

## **PENJADWALAN BATCH PADA *FLOW SHOP* DINAMIS UNTUK MEMINIMASI BIAYA PRODUKSI**

**Tengku Nurainun**

Jurusan Teknik Industri, Fakultas Sains dan Teknologi,  
Universitas Islam Negeri Sultan Syarif Kasim Riau  
Jl. H. R. Soebrantas No. 155 KM. 18, Pekanbaru, 282893, Indonesia  
e-mail: [tn\\_ainun@yahoo.com](mailto:tn_ainun@yahoo.com)

### ***Abstrak***

*Keterbatasan sumber daya yang dimiliki untuk mengerjakan sejumlah tugas pada saat bersamaan memunculkan permasalahan pengambilan keputusan penyusunan jadwal produksi yaitu mengenai urutan pengerjaan tugas (sequencing) dan alokasi dari sumber daya yang akan digunakan. Permasalahan menjadi semakin kompleks manakala dihadapkan pada pencapaian kondisi ideal yaitu penyelesaian pekerjaan tepat pada saat due date. Pada kondisi nyata, sulit sekali untuk menghasilkan jadwal yang tepat waktu karena pekerjaan datang relatif bersamaan dalam jumlah yang banyak. Untuk memenuhi hal tersebut sistem manufaktur terpaksa menjadwalkan pemrosesan sejumlah pekerjaan lebih awal sehingga mendahului due date. Hal ini memunculkan konflik antara minimasi ongkos simpan yang akan mengakibatkan peningkatan ongkos setup, dan sebaliknya. Kedatangan pesanan baru juga menimbulkan konflik penerimaan atau penolakan atas kedatangan pesanan baru tersebut karena penerimaan berarti penyesuaian terhadap jadwal yang telah dibuat sedangkan penolakan berarti kehilangan kesempatan atas pekerjaan yang ditawarkan dimana kedua keputusan ini memuat konsekuensi biaya. Penelitian ini mencoba untuk mengakomodasi permasalahan penjadwalan pekerjaan yang dikelompokkan pengerjaannya ke dalam ukuran batch untuk meminimasi waktu setup dan menjawab permintaan konsumen yang variatif dengan mempertimbangkan kemungkinan kedatangan pekerjaan (job) baru. Kriteria performansi yang digunakan dalam penelitian ini adalah meminimasi total biaya produksi dimana sudah mempertimbangkan biaya-biaya yang berpotensi besar terhadap peningkatan biaya produksi baik pada saat persiapan, pengerjaan, maupun biaya yang muncul setelah produk selesai diproduksi. Keputusan penjadwalan berupa alokasi part pada sumber yang tersedia, pengelompokan part ke dalam batch, dan pengurutan batch yang dihasilkan, serta keputusan penerimaan atau penolakan pekerjaan. Pengujian numerik memperlihatkan bahwa model usulan mampu menyelesaikan permasalahan penjadwalan tersebut dan merupakan solusi yang optimal.*

***Kata Kunci:*** penjadwalan batch, flow shop dinamis, penolakan pekerjaan, biaya produksi

### **1. PENDAHULUAN**

Penelitian mengenai penjadwalan *batch* pada sistem manufaktur *flow shop* telah banyak dilakukan antara lain oleh Cahya (1998), Tejaasih (1998), Zaini (1999), Kurniawan (2003), dan Suryadhini (2006). Namun diantara penelitian-penelitian yang telah dilakukan masih mengabaikan adanya gangguan (*disturbance*) selama horizon perencanaan. Pada kondisi nyata, sistem produksi sering mengalami banyak gangguan antara lain kerusakan mesin, kemunculan pekerjaan (*job*) baru, pembatalan pekerjaan, perubahan *due date*, dan sebagainya sehingga mengakibatkan jadwal yang telah dibuat harus direvisi. Setiawati dan Halim (2003) mengembangkan sebuah model penjadwalan yang bersifat *nonresumable* pada *flow shop* dua mesin dimana model ini telah mempertimbangkan kemungkinan ketidakterersediaan mesin karena mengalami kerusakan maupun *preventif maintenance* untuk meminimasi *mean earliness*.

Ras dan Halim (2002) mengembangkan model penjadwalan *batch* yang mengakomodasi kedatangan pesanan baru pada sistem manufaktur *job shop* dengan menggunakan teknik insersi, yang diusulkan oleh Sotskov *et al.* (1999), untuk meminimasi waktu tinggal aktual. Kriteria performansi ini

dikembangkan oleh Halim dan Ohta (1993) untuk mengakomodasi kasus dinamis, yaitu kenyataan bahwa *demand* bisa datang setiap saat (tidak harus datang pada saat nol), dan tuntutan penyelesaian pekerjaan tepat waktu. Dengan demikian berarti saat selesai *job* tidak diperkenankan melanggar *due date* dan *job* yang akan diproses tidak perlu berada di lantai produksi secara bersamaan pada saat nol, tetapi bisa tiba di lantai produksi pada saat diperlukan.

Pada kondisi nyata, sulit sekali untuk menghasilkan jadwal yang tepat waktu karena pekerjaan datang relatif bersamaan dalam jumlah yang banyak. Untuk memenuhi hal tersebut sistem manufaktur terpaksa menjadwalkan pemrosesan sejumlah pekerjaan lebih awal sehingga mendahului *due date*. Hal ini memunculkan konflik antara minimasi ongkos simpan yang akan mengakibatkan peningkatan ongkos setup, dan sebaliknya. Kedatangan pesanan baru juga menimbulkan konflik penerimaan atau penolakan atas kedatangan pesanan baru tersebut karena penerimaan berarti penyesuaian terhadap jadwal yang telah dibuat sedangkan penolakan berarti kehilangan kesempatan atas pekerjaan yang ditawarkan dimana kedua keputusan ini memuat konsekuensi biaya. Halim dan Zakaria (2002) menjawab permasalahan ini dengan mengembangkan model penjadwalan yang sudah mempertimbangkan biaya produksi, yang merupakan derivasi dari definisi waktu tinggal aktual untuk sistem manufaktur tanpa *tardy jobs*, sebagai ukuran performansi. Biaya produksi yang dipertimbangkan yaitu biaya *set up*, biaya simpan, dan biaya penolakan pekerjaan. Penelitian mengenai penjadwalan *batch* dinamis pada *flow shop* sebenarnya telah dilakukan oleh Halim dan Barnali (1998) namun belum melibatkan biaya produksi sebagai pertimbangan dalam pengambilan keputusan. Oleh karena itu penelitian ini mencoba untuk mengembangkan sebuah model penjadwalan *batch* pada *flow shop* dengan mempertimbangkan kemungkinan kedatangan pekerjaan (*job*) baru untuk meminimasi biaya produksi.

