

PENENTUAN JADWAL PRODUKSI MENGGUNAKAN ALGORITMA *CAMPBELL DUDEK SMITH* DAN *DANNENBRING* DALAM OPTIMASI WAKTU PRODUKSI

Made Febri Ananda Pramana^{1)*}, Hendro Prasetyo²⁾, Yanti Helianty³⁾

^{1,2,3)}Program Studi Teknik Industri, Fakultas Teknologi Industri,
Institut Teknologi Nasional, Bandung, Indonesia

*Corresponding Author : Made.febri@mhs.itenas.ac.id

ABSTRAK

Penjadwalan flow shop dalam industri manufaktur memerlukan metode yang efisien untuk meminimalkan makespan dan keterlambatan produksi. Metode *First Come First Served* (FCFS) sering kali kurang optimal karena tidak mempertimbangkan waktu proses pekerjaan, sehingga menyebabkan inefisiensi. Penelitian ini menerapkan algoritma *Campbell Dudek Smith* (CDS) dan *Dannenbring* untuk mengoptimalkan penjadwalan produksi. Algoritma CDS menghasilkan total *makespan* sebesar 9602 menit, sedangkan algoritma *Dannenbring* menghasilkan total *makespan* sebesar 9480,22 menit dan setelah *overlapping* menjadi 8794,37 menit. Algoritma ini mengurangi keterlambatan dari yang semula 9 *job* terlambat, menjadi tidak ada keterlambatan.

Kata kunci: *Penjadwalan, Campbell Dudek Smith, Dannenbring*

1. PENDAHULUAN

1.1 Latar Belakang

Penjadwalan *flow shop* adalah salah satu masalah penjadwalan yang sering ditemukan dalam dunia manufaktur. Sistem ini melibatkan pekerjaan yang harus diproses melalui serangkaian mesin dalam urutan yang telah ditentukan, di mana setiap mesin mengikuti urutan pengerjaan yang sama (Prasetyo & Heryati, 2024). Berbagai metode telah dikembangkan untuk menyelesaikan masalah penjadwalan *flow shop*, di antaranya adalah metode *Campbell Dudek Smith* (CDS) dan *Dannenbring*. Algoritma CDS adalah pengembangan dari aturan yang telah dikemukakan oleh *Johnson*, di mana setiap pekerjaan atau tugas yang akan diselesaikan harus melewati proses masing-masing mesin untuk mendapatkan harga *makespan* yang terkecil. Algoritma *Dannenbring* merupakan sebuah algoritma yang menggabungkan algoritma CDS dengan konsep *slope index* (Ginting, 2009).

Metode *First Come First Served* (FCFS), yang mengatur urutan pengerjaan berdasarkan kedatangan *job*, sering kali kurang efektif dalam lingkungan produksi dengan variasi proses yang kompleks. Penjadwalan dengan aturan ini banyak menyebabkan keterlambatan karena penjadwalan dilakukan berdasarkan urutan kedatangan dan bukan berdasarkan lamanya waktu proses produksi. Keterlambatan tersebut jika tidak segera diatasi, akan menyebabkan penurunan kepuasan pelanggan sehingga memperbesar resiko kehilangan konsumen. Permasalahan pada perusahaan ini menarik perhatian peneliti untuk melakukan penelitian mengenai penjadwalan produksi untuk meminimumkan persentase keterlambatan dengan meminimumkan waktu produksi (Pinedo, 2016). *makespan* yang rendah diharapkan mampu menurunkan persentase keterlambatan.

