

**PENJADWALAN PRODUKSI
PADA LINGKUNGAN *FLEXIBLE JOB SHOP PROBLEM* (FJSP)
UNTUK MEMINIMASI TOTAL *TARDINESS*
(Studi Kasus di Divisi PPIP PT. Barata Indonesia (Persero), Gresik)**

***PRODUCTION SCHEDULING IN FLEXIBLE JOB SHOP PROBLEM (FJSP)
ENVIRONMENT TO MINIMIZE THE TOTAL TARDINESS
(Case Study: PPIP Division of PT. Barata Indonesia (Persero), Gresik)***

Nimas Elmidina Kartika Sari¹⁾, Ishardita Pambudi Tama²⁾, Ceria Farela Mada Tantrika³⁾

Jurusan Teknik Industri Universitas Brawijaya

Jalan MT. Haryono 167, Malang, 65145, Indonesia

E-mail : nimas.elmidina@gmail.com¹⁾, kangdith@gmail.com²⁾, ceria246@gmail.com³⁾

Abstrak

PT. Barata Indonesia (Persero) Gresik merupakan Badan Usaha Milik Negara (BUMN) yang bergerak dibidang Engineering, Procurement, dan Construction. Divisi Produksi Peralatan Industri Proses (PPIP) sering mengalami keterlambatan dalam memenuhi order atau pesanan dari konsumen dikarenakan oleh metode penjadwalan yang kurang disesuaikan dengan waktu datang kontrak dan waktu pemenuhan due date dari tiap order-nya. Berdasarkan hal tersebut, perusahaan menginginkan solusi usulan agar kondisi buruk ini dapat diperbaiki sehingga keterlambatan pengerjaan order dapat diatasi. Penelitian ini mengusulkan metode Earliest Due Date (EDD), First Come First Serve (FCFS), dan integrasi antara Algoritma Tabu Search dengan Pairwise Interchange Heuristic sebagai keputusan untuk memprioritaskan urutan job sebelum dijadwalkan, kemudian dalam tahapan pengerjaannya menggunakan algoritma Sequencing yang dirancang untuk mendapatkan hasil jadwal usulan yang mendekati optimal, yaitu dengan nilai tardiness dan penalti yang paling minimal. Hasil penelitian menunjukkan bahwa prioritas urutan job yang optimal adalah 1-2-3; 2-3-1; 3-2-1; dan 1-3-2. Seluruh job yang dijadwalkan, yaitu job 1, job 2 dan job 3, secara berurutan membutuhkan waktu produksi selama 87,3 jam; 156 jam; dan 218,6 jam. Nilai ini setara dengan 11 hari, 20 hari dan 28 hari dan seluruh job dapat selesai 9 hari, 12 hari, dan 14 hari sebelum due date-nya. Selanjutnya, penalti diperhitungkan berdasarkan nilai tardiness. Dengan menggunakan metode usulan menghasilkan jadwal yang tidak menghasilkan tardiness, maka penalti juga tidak terjadi sehingga perusahaan dapat menghemat biaya produksi.

Kata Kunci: *Penjadwalan Produksi, First Come First Serve (FCFS), Earliest Due Date (EDD), Algoritma Tabu Search, Tardiness, Penalti.*

1. Pendahuluan

Proses produksi merupakan salah satu hal krusial yang harus diperhatikan perusahaan agar dapat bersaing dengan perusahaan lain karena di dalamnya terdapat perencanaan produksi, penentuan kualitas yang disesuaikan dengan keinginan konsumen, penjadwalan kegiatan produksi, serta pengolahan limbah yang dihasilkan selama proses produksi berlangsung. Penjadwalan produksi merupakan hal yang penting karena kegiatan ini merupakan awal dimulainya proses produksi dan mempengaruhi ketepatan waktu penyelesaian *order*. Penjadwalan yang baik akan menghasilkan waktu penyelesaian yang optimal dan dapat memberikan kepuasan yang bagus terhadap konsumennya.

Penjadwalan dapat didefinisikan sebagai “terdapat *job* yang akan diproses dan tiap *job* mempunyai waktu *set-up*, waktu proses dan waktu *due date*, agar diselesaikan dan setiap *job* harus diproses pada beberapa mesin. Maka dibutuhkan suatu urutan pekerjaan-pekerjaan ini pada mesin-mesin yang ada agar diperoleh performansi optimal menurut kriteria tertentu”. Berdasarkan definisi ini, maka pemahaman penjadwalan sangatlah penting karena sebuah jadwal akan memberikan informasi kapan sebuah *job* akan dimulai dan harus diselesaikan (Andriani, Astuti, & Setyanto, 2010).

PT. Barata Indonesia (Persero) merupakan perusahaan BUMN manufaktur yang bergerak di bidang *Engineering, Procurement, dan Construction* yang memiliki empat *workshop*

Tabel 1. Data Order Produk Laddle dan Kualii Timah Selama Tahun 2012-2014

Thn.	No. Order	Jenis Produk	Order Qty. (unit)	Perencanaan		Lama Waktu Proses	Realisasi	Deviasi (keterlambatan)
				Mulai	Selesai			
2012	PLG 9 12 007	Laddle Timah	10	26-Dec-11	28-Feb-12	23 hari	16-Feb-12	-12 hari
	PLG 9 12 096	Kualii Timah	15	21-Mar-12	30-Apr-12	27 hari	29-Mei-12	29 hari
	PLG 9 12 105	Kualii Timah	3	04-Apr-12	04-Mei-12	22 hari	28-Jun-12	55 hari
	PLG 9 12 127	Kualii Timah	3	24-Mei-12	25-Jun-12	23 hari	30-Jun-12	5 hari
	PLG 9 12 179	Kualii Timah	5	10-Agu-12	05-Sep-12	18 hari	18-Okt-12	43 hari
	PLG 9 12 181	Kualii Timah	5	30-Agu-12	30-Sep-12	23 hari	25-Dec-12	86 hari
	PLG 9 12 284	Laddle Timah	15	11-Dec-12	27-Dec-12	12 hari	23-Apr-13	113 hari
2013	PLG 9 13 022	Kualii Timah	4	14-Jan-13	13-Feb-13	22 hari	11-Mar-13	26 hari
	PLG 9 13 62	Kualii Timah	6	04-Mar-13	10-Mar-13	7 hari	10-Mei-13	61 hari
	PLG 9 13 122	Kualii Timah	6	28-Jun-13	15-Agu-13	30 hari	30-Jul-13	-16 hari
	PLG 9 13 132	Laddle Timah	10	18-Jul-13	21-Agu-13	20 hari	26-Agu-13	5 hari
2014	PLG 9 13 326	Laddle Timah	10	22-Dec-13	22-Jan-14	40 hari	07-Feb-14	12 hari
	PLG 9 14 007	Kualii Timah	10	06-Jan-14	20-Feb-14	32 hari	05-Mar-14	13 hari
	PLG 9 14 011	Laddle Timah	25	16-Jan-14	17-Mar-14	42 hari	15-Apr-14	20 hari

dengan produk yang berbeda-beda. Secara khusus, penelitian ini dilakukan pada *workshop* III Produksi Peralatan Industri Proses (PPIP) yang memproduksi alat-alat berat seperti *pressure vessel*, komponen kapal, struktur kerangka, *head*, kualii timah, dsb.

Berdasarkan data terbaru awal tahun ini, bagian *Plan and Production Controll* (PPC) mempunyai 14 *order* yang harus diselesaikan oleh *workshop* PPIP. Lebih lanjut, selama tiga tahun terakhir terdapat tiga *order* yang selalu dipesan, produk yang dimaksud adalah kualii timah dan *laddle*. Pada Tabel 1 ditampilkan data mengenai *order* selama 3 tahun terakhir untuk produk tersebut. Dapat dilihat bahwa produk paling banyak dipesan sebanyak 7 kali pada tahun 2012 dengan kisaran kuantitas pesanan antara 3-25 unit.

Data deviasi keterlambatan masing-masing *order* pada Tabel 1 selanjutnya ditampilkan pada Gambar 1. Pada gambar tersebut dapat dilihat secara rinci mengenai variasi lama hari keterlambatan pemenuhan *order*, yakni mulai 5-113 hari. Keterlambatan ini berdampak buruk

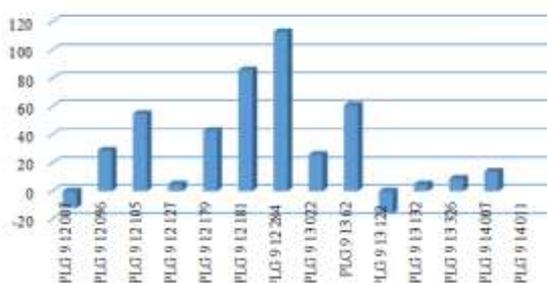
bagi perusahaan khususnya bagi *workshop* PPIP karena setiap *order* yang terlambat akan dikenakan biaya penalti yang harus dibayarkan kepada konsumen sebagai kompensasinya.

