

Penjadwalan Pekerjaan pada *No-Wait Flowshop* dengan Kriteria Minimasi Total Waktu Tinggal Aktual Menggunakan Algoritma Genetik

Urip Sarwo Sambodo, Yuniaristanto dan Cucuk Nur Rosyidi

Laboratorium Sistem Produksi, Jurusan Teknik Industri

Universitas Sebelas Maret Surakarta

Abstract

This research uses genetic algorithm for no-wait flowshop scheduling problem. The shop requires the condition that each job be processed without any interruptions on all m machines and between two consecutive machine. This means that the start of processing a job on a given machine may be delayed in order to make the completion of the job on the machine coincide with this beginning of processing the job on the subsequent machine. The objective for the model is to minimize total actual flow time accommodating the condition that job to be processed arrive at the shop at any times when needed and the condition that the completed job be delivered at a common due-date. Genetic algorithm is randomly and structured search to find optimal solution. Numerical experience shows that for a given scheduling problem, genetic algorithm solution better than before (Yuniaristanto, 2001).

Keywords : *actual flowtime, genetic algorithm, no-wait flowshop*

1. Pendahuluan

Penjadwalan pada *flowshop* terbagi menjadi dua macam yaitu *classical flowshop* dan *no-wait flowshop*. Pada *classical flowshop* mengijinkan adanya pekerjaan yang menunggu diantara dua mesin dan untuk mendapatkan penjadwalan yang optimal tidak memerlukan adanya *delay* pada mesin pertama. Sedangkan pada *no-wait flowshop*, saat pekerjaan selesai diproses pada suatu mesin harus sama dengan saat mulai proses pada mesin berikutnya dan tidak mengijinkan adanya *delay* di antara dua mesin. Terdapat beberapa industri yang sesuai dengan kondisi *no-wait flowshop*, diantaranya industri baja, kimia dan plastik (Aldowaisan dan Allahverdi, 1998).

Banyak literatur yang membahas penjadwalan pekerjaan (*job*) pada sistem produksi *no-wait flowshop*, diantaranya adalah Srikandarajah dan Ladet (1986), Van Der Veen dan van Dal (1991), Kamoun dan Srikandarajah (1993), Allahverdi dan Aldowaisan (2000) dan Yuniaristanto (2001). Srikandarajah dan Ladet (1986) meneliti kompleksitas komputasi dari persoalan penjadwalan *no-wait flowshop*. Van Der Veen dan van Dal (1991) mengembangkan persoalan *no-wait flowshop* dengan kriteria makespan dan diformulasikan sebagai *travelling salesman problem*. Kemudian Kamoun dan Srikandarajah (1993) melakukan penelitian dan menunjukkan bahwa *no-wait flowshop* untuk lebih dari dua mesin dengan kriteria minimasi *cycle time* merupakan masalah *NP-Hard*. Aldowaisan dan Allahverdi (1998) mengembangkan penelitian pada *no-wait flowshop* dengan kriteria total waktu tinggal dan mendefinisikan waktu *setup* terpisah dari waktu proses. Dan kemudian Aldowaisan dan Allahverdi (2000) mengembangkan untuk tiga mesin dengan kriteria *total completion time*. Dari penelitian yang telah dilakukan di atas masih mengasumsikan bahwa saat $t=0$ dan belum memasukkan *due date*

sebagai faktor pembatas. Dan penelitian–penelitian di atas hanya terbatas pada dua atau tiga mesin. Halim dan Ohta (1993, 1994) memperbaiki kekurangan kriteria waktu tinggal dengan mengusulkan kriteria baru yang disebut waktu tinggal aktual (*actual flow time*), yaitu lamanya suatu pekerjaan berada di dalam pabrik dari saat mulai proses sampai *due date*. Kriteria baru ini digunakan untuk mengakomodasikan kondisi yang lebih realistik dan konsep tepat waktu, baik tepat waktu kedatangan material maupun tepat waktu pengiriman.

