

MODEL PENJADWALAN PADA *FLOWSHOP-4-STAGE* DENGAN KRITERIA MINIMISASI *LATENESS* MAKSIMUM DAN JUMLAH *TARDY JOBS*

Alex Saleh, Emsosfi Zaini

Fakultas Teknik Industri, Jurusan Teknik Industri, Institut Teknologi Nasional
Jl. PHH Mustofa No. 23, Bandung 40124
Email: alexaldjaidi@yahoo.com, sosfiez@yahoo.com

ABSTRAK

Penelitian ini membahas model penjadwalan *job* pada *flowshop-4-stage* untuk proses fabrikasi dan perakitan. Setiap *job* terdiri dari 1 *unique component* dan 1 *common component*, keduanya diproses pada tiga mesin pertama secara serial dan dirakit pada mesin keempat. *Unique component* diproses satu per satu sedangkan *common component* diproses dalam *batch* dengan waktu *setup* yang konstan untuk setiap *batch*. Kriteria yang digunakan adalah minimisasi *lateness* maksimum dan jumlah *tardy jobs*. Pengembangan model dilakukan untuk mengakomodasi adanya *job* yang terlambat. Algoritma dimulai dengan proses penjadwalan *job* dengan menggunakan aturan *earliest due date* (EDD) untuk mesin tunggal, dan proses *batching* dengan menggunakan pemrograman dinamis. Jumlah *tardy jobs* diperbaiki dengan Algoritma Hodgson. Algoritma yang diusulkan bukan merupakan algoritma optimal karena proses penjadwalan yang tidak dijamin optimal, meskipun proses *batching* dapat menghasilkan solusi optimal. Penelitian ini dilengkapi dengan contoh numerik untuk menunjukkan perilaku model.

Kata kunci: *flowshop-4-stage*, *unique component*, *common component*, penjadwalan *batch*, *lateness* maksimum, jumlah *tardy jobs*.

ABSTRACT

This research discusses jobs scheduling model on flowshop-4-stage for fabrication and assembly processes. Each job consists of a unique component and a common component. Both the unique and common components are processed on the first three machines and assembled on the fourth machine. The unique components are processed individually, while common components are processed in batches for which respective constant setups are needed. The criteria used for the models are to minimize maximum lateness and number of tardy jobs. The development is executed to accommodate the possibility of tardy jobs. A proposed algorithm is initiated with jobs scheduling using the earliest due date (EDD) rule for the single machine, and batching process using the dynamic programming principle. Number of tardy jobs improved based on Hodgson Algorithm. The proposed algorithm is not an optimal algorithm as we can not guarantee an optimal scheduling process, nevertheless the batching method can bring out an optimal solution. This research also discusses numerical cases to show model behavior.

Keywords: *flowshop-4-stage*, *unique component*, *common component*, *batch scheduling*, *maximum lateness*, *number of tardy jobs*.

1. PENDAHULUAN

Penelitian yang membahas penjadwalan *job* pada sistem produksi *flow shop* telah banyak dilakukan, diantaranya Tansel *et al.* (2001), Xiao dan Li (2002). Permasalahan yang dibahas pada penelitian tersebut hanya masalah penentuan urutan (*sequencing*) tanpa memperhitungkan ukuran

batch. Penelitian lain menggabungkan permasalahan *sequencing* dan *batching* diantaranya Cheng dan Wang (1998), Halim dan Cahya (1999) dan Bukchin *et al.* (2002) menggunakan kriteria meminimalkan *makespan* dan Indrapriyatna *et al.* (2008) meminimalkan total biaya simpan dan biaya kualitas.

Meskipun demikian, semua model yang telah dikemukakan di atas hanya membahas penjadwalan *flowshop* untuk fabrikasi komponen (*parts*) saja dan tidak mempertimbangkan penjadwalan untuk perakitan. Pada kondisi praktis, seringkali suatu perusahaan menerima *job* yang merupakan suatu rakitan dari beberapa komponen. Komponen pembentuk produk bisa terdiri dari komponen yang sama (*common component*) atau komponen yang berbeda (*unique component*).

Di pihak lain, Cheng dan Wang (1999) mengajukan model penjadwalan untuk fabrikasi dan perakitan komponen pada *flowshop-2-stage* untuk meminimalkan *makespan*. Model penjadwalan dengan meminimalkan *makespan* akan efektif apabila saat penyerahan (*due date*) *job* sama. Apabila *due date* setiap *job* berbeda, maka kriteria meminimalkan *makespan* tidak bisa lagi digunakan. Kriteria penjadwalan dengan mempertimbangkan *due date* untuk setiap *job*, contohnya adalah meminimalkan total waktu tinggal aktual, *mean lateness*, *mean tardiness*, *lateness* maksimum dan jumlah *tardy jobs*. Model penjadwalan untuk fabrikasi dan perakitan pada *flowshop-2-stage* dengan kriteria meminimalkan total waktu tinggal aktual telah dikembangkan oleh Halim dan Saleh (2005).

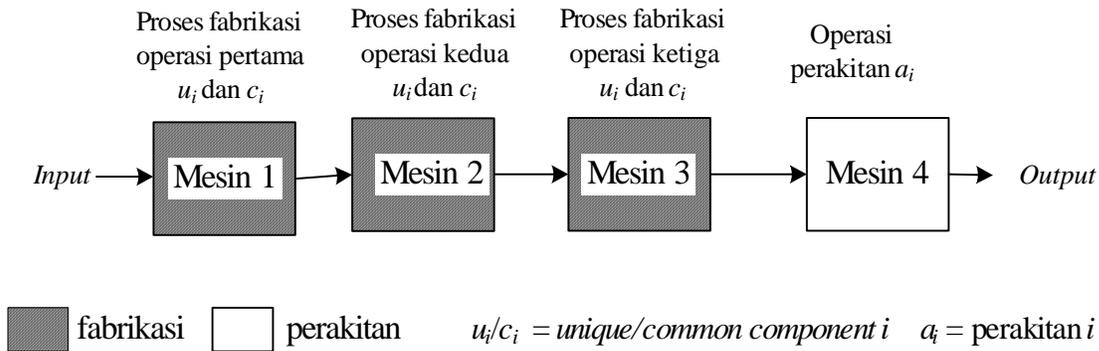
Dalam prakteknya, sistem manufaktur *flowshop* bisa terdiri atas lebih dari 2 mesin. Memperhatikan bahwa untuk mengembangkan model penjadwalan *job* yang merupakan rakitan dari 2 komponen pada *flowshop m* mesin akan menjadi sangat kompleks, maka model yang dibahas dibatasi untuk suatu sistem manufaktur yang terdiri atas 4 mesin. Hal ini didasarkan atas pemikiran Chan dan Bedworth (1990) bahwa suatu *Flexible Manufacturing Cell* (FMC) biasanya terdiri atas 4 mesin.

