

APLIKASI KOMBINASI ALGORITMA GENETIK DAN DATA ENVELOPMENT ANALYSIS PADA PENJADWALAN FLOWSHOP MULTIKRITERIA

Herry Christian Palit¹, Haris Lienardo², I Gede Agus Widyadana³
^{1,2,3}Fakultas Teknologi Industri, Jurusan Teknik Industri, Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya
Email: herry@petra.ac.id, gede@petra.ac.id

ABSTRAK

Artikel ini membahas kombinasi algoritma genetik dengan *Data Envelopment Analysis* (DEA) untuk pemecahan masalah penjadwalan *flowshop* multikriteria. Kriteria-kriteria yang digunakan, yaitu *makespan*, total *weighted tardiness*, dan *mean flow time*. DEA digunakan untuk menghitung nilai keseluruhan kriteria dari setiap *sequence* dengan menggunakan nilai efisiensi relatif sebagai *fitted value* dalam algoritma genetik. Hal ini ditujukan agar nilai keseluruhan dari kriteria-kriteria yang ada tidak terikat pada satu jenis bobot saja. Kombinasi dua metode ini menghasilkan suatu algoritma yang mampu menghasilkan kumpulan solusi optimal dengan nilai efisiensi relatif yang tidak kalah jika dibandingkan dengan hasil dari model *Mixed Integer Programming* (MIP), dimana dari 30 masalah yang dibangkitkan, hanya ada 1 masalah (3,33%) yang memiliki efisiensi relatif di bawah 1.

Kata kunci: penjadwalan *flowshop*, algoritma genetik, *Data Envelopment Analysis*.

ABSTRACT

This article discusses the combination of genetic algorithm (GA) and Data Envelopment Analysis (DEA) to solve the flowshop scheduling problems with multicriteria. The criteria are makespan, total weighted tardiness, and mean flow time. DEA is used to calculate the overall value of criteria from each sequence. Relative efficiency value is employed as the fitted value in genetic algorithm, in order to have overall value that independent to a particular weight. The proposed algorithm that combines GA and DEA attain optimal solutions with relative efficiency as good as analytical solution, i.e., Mixed Integer Programming (MIP). From 30 problems generated, only one problem (3,33%) has relative efficiency less than 1.

Keywords: *flowshop scheduling problems, genetic algorithm, Data Envelopment Analysis.*

1. PENDAHULUAN

Penjadwalan merupakan hal yang penting dalam sistem produksi. Sistem produksi yang umumnya ditemukan adalah sistem *flowshop* dan *jobshop*. Dalam industri yang menggunakan sistem *flowshop*, penjadwalan dilakukan dengan mengatur urutan pengerjaan dari *jobs* yang dimiliki. Berbagai masalah dapat timbul karena penjadwalan yang dilakukan dengan cara yang tidak tepat, seperti waktu penyelesaian produksi yang terlalu lama, keterlambatan dari *due date* yang telah ditentukan, dan *idle* mesin.

Terdapat banyak algoritma yang dapat membantu proses penjadwalan *flowshop* seperti *Genetic Algorithm* (GA), *Simulated Annealing*, *Tabu Search*, maupun dengan *Mixed Integer Programming* (MIP). Algoritma-algoritma ini umumnya hanya dibuat untuk mencapai satu kriteria saja. Pada kenyataannya, suatu jadwal akan lebih baik apabila telah memperhitungkan seluruh kriteria penjadwalan yang ada dan bukan terikat pada satu kriteria saja, oleh karena itu algoritma yang memperhitungkan banyak kriteria (multikriteria) sekaligus diperlukan. Penelitian

sebelumnya mengenai penjadwalan multikriteria oleh Soetanto (1998) dan Pamungkas (2002) dilakukan dengan menggunakan algoritma genetik dengan *fitted value* berupa nilai keseluruhan dari kriteria-kriteria yang ada dengan bobot masing-masing kriteria sama dan solusinya adalah satu jadwal.

Pada kenyataannya, masing-masing pengambil keputusan di tiap departemen pada umumnya akan memiliki penilaian yang berbeda-beda terhadap kriteria-kriteria yang ada. Sebagai contoh, bagian produksi akan lebih menyukai jadwal dengan total waktu produksi terkecil untuk peningkatan kapasitas produksi, sedangkan bagian *marketing* akan lebih memilih jadwal lainnya yang memiliki keterlambatan yang lebih sedikit untuk tetap menjamin kepuasan konsumen, oleh karena itu diperlukan suatu metode penjadwalan yang dapat menghasilkan kumpulan jadwal produksi yang dapat memenuhi beberapa kriteria sekaligus. Arakawa *et al.* (1998) mengkombinasikan algoritma genetik dan *Data Envelopment Analysis* (DEA) untuk memperoleh kumpulan solusi optimal dari model matematik yang mempunyai banyak tujuan (*multiobjective*). Pada artikel ini, kombinasi algoritma genetik dan DEA digunakan untuk menyelesaikan masalah penjadwalan *flowshop* multikriteria, sehingga didapatkan kumpulan jadwal yang optimal dilihat dari keseluruhan kriteria yang ada.

