

Penjadwalan *Batch* Dinamis *Flow Shop* Untuk Meminimasi Rata-Rata Keterlambatan Penyelesaian Order (*Mean Tardiness*) dan Jumlah *Scrap* Tuang di CV. Kembar Jaya

Azizah Aisyati*, Yuniaristanto dan Anik Septiani
Jurusan Teknik Industri, Univeristas Sebelas Maret Surakarta

Abstract

CV. Kembar Jaya is a metal casting manufacturing company which applies Make to Order (MTO) system. It's characteristic of scheduling is dynamic batch on flow shop production system. This scheduling method has risen the tardiness of order completion and high level of pouring scrap. To solve this problem, a rescheduling procedure which able to increase the system's performance is needed. To increase the system performance, a development of batch dynamic scheduling algorithm has been done to minimize the mean tardiness and total pouring scrap. The algorithm consists of setting molding area initialization and five sub algorithms. Sub algorithm of sorting order and determinating batch size will sort and breakdown the order into batches. Sub algorithm of molding area repairing will determine molding area allocation for batches. To prevent the drop temperature, sub algorithm of determination pouring time will identify the pouring temperature for batches. Sub algorithm of batch scheduling will schedule batch to work station. Then sub algorithm of rescheduling will accommodate the dynamic environment when order was inserted. Mean tardiness from the scheduling algorithm application is 7.39 hours or minimize mean tardiness until 65.72 %, from the current scheduling algorithm. Application of the scheduling algorithm can reduce total pouring scrap into 0 kg or 100 %.

Keyword: batch schedule, mean tardiness, pouring scrap, dynamic, flow shop.

1. Pendahuluan

CV. Kembar Jaya merupakan perusahaan yang bergerak di bidang pengecoran logam dengan menerapkan sistem manufaktur *Make-to-order (MTO)*, dimana perusahaan akan memproduksi produk setelah adanya pesanan atau order dari konsumen. Order yang diterima oleh CV. Kembar Jaya adalah benda coran yang dipakai pada onderdil kendaraan bermotor seperti, onderdil mobil, onderdil kereta api, pipa air, perkakas barang-barang elektronik dan sebagainya. Ada dua macam hasil logam coran di CV. Kembar Jaya yaitu jenis FC (*Fero Casting*) dan FCD (*Fero Casting Ductile*). Logam jenis FC mengandung lebih banyak unsur karbon dibanding dengan FCD, sehingga memiliki sifat mekanik yang berbeda. Jenis cetakan yang dipergunakan untuk coran FC dan FCD adalah cetakan pasir dan cetakan CO_2 yang memiliki pola sekali pakai.

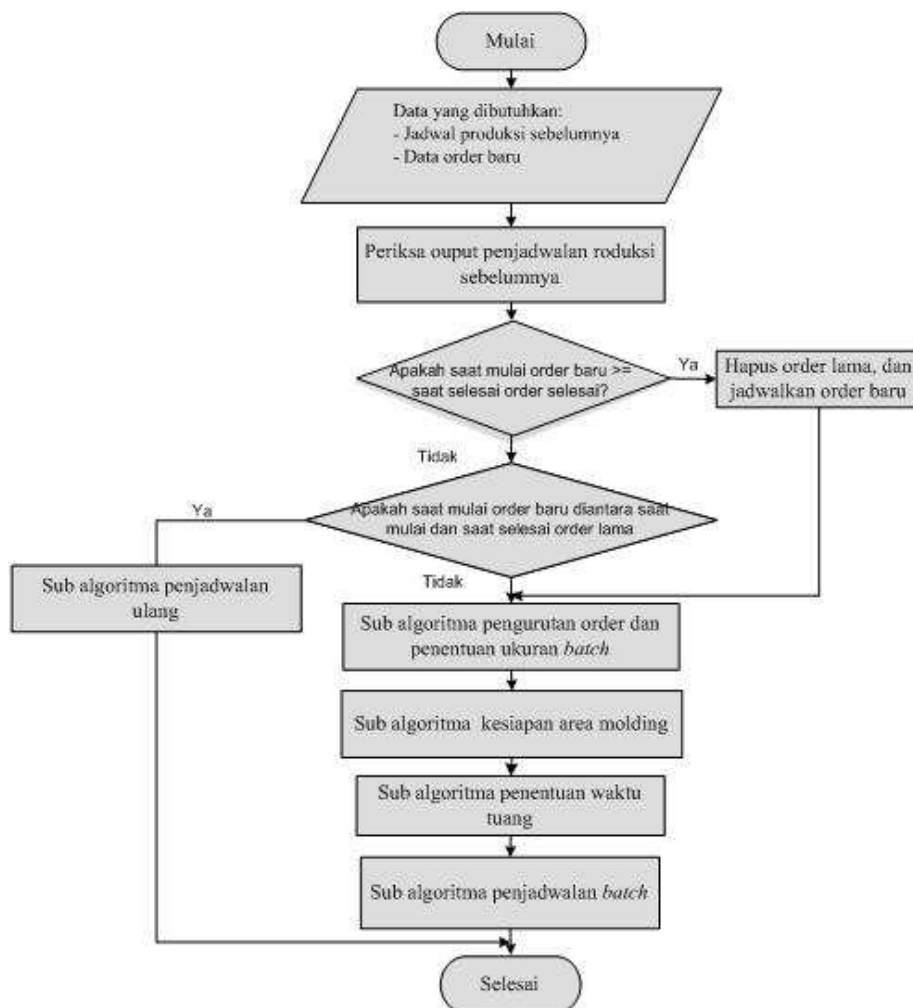
Karakteristik sistem produksi yang diterapkan CV. Kembar Jaya adalah *batch flow shop*, dimana *order* di bagi menjadi beberapa *batch* dan dikerjakan dengan urutan proses kontinyu (*flow shop*). Proses produksi benda coran dilakukan dalam tiga stasiun kerja secara berurutan, yaitu stasiun kerja *Molding*, *Melting*, dan *Finishing*. Proses pembuatan cetakan, proses penuangan logam cair ke dalam cetakan, proses pendinginan dan proses pembongkaran