Kemudian Yuniaristanto (2001) melakukan penelitian untuk mengembangkan suatu model penjadwalan pekerjaan pada *no-wait flowshop* statis m mesin untuk kasus *common due-date* dengan kriteria minimasi total waktu tinggal aktual yang mengasumsikan setiap *stage* hanya terdiri dari satu mesin. Untuk mengurutkan pekerjaan pada permasalahan tersebut digunakan algoritma yang dikembangkan oleh Halim dan Ohta (1993). Algoritma tersebut dimodifikasi untuk mengakomodasi kondisi *no-wait*. Hasil pengurutan pekerjaan kemudian dijadwalkan dengan menggunakan pasangan pekerjaan untuk memilih urutan pekerjaan yang akan dijadwalkan. Algoritma yang dikembangkan mempunyai kelemahan, yaitu :

1. Prioritas pemilihan pekerjaan menggunakan prosedur pasangan pekerjaan sehingga kurang mewakili untuk jumlah pekerjaan banyak
2. *Mean error* meningkat dengan bertambahnya jumlah pekerjaan (lihat gambar 1.1).

Untuk mendapatkan solusi yang optimum pada permasalahan penjadwalan biasanya diselesaikan dengan pendekatan optimasi atau heuristik. Algoritma genetik merupakan salah satu pendekatan heuristik yang dapat digunakan untuk memecahkan permasalahan–permasalahan yang kompleks. Algoritma genetik merupakan algoritma pencarian *random* terstruktur yang didasarkan pada analogi mekanisme seleksi dan informasi genetik alami.

2. Model Penjadwalan Pekerjaan pada No-Wait Flowshop

Model penjadwalan pekerjaan pada *no-wait flowshop* dengan total waktu tinggal aktual dikembangkan oleh Yuniaristanto (2001). Model yang dikembangkan menggambarkan kondisi penjadwalan pekerjaan pada *no-wait flowshop* yang meminimasi waktu tinggal aktual. Pada *no-wait flowshop*, saat mulai proses pekerjaan pada suatu mesin sama dengan saat selesai pekerjaan pada mesin sebelumnya tanpa menunggu. Formulasi matematis model penjadwalan *job common-due-date-flowshop* dengan *no-wait* dirumuskan sebagai berikut :

$$\text{Minimasi } F^a = \sum_{i=1}^n (d - B_{[i,1]}) \quad (1)$$

Pembatas :

$$B_{[n,1]} \geq 0, \quad (2)$$

$$B_{[i,m]} \leq B_{[i-1,m]} - t_{[i,m]} - s_{[i-1,m]}, \quad i = 2, \dots, n, \quad (3)$$

$$B_{[i,k]} = \min\{B_{[i-1,k]} - t_{[i,k]} - s_{[i-1,k]}, B_{[i,k+1]} - t_{[i,k]}\}, \\ i = 2, \dots, n; \quad k = 1, \dots, m-1, \quad (4)$$

$$B_{[i,k]} = B_{[i,k+1]} - t_{[i,k]}, \quad \forall i; \quad k = 1, \dots, m-1 \quad (5)$$

$$B_{[1,m]} = d - t_{[1,m]}, \quad (6)$$

Pembatas (2) menunjukkan bahwa pemrosesan pekerjaan pertama dilakukan pada saat nol atau sesudahnya. Pembatas (3), (4) dan (5) sebagai pembatas kondisi *no-wait* untuk menyusun jadwal produksi dengan pendekatan penjadwalan mundur dan saat mulai proses stasiun kerja

hilir mengendalikan saat mulai proses stasiun kerja hulu. Pembatas (6) menyatakan bahwa proses pekerjaan terakhir J_1 harus selesai tepat pada saat *due date*.

Pada model *job common-due-date-flowshop* dengan *no-wait* (*JCFN*), pekerjaan yang akan diproses dapat datang kapan saja saat diperlukan dan pekerjaan yang sudah diproses harus dikirim saat *common due-date*. Sehingga model ini tidak mengijinkan adanya pekerjaan yang *tardy*.