## **2. FORMULASI MODEL**

### **2.1 Model Dasar**

Dalam merumuskan formulasi model terlebih dahulu dilakukan identifikasi faktor-faktor yang relevan dengan masalah yang dihadapi sesuai dengan tujuan penelitian, serta hubungan antar faktor kemudian diformulasikan menjadi sebuah model. Tujuan dari penelitian ini adalah untuk menghasilkan model penjadwalan *batch* pada *flow shop*  $m$  mesin untuk meminimasi total biaya produksi. Formulasi model yang dikembangkan dinyatakan dalam formulasi *integer non-linear programming* (INLP) dimana formulasi INLP ini sesuai karena pada permasalahan terdapat beberapa variabel keputusan yang dikehendaki bernilai integer, yaitu jumlah dan ukuran *batch*. Selain itu dalam formulasi model juga diperlukan penggunaan variabel biner, yaitu variabel integer yang memiliki nilai 0 dan 1, untuk menyelesaikan permasalahan berupa penentuan jenis *item* yang dimuat dalam suatu *batch* serta penentuan *item* yang akan ditolak pengerjaannya disebabkan karena biaya pengerjaan yang akan ditanamkan lebih berkontribusi terhadap peningkatan total biaya produksi dibandingkan dengan biaya penolakan pekerjaan dari *item* tersebut. Dengan demikian, pada formulasi model digunakan beberapa variabel yang memiliki nilai-nilai integer yaitu jumlah *batch* ( $N$ ), ukuran *batch* ( $Q_i$ ), variabel biner yang menyatakan eksistensi suatu jenis *item* pada suatu *batch* ( $r_{i,g}$ ), dan variabel biner penolakan pekerjaan ( $X_g$ ).

Beberapa penelitian sebelumnya yang dijadikan sebagai dasar pengembangan model antara lain:

1. Halim and Ohta (1996) yang mengembangkan model penjadwalan *batch* dinamis pada *flow shop* dengan mempertimbangkan konsep tepat waktu dimana pekerjaan tidak harus datang seluruhnya secara serentak di awal produksi serta tidak perlu dilakukan pengiriman produk setiap saat kepada konsumen melainkan dapat dilakukan bersamaan pada saat *due date*.
2. Halim dan Barnali (1998) yang mengembangkan model penjadwalan *batch* dinamis pada *flow shop* untuk kasus *multi-item* dengan kriteria untuk meminimasi waktu tinggal aktual.
3. Halim dan Zakaria (2002) yang mengembangkan model penjadwalan *batch* pada *job shop* dinamis dimana kriteria yang ingin dicapai adalah minimasi biaya produksi, yang merupakan derivasi dari definisi waktu tinggal aktual untuk sistem manufaktur tanpa *tardy jobs*, dan biaya penolakan pekerjaan.

4. Aisyati, *et.al* (2007) yang mengembangkan model penjadwalan *batch* pada *flow shop* dinamis untuk meminimasi rata-rata keterlambatan penyelesaian *order* (*mean tardiness*) dimana peneliti sudah mempertimbangkan kedatangan *order* baru berupa pengembangan algoritma penjadwalan ulang yang sudah dilengkapi dengan keputusan penolakan atau penerimaan *order* baru berdasarkan fisibilitas pemenuhan *due date*.

Posisi penelitian yang dilakukan terhadap penelitian sebelumnya dapat dilihat pada Tabel 1 sebagai berikut.

**Tabel 1. Posisi Penelitian**

	<b>Halim dan Ohta (1996)</b>	<b>Halim dan Barnali (1998)</b>	<b>Halim dan Zakaria (2002)</b>	<b>Aisyati, et.al (2007)</b>	<b>Penelitian Usulan</b>
<b>Sistem Manufaktur</b>	FS	FS	JS	FS	FS
<i>Due date</i>	<i>Common</i>	<i>Common</i>	<i>Common</i>	<i>Multi</i>	<i>Common</i>
<i>Earliness</i>	<i>Infeasible</i>	<i>Infeasible</i>	<i>Feasible</i>	<i>Feasible</i>	<i>Feasible</i>
<i>Tardiness</i>	<i>Infeasible</i>	<i>Infeasible</i>	<i>Infeasible</i>	<i>Feasible</i>	<i>Infeasible</i>
<i>Disturbance</i>	Tidak	Tidak	Ya	Ya	Ya
<b>Biaya Produksi</b>	Tidak	Tidak	Ya	Tidak	Ya
<b>Biaya Penolakan Pekerjaan</b>	Tidak	Tidak	Ya	Tidak	Ya
<b>Kriteria Performansi</b>	TAFT	TAFT	TC	MT	TC

Keterangan: FS : *Flow Shop*, JS : *Job Shop*, TAFT : *Total Actual Flow Time*, TC : *Total Cost*, MT : *Mean Tardiness*

Pada Tabel 1 dapat dilihat bahwa penelitian yang dilakukan mengakomodasi beberapa hal yang belum dipertimbangkan dalam penelitian sebelumnya. Pada penelitian Ohta (1996) dan Barnali(1998) penjadwalan dilakukan menggunakan pendekatan mundur sehingga semua *batch* harus memenuhi *due date* secara tepat waktu, namun hal ini akan menyebabkan semua pekerjaan yang melanggar saat mulai ( $T=0$ ) menjadi tidak fisibel atau ditolak pengerjaannya. Padahal kenyataannya sulit sekali mengatur agar semua pekerjaan selesai tepat waktu terutama apabila semua pekerjaan tersebut berada pada *due date* yang ketat.

Penelitian yang dilakukan oleh Halim dan Zakaria (2002) sudah mempertimbangkan faktor toleransi untuk pekerjaan yang tidak diselesaikan tepat pada saat *due date* dimana model yang dikembangkan hanya mengizinkan pekerjaan diselesaikan lebih awal daripada *due date* dan tidak boleh ada pekerjaan yang terlambat. Penelitian ini sudah melibatkan perhitungan biaya produksi di dalam model yang dikembangkan dimana keputusan untuk menerima atau menolak suatu pekerjaan dilakukan berdasarkan perbandingan antara biaya pengerjaan produk dengan resiko kerugian apabila pekerjaan tersebut ditolak pengerjaannya.

Penelitian usulan mencoba untuk mengembangkan model penjadwalan *batch* dimana model tersebut memberikan kemungkinan bagi suatu pekerjaan untuk selesai mendahului *due date* dengan mempertimbangkan resiko terhadap biaya simpan, dan pengerjaan *batch* tidak boleh melampaui *due date*. Kriteria performansi yang digunakan sama dengan Halim dan Zakaria (2002) namun model yang dikembangkan berlaku untuk sistem manufaktur *flow shop*, berbeda dengan Halim dan Zakaria yang mengembangkan model untuk sistem manufaktur *job shop*.