* *Correspondece:* aisyati@yahoo.com

benda coran dilakukan pada stasiun kerja *Molding*. Stasiun kerja *Melting* terdiri dari 2 mesin tanur listrik yang dioperasikan secara bergantian, berfungsi untuk melakukan proses peleburan bahan baku menjadi logam cair. Sedangkan stasiun kerja *finishing* berfungsi untuk membersihkan saluran turun, saluran masuk, saluran penambah dan juga pasir yang menempel pada benda coran.

Permasalahan yang dihadapi oleh CV. Kembar Jaya dilantai produksi saat ini adalah masih tingginya tingkat keterlambatan penyelesaian *order*. Keterlambatan penyelesaian *order* tersebut disebabkan adanya suhu logam cair tidak layak tuang (*drop*) ketika jarak antar cetakan yang akan diisi berjauhan. Hal ini disebabkan karena belum adanya pengaturan cetakan pada area *molding*, sehingga cetakan di stasiun kerja *molding* dilakukan dimana saja pada area *molding*. Selain itu, pembebanan tanur tidak mempertimbangkan jumlah ukuran *batch* pada tiap-tiap *job* yang akan dikerjakan, hal ini mengakibatkan sisa cairan logam (*scrap*) yang cukup tinggi. Tingginya jumlah *scrap* mengakibatkan jumlah *output* produksi berkurang, sehingga target produksi tidak tercapai. Pada penelitian ini akan dikembangkan penjadwalan *batch flowshop* untuk meminimasi rata-rata keterlambatan penyelesaian *order* (*mean tardiness*) dan jumlah *scrap* tuang di CV. Kembar Jaya.

2. METODE PENELITIAN



Gambar 1. Flow chart metode penelitian

2.1. Inisialisasi Penyetingan Area *Molding*

Tahap penyetingan area *molding* adalah untuk menentukan alokasi cetakan pada area *molding*. Kebutuhan area *molding* dihitung selanjutnya dilakukan penyetingan area untuk memperpendek jarak tuang antar cetakan.

2.2. Algoritma Penjadwalan Produksi

Kriteria penjadwalan yang digunakan adalah minimisasi mean tardiness yang dirumuskan sebagai berikut:

$$\text{Min } \bar{T} = \frac{1}{N} \sum_{i=1}^N T_i \quad (1)$$

dimana:

\bar{T} : rata-rata keterlambatan

T_i : jumlah keterlambatan order ke- i

Langkah-langkah algoritma penjadwalan adalah sebagai berikut:

- Langkah 0 : Ambil data jadwal produksi sebelumnya dan data order baru: jenis material item produk j pada order i (g_{ij}), jumlah *demand* (D), *due date* order (d_i), saat kedatangan order ke- i (A_i).
- Langkah 1 : Periksa *output* sebelumnya:
- Jika saat selesai order lama \leq saat mulai order baru, maka hapus order lama, jadwalkan order baru dan lanjutkan ke langkah 3.
 - Jika saat kedatangan order terakhir diantara saat mulai dan saat selesai order awal lanjutkan ke langkah 2.
- Langkah 2 : Jadwalkan order dengan sub algoritma penjadwalan ulang dan selesai.
- Langkah 3 : Urutkan order dan pecah order menjadi *batch* dengan sub algoritma pengurutan order dan penentuan ukuran *batch*.
- Langkah 4 : Jadwalkan *batch* pada area *molding* untuk proses alokasi cetakan dengan sub algoritma kesiapan area *molding*.
- Langkah 5 : Jadwalkan *batch* untuk mengidentifikasi waktu tuang agar tidak drop dengan sub algoritma penentuan waktu tuang.
- Langkah 6 : Distribusikan *batch* pada masing-masing stasiun kerja dengan sub algoritma penjadwalan *batch* dan proses selesai.

Saat mulai order untuk dikerjakan di setiap mesin adalah 2 hari setelah order diterima (sudah dikonfirmasi oleh konsumen untuk dikerjakan, *confirmed order*). Jika saat mulai order tersebut lebih besar dari saat selesai order sebelumnya, maka order akan dijadwalkan setelah 1 hari kedatangannya. Sedangkan untuk *confirmed order* yang memiliki saat mulai lebih kecil dari saat mulai order sebelumnya, maka order tersebut dapat dijadwalkan setelah 1 hari dengan sub algoritma penjadwalan ulang. Dimana periode pengerjaan order pada sub algoritma penjadwalan ulang adalah 2 hari setelah kedatangan order tersebut.

a. Sub Algoritma Pengurutan Order dan Penentuan Ukuran *Batch*

Langkah-langkah sub algoritma pengurutan order dan penentuan jumlah ukuran *batch* adalah :

Langkah 1 : Ambil data order yang meliputi:

No order (i), tanggal kedatangan (A), *due date* (d), item yang di pesan (j), jenis produk material (g), jumlah unit produk yang di pesan (D), dan volume produk per item (kg) (V_j).

