

PENJADWALAN JOB PADA MESIN INJECTION MOLDING UNTUK MEMINIMASI MAKESPAN DAN TARDY JOB DENGAN MENGGUNAKAN ALGORITMA GREEDY

JOB SCHEDULING ON INJECTION MOLDING MACHINE TO MINIMIZE MAKESPAN AND TARDY JOB USING GREEDY ALGORITHM

Lintong Tarnama Lumbantoruan, Pratya Poeri Suryadhini² and Ayudita Oktafiani³

^{1,2,3}Prodi S1 Teknik Industri, Fakultas Rekayasa Industri, Universitas Telkom

¹lintonglumbantoruan@student.telkomuniversity.ac.id, ²pratya@telkomuniversity.ac.id,
³ayuditaoktafiani@telkomuniversity.ac.id

Abstrak

CV Gradient merupakan sebuah perusahaan manufaktur yang memproduksi berbagai jenis produk berbahan dasar plastik. Jumlah pesanan yang diterima dari customer rata-rata mencapai 5.000 – 7.000 *pieces* per hari. Dalam mendukung proses produksi perusahaan ini mengakomodasikan empat buah mesin *Injection Molding*. Keempat mesin *Injection Molding* tersebut diurutkan secara paralel dan memiliki fungsi yang sama sehingga dapat dikategorikan sebagai sistem *job scheduling identical parallel machine*. Berdasarkan hasil observasi dan wawancara ternyata CV Gradient sering mengalami keterlambatan produksi sehingga tidak mampu memenuhi kebutuhan dari customer. Penyebab keterlambatan tersebut ternyata akibat dari pengaturan penjadwalan produksi CV Gradient yang belum optimal. Metode penjadwalan yang diterapkan saat ini adalah *First Come First Serve* (FCFS) artinya pesanan konsumen yang pertama kali datang yang pertama kali dilayani. Kelemahan dari penerapan metode *First Come First Serve* tidak mempertimbangkan waktu penyelesaian pekerjaan maksimum (*Makespan*) dan *waiting time* rata-rata. Berdasarkan permasalahan tersebut dapat dilakukan perancangan usulan penjadwalan produksi dengan menggunakan Algoritma *Greedy* agar penjadwalan keempat mesin tersebut menjadi seimbang. Tujuan penelitian ini adalah merancang penjadwalan produksi dari keempat mesin *Injection Molding* untuk meminimasi *Makespan* dan *Tardy job*. Berdasarkan hasil penelitian, penjadwalan Algoritma *Greedy* dapat menghasilkan solusi optimal yaitu dapat meminimasi *Makespan* dan *Tardy job* dibandingkan dengan penjadwalan metode *First come first serve* (FCFS).

Kata Kunci: *Identical Parallel Machine, First Come First Serve (FCFS), Greedy Algorithm, Makespan, Tardy Job*

Abstract

CV Gradient is a manufacturing company that manufactures various types of plastic-based products. The number of orders received from customers reaches an average of 5,000 - 7,000 *pieces* per day. In supporting the production process, the company accommodates four *Injection Molding* machines. The four *Injection Molding* machines are sorted in parallel and have the same function so that they can be labeled as a *job system scheduling identical parallel machine*. Based on observations and interviews, it turns out that CV Gradient often experiences production delays that are unable to meet the needs of the customer. The cause of the delay was apparently due to the optimal production scheduling arrangement of the CV Gradient. The scheduling method applied today is *First Come First Serve* (FCFS), which means that first-time customer orders are served. The weakness of applying the *First Come First Serve* method does not take into account the maximum completion time (*Makespan*) and average waiting time. Based on these problems, it can be done by designing a production scheduling proposal using the *Greedy Algorithm* so that the scheduling of the four machines is balanced. The purpose of this study is to design production scheduling of the four *Injection Molding* machines to minimize *Makespan* and *Tardy Jobs*. Based on the results of the study, scheduling *Greedy Algorithm* can produce optimal solutions that can minimize the *Makespan* and *Tardy Jobs* compared to scheduling the *First come first serve* (FCFS) method.