2. PENJADWALAN FLOWSHOP

Penjadwalan adalah suatu kegiatan pengalokasian sumber daya (mesin, orang, dan lain-lain) yang dimiliki perusahaan untuk menyelesaikan pekerjaan (*job*) yang ada. Pada penjadwalan *flowshop*, sumber daya yang dialokasikan akan dilewati oleh setiap *job* secara berurutan atau dengan kata lain setiap *job* memiliki rute atau urutan tahap pengerjaan yang sama. Ukuran performansi penjadwalan tergantung pada kriteria yang digunakan, antara lain total waktu untuk penyelesaian semua *job* minimum (*makespan*), rata-rata keterlambatan yang minimum (*mean tardiness*), rata-rata waktu penyelesaian setiap *job* yang minimum (*mean flow time*), dan sebagainya.

3. ALGORITMA GENETIK

Algoritma genetik adalah suatu metode heuristik yang dipakai untuk menemukan solusi berdasarkan pada mekanisme alam yakni *crossover* dan *mutation* untuk menghasilkan generasi baru dan dilanjutkan dengan seleksi alam berdasarkan gen yang dimiliki, sehingga individu yang dapat bertahan hidup di dalam populasi adalah individu yang memiliki kombinasi gen yang baik.

Algoritma genetik dimulai dengan membentuk populasi (kumpulan solusi) awal. Pada algoritma genetik untuk penjadwalan, solusi awal dapat diperoleh secara *random* ataupun dengan algoritma sederhana seperti *Early Due Date* (mengurutkan *job* di mana *job* dengan *due date* yang paling kecil dikerjakan terlebih dahulu), *Shortest Processing Time* (mengurutkan *job* di mana *job* yang dijadwalkan dahulu memiliki total waktu proses yang paling kecil). Algoritma ini melalui banyak iterasi dimana dalam setiap iterasi, anggota dari populasi diseleksi untuk diproses baik secara *crossover* maupun *mutation* sehingga menghasilkan generasi yang baru.

Proses seleksi dapat dilakukan dengan metode *Holland's proportionate selection* atau *roulette wheel selection*. Pada metode ini, setiap anggota populasi dipilih secara *random* dimana setiap individu memiliki besar probabilitas untuk terpilih sesuai dengan nilai *fitted value* yang dimiliki dibandingkan dengan total *fitted value* dari seluruh populasi (Gen dan Cheng, 1997).

$$p_k = \frac{f_k}{\sum_{j=1}^{pop\ size} f_j} \quad (1)$$

dimana:

- p_k : probabilitas terpilihnya individu k dalam populasi
- f_k : nilai *fitted value* dari individu k
- f_j : nilai *fitted value* dari individu ke- j , dimana $j = 1, 2, \dots, pop\ size$
- $pop\ size$: jumlah seluruh individu dalam populasi

Dalam setiap iterasi, generasi yang dihasilkan dari proses *mutation* atau pun *crossover* dievaluasi nilainya. Dalam penjadwalan, nilai yang dievaluasi meliputi *makespan*, *lateness*, *weighted tardiness*, *number of tardy job*, dan nilai-nilai lain yang mempengaruhi keefisienan dari proses produksi. Selanjutnya hasil ini digunakan untuk proses seleksi di mana proses seleksi ini akan membuang *child* atau *parent* yang jelek.

Iterasi akan berjalan terus-menerus untuk menghasilkan anggota populasi yang semakin baik hingga syarat-syarat yang ditentukan terpenuhi misalnya:

- a. Jumlah iterasi maksimal yang telah ditetapkan
- b. Telah tercapainya nilai tujuan yang dikehendaki
- c. Tidak berubahnya anggota populasi selama jumlah iterasi yang ditetapkan

4. DATA ENVELOPMENT ANALYSIS (DEA)

Analisis dengan DEA didesain secara spesifik untuk mengukur efisiensi relatif suatu unit produksi dalam kondisi terdapat banyak *output* maupun banyak *input*, yang biasanya sulit diukur oleh teknik analisis pengukuran efisiensi rasio ataupun analisis regresi. Dalam DEA, efisiensi dinyatakan sebagai rasio antara *total output* tertimbang dan *total input* tertimbang, dimana setiap unit keputusan, yang lazim disebut dengan *Decision Making Unit* (DMU), diasumsikan bebas menentukan bobot untuk setiap variabel-variabel *output* maupun *input* yang ada, asalkan mampu memenuhi dua kondisi yang disyaratkan, yaitu:

- Bobot tidak boleh negatif
- Bobot harus bersifat universal atau tidak menghasilkan indikator efisiensi di atas normal atau lebih besar dari nilai satu

Untuk mencapai tingkat efisiensi yang maksimum, maka setiap DMU cenderung memiliki pola untuk menetapkan bobot tinggi pada *input* yang sedikit digunakan, dan pada *output* yang banyak dihasilkan, dimana bobot yang dipilih tersebut tidak semata-mata menggambarkan suatu nilai ekonomis, tetapi lebih merupakan suatu besaran kuantitatif untuk memaksimalkan efisiensi DMU yang bersangkutan. Berikut ini adalah rumusan perhitungan efisiensi relatif dengan DEA (Ramanathan, 2003).

$$\left. \begin{aligned}
 \text{Max } E_1 &= \frac{\sum_n o_{n1} \times w_n}{\sum_n i_{n1} \times y_n} \\
 \text{Kendala} \\
 \frac{\sum_n o_{na} \times w_n}{\sum_n i_{na} \times y_n} &\leq 1, \text{ untuk setiap } a \\
 w_n, y_n &\geq \varepsilon
 \end{aligned} \right\} \text{efisiensi relatif unit 1} \quad (2)$$

dimana:

- w_n : bobot dari *output* ke n
- a : unit-unit yang dievaluasi efisiensinya
- o_{na} : nilai dari *output* ke n unit a
- y_n : bobot dari *input* ke n
- i_{na} : nilai dari *input* ke n unit a
- E_1 : efisiensi relatif unit 1
- ε : nilai yang sangat kecil (10^{-6})

Model matematis DEA suatu unit dapat dirumuskan ke dalam sebuah program linear fraksional dengan menjadikan bobot-bobot *input* dan *output* dari unit bersangkutan sebagai variabel keputusan seperti terlihat pada persamaan (3). Suatu unit dikatakan efisien apabila memiliki nilai efisiensi relatif sebesar 1, sebaliknya bila nilainya di bawah 1 dikatakan tidak efisien. Perhitungan dengan cara yang sama akan dilakukan untuk unit 2, 3, dan seterusnya sampai hasil dari keseluruhan unit yang akan dievaluasi efisiensinya diperoleh.

$$\left. \begin{aligned} &Max E_1 = \sum_n o_{n1} \times w_n \\ &Kendala \\ &\sum_n i_{n1} \times y_n = 1 \\ &\sum_n o_{na} \times w_n - \sum_n i_{na} \times y_n \leq 0 \quad , \text{ untuk setiap } a \\ &w_n, y_n \geq \varepsilon \end{aligned} \right\} \text{efisiensi relatif unit 1} \quad (3)$$

5. MIP PENJADWALAN FLOWSHOP

Mixed Integer Programming (MIP) adalah suatu program linear yang di dalamnya terdapat variabel integer dan bukan integer. Model ini bertujuan untuk meminimumkan *makespan* dalam penjadwalan *flowshop*. Model ini juga akan dipakai sebagai dasar untuk membuat model yang bertujuan untuk meminimumkan *mean flow time* dan *total weighted tardiness*.

Model MIP untuk permasalahan penjadwalan *flowshop* dengan kriteria *makespan* adalah sebagai berikut (Pinedo, 2002):

- Fungsi tujuan :

$$Min z = F_{max} \quad (4)$$

- Kendala-kendala yang dipakai terdiri atas:

- Kendala suatu *job* hanya dapat dijadwalkan sekali dan satu urutan jadwal hanya terdapat satu *job* yang sama:

$$\sum_{i=1}^n z_{ij} = 1 \quad j = 1, \dots, n \quad (5)$$

$$\sum_{j=1}^n z_{ij} = 1 \quad i = 1, \dots, n \quad (6)$$

dimana n : total jumlah *job*.

- Kendala yang menunjukkan syarat *job* yang berurutan pada setiap mesin:

$$\sum_{i=1}^n p_{ki} z_{i,j+1} + W_{j+1,k} + Id_{j+1,k} = W_{j,k} + \sum_{i=1}^n p_{k+1,i} z_{ij} + Id_{j+1,k+1}$$

$$j = 1, \dots, n-1 \text{ dan } k = 1, \dots, m-1 \quad (7)$$

dimana m : jumlah mesin.

