

PENERAPAN ALGORITMA GENETIKA UNTUK MASALAH PENJADWALAN *JOB SHOP* PADA LINGKUNGAN INDUSTRI PAKAIAN

Fachrudin Afandi, Mahendrawathi ER, S.T, M.Sc, Ph.D, Faizal Mahananto, S.Kom

Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Kampus Keputih, Sukolilo, Surabaya 60111, Indonesia

E-mail: fandi_717@yahoo.com, mahendra_w@its-sby.edu, faizal@is.its.ac.id

Abstrak—Pada industri pakaian khususnya yang proses produksinya bersifat berbaur dan multi produk sering mengalami kesulitan pada penjadwalan job shop. Oleh karena itu, perlu diadakan penelitian untuk penjadwalan job shop yang efektif terutama yang proses produksinya berbaur dan multi produk. Pada jurnal ini akan diajukan metode untuk penjadwalan job shop yang berbaur dan multi produk dengan tujuan meminimalkan total pinalti E/T (*Earliness/Tardiness*) dengan menentukan start pada masing-masing job shop dan bagaimana cara menugaskan operasi-operasi ke operator job shop tersebut. Sebuah algoritma genetika digunakan untuk menyelesaikan masalah penjadwalan job shop yang berbaur dan multi produk dimana pada akhirnya diperoleh penjadwalan job shop yang efektif.

Kata kunci— Algoritma Genetika, Penjadwalan Job Shop, Industri Pakaian

I. LATAR BELAKANG

Pada era globalisasi seperti sekarang, perusahaan diberbagai industri dihadapkan pada kompetisi pasar global yang terus meningkat dan fluktuasi permintaan yang tidak dapat diramalkan. Industri pakaian adalah salah satu yang terkena dampak dari tekanan untuk dapat menghasilkan berbagai macam produk yang sesuai dengan selera konsumen, dengan waktu yang singkat dan biaya yang rendah.

Industri pakaian pada umumnya beroperasi dengan sistem job shop. Penjadwalan job shop untuk industri pakaian adalah suatu penjadwalan yang menggunakan banyak mesin yang fleksibel disertai banyak operasi yang menyertainya.

Berbagai macam penelitian telah dilakukan terhadap masalah penjadwalan job shop pada industri pakaian. Dari penggunaan metoda pemrograman integer yang menggunakan relasi Langrangian sampai algoritma genetika untuk memecahkan permasalahan dalam menjaga keseimbangan lini perakitan pada industri pakaian. Model matematika umum untuk masalah penjadwalan job shop terutama yang produknya berbaur dan multi produk masih jarang diselidiki.

Guo et al. (2006) mengembangkan model matematika untuk *Job Shop Scheduling Problem* (JSSP) yang produknya bersifat berbaur dan multi produk. Dengan menerapkan metode *Genetic Algorithm* (GA), solusi masalah penjadwalan job shop yang berbaur dan multi produk menjadi lebih efektif yaitu dengan meminimalkan total pinalti E/T (*Earliness/Tardiness*) pada job shop dengan menentukan masing-masing pesanan sesuai waktu awal produksi dan tugas operasi pada job shop.

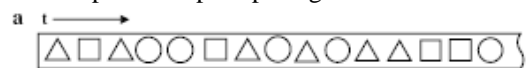
II. TINJAUAN PUSTAKA

A. Penjadwalan Job Shop

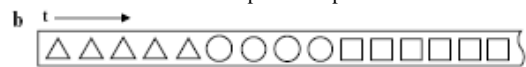
Penjadwalan merupakan proses yang menentukan bagaimana sumberdaya dialokasikan ke dalam berbagai operasi yang mungkin. Penjadwalan dapat juga diartikan adalah suatu proses pengaturan sumber daya untuk menyelesaikan tugas-tugas dengan melibatkan pekerjaan, sumber daya, dan waktu. Pekerjaan diproses pada setiap sumber daya dengan urutan tertentu selama waktu tertentu. Tujuan dari masalah penjadwalan antara lain: meminimumkan waktu penyelesaian semua tugas (*makespan*), meminimumkan keterlambatan pengerjaan, meminimumkan waktu tunggu pada mesin, meminimumkan biaya, dan lain-lain.

Masalah penjadwalan *job shop* merupakan salah satu masalah penjadwalan yang memiliki kendala urutan pemrosesan operasi. Masalah penjadwalan *job shop* adalah penjadwalan yang melibatkan suatu tugas pada seperangkat kerja pada stasiun-kerja (mesin) secara sekuensial, saat mengoptimalkan satu atau lebih sasaran tanpa melanggar batasan yang diterapkan pada *job shop* (Guo et al, 2006).