Penelitian ini menggunakan algoritma CDS dan *Dannenbring* dalam melakukan penjadwalan produksi. Algoritma CDS dapat menangani masalah penjadwalan *flow*

shop dengan lebih dari dua mesin serta lebih mudah diterapkan dan memerlukan waktu komputasi lebih rendah dibandingkan metode optimasi yang kompleks (Campbell dkk., 1970). Sementara algoritma *Dannenbring* mampu menangani lebih banyak mesin dan mengurangi kompleksitas perhitungan dibanding metode CDS. Selain itu, algoritma ini mempertimbangkan waktu pemrosesan rata-rata untuk seluruh mesin sehingga dalam beberapa skenario sering kali lebih unggul dibandingkan algoritma CDS (Dannenbring, 1977).

1.2 Tujuan Penelitian

Penelitian ini bertujuan untuk menentukan jadwal produksi yang menghasilkan *makespan* tersingkat menggunakan algoritma CDS dan *Dannenbring*.

2. METODE

2.1 Metode Pengumpulan Data

Data yang diperlukan dalam penelitian ini diperoleh melalui wawancara. Data-data tersebut antara lain; jumlah mesin pada setiap stasiun kerja, tanggal pesanan masuk dalam rentang waktu 1 bulan, tanggal penyerahan pesanan, urutan proses produksi, dan waktu operasi setiap stasiun kerja.

2.2 Campbell Dudek Smith (CDS)

Algoritma CDS bertujuan untuk memperoleh *makespan* terkecil dari beberapa alternatif yang tersedia (Mashuri dkk., 2019). Langkah-langkah dalam penjadwalan algoritma CDS yaitu:

- Ambil penjadwalan pertama ($k = 1$). Untuk seluruh *job* yang ada, cari harga $t_{i,1}^*$ dan $t_{i,2}^*$ yang minimum yang merupakan waktu proses pada mesin pertama dan kedua dengan rumus berikut:

$$t_{i,1} = t_{i,1}^* \quad (1)$$

$$t_{i,2} = t_{i,2}^* \quad (2)$$

Keterangan:

$t_{i,1}^*$ = waktu proses terbaik di mesin 1

$t_{i,2}^*$ = waktu proses terbaik di mesin m

- Jika waktu minimum didapat pada mesin pertama ($t_{i,1}$), selanjutnya tempatkan tugas tersebut pada awal deret penjadwalan dan bila waktu minimum didapat pada mesin kedua ($t_{i,2}$), tugas tersebut ditempatkan pada posisi akhir dari deret penjadwalan.
- Pindahkan tugas-tugas tersebut dari daftarnya dan susun dalam bentuk deret penjadwalan. Jika masih ada *job* yang tersisa, ulangi kembali langkah (a). Sebaliknya, jika tidak ada lagi *job* yang tersisa, maka penjadwalan telah selesai. Dengan demikian, waktu proses dari kedua mesin yaitu mesin pertama ($t_{i,1}^*$) dan mesin kedua ($t_{i,2}^*$) pada penjadwalan ke- k adalah:

$$t_{i,1}^* = \sum_{k=1}^k t_{i,k}^* \quad (3)$$

$$t_{i,2}^* = \sum_{k=1}^k t_{1,m-k+1}^* \quad (4)$$

- Jika jadwal ke- $k = (m - 1)$ sudah tercapai, maka penjadwalan *job* telah selesai.

2.3 Dannenbring

Algoritma *Dannenbring* ini hanya menghasilkan satu urutan pengerjaan *job* dan bertujuan memberikan solusi dengan hasil yang lebih optimal (Masrurah, 2008). Langkah-langkah penjadwalan algoritma ini yaitu:

- Tentukan nilai P_{j1} dan P_{j2} dengan menggunakan rumus berikut:

$$P_{i1} = \sum_{j=1}^M (M - j + 1) \cdot t_{ij} \quad (5)$$

$$P_{i2} = \sum_{j=1}^M (j) \cdot t_{ij} \quad (6)$$

Keterangan:

P_{i1} = waktu proses *job* ke- *i* dalam mesin ke- 1

P_{i2} = waktu proses *job* ke- *i* dalam mesin ke- 2

j = mesin ke- *j*

- b. Setelah nilai P_{i1} dan P_{i2} didapatkan, urutkan *job* tersebut. Jika waktu terkecil berada pada P_{i1} maka *job* tersebut ditempatkan pada urutan awal. Sebaliknya, jika waktu terkecil berada pada P_{i2} maka *job* ditempatkan pada urutan akhir.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

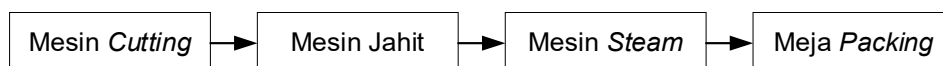
Data yang digunakan dalam penelitian ini adalah sebagai berikut:

- a. Data Permintaan Bulan September 2024
Data permintaan pada bulan September 2024 dapat dilihat pada Tabel 1.

Tabel 1. Data Permintaan Bulan September 2024

<i>Job</i>	Produk	Jumlah (Pcs)	<i>Due Date</i>	<i>Job</i>	Produk	Jumlah (Pcs)	<i>Due Date</i>
1	Sprei	10	12/9/2024	9	Sprei	459	21/9/2024
2	Sprei	10	12/9/2024	10	Sprei	73	17/9/2024
3	Sprei	10	12/9/2024	11	Sprei	65	17/9/2024
4	Sprei	10	12/9/2024	12	<i>T-shirt</i>	1212	30/9/2024
5	Sprei	10	12/9/2024	13	<i>T-shirt</i>	68	20/9/2024
6	Sprei	73	14/9/2024	14	<i>T-shirt</i>	250	20/9/2024
7	Sprei	65	14/9/2024	15	<i>T-shirt</i>	2061	1/10/2024
8	Sprei	224	17/9/2024	16	<i>T-shirt</i>	256	3/10/2024

- b. Data Urutan Proses Pemesinan
Proses pemesinan untuk memproduksi sprei dan *T-shirt* melalui proses dan mesin yang sama. Urutan proses dapat dilihat pada Gambar 1.



Gambar 1. Urutan Proses Pemesinan

- c. Data Jumlah Mesin
Data jumlah mesin yang dimiliki perusahaan dapat dilihat pada Tabel 2.

Tabel 2. Data Jumlah Mesin

Kode Mesin	Nama Mesin	Jumlah (Unit)
A	Mesin <i>Cutting</i>	2
B	Mesin Jahit	30
C	Mesin <i>Steam</i>	2
D	Meja <i>Packing</i>	2

d. Data Waktu Proses

Data waktu proses untuk setiap stasiun kerja dapat dilihat pada Tabel 3.

Tabel 3. Data Waktu Proses

Job (i)	Mesin A (Menit/Pcs)	Mesin B (Menit/Pcs)	Mesin C (Menit/Pcs)	Mesin D (Menit/Pcs)
1	2,08	7,79	5,93	5,32
2	2,51	8,02	5,89	5,65
3	2,56	8,33	6,09	5,58
4	1,97	7,93	5,71	4,93
5	2,35	8,26	5,90	5,08
6	2,57	7,80	6,42	5,76

Tabel 3. Data Waktu Proses (Lanjutan)

Job (i)	Mesin A (Menit/Pcs)	Mesin B (Menit/Pcs)	Mesin C (Menit/Pcs)	Mesin D (Menit/Pcs)
7	2,69	7,93	6,39	5,52
8	2,48	8,25	5,94	5,59
9	2,69	8,18	6,29	5,83
10	2,24	8,21	6,02	4,98
11	2,62	8,39	6,05	5,22
12	3,35	15,74	1,66	2,84
13	3,28	16,07	1,50	3,04
14	3,07	16,62	1,54	3,00
15	2,94	15,93	1,87	2,71
16	3,03	16,47	1,79	2,82

Karena perusahaan memiliki mesin lebih dari satu unit, maka diperlukan penyesuaian terhadap waktu proses. Penyesuaian waktu proses dapat dilihat pada Tabel 4.