Pengembangan model juga dilakukan berdasarkan penelitian Aisyati, *et.al* (2007) dengan menggunakan kriteria yang berbeda yaitu total biaya produksi. Ketiga penelitian yaitu Halim dan Zakaria(2002), Aisyati, *et.al* (2007), dan penelitian yang akan dilakukan lebih realistis karena mempertimbangkan faktor gangguan (*disturbance*) berupa kedatangan *order* baru.

## 2.2 Batasan Masalah dan Asumsi

Dalam pengembangan model terdapat beberapa batasan yang digunakan, yaitu:

1. Model yang dikembangkan hanya mempertimbangkan faktor gangguan (*disturbance*) yang berasal dari kedatangan pekerjaan (*job*) baru di tengah-tengah proses produksi berlangsung.
2. Biaya produksi yang terlibat adalah biaya *setup*, biaya WIP, biaya simpan produk jadi, dan biaya penolakan pekerjaan.
3. Setiap tahapan proses terdiri atas satu mesin.

Sedangkan asumsi yang digunakan dalam model yang dikembangkan adalah:

1. Bahan baku tersedia setiap saat dibutuhkan sehingga biaya simpan dari bahan baku diabaikan.
2. Waktu dan biaya *setup* untuk setiap *batch* pada mesin yang sama adalah sama.
3. Kedatangan *job* baru tidak bisa menginterupsi *batch* yang sedang dikerjakan (harus menunggu hingga pengerjaan *batch* tersebut selesai).
4. Tidak ada *batch* yang dikerjakan pada lebih dari satu mesin pada saat yang bersamaan dan tidak ada mesin yang mengerjakan lebih dari satu *batch* sekaligus pada saat yang sama.

## 2.3 Pengembangan Model

### 2.3.1 Kriteria Kinerja

Kriteria kinerja yang menjadi fungsi tujuan adalah minimasi total biaya produksi pada model penjadwalan *batch* yang diproduksi pada *flow shop* dinamis yang terdiri atas *multi item* dan dikirim pada *due date* yang bersamaan (*common due date*). Nilai variabel keputusan yang akan dicari untuk meminimumkan kriteria kinerja yang telah ditetapkan adalah jumlah *batch* ( $N$ ), ukuran *batch* ( $Q_i$ ), urutan pemrosesan *batch* yang ditandai dengan saat mulai ( $B_{i,m}$ ) dan saat selesai ( $F_{i,m}$ ). Pada model juga terdapat dua variabel biner yaitu variabel biner yang menyatakan eksistensi suatu jenis *item* pada suatu *batch* ( $r_{i,g}$ ), dan variabel biner penolakan pekerjaan ( $X_g$ ) berupa keputusan ya dan tidak. Variabel dependen dalam hal ini adalah total biaya produksi ( $TC$ ).

### 2.3.2 Parameter dan Variabel Keputusan

#### 2.3.2.1 Parameter

Sesuai dengan kriteria kinerja dan variabel keputusan yang telah ditentukan, maka parameter yang digunakan di dalam model usulan adalah sebagai berikut:

- $d$  : batas waktu penyerahan untuk semua *batch*  
 $t_{gm}$  : waktu proses *item* Jenis  $g$  di Mesin  $m$   
 $s_m$  : waktu *setup* di Mesin  $m$   
 $M$  : jumlah mesin (jumlah tahapan proses)  
 $G$  : jumlah jenis *item*  
 $n_g$  : jumlah permintaan dari *item* Jenis  $g$   
 $A_m$  : availabilitas ketersediaan kapasitas pada Mesin- $m$   
 $C_{1,m}$  : ongkos *setup* pada Mesin  $m$  per satuan waktu  
 $C_{2,g}$  : ongkos simpan *item* Jenis  $g$  yang sedang dikerjakan per satuan waktu  
 $C_{3,g}$  : ongkos simpan produk jadi dari *item* Jenis  $g$   
 $C_{4,g}$  : ongkos penolakan pekerjaan *item* Jenis  $g$

#### 2.3.2.2 Variabel Keputusan

Variabel keputusan dari model yang diusulkan adalah sebagai berikut:

- $B_{i,m}$  : saat mulai pengerjaan *batch* urutan ke- $i$  pada Mesin  $m$   
 $F_{i,m}$  : saat selesai pengerjaan *batch* urutan ke- $i$  pada Mesin  $m$   
 $Q_i$  : ukuran *batch* yang diproses pada posisi ke- $i$   
 $N$  : jumlah *batch*  
 $r_{i,g}$  : variabel biner yang menyatakan jenis *item* yang dimuat dalam *batch* urutan ke- $i$   
 $X_g$  : variabel biner penolakan pekerjaan *item* Jenis  $g$   
 $TC$  : total biaya produksi

### 2.3.3 Formulasi Model

Pengembangan model penjadwalan *batch* pada *flow shop* dinamis dikembangkan melalui dua tahap, yaitu pengembangan model penjadwalan *batch* pada *flow shop* tanpa mempertimbangkan penolakan pekerjaan (kasus statis) dan pengembangan model penjadwalan *batch* pada *flow shop* dengan mempertimbangkan penolakan pekerjaan (kasus dinamis). Model penjadwalan *batch* yang dikirim pada *common due date* dan diproses pada sistem manufaktur *flow shop* tanpa mempertimbangkan penolakan pekerjaan, selanjutnya disebut sebagai Model BFSS, dikembangkan sebagai berikut:

#### Fungsi tujuan:

Minimasi

$$TC = \left( N \cdot \sum_{m=1}^M s_m \cdot C_{1,m} \right) + \sum_{g=1}^G \sum_{i=1}^N r_{i,g} \left[ (F_{i,m} - B_{i,1}) Q_i \cdot C_{2,g} \right] + \sum_{g=1}^G \sum_{i=1}^N r_{i,g} \left[ (d - F_{i,M}) Q_i \cdot C_{3,g} \right] \quad (1)$$

#### Pembatas:

$$B_{1,1} = s_1 \quad (2)$$

$$F_{1,1} = B_{1,1} + \sum_{g=1}^G r_{1,g} (t_{g,1} \cdot Q_1) \quad (3)$$

$$B_{i,1} = F_{i-1,1} + s_1 \quad ; \quad i = 2, \dots, N \quad (4)$$

$$F_{i,1} = B_{i,1} + \sum_{g=1}^G r_{i,g} (t_{g,1} \cdot Q_i) \quad ; \quad i = 2, \dots, N \quad (5)$$

$$B_{1,m} = F_{1,m-1} \quad ; \quad m = 2, \dots, M \quad (6)$$

$$F_{1,m} = B_{1,m} + \sum_{g=1}^G r_{1,g} (t_{g,m} \cdot Q_1) \quad ; \quad m = 2, \dots, M \quad (7)$$

$$B_{i,m} = \max(F_{i,m-1}, F_{i-1,m} + s_m) \quad ; \quad i = 2, \dots, N \quad ; \quad m = 2, \dots, M \quad (8)$$