Didalam suatu penjadwalan produk berbaur, dua atau lebih pesanan produksi diproduksi dimanapun urutan berbaur seperti pada gambar 2.1. Sedangkan dua atau lebih produk diproses secara terpisah didalam batch pada penjadwalan masalah multi produk seperti pada gambar 2.2.



Gambar 2.1 Sistem perakitan produk berbaur



Gambar 2.2 Sistem perakitan multi produk

B. Algoritma Genetika

Algoritma genetika adalah algoritma heuristik adaptif yang memiliki dasar pemikiran atau gagasan evolusioner untuk proses seleksi alami dan genetika. Konsep dasar dari algoritma genetika dirancang untuk menirukan proses di dalam sistem alami yang penting bagi evolusi makhluk hidup untuk dapat terus bertahan hidup, yang secara rinci teori ini dicetuskan oleh Charles Darwin yaitu “Survival of the Fittest”. Dengan menirukan teori dari Charles Darwin, algoritma genetika dapat digunakan untuk mencari solusi dari segala macam

permasalahan dalam ilmu pengetahuan seperti dalam mencari solusi penjadwalan *job shop*.

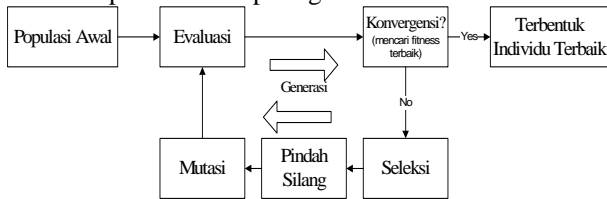
Pelopop pertama penggunaan metode algoritma genetika adalah John Holland pada tahun 60-an. Algoritma genetika menggunakan analogi seleksi alam yang bekerja dari suatu populasi yang terdiri dari berbagai individu (gen), yang masing-masing individu mempresentasikan suatu solusi yang mungkin muncul dari persoalan yang dihadapi. Dalam hal ini, individu yang terpilih dilambangkan dengan sebuah nilai *fitness* yang digunakan untuk mencari solusi terbaik dari persoalan yang ada.

Kemampuan individu yang tinggi memiliki kesempatan untuk dapat melakukan reproduksi melalui kawin silang (*crossover*) dengan individu yang lain pada populasi yang sama. Individu baru yang terbentuk atau dihasilkan membawa sifat-sifat dari induknya. Sedangkan individu didalam populasi yang telah melalui proses seleksi namun tidak terpilih akan mati dengan sendirinya. Beberapa generasi baru yang terbentuk dan memiliki ketahanan hidup yang kuat akan bermunculan didalam populasi tersebut, yang kemudian dari proses seleksi, kawin silang, dan mutasi yang berkelanjutan menghasilkan generasi dengan kemampuan bertahan hidup yang bagus.

Pada umumnya suatu penerapan algoritma genetika secara generasional sederhana terdiri dari 3 bagian, yaitu:

- 1) Memilih populasi awal (inisial populasi)
- 2) Evaluasi nilai *fitness* dari setiap individu didalam populasi
- 3) Ulangi sampai proses berhenti (nilai *fitness* terbaik terpenuhi)
 - a) Pilih individu terbaik berdasar ranking untuk reproduksi
 - b) Bentuk generasi baru melalui pindah silang dan mutasi untuk menghasilkan keturunan baru (*child*)
 - c) Evaluasi nilai *fitness* keturunan yang dihasilkan
 - d) Setelah diatur terlebih dahulu, gantikan individu dengan nilai *fitness* terburuk dengan keturunan yang dihasilkan

Jika diilustrasikan, maka proses algoritma genetika yang dilakukan dapat terlihat seperti gambar 2.3.



Gambar 2.3 Siklus Algoritma Genetika

III. RANCANGAN GA PADA JSSP

A. Asumsi-asumsi yang Digunakan

Pada jurnal ini, model matematika penjadwalan *job shop* yang diusulkan akan menggunakan asumsi-asumsi sebagai berikut:

- 1) Ketika sekali suatu operasi mulai dijalankan, maka tidak dapat disela.
- 2) Tidak ada kasus kekurangan material, gangguan mesin dan ketidakhadiran operator mesin *job shop* dalam kerjanya.

3) *Job shop* digunakan untuk memodelkan adalah dalam keadaan inisialisasi awal *job shop* kosong, dengan kata lain tidak ada kerja yang menumpuk sebelumnya (*Work in Process / WIP*) pada setiap stasiun-kerja.