Terdapat dua kondisi untuk menghitung besar biaya penalti tersebut. Kondisi pertama, yaitu ketika keterlambatan kurang dari 30 hari maka besarnya penalti adalah 5%/₀₀ dari nilai kontrak dihitung setiap harinya. Sedangkan kondisi kedua, yaitu ketika keterlambatan lebih dari 30 hari maka biaya penalti sebesar 5% dari nilai kontraknya.

Alur produksi di *workshop* III termasuk ke dalam *Flexible Job Shop* (FJS) karena pada lantai produksinya terdapat dua *workcentre* (*workshop*), yang salah satunya terdapat mesin-mesin yang dipasang secara paralel. Menurut Mahmudy, Marian, dan Luong (2013), permasalahan pada lingkungan FJS terbilang sulit karena didalamnya terdapat kegiatan mendelegasikan mesin yang akan digunakan dan menentukan alur produksinya (*sequencing*). Oleh karena itu, penelitian ini menggunakan beberapa metode untuk mencari solusi metode penjadwalan yang sesuai bagi *workshop* III.

Metode *First Come First Serve* (FCFS) dan *Earliest Due Date* (EDD) digunakan sebagai keputusan dalam memprioritaskan *order* untuk dijadwalkan (sebagai jadwal inisial). Pinedo (2008) menjelaskan bahwa EDD mengurutkan job berdasarkan waktu due date-nya, selanjutnya menurut Ginting (2009), metode FCFS merupakan sebuah keputusan dalam memberikan prioritas tertinggi pada operasi yang masuk S_i (Stasiun ke-t) lebih dahulu. Kedua metode ini digunakan sebagai langkah awal untuk menemukan jadwal inisiasi.

Grafik Keterlambatan Order (Tardiness)



Gambar 1. Grafik Keterlambatan Waktu Pemenuhan Order (Tardiness) Tahun 2012-2014

Caranya adalah dengan membandingkan hasil dari masing-masing jadwal tersebut kemudian memilih yang terbaik berdasarkan nilai *tardiness* dan penaltinya yang paling minimal.

Jadwal inisiasi yang terpilih diolah lebih lanjut dengan menggunakan Algoritma *Tabu Search* yang diintegrasikan dengan *Pairwise Interchange Heuristic* untuk menentukan solusi *neighbourhood* selanjutnya. Menurut Glover dan Laguna dalam Berlianty dan Arifin (2010), algoritma *Tabu Search* adalah suatu algoritma yang menuntun setiap tahapannya agar dapat menghasilkan kriteria aspirasi yang paling optimum tanpa terjebak ke dalam solusi awal yang telah ditemukan sebelumnya selama tahapan perhitungan berlangsung. Hasil akhir dari penjadwalan dengan menggunakan Algoritma *Tabu Search* inilah yang menjadi saran rekomendasi bagi perusahaan untuk mengatasi masalah *tardiness* dan penalti yang selama ini sering terjadi.

2. Metode Penelitian

Penelitian ini dibagi menjadi empat tahap pengerjaan, yaitu tahap pendahuluan, tahap pengumpulan dan pengolahan data, tahap hasil dan pembahasan, dan terakhir tahap penarikan kesimpulan.

2.1 Tahap Pendahuluan

Pada tahap pendahuluan meliputi:

- a. Studi lapangan
- b. Studi literatur
- c. Identifikasi masalah
- d. Perumusan masalah
- e. Penentuan tujuan penelitian

2.2 Tahap Pengumpulan dan Pengolahan Data

Berikut penjelasan tahap pengumpulan dan pengolahan data secara terpisah.

2.2.1 Tahap Pengumpulan Data

Metode pengumpulan data sebagai berikut:

- a. Studi Literatur (*Library Research*)
Merupakan metode mempelajari literatur di berbagai sumber tertulis yang relevan dengan permasalahan dalam penelitian ini.
- b. Penelitian Lapangan (*Field Research*)
Merupakan metode pengumpulan data secara langsung pada obyek penelitian secara nyata di lapangan. Kegiatannya meliputi observasi, wawancara serta dokumentasi.

Menurut pengertian diatas, rincian kebutuhan data primer dan data sekunder yang dalam penelitian ini adalah sebagai berikut:

- a. Data profil perusahaan
- b. Data jumlah karyawan *workshop* III
- c. Jumlah jam kerja reguler
- d. Data *order* di *workshop* III
- e. Informasi kuantitas *order* dan *due date*-nya
- f. *Bill Of Material* (BOM) *Tree*
- g. Data alur proses produksi
- h. Data mesin yang digunakan
- i. Data waktu proses tiap operasi semua *job* yang akan dijadwalkan
- j. Waktu standar operasi *handling* dan *set-up*
- k. Jadwal induk produksi *existing*
- l. Informasi mengenai prosentase penalti

2.2.2 Tahap Pengolahan Data

Data yang telah terkumpul selanjutnya akan diolah dengan tahapan sebagai berikut:

- a. Membagi kegiatan produksi menjadi dua yaitu *Forming* dan *Assembly*.
- b. Pembentukan dan pendefinisian variabel dan parameter dari penjadwalan.
- c. Penentuan fungsi tujuan dan fungsi kendala dalam penjadwalan.
- d. Menentukan waktu proses tiap operasi dan *precedence*-suksesor tiap operasi.
- e. Perancangan dan pembuatan Algoritma *Sequencing* untuk menugaskan operasi pada mesin yang sesuai dan *eligible* berdasarkan prioritas urutan jadwal dengan menggunakan metode FCFS dan EDD.
- f. Pengoptimalan jadwal dengan menggunakan Algoritma *Tabu Search* dan *Pairwise Interchange Heuristic* untuk menghitung nilai *tardiness* dan penalti yang paling minimal.
- g. Verifikasi dan Validasi.
- h. Menghitung waktu keterlambatan positif.
- i. Menghitung kemungkinan biaya penalti.

2.3 Tahap Hasil dan Pembahasan

Tahap ini merupakan pembahasan dari hasil-hasil yang didapat setelah pengolahan data meliputi *ready time*, *completion time*, *flowtime* masing-masing *job*, *lateness*, *earliness*, *total tardiness*, dan besar biaya penaltinya. Hasil dari penjadwalan ini kemudian dibandingkan

terhadap penjadwalan yang sebelumnya direncanakan oleh perusahaan (*existing*) dan kemudian dianalisis apakah terdapat hasil yang signifikan yang dapat digunakan sebagai saran rekomendasi bagi perusahaan.

2.4 Tahap Penarikan Kesimpulan dan Saran

Setelah diperoleh pemecahan masalah, maka langkah selanjutnya adalah menarik kesimpulan. Kesimpulan yang diambil nantinya dapat menjawab rumusan masalah yang ditentukan di awal penelitian. Selain itu juga dapat memberikan saran untuk perusahaan dan penelitian selanjutnya.

3. Hasil dan Pembahasan

3.1 Data Pesanan Workshop III

Pada Tabel 2 terdapat tiga *job* yang akan dijadwalkan dalam triwulan awal yaitu *order* untuk produk *laddle* timah sebanyak 10 unit, *kuali timah* sebanyak 10 unit, dan *laddle* timah sebanyak 25 unit. Masing-masing tanggal datang *order* dan *due date*-nya juga ditampilkan pada Tabel 2.

Tabel 2. Data *Order* yang Dijadwalkan

No. <i>Order</i>	Jenis Produk	<i>Order Qty.</i> (unit)	Mulai	Deadline	Pemesan
PLG 9.13.526	<i>Laddle</i> Timah	10	22-12-13	22-01-14	PT. Non Ferindo Utama
PLG 9.14.007	<i>Kuali</i> Timah	10	06-01-14	20-02-14	PT. Non Ferindo Utama
PLG 9.14.011	<i>Laddle</i> Timah	25	16-01-14	17-03-14	PT. Non Ferindo Utama

3.2 Prioritas Pengurutan Metode EDD

Conway (1967) mengartikan penjadwalan sebagai “*The task of assigning each operation to a specific position or The time scale of the specific machine*”. Dalam menugaskan seluruh *job* agar bisa dijadwalkan dapat menggunakan beberapa metode yang sesuai. Penelitian ini menggunakan tiga metode, salah satunya adalah metode EDD. Metode ini mengurutkan *job* berdasarkan tanggal *due date*-nya, urutan ini kemudian digunakan sebagai dasar untuk menentukan *job* mana yang diprioritaskan terlebih dahulu di dalam pembuatan jadwal produksi. Untuk lebih jelasnya akan ditampilkan pada Tabel 3.

