JURNAL TEKNOLOGI TECHNOSCIENTIA ISSN: 1979-8415
Vol. 2 No. 2 Februari 2010

ADAPTASI ALGORITMA GENETIKA UNTUK MASALAH PENJADWALAN ULANG DENGAN TEKNIK SISIPAN PADA SISTEM MANUFAKTUR *JOB SHOP* DINAMIS NON DETERMINISTIK

Endang Widuri Asih[1]

[1]Jurusan Teknik Industri, Institut Sains & Teknologi AKPRIND Yogyakarta

Masuk: 8 Nopember 2009, revisi masuk: 29 Desember 2009, diterima: 26 Januari 2010

ABSTRACT

The scheduling of a non-deterministic dynamic job shop was a matter of organizing an amount of operations for certain machines with different sequence for different job, whereas the output of the job varies and was previously could not be determined. The purpose of the research was to minimize the mean tardiness of lateness of a job whether it was either an active job or non-delay, ahead of schedule or the rescheduled, and a prioritized job or the non priority. The research used genetic algorithm approach to determine the problems occurred in the scheduling or the rescheduling of a non-deterministic dynamic job shop. The function of fitness of the research functioned as the goal, whereas the best fitness output was the fitness with the biggest minimum mean tardiness. The best fitness for an ahead of schedule, either active or non-delay, was the ones with the exact fitness value of 0,0256410The same goes for early scheduling, either with or without priority, which had the exact output of 0,0256410. For a new job, either active or non-delay, had the exact fitness output of 0, 0148148. A different resulted for the rescheduling of a prioritized new job that had the best fitness output of 0, 0077295 for active job; and 0,005868 for a non-delay job. Rescheduling could be done by doing a job insertion in the beginning of the schedule. It was seen that in the beginning of the schedule, whether active or non-delay, the fitness was 0,0245610 and afterwards by adding job 6, job 7 and job 8 the output of the fitness differed accordingly with the case of the job.

Keyword: *Rescheduling, Non-deterministic, Genetic Algorithm*

INTISARI

Penjadwalan *job shop* dinamis non deterministik merupakan persoalan pengurutan sejumlah operasi yang diproses pada mesin-mesin tertentu dengan urutan pengerjaan berbeda untuk setiap *job* yang berbeda, dimana kedatangan *job* tersebut bervariasi dan tidak diketahui sebelumnya. Tujuan dari penelitian ini adalah untuk meminimalkan rata-rata keterlambatan *job* baik untuk jadwal aktif maupun *non delay*, penjadwalan awal maupun penjadwalan ulang, serta prioritas atau tidaknya sebuah *job*. Dalam penelitian ini di gunakan pendekatan algoritma genetika untuk masalah penjadwalan dan penjadwalan ulang *job shop* dinamis non deterministik.Pada peneltian ini fungsi *fitness* sebagai fungsi tujuan, dimana *fitness* terbaik adalah *fitness* terbesar yang berarti memiliki keterlambatan rata-rata minimal. *Fitness* terbaik untuk kasus penjadwalan awal, baik jadwal aktif maupun jadwal *non delay* adalah 0,0256410. Demikian juga untuk untuk penjadwalan awal tanpa atau dengan prioritas memiliki hasil yang sama yaitu 0,0256410. Untuk kasus job baru, baik aktif maupun *non delay*, memiliki hasil *fitness* terbaik sama yaitu 0,0148148. Hasil yang ini berbeda terjadi pada penjadwalan ulang job baru dengan prioritas yang memiliki *fitness* terbaik 0,0077295 untuk jadwal aktif dan 0.005868 untuk jadwal *non delay*. Penjadwalan ulang dapat dilakukan dengan melakukan penyisipan *job* pada jadwal awal. Terlihat bahwa *fitness* jadwal awal baik jadwal aktif maupun jadwal *non delay* nilai *fitness*-nya 0,0245610 dan setelah ada penambahan *job* 6, *job* 7 dan *job* 8 nilai *fitness*nya berubah sesuai dengan kasusnya.

Kata kunci *:* Penjadwalan Ulang*, Non Determinstik,* Algoritma Genetika

[1]endang.akprind@gmail.com

PENDAHULUAN

Salah satu aktivitas atau kegiatan industri memegang peranan penting adalah perencanaan dan pengendalian produksi di tingkat lantai produksi (*shop floor*), dan sub bagian dari aktivitas ini salah satunya adalah penjadwalan produksi. Penjadwalan produksi yang baik dapat meningkatkan produktivitas, memaksimalkan utilitas peralatan dan mesin produksi serta menurunkan waktu tinggal *part* dalam sistem produksi sehingga akan dapat menurunkan biaya persediaan. Kesemuanya dapat menghasilkan keunggulan kompetitif bagi perusahaan yang pada akhirnya berdampak positif pada profitabilitas (Kusrini, 2000).

