

Batch Scheduling Model For a Flow Shop Two Stage to Minimize Total Actual Flow Time

Model Penjadwalan *Batch* Pada *Flowshop* Dua Tahap Untuk Meminimumkan Total Waktu Tinggal Aktual

Nisa Noviani Sudarman ¹⁾

¹⁾Masoem University, Jalan Raya Cipacing No 22, kec. Jatinangor, kab. Sumedang, 45363

Email: nisasudarman@gmail.com

*) *Corresponding author*

Abstract: *This research concerns a batch scheduling problem with the process time used in the existing batch scheduling research can vary. One of the factors influencing processing time is machine deterioration and learning-forgetting process. The effect of the deterioration can results in a longer process time due to decreased machine capability, as well as an increase in service life or usage. The effect of the learning process can result in a faster product processing time due to an increase in operator experience in processing the product. Meanwhile, the effect of the forgetting process can result in slower product processing times due to the lag time between processing of the same product. This research proposes a batch scheduling model for a flow shop with a processing tow-item with considering of the deterioration process in Stage One and learning-forgetting process in Stage Two. The objectives is to minimize total actual flow time. The decision variables in this research are number of batch (N), batch size (Q_i), and batch processing sequence. The problem-solving method was developed by proposed heuristic algorithms. The proposed algorithm is consist of batching sub algorithm, and sequence sub algorithm. Numerical testing shows that the proposed algorithm is able to solve the problem of a batch scheduling for a flow shop to minimize total actual flow time and the solution obtained by the algorithm is a feasible solution.*

Keywords: *scheduling, batch, flow shop, common due date, actual flow time.*

Abstrak: Pada penelitian mengkaji terkait dengan permasalahan penjadwalan batch dengan waktu proses yang dapat berubah. Waktu proses yang dapat berubah diakibatkan karena adanya pengaruh dari deteriorasi mesin dan proses belajar-lupa dari operator. Akibat dari efek deteriorasi mesin ini mempengaruhi waktu proses yang menjadi lebih lama karena adanya penurunan kemampuan mesin seiring dengan bertambahnya umur pakai, sedangkan akibat dari adanya efek proses belajar operator adalah waktu proses produksi menjadi lebih cepat diakibatkan oleh meningkatkan pengalaman operator dalam memproses produk tersebut. Sedangkan akibat dari adanya efek proses lupa adalah waktu proses produksi suatu produk menjadi lebih lambat karena adanya waktu jeda di antara pemrosesan suatu produk yang sama untuk mengingat kembali jalan proses produksinya. Model dalam penelitian ini terkait dengan model penjadwalan batch pada flowshop dua tahap yang memproses dua item dengan memperhatikan proses deteriorasi mesin pada Tahap Satu dan proses belajar-lupa pada Tahap Dua. Kriteria tujuan dari penelitian ini adalah meminimumkan total waktu tinggal aktual. Variabel keputusan penelitian ini adalah jumlah batch (N), ukuran batch (Q_i), dan urutan pemrosesan batch. Pemecahan masalah pada penelitian ini adalah dengan mengusulkan sebuah algoritma heuristic. Hasil pengujian yang dilakukan pada penelitian ini adalah secara numerik dengan menunjukkan bahwa model mampu menyelesaikan permasalahan penjadwalan batch pada flowshop dua tahap untuk meminimumkan total waktu tinggal aktual dengan solusi yang dihasilkan merupakan solusi layak.

Kata Kunci: penjadwalan, *batch, flow shop, due date* Tunggal, waktu tinggal aktual.

DOI: <http://dx.doi.org/10.37577/sainteks.v6i02.646>

Received: 03, 2024. Accepted: 07, 2024.

Published: 09, 2024

PENDAHULUAN

Seiring dengan berjalannya waktu, permasalahan penjadwalan pada umumnya menjadi sangat kompleks. Kompleksitas penjadwalan bisa terjadi antara lain karena struktur produk, jumlah mesin di lantai produksi dan lainnya. Sehingga, banyak aturan dan algoritma yang dikembangkan dalam menjadwalkan produksi, akan tetapi untuk memperoleh algoritma terbaik yang dapat menghasilkan solusi optimal ataupun algoritma yang efisien tidaklah mudah (Leung, 2004). Permasalahan penjadwalan yang paling populer salah satunya adalah penjadwalan *flow shop* karena peningkatan penggunaan praktisnya di dalam industri manufaktur (Nakhaeinejad & Nahavandi, 2013). Masalah dalam penjadwalan *flow shop* secara teoritis sebagian besar masih belum terpecahkan sehingga perlu dilakukan penelitian yang dapat menjembatani gap diantara pengembangan teori dan aplikasinya. Penelitian mengenai penjadwalan *flow shop* telah banyak dilakukan, diantaranya oleh Pranzo (2003) mengembangkan penjadwalan *flow shop* dua mesin, (Behnamian et al., 2012) mengembangkan pada masalah penjadwalan *flow shop* tiga mesin, (Damodaran et al., 2013) mengembangkan *permutation flow shop*, dan (Cheng et al., 2015) mengembangkan penjadwalan *batch* pada *flow shop* dua tahap untuk dua jenis produk.

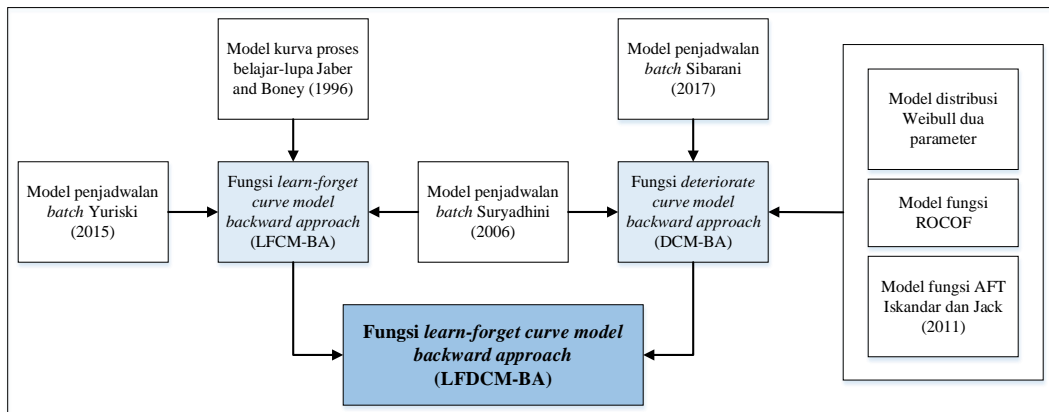
Kriteria performansi yang digunakan pada penelitian yang disebutkan tersebut belum mempertimbangkan *due date* padahal pada kenyataannya pemenuhan *due date* yang telah disepakati bersama pelanggan menjadi lebih penting karena pemenuhan *due date* yang konsisten akan menjaga kesetiaan pelanggan (Halim, 1993). Selain itu, pengiriman produk sesuai dengan jadwal yang telah ditentukan merupakan faktor penting dalam memenangkan persaingan (Halim et al., 1994). Salah satu yang dapat menghambat ketepatan jadwal pengiriman adalah karena adanya efek deteriorasi mesin yang bisa menyebabkan keterlambatan pengiriman akibat waktu proses yang lambat (Janiak et al., 2011). Selain efek deteriorasi yang dapat menghambat ketepatan jadwal pengiriman adalah proses belajar-lupa operator. Efek proses belajar dapat mengakibatkan waktu proses produk menjadi lebih cepat yang disebabkan oleh peningkatan pengalaman operator dalam memproses produk tersebut seiring dengan jumlah produk sejenis yang telah dihasilkannya (Wright, 1936). Sedangkan efek proses lupa dapat mengakibatkan waktu proses produk menjadi lebih lambat yang disebabkan oleh waktu jeda di antara pemrosesan produk yang sama (Carlson, 1976). Jeda antara proses produk yang sama ini menyebabkan operator harus mengingat kembali operasi pengerjaan produk tersebut.