Tabel 4. Waktu Ekuivalensi

Job (i)	Mesin A (Menit/batch)	Mesin B (Menit/pcs)	Mesin C (Menit/pcs)	Mesin D (Menit/pcs)
1	2,08	77,90	59,30	53,20
2	2,51	80,20	58,90	56,50
3	2,56	83,30	60,90	55,80
4	1,97	79,30	57,10	49,30
5	2,35	82,60	59,00	50,80
6	2,57	18,98	234,33	210,24
7	2,69	17,18	207,68	179,40
8	2,48	61,60	665,28	626,08
9	2,69	125,15	1443,56	1337,99
10	2,24	19,98	219,73	181,77
11	2,62	18,18	196,63	169,65
12	10,05	635,90	1005,96	1721,04
13	3,28	36,42	51,00	103,36
14	3,07	138,50	192,50	375,00

15	14,70	1094,39	1927,04	2792,66
16	3,03	140,54	229,12	360,96

3.2 Penjadwalan Produksi Menggunakan Metode FCFS

Berdasarkan data permintaan, order mulai masuk ke lantai produksi pada tanggal 11 September 2024 dan selesai diproduksi pada tanggal 5 Oktober 2024. Total *makespan* yang dihasilkan adalah 25 hari dengan keterlambatan pada 9 *job* yaitu *job* 8, *job* 9, *job* 10, *job* 11, *job* 12, *job* 13, *job* 14, *job* 15, dan *job* 16.

3.3 Penjadwalan Produksi Menggunakan Algoritma Campbell Dudek Smith (CDS)

Penjadwalan produksi dengan algoritma CDS dilakukan untuk beberapa kondisi. Kondisi tersebut antara lain:

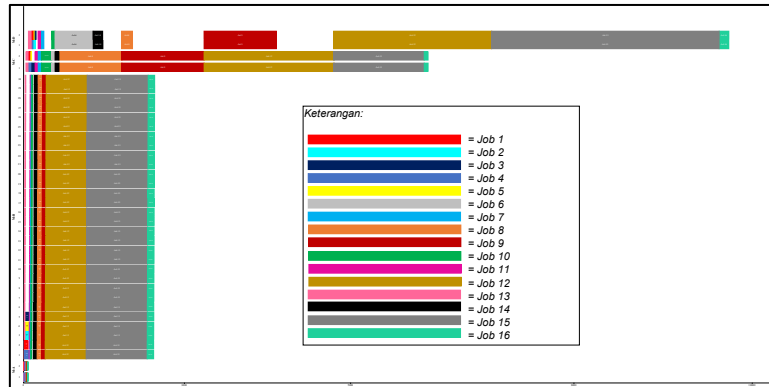
- Berdasarkan waktu ekivalensi
- Berdasarkan waktu ekivalensi dengan modifikasi pembebanan mesin terkecil, dimana penugasan *job* dengan jumlah kecil dilakukan berdasarkan mesin dengan beban kerja terkecil.
- Berdasarkan waktu proses untuk satu mesin
- Berdasarkan waktu proses dengan modifikasi pembebanan mesin terkecil, dimana penugasan *job* dengan jumlah kecil dilakukan berdasarkan mesin dengan beban kerja terkecil.
- Berdasarkan waktu proses dengan modifikasi perbaikan pembebanan mesin terkecil, dimana kondisi ini merupakan versi perbaikan dari kondisi sebelumnya.
- Berdasarkan waktu proses dengan modifikasi *express*, dimana kondisi ini mengasumsikan satu mesin dikhususkan mengerjakan *job* dengan jumlah kecil.

Perhitungan dilakukan dengan mengikuti langkah-langkah pengerjaan algoritma CDS untuk setiap kondisi. Rekapitulasi Alternatif terbaik untuk setiap kondisi dapat dilihat pada Tabel 5.