Memperhatikan hal-hal tersebut di atas, maka terlihat perlunya dikembangkan suatu model penjadwalan di lingkungan FMC yang memproduksi *job* yang memiliki struktur produk dengan kriteria meminimalkan *lateness* maksimum dan jumlah *tardy jobs*.

2. METODE PENELITIAN

2.1 Pengembangan Model

Sistem manufaktur yang dibahas pada penelitian ini merupakan sistem manufaktur *flowshop* yang terdiri dari 4 mesin dan menjalankan kegiatan fabrikasi dan perakitan. Kriteria evaluasi dari model yang dikembangkan adalah meminimalkan *lateness* maksimum dan jumlah *tardy jobs*. Produk yang diproses terdiri atas satu unit *unique component* dan satu unit *common component*. *Unique component* dan *common component* diproses pada tiga mesin pertama secara serial. *Unique component* diproses satu per satu, sedangkan *common component* diproses dalam *batch* dan *setup* dilakukan untuk setiap *batch* dengan waktu *setup batch* konstan. Proses perakitan dilakukan pada mesin keempat dan hanya bisa dilakukan ketika kedua komponen tersedia. Proses fabrikasi *common component* dilakukan dengan cara *batch availability*, yaitu semua *common component* di dalam *batch* tersedia untuk dirakit hanya bila seluruh komponen dalam *batch* selesai diproses. Gambaran sistem produksi pada *flowshop-4-stage* dapat dilihat pada Gambar 1.



Gambar 1. Gambaran sistem produksi *flowshop-4-stage*

Untuk merumuskan model tersebut digunakan notasi berikut:

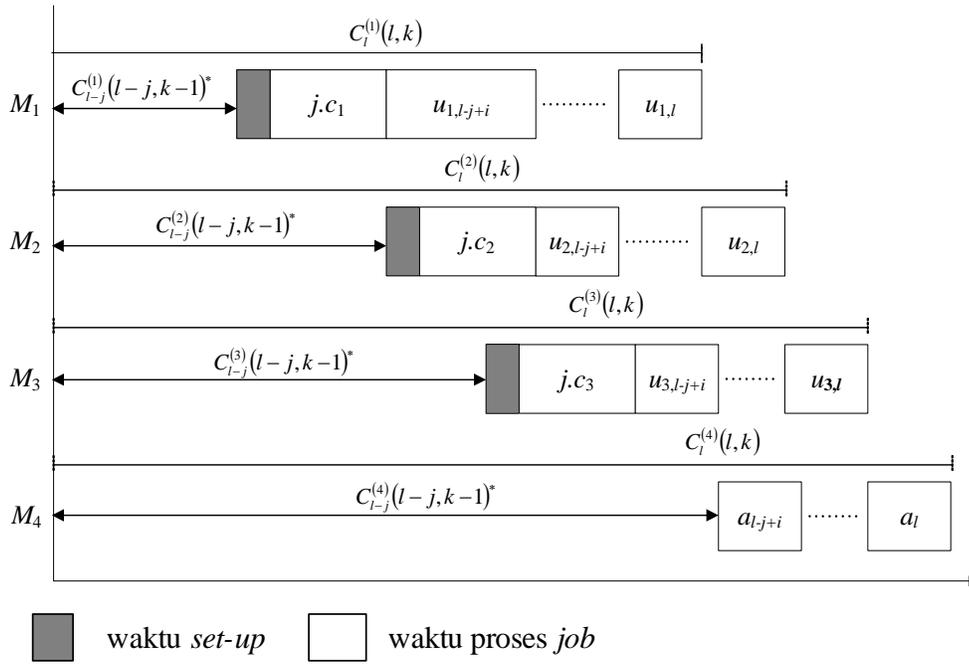
- $C_i^{(m)}(l, k)$ completion time job i di mesin m dengan jumlah job sampai batch k adalah l
- i indeks job ($i = 1, \dots, n$)
- m indeks mesin ($m = 1, 2, 3, 4$)
- k indeks batch
- j ukuran batch
- l jumlah job yang akan dijadwalkan sampai batch k
- d_i due date job ke- i
- N jumlah batch
- $p_{c,m}$ waktu proses common component di mesin m
- $p_{c,m,i}$ waktu proses common component job i di mesin m
- $p_{u,m,i}$ waktu proses unique component i di mesin m
- $p_{a,m,i}$ waktu perakitan job ke- i di mesin m
- s_m waktu set-up pada mesin m
- L_i lateness job i
- $L_{\max}(l, k)$ lateness maksimum sampai batch k dengan jumlah job sampai batch k adalah l
- $[i]$ job urutan ke- i

2.2 Persamaan Umum Model ($F_4/(c, u_i, a_i), b_d/L_{\max}$)

Persamaan umum untuk mendapatkan completion time Model ($F_4/(c, u_i, a_i), b_d/L_{\max}$) dapat dijelaskan dengan Gantt chart pada Gambar 2.

Pada Gambar 2 terlihat bahwa untuk mendapatkan completion time dari batch k diperlukan perhitungan dari batch sebelumnya. Pada perhitungan batch k nilai batch sebelumnya dinotasikan dengan $C_{l-j}^{(1)}(l-j, k-1)^*$, $C_{l-j}^{(2)}(l-j, k-1)^*$, $C_{l-j}^{(3)}(l-j, k-1)^*$, dan $C_{l-j}^{(4)}(l-j, k-1)^*$. Persamaan umum untuk menentukan completion time setiap job di setiap mesin pada Model ($F_4/(c, u_i, a_i), b_d/L_{\max}$) untuk batch k terdapat pada persamaan (1) – persamaan (4).

$$C_i^{(1)}(l, k) = C_{l-j}^{(1)}(l-j, k-1)^* + s_1 + j \cdot p_{c,1} + \sum_{i'=l-j+1}^i p_{u,1,i'} \quad (1)$$