- Kendala untuk menghitung *makespan*:

$$\sum_{j=1}^n \sum_{i=1}^n p_{mi} z_{ij} + \sum_{j=1}^n Id_{jm} = F_{max} \quad (8)$$

- Kendala *idle time* setiap mesin sebelum dimulainya *job* urutan pertama:

$$\sum_{r=1}^{k-1} \sum_{i=1}^n p_{ri} z_{i1} = Id_{1k} \quad \text{dimana } k = 2, \dots, m \quad (9)$$

- Kendala *waiting time job* urutan pertama pada semua mesin:

$$W_{1k} = 0 \quad (10)$$

Variabel keputusan yang digunakan:

$$z_{ij} \begin{cases} = 0, \text{ bila } \textit{job } i \text{ ditempatkan pada urutan ke } j \\ = 1, \text{ bila } \textit{job } i \text{ tidak ditempatkan pada urutan ke } j \end{cases}$$

Id_{jk} : *Idle time* pada mesin k sebelum dimulainya *job* urutan ke j .

W_{jk} : *Waiting time* *job* urutan ke j setelah dikerjakan pada mesin k sebelum dikerjakan pada mesin $k+1$.

P_{ki} : Waktu proses *job* i pada mesin ke k .

F_{max} : *Finish time* dari *job* pada urutan terakhir

Model MIP untuk kriteria *mean flowtime* merupakan sedikit pengembangan dari model MIP *makespan*. Pada kendala di persamaan (9), F_{max} saja yang dihitung. Pengembangan model dilakukan dengan menambah satu perangkat kendala yang menghitung *finish time* (F_i) dari semua urutan *job*.

$$\sum_{j=1}^n \sum_{i=1}^n p_{mi} z_{ij} + \sum_{j=1}^n Id_{jm} = F_i \quad i = 1, 2, 3, \dots, P \quad (11)$$

Berdasarkan kendala ini, maka dapat ditentukan rumus untuk fungsi tujuan *mean flowtime*

$$\text{Min } z = \frac{\sum_{i=1}^P F_i}{P} \quad (12)$$

dimana: P : total *job* yang dikerjakan.

Model matematik untuk kriteria *total weighted tardiness* tidak linear. Perbedaannya dengan model MIP untuk kriteria *mean flowtime* terdapat pada fungsi tujuan dan beberapa tambahan pada fungsi kendala. Selain itu juga terdapat tambahan beberapa variabel yakni:

D_i : *due date* dari *job* i

Dd_j : *due date* dari *job* urutan ke j

W_i : bobot keterlambatan dari *job* i

B_j : bobot keterlambatan dari *job* urutan ke j

T_j : *tardiness job* urutan ke j

Tambahan kendala pada model MIP ini terdiri atas:

- Kendala untuk menghitung bobot keterlambatan pada setiap urutan

$$B_j = \sum_{i=1}^P z_{ij} \cdot W_i \quad j = 1, 2, \dots, P \quad (13)$$

- Kendala untuk menghitung *due date* dari setiap urutan *job*

$$Dd_j = \sum_{i=1}^P z_{ij} \cdot D_i \quad j = 1, 2, \dots, P \quad (14)$$

- Kendala besar keterlambatan dari setiap urutan *job*

$$\begin{aligned} T_j &\geq F_j - Dd_j \\ T_j &\geq 0 \end{aligned} \quad j = 1, 2, \dots, P \quad (15)$$

- Kendala-kendala ini berguna untuk menghitung fungsi tujuan *total weighted tardiness*.