B. Batasan-batasan dalam Pemodelan

Pada produksi pakaian yang nyata mempunyai beberapa karakteristik khusus dan harus sesuai dengan beberapa batasan yang ditetapkan sebelumnya. Pada bagian ini akan diuraikan karakteristik secara matematika dan batasan-batasan yang digunakan sebagai berikut:

1. Batasan waktu kedatangan, menyatakan bahwa pemenuhan order P_i tidak dapat dimulai SP_i sampai order/pesanan tiba A_i , yaitu

$$A_i \leq SP_i \quad (3.1)$$

2. Batasan alokasi. Operasi O_{il} hanya bisa dijalankan pada mesin yang mampu menanganinya, yaitu

$$\sum_{kj, M_{kj} \in SM_{il}} X_{ilkj} = 0 \quad (3.2)$$

Setiap mesin M_{kj} harus memproses setidaknya satu operasi O_{il} , yaitu

$$\sum_{il} X_{ilkj} \geq 1 \quad (3.3)$$

Setiap operasi O_{il} harus diproses, yaitu

$$\sum_{kj} X_{ilkj} \geq 1 \quad (3.4)$$

3. Batasan operasi yang harus didahulukan, menyatakan bahwa suatu operasi tidak dapat dimulai sebelum operasi yang terdahulu telah diselesaikan C_{il} dan diangkut sesuai dengan mesin kerjanya ET_{il} , yaitu

$$C_{il} + ET_{il} + 1 \leq S_{i'l'} \quad (3.5)$$

4. Kebutuhan waktu proses. Operasi O_{il} harus dijalankan dengan waktu mulai S_{il} , waktu *setup* STP_{il} dan waktu proses T_{il} , yaitu

$$C_{il} = S_{il} + STP_{il} + T_{il} - 1 \quad (3.6)$$

C. Fungsi Obyektif

Berdasarkan permintaan produksi yang berbeda-beda, sasaran yang berbeda dapat dibangun atas dasar uraian pada subbab sebelumnya. Pada industri pakaian, memenuhi permintaan tanggal pengiriman, atau tanggal jatuh tempo, adalah sasaran yang paling diinginkan oleh pihak manajemen.

Pada pembelajaran ini, fungsi obyektif dapat diilustrasikan secara matematika sebagai berikut:

$$\min_{\{SP_i\}, \{X_{ilkj}\}} Z, \text{ dengan } Z = \sum_{i=1}^p (\alpha_i \cdot TD_i \cdot \lambda_i + \beta_i \cdot EL_i \cdot (1 - \lambda_i)) \quad (3.8)$$

Pada masing-masing order P_i , tanggal jatuh tempo D_i ditentukan oleh pihak konsumen. Jika order P_i diselesaikan setelah tanggal jatuh tempo D_i , diperlukan pinalti α_i untuk setiap kali unit yang mengalami penundaan. Jika order diselesaikan sebelum tanggal jatuh tempo, tidak akan diterima

oleh konsumen sampai batas waktu itu. Dalam hal ini, gudang pabrik akan dipaksa untuk menyimpan order tersebut sampai tanggal jatuh tempo dan dikeluarkan pinalti β_i untuk setiap unit yang disimpan. Sasaran dari model ini diterapkan dengan pemilihan produksi sesuai waktu awal SP_i untuk setiap order dan membangkitkan penugasan operasi optimal, dimana setara dengan meminimumkan *makespan* jika waktu yang tersedia untuk pemrosesan order cukup singkat (misal: interval waktu antara tanggal sekarang dan tanggal jatuh tempo tidak cukup lama).

D. Inisialisasi Populasi

Pada proses inisialisasi populasi dapat dideskripsikan sebagai berikut:

Langkah 1. Inisialisasi parameter: indeks $i=1$, ukuran populasi u , populasi $PPN = \{\phi\}$ dan kuantitas QTY operasi yang mana setiap mesin dapat memproses.

Langkah 2. Secara acak membangkitkan suatu kromosom string integer CHR_i , $PPN = PPN \cup CHR_i$. (3.9)

Langkah 3. Set $i=i+1$. Jika $i>u$, STOP, kemudian kembali ke langkah 2.

Penjelasan secara detail pada proses membangkitkan kromosom secara acak adalah sebagai berikut:

Langkah 1. Set indeks $j=1$. Untuk setiap mesin, biarkan $PTY=1$, dimana PTY menggambarkan probabilitas suatu mesin dipilih untuk memproses operasi yang relevan.

Langkah 2. Membangkitkan acak integer k antara 1 – jumlah mesin yang memproses operasi j . Set nilai probabilitas $k=1$ jika melebihi dari nilai yang lain.

Langkah 3. Secara acak pilih k -mesin yang dapat memproses operasi ini. Mesin dengan PTY lebih besar akan memiliki probabilitas terbesar untuk terpilih. Jika $PTY=0$, mesin tidak dapat dipilih.