Menurut Gaspersz(2002) mengnyatakan bahwa perusahaan yang meanut sistem *job shop* memiliki kelemahan, salah satunya dalam hal penjadwalan. Pada jadwal produksi yang telah dibuat dan dirancang sedemikian rupa oleh bagian *engineering*, tidak selamanya di taati. Hal ini senada dengan Viera *et.al* (2005), perubahan atau gangguan adalah suatu hal yang mungkin saja terjadi. Hal tersebut karena kondisi dilantai produksi yang sangat fleksibel, sifat kedatangan *job* yang variatif dan tidak diduga sebelumnya, serta *due date* yang variatif atau mungkin adanya tingkat kepentingan atau prioritas yang berbeda yang ditetapkan oleh perusahaan.

Berdasarkan kasus tersebut dalam penelitian ini akan dibahas penjadwalan ulang (*rescheduling*) yang dilakukan dalam lingkungan manufaktur *job shop* dinamis non deterministik dengan mengadaptasikan pada algoritma genetika. Dan Tujuan Penelitian yang ingin dicapai adalah menentukan jadwal produksi yang optimal dan sesuai dengan fungsi obyektifnya yaitu meminimasi rata-rata keterlambatan *job* (*mean tardiness*), serta menentukan penjadwalan ulang bila terdapat *job* baru dan perubahan prioritas job. Penjadwalan didefinisikan sebagai upaya untuk mengatur kegiatan atau job dengan tujuan untuk mencapai efisiensi dari penggunaan fasilitas, waktu dan biaya (Buffa, 1996). Penjadwalan juga dapat dipandang sebagai proses pengalokasian sumber yang tersedia untuk memilih tugas dalam jangka waktu tertentu (Baker, 1974). Fasilitas produksi pada suatu industri yang jumlahnya terbatas, sementara *job* yang akan dilakukan cukup banyak, sehingga keterbatasan fasilitas menuntut pengaturan jadwal *job* yang sesuai dengan sumber daya yang ada. Adanya pengaturan proses atau penjadwalan ini akan mengurangi mesin-mesin yang menganggur sehingga *job* dapat dikerjakan sesuai dengan jadwal yang ditentukan dan memberikan rasa puas terhadap pelanggan.

Masalah penjadwalan *job shop* (PJS) adalah masalah optimasi kombinatorial yang *NP-hard,* dimana sejumlah job akan diproses pada sejumlah mesin, dengan batasan: (1) urutan mesin yang harus dilalui oleh tiap *job* sudah ditentukan, dan (2) setiap mesin hanya dapat memproses paling banyak satu *job* pada suatu waktu tertentu. Pemrosesan sebuah *job* pada sebuah mesin tertentu disebut operasi, durasi tiap operasi adalah sebuah konstanta (Panggabean, 2003).

Berdasarkan waktu kedatangan *job*, penjadwalan *job shop* dapat dikelompokkan sebagai *static job* shop *scheduling* dan *dynamic job shop scheduling*. Pada *static job shop scheduling*, semua *job* diterima pada saat yang sama. Pada *dynamic job shop scheduling*, waktu kedatangan job bervariasi tetapi sudah diketahui sebelumnya (*deterministic*) atau waktu kedatangan *job* bervariasi dan tidak dapat diketahui sebelumnya (*non deterministic* / *stochastic*) (Lin *et al*, 1997).

Algoritma gentika adalah algoritma pencarian heuristic yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi kromosom antar individu antar organisme. Variasi kromosom ini mempengaruhi laju produksi dan tingkat kemampuan organisme ini untuk bertahan hidup (Kusumadewi dan Purnomo, 2005). Individu yang lebih kuat (*fit*) akan memiliki tingkat survival dan tingkat reproduksi yang lebih tinggi jika dibandingkan dengan individu yang kurang *fit*. Pada kurun waktu tertentu (sering dikenal dengan istilah generasi), populasi secara keseluruhan akan lebih banyak memuat organisme yang *fit*. Ada 3 operator yang umumnya diterapkan dalam algoritma genetika, yang sesuai pula operator pada

kromosom dalam teori evolusi, yaitu reproduksi, crossover dan mutasi.