Tabel 3. Prioritas Urutan Metode EDD

	<i>Due date</i>	Notasi	Urutan
<i>Job 1 (laddle)</i>	22-Jan-14	d_1	1
<i>Job 2 (kuali timah)</i>	20-Feb-14	d_2	2
<i>Job 3 (laddle)</i>	17-Mar-14	d_3	3

3.3 Prioritas Pengurutan Metode FCFS

Metode ini mengurutkan *job* berdasarkan tanggal datang dari sekumpulan *job* yang akan dijadwalkan, urutan ini kemudian digunakan sebagai dasar untuk menentukan *job* mana yang diprioritaskan terlebih dahulu di dalam pembuatan jadwal produksi. Untuk lebih jelasnya akan ditampilkan pada Tabel 4.

Tabel 4. Prioritas Urutan Metode FCFS

	Waktu Kedatangan	Notasi	Urutan
<i>Job 1 (laddle)</i>	22-Dec-13	r_1	1
<i>Job 2 (kuali timah)</i>	06-Jan-14	r_2	2
<i>Job 3 (laddle)</i>	16-Jan-14	r_3	3

Pengurutan menggunakan dua metode tersebut mempunyai hasil yang sama yaitu urutan 1-2-3 sehingga dalam penelitian ini secara tidak sengaja menyimpulkan bahwa apabila sekumpulan *job* yang akan dijadwalkan mempunyai waktu datang dan waktu *due date* yang berbanding lurus maka akan menghasilkan urutan *job* yang sama.

3.4 Penjadwalan dengan Menggunakan Metode Sequencing

Perhitungan menggunakan algoritma *Sequencing* mengikuti tahapan-tahapan algoritma sebagai berikut:

1. Definisikan data-data di lapangan ke dalam notasi pengerjaan awal.
Data-data tersebut meliputi :
 - a. Data urutan *job*.
 - b. Data waktu kedatangan *job (arrival time)* dalam jam.
 - c. Data urutan operasi dalam satu *job (precedence dan suksesor operasi)*
 - d. Data waktu proses masing-masing operasi ($t_{i,j,k(x)}$).
 - e. *Precedence dan suksesor* masing-masing operasi
2. Set $i = 1, j = 1$ untuk memulai iterasi. Notasi tersebut dibaca sebagai *job* yang mempunyai urutan pertama (*job 1*) sehingga $i = 1$ dan operasi pertamanya ditugaskan dalam penjadwalan $j = 1$. Notasi dari mesin k bergantung pada operasi j .
3. Set $r_{i,j,k(x)}$ sebagai waktu datang *order* atau tanggal dimana *order* disetujui sesuai kontraknya. Waktu kedatangan *order (arrival time)* ini masih dalam bentuk tanggal dimana selanjutnya akan dikonversikan ke dalam jam sesuai hari aktif yang dihitung mulai dari jadwal induk ini dimulai hingga tanggal datang awal kontraknya.

4. Cek apakah $i = 1$ dan $j = 1$.
 - Jika ya, maka set nilai *ready time* tersebut sebagai waktu dimulainya operasi/job dikerjakan sehingga $S_{i,j,k(x)} = r_{i,j,k(x)}$, karena operasi pertama *job* pertama adalah tugas yang dikerjakan pertama. Lanjut ke langkah 12.
 - Jika tidak, maka lanjut ke langkah 5
5. Set $C_{k(x)}$ di mesin yang akan digunakan oleh *job*/operasi sekarang. Lanjut ke langkah 6.
6. Lakukan identifikasi terhadap operasi/job pendahulu dari operasi/job sekarang (cek *precedence*), catat dan hitung jumlahnya.
7. Cek apakah $precedence \leq 1$?
 - Jika ya maka lanjut ke langkah 10.
 - Jika tidak, maka identifikasi *precedence* operasi sekarang, set sebagai j_γ dimana notasi ini dibaca sebagai operasi pendahulu dari operasi sekarang. Nilai dari γ bergantung pada operasi yang mendahului operasi sekarang. Lanjutkan ke langkah 8.
8. Set nilai *completion time* dari masing-masing operasi *precedence* (j_γ) ke dalam notasi $C_{i,j_\gamma,k(x)}$ berdasarkan data perhitungan sebelumnya.
9. Menentukan nilai $r_{i,j,k(x)}$ dari operasi sekarang.
 $r_{i,j,k(x)} = \text{maks.}\{C_{i,j_\gamma,k(x)}\}$, nilai *ready time* operasi sekarang berdasarkan pemilihan nilai $C_{i,j_\gamma,k(x)}$ yang terbesar. Lanjut ke langkah 12.
10. Cek apakah *precedence* adalah tidak ada atau sama dengan 0?
 - Jika ya, maka nilai $r_{i,j,k(x)}$ tetap sesuai langkah 3. Lanjut ke langkah 12.
 - Jika tidak, maka set *precedence* operasi sekarang sebagai j_γ . Lanjut ke langkah 11.
11. Menentukan nilai $r_{i,j,k(x)}$ operasi sekarang dimana $r_{i,j,k(x)} = C_{i,j_\gamma,k(x)}$ (tidak perlu dibandingkan dengan beberapa *precedence*). Lanjut ke langkah 12.
12. Lakukan cek kondisi mesin dengan syarat berikut,
 - $C_{k(x)} \leq r_{i,j,k(x)}$, maka $S_{i,j,k(x)} = r_{i,j,k(x)}$, → mesin menganggur
 - $C_{k(x)} > r_{i,j,k(x)}$, maka $S_{i,j,k(x)} = C_{k(x)}$, → mesin masih digunakan
 Lanjutkan ke langkah 13.
13. Tugaskan operasi pada mesin $k(x)$ yang *eligible* $k(x)$ sesuai dengan operasinya.
 - Untuk menugaskan operasi ini gunakan waktu $t_{i,j,k(x)}$. Lanjutkan ke langkah 14.
14. Hitung nilai *completion time* untuk operasi j di mesin k yang digunakan menggunakan rumus $C_{i,j,k(x)} = S_{i,j,k(x)} + t_{i,j,k(x)}$.
15. Set nilai $C_{i,j,k(x)}$ pada langkah 14 sebagai $C_{k(x)}$ merupakan waktu acuan bagi operasi/job selanjutnya bahwa waktu selesai paling akhir di mesin k adalah pada saat t /waktu saat mesin k *eligible*.
16. Lanjutkan ke operasi selanjutnya. Set $i=i$; $j=j+1$; $k(x)$, notasi ini dibaca *job* yang sama di operasi selanjutnya yang akan dikerjakan di mesin $k(x)$. Notasi j semakin lama semakin besar hingga seluruh operasi yang terdapat pada *job* urutan pertama selesai dikerjakan atau ketika $j = p$.
 Cek apakah $j = p + 1$?
 - Jika ya, maka set $i = i+1$; $j=1$, lanjut ke langkah 17.
 - Jika tidak, ulangi langkah 5.
17. Cek apakah $i = o+1$?
 - Jika ya maka semua *job* telah terjadwalkan dan lanjut ke langkah 18.
 - Jika tidak maka kembali ke langkah 3.
18. Hitung lama waktu penyelesaian masing-masing *job* i dengan menggunakan rumus: $C_i = C_{i,j=p,k(x)}$; notasi ini dibaca adalah waktu penyelesaian operasi paling akhir di tiap *job*.
19. Hitung nilai *Lateness* : $L_i = C_i - D_i$
20. Hitung nilai *Tardiness* : $T_i = \text{max.}\{0;L_i\}$.
21. Cek apakah $T_i \leq 30$?
 - Jika ya, maka $P_i = 5\%_0 \times \text{nilai kontrak } job \ i \times T_i$
 - Jika tidak, maka $P_i = 5\% \times \text{nilai kontrak } job \ i$

3.5 Perhitungan Tardiness Menggunakan Algoritma Sequencing.

Berikut merupakan penjabaran langkah-langkah dalam menghitung nilai *tardiness* dan penalti.