Model *JCFN* merupakan masalah *NP-hard* (Kamoun dan Srikandarajah, 1993), sehingga perlu dikembangkan pendekatan heuristik untuk menjadwalkan semua pekerjaan. Formulasi matematis model *JCFN* digunakan untuk menjadwalkan urutan pekerjaan terpilih yang dapat meminimasi total waktu tinggal aktual.

3. Algoritma Genetik

Algoritma genetik dikembangkan dengan mengadopsi mekanisme seleksi dan informasi genetik alami. Dalam penggunaannya, algoritma genetik meniru beberapa proses yang ditemukan pada evolusi alamiah. Kemudian algoritma yang telah dikembangkan digunakan untuk menyelesaikan masalah penjadwalan pada *no-wait flowshop* dengan kriteria minimasi total waktu tinggal aktual. Untuk menentukan urutan pekerjaan dibangkitkan populasi yang terdiri dari kromosom-kromosom. Dalam setiap kromosom yang dibangkitkan didalamnya terdapat *gen-gen* yang mewakili urutan pekerjaan. Tahap-tahap dari algoritma genetik untuk model *JCFN* adalah sebagai berikut :

Tahap Inisialisasi

Melakukan pembangkitan kromosom-kromosom sebagai populasi awal. Representasi kromosom berbasis pekerjaan, sehingga kromosom-kromosom yang dibangkitkan adalah mengkodekan urutan pekerjaan yang dikerjakan.

Tahap Penentuan Model Fungsi Evaluasi

Tujuan dari penyelesaian kasus yang dihadapi adalah minimasi total waktu tinggal aktual. Untuk mendapatkan total waktu tinggal aktual, digunakan model penjadwalan pada *no-wait flowshop*. Konsep yang dipakai untuk menyelesaikan permasalahan yang dihadapi adalah konsep minimasi, hal ini berlawanan dengan konsep yang dipakai dalam algoritma genetik. Karena di dalam algoritma genetik, konsep utamanya adalah mencari kromosom yang mempunyai nilai suaian (*fitness value*) yang besar. Sehingga fungsi tujuan yang dipakai untuk mencari nilai kesesuaian setiap kromosom dalam algoritma genetik adalah merupakan fungsi invers dari total waktu tinggal aktual.

Tahap Penentuan Kromosom Induk.

Setelah ditentukan dasar representasi kromosom dan fungsi yang digunakan untuk mencari nilai kesesuaian setiap kromosom, kemudian ditentukan kromosom induk. Dari kromosom awal, yang bertindak sebagai generasi awal, kemudian dilakukan reproduksi dengan memperhatikan nilai kesesuaian masing-masing kromosom. Kromosom-kromosom yang mempunyai nilai kesesuaian yang tinggi mempunyai kesempatan untuk diproduksi dan diduplikasi lebih banyak dari kromosom-kromosom yang mempunyai nilai kesesuaian yang rendah. Metode ini dikenal dengan metode seleksi induk roda rolet (*roulette wheel parents selection*).