Algoritma genetika ini berbeda dengan prosedur optimasi dan pencariansecara konvensional dalam beberapa hal diantaranya (Gen dan Cheng, 1997): yaitu: 1). Algoritma genetika bekerja dengan suatu pengkodean dari sekumpulan solusi, bukan solusi itu sendiri. 2).Algoritma genetika bekerja dari suatu populasi solusi, bukan suatu solusi tunggal. 3).Algoritma genetika menggunakan informasi yang diberikan (fungsi *fitness*), bukan informasi yang lainnya. 4).Algoritma genetika aturan perpindahan probabilitas, bukan aturan deterministik. Ada empat hal dasar yang perlu diperhatikan dalam kaitannya dengan penerpan algoritma genetika, yaitu: pemilihan representasi masalah ke bentuk *string*, operator genetik, fungsi *fitness,* parameter genetik. Representasi *string* bergantung pada masalah yang akan diselesaikan. Pada penjadwalan *job shop* akan dipergunakan *permutation encoding (Saputro dan Yento, 2004)*. Satu karakter mewakili suatu operasi, nilai karakter (*allele*) mewakili nomor *job* dan nomor urut operasi pada *job* tersebut. *Allele* akan dipetakan oleh fungsi yang memasangkan nomor *job* dan nomor urut operasi menjadi nomor mesin dan lama proses.

Operator genetik yang dipakai adalah operator reproduksi gabungan dari *elitism* dan *roulette wheel selection*, operator *Precedence Preservative Crossover* (PPX) (Bierwirth dan Mattfeld,1999) dan operator mutasi *remove and insert* (Manderick, 1991 dalam S*aputro dan Yento, 2004*). Metode *elitism* membuat sejumlah string terbaik tiap generasi akan otomatis diturunkan ke generasi berikutnya (Koza, 2001 dalam S*aputro dan Yento, 2004*). Metode *roulette wheel* ini untuk memilih string-string yang akan dilakukan proses rekombinasi. Pada PPX, string baru disusun secara acak dari *allele* string-string induk. Angka acak 1 atau 2 dipakai untuk memilih induk. Jika 1 diturunkan *allele* paling kiri dari induk pertama, jika 2 diturunkan *allele* paling kiri dari induk kedua. Selanjutnya *allele* yang terpilih tadi dihapus dari kedua induk. Proses dilakukan sampai karakter di kedua induk habis. Contoh 2 induk ABCDEF dan CABFDE, dan angka acak 1 2 1 1 2 2, akan menghasilkan string baru ACBDFE.

Permutasi Induk 1	A	B	C	D	E	F
Permutasi Induk 2	C	A	B	F	D	E
Induk terpilih (1/2):	1	2	1	1	2	2
Permutasi anak	A	C	B	D	F	E

Gambar 1. Ilustrasi PPX (*sumber*: Bierwith dan Mattfeld, 1999)

Pada mutasi, satu *locus* dipilih secara acak dan karakter diposisi tersebut di hapus. Satu *locus* baru dipilih, dan karakter yang telah dihapus tadi disisipkan. Gambar 2 menunjukkan proses mutasi pada *locus* ketiga dan ketujuh.
Pada dynamic scheduling dengan kedatangan *job* tidak dapat diperkirakan sebelumnya, minimalisasi makespan dirasakan kurang berarti (Lin *et al.,* 1997). Pada penelitian Saputro dan Yento (2004), rata-rata waktu penyelesaian sebuah *job*(*average flow time*) pada satu periode penjadwalan digunakan sebagai fungsi fitness. Jika menurut Lin *et al*, (1997), hal terpenting dalam penjadwalan *job shop*

A	**3-1**	**1-1**	**2-2**	**3-2**	**2-1**	**2-3**	**1-3**	**1-2**	**3-3**
Hasil	**3-1**	**1-1**	**3-2**	**2-1**	**2-3**	**1-3**	**2-2**	**1-2**	**3-3**

Gambar 2. Operator mutasi *remove and insert* (Bierwith dan Mattfeld, 1999)

adalah meminimasi *tardiness job* (keterlambatan *job*). Pendapat lain bahwa konsumen lebih mengutamakan ketepatan penyampaian produk daripada lamanya produk yang mengalir dalam sistem produksinya (Wahyuni, 2002). Penelitian ini bertujuan untuk meminimalkan rata-rata keterlambatan (*mean tardine*s) *job* yang masuk dalam penjadwaan produksi. Rumusan yang digunakan sebagai fungsi tujuan adalah sebagai berikut (Kusuma, 2004):

$T_{i,s} = max\{0, C_{i,s} - d_i\}\ i \varepsilon n$ ..... (1)

dengan :
$T_{i,s}$ = *Tardiness* job i dalam jadwal s
$C_{i,s}$ = Waktu penyelesaian (*completion time*) *job* i dalam jadwal s