Langkah 4. Menugaskan operasi kepada mesin yang dipilih dari j ke k . Pada waktu bersamaan, untuk setiap mesin yang

dipilih, set $PTY = PTY - \frac{1}{QTY}$. (3.10)

Langkah 5. Set $j=j+1$. Jika $j>n$, lanjut ke langkah 6. jika tidak kembali ke langkah 2.

Langkah 6. Jika semua mesin sudah ditugaskan, STOP, jika tidak kembali ke langkah 1.

E. Fitness dan Seleksi

Fungsi *fitness* dideskripsikan sebagai tingkat kebugaran dari tiap kromosom untuk menentukan yang akan direproduksi dan bertahan ke dalam generasi berikutnya. Nilai *fitness* pada fungsi *fitness* digambarkan sebagai fungsi obyektif fungsi Z yaitu:

$$fitness = \frac{1}{Z+1} = \frac{1}{\sum_{i=1}^p (\alpha_i.TD_i.\lambda_i + \beta_i.EL_i.(1-\lambda_i)) + 1} \quad (3.11)$$

Proses seleksi turnamen (Goldberg, Korb, & Deb, 1989) pada algoritma genetika, berdasar pada aturan *survival of the fittest*, yaitu proses dimana kromosom dipilih untuk generasi

berikutnya dalam kaitan dengan *fitness*. Skema dan prosedur yang digunakan adalah sebagai berikut:

Langkah 1. Set ukuran turnamen $k \geq 2$.

Langkah 2. Membangkitkan suatu permutasi acak didalam populasi sekarang.

Langkah 3. Bandingkan nilai *fitness* pada kromosom k pertama pada daftar permutasi, dan salin yang terbaik ke dalam generasi berikutnya. Buang string pembandingnya.

Langkah 4. jika permutasi habis terpakai, bangkitkan permutasi lain.

Langkah 5. Ulangi langkah 3 dan 4 sampai tidak diperlukan seleksi lagi untuk generasi berikutnya.

Skema tersebut dapat mengendalikan keanekaragaman populasi dan tekanan seleksi dengan menyesuaikan ukuran turnamen k . Semakin besar nilai k akan meningkatkan tekanan seleksi saat mengurangi keanekaragaman populasi tersebut.

F. Pindah Silang (Crossover)

Proses pindah silang digunakan untuk keturunan sepasang kromosom anak dari sepasang kromosom orang tua dengan metode pindah silang. Operasi modifikasi pindah silang serupa dengan pindah silang *uniform-order* (Davis, 1991) yang diuraikan sebagai berikut:

Langkah 1. Acak beberapa string yang panjangnya sama dengan kromosom.

Langkah 2. Isi beberapa posisi pada anak 1 dengan mencopikan dari orang tua 1 dimana saja bit string yang memuat “1”.

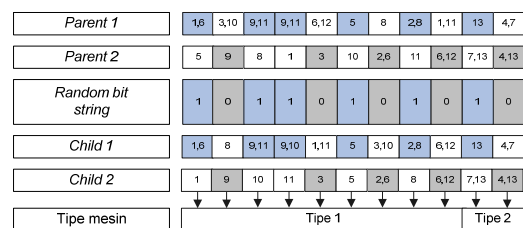
Langkah 3. Buat daftar gen dari orang tua 1 dihubungkan dengan bit string “0”.

Langkah 4. Mengubah urutan daftar gen sehingga urutannya sama dengan urutan gen yang tampak pada orang tua 2. Untuk gen yang mempunyai dua atau lebih operasi, operasi pertama ke orang tua 1 digunakan untuk mengubah posisi dari gen anak 1 mengikuti urutan dari gen orang tua 2. Jika jumlah gen pada daftar lebih dari korespondensi gen dengan operasi sama pada orang tua 2, maka secara sekuensial gen pada orang tua 2 akan diduplikasi dan ditambahkan ke akhir daftar.

Langkah 5. Salin urutan daftar gen ke dalam posisi kosong pada anak 1 sesuai hasil order pada langkah 4.

Langkah 6. Anak 2 diproduksi menggunakan proses serupa dengan proses diatas.

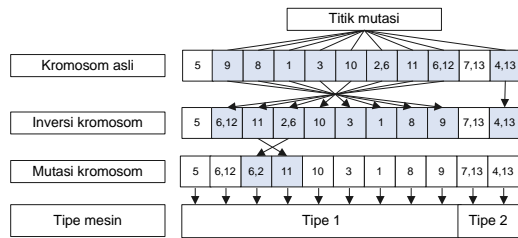
Proses pindah silang adalah proses acak dengan nilai probabilitas dari pindah silang. Pada operator pindah silang nilai probabilitasnya antara 0,6 – 1,0. Pada masalah penjadwalan *job shop*, setiap operasi harus diproses pada mesin bertipe tertentu. Gambar 3.1 menunjukkan proses pindah silang dilakukan.



