

Penjadwalan Pekerjaan pada *No-Wait Flowshop* dengan Pembatas *Common Due-Date*

Yuniaristanto*

Jurusan Teknik Industri, Universitas Sebelas Maret, Surakarta

Abstract

This paper addresses job-scheduling problems in no-wait flowshop. No-wait flowshop gives condition that each job processed without any interruptions on machine or between two machines. Criteria of this problems is to minimize total actual flowtime that accommodate condition that jobs arrive in the shop when they needed and deliver them on due-date. Heuristic algorithm is developed to find feasible solutions. An numerical example is also provided to show how the proposed method work. Example results shows total actual flowtime in no-wait flowshop higher than total actual flowtime in general flowshop..

Keywords : *scheduling, no-wait flowshop, total actual flowtime.*

1. Pendahuluan

Permasalahan penjadwalan pekerjaan (*job*) pada sistem produksi *flowshop* telah banyak diteliti antara lain oleh Chan dan Bedworth (1990), Rajendran dan Chaudhuri (1991), dan Woo dan Yin (1998). Tetapi model-model tersebut masih mengijinkan adanya *delay* di antara dua mesin sedangkan pada industri baja, plastik dan kimia tidak diijinkan adanya *delay* di antara dua mesin. Masalah ini dikenal sebagai *no-wait flowshop*. Kondisi *no-wait* terjadi apabila proses pekerjaan tidak dapat diganggu baik saat berada di suatu mesin atau saat transfer ke mesin berikutnya. Hal ini berarti bahwa saat selesai pekerjaan di suatu mesin harus sama dengan saat mulai proses pekerjaan pada mesin berikutnya. Aldowaisan dan Allahverdi (1998) melakukan penelitian pada *no-wait flowshop* untuk dua mesin dengan kriteria *total flow time*, dan waktu *setup* dilihat terpisah dari waktu proses. Kemudian Allahverdi dan Aldowaisan (2000) mengembangkan untuk tiga mesin dengan kriteria *total completion time*.

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 sedangkan pengiriman pekerjaan ke konsumen dilakukan pada saat *due-date*. 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*.

Sehingga perlu dikembangkan suatu model penjadwalan pekerjaan pada no-wait flowshop statis dengan kriteria minimasi total waktu tinggal aktual. Setiap stage diasumsikan hanya terdiri dari satu mesin. Asumsi-asumsi lain yang digunakan adalah :

1. Waktu proses pekerjaan pada mesin diketahui dan tetap.

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

2. Waktu *setup* mesin diketahui dan terpisah dari waktu proses.
3. Sistem bersifat deterministik.
4. Mesin selalu tersedia.
5. Tidak ada mesin yang memproses lebih dari satu operasi pada saat sama.
6. Tidak ada pekerjaan yang diproses oleh lebih dari satu mesin pada saat sama.

2. Pengembangan Model

Model ini menggambarkan kondisi penjadwalan pekerjaan pada *no-wait flowshop* yang meminimasi total 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 modelnya dapat dirumuskan sebagai berikut :

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

Pembatas :

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

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

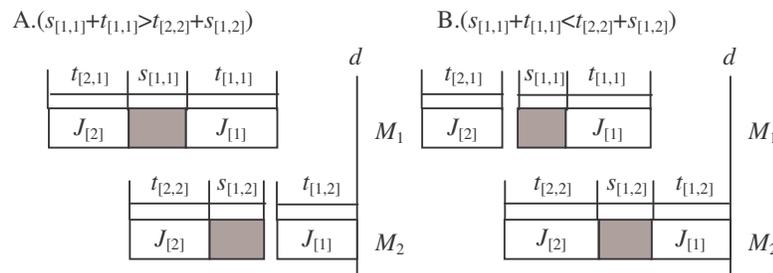
$$B_{[i,k]} = \min\{B_{[i-1,k]} - t_{[i,k]} - s_{[i-1,k]}, B_{[i,k+1]} - t_{[i,k]}\}, \quad i = 2, \dots, n; \quad k = 1, \dots, m-1, \quad (4)$$

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

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

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*.

Penjadwalan pada *no-wait flowshop* lebih dari dua mesin merupakan masalah *NP-hard* (Kamoun dan Sriskandarajah, 1993), sehingga dikembangkan pendekatan heuristik. Untuk mengurutkan semua pekerjaan pada *no-wait flowshop*, digunakan sebuah prosedur heuristik yang dikembangkan oleh Halim dan Ohta (1993). 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 $J_{[1]}$, yaitu dapat dilihat pada Gambar 1.