Langkah 2 : Merekap data order dengan mengelompokkan item produk berdasarkan jenis produknya .

Langkah 3 : Apakah order lebih dari satu, jika ya maka lanjutkan ke langkah 4 dan jika tidak maka lanjutkan ke langkah 5.

Langkah 4 : Urutkan order berdasarkan prioritas *Earlist due date* (EDD). Jika ada order dengan *due date* yang sama, maka pilih order dengan prioritas *Short processing time* (SPT) dan jika masih ada order dengan nilai proses yang sama, maka pilih order dari nomor order teratas.

Langkah 5 : Set $i = 1$, dimana i adalah order terpilih ($i = 1, 2, \dots, N$).

Langkah 6 : Pecah order i menjadi unit *batch* dengan persamaan (2) dan (3).

Jumlah *batch* order i (n_i) adalah:

$$n_i = \text{Round up } \frac{D_i}{Q_{bi}} \quad (2)$$

$$\text{dimana: } \sum_{b=1}^{n_i} Q_{bi} = D_i$$

Ukuran *batch* (dalam kg) adalah:

$$Q_{bi} = M_{pji} = \left(\text{Round down } \frac{Mc}{V_{ji}} \right) \times V_{ji} \quad (3)$$

Ukuran *batch* gabungan ($Q_{(gab)bi}$) adalah:

$$Q_{(gab)bi} = Q_{ni} + \left(\text{Round down } \left(\frac{Mc - Q_{ni}}{V_{j(i+1)}} \right) \right) \times V_{j(i+1)} \quad (4)$$

Jumlah unit item produk pada *batch* b order i adalah:

$$q_{bi} = \sum_{j=1}^{J_i} \frac{Q_{bi}}{V_{ji}} \quad (5)$$

dimana:

Q_{bi} = ukuran *batch* b pada order i (kg)

Q_{ni} = ukuran *batch* terakhir di order i (kg)

M_{pji} = kapasitas maksimal peleburan item produk ke- j , order i (kg)

n_i = jumlah *batch* yang diperlukan untuk memenuhi jumlah permintaan order i

Mc = kapasitas pengecoran (480 kg)

V_{ji} = volume unit untuk item produk j pada order ke- i (kg)

$V_{j(i+1)}$ = volume unit untuk item produk j pada order ke- $i+1$ (kg)

q_{bi} = jumlah item produk pada *batch* b di order i .

$Q_{(gab)bi}$ = ukuran *batch* gabungan untuk *batch* b order ke- i

Penentuan ukuran *batch* disesuaikan dengan kapasitas maksimal item produk untuk setiap peleburan. Total volume *batch* harus sama dengan jumlah *demand* order i .

Langkah 7 : Jika ada $Q_{(gab)bi}$ atau ukuran *batch* dengan order gabungan, maka lanjutkan ke langkah 9. Jika tidak lanjutkan ke langkah 8.

Langkah 8 : Urutkan *batch* dalam pada order i berdasarkan ukuran *batch* yang terbesar.

Langkah 9 : Set *batch* pertama $b = 1$, dimana b adalah nomor *batch* terpilih $b = (1, 2, \dots, n)$.

Langkah 10: Simpan data ukuran *batch* pada order terpilih.

Langkah 11: Apakah $b = n$ adalah urutan *batch* terakhir, jika ya lanjutkan ke langkah 12, jika tidak set $b = b + 1$ dan kembali ke langkah 9.

Langkah 12: Jika ukuran *batch* terakhir di order i (Q_{ni}) $< M_{pji}$, maka lanjutkan ke langkah 14, dan jika tidak maka lanjutkan ke langkah 13.

Langkah 13: Jika $i = N$, yaitu order terakhir pada urutan order maka lanjutkan ke langkah 16, jika tidak set $i = i + 1$ dan lanjutkan ke langkah 14.

Langkah 14: Periksa apakah jenis produk di *ukuran batch terakhir di order ke-i* g_{ni} = jenis produk di *order i+1* (g_{i+1}), jika tidak kembali ke langkah 9. Jika ya lanjutkan ke langkah 15.

Langkah 15: Hitung *demand* aktual pada *order i+1* (D^a_{i+1}) dan kembali ke langkah 5

$$D^a_{i+1} = D_{i+1} + Q_{ni} \quad (6)$$

Langkah 16: Data hasil pengurutan order dan penentuan *ukuran batch* dan proses selesai.

b. Sub Algoritma Kesiapan Area Molding

Sub algoritma ini bertujuan untuk menentukan kesiapan *area molding* yang akan digunakan sebagai dasar penetapan alokasi untuk distribusi cetakan. Kebutuhan *area molding* disesuaikan dengan urutan *batch* yang akan diproses pada proses peleburan. Hal ini disebabkan cetakan harus siap ketika proses peleburan selesai, sehingga langsung dapat didistribusikan ke masing-masing cetakan.