Gambar 3.1 Operator modifikasi crossover Uniform-order

G. Mutasi (Mutation)

Operasi mutasi bersifat kritis terhadap kesuksesan algoritma genetika karena keanekaragaman arah pencarian dan menghindari konvergensi ke optimal lokal. Gambar 3.2 menunjukkan proses mutasi dilakukan.



Gambar 3.2 Operator modifikasi inversi mutasi

Banyak operator mutasi telah dipublikasikan. Pada pembelajaran ini, suatu modifikasi operasi mutasi untuk operator inversi mutasi telah dikembangkan (Holland, 1975) yang diuraikan sebagai berikut:

Langkah 1. Ambil satu kromosom dari populasi sebagai kromosom asli.

Langkah 2. Inversi kromosom dengan cara pada tipe mesin yang sama, jika panjang kromosom n , maka tukar posisi gen ke-2 dengan gen ke- n , gen ke-3 dengan gen ke- $(n-1)$, dan seterusnya.

Langkah 3. Mutasi kromosom dengan cara pada tipe mesin yang sama, tukar secara acak posisi gen yang bersebelahan antara gen ke-2 dan gen terakhir.

Pada operasi ini, pertama menginversi gen antar 2 gen terpilih secara acak pada suatu kromosom dengan terlebih dahulu menentukan nilai probabilitas mutasi.

IV. UJI COBA DAN ANALISIS

A. lingkungan Uji Coba

Uji coba dilakukan pada perangkat keras komputer dengan spesifikasi prosesor: AMD Turion(tm) 64 X2, memori 1 GB dan perangkat lunak: program penjadwalan *job shop* berbasis pemrograman Matlab dengan sistem operasi Windows XP SP2.

B. Data Uji Coba

Data uji coba yang digunakan adalah data produksi dan data efisiensi operasi, dimana data yang digunakan untuk masalah penjadwalan *job shop* yang sifat produknya berbaur dan multi produk serta merupakan produk massal. permasalahan penjadwalan *job shop* terbagi menjadi 2 bagian eksperimen. Pembagian data ini bertujuan untuk melihat kinerja program algoritma genetika dalam menyelesaikan berbagai jenis permasalahan penjadwalan *job shop*. Hasil yang didapatkan dalam program algoritma genetika yang dijalankan berupa lini produksi operasi dari setiap order.

Tabel I mempresentasikan data order berupa bobot tardiness dan bobot earliness saat uji coba dilakukan. Tabel II mempresentasikan data order berupa jumlah order dan tenggat waktu pada setiap kasus yang dijalankan. Tabel III mempresentasikan data efisiensi operasi di stasiun-kerja pada eksperimen 1 sedangkan tabel IV mempresentasikan data efisiensi operasi di stasiun-kerja pada eksperimen 2.

TABEL I
DATA ORDER: BOBOT *TARDINESS* DAN BOBOT *EARLINESS*

	Bobot <i>Tardiness</i>		Bobot <i>Earliness</i>	
	Order1	Order2	Order1	Order2
Eksperimen 1	5000	3000	100	100
Eksperimen 2	6000	4000	100	100

TABLE II
DATA ORDER: JUMLAH ORDER DAN TENGGAT WAKTU

	Kasus	Jumlah Order		Tenggat Waktu	
		Order 1	Order 2	Order 1	Order 2
Eks.1	1	1200	1200	15	12
	2	1000	1500	15	12
	3	1200	1200	18	13
Eks.2	1	1000	1000	15	12
	2	1200	1200	13	10
	3	2000	800	20	15

TABLE III
DATA ORDER: JUMLAH ORDER DAN TENGGAT WAKTU

Tipe Mesin	No. Stasiun-kerja	No. Operasi pada Order 1						No. Operasi pada Order 2					
		1	2	3	4	5	6	7	8	9	10	11	12
Mesin Jahit Lockstitch	1	0	100%	95%	90%	0	0	0	0	0	0	95%	95%
	2	0	70%	70%	70%	0	0	0	0	0	0	70%	70%
	3	0	80%	80%	80%	0	0	0	0	0	0	80%	80%
	4	0	70%	65%	65%	0	0	0	0	0	0	70%	65%
	5	0	85%	85%	80%	0	0	0	0	0	0	85%	85%
	6	0	90%	95%	90%	0	0	0	0	0	0	100%	95%
	7	0	95%	100%	100%	0	0	0	0	0	0	90%	100%
Mesin Jahit Overlock	8	90%	0	0	0	95%	95%	100%	90%	95%	100%	0	0
	9	80%	0	0	0	90%	90%	90%	85%	90%	85%	0	0
	10	70%	0	0	0	80%	80%	75%	80%	80%	75%	0	0
	11	75%	0	0	0	70%	80%	85%	80%	85%	80%	0	0
	12	100%	0	0	0	100%	90%	100%	95%	90%	95%	0	0