Tabel 5. Rekapitulasi Alternatif Terbaik Algoritma CDS

Kondisi	Alternatif Terbaik	Urutan Job	Makespan (Menit)
a	$k = 3$	13 – 4 – 1 – 2 – 5 – 3 – 11 – 7 – 10 – 6 – 14 – 8 – 9 – 12 – 15 – 16	9876,26
b	$k = 3$	2 – 1 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 16 – 10 – 15 – 14 – 12 – 13 – 11	9602
c	$k = 2$	1 – 4 – 2 – 5 – 3 – 7 – 11 – 6 – 10 – 8 – 15 – 9 – 12 – 16 – 14 – 13	69918,94
d	$k = 3$	2 – 1 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 16 – 10 – 15 – 14 – 12 – 13 – 11	42298,53
e	$k = 3$	2 – 1 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 16 – 10 – 15 – 14 – 12 – 13 – 11	16033,14
f	$k = 1$	4 – 1 – 10 – 5 – 8 – 2 – 3 – 6 – 11 – 7 – 9 – 16 – 14 – 13 – 12 – 15	17459,97

Kondisi terbaik yang menghasilkan *makespan* terkecil adalah algoritma CDS berdasarkan waktu ekivalensi dengan modifikasi pembebanan mesin terkecil pada alternatif ketiga yang menghasilkan *makespan* sebesar 9602 menit (20 hari) dengan urutan *job* yaitu *job* 2 – 1 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 16 – 10 – 15 – 14 – 12 – 13 – 11. *Gantt chart* algoritma ini dapat dilihat pada Gambar 2.



Gambar 2. *Gantt Chart* Algoritma CDS Terpilih

3.3 Penjadwalan Produksi Menggunakan Algoritma *Dannenbring*

Penjadwalan produksi dengan algoritma *Dannenbring* dilakukan untuk beberapa kondisi. Kondisi tersebut antara lain:

- Berdasarkan waktu ekivalensi
- Berdasarkan waktu ekivalensi dengan modifikasi pembebanan mesin terkecil, dimana penugasan *job* dengan jumlah kecil dilakukan berdasarkan mesin dengan beban kerja terkecil.
- Berdasarkan waktu proses untuk satu mesin
- Berdasarkan waktu proses dengan modifikasi pembebanan mesin terkecil, dimana penugasan *job* dengan jumlah kecil dilakukan berdasarkan mesin dengan beban kerja terkecil.
- Berdasarkan waktu proses dengan modifikasi perbaikan pembebanan mesin terkecil, dimana kondisi ini merupakan versi perbaikan dari kondisi sebelumnya.
- Berdasarkan waktu proses dengan modifikasi *express*, dimana kondisi ini mengasumsikan satu mesin dikhususkan mengerjakan *job* dengan jumlah kecil.
- Kondisi modifikasi, dimana pada kondisi ini menggunakan *output* urutan *job* terbaik yang memiliki *makespan* terkecil pada algoritma CDS.

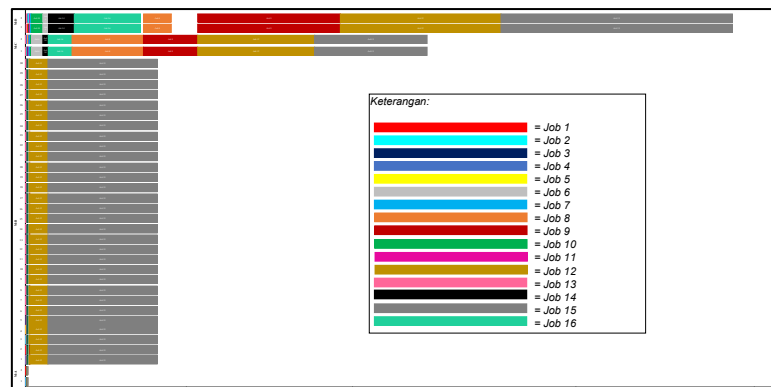
Perhitungan dilakukan dengan mengikuti langkah-langkah pengerjaan algoritma *Dannenbring* untuk setiap kondisi. Rekapitulasi nilai *makespan* untuk setiap kondisi dapat dilihat pada Tabel 6.