Langkah 1 : Ambil data *batch* dari hasil pengurutan order dan penentuan *ukuran batch* di sub algoritma sebelumnya, status masing-masing area, saat selesai lebur *batch* sebelumnya di proses lebur untuk *batch* sebelumnya ($End_{m(b-1)i}$) dan saat selesai proses pembongkaran di *batch b* order ke-*i* (End_{rbi}).

Langkah 2 : Pilih nomor *batch* teratas dari hasil algoritma pengurutan order dan penentuan *ukuran batch* sebelumnya.

Langkah 3 : Set $b=1$, untuk nomor *batch* pertama pada himpunan *batch* yang akan dijadwalkan.

Langkah 4 : Set q_{bi} = jumlah unit produk pada *batch b*, di order ke-*i*.

Langkah 5 : Set $End_{m(b-1)i} = Start_{Abi}$, saat selesai lebur untuk *batch* sebelumnya ($b-1$) order ke-*i* di stasiun kerja melting adalah saat mulai area ke-*A* digunakan untuk *batch* ke-*b* order ke-*i*.

Langkah 6 : Set jenis area yang dibutuhkan sesuai unit item produk pada *batch b* order ke-*i*.

$$\text{Set } A_{FC_a} = 1$$

$$\text{Set } A_{FC_a} = 1$$

$$\text{Set } A_{FC_a} = 1$$

Dalam satu *batch* dimungkinkan dapat membutuhkan lebih dari 1 jenis area, karena dalam satu *batch* tidak menutup kemungkinan akan terdiri dari beberapa jenis produk.

Langkah 7 : Periksa status area ke-1 untuk setiap jenis area.

$$\text{Apakah status } A_{FC_1} = \phi$$

$$\text{Apakah status } A_{Rem_1} = \phi$$

$$\text{Apakah status } A_{FCD_1} = \phi$$

Jika tidak set area untuk masing-masing jenis area dengan $a+1$ dan kembali ke langkah 6. Hal ini disebabkan karena prioritas penggunaan area di untuk setiap jenisnya dimulai dari area dengan jenis produk 1 kemudian 2 dan seterusnya.

Jika status area di masing-masing jenis produk ϕ , maka lanjutkan ke langkah berikutnya.

Langkah 8 : Tampilkan status “isi “ pada setiap area berdasarkan jenis areanya.

Langkah 9 : Hitung saat area kosong dengan persamaan berikut:

$$R_{a_{bi} \phi} = End_{rbi} + s \quad (7)$$

dimana: $R_{a_{bi} \phi}$ = saat area molding *batch b*, order ke-*i* kosong.

$End_{r_{bi}}$ = saat selesai proses bongkar untuk *batch b* pada order *i*

s = *set up* area molding.

Langkah 10: Simpan status masing-masing area.

Langkah 11: Apakah $b = n$, jika ya maka lanjutkan ke langkah berikutnya dan jika tidak set $b = b + 1$ dan kembali ke langkah 3.

Langkah 12: Kelompokkan *batch-batch* tersebut berdasarkan jenis area *molding* kemudian simpan *batch b* yang telah terjadwalkan pada area molding *a* sebagai jadwal inisial area *a* dan selesai.

c. Sub Algoritma Penentuan Waktu tuang

Tahap penentuan waktu penuangan ini digunakan untuk mengidentifikasi total waktu penuangan di setiap *batch*. Besarnya waktu penuangan tergantung pada jumlah unit dalam *batch* yang akan diproses. Berdasarkan hasil pemecahan *batch* sebelumnya, dapat diketahui adanya kemungkinan perbedaan ukuran *batch* satu dengan yang lainnya, sehingga total waktu penuangan juga akan berbeda. Hal ini disebabkan oleh penurunan suhu pada logam paduan sebanding dengan penambahan waktu. Besarnya waktu penuangan tergantung pada proses penguangannya. Jika waktu total penuangan tiap *batch* melebihi suhu dropnya, maka proses penuangan akan dilakukan lebih dari satu kali penuangan.

Langkah-langkah untuk sub algoritma penentuan total waktu penuangan adalah :

Langkah 0 : Tahapan penentuan total waktu tuang dimulai dari pengumpulan data, antara lain: data hasil pengurutan order dan penentuan ukuran *batch* sebelumnya, status area molding yang digunakan pada *batch b*, waktu proses tuang ke ladle, waktu angkut ladle, waktu tuang ke cetakan, waktu drop tiap jenis produk dan $End_{m_{bi}}$ (saat selesai proses peleburan untuk *batch b* order *i*).

Langkah 1 : Set $i = 1$, dimana i adalah nomor *order* dari data *urutan order* yang telah diurutkan sebelumnya $i = (1, 2, \dots, N)$.

Langkah 2 : Set $b = 1$, dimana B adalah nomor *batch* dari data *batch* yang telah diurutkan sebelumnya $b = (1, 2, \dots, n)$.

Langkah 3 : Set $w = 1$, untuk proses penuangan 1 kali, dimana $w = (1, 2 \dots, W)$.