Rumus keterlambatan (*tardiness*) pekerjaan di atas digunakan apabila diasumsikan bahwa batas waktu (due *date*) di ukur dari t=0. oleh karenanya keterlambatan rata-rata (*mean tardiness*) dapat didefinisikan sebagai:

$$\overline{T_s} = \frac{1}{n}\sum_{1=1}^{n} T_{i,s} \quad \text{.....} \qquad (2)$$

dengan:
$\overline{T_s}$ = keterlambatan rata-rata

Fungsi tujuannya adalah mencari minimal *mean tardiness*, maka secara matematis dapat diartikan mencari deviasi (penyimpangan) dari *due date*-nya. Dalam algoritma genetika funsi tujuannya adalah fungsi maksimasi sedangakan dalam penelitian ini tujuannya adalah minimasi, maka fungsi tersebut diubah menjadi fungsi invers, dengan demikian fungsi tujuan yang dipakai dalam pengolahan data adalah sebagai berikut:

$$Fitness = \frac{1}{\overline{T_s}} \quad \text{.....} \qquad (3)$$

Pengolahan dengan perangkat lunak atau program bantu yang di bangun ini menggunakan parameter genetika, baik untuk jadwal aktif maupun *non delay*, sebagai berikut:

Ukuran populasi	: 30 kromosom
Peluang *crossover*	: 80%
Peluang Mutasi	: 25%
Maksimum generasi	: 100 generasi
Jumlah *elitsm*	: 7 *fitness* terbaik
Titik konvergensi	: 7

Pemodelan penjadwalan yang dimaksud adalah proses penyusunan jadwal dari *string*. Permutasi operasi-operasi yang direpresentasikan oleh string akan di-*decode* untuk menghasilkan jadwal. Ada tiga karakteristik jadwal yang dapat dihasilkan, yaitu *feasible*, semi aktif, aktif dan *non delay* (Kusuma, 2002). Penelitian yang ini memakai jadwal *hybrid* yaitu gabungan antara jadwal *active* dan *non delay*. Notasi-notasi yang digunakan pada penelitian ini pada proses penjadwalan *job shop* dengan algoritma genetika ini adalah sebagai berikut (Saputro dan Yento, 2004):

M	: Jumlah mesin
N	: Jumlah *job*
m_i	: Jumlah operasi dari *job* ke-*i*, $1 \le i \le n$, $m_i \le m$
$\mu_i(k)$	:Urutan mesin yang harus dilalui oleh *job* ke-i operasi ke-k.
o_{ik}	: Operasi ke-*k* dari job ke-i, $1 \le k \le m_i$
o_{i1}	: Operasi ke-1 dari job ke-i.
p_{ik}	: Lama proses operasi ke-*k* dari job ke-i.
t_{ik}	: Waktu mulai (*starting time*) operasi ke-*k* dari job ke-i.
r_i	: Waktu tiba job ke-i.
δ	:Parameter yang menunjukan batas lamanya suatu mesin dapat menganggur, δ ε [o,1], δ = untuk jadwal *non delay*, δ = 1 untuk jadwal aktif.

Jadwal aktif dan *non delay* untuk kasus Penjadwalan tanpa prioritas, *Langkah 1: B*uat himpunan operasi yang mengawali pekerjaan: *A={ o_{i1} | 1≤I ≤ n }. Langkah 2:* Pilih O^1, operasi dengan waktu selesai tercepat, $t^1 + p^1 \le t_{ik} + p_{ik}$ untuk semua $o_{ik}\ \varepsilon\ A$. Jika lebih dari satu operasi, pilih operasi yang terletak paling kiri di string. Jika M^1 adalah mesin yang dipakai oleh o^1, buat himpunan B yang berisi semua operasi dari A yang diproses di M^1, B:={oik | A | μi (k) = M^1}. Pilih o^{11}, operasi dengan waktu mulai paling awal di B, $t^{11} < t_{ik}$ untuk semua oik ε B. Jika lebih dari satu operasi, pilih operasi yang terletak paling kiri di string. Hapus operasi di B menurut parameter δ, sehingga himpunan B sekarang B:={oik ε B | $t_{ik} \le t^{11} + \delta\ ((t1+p1)-t^{11})$}. Pilih operasi di B yang terletak paling kiri di string dan hapus dari A. Operasi yang dipilih tersebut adalah o*ik. *Langkah 3:* Masukkan ope-

rasi o^*_{ik} pada jadwal, dan hitung waktu mulai: $t^*_{ik}=max(t^*_{i,k1}+p^*_{i,k-1},t_{hl}+p_{hl})$, t^*_{ik} waktu mulai operasi o^*_{ik}, $t^*_{ik-1}+p^*_{ik-1}$ waktu selesai operasi ke-*(k-1),* yaitu operasi sebelumnya dari pada *job* ke-*i* dan o_{hl} = operasi ke-*l* dari *job* ke-*h* yang mendahului o^*_{ik} pada mesin yang sama. *Langkah 4:*Jika terdapat suksesor dari o^*_{ik}, yaitu $o^*_{i,k+1}$, tambahkan ke *A*. *Langkah 5:* Ulangi langkah 2 sampai isi *A* habis.