1. Waktu siap masing-masing *job* (waktu kontrak *order* disepakati untuk dikerjakan):
 $Job \ 1 \rightarrow r_{1,1,1(1)} = 0$ (23 Desember 2014);
 $Job \ 2 \rightarrow r_{2,1,1(1)} = 6$ Januari 2014 = (10 hari x 8 jam/hari) + 1 jam = 81;
 $Job \ 3 \rightarrow r_{3,1,1(1)} = 16$ Januari 2014 = (10 hari x 8 jam/hari) + 1 jam = 161
 Nilai diatas berarti bahwa *job* 1 akan

- mulai dijadwalkan pada jam ke-0, *job 2* akan dijadwalkan pada jam ke-81, dan *job 3* akan dijadwalkan pada jam ke-161.
- Penjelasan predesessor dan suksesor dari masing-masing operasi dalam *job*. Untuk lebih jelasnya pada Tabel 5.
 - Alokasi waktu produksi dan mesin yang digunakan dengan menggunakan Rumus (1):

$$t_{i,j,k(x)} = \frac{n}{x} \times t_{i,j}; n = \text{kuantitas order.}$$
 Untuk lebih jelasnya pada Tabel 6.
 - Rangkuman perhitungan menggunakan Algoritma *Sequencing*. Perhitungan manual dengan menggunakan algoritma *Sequencing* terbilang sangat mudah dipahami, namun karena notasi yang digunakan cukup banyak maka dibutuhkan ketelitian yang tinggi agar setiap operasi dari *job* dapat dipahami melalui notasi-notasi yang digunakan tersebut. Terdapat notasi *i, j, k, x, j_γ, S, r, p, o* dan *C* dalam setiap iterasi

- pengerjaannya. Penjelasan dari masing-masing notasi sebagai berikut.
- i* = *job/order* yang akan dijadwalkan.
 - j* = sekumpulan dari operasi untuk menyelesaikan *job i*.
 - k* = mesin yang digunakan sesuai dengan operasi *j*.
 - x* = keterangan jumlah mesin, apabila nilai $k > 1$ maka mesin dianggap dipasang secara paralel.
 - j_{γ} = keterangan operasi predesessor dari operasi *j*.
 - S* = waktu mulai operasi *j*.
 - r* = waktu siap *job/operasi j*.
 - p* = nilai acuan untuk menandai bahwa seluruh *job* telah dijadwalkan.
 - o* = nilai acuan untuk menandai bahwa seluruh operasi penyusun satu *job* telah dijadwalkan.
 - C* = waktu penyelesaian masing-masing operasi/*job*.

Tabel 5. Data Predesessor dan Suksesor Maisng-masing Operasi

Operasi Laddle	Prece-dence	Sukse-sor	$r_{i,j,k(x)}$	Operasi Kuali	Prece-dence	Sukse-sor	$r_{i,j,k(x)}$
1	-	3	$r_{i,1,k(x)}$	1	-	3,4	$r_{i,1,k(x)}$
2	-	4,5,6	$r_{i,2,k(x)}$	2	-	5	$r_{i,2,k(x)}$
3	1	12	$r_{i,3,k(x)}$	3	1	6	$r_{i,3,k(x)}$
4	2	7	$r_{i,4,k(x)}$	4	1	7	$r_{i,4,k(x)}$
5	2	18	$r_{i,5,k(x)}$	5	2	8	$r_{i,5,k(x)}$
6	2	20	$r_{i,6,k(x)}$	6	3	9	$r_{i,6,k(x)}$
7	4	8	$r_{i,7,k(x)}$	7	4	16	$r_{i,7,k(x)}$
8	7	9	$r_{i,8,k(x)}$	8	5	24	$r_{i,8,k(x)}$
9	8	10	$r_{i,9,k(x)}$	9	6	10	$r_{i,9,k(x)}$
10	9	11	$r_{i,10,k(x)}$	10	9	11	$r_{i,10,k(x)}$
11	10	13	$r_{i,11,k(x)}$	11	10	12	$r_{i,11,k(x)}$
12	3	13	$r_{i,12,k(x)}$	12	11	13	$r_{i,12,k(x)}$
13	11,12	14	$r_{i,13,k(x)}, r_{i,13,k(x)}$	13	12	14	$r_{i,13,k(x)}$
14	13	15	$r_{i,14,k(x)}$	14	13	15	$r_{i,14,k(x)}$
15	14	16	$r_{i,15,k(x)}$	15	14	24	$r_{i,15,k(x)}$
16	15	17	$r_{i,16,k(x)}$	16	7	17	$r_{i,16,k(x)}$
17	16	18	$r_{i,17,k(x)}$	17	16	18	$r_{i,17,k(x)}$
18	5,17	19	$r_{i,18,k(x)}, r_{i,18,k(x)}$	18	17	19	$r_{i,18,k(x)}$
19	18	20	$r_{i,19,k(x)}$	19	18	20	$r_{i,19,k(x)}$
20	6,19	21	$r_{i,20,k(x)}, r_{i,20,k(x)}$	20	19	21	$r_{i,20,k(x)}$
21	20	22	$r_{i,21,k(x)}$	21	20	22	$r_{i,21,k(x)}$
22	21	23	$r_{i,22,k(x)}$	22	21	23	$r_{i,22,k(x)}$
23	22	-	$r_{i,23,k(x)}$	23	22	29	$r_{i,23,k(x)}$
				24	8,15	25	$r_{i,24,k(x)}, r_{i,24,k(x)}$
				25	24	26	$r_{i,25,k(x)}$
				26	25	27	$r_{i,26,k(x)}$
				27	26	28	$r_{i,27,k(x)}$
				28	27	29	$r_{i,28,k(x)}$
				29	23,28	30	$r_{i,29,k(x)}, r_{i,29,k(x)}$
				30	29	31	$r_{i,30,k(x)}$
				31	30	32	$r_{i,31,k(x)}$
				32	31	33	$r_{i,32,k(x)}$
				33	32	34	$r_{i,33,k(x)}$
				34	33	-	$r_{i,34,k(x)}$

Tabel 6. Data Waktu Proses Maing-masing Operasi

Operasi	Job 1	Job 2	Job 3
1	$t_{2,1,1(1)} = \frac{16 \times 10}{60} = 2,7$ jam	$t_{1,1,1(1)} = \frac{60 \times 10}{60} = 10$ jam	$t_{2,1,1(1)} = \frac{16 \times 25}{60} = 6,7$ jam
2	$t_{2,2,1(1)} = \frac{20 \times 10}{60} = 3,3$ jam	$t_{1,2,1(1)} = \frac{16 \times 10}{60} = 2,7$ jam	$t_{2,2,1(1)} = \frac{20 \times 25}{60} = 8,3$ jam
3	$t_{2,3,9(5)} = \frac{10}{5} \times \frac{12}{60} = 0,4$ jam	$t_{1,3,9(5)} = \frac{10}{5} \times \frac{10}{60} = 0,3$ jam	$t_{2,3,9(5)} = \frac{25}{5} \times \frac{12}{60} = 1$ jam
4	$t_{2,4,9(5)} = \frac{10}{5} \times \frac{10}{60} = 0,3$ jam	$t_{1,4,9(5)} = \frac{10}{5} \times \frac{12}{60} = 0,4$ jam	$t_{2,4,9(5)} = \frac{25}{5} \times \frac{10}{60} = 0,8$ jam
5	$t_{2,5,9(5)} = \frac{10}{5} \times \frac{2}{60} = 0,1$ jam	$t_{1,5,9(5)} = \frac{10}{5} \times \frac{13}{60} = 0,4$ jam	$t_{2,5,9(5)} = \frac{25}{5} \times \frac{2}{60} = 0,2$ jam
6	$t_{2,6,9(5)} = \frac{10}{5} \times \frac{2}{60} = 0,1$ jam	$t_{1,6,2(1)} = \frac{10 \times 30}{60} = 5$ jam	$t_{2,6,9(5)} = \frac{25}{5} \times \frac{2}{60} = 0,2$ jam
7	$t_{2,7,2(1)} = \frac{10 \times 35}{60} = 5,8$ jam	$t_{1,7,2(1)} = \frac{10 \times 40}{60} = 6,7$ jam	$t_{2,7,2(1)} = \frac{25 \times 40}{60} = 14,6$ jam
8	$t_{2,8,6(1)} = \frac{10 \times 120}{60} = 20$ jam	$t_{1,8,2(1)} = \frac{10 \times 35}{60} = 5,8$ jam	$t_{2,8,6(1)} = \frac{25 \times 120}{60} = 50$ jam
9	$t_{2,9,8(5)} = \frac{10}{5} \times \frac{150}{60} = 5$ jam	$t_{1,9,5(1)} = \frac{10 \times 240}{60} = 40$ jam	$t_{2,9,8(5)} = \frac{25}{5} \times \frac{150}{60} = 12,5$ jam
10	$t_{2,10,8(5)} = \frac{10}{5} \times \frac{270}{60} = 9$ jam	$t_{1,10,8(5)} = \frac{10}{5} \times \frac{90}{60} = 3$ jam	$t_{2,10,8(5)} = \frac{25}{5} \times \frac{270}{60} = 22,5$ jam
11	$t_{2,11,9(5)} = 0,6$ jam	$t_{1,11,8(5)} = 3,3$ jam	$t_{2,11,9(5)} = 1,5$ jam
12	$t_{2,12,3(1)} = 40$ jam	$t_{1,12,9(5)} = 1,3$ jam	$t_{2,12,3(1)} = 100$ jam
13	$t_{2,13,8(5)} = 3$ jam	$t_{1,13,8(5)} = 3,3$ jam	$t_{2,13,8(5)} = 7,5$ jam
14	$t_{2,14,8(5)} = 4,5$ jam	$t_{1,14,9(5)} = 1,5$ jam	$t_{2,14,8(5)} = 11,3$ jam
15	$t_{2,15,9(5)} = 1$ jam	$t_{1,15,5(1)} = 22,5$ jam	$t_{2,15,9(5)} = 2,5$ jam
16	$t_{2,16,8(5)} = 4,5$ jam	$t_{1,16,8(5)} = 1,5$ jam	$t_{2,16,8(5)} = 11,3$ jam
17	$t_{2,17,9(5)} = 0,6$ jam	$t_{1,17,8(5)} = 3,3$ jam	$t_{2,17,9(5)} = 1,5$ jam
18	$t_{2,18,8(5)} = 5$ jam	$t_{1,18,9(5)} = 1,7$ jam	$t_{2,18,8(5)} = 12,5$ jam
19	$t_{2,19,8(5)} = 5$ jam	$t_{1,19,8(5)} = 4$ jam	$t_{2,19,8(5)} = 12,5$ jam
20	$t_{2,20,8(5)} = 4$ jam	$t_{1,20,9(5)} = 1,5$	$t_{2,20,8(5)} = 10$ jam
21	$t_{2,21,8(5)} = 3,3$ jam	$t_{1,21,3(1)} = 40$ jam	$t_{2,21,8(5)} = 8,3$ jam
22	$t_{2,22,9(5)} = 1,4$ jam	$t_{1,22,4(1)} = 40$ jam	$t_{2,22,9(5)} = 3,5$ jam
23	$t_{2,23,0(0)} = 8,3$ jam	$t_{1,23,7(1)} = 10$ jam	$t_{2,23,0(0)} = 20,8$ jam
24		$t_{1,24,8(5)} = 3$ jam	
25		$t_{1,25,8(5)} = 5$ jam	
26		$t_{1,26,9(5)} = 1,3$ jam	
27		$t_{1,27,8(5)} = 5,3$ jam	
28		$t_{1,28,9(5)} = 0,7$ jam	
29		$t_{1,29,8(5)} = 5$ jam	
30		$t_{1,30,8(5)} = 7$ jam	
31		$t_{1,31,9(5)} = 1,7$ jam	
32		$t_{1,32,8(5)} = 7$ jam	
33		$t_{1,33,9(5)} = 3$ jam	
34		$t_{1,34,0(0)} = 8,3$ jam	