	13	95%	0	0	0	100%	90%	95%	100%	85%	90%	0	0
	14	90%	0	0	0	85%	100%	90%	95%	100%	85%	0	0
Waktu Standar (detik/potong)		320	342	480	288	310	330	138	158	140	130	288	150

TABLE IV
DATA ORDER: JUMLAH ORDER DAN TENGGAT WAKTU

Tipe Mesin	No. Stasiun-kerja	No. Operasi pada Order 1								No. Operasi pada Order 2				
		1	2	3	4	5	6	7	8	9	10	11	12	13
Mesin Jahit Lockstitch	1	100%	95%	90%	0	100%	100%	0	95%	95%	90%	100%	90%	0
	2	75%	70%	70%	0	75%	80%	0	70%	80%	70%	75%	70%	0
	3	80%	80%	85%	0	80%	60%	0	85%	80%	85%	80%	85%	0
	4	70%	65%	60%	0	70%	75%	0	65%	70%	65%	70%	65%	0
	5	85%	75%	80%	0	85%	95%	0	80%	85%	80%	90%	85%	0
	6	90%	85%	90%	0	90%	90%	0	95%	90%	85%	80%	90%	0
	7	95%	100%	100%	0	95%	85%	0	100%	100%	95%	90%	85%	0
	8	85%	90%	95%	0	85%	90%	0	90%	90%	95%	95%	90%	0
	9	90%	95%	85%	0	90%	95%	0	90%	90%	100%	85%	100%	0
Mesin Jahit Overlock	10	0	0	0	95%	0	0	95%	0	0	0	0	0	
	11	0	0	0	100%	0	0	95%	0	0	0	0	0	
Waktu Standar (detik/potong)		245	200	236	162	240	286	120	298	156	180	208	105	210

C. Metode Uji Coba

Pada setiap eksperimen memiliki data yang berbeda dalam hal bobot *earliness* dan bobot *tardiness* yang dikenakan. Pada setiap eksperimen juga memiliki beberapa kasus yang berbeda-beda dalam hal jumlah order yang dikenakan dan tenggat waktu penyelesaian order tersebut. Hasil yang didapatkan dalam program algoritma genetika yang dijalankan berupa lini produksi operasi dari setiap order, dimana dalam menghitung waktu proses setiap operasi adalah

$$T_{il} = \frac{ST_{il}}{\sum_{kj} X_{ilkj} EM_{ilkj} \gamma_{ilkj}} \quad (4.1)$$

Dimana waktu rata-rata proses operasi T_{il} , didapatkan dari waktu standar operasi ST_{il} tersebut dibagi jumlah efisiensi operasi EM_{ilkj} . $X_{ilkj} = 1$ jika operasi tersebut dikerjakan pada mesin yang mampu memprosesnya dan $X_{ilkj} = 0$ jika sebaliknya. Jika mesin M_{kj} hanya memproses operasi O_{il} , $\gamma_{ilkj} = 1$ dan jika mesin M_{kj} sama sekali tidak memproses operasi O_{il} , $\gamma_{ilkj} = 0$. Jika mesin M_{kj} memproses operasi O_{il} dan operasi lainnya secara simultan serta operasi O_{il} juga dijalankan pada mesin lainnya, $0 < \gamma_{ilkj} < 1$.

Pada uji coba eksperimen yang dilakukan, waktu transportasi (ET) dan waktu pengaturan mesin (STP) dianggap masuk dalam waktu proses (T). Parameter yang digunakan tetap, yaitu jumlah populasi = 200, probabilitas pindah silang = 0.6, probabilitas mutasi = 0.01, dan jumlah generasi = 60.

D. Hasil Uji Coba dan Analisis

Pada setiap eksperimen memiliki masalah penjadwalan sendiri dan pada setiap bagian eksperimen terdapat metode

yang berbeda yaitu mode 1 dan mode 2. Pada mode 1 pengolahan order 1 dijalankan dahulu kemudian selama waktu tertentu ditunda untuk menjalankan pengolahan order hingga selesai. Setelah itu pengolahan order 1 dijalankan kembali hingga selesai. Sifat produk yang dihasilkan multi produk. Sedangkan pada mode 2 pengolahan order 1 dijalankan dahulu kemudian saat waktu tertentu pengolahan order 1 dan order 2 dijalankan secara simultan sampai order 2 terpenuhi. Setelah itu pengolahan order 1 dijalankan kembali hingga selesai. Sifat produk yang dihasilkan berbaur dan multi produk.