Tabel 6. Rekapitulasi Nilai *Makespan* Algoritma *Dannenbring*

Kondisi	Urutan Job	<i>Makespan</i> (Menit)
a	13 - 4 - 1 - 2 - 5 - 3 - 11 - 7 - 10 - 6 - 14 - 16 - 8 - 9 - 12 - 15	9629,69
b	13 - 4 - 1 - 2 - 5 - 3 - 11 - 7 - 10 - 6 - 14 - 16 - 8 - 9 - 12 - 15	9480,22
c	4 - 1 - 2 - 5 - 3 - 7 - 11 - 10 - 6 - 8 - 9 - 15 - 12 - 16 - 14 - 13	69918,83
d	4 - 1 - 2 - 5 - 3 - 7 - 11 - 10 - 6 - 8 - 9 - 15 - 12 - 16 - 14 - 13	42297,90
e	4 - 1 - 2 - 5 - 3 - 7 - 11 - 10 - 6 - 8 - 9 - 15 - 12 - 16 - 14 - 13	16032,52
f	4 - 1 - 2 - 5 - 3 - 7 - 11 - 10 - 6 - 8 - 9 - 15 - 12 - 16 -	18637,25

	14 - 13	
g	13 - 4 - 1 - 2 - 5 - 3 - 11 - 7 - 10 - 6 - 14 - 16 - 8 - 9 - 12 - 15	9629,69

Kondisi terbaik yang menghasilkan *makespan* terkecil adalah algoritma *Dannenbring* berdasarkan waktu ekivalensi dengan modifikasi pembebanan mesin terkecil dengan *makespan* sebesar 9480,22 menit (sekitar 20 hari). Urutan *job* kondisi ini yaitu *job* 13 - 4 - 1 - 2 - 5 - 3 - 11 - 7 - 10 - 6 - 14 - 16 - 8 - 9 - 12 – 15 pada awal proses produksi dan *job* 13 - 4 - 1 - 5 - 2 - 3 - 11 - 7 - 10 - 6 - 14 - 16 - 8 - 9 - 12 – 15 pada saat selesai. *Gantt chart* ini dapat dilihat pada Gambar 3.



Gambar 3. *Gantt Chart* Algoritma *Dannenbring* Terpilih

3.4 Penjadwalan Produksi menggunakan *Overlapping* pada Algoritma Terpilih

Berdasarkan perhitungan yang dilakukan terhadap kedua algoritma, algoritma *Dannenbring* memiliki *makespan* terkecil dibandingkan algoritma CDS. Akan tetapi, hasil tersebut masih bisa dilakukan *overlapping* guna lebih meminimasi *makespan*. *Overlapping* adalah proses pembagian sebuah lot menjadi sedikitnya dua *batch* untuk memangkas waktu produksi. *Overlapping* dilakukan pada *job* 9, *job* 12, dan *job* 15 pada mesin *steam* karena pada *job* tersebut ditemukan *idle time* atau waktu menganggur. Perhitungan *overlapping* menggunakan persamaan sebagai berikut:

$$Q_1 \geq \frac{Q_{PA} - S_B}{P_B + P_A} \quad (5)$$

$$Q_2 \leq Q - Q_1 \quad (6)$$

Keterangan:

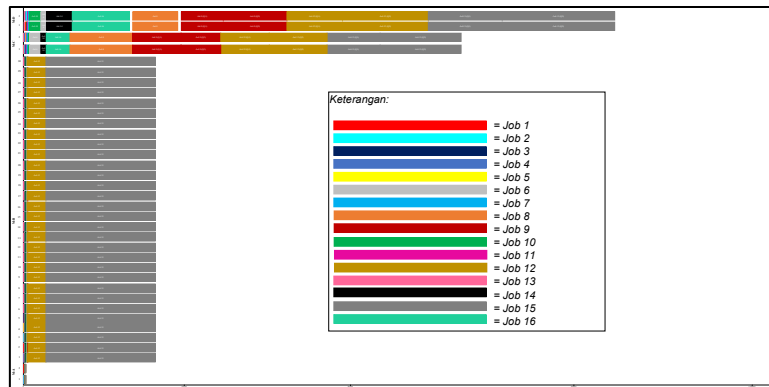
- Q = ukuran lot total
- Q_1 = ukuran minimum dari *batch* pertama
- Q_2 = ukuran maksimum dari *batch* kedua
- S_B = waktu *setup* dari operasi B
- P_A = waktu proses per unit, operasi A
- P_B = waktu proses per unit, operasi B

Rekapitulasi ukuran lot dan waktu proses kondisi *overlapping* ini dapat dilihat pada Tabel 7.

Tabel 7. Ukuran Lot dan Waktu Proses *Overlapping*

Job (i)		Ukuran Lot (Pcs)	Mesin C (Menit)	Mesin D (Menit)
9	Q ₁	239	751,66	696,69
	Q ₂	220	691,90	641,30
12	Q ₁	448	371,84	638,40
	Q ₂	764	634,12	1088,70
15	Q ₁	842	787,27	1140,91
	Q ₂	1219	1139,77	1651,75

Setelah waktu proses didapatkan, langkah selanjutnya adalah menghitung nilai *makespan* dengan menggunakan waktu *job* 9, *job* 12, dan *job* 15 terbaru. Berdasarkan perhitungan yang dilakukan, setelah dilakukan *overlapping* diperoleh total nilai *makespan* sebesar 8794,37 menit atau sekitar 18 hari. *Gantt chart* algoritma *Dannenbring* setelah dilakukan *overlapping* dapat dilihat pada Gambar 4.



Gambar 4. *Gantt Chart* Algoritma *Dannenbring* Terpilih dengan *Overlapping*

3.5 Analisis

Alternatif terpilih untuk algoritma CDS adalah pada kondisi waktu ekivalensi dengan modifikasi pembebanan mesin terkecil yang menghasilkan *makespan* sebesar 9602 menit (sekitar 20 hari), sedangkan untuk algoritma *Dannenbring* adalah pada kondisi serupa dengan *makespan* sebesar 9480,22 menit (sekitar 20 hari) dan setelah dilakukan *overlapping* menjadi 8794,37 menit (sekitar 18 hari). Algoritma *Dannenbring* lebih unggul dibandingkan algoritma CDS karena algoritma tersebut secara eksplisit menempatkan pekerjaan dengan total waktu terlama di belakang, sementara algoritma CDS tidak memiliki aturan eksplisit untuk penempatan *job* dengan waktu terlama. Algoritma ini mengurangi keterlambatan dari yang semula 9 *job* terlambat, menjadi tidak ada keterlambatan. Waktu pengerjaan *job* dengan algoritma ini dapat dilihat pada Tabel 8.

Tabel 8. Rekapitulasi Waktu Pengerjaan *Job* Algoritma *Dannenbring*

Jo b	Completi on Time	Due Date	Keterang an	Jo b	Completi on Time	Due Date	Keterang an
1	11/9/2024	12/9/2024	Tidak Terlambat	9	21/9/2024	21/9/2024	Tidak Terlambat
2	11/9/2024	12/9/2024	Tidak Terlambat	10	12/9/2024	17/9/2024	Tidak Terlambat
3	11/9/2024	12/9/2024	Tidak Terlambat	11	12/9/2024	17/9/2024	Tidak Terlambat
4	11/9/2024	12/9/2024	Tidak Terlambat	12	25/9/2024	30/9/2024	Tidak Terlambat