Jadwal aktif dan *non delay* kasus penjadwalan dengan prioritas.Prosedur yang digunakan untuk kasus ini pada dasarnya sama pada kasus penjadwalan awal, hanya ada sedikit perubahan dalam pemilihan o^1 dan o^{11}, pada Langkah 2 poin i dan iii yaitu menjadi: Pilih O^1, operasi dengan waktu selesai tercepat, $t^1 + p^1 \le t_{ik} + p_{ik}$ untuk semua o_{ik} *ε A* yang memiliki prioritas utama. Jika lebih dari satu operasi yang memiliki prioritas utama, pilih operasi yang terletak paling kiri di string. Jika tidak ada operasi dengan priorits utama, Pilih O^1, operasi dengan waktu selesai tercepat, $t^1 + p^1 \le t_{ik} + p_{ik}$ untuk semua o_{ik} *ε A* dan jika lebih dari satu operasi, pilih operasi yang terletak paling kiri di string.

Pilih o^{11}, operasi dengan waktu mulai yang paling awal di B, $t^{11}<t_{ik}$ untuk semua oik ε B yang memiliki prioritas utama. Jika lebih dari satu operasi yang memiliki prioritas utama, pilih operasi yang terletak paling kiri di *string*. Jika tidak ada operasi dengan prioritas utama, Pilih o^{11} operasi dengan waktu mulai paling awal di B, $t^{11}<t_{ik}$ untuk semua oik ε B. Jika lebih dari satu operasi, pilih operasi yang terletak paling kiri di *string*.

Jadwal aktif dan *non delay* untuk kasus Penjadwalan ulang Jadwal aktif dan *non delay* untuk kasus Penjadwalan ulang karena *job* baru dengan atau tanpa prioritas pada dasarnya hanya untuk menentukan waktu tiba *job* yang baru, initial *setup time,*dan waktu mulai untuk mesin yang prtama kali dapat dipakai. Bila ada tambahan *job* baru pada saat t_1 : Simpan jadwal dari operasi o_{ik} yang dimulai sebelum t_1. Jadwal ini tidak perlu diubah lagi saat penjadwalan ulang.

Hapus semua operasi o_{ik} yang dimulai sebelum t_1 ($t_{ik}<t_1$) dari kumpulan *job* lama. Suatu *job* telah selesai dikerjakan seluruhnya bila m_i = 0. Bila ada *job* ke-i yang belum selesai, hitung waktu tiba yang baru :

$$r_i = \max_{1\le k<m_i} \left\{ t_{ik} + p_{ik} \mid t_{ik} < t_1 \right\} \quad \text{.....(4)}$$

Jika ada operasi o_{ik} yang masih diproses pada t_1 dan belum selesai ($t_{ik}<t_1< t_{ik}+p_{ik}$), mesin tidak dapat dipakai sampai operasi tersebut selesai. Perlu dihitung *initial set-up time* s_j, yaitu waktu dimana mesin ke-j baru dapat mulai digunakan lagi.

$$s_j = \max\left(\max_{1\le i\le n} \left\{ t_{ik} + p_{ik} \mid t_{ik} < t_1 \right\}, t_1 \right) \text{...(5)}$$

Waktu setup ini akan dipergunakan saat penjadwalan ulang dan dipakai saat menghitung waktu mulai (*starting time*) dari operasi yang memakai mesin tersebut pertama kali. Waktu mulai untuk mesin ke-j saat dipergunakan pertama kali oleh operasi o_{ik} dapat dihitung: t_{ik} = max($t_{i,k-1}$ + $pi_{,k-1,}s_j$). a).Bentuk string baru dari sisa operasi yang belum dikerjakan pada jadwal lama, dan dari operasi-operasi dari *job-job* yang tiba pada saat t_1. b).Bentuk populasi awal yang baru secara acak sebanyak *Nmax string*, susun jadwal tiap string dengan prosedur *hybrid* dan hitung nilai *fitness* tiap string.

Spesifikasi *job* awal pada penelitian ini yang akan dijadwalkan sebagai berikut: *Ready time* adalah saat dimana *job* siap untuk dikerjakan pada mesin produksi, sedangakan *due date* adalah batas waktu bahwa *job* harus selesai dikerjakan.