Keywords: *Identical Parallel Machine, First Come First Serve (FCFS), Greedy Algorithm, Makespan, Tardy Job*

1. Pendahuluan

1.1 Latar Belakang

CV Gradient merupakan sebuah perusahaan manufaktur yang memproduksi berbagai jenis produk berbahan dasar plastik seperti *spring guide* untuk kendaraan bermotor, *cones* sebagai tempat gulungan benang dan lain-lain. CV Gradient juga bekerjasama dengan beberapa perusahaan seperti PT. Showa Mfg, PT Yamaha, Astra Group, PT Chandra Asri, PT Indah Jaya dan lain-lain. Jumlah pesanan yang

diterima dari customer rata-rata mencapai 5.000 – 7.000 *pieces* per hari. Sistem produksi yang diterapkan oleh CV Gradient adalah *make to order* (MTO). *Make to order* yaitu mengadakan produksi ketika ada pesanan dari konsumen sesuai dengan spesifikasi yang diinginkan (Baker, 1974). Dalam mendukung proses produksi perusahaan ini mengakomodasikan empat buah mesin *Injection Molding*. Keempat mesin *Injection Molding* tersebut diurutkan secara paralel dan memiliki fungsi yang sama sehingga dapat dikategorikan sebagai sistem *job scheduling identical parallel machine*. *Penjadwalan identical parallel machine* merupakan mesin yang mempunyai *speed factor* yang sama dan semua mesin dapat memproses semua *job* yang dipesan oleh customer (German, et al., 2016). Berdasarkan hasil observasi dan wawancara ternyata CV Gradient sering mengalami keterlambatan produksi sehingga tidak mampu memenuhi kebutuhan dari customer. Berikut ini merupakan salah satu data keterlambatan yang terjadi pada PT Showa Mfg tanggal 17 Desember 2018 – 21 Desember 2018:

Tabel 1 Data keterlambatan perusahaan PT Showa Mfg Tanggal 17 Desember- 21 Desember 2018

Data Pesanan Produk 17 Desember 2018 - 21 Desember 2018										
Job	Part Name	Part Code	Cycle Time (Second)	Quantity (Unit)	Processing Time (Hour)	Ordering Day	Ordering Time	Due Date (Hour)	Actual Created (Unit)	Not Created (Unit)
1	Under Case	HKVB2-425	21	2000	11,67	Monday, 17/12/18	7:30:00	38	2000	-
2	Spring Guide	HK592-400	24	2100	14,00	Monday, 17/12/18	7:30:00	38	2100	-
3	Spring Guide	HK932-400	24	2000	13,33	Monday, 17/12/18	7:30:00	36	2000	-
4	Under Case	HKZK L2-425	20	2500	13,89	Monday, 17/12/18	7:30:00	39	2500	-
5	Spring Guide	HKWB2-400	20	2500	13,89	Tuesday, 18/12/18	7:30:00	39	2500	-
6	Under Case	HK932-425	25	1500	10,42	Tuesday, 18/12/18	7:30:00	38	1500	-
7	Spring Guide	HK152-400	21	2000	11,67	Tuesday, 18/12/18	7:30:00	38	2000	-
8	Spring Guide	HK182-400	20	2100	11,67	Tuesday, 18/12/18	7:30:00	39	2100	-
9	Spring Guide	HKWN2-400	24	2000	13,33	Wednesday, 19/12/18	7:30:00	36	1837	163
10	Spring Guide	S4272-400	23	1500	9,58	Thursday, 20/12/18	7:30:00	38	1500	-
11	Spring Guide	S4292-400	23	1500	9,58	Thursday, 20/12/18	7:30:00	38	1500	-
12	Spring Guide	S4402-400	22	2500	15,28	Thursday, 20/12/18	7:30:00	40	2345	155
13	Spring Guide	S2463-400	22	2500	15,28	Friday, 20/12/18	7:30:00	40	795	794

