

Model Penjadwalan Pekerjaan pada *Zero-buffer Flowshop* Tipe (1) dengan Kriteria Minimasi Total Waktu Tinggal Aktual

Yuniaristanto*

Jurusan Teknik Industri, Universitas Sebelas Maret, Surakarta

Abstract

This paper proposes a scheduling model to determine a sequence of the jobs processed in zero-buffer flowshop with separated setup times. In this case, the setup for the next job on current machine can start as soon as this machine completes its processing of the current job. The objective for the model is to minimize total actual flow time accommodating the condition that jobs to be processed arrive at the shop at any times when needed and the condition that the completed jobs be delivered at a common due-date. A heuristic algorithm searching the best solution from the resulting feasible solution has been developed. The numerical experience showing the characteristic of the problem is also presented.

Keywords : *zero-buffer flowshop, scheduling, actual flowtime.*

1. Pendahuluan

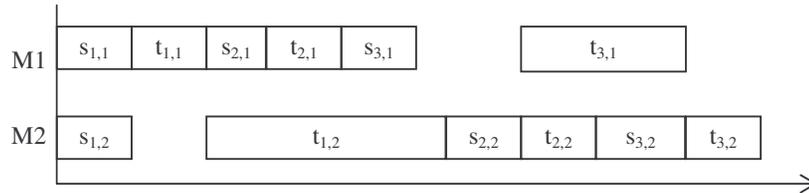
Literatur-literatur yang membahas penjadwalan pekerjaan (*job*) pada sistem produksi *flowshop*, antara lain penelitian Chan dan Bedworth (1990), Rajendran dan Chaudhuri (1991) dan Woo dan Yin (1998). Penelitian-penelitian ini masih mengasumsikan bahwa seluruh pekerjaan telah siap pada saat $t=0$ dan belum mempertimbangkan *due-date* sebagai pembatas. Pada kenyataannya, pekerjaan tidak perlu datang seluruhnya pada saat $t=0$ tetapi dapat datang pada saat pekerjaan tersebut dibutuhkan. Demikian juga, pengiriman pekerjaan ke konsumen tidak dapat dilakukan setiap saat tetapi harus pada saat *due-date*.

Halim dan Ohta (1993, 1994) memperbaiki kekurangan kriteria waktu tinggal (*flow time*) dengan mengusulkan kriteria waktu tinggal aktual (*actual flow time*), yaitu lamanya pekerjaan berada di dalam pabrik dari saat mulai proses sampai *due-date*. Kriteria baru ini digunakan untuk model yang diusulkan agar dapat mengakomodasikan kondisi yang lebih realistis dan konsep tepat waktu, baik tepat waktu kedatangan material maupun tepat waktu pengiriman.

Yuniaristanto (2002) telah mengembangkan suatu model penjadwalan pekerjaan pada *flowshop* statis m mesin untuk kasus *common due-date* dengan kriteria minimasi total waktu tinggal aktual. Model ini telah mempertimbangkan waktu *setup* yang terpisah dari waktu proses dan mengakomodasikan konsep tepat waktu, baik tepat waktu kedatangan material maupun tepat waktu pengiriman. Akan tetapi penelitian ini masih mengijinkan adanya *buffer* diantara kedua mesinnya. *Buffer* diperlukan untuk menyimpan sementara pekerjaan yang telah dikerjakan, sehingga mesin tersebut dapat langsung mengerjakan pekerjaan berikutnya tanpa harus menunggu mesin berikutnya kosong atau menganggur.

* E-mail: utan@uns.ac.id

Memperhatikan uraian di atas, perlu dikembangkan suatu model penjadwalan pekerjaan pada *flowshop* statis m mesin untuk kasus *zero-buffer* dengan kriteria minimasi total waktu tinggal aktual. Pada penelitian ini, dipertimbangkan *zero-buffer* tipe (1) yaitu waktu *setup* boleh dilakukan sebelum pekerjaan meninggalkan mesin tersebut seperti pada gambar 1 (Aldowaisan dan Allahverdi, 1998).



Gambar 1. Waktu *setup* boleh dilakukan sebelum pekerjaan meninggalkan mesin 1

Asumsi-asumsi yang digunakan adalah :

1. Setiap *stage* diasumsikan hanya terdiri dari satu mesin.
2. Waktu proses pekerjaan dan waktu *setup* pada mesin diketahui dan tetap.
3. Waktu *setup* mesin terpisah dari waktu proses dan *independent*.
4. Sistem yang diteliti bersifat statis-deterministik.
5. Pemrosesan pekerjaan tidak dapat diganggu (interupsi).
6. Mesin selalu tersedia.
7. Tidak ada mesin yang memproses lebih dari satu operasi pada saat sama.