Dalam penelitian ini, *due date job* dibagi dua jenis, yaitu *due date* yang langsung ke konsumen dan yang kedua adalah *due date* karena adanya proses lain yang juga dikenakan pada *job* terse-but pada departemen atau bagian lain diluar bagian fabrikasi. Pada dasarnya semua proses pada setiap kasus pen-jadwalan adalah sama, letak perbedaan-nya adalah pada prosedur *hybrid* untuk perhitungan *fitness*-nya. Berikut ini hasil terbaik dari setiap generasi untuk setiap kasus penjadwalan. Berikut adalah hasil akhir kasus penjadwalan ini berdasarkan *fitness* terbaiknya.

Tabel 1. Spesifikasi Job Awal

No. *Job*	Nama *Job*	*Routing Job* (No. Mesin) dan Lama Operasi (Menit)				
		Operasi 1	Operasi 2	Operasi 3	Operasi 4	Operasi 5
Job 1	*Disc Column*	2, 240	9, 690	5, 180	5, 60	
Job 2	*Mach. Flange*	2, 240	10., 60	2, 240	11, 60	2, 120
Job 3	*Fab. Inter Column*	2, 240	5, 180	12, 120	10, 180	1, 240
Job 4	*Column Modification*	3, 720	5, 480	11, 480	1, 480	
Job 5	*Mach. Adapter*	1, 420	6, 360			

Tabel 2. *Ready Time* dan *Due Date Job* Awal

No. *Job*	Nama *Job*	*Ready Time*	*Due Date*
Job 1	*Disc Column*	05-12-2008 08:15	08-12-2008 12:00
Job 2	*Mach. Flange*	05-12-2008 08:15	06-12-2008 16:00
Job 3	*Fab. Inter Column*	05-12-2008 08:15	07-12-2008 10:00
Job 4	*Column Modification*	05-12-2008 08:15	10-12-2008 12:00
Job 5	*Mach. Adapter*	05-12-2008 08:15	06-12-2008 16:00

Tabel 3. Spesifikasi *Job* Baru

No. *Job*	Nama *Job*	*Routing Job* (No. Mesin) dan Lama Opersi (Menit)		
		Operasi 1	Operasi 2	Operasi 3
Job 6	*Mach. Column*	2, 240	9, 240	5, 60
Job 7	*Fab. Pump Shaft*	3, 66	4, 240	9, 180
Job 8	*Machinig Column*	2, 330	7, 390	

Tabel 4. *Ready Time* dan *Due Date Job* Baru

No. *Job*	Nama *Job*	*Ready Time*	*Due Date*
Job 6	*Mach. Column*	07-12-2008 10:00	08-12-2008 12:00
Job 7	*Fab. Pump Shaft*	06-12-2008 08:00	08-12-2008 16:00
Job 8	*Machinig Column*	08-12-2008 11:00	10-12-2008 10:00

Tabel 5. Jadwal terbaik berdasrkan fitness terbaik tiap kasus penjadwalan

Kasus	*Fitness* terbaik
Jadwal aktif kasus penjadwalan awal	**0,0256410**
Jadwal aktif kasus penjadwalan awal dengan prioritas	**0,0256410**
Jadwal *nondelay* kasus penjadwalan awal	**0,0256410**
Jadwal *nondelay* kasus penjadwalan awal dengan prioritas	**0,0256410**
jadwal aktif kasus penjadwalan ulang *job* baru	**0,0148148**
jadwal aktif kasus penjadwalan ulang *job* baru dengan prioritas	**0,0077295**
jadwal *non delay* kasus penjadwalan ulang *job* baru	**0,0148148**
jadwal *non delay* kasuss penjadwalan ulang *job* baru dengan prioritas	**0,0058608**

Setelah diketahui *fitness* terbaik dari setiap kasus penjadwalan maka da-pat kita ketahui jadwal mana yang terbaik berdasarkan *fitness* dalam hal meminimumkan rata-rata keterlambatan *job*. Tabel 6-9 adalah hasil terbaik dari setiap kasus panjadwalan berdasarkan *fitness* terbaik yang diolah dengan perangkat lunak.

PEMBAHASAN

Perbandingan Jadwal Aktif dan *Non Delay, b*erdasarkan hasil yang telah didapat dari pengolahan data maka dapat diambil suatu analisa bahwa *fitness* terbaik dari jadwal aktif dan *non delay* memiliki hasil yang sama untuk beberapa kasus penjadwalan, artinya antara jadwal dan *non delay* memiliki jadwal terbaik yang sama. Pada teori, seharusnya jadwal *non delay* akan lebih ketat dibandingkan dengan jadwal aktif karena tidak ada waktu menunggu pada mesin. Pada penelitian ini, hasil sama didapatkan karena variasi dan jumlah *job* yang sedikit serta jumlah mesin yang banyak menyebabkan tidak ada beban mesin yang berlebihan. Tetapi untuk *fitness* terburuk dan *fitness* rata-rata berbeda, walau ini tidak ada pengaruhnya terhadap hasil akhir karena hasil akhir yang dipilih adalah *fitness* terbaik. Jadwal aktif dan *non delay* memiliki *fitness* terbaik berbeda pada kasus penjadwalan ulang *job* baru dengan prioritas.