Berdasarkan Tabel 1.1 terdapat *job* yang terlambat diantaranya *job* 9 (HKWN2-400) mengalami keterlambatan sebanyak 163 *pieces*, *job* 12 (S4402-400) mengalami keterlambatan sebanyak 155 *pieces* dan *job* 13 (S2463-400) mengalami keterlambatan sebanyak 794 *pieces*. Setelah dilakukan observasi ternyata penyebab keterlambatan tersebut akibat dari penjadwalan produksi CV Gradient yang belum optimal. Metode penjadwalan yang diterapkan oleh CV Gradient adalah *First Come First Serve* (FCFS). Artinya, pesanan konsumen yang pertama kali diterima yang akan pertama kali diproses. Kelemahan dari penerapan metode *First Come First Serve* (FSFS) adalah tidak mempertimbangkan waktu penyelesaian pekerjaan maksimum (*Makespan*) dan rata-rata waktu tunggu (*average waiting time*). Setiap *job* memiliki waktu proses yang berbeda-beda, ada *job* yang memiliki waktu proses yang lama maupun sebaliknya. Ketika pesanan tiba biasanya keputusan yang diambil oleh manajer produksi CV Gradient adalah menjadwalkan pekerjaan (*job*) ke dalam mesin yang sudah menyelesaikan pekerjaan tanpa mempertimbangkan waktu penyelesaian maksimum (*Makespan*) dan rata-rata waktu tunggu (*average waiting time*). Dari kedua masalah tersebut diindikasikan bahwa hal ini penyebab dari keterlambatan yang terjadi pada perusahaan CV Gradient. Berdasarkan permasalahan tersebut dapat dilakukan perancangan usulan penjadwalan produksi pada keempat mesin *injection molding* agar penjadwalan keempat mesin tersebut menjadi seimbang dengan kriteria meminimasi *Makespan* dan *Tardy job*.

2. Dasar Teori dan Metodologi

2.1 Istilah dalam Penjadwalan

1. Processing Time (t_i)

Processing Time adalah waktu yang dibutuhkan untuk menyelesaikan suatu tugas. Jika dalam penjadwalan memberikan waktu *set-up* maka total waktu *processing time* di jumlahkan dengan waktu *set-up* nya. *Processing time* dinyatakan dengan t_i .

2. Due-date (d_i)

Due-date adalah batas waktu maksimal yang dapat diterima untuk menyelesaikan suatu pekerjaan. *Due date* dinyatakan dalam d_i .

3. *Slack time* (SL_i)

Slack time adalah waktu yang dihasilkan dari pengurangan waktu *processing time* dengan *due-date*.

$$SL_i = d_i - t_i \dots \dots \dots (1)$$

4. *Flow time* (F_i)

Flow time adalah rentang waktu antara satu titik di mana tugas tersedia untuk diproses dengan suatu titik ketika tugas tersebut selesai. Jadi, *flowtime* sama dengan *processing time* dijumlahkan dengan waktu ketika tugas menunggu sebelum diproses. *Flow time* dinyatakan dalam F_i .

5. *Completion time* (C_i)

Waktu dari mulai bekerja menyelesaikan tugas pertama ($t = 0$) sampai dengan tugas ke- n selesai. *Completion time* dinyatakan dalam C_i .

6. Keterlambatan (*Lateness*), L_i

Lateness adalah selisih antara *completion time* (C_i) dengan *due-date* nya (d_i).

Apabila suatu pekerjaan diselesaikan setelah *due-date* nya maka *lateness* bernilai positif atau disebut dengan *tardiness*.

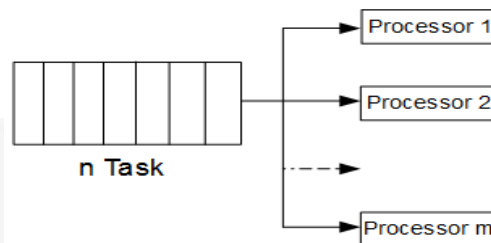
$$L_i = C_i - D_i > 0 \text{ (positif)} \dots \dots \dots (2)$$

Sedangkan apabila suatu pekerjaan diselesaikan sebelum *due-date* nya maka *lateness* bernilai negatif atau disebut dengan *earliness*.

$$L_i = C_i - D_i < 0 \text{ (negatif)} \dots \dots \dots (3)$$

2.2 Pengurutan n Pekerjaan Terhadap m Mesin Paralel

Pada n buah pekerjaan dapat beroperasi pada m buah mesin / processor.