Tahap Operasi genetik

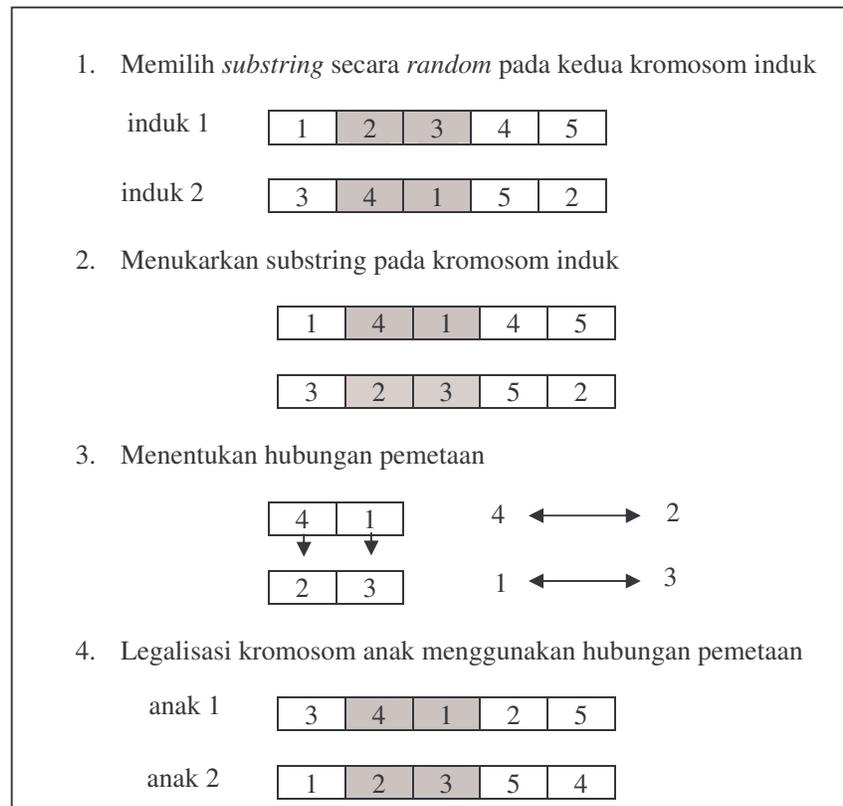
Dalam penelitian ini operasi genetik yang digunakan adalah persilangan (*crossover*) dan mutasi.

a. Persilangan

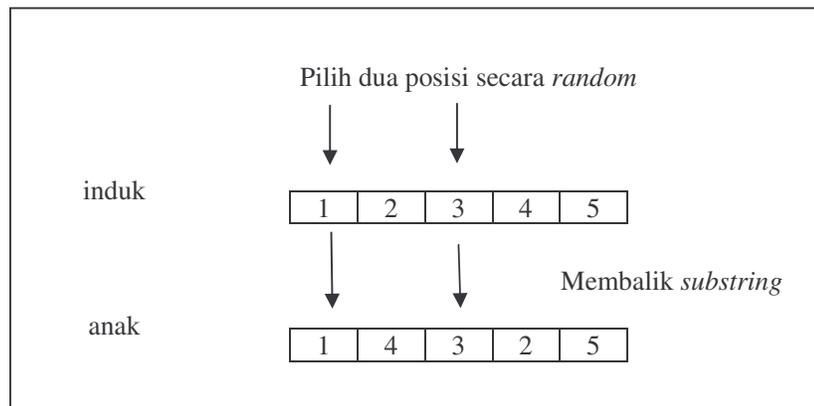
Induk-induk yang akan melakukan persilangan dipilih secara *random*. Apabila jumlah induk yang terpilih untuk melakukan persilangan adalah genap, maka langsung dilakukan pemasangan induk-induk tadi secara *random* untuk melakukan persilangan. Apabila jumlah induk-induk yang akan melakukan persilangan adalah ganjil, maka induk yang dengan nilai suaian terkecil dihapus dari populasi dan kemudian baru dilakukan pemasangan secara *random* untuk melakukan persilangan. Metode persilangan yang dipakai dalam penelitian ini adalah *Partial-Mapped Crossover (PMX)*. Mekanisme dari metode persilangan ini adalah dengan memilih dua posisi (*substring*) yang akan disilangkan pada kedua kromosom induk secara *random*, kemudian menukarkan dua *substring* antara kedua induk. Langkah berikutnya menentukan hubungan pemetaan antara dua *substring* yang dipetakan dan kemudian melakukan legalisasi kromosom anak menggunakan hubungan pemetaan, mekanisme metode persilangan ini dapat dilihat pada gambar 1.

b. Mutasi

Induk-induk yang terpilih dari generasi awal sebagai hasil dari reproduksi generasi awal siap pula terkena mutasi *gen*. Pemilihan kromosom-kromosom yang akan terkena mutasi dilakukan secara *random*. Aturan mutasi *gen* dari kromosom-kromosom yang terkena mutasi yang digunakan dalam penelitian ini adalah mengikuti aturan mutasi inversi (*inversion mutation*). Mekanisme dari aturan ini adalah memilih dua posisi di dalam kromosom secara *random*, kemudian membalik *gen-gen* yang berada diantara dua posisi tersebut seperti pada gambar 2.