Pada eksperimen 1, dua pengolahan order dijalankan pada 14 mesin dengan dua tipe mesin, yaitu mesin *lockstitch* dan mesin *overlock*. Terdapat tujuh mesin setiap tipe mesin. Mesin *lockstitch* antara mesin no. 1-7 dan mesin *overlock* antara mesin no. 8-12. Proses perakitan pakaian pada order 1 adalah dari operasi 1-6 dan pada order 2 dari operasi 7-12. Pada eksperimen ini, setiap mesin hanya dapat menampung satu operasi.

Pada eksperimen 2, dua pengolahan order dijalankan pada 13 mesin dengan dua tipe mesin, yaitu mesin *lockstitch* dan mesin *overlock*. Terdapat sembilan mesin tipe *lockstitch* dan dua mesin tipe *overlock*. Mesin *lockstitch* antara mesin no. 1-9 dan mesin *overlock* antara mesin no. 10-11. Proses perakitan pakaian pada order 1 adalah dari operasi 1-8 dan pada order 2 dari operasi 9-13. Pada eksperimen ini, diasumsikan setiap mesin dapat menampung maksimal dua operasi.

Pada tabel V merupakan hasil pada eksperimen 1 berupa lini operasi produksi dari mesin 1-14 dan terbagi dalam setiap pengolahan order yang dijalankan. Pada tabel VI merupakan hasil pada eksperimen 2 berupa lini operasi produksi dari mesin 1-11 dan setiap mesin mampu menampung maksimal 2 operasi serta terbagi dalam setiap pengolahan order yang dijalankan.

TABLE V
PENUGASAN MESIN PADA EKSPERIMEN 1

Penempatan operasi			No. mesin													
			1	2	3	4	5	6	7	8	9	10	11	12	13	14
Kasus 1	Mode 1	Order1	3	2	2	4	4	3	3	6	1	1	5	6	6	5
		Order2	12	11	11	12	11	11	12	10	7	7	9	8	8	9
	Mode 2	Order1	2	4	2	3	4	3	3	1	1	1	5	5	6	6
		duaorder	11	3	2	12	4	4	11	8	1	7	5	6	9	10
Kasus 2	Mode 1	Order1	2	4	2	3	4	3	3	6	5	1	5	6	1	1
		Order2	11	12	12	11	11	11	12	10	9	10	7	7	8	9
	Mode 2	Order1	4	3	4	2	4	2	3	6	1	1	5	6	6	5
		duaorder	12	3	2	4	11	12	11	6	7	1	10	8	9	5
Kasus 3	Mode 1	Order1	4	2	2	3	4	3	4	5	1	1	5	6	6	5
		Order2	12	11	11	11	12	12	12	10	8	7	8	9	9	7
	Mode 2	Order1	3	2	4	2	4	3	3	6	1	1	5	6	6	5
		duaorder	12	3	4	2	4	11	12	9	6	1	5	10	8	7

TABLE VI
PENUGASAN MESIN PADA EKSPERIMEN 2

Penempatan operasi			No. mesin										
			1	2	3	4	5	6	7	8	9	10	11
Kasus 1	Mode 1	Order1	1,3	8	2,6	3	8	1	2,5	5	6	4,7	4,7
		Order2	11	11,12	12	9,10	9,11	12	10,11	10,11	10,12	13	13
	Mode 2	Order1	2	8	6	5,8	2	1	6	5	3	4	7
		duaorder	1,12	10	2,11	9	5	11	3,6	1	8	4,7	13
Kasus 2	Mode 1	Order1	6	1,3	3	2	1	2,8	8	5	5,6	4	7
		Order2	10	11,12	9	12	10	11	10	9	11	13	13
	Mode 2	Order1	6	3	6,8	3	8	5	2	1,5	1	4	7
		duaorder	9	6	5	3	8	10	1,2	11	1,12	4,13	7,13
Kasus 3	Mode 1	Order1	6	1	6	1,3	8	2,5	2,8	5	3	4	7
		Order2	11	10,12	9	10	10,11	12	12	9	9	13	13
	Mode 2	Order1	8	2	5	2,3	3	6	1	5,6	1,8	4	7
		duaorder	6	8	6,9	5,8	3	11	9,10	1,10	2,12	4,7	13