Gambar 1 Penjadwalan Paralel

Pada gambar 1 masing-masing *job* hanya dikerjakan oleh 1 mesin.

2.3 Penjadwalan dengan Algoritma Greedy

Salah satu metode heuristik yang dapat menyelesaikan masalah optimasi adalah Algoritma *Greedy*. Persoalan optimasi (*optimization problems*) dibagi menjadi dua yaitu persoalan maksimasi (*maximization*) dan persoalan minimasi (*minimization*). Prinsip utama Algoritma *Greedy* adalah “*take what you can get now*”. Maksud dari prinsip tersebut adalah pada setiap langkah dalam Algoritma *Greedy*, diambil keputusan yang optimal untuk langkah tersebut tanpa memperhatikan konsekuensi pada langkah selanjutnya (Ahmad Juniar, 2015). Nilai optimum sementara ini dikenal dengan istilah *local optimum*. Terdapat banyak pilihan yang dapat dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya. Semua himpunan optimal local (*local-optimum*) akan mengarah kepada solusi optimal global (*global-optimum*).

Algoritma *Greedy* disusun oleh elemen-elemen sebagai berikut (Ahmad Juniar, 2015):

1. Himpunan Kandidat
Himpunan kandidat adalah himpunan yang berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi
Himpunan solusi adalah himpunan yang berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi seleksi (*selection function*)
Memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah terpilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi kelayakan (*feasible*)
Memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar

kendala (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak tidak dipertimbangkan lagi.

5. Fungsi obyektif

Fungsi obyektif adalah fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjang lintasan, waktu penyelesaian, keuntungan, dan lain-lain).

Berikut ini merupakan salah satu contoh penerapan Algoritma *Greedy* dalam kasus penukaran uang kertas:

Diberikan uang kertas senilai 32 sedangkan koin-koin yang tersedia adalah bernilai 1,5,10 dan 25. Berapa jumlah koin minimal untuk dapat menukar uang kertas senilai 32. Maka uang kertas senilai 32 dapat ditukar dengan cara berikut:

Langkah 1: pilih 1 buah koin 25 (Total =25)

Langkah 2: Pilih 1 buah koin 5 (Total= 25+5 = 30)

Langkah 3: Pilih 2 buah koin 1 (Total = 25+5+1+1 = 32)

Solusi: jumlah koin minimum = 4 (solusi optimal).

Dari contoh kasus diatas terlihat bahwa pada setiap langkah kita memperoleh *local-optimum* yaitu langkah ke-1 dan langkah ke-2, dan pada akhir algoritma kita memperoleh *global-optimum* pada langkah ke-3 (tiga).

Elemen Algoritma *Greedy* pada masalah penukaran uang tersebut adalah sebagai berikut:

1. Himpunan kandidat
Himpunan koin yang merepresentasikan nilai 1,5,10,25, paling sedikit mengandung satu koin untuk setiap nilai.
2. Himpunan solusi
Total nilai koin yang dipilih tepat sama jumlahnya dengan nilai uang yang ditukarkan
3. Fungsi seleksi
Memilih koin yang bernilai tertinggi dari himpunan kandidat yang tersisa.
4. Fungsi layak feasible)
Memeriksa apakah nilai total dari himpunan koin yang dipilih tidak melebihi jumlah uang yang harus ditukar.
5. Fungsi obyektif:
Jumlah koin yang digunakan minimum.