$$F_{i,m} = B_{i,m} + \sum_{g=1}^G r_{i,g} (t_{g,m} \cdot Q_i) \quad ; \quad i = 2, \dots, N \quad ; \quad m = 2, \dots, M \quad (9)$$

$$F_{N,M} \leq d \quad (10)$$

$$r_{i,g} \in \{0,1\} \quad ; \quad i = 1, \dots, N \quad ; \quad g = 1, \dots, G \quad (11)$$

$$\sum_{g=1}^G r_{i,g} = 1 \quad ; \quad i = 1, \dots, N \quad (12)$$

$$\sum_{g=1}^G r_{i,g} \cdot Q_i = n_g \quad ; \quad i = 1, \dots, N \quad (13)$$

$$N \geq G \quad (14)$$

$$Q_i \geq 1, \text{ integer} \quad (15)$$

Formulasi (1) menyatakan tujuan model yaitu meminimasi total biaya produksi. Pembatas (2) menyatakan saat mulai *batch* urutan pertama di Tahap 1. Pembatas (3) menyatakan saat selesai *batch* urutan pertama di Tahap 1. Pembatas (4) menyatakan saat mulai *batch* urutan ke-*i* di Tahap 1. Pembatas (5) menyatakan saat selesai *batch* urutan ke-*i* di Tahap 1. Pembatas (6) menyatakan saat mulai *batch* urutan pertama di Mesin *m*. Pembatas (7) menyatakan saat selesai *batch* urutan pertama di Mesin *m*. Pembatas (8) menyatakan saat mulai *batch* urutan ke-*i* di Mesin *m*. Pembatas (9) menyatakan saat selesai *batch* urutan ke-*i* di Mesin *m*. Pembatas (10) menjamin bahwa

saat selesai pengerjaan *batch* terakhir yang diproses di mesin terakhir tidak menabrak *due date*. Pembatas (11) merupakan variabel biner untuk pemilihan jenis *item* yang ditempatkan pada suatu *batch*. Pembatas (12) menjamin bahwa *batch* pada urutan ke-*i* hanya terdiri atas satu jenis *item* tertentu. Pembatas (13) menjamin bahwa jumlah *part* dari suatu jenis *item* yang diproses harus sama dengan jumlah total *part* dari jenis *item* tersebut yang harus diproduksi. Pembatas (14) menyatakan bahwa jumlah *batch* minimal sama dengan jumlah jenis *item* yang akan diproses. Pembatas (15) menjamin bahwa ukuran *batch* harus lebih besar atau sama dengan satu, serta jumlah dan ukuran *batch* bernilai integer.

Sedangkan model penjadwalan *batch* dengan mempertimbangkan penolakan pekerjaan, selanjutnya disebut sebagai Model BFSD, dikembangkan sebagai berikut:

**Fungsi tujuan:**

Minimasi

$$TC = \left( N \cdot \sum_{m=1}^M s_m \cdot C_{1,m} \right) + \sum_{g=1}^G \sum_{i=1}^N r_{i,g} \left[ (F_{i,m} - B_{i,1}) Q_i \cdot C_{2,g} \right] + \sum_{g=1}^G \sum_{i=1}^N r_{i,g} \left[ (d - F_{i,M}) Q_i \cdot C_{3,g} \right] + \sum_{g=1}^G X_g \cdot C_{4,g} \quad (16)$$

**Pembatas:**

$$B_{1,1} = A_1 + s_1 \quad (17)$$

$$F_{1,1} = B_{1,1} + \sum_{g=1}^G r_{1,g} (t_{g,1} \cdot Q_1) \quad (18)$$

$$B_{i,1} = F_{i-1,1} + s_1 \quad ; \quad i = 2, \dots, N \quad (19)$$

$$F_{i,1} = B_{i,1} + \sum_{g=1}^G r_{i,g} (t_{g,1} \cdot Q_i) \quad ; \quad i = 2, \dots, N \quad (20)$$

$$B_{1,m} = \max(F_{1,m-1}, A_m) \quad ; \quad m = 2, \dots, M \quad (21)$$

$$F_{1,m} = B_{1,m} + \sum_{g=1}^G r_{1,g} (t_{g,m} \cdot Q_1) \quad ; \quad m = 2, \dots, M \quad (22)$$

$$B_{i,m} = \max(F_{i,m-1}, F_{i-1,m} + s_m) \quad ; \quad i = 2, \dots, N \quad ; \quad m = 2, \dots, M \quad (23)$$

$$F_{i,m} = B_{i,m} + \sum_{g=1}^G r_{i,g} (t_{g,m} \cdot Q_i) \quad ; \quad i = 2, \dots, N \quad ; \quad m = 2, \dots, M \quad (24)$$

$$F_{N,M} \leq d \quad (25)$$

$$r_{i,g} \in \{0,1\} \quad ; \quad i = 1, \dots, N \quad ; \quad g = 1, \dots, G \quad (26)$$

$$\sum_{g=1}^G r_{i,g} = 1 \quad ; \quad i = 1, \dots, N \quad (27)$$

$$\sum_{g=1}^G r_{i,g} \cdot Q_i = n_g \quad ; \quad i = 1, \dots, N \quad (28)$$

$$X_g \begin{cases} 1, & \text{jika } \sum_{i=1}^N r_{i,g} = 0 \quad ; \quad g = 1, \dots, G \\ 0, & \text{jika lainnya} \end{cases} \quad (29)$$

$$N \geq G \quad (30)$$

$$Q_i \geq 1, \text{ integer} \quad (31)$$

Formulasi (16) menyatakan tujuan model yaitu meminimasi total biaya produksi. Pembatas (17) menyatakan saat mulai *batch* urutan pertama di Tahap 1. Pembatas (18) menyatakan saat selesai *batch* urutan pertama di Tahap 1. Pembatas (19) menyatakan saat mulai *batch* urutan ke-*i* di Tahap 1. Pembatas (20) menyatakan saat selesai *batch* urutan ke-*i* di Tahap 1. Pembatas (21) menyatakan saat mulai *batch* urutan pertama di Mesin *m*. Pembatas (22) menyatakan saat selesai *batch* urutan pertama di Mesin *m*. Pembatas (23) menyatakan saat mulai *batch* urutan ke-*i* di Mesin *m*. Pembatas (24) menyatakan saat selesai *batch* urutan ke-*i* di Mesin *m*. Pembatas (25) menjamin bahwa saat selesai pengerjaan *batch* terakhir yang diproses di mesin terakhir tidak menabrak *due date*. Pembatas (26) merupakan variabel biner untuk pemilihan jenis *item* yang ditempatkan pada suatu *batch*. Pembatas (27) menjamin bahwa *batch* pada urutan ke-*i* hanya terdiri atas satu jenis *item* tertentu. Pembatas (28) menjamin bahwa jumlah *part* dari suatu jenis *item* yang diproses harus sama dengan jumlah total *part* dari jenis *item* tersebut yang harus diproduksi. Pembatas (29) merupakan variabel biner untuk menentukan penerimaan atau penolakan pekerjaan baru. Pembatas (30) menyatakan bahwa jumlah *batch* minimal sama dengan jumlah jenis *item* yang akan diproses. Pembatas (31) menjamin bahwa ukuran *batch* harus lebih besar atau sama dengan satu, serta jumlah dan ukuran *batch* bernilai integer.