Gambar 1. Mekanisme partial-mapped crossover



Gambar 2. Mekanisme mutasi inversi

Tahap Seleksi

Dari tahap operasi genetik didapat kromosom-kromosom anak, karena ruang *sampling* yang digunakan dalam penelitian ini adalah ruang *sampling* yang diperluas, maka kromosom-kromosom anak yang dihasilkan kemudian dibawa ke dalam ruang *sampling* yang sebagian sudah ditempati oleh kromosom-kromosom induk, yang kemudian akan dilakukan seleksi sebanyak ukuran populasi (Z) kromosom terbaik.

4. Analisis

Untuk mengetahui performansi dari algoritma genetik yang telah dikembangkan untuk model *JCFN* maka dilakukan pengujian untuk jumlah pekerjaan sedikit yaitu 5, 6, 7, 8 dengan jumlah mesin 5, 10, 15, 20 dan pekerjaan banyak yaitu 10, 20, 30, 40 dengan jumlah mesin 5, 10, 15. Pada kasus jumlah pekerjaan sedikit performansi didapat dengan menghitung *mean error* dari algoritma genetik, solusi optimal dapat diperoleh melalui enumerasi total sedangkan untuk jumlah pekerjaan banyak performansi dihitung dengan menggunakan ukuran *mean deviation* dari solusi algoritma pasangan pekerjaan. Data waktu proses dan waktu *setup* dibangkitkan secara *random* dari distribusi *uniform* dengan *range* antara 1 dan 10 dan *range* antara 1 dan 100 dan dilakukan 50 replikasi dengan mengubah *seed* dalam pembangkit bilangan *random*. Data diselesaikan dengan $d = 10000$, $Z = 20$, $g = 100$, $P_c = 0,4$, dan $P_m = 0,2$. Penentuan P_c dan P_m yang terlalu besar akan mengakibatkan pemborosan waktu komputasi dan jika terlalu kecil akan menghilangkan peluang kromosom-kromosom yang baik untuk melakukan operasi genetik (Gen dan Cheng, 1997).

Analisis Performansi Algoritma Genetik untuk Model *JCFN* pada Kasus Pekerjaan Sedikit

1. Semakin banyak jumlah pekerjaan maka semakin besar pula *mean error* solusi algoritma genetik untuk *JCFN*, dan sebaliknya semakin sedikit jumlah pekerjaan maka semakin kecil *mean error* solusi model *JCFN*.
2. Semakin banyak jumlah mesin maka *mean error* solusi algoritma genetik untuk *JCFN* cenderung semakin kecil, dan sebaliknya semakin sedikit jumlah mesin maka *mean error* solusi algoritma genetik untuk *JCFN* cenderung semakin besar untuk *range* bilangan *random* $U(1,10)$.

3. Semakin besar *range* bilangan *random* yang dibangkitkan maka cenderung semakin besar pula *mean error* solusi model *JCFN*, dan sebaliknya semakin kecil *range* bilangan *random* yang dibangkitkan maka cenderung semakin kecil pula *mean error* solusi model *JCFN*.

Analisis Performansi Algoritma Genetik untuk Model *JCFN* pada Kasus Pekerjaan Banyak

1. Semakin besar jumlah pekerjaan maka *mean deviation* semakin kecil, dan sebaliknya semakin kecil jumlah pekerjaan semakin besar *mean deviation*
2. Semakin besar *range* bilangan *random* yang dibangkitkan maka cenderung semakin besar pula *mean deviation* solusi model *JCFN*, dan sebaliknya semakin kecil *range* bilangan *random* yang dibangkitkan maka cenderung semakin kecil pula *mean deviation* solusi model *JCFN*.
3. Semakin besar jumlah mesin maka *mean deviation* cenderung semakin besar dan sebaliknya semakin kecil jumlah mesin maka cenderung semakin besar pula *mean deviation*.