Langkah 4 : Hitung total waktu proses tuang untuk *batch b* order ke- i ($P_{t_{bi}}$)

$$P_{t_{bi}} = \sum_{w=1}^W P_{t_{il}} + P_{al} + P_{tc} \quad (8)$$

dimana: $P_{t_{il}}$ = waktu tuang ke ladle.

P_{al} = waktu angkut ladle ke area cetakan

P_{tc} = waktu proses tuang ke dalam cetakan.

Langkah 5 : Periksa apakah waktu tuang *batch* ke- b order i , lebih kecil dibandingkan waktu drop (waktu dimana suhu logam tidak layak tuang) $P_{t_{bi}} < P_{drop}$, jika tidak lanjutkan ke langkah 7 dan jika ya lanjutkan ke langkah 6.

Langkah 6 : Set $w = w + 1$, untuk proses penuangan dilakukan lebih dari 1 kali.

Langkah 7 : Hitung jumlah volume tiap proses penuangan ke- w , dan kembali ke langkah 4.

$$V_{t_w} = \frac{Q_{bi}}{w} \quad (9)$$

dimana: w = jumlah proses penuangan.

V_{t_w} = volume penuangan ke- w (kg).

Langkah 8 : Hitung saat selesai proses tuang untuk *batch b* order ke-*i* (End_{tbi})

$$End_{tbi} = End_{mbi} + P_{tbi}$$

Saat selesai tuang untuk *batch b*, di order *i* adalah saat selesai proses peleburan *batch b*, order *i* ditambah waktu tuang *batch*.

Langkah 9 : Simpan data waktu tuang tiap *batch* dan saat selesai tuang *batch*.

Langkah 10: Apakah $b = n$, untuk urutan *batch* terakhir, jika tidak maka set $b = b + 1$ dan kembali ke langkah 2 dan jika ya lanjutkan ke langkah 11.

Langkah 11: Jika $i =$ order terakhir, maka lanjutkan ke langkah 12. Jika tidak set $i = i + 1$, yaitu periode selanjutnya dan kembali ke langkah 1.

Langkah 12: Kelompokkan data waktu tuang dan saat selesai tuang di masing-masing *batch* dan proses selesai.

d. Sub Algoritma Penjadwalan *Batch*

Penjadwalan *batch* dilakukan mulai dari proses peleburan di stasiun kerja *melting*, proses pendinginan dan pembongkaran di stasiun kerja *molding* dan proses pengerjaan akhir di stasiun kerja *finishing*. Proses peleburan, pendinginan dan pembongkaran merupakan proses kontinyu, dimana logam yang dilebur dalam satu *batch* peleburan kemudian dituang ke cetakan dan mengalami proses pendinginan dan pembongkaran. Penentuan urutan *batch* di peroleh dari data hasil algoritma pengurutan order dan pemecahan *batch*. Pengurutan *batch* pada proses pengerjaan akhir menggunakan aturan minimasi *slack time*. Minimasi *slack time* digunakan untuk memilih *batch* yang memiliki deviasi terkecil antara waktu proses dengan *due date*.

Langkah-langkah penjadwalan *batch* adalah sebagai berikut:

Langkah 0 : Ambil data hasil pengurutan order dan pemecahan *batch* sebelumnya, waktu proses peleburan (P_m), waktu proses pendinginan (P_c), pembongkaran (P_r), waktu proses finishing (P_f), waktu tuang (P_t) (jam kerja produktif (W_{on}) dan non produktif di shift 1 ($W_{off_{s_1}}$) dan jam kerja non produktif di shift 2 ($W_{off_{s_2}}$).

Langkah 1 : Pilih *batch* teratas di order pertama dari data hasil pengurutan order dan penentuan ukuran *batch* sebelumnya.

Langkah 2 : Set $t = 0$, sebagai periode penjadwalan, dimana $t = 0, 1, 2, \dots, T$.

Langkah 3 : Set $b = 1$, sebagai *batch* pertama, dimana b adalah nomor *batch* terpilih $b = (1, 2, \dots, n)$.

Langkah 4 : Set $End_{m(b-1)i} = Start_{mbi}$, dimana saat selesai *batch* sebelumnya atau *batch* ke-($b-1$) pada proses *melting* merupakan saat mulai *batch b*, order ke-*i* pada proses *melting* berikutnya.

Langkah 5 : Jika saat mulai operasi peleburan lebih besar dari jam kerja non produktif di shift 1 dan shift 2 (W_{off}), $Start_{mbi} \geq W_{off_{s_1}}$, atau $Start_{mbi} \geq W_{off_{s_2}}$ maka saat mulai operasi menjadi :

$$Start_{mbi}^a = Start_{mbi} + 1 \text{ jam} \quad (10)$$

dimana:

$$W_{off_{s_1}} = \text{waktu non produktif shift 1: } 11.30 - 12.30 = 1 \text{ jam}$$

$$W_{off_{s_2}} = \text{waktu non produktif shift 2: } 23.30 - 00.30 = 1 \text{ jam}$$

$Start_{mbi}$ = saat mulai aktual, yaitu saat mulai operasi peleburan hasil perhitungan awal yang dikurangi total jam kerja non produktif ($\sum W_{off}$). Waktu non produktif dipertimbangkan karena stasiun peleburan dikerjakan secara manual.