$$\text{Min } z = \sum_{j=1}^P B_j \times T_j \quad (16)$$

6. ALGORITMA GA-DEA PADA PENJADWALAN FLOWSHOP

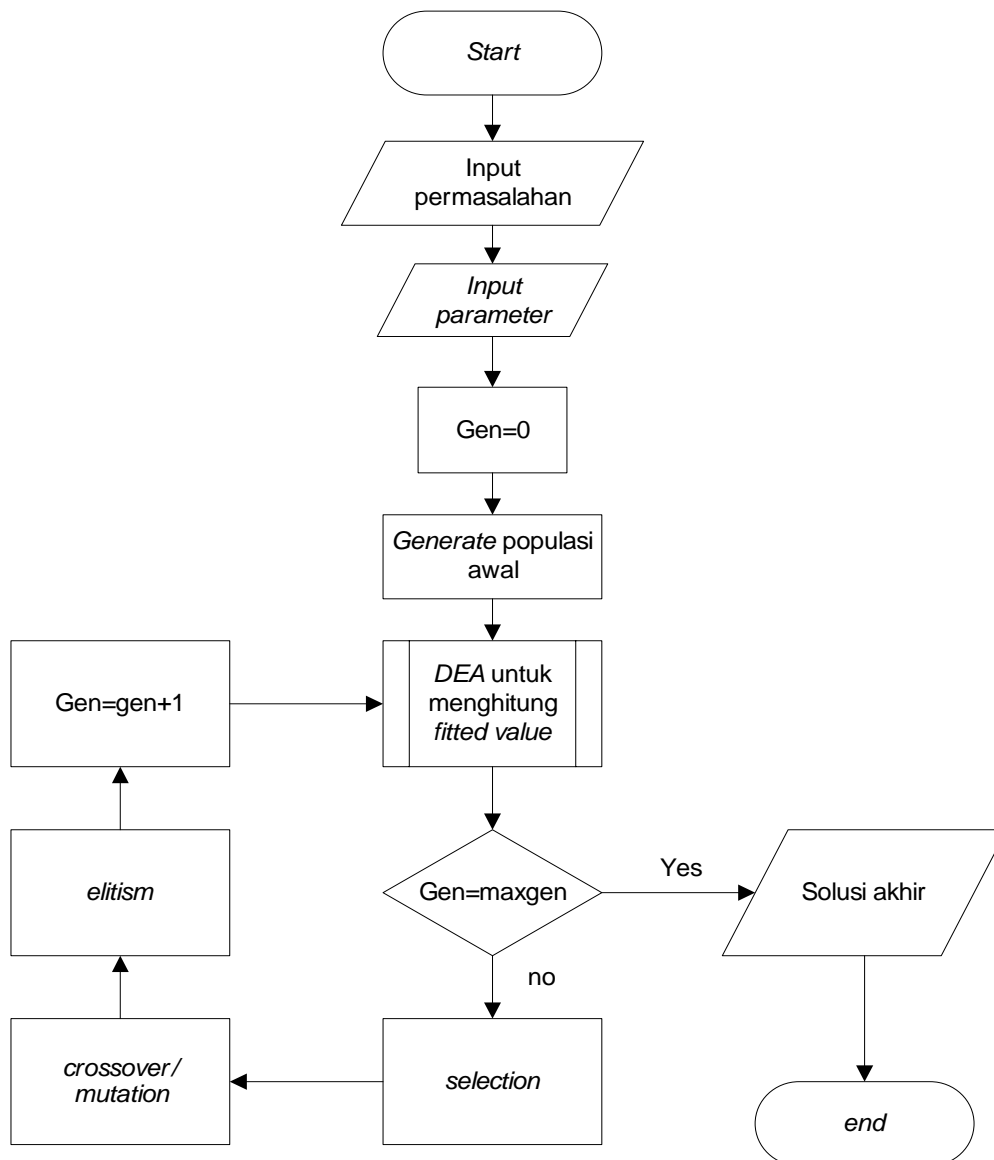
Kombinasi antara algoritma genetik dan *Data Envelopment Analysis* ditujukan untuk menyelesaikan permasalahan penjadwalan multikriteria. Metode *Data Envelopment Analysis* digunakan untuk menentukan *fitted value* dari algoritma genetik, yakni dengan menghitung nilai efisiensi relatif dari setiap individu dalam satu generasi. Secara keseluruhan, algoritma ini dapat dilihat pada Gambar 1.

Algoritma ini dimulai dengan memasukkan data permasalahan. Pada penelitian ini, permasalahan yang dibahas adalah penjadwalan *flowshop* dengan kriteria-kriteria *makespan*, *total weighted tardiness*, dan *mean flow time*. Data yang digunakan meliputi jumlah mesin, jumlah *job* yang akan dijadwalkan, lama penyelesaian setiap *job* di setiap mesin, *due date*, dan bobot keterlambatan dari setiap *job*.

Algoritma genetik juga memiliki parameter awal yang harus di-*input*-kan dahulu yakni jumlah maksimum generasi, maksimum populasi, dan besar probabilitas mutasi atau *crossover*. Maksimum generasi menandakan jumlah iterasi maksimal yang dijalankan dalam algoritma ini. Maksimum generasi dapat ditentukan dengan melihat jumlah generasi yang diperlukan untuk mencapai *local optimum*. *Local optimum* tercapai pada saat seluruh individu dalam populasi di suatu generasi memiliki kombinasi gen yang hampir sama satu sama lain (*nominal convergence*). Nilai maksimum populasi menunjukkan jumlah maksimum individu yang boleh berada di dalam suatu generasi. Probabilitas mutasi menunjukkan besar kemungkinan dari anak yang dihasilkan pada proses seleksi untuk mengalami mutasi atau *crossover*. Pada proses seleksi alam kemungkinan terjadinya mutasi relatif kecil, oleh karena itu pada penelitian ini nilai probabilitas mutasi ditentukan sebesar 0,1. Pada penelitian ini parameter awal dan kombinasi *job*-mesin yang digunakan dalam perhitungan seperti terlihat pada Tabel 1. Pada masing-masing kombinasi akan dilakukan replikasi sebanyak 5 kali dengan data yang berbeda yang dibangkitkan secara random, sehingga secara total terdapat 30 masalah.