### 3. ALGORITMA PENCARIAN SOLUSI

Algoritma pencarian solusi dikembangkan untuk menyelesaikan permasalahan dimana algoritma ini secara garis besar terdiri atas dua jenis yaitu Algoritma Statis (BFSS) untuk menyelesaikan permasalahan penjadwalan di awal periode penjadwalan (belum melibatkan gangguan), dan Algoritma Dinamis (BFSD) untuk mengakomodasi kedatangan pekerjaan baru di tengah proses produksi berlangsung. Kedua algoritma ini menggunakan Sub Algoritma pemecahan *batch* untuk mengenumerasi jumlah dan ukuran *batch* serta jenis *item* yang ditempatkan dalam setiap *batch*.

#### 4.1 Algoritma Statis (BFSS)

Algoritma statis dikembangkan untuk menyelesaikan permasalahan penjadwalan *batch* dalam kondisi tidak ada gangguan berupa kedatangan pekerjaan baru. Algoritma pencarian solusi mengikuti aturan sebagai berikut :

- Langkah 1 : Siapkan data awal yang merupakan parameter model, yaitu  $d, t_{gm}, s_m, M, G, n_g, C_{1,m}, C_{2,g}, C_{3,g}$ .
- Langkah 2 : Tetapkan jumlah *batch* sama dengan jumlah jenis item ( $N = G$ ).
- Langkah 3 : Selesaikan permasalahan menggunakan Model BFSS.
- Langkah 4 : Lakukan pemecahan *batch* menggunakan Sub Algoritma Pemecahan *Batch*.
- Langkah 5 : Tetapkan jadwal yang menghasilkan total biaya produksi terkecil sebagai jadwal terpilih.
- Langkah 6 : Selesai.

#### 4.2 Algoritma Dinamis (BFSD)

Algoritma dinamis digunakan untuk menyelesaikan permasalahan penjadwalan pekerjaan baru yang kedatangannya pada saat proses produksi sedang berlangsung. Secara umum algoritma pencarian solusi ini disusun sebagai berikut:

1. Pengecekan ketersediaan kapasitas dengan menghitung pekerjaan sisa yang sedang dikerjakan dan menentukan saat mulai yang paling memungkinkan untuk melakukan penjadwalan ulang.
2. Mengelompokkan pekerjaan baru yang sejenis dengan pekerjaan lama dan menghitung total jumlah produksi dari masing-masing jenis *item* yang akan dikerjakan.
3. Melakukan penjadwalan ulang.

Adapun secara rinci tahap demi tahap algoritma pencarian solusi penjadwalan *batch* pada *flow shop* dinamis adalah sebagai berikut:

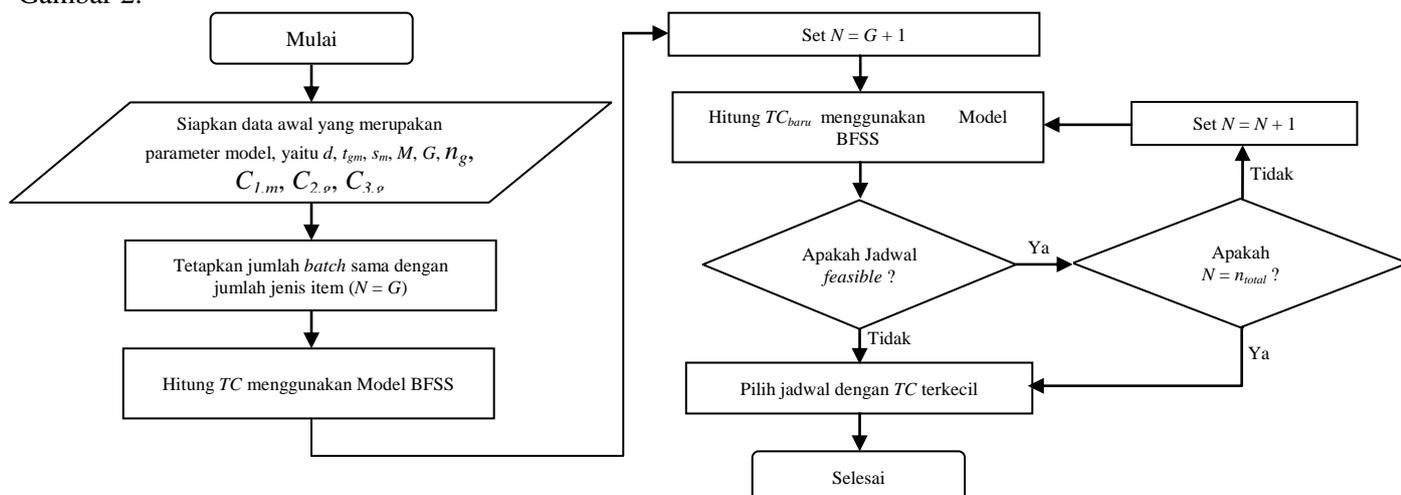
- Langkah 1 : Identifikasi pekerjaan (*batch*) yang belum selesai diproses.
- Langkah 2 : Tetapkan saat selesai pengerjaan *batch* yang sedang dikerjakan di setiap mesin sebagai availabilitas kapasitas mesin ( $A_m$ ).
- Langkah 3 : Cek pekerjaan sisa yang identik dengan pekerjaan baru dan gabungkan dalam satu kelompok pekerjaan.
- Langkah 4 : Siapkan data baru (data gabungan dari sisa pekerjaan lama dan pekerjaan baru) yang merupakan parameter model, yaitu  $d, t_{gm}, s_m, M, G, n_g, C_{1,m}, C_{2,g}, C_{3,g}, C_{4,g}$ .
- Langkah 5 : Tetapkan jumlah *batch* sama dengan jumlah jenis item ( $N = G$ ).
- Langkah 6 : Selesaikan permasalahan menggunakan Model BFSD.
- Langkah 7 : Lakukan pemecahan *batch* menggunakan Sub Algoritma Pemecahan *Batch*.
- Langkah 8 : Tetapkan jadwal yang menghasilkan total biaya produksi terkecil sebagai jadwal terpilih.
- Langkah 9 : Periksa kelayakan pekerjaan. Jika seluruh pekerjaan baru tertolak kembali ke jadwal semula. Jika terdapat pekerjaan baru yang layak, revisi jadwal produksi berdasarkan hasil penjadwalan ulang.
- Langkah 10 : Selesai.