Gambar 2. Gantt chart batch *k* Model $(F_4/(c, u_i, a_i), b_\alpha/L_{\max})$

$$C_i^{(2)}(l, k) = \max \left\{ \begin{array}{l} C_{l-j}^{(1)}(l-j, k-1)^* + \sum_{m=1}^2 s_m + j \cdot \sum_{m=1}^2 p_{c,m} + \sum_{i'=l-j+1}^i p_{u,2,i'} , \\ \max_{l-j+1 \leq z \leq i} \left\{ C_{l-j}^{(1)}(l-j, k-1)^* + s_1 + j \cdot p_{c,1} + \sum_{i'=l-j+1}^z p_{u,1,i'} + \sum_{i'=z}^i p_{u,2,i'} \right\} , \\ C_{l-j}^{(2)}(l-j, k-1)^* + s_2 + j \cdot p_{c,2} + \sum_{i'=l-j+1}^i p_{u,2,i'} \end{array} \right\} \quad (2)$$

$$C_i^{(3)}(l, k) = \max \left\{ \begin{array}{l} C_{l-j}^{(1)}(l-j, k-1)^* + \sum_{m=1}^3 s_m + j \cdot \sum_{m=1}^3 p_{c,m} + \sum_{i'=l-j+1}^i p_{u,3,i'} , \\ \max_{l-j+1 \leq z \leq i} \left\{ C_{l-j}^{(1)}(l-j, k-1)^* + \sum_{m=1}^2 s_m + j \cdot \sum_{m=1}^2 p_{c,m} + \sum_{i'=l-j+1}^z p_{u,2,i'} + \sum_{i'=z}^i p_{u,3,i'} \right\} , \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq y \leq i}} \left\{ C_{l-j}^{(1)}(l-j, k-1)^* + s_1 + j \cdot p_{c,1} + \sum_{i'=l-j+1}^z p_{u,1,i'} + \sum_{i'=z}^y p_{u,2,i'} + \sum_{i'=y}^i p_{u,3,i'} \right\} , \\ C_{l-j}^{(2)}(l-j, k-1)^* + \sum_{m=2}^3 s_m + j \cdot \sum_{m=2}^3 p_{c,m} + \sum_{i'=l-j+1}^i p_{u,3,i'} , \\ \max_{l-j+1 \leq z \leq i} \left\{ C_{l-j}^{(2)}(l-j, k-1)^* + s_2 + j \cdot p_{c,2} + \sum_{i'=l-j+1}^z p_{u,2,i'} + \sum_{i'=z}^i p_{u,3,i'} \right\} , \\ C_{l-j}^{(3)}(l-j, k-1)^* + s_3 + j \cdot p_{c,3} + \sum_{i'=l-j+1}^i p_{u,3,i'} \end{array} \right\} \quad (3)$$

$$C_i^{(4)}(l, k) = \max \left\{ \begin{array}{l} \max_{\substack{l-j+1 \leq z \leq i \\ z \leq i}} \left\{ C_{l-j}^{(1)}(l-j, k-1)^* + \sum_{m=1}^3 s_m + j \cdot \sum_{m=1}^3 p_{c,m} + \sum_{i'=l-j+1}^z p_{u,3,i'} + \sum_{i'=z}^i p_{a,i'} \right\}, \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq y \leq i}} \left\{ C_{l-j}^{(1)}(l-j, k-1)^* + \sum_{m=1}^2 s_m + j \cdot \sum_{m=1}^2 p_{c,m} + \sum_{i'=l-j+1}^z p_{u,2,i'} + \sum_{i'=z}^y p_{u,3,i'} + \sum_{i'=y}^i p_{a,i'} \right\}, \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq y \leq i \\ y \leq w \leq i}} \left\{ C_{l-j}^{(1)}(l-j, k-1)^* + s_1 + j \cdot p_{c,1} + \sum_{i'=l-j+1}^z p_{u,1,i'} + \sum_{i'=z}^y p_{u,2,i'} + \sum_{i'=y}^w p_{u,3,i'} + \sum_{i'=w}^i p_{a,i'} \right\}, \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq i}} \left\{ C_{l-j}^{(2)}(l-j, k-1)^* + \sum_{m=2}^3 s_m + j \cdot \sum_{m=2}^3 p_{c,m} + \sum_{i'=l-j+1}^z p_{u,3,i'} + \sum_{i'=z}^i p_{a,i'} \right\}, \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq y \leq i}} \left\{ C_{l-j}^{(2)}(l-j, k-1)^* + s_2 + j \cdot p_{c,2} + \sum_{i'=l-j+1}^z p_{u,2,i'} + \sum_{i'=z}^y p_{u,3,i'} + \sum_{i'=y}^i p_{a,i'} \right\}, \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq i}} \left\{ C_{l-j}^{(3)}(l-j, k-1)^* + s_3 + j \cdot p_{c,3} + \sum_{i'=l-j+1}^z p_{u,3,i'} + \sum_{i'=z}^i p_{a,i'} \right\}, C_{l-j}^{(4)}(l-j, k-1)^* + \sum_{i'=l-j+1}^i p_{a,i'} \end{array} \right\} \quad (4)$$

Setelah menghitung *completion time* di setiap mesin, langkah selanjutnya adalah menghitung *lateness* maksimum untuk setiap *job*. Persamaan umum untuk menghitung *lateness* untuk setiap *job* $i = 1, \dots, n$ adalah sebagai berikut:

$$L_i = C_i^{(4)}(l, k) - d_i \quad (5)$$

Persamaan untuk menghitung *lateness* maksimum diperoleh dengan memisalkan *lateness* maksimum untuk *batch* sebelumnya $(k - 1)$ adalah $L_{\max}(k - 1)$. Sehingga, persamaan umum *lateness* maksimum adalah sebagai berikut:

$$L_{\max}(l, k) = \max \left\{ \begin{array}{l} L_{\max}(l-j, k-1)^*, \\ \max_{\substack{l-j+1 \leq i \leq l}} \left\{ 0, \left(C_i^{(4)}(l, k) - d_i \right) \right\} \end{array} \right\} \quad (6)$$

2.3 Algoritma Model $(F_d/(c, u_i, a_i), b_d/L_{\max})$

Setelah diperoleh rumusan umum menghitung nilai *completion time* dan nilai *lateness* untuk *batch* k , maka langkah selanjutnya adalah menyusun algoritma penjadwalan untuk fabrikasi dan perakitan pada *flowshop-4-stage*. Algoritma disusun untuk menentukan urutan pengerjaan *job*, ukuran dan jumlah *batch*. Urutan pengerjaan *job* disusun berdasarkan *earliest due date* (EDD). Sedangkan penentuan ukuran dan jumlah *batch* dilakukan dengan mengikuti pendekatan yang dilakukan oleh Cheng dan Wang (1999), penjadwalan dilakukan dalam horizon perencanaan tertentu. Secara lengkap, algoritma Model $(F_d/(c, u_i, a_i), b_d/L_{\max})$ terdapat pada Gambar 3.