METODOLOGI

Pada tahap ini dilakukan pengembangan model penjadwalan batch pada flow shop dua tahap yang memproses dua item. Pemilihan model acuan ini didasarkan atas kesamaan taksonomi penelitian dari masing-masing penelitian. Terdapat tiga model yang akan dijadikan sebagai model acuan, yaitu :

1. Model penjadwalan batch pada flowshop dengan mesin umum (*common machine*) pada Tahap 1 dan mesin khusus (*dedicated machine*) pada Tahap 2 dengan kriteria minimasi *actual flowtime* yang dikembangkan oleh (Suryadhini, 2015).
2. Model penjadwalan batch pada single machine dengan efek belajar-lupa operator untuk kasus single item dengan common due date dengan kriteria minimasi actual flowtime yang dikembangkan oleh (Yusriski et al., 2015).
3. Model penjadwalan *batch* pada *flow shop* dengan mesin terdeteriorasi yang memroses multi item dengan *common due date* yang dikembangkan oleh (Sibarani, 2016).

Ketiga model ini akan menjadi dasar untuk pengembangan model penjadwalan yang diusulkan dalam penelitian ini. Hubungan antara model acuan terhadap pengembangan model dapat dilihat pada Gambar 1 tersebut.



Gambar 1 Hubungan Model Konseptual Pengembangan Kurva Waktu Proses

HASIL PENELITIAN DAN PEMBAHASAN

Permasalahan dalam penelitian ini adalah menentukan jumlah, ukuran *batch* serta urutan *batch*. Parameter yang digunakan adalah waktu proses per unit, waktu setup antar *batch*, jumlah permintaan dari masing-masing item, jumlah unit yang akan dijadwalkan, parameter deteriorasi mesin, parameter proses belajar-lupa operator, dan saat penyerahan seluruh unit (*due date*). Asumsi penyerahan seluruh unit pada penelitian adalah dilakukan bersamaan (*common due date*), dan waktu setup antar *batch* diasumsikan hanya tergantung pada jenis item dan mesin, tidak tergantung pada ukuran *batch*. Notasi lengkap dalam penelitian ini adalah sebagai berikut :

Indeks Model

- i, j = Indeks yang menyatakan urutan *batch*;
- G = Indeks yang menyatakan jenis *item*.

Parameter Model

- D = Batas waktu penyerahan untuk semua *batch* (waktu);
- t_0 = Waktu proses inisial perunit pada mesin umum (waktu/unit);
- $t_g[M]$ = Waktu proses inisial *item* jenis g pada mesin unik (waktu/unit);
- S = Waktu *setup*;
- G = Jumlah jenis *item*;
- N_g = Jumlah permintaan dari *item* jenis g (unit);
- γ_0 = Parameter AFT pada mesin umum;
- α_{00} = Diasumsikan bahwa α_0 adalah kegagalan pertama dari mesin umum (ekspektasi sisa umur mesin inisial);
- β_0 = Parameter bentuk mesin umum, menyatakan gradient *Weibull*, diasumsikan bahwa β adalah parameter deteriorasi mesin ($\beta > 1$);
- M = Gradien proses belajar;
- Q = Minimal jumlah produksi untuk mencapai tahap keadaan tunak;
- δ = Parameter proses belajar;
- t_b = Minimal waktu jeda operator sebelum mengalami efek lupa secara total.

Variabel Model

- $T_0[i]$ = Waktu proses perunit *part* di *batch* ke- i pada mesin umum yang dipengaruhi oleh proses deteriorasi mesin;
- $T_g[i]$ = Waktu proses perunit *part* di *batch* ke- i pada mesin unik yang dipengaruhi oleh proses belajar-lupa;
- $\alpha_{0[i]}$ = Ekspektasi sisa umur mesin umum pada tahap awal mesin mulai memroses *batch* ke- i (waktu/unit);
- $x_{0[i]}$ = Waktu proses maksimum perunit *part* di *batch* ke- i pada mesin umum yang mengalami kerusakan (waktu/unit);

- $E_{g[i]}$ = Pengalaman operator pada indeks posisi ke- $[i]$ yaitu setelah mengerjakan sejumlah komponen dalam *batch* ke- $[N]$ sampai dengan ke- $[i+1]$ pada pendekatan penjadwalan mundur;
- V = Ambang batas waktu proses belajar (*learning threshold*);
- $Y_{g[i]}$ = Ukuran *batch* yang dijadwalkan pada posisi ke- $[i]$, jika $\delta = 1$ maka nilai $Y_{0[i]}$ tidak dipengaruhi proses belajar-lupa, tetapi jika $\delta \neq 1$ maka nilai $Y_{0[i]}$ dipengaruhi oleh proses belajar-lupa;
- $X_{g[i]}$ = Ekspektasi jumlah produk yang dapat diproses oleh operator yang mengalami efek proses belajar dan mesin yang mengalami deteriorasi untuk *batch* ke- $[i+1]$ dan selama *setup batch* ke- $[i]$;
- $P_{g[i]}$ = Ekpektasi efisiensi mesin saat akan memproses *batch* pada indeks posisi ke- $[i]$;
- $f_{g[i]}$ = Faktor proses lupa pada posisi ke- $[i]$;
- $H_{g[i]}$ = Jumlah waktu yang dibutuhkan untuk memproduksi *batch* pada posisi ke- $[i+1]$ oleh operator yang telah mengalami proses belajar-lupa;
- $\Lambda(B_{0[i]})$ = Ekspektasi jumlah kerusakan mesin umum pada tahap awal mesin umum mulai memproses *batch* indeks ke- i ;
- $U_{0[i]}$ = Umur pemakaian mesin umum pada indeks ke- i yaitu setelah mengerjakan sejumlah komponen dalam *batch* indeks ke- N sampai dengan ke- $i+1$ pada mesin umum.

Variabel Keputusan

- B_i = Saat mulai *batch* urutan ke- i yang dimulai untuk $i = 2$;
- Q_i = Ukuran *batch* yang diproses pada posisi ke- i (*unit*);
- N = Jumlah *batch*;
- $z_{g[i]}$ = Variabel biner yang menyatakan jenis *item* yang dimuat dalam *batch* urutan ke- i .

Formulasi untuk model penjadwalan *batch* pada *flowshop* dua tahap dengan kriteria meminimasi total waktu tinggal aktual :

Minimasi

$$F^a = \left\{ \sum_{i=1}^N \sum_{j=1}^i T_{0[i]} Q_{[j]} + \max \left(s + F_{[i-1]}^a, \sum_{g=1}^G z_{g[i]} \left(\sum_{g=1}^G \sum_{j=1}^i z_{g[j]} (T_{g[j]} Q_{[j]} + s) - s \right) \right) \right\} Q_{[i]} \tag{1}$$

Fungsi (1) menyatakan tujuan model dari penelitian ini adalah meminimasi *total actual flowtime* seluruh *part* yang akan diproses dengan $B_{[i]}$ pada penelitian ini dihitung dengan cara :

$$B_{[1]} = d - (T_{0[i]} Q_{[1]} + T_{g[j]} Q_{[1]}) \tag{2}$$

$$B_{[i]} = \left(\min \left[d - (s + F_{[i-1]}^a), d - \sum_{g=1}^G z_{g[i]} \left(\sum_{g=1}^G \sum_{j=1}^i z_{g[j]} (T_{g[j]} Q_{[j]} + s) - s \right) \right] \right) - T_{0[i]} Q_{[i]} \tag{3}$$

Persamaan (2) suku pertama menyatakan batas waktu penyerahan dan suku kedua menyatakan lamanya batch pada posisi pertama berada pada rantai produksi. Sedangkan persamaan (3) untuk elemen pertama pada suku pertama menyatakan lamanya batch ke- i di rantai produksi ditambah dengan waktu setup dan elemen kedua menyatakan lamanya batch ke i di tahap ke dua sedangkan suku kedua menyatakan lamanya batch ke i di tahap ke satu.