### 3.3 Sub Algoritma Pemecahan *Batch*

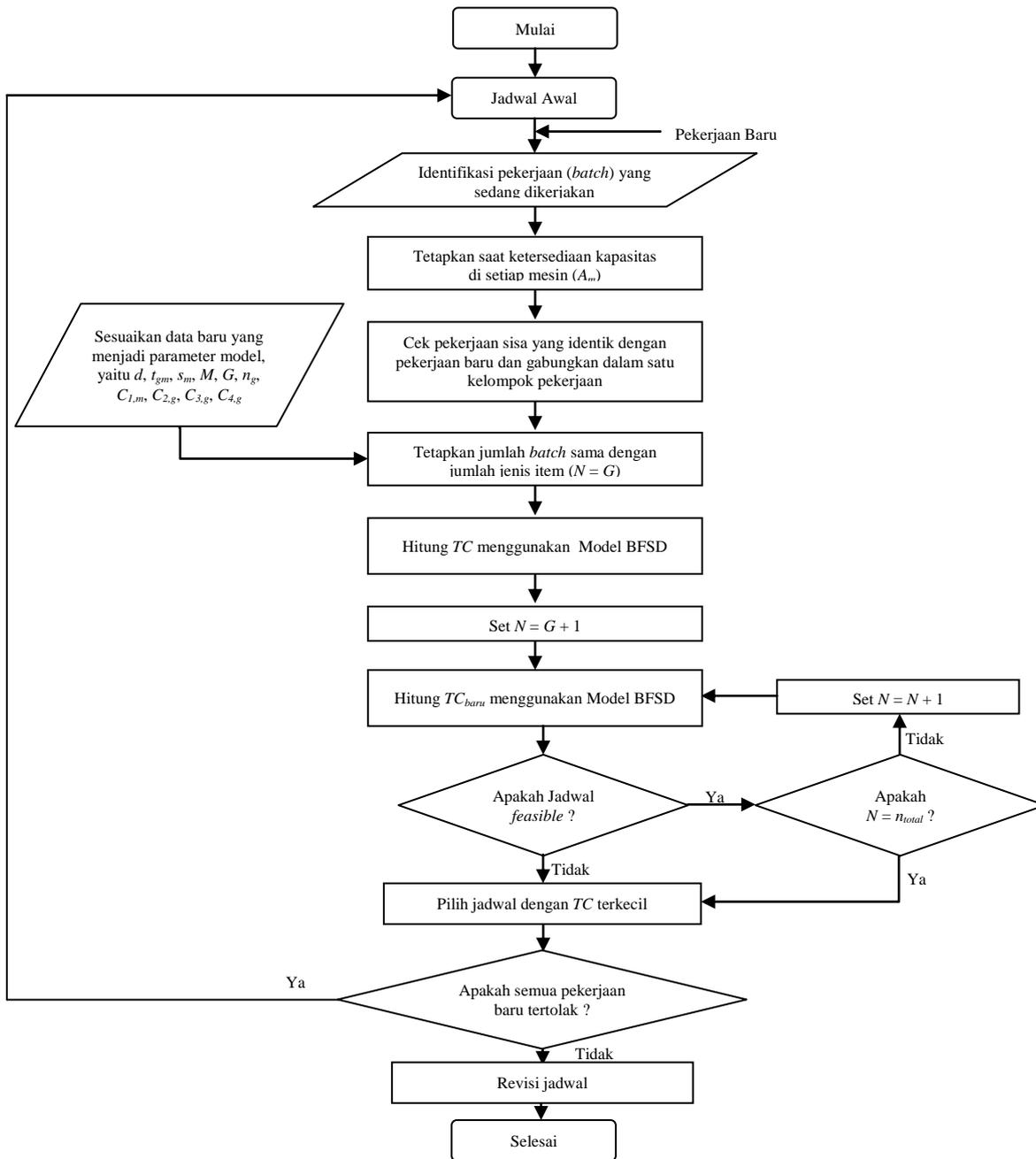
Pada model yang dikembangkan, jumlah dan ukuran *batch* merupakan variabel keputusan yang akan dicari nilainya. Pada pencarian solusi awal ditetapkan jumlah *batch* sama dengan jumlah jenis *item*. Kemudian dilakukan pemecahan *batch* secara bertahap hingga dicapai jumlah *batch* sama dengan total jumlah produk. Secara lengkap Algoritma Pemecahan *batch* disusun sebagai berikut:

- Langkah 1 : Set  $N = G + 1$ .
- Langkah 2 : Selesaikan permasalahan menggunakan formulasi matematis.
- Langkah 3 : Catat total biaya produksi yang dihasilkan.
- Langkah 4 : Apakah saat penyelesaian *batch* terakhir lebih besar dari *due date*? Jika ya hentikan pencarian solusi, lanjut ke Langkah 7. Jika tidak lanjut ke Langkah 5.
- Langkah 5 : Apakah  $N = \sum_{g=1}^G n_g$  ? Jika ya lanjut ke Langkah 7. Jika tidak lanjut ke Langkah 6.
- Langkah 6 : Set  $N = N + 1$ . Kembali ke Langkah 2.
- Langkah 7 : Pilih jadwal dengan total biaya produksi paling kecil.
- Langkah 8 : Selesai.

Secara lengkap alur penggunaan Algoritma BFSS dan BFSD dapat dilihat pada Gambar 1 dan Gambar 2.



**Gambar 1. Flow Chart Algoritma BFSS**



**Gambar 2. Flow Chart Algoritma BFSD**

#### 4. CONTOH NUMERIK

Penggunaan model dan algoritma yang dikembangkan dapat dilihat pada contoh kasus sebagai berikut:

##### 1. Kasus Statis

Terdapat 2 jenis produk yang akan diproduksi, yaitu produk A dan B. Produk A akan diproduksi sebanyak 5 unit dan produk B sebanyak 10 unit. Kedua produk akan diproses pada sistem manufaktur *flow shop* 3 tahap dimana masing-masing tahap terdiri atas mesin tunggal. *Setup* dilakukan pada masing-masing mesin dengan waktu masing-masing adalah 20 menit, 25 menit, dan 10 menit. Kedua jenis produk harus selesai pada *due date* yang sama yaitu 800 menit. Data waktu proses dari kedua jenis produk dan Biaya *setup* pada setiap mesin disajikan pada Tabel 2. Sedangkan data biaya WIP dan biaya simpan disajikan pada Tabel 3.