Langkah 6 : Hitung saat selesai *batch* *b* pada proses peleburan

$$End_{mbi} = Start_{mbi} + P_m \quad (11)$$

Saat selesai *batch* *b*, order ke-*i* pada proses peleburan adalah saat mulai aktual *batch* *b*, order ke-*i* pada proses peleburan ditambah waktu proses peleburan.

Langkah 7: Hitung waktu transfer (waktu tuang ke cetakan) dengan menggunakan sub algoritma penentuan waktu tuang, $Start_{cbi} = End_{tbi}$.

Saat mulai *batch* *b* di order *i* pada proses pendinginan ($Start_{cbi}$) adalah saat selesai *batch* *b* di order *i* pada proses penuangan (End_{tbi}).

Langkah 8 : Hitung saat selesai *batch* pada proses pembongkaran di stasiun kerja *molding*.

$$End_{rbi} = Start_{cbi} + P_c + P_r \quad (12)$$

Saat selesai *batch* *b* order *i* pada proses pembongkaran (End_{rbi}) adalah saat mulai *batch* *b* order *i* pada proses pendinginan ditambah waktu proses pendinginan dan waktu proses pembongkaran.

Langkah 9 : Jika selesai *batch* *b* order *i* pada proses pembongkaran (End_{rbi}) lebih besar dari jam kerja non produktif di shift 1 dan shift 2 (W_{off}), $End_{rbi} \geq W_{off_{s_1}}$, atau

$End_{rbi} \geq W_{off_{s_2}}$ maka saat mulai operasi menjadi :

$$End_{rbi}^a = End_{rbi} + 1 \text{ jam} \quad (13)$$

dimana: $W_{off_{s_1}}$ = waktu non produktif shift 1: 11.30 – 12.30 = 1 jam

$W_{off_{s_2}}$ = waktu non produktif shift 2: 23.30 – 00.30 = 1 jam

Langkah 10: Simpan saat mulai dan saat selesai *batch* di masing-masing proses.

Langkah 11: Jika $b = n$, lanjutkan ke langkah 13, jika tidak set $b = b + 1$ dan kembali ke langkah 4.

Langkah 12: Data hasil penjadwalan *batch* di stasiun kerja melting dan molding.

Langkah 13: Set R_{sf} sebagai saat siap *server* di proses finishing.

Langkah 14: Pilih saat selesai *batch* diproses bongkar tercepat (End_{rbi}), jika saat selesai *batch* diproses bongkar tercepat (End_{rbi}) lebih dari satu maka pilih *batch* berdasarkan nilai *slack time* terkecil

$$ST_{bi} = d_i - R_{sf} - p_f \quad (14)$$

Langkah 15: Set $b_f = 1$

Langkah 16: Jika saat mulai proses finishing lebih besar dari jam kerja non produktifnya $Start_{fbi} \geq W_{off_{s_1}}$ maka saat mulai operasi menjadi :

$$Start_{fbi}^a = Start_{fbi} + 1 \text{ jam} \quad (15)$$

W_{off_f} adalah waktu non produktif proses *finishing*: 11.30 – 12.30 = 1 jam.

Langkah 17: Hitung saat selesai pada proses *finishing*

$$Start_{bif}^a = R_{sf}$$

$$Start_{fbi} = Start_{bif}^a + P_f \quad (16)$$

Langkah 18: Simpan data saat mulai dan saat selesai *batch* di proses *finishing*.

Langkah 19: Jika $b_f = n$, maka lanjutkan ke langkah berikutnya, jika tidak set $b_f = b_f + 1$ dan kembali ke langkah 16.

Langkah 20: Hasil penjadwalan *finishing* dan proses selesai.

e. Sub Algoritma Penjadwalan Ulang

Sub algoritma penjadwalan ulang merupakan algoritma untuk menjadwalkan order baru yang datang pada saat suatu order sedang dikerjakan. Kedatangan order baru menyebabkan adanya perubahan pada jumlah operasi yang akan dikerjakan. Kedatangan order baru akan dipertimbangkan untuk dijadwalkan pada awal periode untuk tiap shift pertama. Langkah-langkah yang dilakukan dengan adanya order baru yang datang pada saat $t = A$ pada sub algoritma penjadwalan ulang adalah sebagai berikut:

Langkah 0 : Periksa *output* penjadwalan sebelumnya?, apakah masih ada *batch* yang belum selesai diproses? Jika ya lanjutkan ke langkah berikutnya, dan jika tidak maka selesai.

Langkah 1 : Periksa mulai dari baris paling atas nomor *batch* b , ukuran *batch* yang belum selesai diproses tersebut. dan nyatakan $b = 1$, untuk $b = 1, 2, \dots, n$.

Langkah 2 : Periksa apakah saat selesai proses peleburan untuk *batch* b di order i ($End_{mbi} \geq A$) (saat kedatangan order terakhir). Jika tidak lanjutkan ke langkah 3, jika ya lanjutkan ke langkah 4.

Langkah 3 : Set $b = b+1$, jika $b \leq n$ ulangi langkah 1, jika tidak lanjutkan ke langkah 5.

Langkah 4 : Periksa apakah saat mulai proses peleburan untuk *batch* b order i di proses melting $Start_{mbi} \geq A$ (saat kedatangan order terakhir), jika tidak set $Tarr = End_{mbi}$ dan kembali kelangkah 3, dan jika ya lanjutkan ke langkah 5.