2.4 Persamaan Umum Model $(F_d/(c, u_i, a_i), b_d/\Sigma T)$

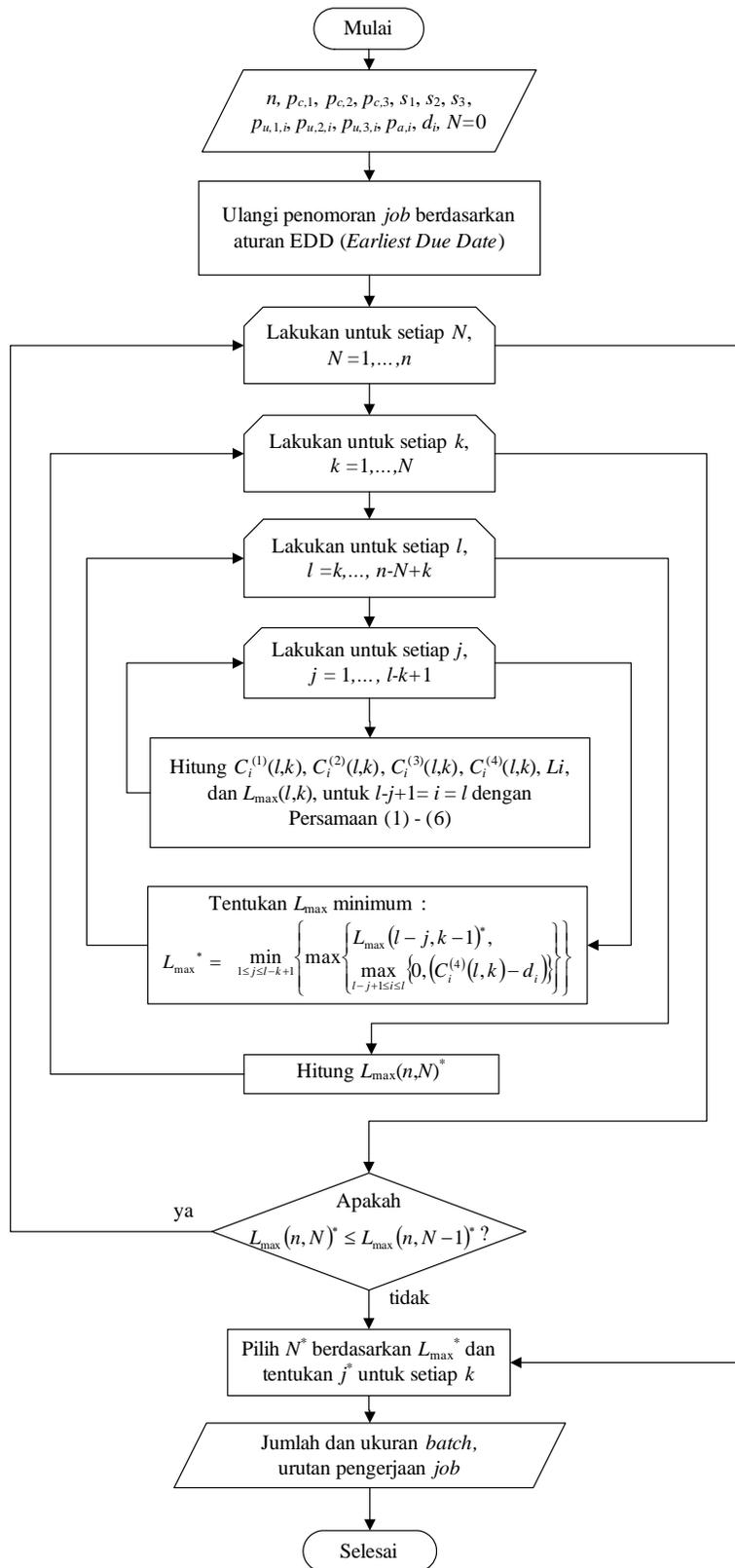
Persamaan umum untuk menghitung *completion time* Model $(F_d/(c, u_i, a_i), b_d/\Sigma T)$ di Mesin 1 sampai dengan Mesin 4 hampir sama dengan persamaan (1) sampai dengan persamaan (4), perbedaannya terletak pada penambahan *subscript*, misalnya notasi $C_i^{(1)}(l, k)$ berubah menjadi $C_{x,[i]}^{(1)}(l, k)$. Penambahan *subscript* ini bertujuan mengakomodasi pertukaran urutan *job* untuk kriteria minimisasi jumlah *tardy jobs*. Persamaan tersebut adalah sebagai berikut:

$$C_{x[i]}^{(1)}(l, k) = C_{x[l-j]}^{(1)}(l-j, k-1)^* + s_1 + j \cdot p_{c,1} + \sum_{i'=l-j+1}^i p_{u,1,x[i']} \quad (7)$$

$$C_{x[i]}^{(2)}(l, k) = \max \left\{ \begin{array}{l} C_{x[l-j]}^{(1)}(l-j, k-1)^* + \sum_{m=1}^2 s_m + j \cdot \sum_{m=1}^2 p_{c,m} + \sum_{i'=l-j+1}^i p_{u,2,x[i']}, \\ \max_{l-j+1 \leq z \leq i} \left\{ C_{x[l-j]}^{(1)}(l-j, k-1)^* + s_1 + j \cdot p_{c,1} + \sum_{i'=l-j+1}^z p_{u,1,x[i']} + \sum_{i'=z}^i p_{u,2,x[i']} \right\}, \\ C_{x[l-j]}^{(2)}(l-j, k-1)^* + s_2 + j \cdot p_{c,2} + \sum_{i'=l-j+1}^i p_{u,2,x[i']} \end{array} \right\} \quad (8)$$

$$C_{x[i]}^{(3)}(l, k) = \max \left\{ \begin{array}{l} C_{x[l-j]}^{(1)}(l-j, k-1)^* + \sum_{m=1}^3 s_m + j \cdot \sum_{m=1}^3 p_{c,m} + \sum_{i'=l-j+1}^i p_{u,3,x[i']}, \\ \max_{l-j+1 \leq z \leq i} \left\{ C_{x[l-j]}^{(1)}(l-j, k-1)^* + \sum_{m=1}^2 s_m + j \cdot \sum_{m=1}^2 p_{c,m} + \sum_{i'=l-j+1}^z p_{u,2,x[i']} + \sum_{i'=z}^i p_{u,3,x[i']} \right\}, \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq y \leq i}} \left\{ C_{x[l-j]}^{(1)}(l-j, k-1)^* + s_1 + j \cdot p_{c,1} + \sum_{i'=l-j+1}^z p_{u,1,x[i']} + \sum_{i'=z}^y p_{u,2,x[i']} + \sum_{i'=y}^i p_{u,3,x[i']} \right\}, \\ C_{x[l-j]}^{(2)}(l-j, k-1)^* + \sum_{m=2}^3 s_m + j \cdot \sum_{m=2}^3 p_{c,m} + \sum_{i'=l-j+1}^i p_{u,3,x[i']}, \\ \max_{l-j+1 \leq z \leq i} \left\{ C_{x[l-j]}^{(2)}(l-j, k-1)^* + s_2 + j \cdot p_{c,2} + \sum_{i'=l-j+1}^z p_{u,2,x[i']} + \sum_{i'=z}^i p_{u,3,x[i']} \right\}, \\ C_{x[l-j]}^{(3)}(l-j, k-1)^* + s_3 + 2p_{c,3} + \sum_{i'=l-j+1}^i p_{u,3,x[i']} \end{array} \right\} \quad (9)$$