**Tabel 2. Data Waktu Proses dan Biaya Setup Kasus Statis**

Mesin	Waktu Proses		Waktu Setup	Biaya Setup
	Produk A	Produk B		
1	4	6	20	250
2	10	12	25	300
3	8	10	10	200

**Tabel 3. Data Biaya WIP dan Biaya Simpan Kasus Statis**

Jenis Biaya	Jenis Produk	
	Produk A	Produk B
Biaya WIP	50	80
Biaya Simpan	100	140

**Penyelesaian:**

Proses pencarian solusi adalah sebagai berikut:

- Langkah 1 : Siapkan data awal yang merupakan parameter model, yaitu  $d=1000$ ,  $t_{1,1} = 4$ ,  $t_{1,2} = 10$ ,  $t_{1,3} = 8$ ,  $t_{2,1} = 6$ ,  $t_{2,2} = 12$ ,  $t_{2,3} = 10$ ,  $s_1 = 20$ ,  $s_2 = 25$ ,  $s_3 = 10$ ,  $M = 3$ ,  $G = 2$ ,  $n_1 = 5$ ,  $n_2 = 10$ ,  $C_{1,1} = 250$ ,  $C_{1,2} = 300$ ,  $C_{1,3} = 200$ ,  $C_{2,1} = 50$ ,  $C_{2,2} = 80$ ,  $C_{3,1} = 100$ ,  $C_{3,2} = 140$ .
- Langkah 2 : Tetapkan jumlah *batch* sama dengan jumlah jenis item ( $N = G$ ). Pada kasus ini Terdapat 2 jenis produk, sehingga  $N = 2$ .
- Langkah 3 : Selesaikan permasalahan menggunakan Model BFSS. Total biaya yang dihasilkan adalah Rp. 1.699.250,- dengan jadwal yang diperoleh adalah sebagai berikut:
- Langkah 4 : Lakukan pemecahan *batch* menggunakan Sub Algoritma Pemecahan *Batch*. Solusi yang diperoleh adalah sebagai berikut:

**Tabel 5. Total Biaya Produksi Setiap *Batch* (Kasus Statis)**

Jumlah <i>Batch</i> ( $N$ )	Total Biaya Produksi ( $TC$ )	Jumlah <i>Batch</i> ( $N$ )	Total Biaya Produksi ( $TC$ )
3	1.692.000	10	1.612.800
4	1.677.350	11	1.613.000
5	1.655.450	12	1.616.060
6	1.640.000	13	1.621.980
7	1.629.360	14	1.630.760
8	1.620.980	15	1.642.400
9	1.615.460		

- Langkah 5 : Tetapkan jadwal yang menghasilkan total biaya produksi terkecil sebagai jadwal terpilih. Berdasarkan Tabel 5 diperoleh Total Biaya Produksi terendah pada saat *batch* dipecah menjadi 10 *batch* dengan jadwal sebagai berikut:

**Tabel 6. Solusi Optimal untuk Kasus Statis**

Batch ke-	Ukuran Batch	Jenis Produk	Saat Mulai		Saat Selesai	Batch ke-	Ukuran Batch	Jenis Produk	Saat Mulai		Saat Selesai
			Mesin	$B_{im}$					Mesin	$B_{im}$	
1	1	A	1	20	24	6	1	B	1	140	146
			2	24	34				2	199	211
			3	34	42				3	211	221
2	1	A	1	44	48	7	1	B	1	166	172
			2	59	69				2	236	248
			3	69	77				3	248	258
3	1	A	1	68	72	8	1	B	1	192	198
			2	94	104				2	273	285
			3	104	112				3	285	295
4	1	A	1	92	96	9	1	B	1	218	224
			2	129	139				2	310	322
			3	139	147				3	322	332
5	1	A	1	116	120	10	6	B	1	244	280
			2	164	174				2	347	419
			3	174	182				3	419	479

- Langkah 6 : Selesai.

**2. Kasus Dinamis**

Berdasarkan pada contoh kasus statis di atas, jika pada menit ke 255 terjadi kedatangan *order* baru berupa pekerjaan *item* jenis B sebanyak 5 unit dan *item* jenis C sebanyak 8 unit dengan data *order* sebagai berikut:

**Tabel 7. Data Waktu Proses dan Biaya Setup Kasus Dinamis**

Mesin	Waktu Proses Produk C	Waktu Setup	Biaya Setup
1	3	10	250
2	8	15	300
3	5	20	200

**Tabel 8. Data Biaya WIP dan Biaya Simpan Kasus Dinamis**

Jenis Biaya	Produk C
Biaya WIP	30
Biaya Simpan	120

Adapun biaya penolakan pekerjaan baru adalah Rp. 500.000,- untuk *item* jenis B dan Rp. 250.000,- untuk *item* jenis C.

**Penyelesaian:**

Proses pencarian solusi adalah sebagai berikut:

- Langkah 1 : Identifikasi pekerjaan (*batch*) yang belum selesai diproses. Berdasarkan jadwal terpilih pada Tabel 6 dapat dilihat bahwa pada menit ke 255 sedang dilaksanakan pemrosesan *batch* ke-7 di mesin 3. Dengan demikian terdapat 4 *batch* yang belum selesai diproses.
- Langkah 2 : Tetapkan saat selesai pengerjaan *batch* yang sedang dikerjakan di setiap mesin sebagai availabilitas kapasitas mesin ( $A_m$ ). Jika dilihat pada jadwal, pengerjaan *batch* ke-7 yang belum selesai terdapat pada mesin 3. Pekerjaan tersebut tidak bisa diinterupsi (harus menunggu hingga pengerjaan *batch* ke-7 selesai) sehingga saat paling memungkinkan untuk memulai pengerjaan pekerjaan sisa dan pekerjaan baru pada masing-masing mesin adalah sebagai berikut:

**Tabel 9. Availabilitas Mesin**

Mesin ke-	Saat Mulai ( $A_m$ )
1	172
2	248
3	258

- Langkah 3 : Cek pekerjaan sisa yang identik dengan pekerjaan baru dan gabungkan dalam satu kelompok pekerjaan. Dalam hal ini pekerjaan baru yang identik dengan pekerjaan lama yang belum selesai dikerjakan adalah *item* jenis B. Kuantitas gabungan dari kedua produk adalah sebagai berikut:

**Tabel 10. Kuantitas Produk Gabungan**

Order Lama	Order Baru	Jumlah
8	5	13

- Langkah 4 : Siapkan data baru (data gabungan dari sisa pekerjaan lama dan pekerjaan baru) yang merupakan parameter model, yaitu  $d=1000$ ,  $t_{1,1} = 6$ ,  $t_{1,2} = 12$ ,  $t_{1,3} = 10$ ,  $t_{2,1} = 3$ ,  $t_{2,2} = 8$ ,  $t_{2,3} = 5$ ,  $s_1 = 20$ ,  $s_2 = 25$ ,  $s_3 = 10$ ,  $M = 3$ ,  $G = 2$ ,  $n_1 = 13$ ,  $n_2 = 8$ ,  $C_{1,1} = 250$ ,  $C_{1,2} = 300$ ,  $C_{1,3} = 200$ ,  $C_{2,1} = 80$ ,  $C_{2,2} = 30$ ,  $C_{3,1} = 140$ ,  $C_{3,2} = 120$
- Langkah 5 : Tetapkan jumlah *batch* sama dengan jumlah jenis *item* ( $N = G$ ). Pada kasus ini Terdapat 2 jenis produk, sehingga  $N = 2$ .
- Langkah 6 : Selesaikan permasalahan menggunakan Model BFSD. Total biaya yang dihasilkan adalah Rp. 1.873.420,-.

