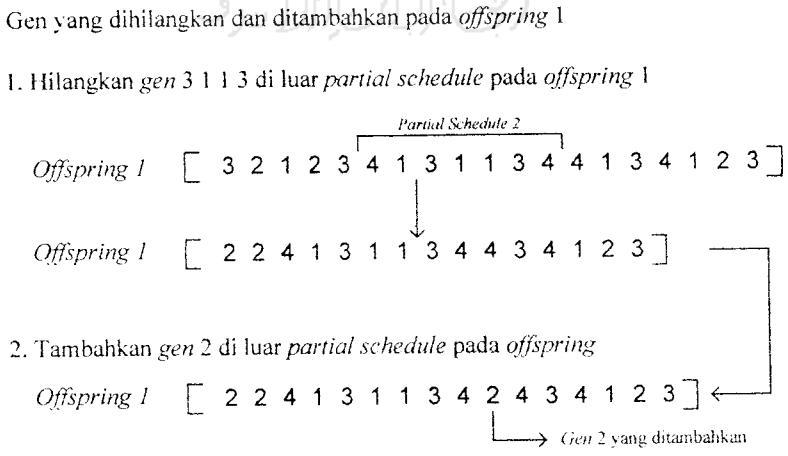
Langkah berikutnya adalah untuk melegalkan *offspring* yang dihasilkan, dilakukan penghapusan *gen* yang lebih dan melakukan penambahan *gen* yang

kurang. Untuk *offspring* 1, *gen* yang berlebih adalah operasi (3 1 1 3). *Offspring* 1 menerima *partial schedule* 2 dari *parent* 2.

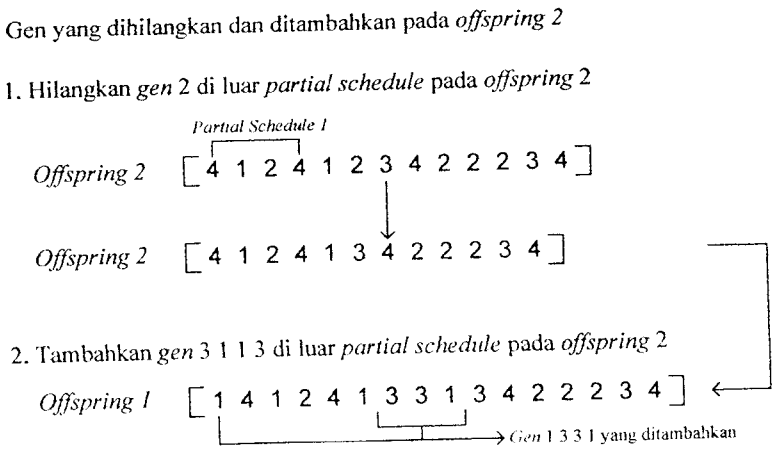


Gambar 2.9 *Gen* yang dihilangkan dan ditambahkan pada *offspring*

Karena *job* 1 dan *job* 3 pada *partial schedule* 2 adalah pekerjaan pertama yang akan diproses, maka operasi sebelum *partial schedule* 2 pada *offspring* 1 harus dihapus untuk menjaga urutan pekerjaan pada *partial schedule* 2 agar tetap sama dengan *parent* 2, untuk mewarisi sifat yang dimiliki induknya.



Gambar 2.10 Legalisasi *offspring* 1



Gambar 2.11 Legalisasi *offspring* 2

Untuk *offspring* 1, gen yang hilang adalah *job* 2. Karena tidak terdapat *job* 2 pada *partial schedule* 2, kita dapat memasukkan *job* 2 tersebut ke urutan mana saja, kecuali di dalam *partial schedule* 2, agar urutan *gen* tidak berubah.

Crossover ini dilakukan sesuai dengan probabilitas yang ditentukan, sehingga apabila $p_c = 0{,}25$ maka dilakukan prosedur sebagai berikut :

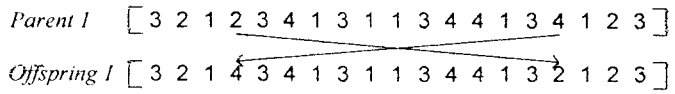
Langkah 1 : Bangkitkan bilangan *random* r sesuai dengan jumlah kromosom dengan *range* [0, 1], dimana k = 1, 2, ..., ukuran populasi

Langkah 2 : Jika $r_k < 0{,}25$ maka pilih sebagai induk untuk disilangkan

2.4.2.d Mutasi

Pada permasalahan ini, operator mutasi yang digunakan adalah *job-pair exchange mutation* yang diperkenalkan oleh Gen, Tsujimura dan Kubota. Mutasi ini dilakukan dengan cara memilih secara acak dua pekerjaan yang tidak sama dalam sebuah kromosom, dan dilakukan penukaran posisi *gen* tersebut. Pada

representasi yang berbasis operasi, penukaran *gen* yang berjarak terlalu dekat akan menghasilkan perubahan yang tidak signifikan pada solusi yang terbentuk. Oleh karena itu, semakin jauh jarak *gen* dalam kromosom yang ditukar, maka hasil yang diperoleh akan semakin baik (Gen dan Cheng, 1997)



Gambar 2.12 Mutasi

Mutasi ini dilakukan pada setiap *gen*, sesuai dengan probabilitas yang ditentukan, sehingga apabila $p_m = 0{,}01$ maka dilakukan prosedur sebagai berikut :

Langkah 1 : Bangkitkan bilangan *random* r sesuai dengan jumlah *gen* dengan *range* [0, 1], dimana k = 1, 2, ..., jumlah *gen*