Setelah mengetahui penjelasan dari masing-masing notasi, maka langkah selanjutnya adalah menghitung secara manual dan membuat jadwal berdasarkan Algoritma *Sequencing*. Langkah pengerjaan dengan menggunakan metode ini terbilang cukup banyak karena dihitung satu iterasi untuk satu operasi *job i*. Maka dari itu, dalam jurnal ini hanya dicontohkan beberapa langkah awalnya saja sebagai panduan dalam merencanakan jadwal dengan menggunakan Algoritma *Sequencing*. Berikut contoh pengerjaannya pada dua iterasi awal, yaitu operasi pertama dan kedua dari *job* dengan prioritas urutan yang pertama (*job 1*).

- Urutan *job* berdasarkan aturan EDD adalah *job 1, job 2, dan job 3*. Maka *job* yang

pertama dijadwalkan dalam mesin yang *eligible* adalah *job 1* sehingga $i = 1$.

- Set $i = 1$ dan $j = 1$; $k(x) = 1(1)$ karena mesin yang digunakan adalah mesin *copier* (mesin nomor 1) dan banyaknya hanya satu (*single*).
- Set $i = \text{job 1}$ sehingga $i = 1, j = 1, k = 1$ dan $x = 1$, karena mesin 1 adalah mesin *copier* (mesin nomor 1) dan banyaknya hanya satu (*single*).
- Set nilai $r_{1,1,1(1)} =$ waktu siap *order/job 1* = 0 (23 Desember 2013).
- Apakah $i = 1$ dan $j = 1$? Ya, maka set $S_{1,1,1(1)} = r_{1,1,1(1)} = 0$.
- $C_{1,1,1(1)} = S_{1,1,1(1)} + t_{1,1,1(1)} = 0 + 2,7 \text{ jam} = 2,7 \text{ jam}$
Set $C_{1,1,1(1)} = C_{1(1)} = 2,7 \text{ jam}$
- Set $i = 1; j = 1+1 = 2; k(x) = 1(1)$

$j = 24$? Tidak, maka set nilai $C_{1(1)} = 2,7$ jam.
 Cek *precedence job* sekarang,
 Apakah *precedence* ≤ 1 ? Ya, maka lakukan pengecekan lanjut.
 Cek apakah $C_{k(x)} \leq r_{i,j,k(x)}$? ; $C_{1(1)} \leq r_{1,2,1(1)}$?
 Tidak $\rightarrow 2,7 > 0$,
 maka $S_{1,2,1(1)} = C_{1(1)} = 2,7$ jam
 $C_{1,2,1(1)} = S_{1,2,1(1)} + t_{1,2,1(1)} = 2,7$ jam + 3,3 jam = 6 jam
 Set $C_{1,2,1(1)} = C_{1(1)} = 6$ jam
 Apakah *precedence* tidak ada atau sama dengan 0? Ya, maka $r_{1,2,1(1)} = 0$

- Set $i = 1$; $j = 2+1 = 3$; $k(x) = 9(5)$
 $j = 24$? Tidak, maka set nilai $C_{9(5)} = 0$ jam.
 Cek *precedence job* sekarang,
 Apakah *precedence* ≤ 1 ? Ya, maka lakukan pengecekan lanjut.
 Apakah *precedence* tidak ada atau sama dengan 0? Tidak, maka set *precedence* operasi sekarang, $j_y = 1$.

$r_{i,j,k(x)} = C_{i,j,k(x)} \rightarrow r_{1,3,9(5)} = C_{1,1,1(1)} = 2,7$ jam
 Cek apakah $C_{k(x)} \leq r_{i,j,k(x)}$? ; $C_{9(5)} \leq r_{1,3,9(5)}$?
 Ya $\rightarrow 0 \leq 2,7$;
 maka $S_{1,3,9(5)} = r_{1,3,9(5)} = 2,7$ jam
 $C_{1,3,9(5)} = S_{1,3,9(5)} + t_{1,3,9(5)} = 2,7$ jam + 0,4 jam = 3,1 jam
 Set $C_{1,3,9(5)} = C_{9(5)} = 3,1$ jam.

Pada contoh perhitungan menggunakan algoritma *sequencing* tersebut terlihat beberapa langkah yang diulang (*looping*), yaitu ketika kondisi seluruh operasi dalam satu *job* belum dijadwalkan dalam mesin dan ketika seluruh *job* belum dijadwalkan. Ketika kondisi ini belum dipenuhi maka perhitungan algoritma akan terus dilakukan dengan mengulang langkah 5 dan 3 secara terus menerus. Namun apabila kondisi ini telah dipenuhi maka langkah pengerjaan akan lanjut ke langkah 18, yaitu menghitung lama waktu penyelesaian masing-masing *order* atau *job*.