Langkah 7 : Lakukan pemecahan *batch* menggunakan Sub Algoritma Pemecahan *Batch*. Solusi yang diperoleh adalah sebagai berikut:

**Tabel 11. Total Biaya Produksi Setiap *Batch* (Kasus Dinamis)**

Jumlah <i>Batch</i> (N)	Total Biaya Produksi (TC)	Jumlah <i>Batch</i> (N)	Total Biaya Produksi (TC)
3	1.872.380	13	1.727.020
4	1.873.560	14	1.715.860
5	1.879.120	15	1.714.820
6	1.882.580	16	1.716.000
7	1.854.040	17	1.719.400
8	1.825.720	18	1.823.460
9	1.800.260	19	1.732.860
10	1.777.660	20	1.742.920
11	1.757.920	21	1.755.200
12	1.741.040		

Langkah 8 : Tetapkan jadwal yang menghasilkan total biaya produksi terkecil sebagai jadwal terpilih. Berdasarkan Tabel 11 diperoleh Total Biaya Produksi terendah pada saat *batch* dipecah menjadi 15 *batch* dengan jadwal sebagai berikut:

**Tabel 12. Solusi Optimal untuk Kasus Dinamis**

Batch ke-	Ukuran Batch	Jenis Produk	Saat Mulai		Saat Selesai	Batch ke-	Ukuran Batch	Jenis Produk	Saat Mulai		Saat Selesai	
			Mesin	$B_{im}$					Mesin	$B_{im}$		
1	1	B	1	192	1	9	1	B	1	228	234	
			2	273					285	2	569	581
			3	285					295	3	581	591
2	1	B	1	46	10	1	B	1	254	260		
			2	310				322	2	606	618	
			3	322				332	3	618	628	
3	1	B	1	72	11	1	B	1	280	286		
			2	347				359	2	643	655	
			3	359				369	3	655	665	
4	1	B	1	98	12	1	B	1	306	312		
			2	384				396	2	380	692	
			3	396				406	3	692	702	
5	1	B	1	124	13	1	B	1	332	338		
			2	421				433	2	717	729	
			3	433				443	3	729	739	
6	1	B	1	150	14	1	C	1	358	361		
			2	458				470	2	754	762	
			3	470				480	3	762	767	
7	1	B	1	176	15	7	C	1	381	402		
			2	495				507	2	787	843	
			3	507				517	3	843	878	
8	1	B	1	202								
			2	532				544				
			3	544				554				

Langkah 9 : Periksa kelayakan pekerjaan. Pada kasus ini seluruh pekerjaan lama dan baru layak dikerjakan, sehingga perlu dilakukan revisi jadwal produksi berdasarkan hasil penjadwalan ulang pada Tabel 12.

Langkah 10 : Selesai.

### 3. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Model yang dikembangkan mampu menyelesaikan permasalahan penjadwalan *batch* pada sistem manufaktur *flowshop* dinamis yang memproduksi lebih dari satu jenis *item*. Penentuan urutan pemrosesan jenis *item*, *batching*, dan *sequencing* dapat diselesaikan secara simultan.

2. Model yang dikembangkan mampu mengakomodasi kedatangan pekerjaan baru di saat proses produksi sedang berlangsung dengan keputusan penerimaan atau penolakan pekerjaan dan menghasilkan jadwal revisi yang meminimasi total biaya produksi.
3. Peningkatan waktu *setup* lebih berpengaruh terhadap penentuan jumlah *batch* yang dihasilkan serta total biaya produksi dibandingkan peningkatan waktu proses.
4. Peningkatan jumlah jenis *item* dan jumlah tahapan (mesin) memberikan dampak yang signifikan terhadap waktu komputasi. Dampak dari jumlah jenis *item* terlihat lebih nyata dibandingkan dampak penambahan jumlah mesin.
5. Penjadwalan *batch* terbukti memberikan solusi yang lebih baik dibandingkan dengan penjadwalan *job* namun meningkatkan kompleksitas permasalahan yang berpengaruh terhadap waktu komputasi.

Adapun keterbatasan dari model yang dikembangkan dalam penelitian ini menghasilkan celah untuk penelitian lanjutan antara lain:

1. Pengembangan model yang melibatkan biaya *penalty* keterlambatan pengerjaan produk (*tardiness*).
2. Pengembangan model dengan *due date* yang berbeda (*multiple due date*) dan waktu *setup* yang berbeda terhadap masing-masing produk sehingga lebih mendekati kondisi nyata.
3. Pengembangan model dengan memasukkan faktor-faktor gangguan lainnya seperti pembatalan order atau ketidaktersediaan sumber daya.
4. Pengembangan algoritma heuristik untuk efisiensi waktu komputasi.

## References

### Journal:

- [1] Halim, A.H., Ohta, H. Batch Scheduling Problems of Multiple Items through The Flow Shop with Both Receiving and Delivery Just in Times. *International Journal of Production Research*. 1993; 31: 1943-1955.
- [2] Setiawati, L., Halim, A.,H., Model Penjadwalan Flow Shop 2 Mesin dengan Interval Ketidaktersediaan untuk Meminimasi Mean Earliness. *Jurnal TMI*. 2003; 23(2): 27-39.
- [3] Halim, A.,H., Barnali, L., Model Penjadwalan Batch Dinamis pada Flow Shop untuk Kasus Multi Item dengan Kriteria Waktu Tinggal Aktual. *Jurnal TMI*. 1998; 18(1): 14-25.

### Thesis/Disertation:

- [1] Cahya, B.I. Model Overlap Operasi (Operation Overlapping) untuk Kasus Multi-Batch dan Multi Item. Tugas Akhir. Bandung: Fakultas Teknik Industri ITB; 1998.
- [2] Kurniawan, D. Model Penjadwalan Batch untuk Meminimumkan Total Flow Time pada Sistem Produksi Flow Shop. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 2003.
- [3] Suryadhini, P.P. Model Penjadwalan Batch pada Flow Shop dengan Mesin Umum pada Tahap Satu dan Mesin Unik pada Tahap Dua untuk Meminimasi Total Actual Flow Time. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 2006.
- [4] Tejaasih, I. Pengembangan Model Penjadwalan Batch Sumber Serentak untuk Sistem Produksi Flow Shop. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 1998.
- [5] Wahyuni, S. Penjadwalan Flow Shop Kelompok Mesin Heterogen untuk Meminimasi Total Actual Flow Time. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 2001.
- [6] Zaini, E. Model Penjadwalan Batch pada Flow Shop Dinamis Sumber Simultan dengan Kriteria Minimasi Total Actual Flow Time. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 1999.