Tabel 6. Perbandingan jadwal aktif dan *non delay*

No.	Kasus Penjadwalan	Nilai *fitness*	
		Aktif	*Non Delay*
1	Penjadwalan awal	0,0256410	0,0256410
2	Penjadwlan awal dengan prioritas	0,0256410	0,0256410
3	Penjadwalan ulang *job* baru	0,0148148	0,0148148
4	Penjadwalan ulang *job* baru dengan priorits	0,0077295	0,0058608

Perbandingan penjadwalan ulang karena adanya *job* berpengaruh terhadap jadwal akhir, ini dapat terlihat pada jadwal yang terjadi penyisipan *job* baru pada jadwal awal. Akibat lain adalah, penyelesaian jadwal akhir pun tentu berbeda yang ditunjukkan dengan nilai *fitness*-nya. *Job* baru yaitu 6,7, dan 8 dapat disisipkan pada jadwal lama

Tabel 7. Perbandingan Penjadwalan Awal dan Penjadwalan Ulang

No.	Kasus	Nilai *fitness*	
		Jadwal Awal	Jadwal Ulang
1	Aktif awal	0,0256410	0,0148148
2	Aktif awal	0,0256410	0,0077295
3	*Non delay* awal	0,0256410	0,0148148
4	*Non delay*	0,0256410	0,0058608

Berdasar Tabel 7. tampak bahwa penjadwalan awal memiliki pengaruh terhadap jadwal awal yang ditunjukkan dengan nilai *fitness* yang berbeda. Perbedaan atau perubahan nilai *fitness* ini disebabkan oleh adanya *job* sisipan yang mengakibatkan jadwal awal juga berubah. Terlihat bahwa nilai *fitness* semakin kecil yang berarti bahwa rata-rata keterlambatan semakin besar. Pengaruh prioritas *job* berpengaruh besar terhadap jadwal yang terbentuk, karena pada *job* yang memiliki prioritas utama akan dikerjakan terlebih dahulu. Kemudian *job* yang

tidak memiliki prioritas baru dikerjakan setelah *job* dengan prioritas selesai dikerjakan. Aturan ini berlaku tapi tetap mengacu pada prosedur *hybrid.* Ini ditunjukkan dengan nilai *fitness* yang lebih kecil dari pada jadwal tanpa prioritas, artinya adanya keterlambtan *job* yang lebih besar. Rata-rata keterlambatan yang lebih besar ini disebabkan *due date* tidak terlalu diperhitungkan. Yang terpenting adalah bahwa *job* dengan prioritas utama terjadwalkan terlebih dahulu.

Tabel 8. Pengaruh Prioritas *Job* terhadap nilai *fitness*

No.	Kasus	Nilai *fitness*	
		Tanpa prioritas	Dengan prioritas
1	Aktif awal	0,0256410	0,0256410
2	*Non delay* awal	0,0256410	0,0256410
3	Aktif *job* baru	0,0148148	0,0077295
4	*Non delay job* baru	0,0148148	0,0058608

Pada jadwal awal baik aktif maupun *non delay*, aturan prioritas tidak berpengaruh pada hasil *fitness*. Sedangkan pada *job* baru, baik aktif maupun *non delay,* mempunyai pengaruh yang cukup signifikan dengan ditunjukan perubahan nilai *fitness* yang semakin kecil. Ini berarti rata-rata keterlambatan semakin besar.

Parameter genetika seperti jumlah populasi, peluang *crossover*(P_c,), peluang mutasi(P_m), *maxgen*, jumlah e*litsm*, dan titik konvergensi (*convergency point)* mempunyai pengaruh yang signifikan. Parameter genetika paling signifikan adalah e*litism* dimana mampu mempertahankan *fitness* terbaik dari setiap generasi. Kecenderungan bahwa tiap generasi dari semua kasus penjadwalan tidak mengalami perubahan karena memang variasi jadwal yang tidak terlalu tinggi. Artinya walau *fitness* tiap kromosom selalu berubah tiap generasi namun hasil terbaik tetaplah sama. Dengan parameter genetika yang berbeda akan dimungkinkan menghasilkan *fitness* yang berbeda.