$$C_{x[i]}^{(4)}(l, k) = \max \left\{ \begin{array}{l} \max_{l-j+1 \leq z \leq i} \left\{ C_{x[l-j]}^{(1)}(l-j, k-1)^* + \sum_{m=1}^3 s_m + j \cdot \sum_{m=1}^3 p_{c,m} + \sum_{i'=l-j+1}^z p_{u,3,x[i']} + \sum_{i'=z}^i p_{a,x[i']} \right\}, \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq y \leq i}} \left\{ C_{x[l-j]}^{(1)}(l-j, k-1)^* + \sum_{m=1}^2 s_m + j \cdot \sum_{m=1}^2 p_{c,m} + \sum_{i'=l-j+1}^z p_{u,2,x[i']} + \sum_{i'=z}^y p_{u,3,x[i']} + \sum_{i'=y}^i p_{a,x[i']} \right\}, \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq y \leq i \\ y \leq w \leq i}} \left\{ C_{x[l-j]}^{(1)}(l-j, k-1)^* + s_1 + j \cdot p_{c,1} + \sum_{i'=l-j+1}^z p_{u,1,x[i']} + \sum_{i'=z}^y p_{u,2,x[i']} + \sum_{i'=y}^w p_{u,3,x[i']} + \sum_{i'=w}^i p_{a,x[i']} \right\}, \\ \max_{l-j+1 \leq z \leq i} \left\{ C_{x[l-j]}^{(2)}(l-j, k-1)^* + \sum_{m=2}^3 s_m + j \cdot \sum_{m=2}^3 p_{c,m} + \sum_{i'=l-j+1}^z p_{u,3,x[i']} + \sum_{i'=z}^i p_{a,x[i']} \right\}, \\ \max_{\substack{l-j+1 \leq z \leq i \\ z \leq y \leq i}} \left\{ C_{x[l-j]}^{(2)}(l-j, k-1)^* + s_2 + j \cdot p_{c,2} + \sum_{i'=l-j+1}^z p_{u,2,x[i']} + \sum_{i'=z}^y p_{u,3,x[i']} + \sum_{i'=y}^i p_{a,x[i']} \right\}, \\ \max_{l-j+1 \leq z \leq i} \left\{ C_{x[l-j]}^{(3)}(l-j, k-1)^* + s_3 + j \cdot p_{c,3} + \sum_{i'=l-j+1}^z p_{u,3,x[i']} + \sum_{i'=z}^i p_{a,x[i']} \right\}, \\ C_{x[l-j]}^{(4)}(l-j, k-1)^* + \sum_{i'=l-j+1}^i p_{a,x[i']} \end{array} \right\} \quad (10)$$



Gambar 3. Flow chart Algoritma $(F_4/(c, u_i, a_i), b_d/L_{max})$

Sedangkan persamaan untuk menghitung *lateness* untuk setiap *job* ($i = 1, \dots, n$) adalah sebagai berikut:

$$L_{x[i]}(l, k) = C_{x[i]}^{(4)}(l, k) - d_{x[i]} \tag{11}$$

Persamaan umum perhitungan jumlah *tardy jobs* adalah:

$$T_j = \sum_{i=l-j+1}^l \delta_i, \quad \delta_i = \begin{cases} 1, & \text{jika } L_{x[i]}(l, k) > 0, \\ 0, & \text{untuk lainnya} \end{cases} \tag{12}$$

Algoritma penjadwalan dan penentuan jumlah dan ukuran *batch* Model $(F_4/(c, u_i, a_i), b_d/\Sigma T)$ dapat dilihat pada Gambar 4.

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Model dan Analisis

Untuk mengetahui performansi model yang dikembangkan, dilakukan pengujian model dengan menggunakan beberapa skenario. Pengujian model terdiri dari dua tahap. Tahap pertama adalah menguji model untuk melihat cara kerja model dalam hal melakukan penjadwalan yang menghasilkan *lateness* maksimum yang minimum. Tahap kedua adalah menguji perilaku model yang dikembangkan dengan menggunakan beberapa nilai parameter yang berbeda, yaitu nilai waktu *set up* dan waktu proses *common component*.