Tabel 7. Rekap Perhitungan Waktu Penyelesaian Masing-masing Operasi

Operasi	Job 1			Job 2			Job 3		
	$S_{i,j,k(x)}$	$t_{i,j,k(x)}$ (jam)	$C_{i,j,k(x)}$	$S_{i,j,k(x)}$	$t_{i,j,k(x)}$ (jam)	$C_{i,j,k(x)}$	$S_{i,j,k(x)}$	$t_{i,j,k(x)}$ (jam)	$C_{i,j,k(x)}$
1	0,0	2,7	2,7	81,0	10,0	91,0	161,0	6,7	167,7
2	2,7	3,3	6,0	91,0	2,7	93,7	167,7	8,3	176,0
3	2,7	0,4	3,1	91,0	0,3	91,3	167,7	1,0	168,7
4	6,0	0,3	6,3	91,3	0,4	91,7	176,0	0,8	176,8
5	6,3	0,1	6,4	93,7	0,4	94,1	176,8	0,2	177,0
6	6,4	0,1	6,5	91,3	5,0	96,3	177,0	0,2	177,2
7	6,3	5,8	12,2	96,3	6,7	103,0	176,8	14,6	191,4
8	12,2	20,0	32,2	103,0	5,8	108,8	191,4	50,0	241,4
9	32,2	5,0	37,2	96,3	40,0	136,3	241,4	12,5	253,9
10	37,2	9,0	46,2	136,3	3,0	139,3	253,9	22,5	276,4
11	46,2	0,6	46,8	139,3	3,3	142,6	276,4	1,5	277,9
12	3,1	40,0	43,1	142,6	1,3	143,9	168,7	100,0	268,7
13	46,8	3,0	49,8	143,9	3,3	147,2	277,9	7,5	285,4
14	49,8	4,5	54,3	147,2	1,5	148,7	285,4	11,3	296,7
15	54,3	1,0	55,3	148,7	22,5	171,2	296,7	2,5	299,2
16	55,3	4,5	59,8	103,0	1,5	104,5	299,2	11,3	310,4
17	59,8	0,6	60,4	104,5	3,3	107,8	310,4	1,5	311,9
18	60,4	5,0	65,4	107,8	1,7	109,5	311,9	12,5	324,4
19	65,4	5,0	70,4	109,5	4,0	113,5	324,4	12,5	336,9
20	70,4	4,0	74,4	113,5	1,5	115,0	336,9	10,0	346,9
21	74,4	3,3	77,7	115,0	40,0	155,0	346,9	8,3	355,3
22	77,7	1,4	79,1	155,0	40,0	195,0	355,3	3,5	358,8
23	79,1	8,3	87,4	195,0	10,0	205,00	358,8	20,8	379,6
24				171,2	3,0	174,2			
25				174,2	5,3	179,2			
26				179,2	1,3	180,5			
27				180,5	5,3	185,8			
28				185,8	0,7	186,5			
29				205,0	5,0	210,0			
30				210,0	7,0	217,0			
31				217,0	1,7	218,7			
32				218,7	7,0	225,7			
33				225,7	3,0	228,7			
34				228,7	8,3	237,0			

Selanjutnya, rekap data hasil penjadwalan secara rinci ditampilkan pada Tabel 7.

Completion time pada Tabel 7 masih merupakan data mentah dari penjadwalan ini karena waktu ini merupakan waktu penyelesaian bukan waktu produksi yang dibutuhkan oleh masing-masing *job*. Untuk mengetahui lamanya waktu produksi yang dibutuhkan oleh masing-masing produk dapat dihitung sesuai dengan algoritma pada langkah 18 atau dengan menggunakan rumus (2) dan akan ditampilkan sebagai berikut.

Rumus (2) : $C_i = (C_{i,j=p,k(x)}) - r_i$

- *Job 1*

$$C_1 = (C_{1,23,k(x)}) - r_1 = 87,3 - 0 = 87,3 \text{ jam}$$

- *Job 2*

$$C_2 = (C_{2,34,k(x)}) - r_2 = 237 - 81 = 156 \text{ jam}$$

- *Job 3*

$$C_3 = (C_{3,23,k(x)}) - r_3 = 379,6 - 161 = 218,6 \text{ jam}$$

Dari perhitungan tersebut *job 1* membutuhkan waktu produksi selama 87,3 jam, *job 2* selama 156 jam, dan *job 3* selama 218,6 jam. Nilai ini didapatkan dari selisih antara waktu *completion* dengan waktu siap dari masing-masing *order*-nya dimana waktu siap dari tiap *order* adalah berbeda sehingga akan mempengaruhi nilai dari *completion time* dan lama waktu produksinya. Selanjutnya keterangan tanggal dimana setiap *job* dapat selesai ditampilkan pada Tabel 8.

Tabel 8. Rekap Perhitungan Lama Penjadwalan dan Tanggal Penyelesaiannya

<i>Job</i>	Tanggal <i>Order</i>	C_i (jam)	Jam Kerja	Lama hari	Tanggal Selesai
1	22-12-2013	87,3	8 jam	11	8-01-2014
2	06-01-2014	156	8 jam	20	04-02-2014
3	16-01-2014	218,6	8 jam	28	25-02-2014

Berdasarkan Tabel 8 didapat keterangan mengenai lama hari penyelesaian dan tanggal selesai dari masing-masing *job*. Nilai lama hari penyelesaian didapatkan dari perhitungan waktu produksi yang dibutuhkan oleh masing-masing *job* sesuai dengan rumus (4-12). Nilai ini kemudian dikonversikan ke dalam satuan hari dan diperkirakan tanggal selesainya tanpa memperhatikan hari sabtu dan minggu, dengan kata lain perusahaan tidak melakukan lembur untuk menyelesaikan tiga *job* tersebut. Selanjutnya, dapat dihitung nilai dari *lateness*, *tardiness* hingga besarnya penalti dari *job* tersebut. Berikut tahapan dalam menghitung *lateness*, *tardiness*, dan penalti dari penjadwalan dengan berdasarkan urutan EDD.

- Perhitungan *lateness*.

$$Job\ 1 \rightarrow L_1 = C_1 - D_1 = 8 \text{ Januari } 2014 - 22 \text{ Januari } 2013 = -9 \text{ hari}$$

$$Job\ 2 \rightarrow L_2 = C_2 - D_2 = 04 \text{ Februari } 2014 - 20 \text{ Februari } 2014 = -12 \text{ hari}$$

$$Job\ 3 \rightarrow L_3 = C_3 - D_3 = 25 \text{ Februari } 2014 - 17 \text{ Maret } 2014 = -14 \text{ hari}$$

- Perhitungan *tardiness*

$$Job\ 1 \rightarrow T_1 = \text{Max}\{0, L_1\} = \text{Max}\{0, -9\} = 0 \text{ hari}$$

$$Job\ 2 \rightarrow T_2 = \text{Max}\{0, L_2\} = \text{Max}\{0, -12\} = 0 \text{ hari}$$

$$Job\ 3 \rightarrow T_3 = \text{Max}\{0, L_3\} = \text{Max}\{0, -14\} = 0 \text{ hari}$$

- Perhitungan penalti

Job 1 → $P_1 = 5^0/00 \times T_1 \times \text{nilai kontrak} = 0$, karena tidak terjadi keterlambatan, maka penjadwalan untuk *job 1* tidak menimbulkan biaya penalti.

Job 2 → $P_2 = 5^0/00 \times T_2 \times \text{nilai kontrak} = 0$, karena tidak terjadi keterlambatan, maka penjadwalan untuk *job 2* tidak menimbulkan biaya penalti.

Job 3 → $P_3 = 5^0/00 \times T_3 \times \text{nilai kontrak} = 0$, karena tidak terjadi keterlambatan, maka penjadwalan untuk *job 3* tidak menimbulkan biaya penalti.

Berdasarkan perhitungan poin diatas dapat diketahui hasil dari perhitungan masing-masing *job* untuk *completion time*, *lateness*, dan *tardiness*-nya. Lama waktu penyelesaian (*completion time*) *job 1* adalah 87,3 jam; *job 2* adalah 156 jam; dan *job 3* adalah 218,6 jam. Lama waktu keterlambatan (*lateness*) untuk *job 1* adalah -9 hari, *job 2* adalah -12 hari, dan *job 3* adalah -14 hari. Nilai negatif (-) dari hasil perhitungan *lateness* menandakan bahwa *job* yang bersangkutan dapat selesai lebih awal dari *due date* yang telah ditentukan, untuk ketiga *job* ini semuanya memiliki nilai yang negatif sehingga dapat dikatakan bahwa seluruh *job* dapat selesai sebelum tanggal *due date*-nya dan tidak menyebabkan keterlambatan. Total waktu keterlambatan (*tardiness*) untuk semua *job* adalah nol atau tidak terjadi sehingga prioritas urutan penjadwalan ini tidak menimbulkan biaya penalti juga. Untuk lebih ringkasnya, hasil yang didapat pada poin diatas akan ditampilkan pada Tabel 9 agar memberikan informasi dari penjadwalan ini secara keseluruhan.

Metode selanjutnya yang digunakan

untuk menentukan jadwal inisiasi adalah dengan menggunakan metode FCFS dimana *job* yang datang terlebih dahulu akan langsung diproses pada mesin yang *eligible*. Sesuai dengan Tabel 4, pengurutan dengan menggunakan metode ini menghasilkan urutan penjadwalan *job* yang sama dengan metode EDD, yaitu *job* 1-*job* 2-*job* 3, sehingga apabila dilakukan perhitungan untuk penjadwalan maka akan memberikan hasil yang sama. Hal ini dikarenakan waktu kedatangan *job* dan waktu *due date* dari masing-masing *job* berbanding lurus, dengan kata lain *job* yang datang lebih awal juga memiliki waktu *due date* yang lebih awal juga. Mempertimbangkan hal ini, maka perhitungan menggunakan metode FCFS tidak dilakukan karena akan memberikan hasil yang sama. Namun untuk kasus yang berbeda, perhitungan dengan menggunakan metode FCFS bisa saja tetap diperhitungkan apabila waktu *order* datang tidak berbanding lurus dengan waktu *due date*-nya.