KESIMPULAN

Dari hasil analisa dapat disimpulkan beberapa hal yaitu jadwal produksi yang optimal pada kasus penjadwalan didasarkan pada *fitness* terbaik yang dihasilkan. *Fitness* terbaik untuk kasus pejadwalan awal ini, baik jadwal aktif maupun jadwal *non delay*, memiliki nilai *fitness* yang sama yaitu 0,0256410 atau rata-rata keterlambatan 1/0,0256410 = 39 menit. Demikian juga untuk untuk penjadwalan awal tanpa atau dengan prioritas memiliki hasil sama yaitu 0,0256410. Untuk kasus *job* baru, baik aktif maupun *non delay*, memiliki hasil *fitness* terbaik yaitu 0,0148148 atau rata-rata keterlambatan 1/0,0148148= 67,5 menit. Dari hasil yang berbeda terjadi pada penjadwalan ulang *job* baru dengan prioritas yang memiliki *fitness* terbaik 0,0077295 untuk jadwal aktif dan 0,005868 untuk jadwal *non delay*.

Penjadwalan ulang dapat dilakukan dengan melakukan penyisipan *job* pada jadwal awal. Walaupun *job* baru ini akan berbepangruh pada jadwal awal yang telah dibuat sebelumnya akan tetapi respon terhadap perubahan lantai produksi akan segera dapat dengan cepat dilakukan. Terlihat bahwa *fitness* jadwal awal baik jadwal aktif maupun jadwal *non delay* nilai *fitness*-nya 0,0245610 dan setelah ada penambahan *job* 6, job 7 dan *job* 8 nilai *fitness*-nya berubah sesuai dengan kasusnya.

Hasil terbaik pada setiap kasus penjadwalan adalah dengan *fitness* terbaiknya. Terlihat bahwa ada perbedaan untuk beberapa kasus walaupun dalam penelitian ini hasilnya cenderung sama. Sedangkan jadwal yang dapat kita pilih untuk keputusan terbaik adalah apakah kita akan memberikan prioritas pada sebuah *job* apa tidak, atau juga memilih

jadwal aktif atau *non delay*. Akan tetapi semua tetap berdasarkan nilai *fitness* terbaik.

DAFTAR PUSTAKA

Baker, K.R, 1974, *Introduction to Sequencing and Scheduiling*, John Wiley & Sons, New York.

Bierwirth, C., D.C. Mattfield, 1999. Production Scheduling and Rescheduling with Genetic Algorithm, *Evolutionary Computation*, 7 (1), 1999, 1-17.

Buffa, E.S., 1996, *Manajemen Operasi dan Produksi Modern*, Binarupa Aksara, Jakarta.

Gasperz, V., 2002, *Production Planning and Inventory Control : Berdasarkan Pendekatan Sistem Terintegrasi MRP dan JIT Menuju Manufaktur 21*, PT. Gramedia Pustaka Utama, Jakarta.

Gen, M., dan Cheng, R., 1997, *Genetic Algorithms & Engineering Design*, A Wiley Intersciense Publication, John Wiley & Sons, Inc.

Kusrini, E., 2000, Pengembangan Model Penjadwalan dan Penjadwalan Ulang Job Shop Yang Dinamis dengan Pendekatan Time Window, *Jurnal Teknologi Industri*, Vol. 5, No, 3, 213-240.

Kusuma, H., 2002, *Manajemen Produksi: Perencanaan dan Pengendalian Produksi*, Andi Offset, Yogyakarta.

Kusumadewi dan Purnomo, H., 2005, *Teknik-Teknik Optimasi Heuristik*, Penerbit Graha Ilmu, Yogyakarta.

Lin, S., Goodman, E., Punch, W., 1997. A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems, *Proceedings of the Seventh International Conference on Genetic Algorithms*, 481 – 489, Morgan Kaufmann.

Panggabean, 2003, Adaptasi Simulated Annealing ke dalam Prosedur Shifting Bottleneck Untuk Masalah Job Shop, *Jurnal Integral*, Vol. 8 No. 1, April 2003, 53-62.

Saputro, N., dan Yento, 2004, Pemakaian Algoritma Genetik untuk Penjadwalan Job Shop Dinamis Non Determenistik, *Jurnal Teknik Industri* vol. 6, no, 1, 61-70.

Vieira,et.al, 2005, *Reschduling Manufacturing Systems: A Framework Of Strategies, Policies, And Methods,* Departement Of Mechanical Engineering, University Of Maryland.

Wahyuni, D., 2002, Pengembangan Model Penjadwalan Menggunakan Teknik Sisipan (Insertion Technique), *Digittal library*, USU. http://www.library.usu.ac.id/modules.php