Gambar 1. Gantt chart pasangan pekerjaan untuk 2 mesin (*no-wait*)

Jika dua pekerjaan diproses di tiga mesin, maka ada tiga alternatif keadaan yang terjadi dan jika diproses di empat mesin, maka ada empat alternatif keadaan yang terjadi. Rumusan umum waktu tinggal aktual pasangan pekerjaan $J_{[1]}$ dan $J_{[2]}$ yang diproses pada m mesin yaitu:

$$F^a = t_{[2,1]} + \sum_{k=1}^m t_{[1,k]} + t_{[1,m]} + R_m, \quad m \geq 2, \quad (7)$$

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

$$R_m = \max \{ t_{[1,m-1]} + R_{m-1}, \sum_{k=2}^m t_{[2,k]} + s_{[1,m]} \} \quad (8)$$

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

3. Algoritma Penjadwalan

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.

Untuk menentukan urutan pekerjaan digunakan persamaan (7) 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 penjadwalan layak jika jadwal $B_{[n,1]} \geq 0$. Algoritma penjadwalan model ini dapat dijelaskan sebagai berikut :

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

Langkah 1. Hitung pasangan pekerjaan $(R) = n(n-1)/2$. Tiap pasangan *job* mempunyai dua urutan.

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 dengan persamaan (7). 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, ke Langkah 5.

Langkah 5. Jadwalkan mundur pekerjaan me-nurut *decreasing order* dari $a_i, \forall i$ dengan algoritma H1. Hitung total waktu tinggal aktual pekerjaan dengan persamaan (1).

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

Untuk menentukan total waktu tinggal aktual pada persamaan (1), harus diketahui $B_{[i,1]}$ untuk $\forall i$. Akan tetapi dari semua persamaan pembatas tersebut tidak ada satupun persamaan yang dapat digunakan untuk menghitung $B_{[i,1]}$. Hal ini disebabkan karena pada kasus *no-wait*, saat mulai proses pekerjaan pada suatu mesin dapat ditunda supaya saat selesai proses pekerjaan pada mesin tersebut tepat sama dengan saat mulai proses pekerjaan pada mesin berikutnya sehingga persamaan (5) dapat terpenuhi. Karena itu perlu dibuat algoritma untuk menyederhanakan penghitungan $B_{[i,1]}$, yaitu algoritma H1.

Algoritma H1 dimulai dengan menghitung waktu tinggal aktual dua pekerjaan yang terdekat dengan *due-date*. Kemudian hitung saat mulai kedua pekerjaan tersebut pada mesin M_1 . Proses ini diulang dengan menggeser saat *due-date* ke arah mendekati $t=0$ sampai semua pekerjaan terjadwalkan. Rumusan umum untuk menghitung waktu tinggal aktual dari pekerjaan $J_{[1]}$ dan $J_{[2]}$ jika diproses pada sejumlah m mesin adalah:

$$1. \text{ Waktu tinggal aktual untuk } J_{[1]} = \sum_{k=1}^m t_{[1,k]} \text{ sehingga } B_{[1,1]} = d - \sum_{k=1}^m t_{[1,k]} \quad (9)$$

2. Waktu tinggal aktual untuk $J_{[2]} = t_{[2,1]} + t_{[1,m]} + R_m$, $m \geq 2$,
 dengan: $R_1 = s_{[1,1]}$

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

waktu tinggal aktual ini nantinya selalu diulang untuk $i \geq 2$.

Jika J_i dan J_j berdekatan dan J_j mendahului J_i dan jika d_i adalah *due-date* untuk J_i maka:

$$B_{[j,1]} = d_i - (t_{[j,1]} + t_{[i,m]} + R_m) \tag{10}$$

$$d_j = B_{[j,1]} + \sum_{k=1}^m t_{[j,k]} \tag{11}$$

Algoritma H1

Langkah 1. Hitung $B_{[1,1]}$ dengan persamaan (8).

Langkah 2. Set $i = 1$.

Langkah 3. Set $j = i+1$.

Langkah 4. Hitung $B_{[j,1]}$ dengan persamaan (9).

Langkah 5. Hitung d_j dengan persamaan (10).

Langkah 6. Jika $i < n$ maka $i = i+1$. Kembali ke Langkah 3. Selainnya, lanjutkan ke Langkah 7.