Sehingga formulasi model menjadi :

Minimasi

$$F^a = \sum_{i=1}^N [(d - B_{[i]}) Q_{[i]}] \tag{4}$$

Fungsi (4) menyatakan tujuan model dari penelitian ini yaitu minimasi *total actual flowtime* semua *part* yang akan diproses.

Variabel bilangan biner akan bernilai 1 apabila *part* yang berada dalam *batch* $b[i]$ adalah *item* g dan bernilai 0 apabila bukan *item* g .

$$z_{g[i]} = 0 \text{ atau } 1 \quad \forall g \text{ dan } i \quad (5)$$

Terdapat beberapa kondisi yang harus dipenuhi serta membentuk beberapa pembatas agar fungsi tujuan yang dibentuk dapat tercapai, adalah sebagai berikut :

1. Pembatas ukuran *batch* harus lebih atau sama dengan 1 dan integer.

$$Q_{[i]} \geq 1, \text{ integer} \quad \forall i \quad (6)$$

2. Pembatas Jumlah *Batch Item* harus lebih dari atau sama dengan jumlah jenis produk yang akan diproses

$$N \geq G \quad (7)$$

3. Pembatas Jumlah *Part Item* yang Diproduksi

$$\sum_{i=1}^N Q_{[i]} \cdot z_{g[i]} = n_g \quad \forall g \quad (8)$$

Pembatas 8 menyatakan bahwa ukuran batch pada batch untuk jenis produk g pada urutan ke i dan menyatakan jumlah part untuk semua batch harus sama dengan jumlah part total yang harus diproduksi.

4. Pembatas *Actual Flowtime Batch*

$$F^a = T_{0[i]} Q_{[i]} + \max \left(s + F_{[i-1]}^a, \sum_{g=1}^G z_{g[i]} \left(\sum_{g=1}^G \sum_{j=1}^i z_{g[j]} (T_{g[j]} Q_{[j]} + s) - s \right) \right) \quad (9)$$

Pembatas (9) menyatakan bahwa *actual flowtime* untuk *batch* pada posisi ke i .

Fungsi tujuan minimasi F^a dapat tercapai apabila memenuhi beberapa kondisi sebagai berikut :

- 1) Seluruh *batch* harus diproses pada rantai produksi dalam rentang waktu yang tidak melampaui saat tenggat (*due date*) dan tidak mendahului $t = 0$ sebagai *earliest ready time* setelah menyelesaikan semua aktivitas yang dibutuhkan untuk pemrosesan.
- 2) Pemrosesan *batch* terakhir harus selesai tepat pada saat tenggat (*due date*). Hal ini merupakan konsekuensi dari penerapan penjadwalan mundur.

Kondisi (1) yang menyatakan bahwa seluruh *batch* harus diproses pada rantai produksi dalam rentang waktu yang tidak melampaui saat tenggat (*due date*) dan tidak mendahului $t = 0$ diperlukan pembatas sebagai berikut :

1. Pembatas yang menyatakan bahwa saat mulai *batch* pertama yang diproses harus pada saat nol atau setelah saat nol sebagai *earliest ready time* atas setelahnya. Pembatas ini merujuk pada model (Suryadhini, 2015).

$$B_{[N]} \geq 0 \quad (10)$$

2. Pembatas yang menyatakan bahwa saat mulai *batch* ke i , untuk i lebih dari 2. Pembatas ini merujuk pada model (Suryadhini, 2015).

$$B_{[i]} = \left(\min \left[d - (s + F_{[i-1]}^a), d - \sum_{g=1}^G z_{g[i]} \left(\sum_{g=1}^G \sum_{j=1}^i z_{g[j]} (T_{g[j]} Q_{[j]} + s) - s \right) \right] \right) - T_{0[i]} Q_{[i]} \quad i \geq 2 \quad (11)$$

Kondisi (2) yang menyatakan bahwa pemrosesan *batch* terakhir harus selesai tepat pada saat tenggat (*due date*). Hal ini merupakan konsekuensi dari penerapan penjadwalan mundur diperlukan pembatas sebagai berikut :

1. Pembatas yang menyatakan bahwa *batch* terakhir yang diproses harus selesai tepat pada saat tenggat (*due date*).

$$B_{[1]} + T_{0[i]}Q_{[1]} + T_{g[j]}Q_{[1]} = d \quad (12)$$

Pada penelitian ini, nilai total waktu tinggal aktual akan dipengaruhi oleh proses deteriorasi mesin pada tahap 1 dan proses belajar-lupa pada tahap 2.

A. Waktu Proses yang Dipengaruhi Deteriorasi Mesin

Fenomena deteriorasi ditunjukkan dengan menurunnya keandalan mesin atau alat seiring dengan waktu atau jumlah produk yang telah diproduksi. Menurut Kaminskiy dan Krivtsov (2008), ROCOF (*the rate of occurrence of failures*) dapat digunakan untuk mengestimasi jumlah terjadinya kerusakan alat pada selang interval $[0, t)$. dengan menggunakan distribusi Weibull dua parameter (α, β) maka ROCOF dapat dituliskan sebagai berikut :

$$\Lambda(t) = (t/\alpha)^\beta \quad (13)$$

Waktu proses dalam penelitian ini merujuk pada waktu proses yang dikembangkan pada model (Sibarani, 2016). $T_{m[i]}$ merupakan waktu proses perunit di *batch* ke-*i* pada mesin *m* yang dipengaruhi oleh deteriorasi mesin. Formulasi untuk mencari waktu proses yang dipengaruhi deteriorasi mesin yang merujuk pada model (Sibarani, 2016) dituliskan sebagai berikut :

$$T_{m[i]} = t_{m[i]}(1 + \Lambda(t)) \quad (14)$$

Pada model penelitian ini kasus yang terjadi adalah waktu proses perunit yang dipengaruhi oleh deteriorasi mesin hanya terjadi pada *common machine* sehingga dinotasikan dengan $T_{0[i]}$. Sehingga model berubah menjadi sebagai berikut :

$$T_{0[i]} = t_{0[i]}(1 + \Lambda(t)) \quad (15)$$

Pengaruh deteriorasi pada waktu proses ditunjukkan berdasarkan nilai ekspektasi jumlah kerusakan pada saat $B_{[i]}$ pada *common machine* yang dinotasikan dengan $\Lambda(B_{0[i]})$. Jika $\Lambda(B_{0[i]}) = 0$ maka mesin berada dalam kondisi terbaik. $\Lambda(B_{0[i]}) = 1$ maka mesin berada dalam kondisi rusak. Waktu proses pada interval $[0, B_{[i]})$ pada *common machine* dengan berdasarkan Persamaan 15, diformulasikan sebagai berikut :

$$T_{0[i]} = t_{0[i]}(1 + \Lambda(B_{0[i]})) \quad (16)$$

Pada saat $(\Lambda(B_{0[i]}) \geq 1)$ diasumsikan waktu proses akan menjadi dua kali dari waktu proses inisial yang dinotasikan sebagai $x_{0[i]}$. Sehingga Persamaan 16 berubah menjadi sebagai berikut :

$$T_{0[i]} = \min \{t_{0[i]}(1 + \Lambda(B_{[i]0})), x_{0[i]}\} \quad i = N, \dots, 2, 1 \quad (17)$$