Tabel 1. Parameter perhitungan kombinasi GA-DEA

Jumlah <i>job</i>	Jumlah mesin	Max populasi	Max generasi
5	5	10	15
	10	10	
	15	10	
10	5	20	50
	10	20	
	15	20	



Gambar 1. Flowchart algoritma GA-DEA

Kromosom dalam algoritma genetik ini terdiri atas gen yang berisi *job*. Posisi dari gen menunjukkan urutan pengerjaan dari *job* yang ada. Setiap *job* hanya mengisi satu gen saja.

Populasi awal dibuat sebanyak jumlah maksimal populasi yang telah ditentukan sebelumnya. Dalam penelitian ini, individu pertama dalam populasi awal ditentukan dengan menggunakan metode *Early Due Date* (EDD) dan *Shortest Processing Time* (SPT). Metode EDD menentukan *sequences job* berdasarkan *due date* yang dikehendaki oleh konsumen di mana *job* dengan *due date* terdekat dikerjakan terlebih dahulu hingga *job* dengan *due date* terlama. Pada metode SPT, urutan *job* ditentukan berdasarkan total waktu kerja masing-masing *job* di semua mesin. *Job* dengan total waktu keseluruhan terpendek akan dikerjakan terlebih dahulu, dan dilanjutkan dengan *job* lainnya hingga *job* dengan waktu keseluruhan terbesar. Individu-individu lain dalam populasi awal akan dibentuk urutannya secara random. Semua individu dalam populasi dihitung nilai *makespan*, *total weighted tardiness*, dan *mean flowtime*. Nilai-nilai ini akan digunakan dalam perhitungan *fitted value* dengan metode DEA.

Fitted value diperoleh dengan menghitung nilai efisiensi relatif dari setiap individu di dalam suatu generasi. Dalam perhitungan efisiensi relatif diperlukan nilai *input* dan *output*. Dalam masalah penjadwalan *flowshop* ini, nilai *makespan*, *total weighted tardiness*, dan *mean flowtime* dari setiap individu akan digunakan sebagai *input*. Sedangkan *output* yang dihasilkan adalah sama yakni terselesaikannya semua *job* yang dimiliki. Terlihat bahwa *output* yang dihasilkan oleh setiap unit individu adalah sama karena itu *virtual output* dapat digunakan, di mana nilai *output* dari setiap unit yang dievaluasi adalah 1.

$$\left. \begin{aligned}
 & \text{Min } z = y_1 \times i_{1a} + y_2 \times i_{2a} + y_3 \times i_{3a} \\
 & \text{Kendala} \\
 & y_1 \times i_{1l} + y_2 \times i_{2l} + y_3 \times i_{3l} \geq 1 \\
 & \qquad \qquad \qquad , \text{ untuk setiap individu } l \\
 & y_1, y_2, y_3 \geq 10^{-6} \\
 & \text{fitted value (eff relatif) individu } a = \frac{1}{\min z}
 \end{aligned} \right\} \text{fitted value individu } a \quad (17)$$

dimana:

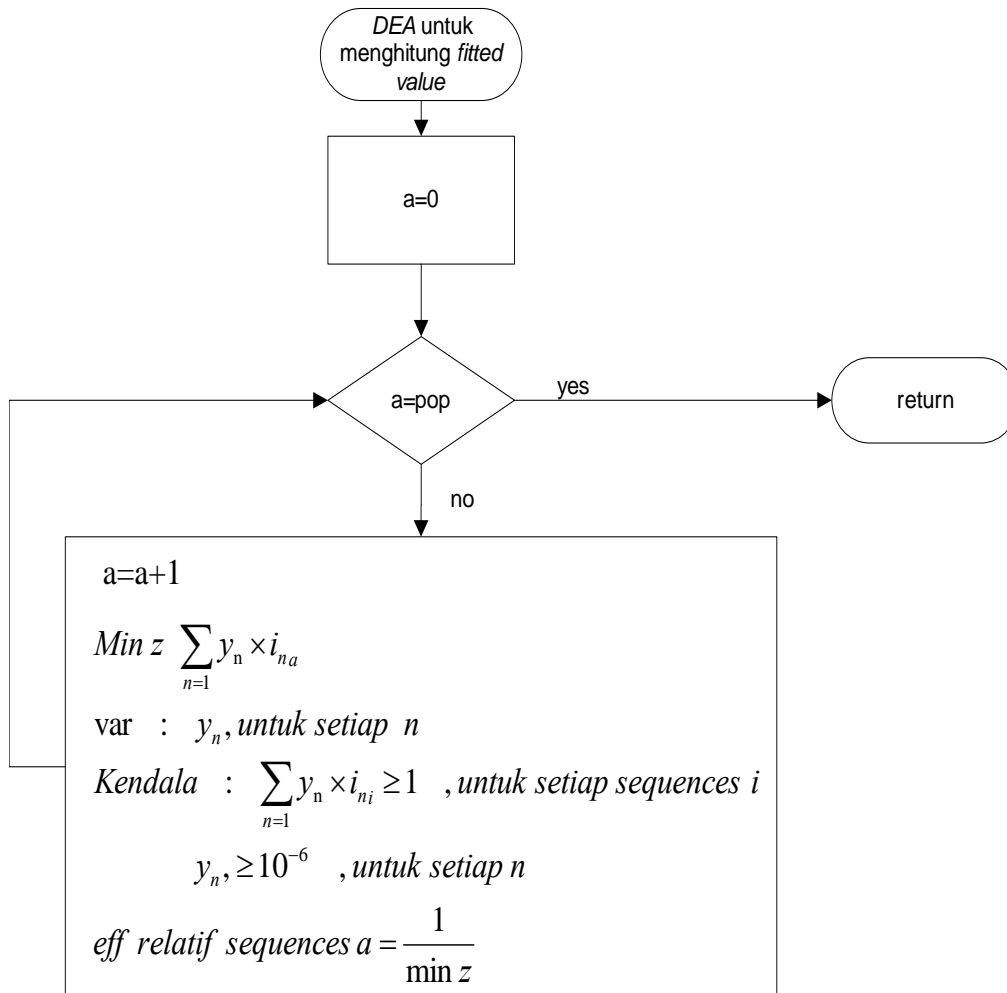
- y_1 : bobot dari *makespan*
- y_2 : bobot dari *total weighted tardiness*
- y_3 : bobot dari *mean flow time*
- i_{1l} : nilai *makespan* pada individu ke l
- i_{2l} : nilai *total weighted tardiness* pada individu ke l
- i_{3l} : nilai *mean flow time* pada individu ke l

Secara lengkap skema perhitungan *fitted value* dari suatu generasi dapat dilihat pada Gambar 2.

Nilai *fitted value* yang diperoleh akan dipakai untuk proses seleksi. Proses seleksi bertujuan untuk memilih pasangan *parent* yang mengalami *crossover*. Probabilitas terpilihnya *parent* ditentukan oleh nilai *fitted value* yang dimiliki sehingga semakin baik *fitted value*, semakin besar pula peluangnya untuk dipilih sebagai *parents* (Holland's *proportionate selection* atau *roulette wheel selection*). Mengingat proses *crossover* akan menghasilkan dua *offspring*/anak serta adanya batasan jumlah maksimal individu dalam populasi maka jumlah pasangan *parents* yang dipilih adalah setengah dari ukuran populasi.

Setiap pasangan *parents* dikondisikan untuk mengalami mutasi atau *crossover*. Pasangan *parents* akan mengalami mutasi bila angka acak yang dihasilkan lebih kecil dari probabilitas

mutasi yang telah ditetapkan. Pasangan *parents* akan mengalami *crossover* bila angka acak lebih besar dari probabilitas mutasi. Ada pun metode *crossover* yang dipakai adalah *Partial Mapped Crossover* (PMX) sedangkan metode mutasi yang dipakai adalah *displacement mutation*.



Gambar 2. Flowchart perhitungan *fitted value*

Elitism membuat hasil terbaik pada generasi sebelumnya ikut masuk ke dalam populasi pada generasi yang baru dan ikut dievaluasi ulang *fitted value* yang dimilikinya. Hasil terbaik adalah individu-individu (dapat lebih dari satu) pada generasi sebelumnya yang memiliki nilai efisiensi relatif 1. Proses *elitism* ini juga disertai dengan satu syarat yaitu bila terdapat anak pada generasi yang baru yang sama dengan individu elit pada generasi sebelumnya, maka individu elit ini tidak perlu disisipkan lagi pada generasi yang baru. Hal ini ditujukan untuk mencegah adanya individu yang serupa pada generasi yang baru.

Proses *elitism* juga berdampak pada penambahan jumlah individu yang melebihi jumlah maksimal populasi. Untuk mengatasi hal ini, maka dilakukan pembuangan individu-individu dengan nilai *fitted value* yang terjelek untuk digantikan dengan individu elit dari generasi sebelumnya. Solusi akhir dari algoritma ini adalah semua urutan *job* yang memiliki *fitted value* sebesar 1 pada generasi terakhir yang telah ditentukan pada parameter awal.