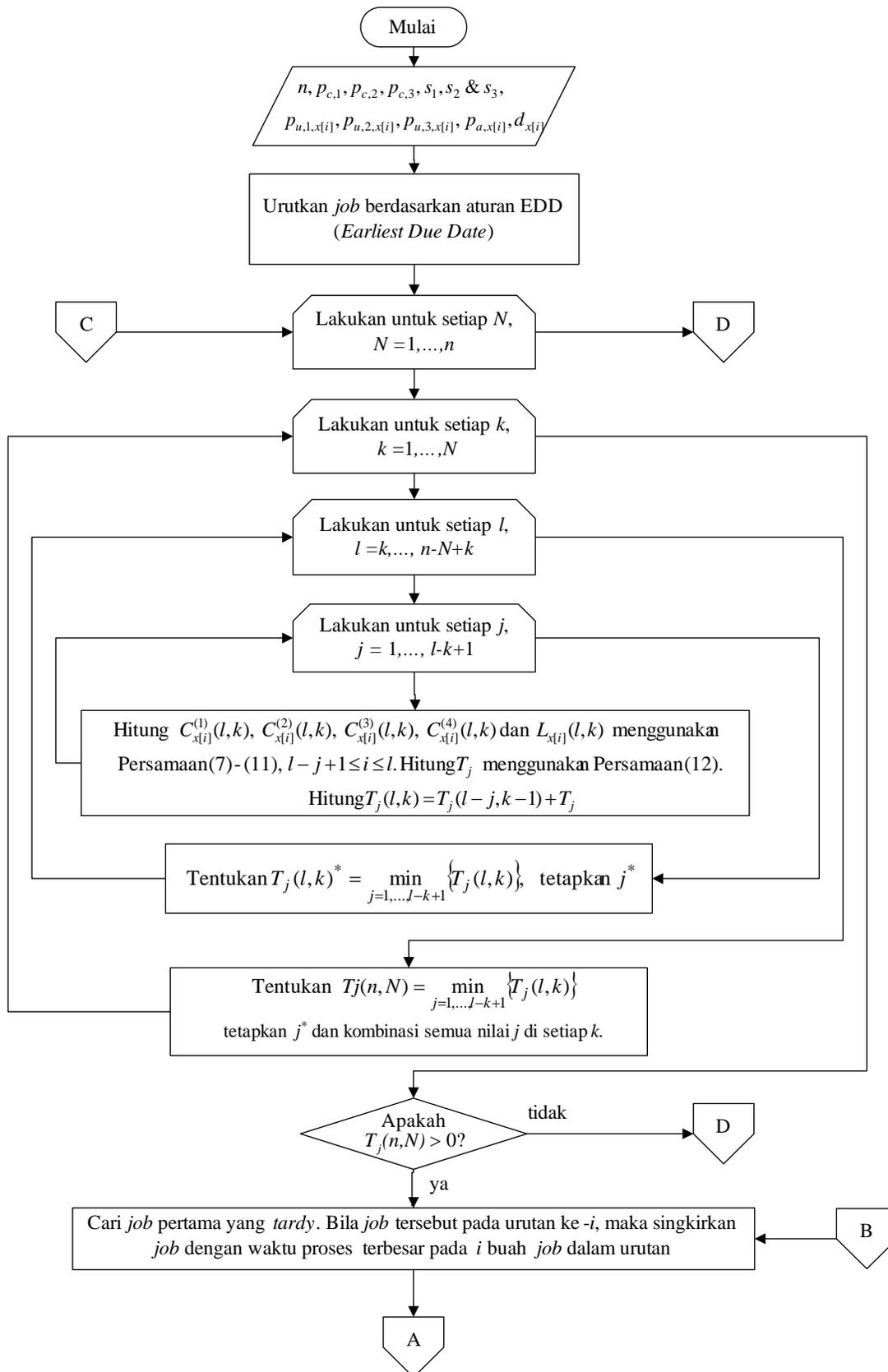
3.2 Pengujian Tahap 1

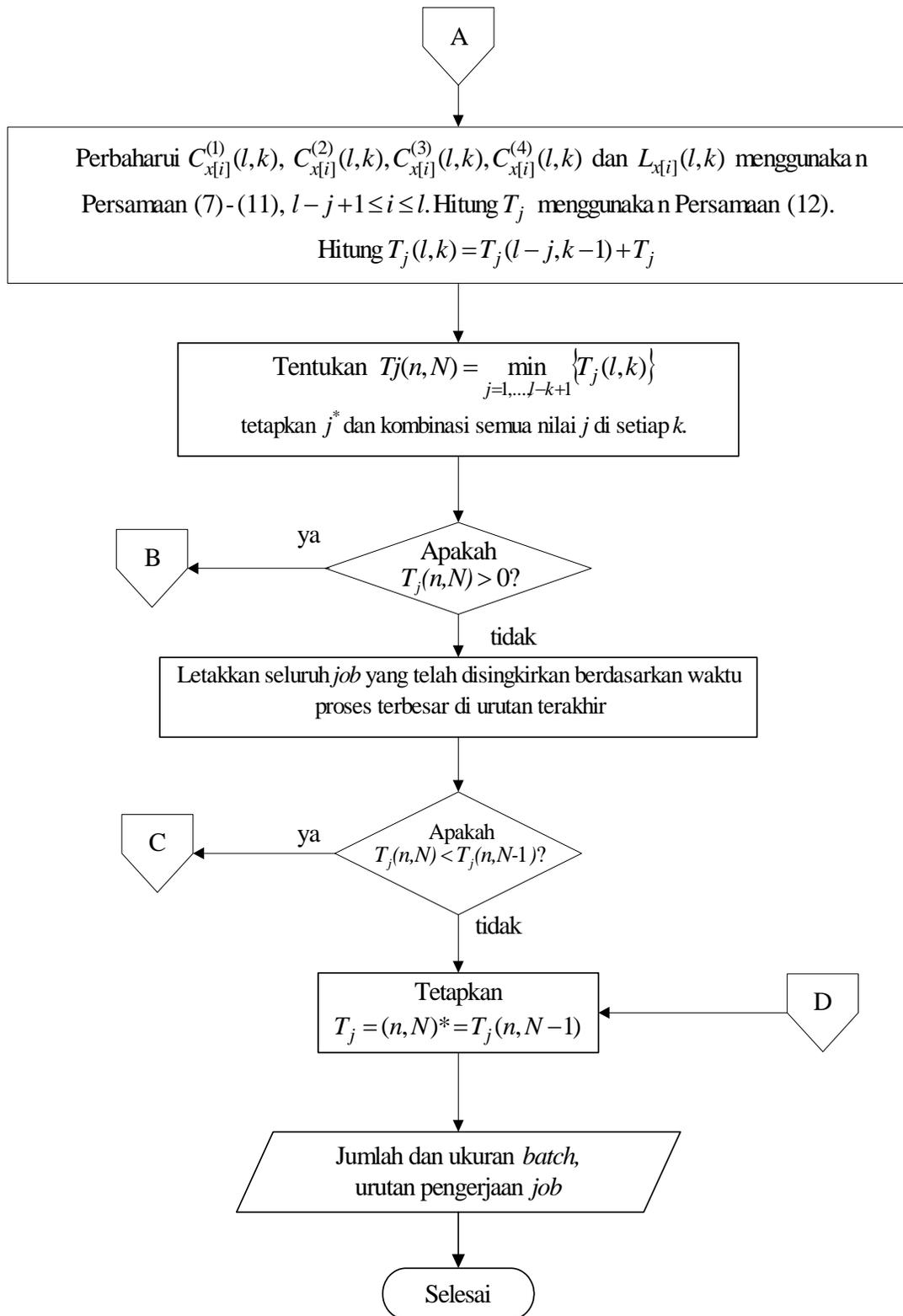
Pada tahap pertama, pengujian dilakukan untuk melihat cara kerja Model $(F_4/(c, u_i, a_i), b_d/L_{\max})$ dan Model $(F_4/(c, u_i, a_i), b_d/\Sigma T)$. Data pengujian tahap 1 untuk Model $(F_4/(c, u_i, a_i), b_d/L_{\max})$ disusun dengan membangkitkan secara random data waktu proses *common component*, waktu proses *unique component*, waktu operasi perakitan dan data waktu *setup* pada interval nilai 1 sampai 15, sedangkan data *due date* dibangkitkan pada interval 20 sampai 50. Data pengujian Tahap 1 terdiri dari 10 *job* dan data lengkapnya terdapat pada Tabel 1.