Ketika pemrosesan *batch* ke-*N*, mesin belum terpengaruh deteriorasi mesin sehingga $\Lambda(B_{0[i]}) = 0$. Sehingga, waktu proses yang dipengaruhi oleh deteriorasi mesin untuk *batch* ke-*N* dapat diformulasikan sebagai berikut :

$$\begin{aligned} T_{0[N]} &= \min \{t_{0[N]}(1 + \Lambda(B_{[N]0})), x_{0[N]}\} \\ T_{0[N]} &= \min \{t_{0[N]}(1 + 0), 2t_{0[N]}\} \\ T_{0[N]} &= \min \{t_{0[N]}, 2t_{0[N]}\} \\ T_{0[N]} &= t_{0[N]} \end{aligned} \quad (18)$$

Agar mendapatkan nilai ekspektasi jumlah kerusakan pada saat $B_{[i]}$ di *common machine* yang dinotasikan dengan $\Lambda(B_{0[i]})$ maka dapat diperoleh dengan persamaan berikut :

$$\Lambda(B_{0[i]}) = \left(\frac{U_{0[i]}}{\alpha_{0[i]}}\right)^{\beta_0} \quad (19)$$

$U_{0[i]}$ yang merupakan umur pemakaian mesin setelah mengerjakan sejumlah *part* dalam *batch* ke-*i* pada *common machine*, diformulasikan sebagai persamaan berikut :

$$U_{0[i]} = \sum_{h=i}^N T_{0[h+1]}Q_{[h+1]} \quad (20)$$

Dengan i adalah indeks untuk menyatakan posisi *batch* pada *common machine* maka formulasi umur pemakaian mesin pada posisi ke- i , yaitu setelah mengerjakan sejumlah *part* dalam *batch* ke- N sampai ke - $i+ 1$. Untuk memperoleh $\alpha_{0[i]}$ yang merupakan akselesarsi kerusakan (degredasi) mesin pada interval $[0, B_{[i]})$ pada *common machine*, diformulasikan sebagai berikut:

$$\alpha_{0[i]} \approx \alpha_{00} \left(\frac{d}{d + U_{0[i]}} \right)^{\gamma_0} \quad (21)$$

Beban penggunaan pada saat $B_{[i]}$ pada *common machine* adalah $U_{0[i]}$, $\alpha_{0[i]}$ adalah parameter skala standar pada *common machine*, dengan asumsi bahwa beban penggunaan standar (γ_0) sama dengan *due date* (d) dan γ_m adalah parameter AFT di mesin m .

Fungsi waktu proses perunit di batch ke I secara lengkap dituliskan sebagai berikut :

$$T_{0[i]} = \min \{t_{0[i]} (1 + \Lambda(B_{0[i]})), x_{0[i]}\} \quad i = N, \dots, 2, 1 \quad (22)$$

Dengan :

$$T_{0[N]} = t_{0[N]} \quad (23)$$

$$T_{0[N+1]} = 0, \quad Q_{[N+1]} = 0 \quad (24)$$

$$\Lambda(B_{0[i]}) = \left(\frac{U_{0[i]}}{\alpha_{0[i]}} \right)^{\beta_0} \quad (25)$$

$$U_{0[i]} = \sum_{h=i}^N T_{0[h+1]} Q_{[h+1]} \quad (26)$$

$$\alpha_{0[i]} = \alpha_{00} \left(\frac{d}{d + U_{0[i]}} \right)^{\gamma_0} \quad (27)$$

$$\beta_0 > 1, \gamma_0 \geq 1 \quad (28)$$

$$x_{0[i]} = 2t_{0[i]} \quad (29)$$

B. Waktu Proses yang Dipengaruhi Proses Belajar-Lupa

Waktu proses *batch* yang mempertimbangkan proses belajar-lupa dalam penelitian memakai model acuan yang digunakan oleh (Yusriski et al., 2015). Waktu proses *batch* yang mengalami proses belajar-lupa dihitung berdasarkan nilai maksimum dari fungsi kurva proses belajar dan waktu proses terkecil (minimum) sebagai ambang batas waktu proses belajar (v).

$$T_{g[j]} = \max \{T_{g[N]} (1 + E_{g[j]})^{-m}, v\} \quad \text{untuk} \quad j=N, \dots, 2, 1 \quad (30)$$

Nilai pengalaman operator tersebut diformulasikan sebagai berikut :

$$E_{g[j]} = \sum_{j=1}^N Y_j \quad (31)$$

Dimana,

$$Y_{g[j]} = \begin{cases} Q_{g[j+1]}, & \delta = 1 \\ [(Q_{[j+1]})^{(m+f_{[j]})/m} (X_{g[j]})^{-f_{[j]}/m}], & 0.5 < \delta < 1 \end{cases} \quad (32)$$

Persamaan 31 menyatakan bahwa nilai $Y_{g[j]}$ dipengaruhi oleh ukuran *batch* sebelumnya ($Q_{[j+1]}$) dan jumlah komponen yang dapat diproduksi bila tidak terjadi *setup* ($X_{g[j]}$). Nilai $X_{g[j]}$ dihitung dengan formula sebagai berikut :

$$X_{g[j]} = [(Q_{[j+1]})^{1-m} + s(1 - m)/T_{g[j+1]}]^{1/(1-m)} \quad (33)$$

Dengan mengetahui nilai $X_{g[j]}$ dan $Q_{[j]}$ maka akan diketahui dampak proses lupa selama *setup* terhadap operator yang mengalami proses belajar. Dampak tersebut akan lebih kecil bila dibandingkan dengan perhitungan tanpa mempertimbangkan efek proses belajar.

Apabila waktu proses dipengaruhi oleh operator yang mengalami proses belajar dan diasumsikan tidak terdapat *setup* maka pada saat pengalaman, $E_{g[j]} = X_{g[j]}$ waktu proses $T_{g[j]}$ akan sama dengan $T_{g[B]}$. Sedangkan apabila waktu tidak mempertimbangkan proses belajar maka $T_{g[j]}$ akan

sama dengan $T_{g[L]}$ dan apabila mempertimbangkan proses belajar-lupa maka $T_{g[j]}$ akan sama dengan $T_{g[BL]}$.

Pada model kurva proses belajar-lupa, waktu proses *batch* pada posisis ke- $[M]$ tidak hanya berfungsi sebagai waktu proses inisial tetapi juga berfungsi sebagai ambang batas waktu proses maksimum. Sehingga, kurva proses belajar-lupa bergerak turun pada tahap akselerasi dan kemudian bergerak konstan setelah mencapai nilai v pada tahap tunak. Waktu proses model belajar-lupa dapat diformulasikan sebagai berikut :

$$T_{g[j]} = \max\{T_{g[N]}(1 + E_{[j]})^{-m}, v\} \text{ untuk } j=N, \dots, 2, 1 \quad (34)$$

Dengan,

$$E_{[j]} = \sum_{j=1}^N Y_j \quad (35)$$

$$Y_{[j]} = \begin{cases} Q_{[j+1]}, & \delta = 1 \\ [(Q_{[j+1]})^{(m+f_{[j]})/m} (X_{[j]})^{-f_{[j]}/m}], & 0.5 < \delta < 1 \end{cases} \quad (36)$$

$$X_{[j]} = [(Q_{[j+1]})^{1-m} + s(1-m)/T_{g[j+1]}]^{1/(1-m)} \quad (37)$$

$$m = -\log \delta / \log 2 \quad (38)$$

$$f_{[j]} = \{m(1-m) \log(Q_{[j+1]} + 1) / \log(1 + C_{[j]})\} \quad (39)$$

$$C_{[j]} = (t_b/H_{[j]}) \quad (40)$$

$$H_{[j]} = (T_{g[j+1]}/1-m)(Q_{[j+1]})^{1-m} \quad (41)$$

$$v = T_{g[N]}(q)^{-m} \quad (42)$$

C. Formulasi Lengkap

Secara lengkap model penjadwalan batch pada flow shop pada dua tahap untuk meminimumkan total waktu tinggal aktual dapat diformulasikan sebagai berikut:

Minimasi

$$F^a = \sum_{i=1}^N [\{d - B_{[i]}\} Q_{[i]}] \quad (43)$$

Pembatas

$$B_{[1]} + T_{0[1]} Q_{[1]} + T_{g[1]} Q_{[1]} = d, \quad \forall g = 1, 2, \dots, G \quad (44)$$

$$B_{[i]} = \left(\min \left[d - (s + F_{[i-1]}^a), d - \sum_{g=1}^G z_{g[i]} \left(\sum_{j=1}^G \sum_{j=1}^i z_{g[j]} (T_{g[j]} Q_{[j]} + s) - s \right) \right] \right) - T_{0[i]} Q_{[i]} \quad (45)$$

$$B_{[N]} \geq 0 \quad (46)$$

$$F_{[i]}^a = T_{0[i]} Q_{[i]} + \max \left(s + F_{[i-1]}^a, \sum_{g=1}^G z_{g[i]} \left(\sum_{j=1}^G \sum_{j=1}^i z_{g[j]} (T_{g[j]} Q_{[j]} + s) - s \right) \right) \quad (47)$$

$$z_{g[i]} = 0 \text{ atau } 1 \quad \forall g \text{ dan } i \quad (48)$$

$$Q_{[i]} \geq 1, \text{ integer} \quad \forall i \quad (49)$$

$$N \geq G \quad (50)$$

$$\sum_{i=1}^N Q_{[i]} \cdot z_{g[i]} = n_g \quad \forall i = 1, 2, \dots, N \text{ dan } g \quad (51)$$

$$T_{0[i]} = \min \{t_{0[i]} (1 + \Lambda(B_{0[i]})), x_{0[i]}\} \quad \forall i = N, \dots, 2, 1 \quad (52)$$

$$T_{0[N]} = t_{0[N]} \quad (53)$$

$$T_{0[N+1]} = 0, \quad Q_{[N+1]} = 0 \quad (54)$$

$$\Lambda(B_{0[i]}) = \left(\frac{U_{0[i]}}{\alpha_{0[i]}}\right)^{\beta_0} \quad \forall i = 1, 2, \dots, N \quad (55)$$

$$U_{0[i]} = \sum_{i=1}^N T_{0[i+1]} Q_{[i+1]} \quad \forall i = 1, 2, \dots, N \quad (56)$$

$$\alpha_{0[i]} = \alpha_{00} \left(\frac{d}{d+U_{0[i]}}\right)^{\gamma_0} \quad \forall i = 1, 2, \dots, N \quad (57)$$

$$\beta_0 > 1, \gamma_0 \geq 1 \quad (58)$$

$$x_{0[i]} = 2t_{0[i]} \quad \forall i = 1, 2, \dots, N \quad (59)$$

$$T_{g[j]} = \max\{T_{g[N]}(1 + E_{[j]})^{-m}, v\} \quad \forall j = N, \dots, 2, 1 \quad (60)$$

$$E_{[j]} = \sum_{j=1}^N Y_j \quad \forall j = 1, 2, \dots, N \quad (61)$$

$$Y_{[j]} = \begin{cases} Q_{[j+1]}, & \delta = 1 \\ \left[(Q_{[j+1]})^{(m+f_{[j]})/m} (X_{[j]})^{-f_{[j]}/m} \right], & 0.5 < \delta < 1 \end{cases} \quad \forall j = 1, 2, \dots, N \quad (62)$$

$$X_{[j]} = \left[(Q_{[j+1]})^{1-m} + s(1-m)/T_{g[j+1]} \right]^{1/(1-m)} \quad \forall j = 1, 2, \dots, N \quad (63)$$

$$m = -\log \delta / \log 2 \quad (64)$$

$$f_{[j]} = \{m(1-m) \log(Q_{[j+1]} + 1) / \log(1 + C_{[j]})\} \quad \forall j = 1, 2, \dots, N \quad (65)$$

$$C_{[j]} = (t_b / H_{[j]}) \quad \forall j = 1, 2, \dots, N \quad (66)$$

$$H_{[j]} = (T_{g[j+1]} / (1-m)) (Q_{[j+1]})^{1-m} \quad \forall j = 1, 2, \dots, N \quad (67)$$

$$v = T_{g[N]}(q)^{-m} \quad (68)$$

Fungsi tujuan (43) adalah minimasi total waktu tinggal actual seluruh part yang akan diproses. Pembatas (44) menyatakan bahwa batch terakhir yang diproses harus selesai tepat pada due date. Pembatas (45) menyatakan bahwa saat mulai batch ke i , untuk i lebih dari 2. Pembatas (46) menyatakan bahwa saat mulai batch pertama yang diproses harus pada saat nol atau setelah saat nol. Pembatas (47) menyatakan bahwa actual flowtime untuk batch pada posisi ke i . Pembatas (48) menyatakan bahwa eksistensi suatu jenis produk pada sebuah batch dan urutan batch tersebut, jika $X_{g[i]} = 1$ sehingga batch tersebut adalah batch yang berasal dari produk jenis g dan berada pada posisi i , akan tetapi apabila $X_{g[i]} = 0$ maka batch tersebut bukan merupakan batch yang berasal dari produk jenis g . Pembatas (49) menyatakan bahwa ukuran batch harus lebih atau sama dengan 1 dan integer. Pembatas (50) menyatakan bahwa jumlah batch harus lebih dari satu atau sama dengan jumlah jenis produk yang akan diproses. Pembatas (51) menyatakan bahwa ukuran batch pada batch untuk jenis produk g pada urutan ke i dan menyatakan jumlah part untuk semua batch harus sama dengan jumlah part total yang harus diproses. Pembatas (52) sampai dengan pembatas (59) menyatakan bahwa waktu proses part di batch ke i pada mesin umum dipengaruhi oleh efek deteriorasi mesin. Pembatas (60) sampai dengan pembatas (68) menyatakan bahwa waktu proses part di batch ke i pada mesin unik dipengaruhi oleh proses belajar-lupa.

D. Pengembangan Sub Algoritma Penentuan Ukuran Batch

Langkah pertama dalam merancang algoritma adalah dengan menetapkan sejumlah nilai layak untuk jumlah batch (N) dan mencari solusi ukuran batch ($Q_{[i]}$ untuk $i = 1, 2, \dots, N$) untuk setiap nilai N . Pencarian nilai N_{max} dapat dilihat pada persamaan (69).

$$N_{max} = G \quad (69)$$

Ketika arah perbaikan dilakukan dengan yang semula adalah G menjadi $G + 1$ maka akan terdapat beberapa kondisi dalam penentuan N_g . Contoh, ketika nilai $N = 3$ sedangkan jenis item hanya 2 item, maka akan terdapat 2 kondisi untuk menentukan n_g yaitu ketika item 2 dibagi menjadi 2 batch dan item 1 tetap atau sebaliknya. Maka penentuan N_g ditetapkan terlebih dahulu dengan cara :

$$N_g = i, \quad i = 1, 2, \dots, N_g \quad (70)$$

$$N_{[g+1]} = N - N_g \quad (71)$$

Setelah jumlah *batch* dapat ditentukan maka dapat dilakukan solusi untuk ukuran *batch* ($Q_{[i]}$) dengan mengadopsi persamaan penentuan ukuran pada algoritma solusi Halim dkk (1994a) sebagai berikut.