7. ANALISA

Perbandingan algoritma GA-DEA dengan MIP dilakukan dengan menghitung nilai efisiensi relatif dari jadwal (*sequence*) hasil algoritma GA-DEA dibandingkan dengan jadwal dari masing-masing kriteria yang diperoleh dengan MIP. Jadi kumpulan solusi jadwal hasil algoritma GA-DEA dari 30 masalah yang dibangkitkan akan dibandingkan dengan MIP untuk masing-masing kriteria menggunakan DEA seperti yang terdapat pada persamaan 17. Dari hasil perhitungan yang dilakukan terhadap 30 masalah, hanya terdapat satu masalah yang memiliki nilai efisiensi relatif di bawah 1, yaitu pada penjadwalan 5 job 5 mesin masalah pertama seperti terlihat pada Tabel 2.

Tabel 2. Tabel output 5 job 5 mesin masalah 1

<i>Sequence</i>	Solusi GA-DEA ke-							Solusi MIP		
	1	2	3	4	5	6	7	<i>Makespan</i>	<i>Weighted tardiness</i>	<i>Mean flowtime</i>
1	3	3	3	5	3	3	5	1	3	5
2	2	2	2	3	2	2	1	5	2	1
3	4	5	4	2	4	5	4	3	4	4
4	5	4	5	4	5	4	3	2	5	3
5	1	1	1	1	1	1	2	4	1	2
<i>Makespan</i>	307	307	307	314	307	307	311	297	307	311
<i>Total weighted tardiness</i>	87	91	87	404	87	91	1480	936	87	1480
<i>Mean flowtime</i>	233	232,8	233	231,2	233	232,8	225,8	226,2	233	225,8
<i>Eff relatif</i>	1	1	1	0,996	1	1	1	1	1	1

Pada Tabel 2 tampak bahwa *sequence* keempat dari algoritma GA-DEA memiliki nilai efisiensi relatif yang lebih jelek dari output MIP, akan tetapi secara keseluruhan algoritma GA-DEA masih memiliki efisiensi relatif yang sebanding dengan MIP dan bisa dikatakan GA-DEA tidak kalah dari MIP.

Solusi dari MIP untuk kriteria *makespan*, *mean flow time*, *total weighted tardiness* memiliki nilai efisiensi relatif 1, karena jadwal yang dihasilkan hanya fokus pada satu kriteria. GA-DEA memiliki perbedaan dengan MIP dalam hal penitikberatan kriteria untuk mencapai nilai efisiensi relatif 1. GA-DEA memiliki beberapa macam fokus dalam sekali jalannya algoritma ini. GA-DEA dapat memfokuskan pada satu jenis kriteria, dua kriteria, atau pun tiga kriteria secara sekaligus.

8. KESIMPULAN

Algoritma GA-DEA memiliki kemampuan untuk menghasilkan solusi yang memiliki nilai efisiensi relatif 1 setelah dibandingkan dengan solusi optimal dari masing-masing kriteria yang diperoleh dengan MIP. Hal ini dapat dilihat dengan hanya ada 1 dari 30 masalah (3,33%) yang memiliki efisiensi relatif terbaik dibawah 1. Berarti ini menunjukkan bahwa nilai keseluruhan dari output algoritma GA-DEA tidak kalah dari MIP, selain itu GA-DEA juga dapat memberikan sekumpulan solusi jadwal yang baik secara sekaligus bagi tiap pengambil keputusan sesuai dengan kebutuhan dari departemen yang bersangkutan.

DAFTAR PUSTAKA

- Arakawa, M., Nakayama, H., Hagiwara, and I., Yamakawa, H., 1998. "Multiobjective Optimization Using Adaptive Range Genetic Algorithms with Data Envelopment Analysis." *American Institute in Aeronautics and Astronautics*, p. 1–9.
- Gen, M., and Cheng, R., 1997. *Genetic Algorithms and Engeneering Design*, John Wiley & Sons, Canada.
- Pamungkas, A., 2002. *Perbandingan Kinerja Algoritma Genetik dan Simulated Annealing untuk Masalah Multiple Objectives pada Penjadwalan Flowshop*, Tugas Akhir, Jurusan Teknik Industri, Universitas Kristen Petra, Surabaya.
- Pinedo, M., 2002. *Scheduling Theory, Algorithms, and Systems*, 2nd edition, Prentice Hall Inc., New Jersey.
- Ramanathan, R., 2003. *An Introduction to Data Envelopment Analysis*, Sage Publications, New Delhi.
- Soetanto, T.V., 1998. *Penjadwalan Flowshop dengan Algoritma Genetik*, Tugas Akhir, Jurusan Teknik Industri, Universitas Kristen Petra, Surabaya.