Langkah 5 : *Update* status nomer *batch* menjadi $Tarr$.

Langkah 6 : Hitung jumlah *batch* yang selesai diproses hingga $Tarr$, nyatakan sebagai F_i .

Langkah 7 : Update nilai D_i yaitu jumlah permintaan order i yang dijadwalkan menjadi.

$$D^a_i = D_i - F_i \quad (17)$$

D^a_i adalah jumlah permintaan aktual order i yang belum dijadwalkan.

Langkah 8 : Jadwalkan sisa unit di D_i dan order baru dengan sub algoritma pengurutan order dan penentuan ukuran *batch*.

Langkah 9 : Jika semua *due date* dapat dipenuhi, maka order baru dapat diterima, dan lanjutkan ke langkah 10, dan jika tidak maka tampilkan peringatan "diundur *due datenya*" dan kembali ke jadwal sebelum terjadinya kedatangan order.

Langkah 10: Data revisi jadwal produksi yang baru.

3. Hasil Penelitian

Hasil dari aplikasi algoritma penjadwalan *batch* dinamis *flowshop* pada data kasus di CV. Kembar Jaya adalah sebagai berikut:

1. Perhitungan rata-rata keterlambatan penyelesaian pesanan (*mean tardiness*) pada 2 hasil penjadwalan yaitu:
 - a. Hasil penjadwalan produksi dinamis dengan menerapkan keputusan penolakan order dengan menggunakan sub algoritma penjadwalan ulang.

Tabel 1. Rata-rata keterlambatan penyelesaian pesanan
(*Mean Tardiness*) dengan penolakan order

Order (<i>i</i>)	r_i		C_i		d_i		Keterlambatan order (T_i)	
	jam	tgl	jam	tgl	jam	tgl	Penelitian	Perusahaan
							jam	Jam
1	13:00	2-Sep	13:06	8-Sep	14:00	8-Sep	0,0	23
2	7:00	2-Sep	15:35	2-Sep	14:00	6-Sep	0,0	0,0
3	7:00	4-Sep	16:16	4-Sep	14:00	7-Sep	0,0	0,0
4	3:00	8-Sep	12:08	30-Sep	14:00	30-Sep	0,0	114
5	17:00	21-Sep	9:09	22-Sep	14:00	25-Sep	0,0	0,0
6	7:40	21-Sep	8:36	22-Sep	14:00	25-Sep	0,0	24
7	Diundur due datenya	-	-	-	-	-	-	26,5
8	Diundur due datenya	-	-	-	-	-	-	0,0
9	Diundur due datenya	-	-	-	-	-	-	0,0
10	Diundur due datenya	-	-	-	-	-	-	0,0
11	Diundur due datenya	-	-	-	-	-	-	49,66
<i>Mean tardiness</i>							0,0	21,56

- b. Hasil penjadwalan produksi dinamis dengan menerapkan keputusan penerimaan seluruh order meskipun akan menimbulkan *tardiness* (keterlambatan pada order sebelumnya). Hal ini dilakukan untuk mengetahui kehandalan dari algoritma yang dibuat.

Tabel 2. Rata-rata Keterlambatan Penyelesaian Pesanan
(*Mean Tardiness*) dengan menerima semua order

Order (<i>i</i>)	r_i		C_i		d_i		Keterlambatan order (T_i)	
	jam	tgl	jam	tgl	jam	tgl	Penelitian	Perusahaan
							jam	Jam
1	13:00	2-Sep	13:06	8-Sep	14:00	8-Sep	0,0	23
2	7:00	2-Sep	15:35	2-Sep	14:00	6-Sep	0,0	0,0
3	7:00	4-Sep	16:16	4-Sep	14:00	7-Sep	0,0	0,0
4	3:00	8-Sep	15:14	4-Okt	14:00	30-Sep	64,3	114
5	17:00	21-Sep	8:34	23-Sep	14:00	25-Sep	0,0	0,0
6	7:40	21-Sep	7:59	22-Sep	14:00	25-Sep	0,0	24
7	7:40	22-Sep	10:12	25-Sep	14:00	25-Sep	0,0	26,5
8	5:20	24-Sep	12:49	25-Sep	14:00	26-Sep	0,0	0,0
9	7:00	25-Sep	12:49	25-Sep	14:00	26-Sep	0,0	0,0
10	0:20	24-Sep	11:29	25-Sep	14:00	26-Sep	0,0	0,0
11	3:20	3 Okt	9:14	5-Sep	14:00	2-Okt	17,3	49,66
<i>Mean tardiness</i>							7,39	21,56

2. Perhitungan jumlah *scrap* tuang

Berdasarkan aplikasi algoritma penjadwalan diperoleh total jumlah ukuran *batch* yang dilebur untuk memenuhi order bulan September 2006 adalah 164110 kg. Output produksi dengan berdasarkan aplikasi algoritma penjadwalan *batch* adalah 164110 kg, sehingga diketahui tidak ada *scrap* tuang.