$$Q_i = \frac{n}{N} + \frac{s}{2t} (N + 1) - i \frac{s}{t}, \quad i = 1, \dots, N \quad (72)$$

$$N = -\frac{z}{s} + \left(\frac{\sqrt{z^2 + 2.n.s.t}}{s} \right) \quad (73)$$

$$\text{dengan } z = \frac{n.t}{2} + \left(\frac{n.s}{\frac{d}{t} + \frac{s}{t} - n} \right) - \frac{d}{2} - \frac{s}{2}$$

Modifikasi dilakukan dengan cara perhitungan nilai (Q_i) dalam algoritma Halim, dkk (1994a) dengan menggunakan teknik perhitungan mundur, yaitu perhitungan dimulai dari indeks posisi ke- $[M]$ menuju indeks ke- $[1]$ ($i = N, \dots, 1$). Hasil yang dikembangkan oleh Yuriski (2015) dapat dilihat pada persamaan (73).

$$Q_i = \frac{G_{[i]}}{i} + \frac{s}{2T_{[i]}} (i + 1) - i \frac{s}{T_{[i]}}, \quad i = N, \dots, 1 \quad (74)$$

$$\text{dengan } G_{[i]} = n - \sum_{k=i}^N Q_{[k+1]}$$

Modifikasi dilakukan dengan cara menambahkan indeks jenis produk pada persamaan, yaitu sistem produksi *flowshop* yang memproduksi *multi item* dan waktu proses yang digunakan pada Persamaan 72) menggunakan waktu proses pada mesin di Tahap 1 yang mengalami proses deteriorasi dan waktu proses pada mesin di Tahap 2 yang mengalami efek proses belajar-lupa. Persamaan (72) berubah menjadi persamaan sebagai berikut :

$$Q_{g[i]} = \frac{G_{g[i]}}{i} + \frac{s}{2T_{0[i]}} (i + 1) - i \frac{s}{T_{0[i]}}, \quad i = N, \dots, 1 \quad (75)$$

$$\text{dengan } G_{g[i]} = n_g - \sum_{k=i}^{N_g} Q_{g[k+1]}$$

Berikut merupakan sub algoritma yang dikembangkan untuk menentukan jumlah serta ukuran *batch* yang dicari dalam menyelesaikan permasalahan.

[Sub Algoritma Penentuan Jumlah dan Ukuran *Batch*]

Langkah 0 : Set nilai untuk parameter $t_{0[N]}$, s , n_g . Lanjutkan ke Langkah 1.

Langkah 1 : Hitung nilai N_g dengan Persamaan (70) dan Persamaan (71). Lanjutkan ke Langkah 2.

Langkah 2 : Hitung nilai $Q_{g[i]}$ ($g = 1, 2, \dots, G$; $i = N, \dots, 1$) dengan Persamaan (75) untuk setiap kondisi yang dihasilkan dari Langkah 1. Lanjutkan ke Langkah 3.

Langkah 3 : Lakukan pembulatan. BERHENTI.

E. Pengembangan Sub Algoritma Penentuan Urutan *Batch*

Pendekatan CDS (Campbell, Dudek, dan Smith) adalah salah satu solusi yang dapat digunakan untuk permasalahan penjadwalan n *job* pada sistem produksi *flowshop* dengan m mesin. CDS pada penentuan urutan *batch* dalam penelitian dilakukan dengan menghitung Persamaan (76).

$$R_{mj} = \left(\frac{t_{mg} Q_{jg} + s_{mg}}{Q_{jg}} \right), \text{ dengan } m (m = 1, 2, \dots, M) \quad (76)$$

Modifikasi dilakukan hanya pada indeks yang digunakan. Waktu *setup* yang semula tidak konstan menjadi konstan sehingga terdapat penghilangan indeks yang semula s_{mg} menjadi s . Perhitungan rasio dapat dilihat pada Persamaan (77).

$$R_{mj} = \left(\frac{t_{gm} Q_{gj} + s}{Q_{gj}} \right) \quad (77)$$

Pada tahap inisialisasi dilakukan *input* data parameter, meliputi *common due date* (d), waktu proses setiap jenis *item* di masing-masing mesin (t_{gm}), waktu *setup* (s), jumlah *item* (G), jumlah mesin (M), permintaan setiap *item* (n_g), ukuran *batch* hasil Sub Algoritma Penentuan Ukuran

Batch (Q_{gj}), parameter AFT pada mesin umum (γ_0), ekspektasi sisa umur mesin inisial (α_{00}), parameter deteriorasi mesin (β_0), gradien proses belajar (m), minimal jumlah produksi untuk mencapai tahap keadaan tunak (q), parameter proses belajar (δ), dan minimal waktu jeda operator sebelum mengalami efek lupa secara total (t_b). Setelah dilakukan *input* parameter selanjutnya ditentukan himpunan dalam algoritman solusi ini, antara lain :

1. Himpunan rasio *batch* di masing – masing mesin (R)
2. Himpunan rasio untuk pengurutan *batch* (R')
3. Himpunan alternatif urutan (HS)

Berikut merupakan sub algoritma menentukan urutan *batch* dalam menyelesaikan permasalahan.

[Sub Algoritma Penentuan Urutan *Batch*]

- Langkah 0 : Inisialisasi berupa set nilai untuk parameter $d, t_{gm}, s, G, M, n_g, Q_{gj}, \gamma_0, \alpha_{00}, \beta_0, b, q, \delta, t_b$. serta tetapkan $R = \{ \}$, $R' = \{ \}$, dan $HS = \{ \}$. Lanjutkan ke Langkah 1.
- Langkah 1 : Hitung rasio untuk setiap *batch item* di masing-masing mesin m ($m = 1, 2, \dots, M$) dengan menggunakan Persamaan (77), masukan nilai rasio ke dalam himpunan $R = \{ \}$. Lanjutkan ke Langkah 2.
- Langkah 2 : Set $k = 1$. Hitung R'_1 dan R'_2 dengan menggunakan Persamaan (78) serta masukkan nilai ke dalam himpunan $R' = \{ \}$. Lanjutkan ke Langkah 3.
- $$\begin{aligned} R'_1 &= R_{1j} \\ R'_2 &= R_{Mj} \end{aligned} \tag{78}$$
- Langkah 3 : Mulai dari rasio yang paling terkecil. Jika *batch* memiliki R'_1 minimum maka urutkan *batch* dari yang paling jauh dengan *due date*. Jika R'_2 minimum maka urutkan *batch* dari yang paling dekat dengan *due date*. Sisihkan *batch* yang telah diurutkan. Ulangi hingga semua *batch* selesai diurutkan. Cek jika $k = M - 1$ lanjutkan ke Langkah 5. Jika tidak, lanjutkan ke Langkah 4.
- Langkah 4 : Set $k = k + 1$. Hitung R'_1 dan R'_2 dengan menggunakan Persamaan (IV.87), serta masukkan nilai ke dalam himpunan $R' = \{ \}$, ulangin Langkah 3. Jika $k = M - 1$, lanjutkan ke Langkah 5. Jika tidak, ulangin Langkah 4.
- $$\begin{aligned} R'_1 &= \sum_{p=1}^k R_{[p]j} \\ R'_2 &= \sum_{p=1}^k R_{[M-k+1]j} \end{aligned} \tag{79}$$
- Langkah 5 : Susun urutan yang diperoleh untuk setiap nilai k . Masukkan urutan pada himpunan HS . BERHENTI.

F. Algoritma Solusi

Pengembangan algoritma solusi dimulai dengan memeriksa kelayakan masalah, apabila masalah penelitian dapat dikatakan layak maka permasalahan penjadwalan dapat diselesaikan. Setelah permasalahan dikatakan layak, tahap selanjutnya adalah menentukan ukuran *batch* yang akan dijadikan *input* sebagai penentuan urutan *batch*. Setelah itu, menjadwalkan *batch* yang diperoleh dari penentuan ukuran dan urutan *batch*.