2. Pengembangan Model

Pada model Yuniaristanto (2002) belum mempertimbangkan kendala *zero-buffer* tipe (1) yaitu waktu *setup* boleh dilakukan sebelum pekerjaan meninggalkan mesin tersebut. Model berikut menggambarkan kondisi penjadwalan pekerjaan pada *zero-buffer flowshop* yang meminimasi total waktu tinggal aktual. Formulasi matematis model penjadwalan pekerjaan pada *zero-buffer-flowshop* (ZBF) dapat dirumuskan sebagai berikut :

$$\text{Minimasi } F^a = \sum_{i=1}^n (d - B_{[i,1]}) \quad (3)$$

Pembatas :

$$B_{[n,1]} \geq 0, \quad (4)$$

$$B_{[i,m]} = \min \{ B_{[i-1,m]} - t_{[i,m]} - s_{[i-1,m]}, B_{[i-1,m]} - t_{[i-1,m-1]} \} \quad i = 2, \dots, n, \quad (5)$$

$$B_{[i,k]} = \min \{ B_{[i-1,k]} - t_{[i,k]} - s_{[i-1,k]}, B_{[i,k+1]} - t_{[i,k]} \}, \quad i = 2, \dots, n; \quad k = 1, \dots, m-1, \quad (6)$$

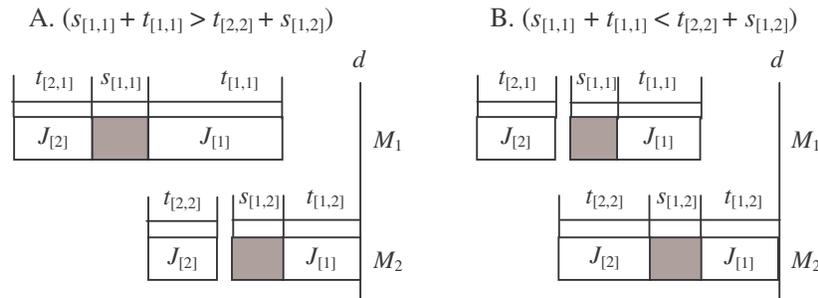
$$B_{[1,k]} = B_{[1,k+1]} - t_{[1,k]}, \quad k = 1, \dots, m-1, \quad (7)$$

$$B_{[1,m]} = d - t_{[1,m]}, \quad (8)$$

Persamaan (3) menyatakan minimasi total waktu tinggal aktual yang merupakan jumlah waktu tinggal aktual semua pekerjaan di mesin M_1 . Pembatas (4) menyatakan bahwa saat mulai pekerjaan yang diproses pertama kali, J_n di mesin M_1 lebih besar atau sama dengan nol. Pembatas (5), (6) dan (7) digunakan untuk menyusun jadwal produksi dengan pendekatan penjadwalan mundur dan saat mulai proses stasiun kerja hilir mengendalikan saat mulai proses stasiun kerja hulu. Pembatas (8) menyatakan bahwa pekerjaan terakhir J_1 harus selesai tepat pada saat *due-date*.

Model penjadwalan pekerjaan pada *flowshop* dengan m mesin merupakan masalah *NP-hard* (French, 1982), sehingga perlu dikembangkan pendekatan heuristik untuk menjadwalkan semua pekerjaan. Formulasi matematis model *ZBF* digunakan untuk menjadwalkan urutan pekerjaan terpilih yang dapat meminimasi total waktu tinggal aktual.

Untuk mengurutkan semua pekerjaan di sistem produksi *flowshop*, digunakan prosedur heuristik yang dikembangkan oleh Yuniaristanto (2002). Prosedur ini merupakan modifikasi dari prosedur yang dikembangkan Chan dan Bedworth (1990). Modifikasi dilakukan dengan dasar pendekatan penjadwalan mundur dimulai dari *due-date* ke arah $t=0$ dan menggunakan persamaan pembatas untuk memenuhi kondisi model tersebut. Penyelesaian model akan dimulai pada persoalan penjadwalan dua pekerjaan yang merupakan model dasar pada masalah *sequencing* dan mengamati variasi karakteristiknya. Pada saat dua pekerjaan diproses pada semua mesin dengan urutan sama, maka keputusan penjadwalannya adalah pekerjaan mana yang harus dijadwalkan pertama kali untuk meminimasi waktu tinggal aktual. Persoalan ini dapat diterapkan pada dua, tiga, empat sampai m mesin.