TABLE VII
HASIL PENJADWALAN OPTIMASI GENETIKA

Eksperimen 1		Waktu penyelesaian order (hari)		Tenggat waktu (hari)		Total pinalti (\$)
		Order 1	Order 2	Order 1	Order 2	
Kasus 1	Mode 1	14,96	12	15	12	4
	Mode 2	14,98	11,99	15	12	3
Kasus 2	Mode 1	15	12	15	12	0
	Mode 2	14,97	12	15	12	3
Kasus 3	Mode 1	18	13	18	13	0
	Mode 2	17,97	13	18	13	3
Eksperimen 2						
Kasus 1	Mode 1	14,99	12	15	12	1
	Mode 2	14,97	12	15	12	3
Kasus 2	Mode 1	12,97	9,99	13	10	4
	Mode 2	12,97	10	13	10	3
Kasus 3	Mode 1	19,97	14,99	20	15	4
	Mode 2	19,95	15	20	15	5

Secara keseluruhan hasil menunjukkan bahwa program algoritma genetika yang diujikan dapat menyelesaikan permasalahan penjadwalan *job shop* yang produknya bersifat berbaur dan multi produk secara efektif. Pada tabel VII terlihat bahwa program algoritma genetika untuk masalah penjadwalan *job shop* mampu meminimalkan total pinalti E/T (*Earliness/Tardiness*).

V. CONCLUSIONS

Program algoritma genetika yang dijalankan dapat berjalan dengan efektif di dua eksperimen yang berbeda. Dimana eksperimen 1: setiap mesin hanya dapat menampung satu operasi sedangkan eksperimen 2: setiap mesin dapat menampung maksimal dua operasi.

Program algoritma genetika dapat digunakan pada persoalan penjadwalan *job shop* yang produknya multi produk (mode 1) maupun persoalan penjadwalan *job shop* yang produknya berbaur dan multi produk (mode2).

Program algoritma genetika dapat dijalankan pada berbagai data order dan data efisiensi operasi yang berbeda dengan ketentuan data-data tersebut memenuhi batasan-batasan dalam pemodelan yang sesuai dengan masalah penjadwalan *job shop*.

Pengembangan yang dapat dilakukan terhadap program algoritma genetika pada permasalahan penjadwalan *job shop* terutama produknya yang dihasilkan bersifat berbaur dan multi produk dengan mempertimbangkan efek ketidakpastian yang terjadi pada proses penjadwalan. Diantaranya adalah ketidakpastian permintaan konsumen, gangguan mesin, kekurangan bahan baku, ketidakhadiran operator mesin, dan lainnya. Hal-hal tersebut sangat berpengaruh dan menyebabkan perubahan yang cukup signifikan terhadap proses penjadwalan *job shop*. Begitu juga dengan pembatalan

order yang telah terjadwal akan berakibat pada perubahan susunan operasi yang dikerjakan pada mesin produksi.

REFERENCES

- [1] Watanabe, M., Ida, K., & Gen, M. (2005). A genetic algorithm with modified crossover operator and search area adaption for the jobshop scheduling problem. *Computers and Industrial Engineering*, 48(4), 743-752.
- [2] Ventura, J. A., & Kim, D. (2003). *Parallel machine scheduling with earliness-tardiness penalties and additional resource constraints*. *Computers and Operations Research*, 30(13), 1945-1958.
- [3] Seo, D. K., Klein, C. A., & Jang, W. (2005). *Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models*. *Computers and Industrial Engineering*, 48(2), 153-161.
- [4] Poon, P. W., & Carter, J. N. (1995). *Genetic algorithm crossover operators for ordering applications*. *Computers and Operations Research*, 22(1), 135-147.
- [5] Park, B. J., Choi, H. R., & Kim, H. S. (2003). *A hybrid genetic algorithm for the job shop scheduling problems*. *Computers and Industrial Engineering*, 45(4), 597-613.
- [6] Gordon, V., Proth, J., & Chu, C. (2002). *A survey of the state-of-the-art of common due date assignment and scheduling research*. *European Journal of Operational Research*, 139(1), 1-25.
- [7] Guo, Z.X., Wong, W.K., Leung, S.Y.S., Fan, J.T., Chan, S.F. (2006). *Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: A case study based on the apparel industry*. *Computers & Industrial Engineering* 50 (2006) 202-219.
- [8] Cheng, R. W., Gen, M., & Tsujimura, Y. (1996). *A tutorial survey of job-shop scheduling problems using genetic algorithms. I. Representation*. *Computers and Industrial Engineering*, 30(4), 983-997.
- [9] Lauff, V., & Werner, F. (2004). *Scheduling with common due date, earliness and tardiness penalties for multimachine problems: A survey*. *Mathematical and Computer Modeling*, 40(5-6), 637-655.
- [10] Brucker, P., Jurisch, B., & Sievers, B. (1994). *A branch-and-bound algorithm for the job-shop scheduling problem*. *Discrete Applied Mathematics*, 49(1-3), 107-127.