5. Kesimpulan

Kesimpulan yang dapat diambil dari hasil pembahasan di atas adalah

1. Algoritma genetik dapat digunakan untuk menyelesaikan permasalahan penjadwalan pada *no-wait flowshop* dengan kriteria minimasi waktu tinggal aktual.
2. Dengan pengembangan algoritma genetik dapat meningkatkan performansi dari algoritma pasangan pekerjaan yang telah dikembangkan sebelumnya dalam pencarian solusi yang mendekati optimal.
3. Pada algoritma genetik untuk model *JCFN*, solusi yang mendekati optimal yang didapat dipengaruhi oleh besarnya penentuan ukuran populasi dan generasi yang dibangkitkan.
4. Algoritma genetik yang dikembangkan tidak menjamin bahwa solusi optimal akan didapatkan, akan tetapi dari penelitian yang telah dilakukan, solusi menunjukkan *mean error* yang kecil dibandingkan algoritma yang dikembangkan sebelumnya.

Beberapa hal yang dapat ditindaklanjuti dari hasil penelitian antara lain :

1. Pengembangan dengan memperbaiki langkah-langkah pemrograman komputer untuk mendapatkan solusi yang mendekati optimal dengan waktu komputasi yang lebih pendek.
2. Karena algoritma yang dikembangkan tidak menjamin bahwa solusi akan optimal dan model penjadwalan pada *no-wait flowshop* dengan $m > 2$ merupakan masalah *NP-hard*, maka perlu dikembangkan algoritma heuristik untuk pengurutan pekerjaan yang dapat menghasilkan solusi dengan kualitas lebih baik.

Daftar Pustaka

- Aldowaisan, T. dan Allahverdi, A.,(1998), "Total flowtime in no-wait flowshops with separated setup times", *Computers Ops. Res.*, 25(9), pp.757-763.
- Allahverdi, A. and Aldowaisan, T., (2000), "No-wait and separate setup three-machine flowshop with total completion time criterion", *Intl. Trans. in Opl.Res.*, 7, pp.245-264.
- Baker, K. R., (1974), *Introduction to Sequencing and Scheduling*, John Wiley and Sons, New York.
- Davis, L., (1991), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.
- French, S., (1982), *Sequencing and Scheduling: An Introduction to the Mathematics of Job-Shop*, Ellis Horwood Limited, Chichester.

- Halim, A. H. dan Ohta, H., (1993), "Batch-scheduling problems through flow shop with both receiving and delivery just in time", *Int. J. Prod. Res.*, 31 , pp.1943-1955.
- Halim, A. H. dan Ohta, H., (1994), "Batch-scheduling problems to minimize inventory cost in the shop with both receiving and delivery just in time", *Int. J. Prod. Eco.*,33 , pp.185-195.
- Gen, Mitsuo dan Cheng, R., (1997), *Genetic Algorithms & Engineering Design*. A Wiley-Interscience Publication. John Wiley & Sons, Inc.
- Goldberg, D., (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison – Wesley, Reading, MA.
- Kamoun, H. dan Sriskandarajah, C.,(1993), "The complexity of scheduling jobs in repetitive manufacturing systems", *Eur. J. Opl. Res.*,70, pp.350-364.
- Morton, T.E dan Pentico, D.W., (1993), *Heuristic Scheduling System: with Application to Production System and Project Management*, John Wiley & Sons.
- Sriskandarajah, C. dan Ladet, P., (1986), "Some no-wait shops scheduling problems: Complexity aspect", *Eur. J. Opl. Res.*,24, pp.424-438
- Van Der Veen, J. A. A. dan van Dal, R., (1991), "Solvable Cases of The No-wait Flow-shop Scheduling Problem", *J. Opl. Res.* 42(11), pp. 971-980.
- Yuniaristanto, (2001), "Model Penjadwalan *Batch* pada *No-Wait Flowshop* dengan Kriteria Waktu Tinggal Aktual", Thesis, Program Pasca Sarjana Program Studi Teknik dan Manajemen Industri, ITB, Bandung.