Gambar 3. Gantt chart pasangan pekerjaan untuk 2 mesin

Misalkan dua pekerjaan, yaitu $J_{[1]}$ dan $J_{[2]}$ diproses di dua mesin, M_1 dan M_2 . Dua alternatif keadaan yang dapat terjadi jika $J_{[2]}$ diproses sebelum $J_{[1]}$ dapat dilihat pada gambar 2. Jika dua pekerjaan tersebut diproses di tiga mesin, maka ada tiga alternatif keadaan yang dapat terjadi dan jika diproses di empat mesin, maka ada empat alternatif keadaan yang dapat terjadi. Waktu tinggal aktual pasangan pekerjaan dapat dilihat pada tabel 1. Tabel 1 juga menunjukkan waktu tinggal aktual dari pasangan pekerjaan yang diproses pada tiga mesin dan empat mesin.

Tabel 1. Penghitungan waktu tinggal aktual pasangan pekerjaan

M	c	F^a job 1	F^a job 2	F^a pasangan job
2	1	$t_{[1,1]}+t_{[1,2]}$	$t_{[2,1]}+s_{[1,1]}+t_{[1,1]}+t_{[1,2]}$	$t_{[1,1]}+2t_{[1,2]}+t_{[2,1]}+\max$
	2	$t_{[1,1]}+t_{[1,2]}$	$t_{[2,1]}+t_{[2,2]}+s_{[1,2]}+t_{[1,2]}$	$\{s_{[1,1]}+t_{[1,1]}, t_{[2,2]}+s_{[1,2]}\}$
3	1	$t_{[1,1]}+t_{[1,2]}+t_{[1,3]}$	$t_{[2,1]}+s_{[1,1]}+t_{[1,1]}+t_{[1,2]}+t_{[1,3]}$	$t_{[1,1]}+t_{[1,2]}+2t_{[1,3]}+t_{[2,1]}+\max$
	2	$t_{[1,1]}+t_{[1,2]}+t_{[1,3]}$	$t_{[2,1]}+t_{[2,2]}+s_{[1,2]}+t_{[1,2]}+t_{[1,3]}$	$[t_{[1,2]}+\max\{s_{[1,1]}+t_{[1,1]},$
	3	$t_{[1,1]}+t_{[1,2]}+t_{[1,3]}$	$t_{[2,1]}+t_{[2,2]}+t_{[2,3]}+s_{[1,3]}+t_{[1,3]}$	$t_{[2,2]}+s_{[1,2]}, t_{[2,2]}+t_{[2,3]}+s_{[1,3]}\}$
4	1	$t_{[1,1]}+t_{[1,2]}+t_{[1,3]}+t_{[1,4]}$	$t_{[2,1]}+s_{[1,1]}+t_{[1,1]}+t_{[1,2]}+t_{[1,3]}+t_{[1,4]}$	$t_{[1,1]}+t_{[1,2]}+t_{[1,3]}+2t_{[1,4]}+t_{[2,1]}+$
	2	$t_{[1,1]}+t_{[1,2]}+t_{[1,3]}+t_{[1,4]}$	$t_{[2,1]}+t_{[2,2]}+s_{[1,2]}+t_{[1,2]}+t_{[1,3]}+t_{[1,4]}$	$\max < t_{[1,3]}+\max[t_{[1,2]}+\max$
	3	$t_{[1,1]}+t_{[1,2]}+t_{[1,3]}+t_{[1,4]}$	$t_{[2,1]}+t_{[2,2]}+t_{[2,3]}+s_{[1,3]}+t_{[1,3]}+t_{[1,4]}$	$\{s_{[1,1]}+t_{[1,1]}, t_{[2,2]}+s_{[1,2]},$
	4	$t_{[1,1]}+t_{[1,2]}+t_{[1,3]}+t_{[1,4]}$	$t_{[2,1]}+t_{[2,2]}+t_{[2,3]}+t_{[2,4]}+s_{[1,4]}+t_{[1,4]}$	$t_{[2,2]}+t_{[2,3]}+s_{[1,3]}, t_{[2,2]}+t_{[2,3]}$ $+t_{[2,4]}+s_{[1,4]}>$