Langkah 2 : Jika $r_k < 0{,}01$ maka pilih sebagai induk untuk mutasi

2.4.2.e Seleksi

Teknik seleksi dengan menggunakan *roulette wheel* berdasarkan nilai *fitness* yang diperoleh dari fugsi tujuan yang telah ditetapkan pada permasalahan ini. Langkah-langkah untuk melakukan seleksi dengan menggunakan *roulette-wheel* adalah sebagai berikut (Gen dan Cheng, 1997):

1. Evaluasi nilai *fitness* yang ada pada tiap-tiap kromosom

$$\max_{1 \le g \le n} \left\{ \max_{1 \le p \le pop\_size} \left(\frac{S_{AT(gp)} + S_{NT(gp)}}{2} \right) \right\},$$

g = 1, 2, 3...n , p = 1, 2, 3, ..., ukuran populasi

2. Hitung total *fitness* yang diperoleh dari semua kromosom

$$F_{TOT} = \sum_{k=1}^{\text{ukuran populasi}} F_k \quad ; k = 1, 2, ..., \text{ ukuran populasi}$$

3. Hitung probabilitas seleksi p_k untuk setiap kromosom

$$p_k = \frac{F_k}{F_{TOT}} \quad ; k = 1, 2, ..., \text{ ukuran populasi}$$

4. Hitung probabilitas kumulatif q_k untuk setiap kromosom

$$q_k = \sum_{j=1}^{k} p_j \quad ; k \text{ dan } J = 1, 2, ..., \text{ ukuran populasi}$$

Setelah diperleh hasil penghitungan untuk tiap-tiap poin, dilakukan prosedur berikut :

Langkah 1 : Bangkitkan bilangan *random* r dengan *range* [0, 1]

Langkah 2 : Jika $r \leq q_1$, kemudian pilih kromosom pertama (k = 1), selain itu, pilih kromosom ke - k dimana $(q_{k-1} \leq r \leq q_k)$.

2.4.2.f Strategi Elit (*Elitist Strategy*)

Strategi ini digunakan untuk memilih kromosom terbaik pada setiap generasi untuk tetap bertahan pada generasi berikutnya, sehingga solusi yang dihasilkan pada setiap generasi memiliki nilai yang cenderung naik.

2.4.3 PENENTUAN PARAMETER

Yang merupakan parameter di sini adalah parameter kontrol Algoritma Genetik, yaitu : ukuran populasi (*pop_size*), peluang *crossover* (p_c), dan peluang mutasi (p_m). Nilai prameter ini ditentukan juga berdasarkan permasalahan yang akan dipecahkan. Ada beberapa rekomendasi yang digunakan, antara lain (Kusumadewi, 2003) :

- Untuk permasalahan yang memiliki kawasan solusi cukup besar, De Jong merekomendasikan untuk nilai parameter kontrol :

  (*pop_size*; p_c; p_m) = (50; 0.6; 0.001)

- Bila rata-rata *fitness* setiap generasi digunakan sebagai indikator, maka Grefenstette merekomendasikan :

  (*pop_size*; p_c; p_m) = (30; 0.95; 0.01)

- Bila *fitness* dari individu terbaik dipantau pada setiap generasi, maka usulannya adalah :

  (*pop_size*; p_c; p_m) = (80; 0.45; 0.01)

Ukuran populasi sebaiknya tidak lebih kecil dari 30 untuk sembarang jenis permasalahan.

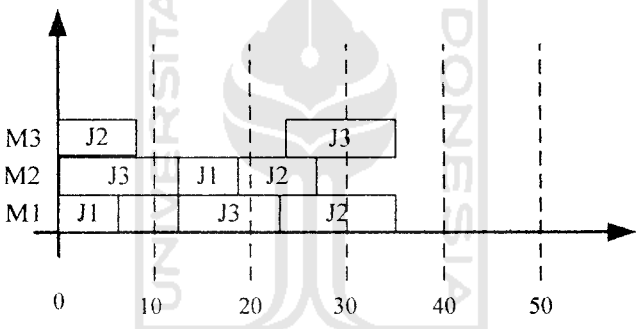
2.4.4 PENJADWALAN DINAMIS

Proses penjadwalan ulang dilakukan dengan memperhitungkan populasi akhir pada periode sebelumnya dengan mempertimbangkan bahwa dengan mengubah beberapa bagian dari kromosom yang mewakili material yang belum

diproses pada saat masuknya *job* baru, maka solusi optimal hasil pencarian awal masih tetap terjaga karena hanya sedikit bagian dari hasil optimal atau sub optimal yang berubah.

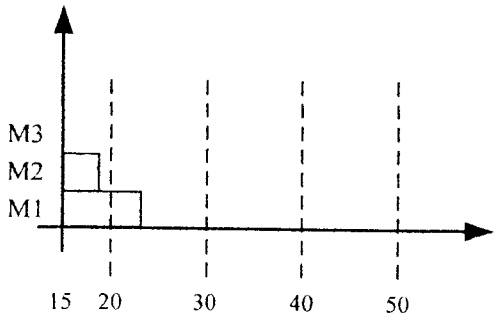
Alasan lainnya adalah sangat tidak efektif apabila dilakukan pembuatan jadwal ulang secara keseluruhan, karena pada saat masuknya *job* baru tersebut terdapat material yang sudah atau sedang di proses, sehingga akan mengakibatkan proses produksi akan terganggu secara keseluruhan.

Untuk lebih jelasnya mengenai *rescheduling* ini, dapat dilihat pada gambar di bawah ini :



Gambar 2.13 Kondisi jadwal awal

Dari kondisi pada jadwal awal, apabila terbdapat *job* masuk pada t = 15, maka waktu mulai *job* untuk setiap mesin adalah sebagai berikut :



Gambar 2.14 Kondisi awal setelah terjadi kejadian dinamis