2.4 Langkah-langkah Algoritma *Greedy* pada penjadwalan *identical parallel machine*

Penjadwalan *identical parallel machine* menggunakan Algoritma *Greedy* memiliki langkah-langkah sebagai berikut:

Langkah 1:

Menentukan Himpunan kandidat. Himpunan kandidat adalah elemen-elemen pembentuk solusi. Pengurutan himpunan kandidat ditentukan berdasarkan waktu siklus dari setiap *job*.

$$CT_1 \geq CT_2 \geq \dots \geq CT_n \dots \dots \dots (4)$$

$CT = \text{Cycle Time}$

Langkah 2:

Menentukan himpunan kandidat (C). Himpunan kandidat adalah himpunan yang berisi elemen-elemen pembentuk solusi, dalam kasus penjadwalan *identical parallel machine* himpunan kandidat merupakan himpunan dari seluruh *job* yang telah diurutkan berdasarkan cycle time secara descending (diurutkan secara menurun).

Langkah 3:

Fungsi seleksi (*Selection function*). Fungsi seleksi dilakukan untuk memilih kandidat yang paling memungkinkan untuk mencapai solusi optimal. Berikut langkah-langkah yang dilakukan untuk memilih himpunan kandidat:

- Melakukan proses pengecekan apakah himpunan kandidat masih berisi atau sudah habis. Jika himpunan kandidat sudah habis, maka dilanjutkan langkah ke-7.
- Jika himpunan kandidat masih ada, maka ambil himpunan kandidat pada posisi pertama (C1), setelah mengambil himpunan kandidat selanjutnya akan diproses pada fungsi kelayakan.

Langkah 4:

Fungsi kelayakan. Fungsi kelayakan dilakukan untuk mengecek apabila kandidat yang dipilih sudah tepat, maka kandidat akan dimasukkan kedalam himpunan solusi.

Langkah 5:

Menghitung Kumulatif beban Mesin. Kumulatif beban Mesin dapat dicari dengan menggunakan rumus:

$$\text{Kumulatif Beban Mesin} = \sum_{j=1}^n Q_j \times CT_j \dots \dots \dots (5)$$

$Q_j = \text{Quantity of job (j) = (j = 1, \dots n)}$

$CT_j = \text{Cycle time of job (j) = (j = 1, \dots n)}$

Langkah 6:

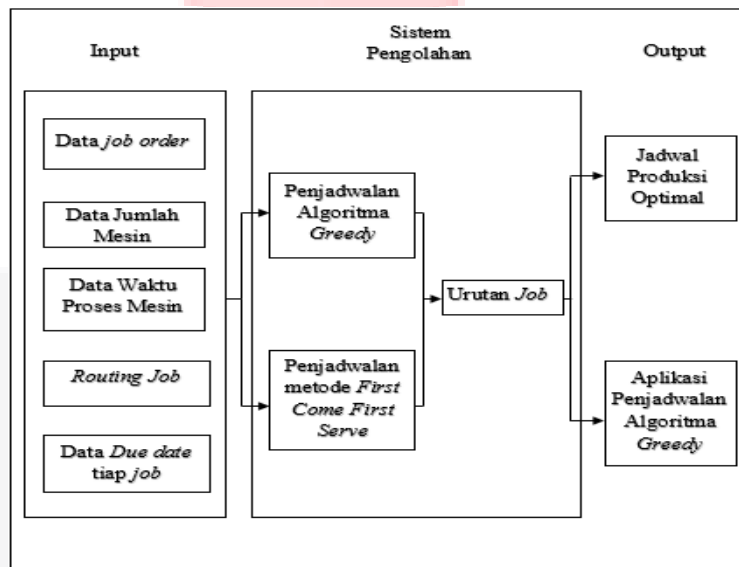
Ulangi iterasi pada langkah ke-3 (tiga) dan ke-4 (empat) hingga seluruh *job* pada himpunan kandidat telah habis.

Langkah 7:

Iterasi selesai dan solusi optimal sudah didapatkan.

Dengan kata lain, Algoritma Greedy melibatkan pencarian sebuah himpunan bagian (S) dari himpunan kandidat (C) yang dalam hal ini (S) harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan (S) dioptimisasi oleh fungsi obyektif.