Berikut merupakan langkah-langkah algoritma yang diusulkan untuk menyelesaikan permasalahan penjadwalan *batch* pada *flow shop* dua tahap.

[Algoritma Solusi]

- Langkah 0 : Set nilai untuk parameter $d, t_0, t_g, s, G, n_g, \gamma_0, \alpha_{00}, \beta_0, m, q, \delta, t_b$. Lanjutkan ke Langkah 1.
- Langkah 1 : Tetapkan N_{max} dengan N_{max} = jumlah seluruh permintaan. Lanjutkan ke Langkah 2.

- Langkah 2 : Tetapkan banyaknya jenis produk yang akan diproduksi dengan notasi g , untuk $g = 1, 2, \dots, G$. Pada langkah ini banyaknya *batch* sama dengan banyaknya jenis produk (G). Lanjutkan ke Langkah 3.
- Langkah 3 : Kelompokkan *item* yang sejenis ke dalam satu *batch*. Pada langkah ini banyaknya *batch* sama dengan banyaknya jenis produk (G). Set $N = G$. Lanjutkan ke Langkah 4.
- Langkah 4 : Tentukan ukuran *batch item g*. Pada langkah ini ukuran *batch item g* sama dengan banyaknya permintaan *item g*. Lanjutkan ke Langkah 5.
- Langkah 5 : Urutkan *batch* dengan menggunakan **Sub Algoritma Penentuan Urutan *Batch***. Indeks ulang *batch* yang diperoleh dengan indeks i ($i = 1, 2, \dots, N$), dimulai dari urutan yang paling dekat dengan *due date*. Lanjutkan ke Langkah 6.
- Langkah 6 : Hitung total waktu tinggal aktual dengan menyelesaikan Persamaan 43 sampai dengan Persamaan 68 untuk setiap alternatif urutan yang ada. Lanjutkan ke Langkah 7.
- Langkah 7 : Pilih urutan yang memberikan nilai total waktu tinggal aktual terkecil. Apakah yang dihasilkan melanggar saat 0 ?
- Jika tidak, maka solusi dikatakan layak dan lanjutkan ke Langkah 8.
 - Jika ya, maka solusi dikatakan tidak layak. Maka order yang diterima tidak dapat diproses (order ditolak). SELESAI.
- Langkah 8 : Pecahkan *batch* secara bertahap. Set $N = G + 1$. Lanjutkan ke Langkah 9.
- Langkah 9 : Tentukan ukuran *batch* menggunakan **Sub Algoritma Penentuan Ukuran *Batch*** dan urutkan berdasarkan **Sub Algoritma Penentuan Urutan *Batch***. Lalu hitung total waktu tinggal actual dengan menyelesaikan Persamaan 43 sampai dengan Persamaan 68 untuk setiap alternative urutan yang ada. Lanjutkan ke Langkah 10.
- Langkah 10 : Pilih yang menghasilkan nilai total waktu tinggal aktual terkecil. Apakah penjadwalan yang dihasilkan melanggar saat 0 ?
- Jika tidak, maka solusi dikatakan layak dan lanjutkan ke Langkah 11.
 - Jika ya, maka solusi dikatakan tidak layak, dan lanjutkan ke Langkah 14.
- Langkah 11 : Apakah $F_N^a \leq F_{N-1}^a$?
- Jika ya, lanjutkan ke langkah 12.
 - Jika tidak, stop algoritma, Nterpilih = $N - 1$.
- Langkah 12 : Apakah $N = n_{total}$?
- Jika ya, *stop* algoritma, N terpilih = N
 - Jika tidak, lanjutkan ke Langkah 13.
- Langkah 13 : Set $N = N + 1$ dan lakukan kembali Langkah 9.
- Langkah 14 : Set $N = N + 1$. Hitung total waktu tinggal aktual (*total actual flowtime*) dengan menyelesaikan Persamaan 43 sampai Persamaan 68. Lanjutkan ke Langkah 15.
- Langkah 15 : Apakah penjadwalan yang dihasilkan melanggar saat 0 ?
- Jika tidak, maka solusi dikatakan layak dan lanjutkan ke Langkah 11.
 - Jika ya, maka solusi dikatakan tidak layak, stop algoritma, dan Nterpilih adalah N terakhir yang layak.

G. Hasil Pengujian Skenario

Hasil pengujian untuk skenario 1 dengan kondisi jumlah permintaan untuk *item 1* dan *item 2* adalah 9 unit dan 12 unit dengan *due date* 50, 35, 20, 11, dan 10 menghasilkan jumlah *batch*, ukuran *batch*, dan urutan *batch* yang sama akan tetapi berbeda untuk nilai *due date* 10. Namun total waktu tinggal actual yang dihasilkan berbeda.

Pada skenario 2 dilakukan beberapa set data dengan jumlah permintaan jenis *item 1* lebih besar dari permintaan jenis *item 2*. Hasil dari skenario 2 adalah permintaan jumlah *batch*, ukuran *batch*, dan komposisi jumlah *batch* untuk masing-masing *item* yang dihasilkan tidak selalu sama serta tidak mengikuti pola tertentu.

Pada skenario 3 model diuji dengan menggunakan beberapa set data parameter deteriorasi yang bervariasi, dengan nilai β_0 , γ_0 , dan α_{00} yang berbeda sementara data yang lainnya sama dengan data skenario 1 dengan due date 100. Nilai β_0 , γ_0 , dan α_{00} diberikan dengan aturan pertama : β_0 , γ_0 konstan dan , dan α_{00} meningkat. Aturan kedua : α_{00} dan γ_0 konstan dan β_0 meningkat. Aturan ketiga yaitu β_0 dan α_{00} konstan dan γ_0 meningkat. Hasil pengujian Skenario 3 menunjukkan bahwa perubahan parameter deteriorasi tidak berpengaruh terhadap jumlah batch dan ukuran batch yang dihasilkan. Hal ini disebabkan karena dalam penentuan jumlah dan ukuran batch waktu proses yang digunakan adalah waktu proses proses inisial (*standard*) dari setiap item. Sehingga dapat disimpulkan bahwa nilai total waktu tinggal aktual akan semakin besar apabila kondisi parameter α_{00} semakin kecil, gradient Weibull β_0 semakin kecil, dan akselerasi kerusakan γ_0 semakin besar.

Skenario 4 model diuji dengan menggunakan beberapa set data parameter proses belajar-lupa opertaor yang bervariasi, yaitu dengan memberikan nilai q , t_b , dan δ yang berbeda sementara data yang lainnya sama dengan data skenario 1 dengan due date 100. Nilai q , t_b , dan δ diberikan dengan aturan pertama q dan t_b konstan dan δ berkurang. Aturan kedua adalah q dan δ konstan, dan t_b meningkat. Aturan ketiga adalah t_b dan δ konstan, dan q meningkat. Berdasarkan hasil dari pengujian skenario 4 untuk kondisi q dan t_b konstan dan δ berkurang menghasilkan jumlah *batch* dan ukuran *batch* yang berbeda. Akan tetapi peningkatan nilai δ terlihat berpengaruhnya terhadap nilai *total actual flow time*, dimana semakin besar nilai δ maka akan semakin besar nilai *total actual flow time*. Skenario 4 untuk kondisi q dan δ konstan, dan t_b meningkat menghasilkan jumlah *batch* dan ukuran *batch* yang berbeda. Akan tetapi peningkatan nilai t_b terlihat berpengaruhnya terhadap nilai *total actual flow time*, dimana semakin besar nilai t_b maka akan semakin kecil nilai *total actual flow time*. Skenario 4 untuk kondisi nilai t_b dan δ konstan, dan q meningkat, menghasilkan jumlah *batch*, ukuran *batch*, dan nilai *total actual flow time* yang sama.