Tabel 3. Hasil perhitungan jumlah *scrap* tuang

Hasil penjadwalan	Jumlah <i>demand</i> (kg)	Jumlah yang dilebur (kg)	Output produksi (kg)	Jumlah <i>scrap</i> tuang (kg)
Perusahaan	164110	167674	164110	5126
Penelitian	164110	164110	164110	0

Hasil peleburan yang dilakukan oleh perusahaan untuk memenuhi order bulan September 2006 diketahui sebesar 167674 kg dengan jumlah *scrap* tuang sebesar 5126 kg. Dengan menggunakan algoritma penjadwalan hasil penelitian jumlah *scrap* tuang dapat dikurangi menjadi 0 kg atau dalam satuan prosentase dapat mengurangi jumlah *scrap* tuang sebesar 100%.

3.1. Analisis Rata-rata Keterlambatan Penyelesaian Order

Berdasarkan data perusahaan rata-rata keterlambatan penyelesaian order pada bulan pengerjaan September dengan menjadwalkan 11 order diperoleh rata-rata keterlambatan penyelesaian order adalah 21,56 jam. Sedangkan dengan mengaplikasikan algoritma penjadwalan yang telah dikembangkan rata-rata keterlambatan bisa berkurang menjadi 7,39 jam. Rata-rata keterlambatan penyelesaian order (*mean tardiness*) bisa diperbaiki hingga mencapai 65,72%. lebih efektif jika dibandingkan dengan sistem penjadwalan dari perusahaan.

3.2. Analisis Jumlah *Scrap* Tuang

Berdasarkan hasil peleburan yang dilakukan oleh perusahaan untuk memenuhi order tersebut sebesar 167.674 kg dengan jumlah *scrap* tuang sebesar 5.126 kg. Sedangkan berdasarkan aplikasi algoritma penjadwalan yang dikembangkan diperoleh total hasil peleburan untuk memenuhi permintaan order bulan September adalah 164.110 kg tanpa ada *scrap* tuang.

4. Kesimpulan

Berdasarkan penelitian tentang penjadwalan produksi *batch* dinamis *flow shop* dengan mempertimbangkan rata-rata keterlambatan penyelesaian (*mean tardiness*) dan meminimasi jumlah *scrap* tuang di CV. Kembar Jaya dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan aplikasi algoritma penjadwalan dengan set data order bulan September 2006 diperoleh hasil rata-rata keterlambatan penyelesaian order (*mean tardiness*) sebesar 7,39 jam, sedangkan berdasarkan metode penjadwalan perusahaan diperoleh *mean tardiness* sebesar 21,56 jam. Sehingga dengan mengaplikasikan algoritma penjadwalan yang telah dikembangkan dapat meminimasi *mean tardiness* sebesar 14,17 jam atau sebesar 65,72%.
2. Total peleburan untuk memenuhi permintaan order bulan September adalah 164110 kg. Berdasarkan aplikasi algoritma penjadwalan yang telah dikembangkan diperoleh total peleburan untuk memenuhi order bulan September adalah 164110 kg, sehingga tidak ada *scrap* tuang. Sedangkan berdasarkan sistem penjadwalan perusahaan total peleburan untuk memenuhi order bulan September adalah 167674 kg dengan jumlah *scrap* tuang sebesar 5126kg.. Oleh karena itu dengan menerapkan algoritma penjadwalan yang telah dikembangkan dapat meminimasi jumlah *scrap* tuang yaitu menjadi 0 kg atau 100%.

Daftar Pustaka

- Aji, Anselmus. *Penjadwalan Produksi Minibus Jenis Varian dengan Algoritma Drum-Buffer-Rope untuk Meminimasi mean Tardiness dan Work in Process*. Skripsi Teknik Industri, UNS, Surakarta, 2002.
- Baker, K.R dan College D. *Introduction to Sequencing and Scheduling*. John Wiley & Sons, 1974.
- Bedworth, D.D, Bailey. *Integrated Production Control System Management*. Singapore : John Wiley & Sons, 1982.
- Bakrun. dkk. *Pembuatan Jadwal Material Release: Suatu Pendekatan dengan Theory Of Constraints (TOC)*: Preceedings Seminar Sistem Produksi VI, 1999.
- Fogarty, D.W, Blackstone, John H. dan Hoffman, Thomas R. *Production and Inventory Management*. Cincinnati, Ohio: South-Western Publising, 1999.
- Pariyanti, Yuni. *Penjadwalan Produksi Pada Proses Persiapan Pertenunan dengan mempertimbangkan Due (Studi Kasus PT. Kusuma Hadi Santosa)*. Tugas Akhir Sarjana Program Studi Teknik Industri, Jurusan Teknik Industri, Sebelas Maret Surakarta, 2004.
- Tejaasih *et al.* *Pengembangan Model Penjadwalan Batch pada Sistem Produksi Flow Shop Heterogenous Machine Dinamis dengan Ukuran Batch Integer untuk Meminimasi Total Actual Flow Time. Pendekatan Mundur pada Sistem produksi Flow Shop yang Dinamis (Studi Kasus PT. Supramas Inti Kemilau)*. Preceedings Seminar Sistem Produksi V, 2001.
- Uzsoy. *et al.* *Alternative Loading and Dispatching Policies for Furnace Operation in Semiconductor Manufacturing*: In Proceeding of the Winter Simulation Conference, 2000.
- Wignjosoebroto, Sritomo. *Ergonomi, Studi Gerak dan Waktu: Teknik Analisis untuk Peningkatan Produktivitas Kerja*. Surabaya: Guna Widya, 2000.