Langkah 7. Hitung $B_{[j,k]}$ dengan persamaan (5)

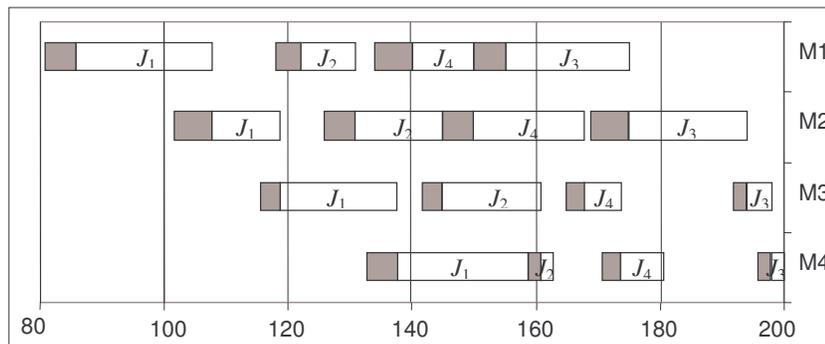
4. Contoh Numerik

Contoh numerik membandingkan model penjadwalan pekerjaan pada *no-wait flowshop* dengan model penjadwalan pekerjaan pada *flowshop* tanpa *no-wait*. Untuk membandingkan kedua model tersebut digunakan data yang sama seperti pada tabel 1 dengan *due-date* = 200.

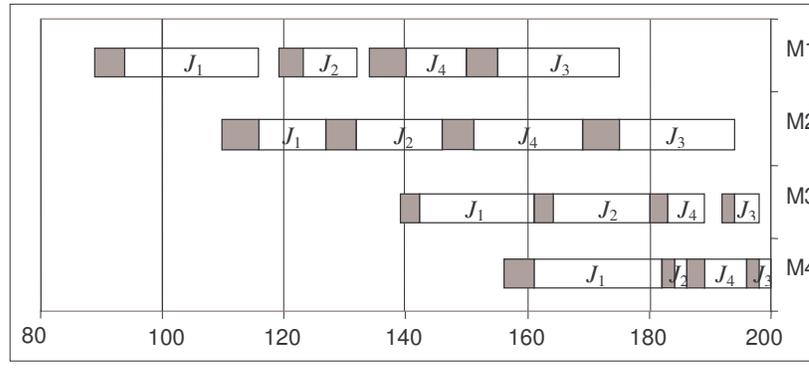
Tabel 1. Waktu proses dan waktu *setup*

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

Solusi model penjadwalan pekerjaan pada *no-wait flowshop* dapat dilihat pada gambar 2 dengan total waktu tinggal aktual = 297. Solusi model penjadwalan pekerjaan tanpa *no-wait* dapat dilihat pada gambar 3 dengan total waktu tinggal aktual = 288.



Gambar 2. Gantt chart solusi model penjadwalan pekerjaan pada *no-wait flowshop*



Gambar 3. Gantt chart solusi model penjadwalan pekerjaan pada *flowshop*

Hasil menunjukkan peningkatan total waktu tinggal aktual model penjadwalan pekerjaan dengan *no-wait* sebesar 3,125% dari solusi model penjadwalan tanpa *no-wait*.

5. Kesimpulan

Model penjadwalan pekerjaan pada *no-wait flowshop* menghasilkan total waktu tinggal aktual yang lebih besar dari model penjadwalan pekerjaan pada *flowshop*. Hal ini disebabkan karena model penjadwalan pekerjaan dengan *no-wait* menghasilkan solusi dengan waktu *idle* yang lebih panjang dari model penjadwalan pekerjaan tanpa *no-wait*. Waktu *idle* ini terjadi karena saat mulai proses pekerjaan pada suatu mesin mungkin ditunda supaya saat selesainya sama dengan saat mulai proses pekerjaan pada mesin berikutnya.

Selanjutnya perlu dikembangkan algoritma heuristik untuk penjadwalan pekerjaan yang dapat menghasilkan solusi dengan performansi yang lebih baik.

Daftar Pustaka

- Aldowaisan, T. and Allahverdi, A. (1998), "Total flowtime in no-wait flowshops with separated setup times", *Comp. 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.
- 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.
- 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. and 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.
- Kamoun, H. and Sriskandarajah, C. (1993), "The complexity of scheduling jobs in repetitive manufacturing systems", *Eur. J. Opl. Res.*,70, pp.350-364.
- 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", *Comp. Ops. Res.*, 25(3), pp.175-182.