Dari tabel 1 dapat diperoleh rumusan umum waktu tinggal aktual pasangan pekerjaan $J_{[1]}$ dan $J_{[2]}$ yang diproses pada m mesin yaitu

$$F^a = t_{[2,1]} + \sum_{k=1}^m t_{[1,k]} + t_{[1,m]} + R_m, \quad m \geq 2, \quad (9)$$

dengan: $R_1 = s_{[1,1]}$

$$R_m = \max \{ t_{[1,m-1]} + R_{m-1}, \sum_{k=2}^m t_{[2,k]} + s_{[1,m]} \}$$

Jika $J_{[1]}$ diproses $J_{[2]}$, maka indeks 1 dan 2 dipertukarkan.

Penghitungan waktu tinggal aktual pasangan pekerjaan pada kasus penjadwalan *zero-buffer flowshop* sama seperti pada kasus penjadwalan *flowshop* dengan *buffer*. Akan tetapi pada penentuan jadwal produksi pekerjaan akan berbeda hasilnya karena adanya perbedaan kendala-kendala yang membatasi sistem penjadwalan tersebut (lihat Yuniaristanto, 2002).

3. Algoritma Penjadwalan ZBF

Untuk menjadwalkan n pekerjaan pada m mesin, maka digunakan persoalan penjadwalan dua pekerjaan pada m mesin sebagai dasar penjadwalan. Kemudian dikembangkan suatu algoritma agar persoalan penjadwalan n pekerjaan pada m mesin dapat diselesaikan sehingga dapat meminimasi total waktu tinggal aktual. Algoritma berikut diambil dari algoritma yang dikembangkan oleh Yuniaristanto (2002).

Untuk menentukan urutan pekerjaan digunakan persamaan (9) pada setiap pasangan pekerjaan yang mungkin. Kemudian pilih pekerjaan awal dari pasangan pekerjaan yang mempunyai waktu tinggal aktual terkecil. Jadwalkan secara mundur pekerjaan menurut *decreasing order* dari frekuensi suatu pekerjaan terpilih. Problem ZBF layak jika $B_{[n,1]} \geq 0$. Algoritma penjadwalan ZBF dapat dijelaskan sebagai berikut :

Langkah 0. Inisialisasi dengan menentukan $n, d, m, s_{[i,k]}, t_{[i,k]}$.

Langkah 1. Hitung jumlah pasangan pekerjaan $(R) = n(n-1)/2$. Tiap pasangan pekerjaan mempunyai dua urutan yang mungkin.

Langkah 2. Set $f = 1$ dan frekuensi pekerjaan terpilih $(a_i) = 0, \forall i$

Langkah 3. Untuk tiap pasangan f yang mungkin, hitung waktu tinggal aktual pasangan pekerjaan di dalam *shop* dengan menggunakan persamaan (9). Pilih urutan dengan waktu tinggal aktual terkecil. Pilih pekerjaan awal dari urutan yang terpilih. Set $a_i = a_i + 1$.

Langkah 4. Jika $f < R$ maka set $f = f + 1$ dan kembali ke Langkah 3. Selainnya, lanjutkan ke Langkah 5.

Langkah 5. Jadwalkan secara mundur pekerjaan menurut *decreasing order* dari $a_i, \forall i$ dengan menggunakan persamaan (5), (6), (7), dan (8). Kemudian hitung total waktu tinggal aktual pekerjaan dengan menggunakan persamaan (3).

Langkah 6. Jika tidak ada solusi untuk $B_{[n,1]} \geq 0$, maka problem tak layak dan *stop*. Selainnya, pilih solusi dengan total waktu tinggal aktual minimum.

4. Contoh Numerik

Contoh numerik ini bertujuan untuk mengetahui karakteristik model penjadwalan ZBF. Untuk melihat karakteristik model digunakan data yang meliputi data waktu proses, waktu *setup*, *due-date*, jumlah *job* dan jumlah mesin seperti pada tabel 2 dengan *due-date* = 200. Data dalam contoh numerik ini sama dengan data pada penelitian Yuniaristanto (2002) dengan tujuan untuk menunjukkan perbedaan antara kasus penjadwalan *zero-buffer flowshop* dan penjadwalan *flowshop* dengan *buffer*.