Tabel 8. Rekapitulasi Waktu Pengerjaan *Job* Algoritma *Dannenbring* (Lanjutan)

Jo b	Completi on Time	Due Date	Keterang an	Jo b	Completi on Time	Due Date	Keterang an
5	11/9/2024	12/9/2024	Tidak Terlambat	13	11/9/2024	20/9/2024	Tidak Terlambat
6	13/9/2024	14/9/2024	Tidak Terlambat	14	15/9/2024	20/9/2024	Tidak Terlambat
7	12/9/2024	14/9/2024	Tidak Terlambat	15	1/10/2024	1/10/2024	Tidak Terlambat
8	17/9/2024	17/9/2024	Tidak Terlambat	16	15/9/2024	3/10/2024	Tidak Terlambat

Terdapat suatu permasalahan ketika akan mengimplementasikan hasil penelitian ini. Permasalahan tersebut adalah jumlah stasiun kerja jahit yang tersedia lebih banyak dibandingkan dengan stasiun kerja berikutnya yaitu stasiun kerja *steam*. Hal ini menyebabkan pengalokasian lot pada stasiun kerja *steam* memerlukan suatu petunjuk bagi operator untuk pengambilan lot. Solusi yang disarankan adalah dengan membuat *kanban tags*. Contoh *kanban tags* yang dapat digunakan perusahaan dapat dilihat pada Gambar 5.



Gambar 5. *Kanban Tags*

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, algoritma CDS dan *Dannenbring* adalah solusi yang efektif dalam mengurangi *makespan* produksi dari yang semula sebesar 25 hari menjadi 20 dan 18 hari. Algoritma yang terpilih dengan nilai *makespan* terkecil adalah algoritma *Dannenbring* dengan modifikasi pembebanan mesin terkecil

dan *overlapping* yang menghasilkan *makespan* sebesar 8794,37 menit (sekitar 18 hari). Algoritma ini dapat mengurangi keterlambatan dari yang semula 9 *job* terlambat, menjadi tidak ada keterlambatan.

DAFTAR PUSTAKA

- Campbell, H. G., Dudek, R. A., & Smith, M. L. (1970). A Heuristic Algorithm for the n Job, m Machine Sequencing Problem. *Management Science*, 16(10), B-630-B-637. <https://doi.org/10.1287/mnsc.16.10.B630>
- Dannenbring, D. G. (1977). An Evaluation of Flow Shop Sequencing Heuristics. *Management Science*, 23(11), 1174–1182. <https://doi.org/10.1287/mnsc.23.11.1174>
- Ginting, R. (2009). *Penjadwalan Mesin* (Edisi Pertama). Graha Ilmu.
- Mashuri, C., Mujiyanto, A., Sucipto, H., Arsam, R., & Permadi, G. (2019). Production Time Optimization using Campbell Dudek Smith (CDS) Algorithm for Production Scheduling. *E3S Web of Conferences*, 125. <https://doi.org/10.1051/e3sconf/201>
- Masruroh, N. (2008). ANALISA PENJADWALAN PRODUKSI DENGAN MENGGUNAKAN METODA CAMPBELL DUDECK SMITH, PALMER, DAN DANNENBRING DI PT LOKA REFRAKTORIS SURABAYA. *Jurnal Tekmapro*, 3(2), 158–171.
- Pinedo, M. (2016). *Scheduling: Theory, Algorithms, and Systems* (5th ed.). Springer International Publishing. <https://doi.org/10.1007/978-3-319-26580-3>
- Prasetyo, H., & Heryati, F. (2024). The Effectiveness of Genetic Algorithm, And the CDS Method In Solving Flowshop Scheduling Problems. *E3S Web of Conferences*, 484, 01008. <https://doi.org/10.1051/e3sconf/202448401008>