Tabel 9. Rekap Perhitungan Waktu Penyelesaian dan Penalti

Job	Due date (D _i)	C _i	Lateness (L _i)	Tardiness (T _i)	Penalti (P _i)
1	22-01-2014	08-01-2014	-9 hari	0	0
2	20-02-2014	04-02-2014	-12 hari	0	0
3	17-03-2014	25-02-2014	-14 hari	0	0

3.6 Prioritas Pengurutan Algoritma Tabu Search

Sebelum memulai menghitung dengan menggunakan Algoritma Tabu Search, dimulai dengan mendefinisikan notasi-notasi yang dipergunakan selama perhitungan, antara lain sebagai berikut.

$k = 1$; $Move = 2$

$N = 6$, nilai ini didapat dari peluang berdasarkan aturan permutasi tanpa perulangan. $G(S_0) = 0$, merupakan nilai optimal awal yang digunakan sebagai acuan untuk menentukan solusi selanjutnya.

Berdasarkan solusi tersebut, maka perhitungan dengan menggunakan algoritma Tabu Search dapat dimulai dengan langkah-langkah sebagai berikut (Pinedo, 2009).

Langkah 1

Atur nilai $k=1$

Pilih satu urutan inisiasi S_1 menggunakan metode heuristik tertentu

Atur $S_0 = S_1$

Langkah 2

Pilih satu jadwal kandidat S_c dari solusi tetangga S_k

Jika *move* dari $S_k \rightarrow S_c$ terlarang oleh mutasi dalam *tabu-list*

Maka atur $S_{k+1} = S_k$ dan lanjut ke langkah 3

Jika *move* dari $S_k \rightarrow S_c$ tidak terlarang oleh mutasi manapun dalam *tabu-list*

Maka atur $S_{k+1} = S_c$ dan

masukkan mutasi kebalikannya dalam *tabu-list* yang paling atas

Pindah semua masukan dalam *tabu-list* satu posisi ke bawah dan hapus masukan paling akhir di dalam *tabu-list*

Jika $G(S_c) < G(S_0)$, atur $S_0 = S_c$;

Lanjut ke langkah 3

Langkah 3

Tambahkan nilai k dengan 1

Jika nilai $k = N$ maka BERHENTI

Jika tidak, kembali ke langkah 2

Dalam jurnal ini hanya diberikan contoh perhitungan iterasi pertama saja, hasil yang lebih lengkap akan diberikan penjelasan setelah Tabel 11.

1. Iterasi Pertama

- $k = 1$

S_1 menggunakan metode EDD = 1-2-3

$S_0 = S_1$

- $S_c = 2-1-3$ dan $S_c = 1-3-2$

Move terlarang? Tidak.

Maka $S_{c1} = 2-1-3 \rightarrow Tabu\ list = \{(1,2)\}$

$S_{c2} = 1-3-2 \rightarrow Tabu\ list = \{(2,3)\}$

Cek $G(S_c) < G(S_0)$? $G(S_0) = 0$ hari

$G(S_{c1}) = 0, \rightarrow G(S_{best})$

$G(S_{c2}) = 0,$

Tabel 10. Hasil Jadwal Algoritma Tabu Search Iterasi Pertama Solusi 1 ($G(S_{c1})$)

	s_i	Due date	C	Lateness	Tardiness	Penalti
Job 1	22-Dec-13	22-Jan-13	08-Jan-14	-9	0	0
Job 2	06-Jan-14	20-Feb-14	04-Feb-14	-12	0	0
Job 3	16-Jan-14	17-Mar-14	25-Feb-14	-14	0	0

Tabel 11. Hasil Jadwal Algoritma Tabu Search Iterasi Pertama Solusi 2 ($G(S_{c2})$)

	s_i	Due date	C _i	Lateness	Tardiness	Penalti
Job 1	22-Dec-13	22-Jan-13	08-Jan-14	-9	0	0
Job 2	06-Jan-14	20-Feb-14	04-Feb-14	-12	0	0
Job 3	16-Jan-14	17-Mar-14	25-Feb-14	-14	0	0

Tabu-list: $\{(1,2)\}$

- $k = 1+1 = 2,$

$k = 6$? Tidak, maka lanjut ke iterasi kedua.

Dapat dilihat pada iterasi pertama ada dua alternatif solusi penyelesaian yang bisa dipertimbangkan, yaitu jadwal dengan prioritas urutan 2-1-3 dan 1-3-2. Kedua solusi ini dihitung nilai *tardiness*-nya dan dibandingkan dengan nilai *tardiness* yang dimiliki oleh solusi acuannya (G_{best}). Hasilnya, kedua solusi ini memiliki nilai optimal yang sama dengan solusi inisiasinya. Namun karena pada iterasi ini hanya dipilih satu solusi saja maka, jadwal dengan urutan 2-1-3 dipilih. Jadwal ini memiliki *move* (1,2) dan dimasukkan ke dalam *tabu List*. Langkah selanjutnya dapat dilakukan dengan mendefinisikan nilai k yang baru, yakni 2 yang menandakan iterasi kedua.

Prioritas perhitungan menggunakan algoritma ini mempunyai 6 iterasi dan masing-masing memiliki satu solusi optimal yang akan dibandingkan dengan nilai *tardiness* dari solusi inisiasinya (acuannya). Berdasarkan 6 iterasi yang dilakukan menghasilkan 3 solusi jadwal dengan urutan 2-1-3, 3-2-1, dan 1-3-2 dimana tiap jadwal ini mempunyai nilai yang sama dengan jadwal inisialnya, yaitu membutuhkan waktu produksi untuk *job* 1 selama 87,3 jam; *job* 2 membutuhkan waktu selama 156 jam; dan *job* 3 membutuhkan waktu produksi selama 218,6 jam.

Disamping perhitungan manual dengan menggunakan algoritma sequencing, jadwal juga dibuat dengan menginterpretasikannya ke dalam *Gantt Chart* untuk mengetahui apakah Algoritma *Sequencing* telah memenuhi logika penjadwalan. Berdasarkan hasil penjadwalan dengan menggunakan *Gantt Chart*, dapat dilihat bahwa waktu produksi yang dibutuhkan untuk masing-masing *job* menunjukkan hasil yang sama dari perbandingan antara pengerjaan dengan menggunakan algoritma *sequencing* dan hasil *Gantt Chart*-nya, sehingga hasil ini menunjukkan bahwa logika algoritma *sequencing* telah sesuai dengan tujuan penelitian, yaitu untuk meminimasi *tardiness* dan penalti. Gambar *Gantt Chart* lebih jelasnya dapat dilihat pada Lampiran 1.

Hasil ini menunjukkan bahwa seluruh *job* dapat selesai sebelum waktu *due date*-nya, dengan kata lain tujuan penelitian untuk memenuhi waktu *due date*-nya telah tercapai. Selanjutnya perhitungan terhadap kemungkinan membayar penalti dilakukan. Besarnya penalti bergantung pada lamanya keterlambatan dan nilai kontraknya. Hasil penelitian menunjukkan bahwa tidak ada *job* yang terlambat sehingga

perusahaan pun tidak perlu menanggung biaya penalti akibat keterlambatan, dengan kata lain tujuan kedua dari penelitian ini juga terpenuhi. Berdasarkan beberapa penjelasan tersebut menyimpulkan bahwa metode yang diusulkan pada penelitian ini dapat digunakan oleh perusahaan sebagai salah satu alternatif saran perbaikan untuk mengatasi permasalahan *tardiness* dan penalti yang sering terjadi sebelumnya.