Tabel 1. Data pengujian Tahap 1 Model $(F_4/(c, u_i, a_i), b_d/L_{\max})$

<i>Job</i>	$p_{c,1}$	$p_{c,2}$	$p_{c,3}$	$p_{u,1,i}$	$p_{u,2,i}$	$p_{u,3,i}$	$p_{a,i}$	d_i
1	2	3	4	8	8	12	10	44
2	2	3	4	6	13	9	7	31
3	2	3	4	7	12	5	13	36
4	2	3	4	12	12	8	12	31
5	2	3	4	8	8	6	12	46
6	2	3	4	7	10	7	10	38
7	2	3	4	14	11	7	15	50
8	2	3	4	9	8	13	12	34
9	2	3	4	12	7	12	11	27
10	2	3	4	7	15	8	12	27

$s_1=4, s_2=5$ dan $s_3=6$





Gambar 4. Flow chart Algoritma $(F_4/(c, u_i, a_i), b_d / \Sigma T)$

Dengan mengikuti seluruh langkah pada Algoritma Model $(F_4/(c,u_i,a_i),b_d/L_{max})$ maka hasil yang diperoleh dapat dilihat pada Tabel 2.

Data pengujian Tahap 1 untuk Model $(F_4/(c,u_i,a_i),b_d/\Sigma T)$ disusun dengan membangkitkan secara random data waktu proses *common component*, waktu proses *unique component*, waktu operasi perakitan dan data waktu *setup* pada interval nilai 1 sampai 16, sedangkan data *due date* dibangkitkan pada interval 50 sampai 200. Data pengujian Tahap 1 terdiri dari 10 *job* dan data lengkapnya terdapat pada Tabel 3.

Dengan mengikuti langkah-langkah algoritma Model $(F_4/(c,u_i,a_i),b_d/\Sigma T)$, maka hasil yang diperoleh dari pengujian Model $(F_4/(c,u_i,a_i),b_d/\Sigma T)$ dapat dilihat pada Tabel 4.

Berdasarkan Tabel 4, dapat dilihat bahwa terdapat penurunan jumlah *tardy jobs* dari $N = 1$ hingga $N = 3$, namun pada $N = 4$ jumlah *tardy jobs* mengalami kenaikan, sehingga jumlah *batch* yang terpilih adalah sebesar 3 *batch*, dengan jumlah *job* pada *Batch 1*: 2 *job*, *Batch 2*: 3 *job*, dan *Batch 3*: 5 *job*.

Tabel 2. Hasil pengujian Tahap 1 Model $(F_4/(c,u_i,a_i),b_d/L_{max})$

Jumlah <i>batch</i>	<i>batch</i> ke-	<i>j</i>	L_{max}^*
$N=1$	$k=1$	10	181
$N=2$	$k=1$	5	151
	$k=2$	5	
$N=3$	$k=1$	5	147
	$k=2$	4	
	$k=3$	1	
$N=4$	$k=1$	2	143
	$k=2$	4	
	$k=3$	3	
	$k=4$	1	
$N=5$	$k=1$	2	147
	$k=2$	4	
	$k=3$	2	
	$k=4$	1	
	$k=5$	1	

Tabel 3. Data pengujian Tahap 1 Model $(F_4/(c,u_i,a_i),b_d/\Sigma T)$

<i>x</i>	$P_{c,1}$	$P_{u,1,x[i]}$	$P_{c,2}$	$P_{u,2,x[i]}$	$P_{c,3}$	$P_{u,3,x[i]}$	$P_{a,x[i]}$	$d_{x[i]}$
1	4	9	5	6	6	12	15	130
2	4	15	5	16	6	5	9	180
3	4	14	5	7	6	8	6	120
4	4	5	5	7	6	7	5	60
5	4	9	5	9	6	15	10	150
6	4	10	5	8	6	16	14	170
7	4	10	5	11	6	9	16	160
8	4	6	5	8	6	14	12	110
9	4	6	5	6	6	13	13	90
10	4	16	5	15	6	15	11	190

$S_1 = 2, S_2 = 3$ dan $S_3 = 4$

Tabel 4. Hasil pengujian Tahap 1 Model $(F_4/(c,u_i,a_i),b_d/\Sigma T)$

Jumlah <i>batch</i> (<i>N</i>)	<i>Batch</i> ke- (<i>k</i>)	Ukuran <i>batch</i> (<i>j</i>)	Jumlah <i>tardy jobs</i> $T_j(n,N)^*$
1	1	10	8
2	1	3	5
	2	7	
3	1	2	3
	2	3	
	3	5	
4	1	2	4
	2	3	
	3	1	
	4	4	

3.3 Pengujian Tahap 2

Pada tahap kedua, pengujian perilaku model dilakukan dengan cara mengubah nilai-nilai parameter berbeda terhadap nilai waktu *setup* dan waktu proses *common component*. Pengujian tahap kedua ini disusun ke dalam 3 skenario.

Pada Skenario 1, akan dilakukan pengujian untuk melihat perilaku model terhadap perubahan nilai waktu *setup*. Tujuan pengujian ini adalah untuk mengetahui pengaruh kenaikan waktu *setup* terhadap jumlah *batch* dan nilai *lateness* maksimum. Data pengujian skenario ini sama dengan Tabel 1 kecuali variasi waktu *setup*-nya.

Tabel 5. Data pengujian Tahap 2 Skenario 3

<i>Job</i>	$p_{c,1}$	$p_{c,2}$	$p_{c,3}$	$p_{u,1,i}$	$p_{u,2,i}$	$p_{u,3,i}$	$p_{a,i}$	d_i
1	5	9	9	6	12	5	7	65
2	5	15	9	3	12	8	6	90
3	5	16	9	6	12	7	14	150
4	5	9	9	9	12	9	5	30
5	5	10	9	10	12	3	12	110
6	5	16	9	12	12	6	11	125
7	5	14	9	14	12	8	15	145
8	5	7	9	3	12	6	8	45
9	5	16	9	6	12	4	10	115
10	5	16	9	8	12	2	16	180
11	5	12	9	4	12	4	12	1000
12	5	16	9	2	12	7	13	1200
13	5	11	9	7	12	4	14	120
14	5	10	9	9	12	3	21	570
15	5	9	9	10	12	3	25	90
16	5	6	9	12	12	3	4	87
17	5	4	9	13	12	4	6	67
18	5	12	9	14	12	4	8	90
19	5	10	9	3	12	4	2	120
20	5	8	9	5	12	4	6	200

$s_1=4, s_2=5$ dan $s_3=6$

Tabel 6. Hasil pengujian Tahap 2 Skenario 1

Waktu <i>setup</i>			N^*	L_{max}^*
s_1	s_2	s_3		
4	5	6	4	143
5	6	7	4	148
6	7	8	4	154
7	8	9	3	159
8	9	10	3	164
9	10	11	3	169
10	11	12	3	174
11	12	13	2,3	179
12	13	14	2	183
13	14	15	2	187

Pada Skenario 2, akan dilakukan pengujian untuk melihat perilaku model terhadap perubahan nilai waktu proses *common component*. Pengujian ini bertujuan untuk mengetahui pengaruh kenaikan waktu proses *common component* terhadap jumlah *batch* dan nilai *lateness* maksimum. Data pengujian juga sama dengan Tabel 1, kecuali variasi waktu proses *common component*-nya.