H. Contoh Numerik dan Analisis Model

Hasil perhitungan dari contoh numerik berupa jumlah *batch*, ukuran *batch*, dan urutan pemrosesan. Kemudian digambarkan dalam bentuk *gantt chart*. Berikut merupakan hasil perhitungan contoh numerik. Langkah 1 Algoritma solusi memberikan $T_{min} = 1.9258$ dimana $T_{min} \leq d$, sehingga contoh numerik ini layak untuk model. Langkah 2 sampai dengan Langkah 16 menghasilkan solusi yang ditampilkan dalam Tabel 1.

Tabel 1 Rekap Hasil Contoh Numerikal

i	Jenis Produk	Q_i	i	Jenis Produk	Q_i	i	Jenis Produk	Q_i	i	Jenis Produk	Q_i	i	Jenis Produk	Q_i	i	Jenis Produk	Q_i
1	1	3	1	1	3	1	1	2	1	1	1	1	1	1	1	1	1
2	2	4	2	2	2	2	1	1	2	1	1	2	1	1	2	1	1
			3	2	2	3	2	2	3	1	1	3	1	1	3	1	1
						4	2	2	4	2	2	4	2	2	4	2	1
									5	2	2	5	2	1	5	2	1
												6	2	1	6	2	1
															7	2	1
Fa =		18,88	Fa =		17,984	Fa =		15,338	Fa =		13,473	Fa =		13,323	Fa =		13,475

Tabel 2 Saat Mulai *Batch* Contoh Numerikal

Batch Urutan ke	Jenis Produk	Tahap 1		Tahap 2	
		Saat Mulai	Saat Selesai	Saat Mulai	Saat Selesai
1	1	5999,4298	5999,7341	5999,7341	6000
2	1	5998,9768	5999,2798	5999,3168	5999,5892
3	1	5998,5248	5998,8268	5998,8268	5999,1668
4	2	5997,7739	5998,3748	5998,3748	5999,1545
5	2	5997,3238	5997,6239	5997,8238	5998,0784
6	2	5996,8738	5997,1738	5997,1738	5997,6738

Berdasarkan Tabel IV.2 didapat bahwa solusi contoh numerik model ini diberikan oleh iterasi ke enam dengan nilai $F^a = 13.323$, jumlah *batch* (N) adalah 6 serta ukuran *batch* masing-masing adalah $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 1$, $Q_4 = 2$, $Q_5 = 1$, dan $Q_6 = 1$. Berdasarkan Tabel IV.3 dapat terlihat

bahwa tidak terjadi proses yang tumpang tindih dan saat mulai proses *batch* tidak terdapat yang melanggar $t = 0$. *Gantt chart* penjadwalan yang terbentuk dapat dilihat pada Gambar 2.



Gambar 2 *Gantt Chart* Contoh Numerikal

SIMPULAN

Pengembangan model yang dihasilkan berhasil dikembangkan dengan memperhatikan *flow shop* dua tahap untuk meminimumkan total waktu tinggal aktual. Solusi yang dihasilkan adalah solusi layak.

Nilai total waktu tinggal aktual semakin besar apabila efek deteriorasi semakin membesar pula. Efek deteriorasi membesar ketika semakin kecil nilai parameter α_0 , semakin kecil nilai gradient Weibull β_0 , dan semakin besar nilai akselerasi kerusakan γ_0 . Selain itu, nilai total waktu tinggal aktual akan konstan apabila parameter nilai q semakin meningkat. Akan tetapi, nilai total waktu tinggal aktual akan semakin besar apabila beberapa efek proses belajar-lupa operator semakin membesar pula. Efek proses belajar-lupa operator membesar ketika nilai δ semakin besar dan nilai t_b semakin kecil.

Perlu dilakukan pengembangan yang dapat dilakukan dari penelitian ini dengan mempertimbangkan waktu tenggat dan waktu *setup* yang bervariasi karena pada sistem nyata terdapat sistem manufaktur yang memproduksi produk atau *item* dengan waktu tenggat yang bervariasi dan waktu *setup* yang berbeda untuk setiap mesin.

DAFTAR PUSTAKA

- Behnamian, J., Fatemi Ghomi, S. M. T., Jolai, F., & Amirtaheri, O. (2012). Minimizing makespan on a three-machine flowshop batch scheduling problem with transportation using genetic algorithm. *Applied Soft Computing Journal*, 12(2), 768–777. <https://doi.org/10.1016/j.asoc.2011.10.015>
- Carlson, J. G. & R. A. J. (1976). How much does forgetting cost? *Industrial Engineering*, 8, 40–47.
- Cheng, M., Tadikamalla, P. R., Shang, J., & Zhang, B. (2015). Two-machine flow shop scheduling with deteriorating jobs: Minimizing the weighted sum of makespan and total completion time. *Journal of the Operational Research Society*, 66(5), 709–719. <https://doi.org/10.1057/jors.2014.58>
- Damodaran, P., Rao, A. G., & Mestry, S. (2013). Particle swarm optimization for scheduling batch processing machines in a permutation flowshop. *International Journal of Advanced Manufacturing Technology*, 64(5–8), 989–1000. <https://doi.org/10.1007/s00170-012-4037-z>
- Halim, A. H. (1993). *Batch Scheduling for production system under Just in Time Environment*. University of Osaka Prefecture.
- Halim, A. H., Miyazaki, S., & Ohta, H. (1994). Theory and Methodology Batch-scheduling problems to minimize actual flow times of parts through the shop under JIT environment. In *European Journal of Operational Research* (Vol. 72).
- Janiak, A., Lichtenstein, M., & Rusoń, A. (2011). Scheduling Jobs with Linear Model of Simultaneous Ageing and Learning Effects. In *Decision Making in Manufacturing and Services* (Vol. 5, Issue •).
- Leung, J. Y.-T. (2004). *Handbook of SCHEDULING Algorithms, Models, and Performance Analysis* (CRC Press, Ed.). Chapman & Hall.
- Nakhaeinejad, M., & Nahavandi, N. (2013). An interactive algorithm for multi-objective flow shop scheduling with fuzzy processing time through resolution method and TOPSIS.

- International Journal of Advanced Manufacturing Technology*, 66(5–8), 1047–1064. <https://doi.org/10.1007/s00170-012-4388-5>
- Pranzo, M. (2003). Batch scheduling in a two-machine flow shop with limited buffer and sequence independent setup times and removal times. *European Journal of Operational Research*, 153(3 SPEC. ISS.), 581–592. [https://doi.org/10.1016/S0377-2217\(03\)00264-9](https://doi.org/10.1016/S0377-2217(03)00264-9)
- Sibarani, A. A. (2016). *Model Penjadwalan Batch pada Flowshop dengan Mesin Terdeteriorasi yang Memroses Multi Item dengan Due Date Tunggal untuk Meminimumkan Total Waktu Tinggal Aktual*. Institut Teknologi Bandung.
- Suryadhini, P. (2015). MODEL PENJADWALAN BATCH PADA FLOWSHOP DUA TAHAP DENGAN VARIASI JUMLAH PART UNTUK MEMINIMASI TOTAL ACTUAL FLOW TIME. *Jurnal Rekayasa Sistem & Industri*, 2.
- Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of Aeronautical Sciences*, 3, 122–128.
- Yusriski, R., Sukoyo, S., Ari Samadhi, T. M. A., & Halim, A. H. (2015). Integer batch scheduling problems for a single-machine with simultaneous effects of learning and forgetting to minimize total actual flow time. *International Journal of Industrial Engineering Computations*, 6(3), 365–378. <https://doi.org/10.5267/j.ijiec.2015.2.005>