4. Kesimpulan

Dari hasil pengolahan data dan analisisnya, terdapat beberapa kesimpulan, yaitu sebagai berikut:

1. Penjadwalan yang diusulkan dalam penelitian ini adalah dengan menggunakan perbandingan antara metode FCFS dan EDD sebagai jadwal inisiasi awal. Selanjutnya, untuk pengoptimalan jadwal lebih lanjut menggunakan Algoritma *Tabu Search* dimana dalam setiap tahapan iterasinya menggunakan prinsip *Pairwise Interchange Heuristic*. Hasil dari masing-masing tahap akan dijelaskan lebih lanjut sebagai berikut:
 - a. Prioritas jadwal dengan menggunakan metode EDD mempunyai hasil yang sama dengan metode FCFS, yaitu 1-2-3. Hal ini karena secara kebetulan waktu datang tiap *order* berbanding lurus dengan *due date*-nya, sehingga *order* yang datang pertama kali juga mempunyai waktu *due date* yang lebih awal juga. Urutan menggunakan metode EDD (1-2-3) membutuhkan waktu produksi secara berurutan selama 87,3 jam; 156 jam; dan 218,6 jam, atau setara dengan 11 hari, 20 hari dan 28 hari. Selain itu, *job* 1, *job* 2, dan *job* 3 dapat diselesaikan 9 hari, 12 hari dan 14 hari lebih cepat sebelum *due date*-nya.
 - b. Hasil prioritas urutan jadwal lebih lanjut menggunakan Algoritma *Tabu Search* menghasilkan jadwal yang nilai *tardiness* dan penaltinya sama dengan jadwal inisialnya. Hasilnya terdapat tiga alternatif urutan jadwal yang bisa diusulkan, yaitu 2-1-3, 3-2-1, dan 1-3-2. Hal ini dapat terjadi karena setiap *job* datang satu persatu dan tidak ada beberapa *job* yang dikerjakan secara bersamaan dalam satu waktu. Sehingga

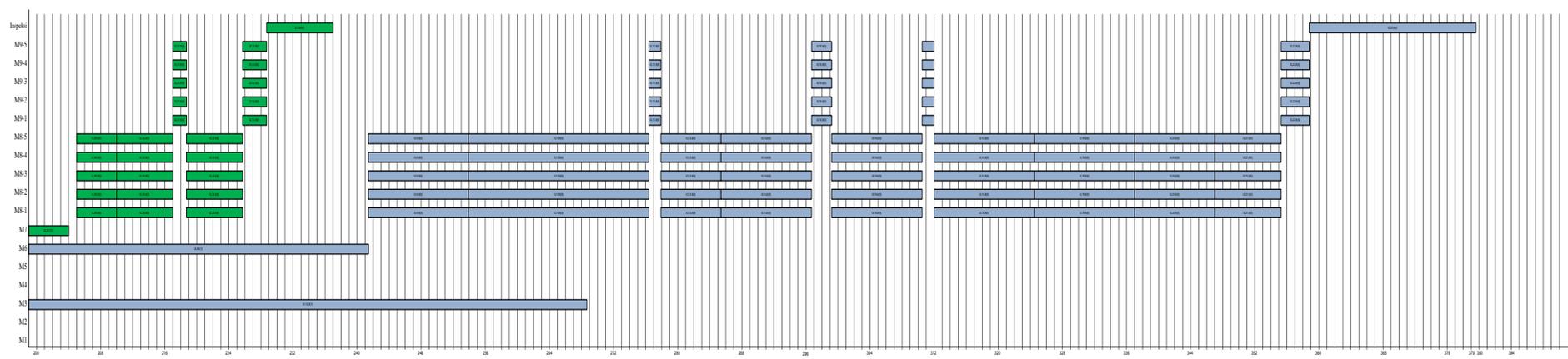
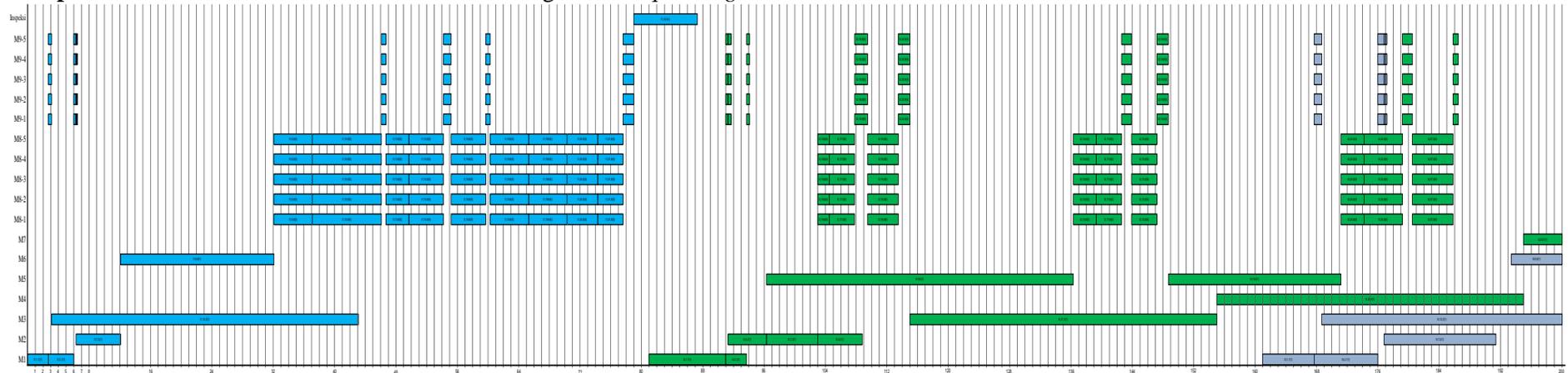
- hal ini dapat menghasilkan beberapa alternatif jadwal yang tetap bisa meminimasi *tardiness* dan penalti.
2. Berdasarkan hasil pengolahan data dengan menggunakan prioritas penjadwalan yang diusulkan, seluruh *job* dapat diselesaikan sebelum *due date*-nya Berikut ringkasan hasil perhitungan tiap *order*:
 - a. Lama waktu produksi yang dibutuhkan oleh *job* 1 adalah 11 hari, dapat selesai 9 hari sebelum *due date*, dan tidak menyebabkan *tardiness*.
 - b. Lama waktu produksi yang dibutuhkan oleh *job* 2 adalah 20 hari, dapat selesai 12 hari sebelum *due date*, dan tidak menyebabkan *tardiness*.
 - c. Lama waktu produksi yang dibutuhkan oleh *job* 3 adalah 28 hari, dapat selesai 14 hari sebelum *due date*, dan tidak menyebabkan *tardiness*.
 3. Hasil penjadwalan usulan terhadap jadwal *existing* menunjukkan pengurangan nilai *tardiness* yang cukup signifikan, dijelaskan sebagai berikut:
 - a. Pada jadwal *existing* *job* 1, *job* 2 dan *job* 3 secara berurutan mengalami keterlambatan (*tardiness*) selama 12 hari, 13 hari dan 20 hari.
 - b. Pada jadwal dengan menggunakan metode usulan tidak menyebabkan keterlambatan (*tardiness*) bagi seluruh *job* yang dijadwalkan.
 4. Berdasarkan hasil jadwal usulan, tidak terjadi penalti karena besarnya nilai penalti bergantung pada nilai *tardiness*. Jadi, apabila nilai *tardiness* sama dengan nol (tidak terjadi keterlambatan) maka jadwal juga tidak menyebabkan penalti yang harus dibebankan kepada perusahaan. Sehingga

metode yang diusulkan dalam penelitian ini dapat digunakan sebagai alternatif saran rekomendasi perbaikan bagi perusahaan.

Daftar Pustaka

- Andriani, Debrina Puspita, Astuti, Murti & Setyanto, Nasir Widha. (2010). *Penjadwalan Produksi dengan Kendala Resource Sharing pada Proses Sakarifikasi Menggunakan Metode Backward Scheduling* (Studi Kasus di PT. Sorini Agro Asia Corporindo Tbk, Pasuruan). Skripsi tidak dipublikasikan. Universitas Brawijaya, Malang.
- Berlianty, Intan & Miftahol Arifin. (2010). *Teknik-teknik Optimasi Heuristik*. Yogyakarta : Graha Ilmu.
- Conway, Richard Walter, dkk. (1967). *Theory of Scheduling*. New York : Dover Publication, Inc.
- Ginting, Rosnani. (2009). *Penjadwalan Mesin*. Yogyakarta : Graha Ilmu
- Mahmudy, Wayan F., Romeo M. Marian & Lee H. S. Luong. (2013). *Real Coded Genetic Algorithms for solving Flexible Job-Shop Scheduling Problem-Part I : Modelling*. Advanced Material Research Vol.701 (2013) Pp 359-363.
- Pinedo, M. (2008). *Scheduling*. New York : Springer.
- Pinedo, M. (2009). *Planning and Scheduling in Manufacturing and Services*. New York : Springer.

Lampiran 1. Gambar *Gantt Chart* Jadwal Produksi Algoritma *Sequencing* Prioritas *Job 1-2-3*



- | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <p>Axix = Mesin-mesin yang tersedia</p> <p>Ordinat = Waktu Produksi (jam)</p> <p> = Job 1</p> <p> = Job 2</p> | <p> = Job 3</p> <p>M1 = Mesin Gas Cutting</p> <p>M2 = Mesin Bevelling</p> | <p>M3 = Mesin Head Dishing</p> <p>M4 = Mesin Head Flanging</p> <p>M5 = Mesin Roll and Bend</p> <p>M6 = Mesin Press Hydraulic</p> | <p>M7 = Mesin Gas Turning Table</p> <p>M8 = Mesin Paralel Las</p> <p>M9 = Mesin Paralel Gerind</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|