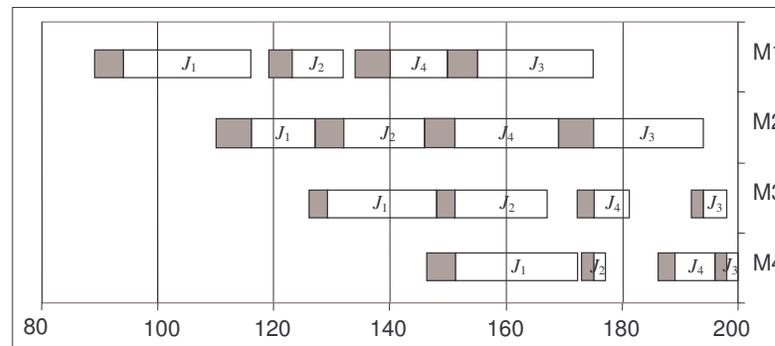
Tabel 2. Waktu proses dan waktu *setup* pekerjaan

M_k	$s_{[1,k]}$	$t_{[1,k]}$	$s_{[2,k]}$	$t_{[2,k]}$	$s_{[3,k]}$	$t_{[3,k]}$	$s_{[4,k]}$	$t_{[4,k]}$
1	5	22	4	9	5	20	6	10
2	6	11	5	14	6	19	5	18
3	3	19	3	16	2	4	3	6
4	5	21	2	2	2	2	3	7

Untuk mendapatkan solusi model penjadwalan ZBF dijalankan algoritma penjadwalan ZBF. Solusi model ZBF menunjukkan bahwa urutan pekerjaan yang dihasilkan mulai dari *due-date* adalah 3 – 4 – 2 – 1 (jika mulai dari saat nol adalah 1 – 2 – 4 – 3). Kemudian urutan pekerjaan tersebut dijadwalkan dengan menggunakan pembatas (5), (6), (7), dan (8) dan hasil perhitungannya dapat dilihat pada tabel 3. Setelah semua pekerjaan dijadwalkan, maka diperoleh total waktu tinggal aktual = 288. Hasil penjadwalan model penjadwalan ZBF dapat dilihat pada gambar 3.

Tabel 3. Hasil penghitungan $B_{[i,k]}$ model penjadwalan ZBF

$i \backslash k$	1	2	3	4
1	155	175	194	198
2	140	151	175	189
3	123	132	151	175
4	94	116	129	151



Gambar 3. Gantt chart solusi model penjadwalan ZBF

5. Kesimpulan

Dengan contoh numerik seperti pada Yuniaristanto (2002) diperoleh hasil urutan pekerjaan sama untuk model *ZBF* dan *JCF* dan menghasilkan total waktu tinggal aktual yang sama pula. Akan tetapi saat mulai tiap pekerjaan pada tiap mesin antara model *ZBF* dan *JCF* berbeda. Hal ini disebabkan adanya batasan dimana tidak diperbolehkan adanya *buffer* diantara mesin-mesin.

Pada model penjadwalan *ZBF* tidak ada satupun pekerjaan yang *tardy* karena menggunakan pendekatan penjadwalan mundur. Algoritma penjadwalan *ZBF* tidak menjamin bahwa solusi akan optimal karena urutan optimal untuk $n-1$ pekerjaan tidak selalu memberikan titik awal yang baik untuk masalah n pekerjaan. Perbandingan kualitas solusi tidak dapat dilakukan karena tidak adanya model pembandingan.

Daftar Pustaka

- Aldowaisan, T. and Allahverdi, A. (1998). Total flowtime in no-wait flowshops with separated setup times. *Computers Ops. Res.*, 25(9), pp.757-765.
- Chan, D. Y. and Bedworth, D. D. (1990). Design of a scheduling system for flexible manufacturing cell. *Int. J. Prod. Res.*, 28, pp.2037-2049.
- French, S. (1982). *Sequencing and Scheduling: An Introduction to the Mathematics of Job-Shop*. Chichester: Ellis Horwood Limited.
- Halim, A. H. and 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. & 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.
- Rajendran, C. and Chaudhuri, D. (1991). An efficient heuristic approach to the scheduling of jobs in a flowshop. *Eur. J. Opl. Res.*, 61, pp.318-325.
- Woo, H. S. and Yim, D. S. (1998). A heuristic algorithm for mean flowtime objective in flowshop scheduling. *Computers Ops. Res.*, 25(3), pp.175-182.
- Yuniaristanto (2002). Model Penjadwalan Pekerjaan pada *Flowshop* dengan Kriteria Minimasi Total Waktu Tinggal Aktual. *Performa*, 1(1), pp.20-25.