Pada Skenario 3, akan dilakukan pengujian untuk melihat perilaku model terhadap kenaikan jumlah *job*. Tujuan pengujian ini adalah untuk mengetahui pengaruh kenaikan jumlah *job* terhadap jumlah *batch* dan nilai *lateness* maksimum. Jumlah *job* mulai dari 10 *job* sampai dengan 20 *job*. Untuk jumlah data 10 *job*, data yang diambil adalah 10 *job* pertama dan jumlah data 12 *job*, data yang diambil adalah 12 *job* pertama. Data untuk Skenario 3 terdapat pada Tabel 5.

Pengujian model dilakukan dengan menggunakan program bantu *Visual Basic Application* (VBA) for Excel dan dijalankan pada Personal Computer Intel Pentium Core Duo dengan RAM 1 GB pada sistem Microsoft Window XP. Hasil pengujian Skenario 1 dapat dilihat pada Tabel 6, hasil pengujian Skenario 2 dapat dilihat pada Tabel 7 dan hasil pengujian Skenario 3 terdapat pada Tabel 8.

Tabel 6 memperlihatkan bahwa kenaikan waktu *setup* mempengaruhi jumlah *batch*. Jumlah *batch* semakin kecil dengan waktu *setup* yang semakin besar. Tabel 7 memperlihatkan bahwa kenaikan waktu proses *common component* mempengaruhi jumlah *batch*, jumlah *batch* optimal akan semakin besar seiring dengan meningkatnya waktu proses *common component*.

Tabel 7. Hasil pengujian Tahap 2 Skenario 2

Waktu proses <i>common component</i>			N^*	L_{max}^*
$p_{c,1}$	$p_{c,2}$	$p_{c,3}$		
2	3	4	4	143
3	4	5	4	157
4	5	6	4	172
5	6	7	4	187
6	7	8	4	202
7	8	9	4	218
8	9	10	4	234
9	10	11	5	248
10	11	12	5	262
11	12	13	5	276

Tabel 8. Hasil pengujian Tahap 2 Skenario 3

Jumlah <i>job</i>	N^*	L_{\max}^*	Lama <i>running</i> program
10	5	110	1 menit
12	7	110	2 menit
14	8	130	5 menit
16	10	162	12 menit
18	11	200	26 menit
20	11	220	75 menit

Kenaikan jumlah *job* juga akan mempengaruhi jumlah *batch*. Jumlah *batch* akan makin besar dengan *job* yang makin besar. Alasannya adalah dengan jumlah *job* makin besar, bila tetap mempertahankan jumlah *batch* yang kecil, maka akan memperbesar ukuran *batch*. Hal ini menyebabkan saat selesainya lebih lambat, sehingga *lateness*-nya membesar. Sehingga, untuk meminimalkan nilai *lateness*, jumlah *batch* dibuat lebih banyak.

4. KESIMPULAN

Model penjadwalan *job* pada *flowshop-4-stage* untuk proses fabrikasi dan perakitan yang dikembangkan pada penelitian ini bekerja dalam dua tahap yaitu proses *sequencing* dan proses *batching*. Algoritma yang diusulkan bukan merupakan algoritma optimal karena proses penjadwalan yang tidak dijamin optimal, meskipun proses *batching* dapat menghasilkan ukuran *batch* optimal. Model yang diusulkan tidak bisa dibandingkan dengan model lain karena model yang telah ada mempunyai karakteristik tidak komparatif.

Perilaku model menunjukkan sifat sebagai berikut: jika waktu *setup* diperbesar, maka jumlah *batch* makin kecil. Selain itu, jika waktu proses *common component* diperbesar, maka jumlah *batch* makin besar dan jika jumlah *job* makin besar, maka jumlah *batch* makin besar.

Pengembangan yang dapat dilakukan berbasis penelitian ini adalah pengembangan model untuk *flowshop m* mesin; pengembangan yang mengakomodasi kondisi sebuah produk terdiri lebih dari 1 unit *common component* dan 1 unit *unique component*; dan pengembangan terhadap metode pemecahan masalah menggunakan algoritma metaheuristik seperti algoritma genetik, *simulated annealing* dan *greedy randomized adaptive search procedure* (GRASP).

DAFTAR PUSTAKA

- Bukchin, J., Tzur, M., and Jaffe, M., 2002. "Lot Splitting to Minimize Average Flow-Time in a Two-Machine Flow-Shop." *IIE Transactions*, Vol. 34, pp. 953–970.
- Chan, D., and Bedworth, D. D., 1990. "Design of Scheduling Systems for Flexible Manufacturing Cells." *International Journal of Production Research*, Vol. 28, pp. 2037– 2049.
- Cheng, T. C. E., and Wang, G., 1998. "Batching and Scheduling to Minimize the Makespan in the Two Machine Flowshop." *IIE Transactions*, Vol. 30, pp. 447-453.
- Cheng, T. C. E., and Wang, G., 1999. "Scheduling the Fabrication and Assembly of Components in a Two-Machine Flowshop." *IIE Transactions*, Vol. 31, pp. 135-143.
- Halim, A. H., dan Cahya, B. I., 1999. "Model *Overlapping* pada *Flowshop-4-Stage* yang Memproses Multi-Item." *Proceeding Seminar Sistem Produksi IV*, Bandung.

- Halim, A. H., dan Saleh, A., 2005. "Model Penjadwalan untuk Fabrikasi dan Perakitan pada *Flow Shop* 2 Mesin dengan Kriteria Minimisasi Total Waktu Tinggal Aktual." *Jurnal ITENAS: Jurnal Ilmu Pengetahuan, Teknologi dan Seni*, Vol. 8, No. 3, pp. 94-106.
- Indrapriyatna, A. S., Suprayogi, Iskandar, B. P., dan Halim, A. H., 2008, "Model Penjadwalan *Batch* pada Satu Mesin yang Mengalami Deteriorasi untuk Minimasi Total Biaya Simpan dan Biaya Kualitas." *Jurnal Teknik Industri*, Vol. 10, No. 1, pp. 26-37.
- Tansel, B.Ç., Kara, B.Y., and Sabuncuoglu, I., 2001. "An Efficient Algorithm for the Single Machine Total Tardiness Model." *IIE Transactions*, Vol. 33, pp. 661–674.
- Xiao, W., and Li, C., 2002. "Approximation Algorithms for Common due Date Assignment and Job Scheduling on Parallel Machines." *IIE Transactions*, Vol. 34, pp. 467–477.