2.5 Metodologi Penelitian



Gambar 2 Metodologi Penelitian

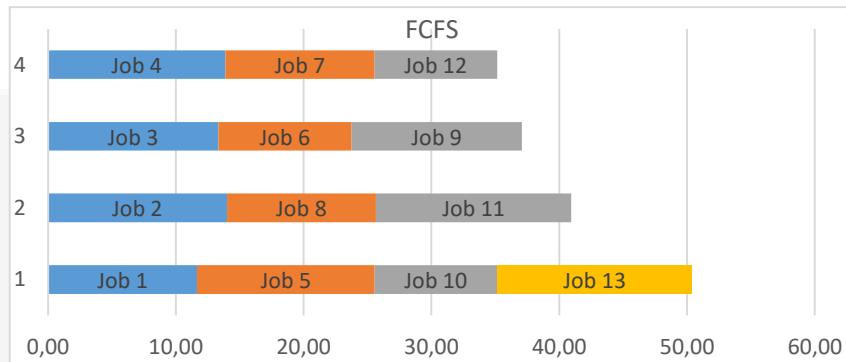
3. Pembahasan

3.1 Penjadwalan Metode First Come First Serve (Kondisi Aktual)

Sistem penjadwalan yang diterapkan oleh CV Gradient adalah *First Come First Serve* (FCFS). Artinya, pesanan yang diterima pertama kali datang akan diproses terlebih dahulu. Biasanya job HKVB2-425, HK592-400, HK932-400, HKZKL-425, HKWB2-400, HK932-425, HK182-400, HK152-400, HKWN2-400 akan tiba terlebih dahulu dari pada job S4292-400, S4272-400, S4402-400, S2463-400. Penerapan penjadwalan dengan menggunakan metode *first come first serve* (FCFS) didapatkan *Makespan* dibantu dengan menggunakan *Gantt Chart* sebesar 50,42 jam atau 50 jam 25 menit, *tardinees* 12,44 jam atau 12 jam 26 menit 40 detik, *Total quantity job* yang terlambat sebanyak 2023 unit dengan jumlah job yang terlambat sebanyak 3 buah *job* yaitu *job* 9 (HKWN2-400,) *job* 11 (S4272-400), dan *job* 12 (S4402-400). Berikut Tabel 2 Hasil penjadwalan kondisi awal perusahaan atau metode *first come first serve* (FCFS) dan Gambar 3 *Gantt Chart* dari metode *First Come First Serve*:

Tabel 2 Penjadwalan *First Come First Serve* (Kondisi Aktual)

Mesin	Job	Part Name	Cycle Time (Second)	Quantity (Unit)	Processing Time (Second)	Processing Time (Minute)	Processing Time (Hour)	Completion Time	Due Date (Hour)	Lateness (Hour)	Total Quantity Job yang terlambat
1	1	HKVB2-425	21	2000	42000	700	11,67	11,67	38	-26,33	-
	5	HKWB2-400	20	2500	50000	833,3333333	13,89	25,56	39	-13,44	-
	10	S4272-400	23	1500	34500	575	9,58	35,14	38	-2,86	-
	13	S2463-400	22	2500	55000	916,6666667	15,28	50,42	40	10,42	1705
2	2	HK592-400	24	2100	50400	840	14	14	38	-24	-
	8	HK182-400	20	2100	42000	700	11,67	25,67	39	-13,33	-
	12	S4402-400	22	2500	55000	916,6666667	15,28	40,94	40	0,94	155
3	3	HK932-400	24	2000	48000	800	13,33	13,33	36	-22,67	-
	6	HK932-425	25	1500	37500	625	10,42	23,75	38	-14,25	-
	9	HKWN2-400	24	2000	48000	800	13,33	37,08	36	1,08	163
4	4	HKZKL2-425	20	2500	50000	833,3333333	13,89	13,89	39	-25,11	-
	7	HK152-400	21	2000	42000	700	11,67	25,56	38	-12,44	-
	11	S4292-400	23	1500	34500	575	9,58	35,14	38	-2,86	-
Maksimum Makespan (Hour)		50,42	Makespan = C max								
Total Tardy Job (Hour)		12,44	Tardy Job = {0, Li} : hanya melihat L yang > 0, dengan 1 ≤ i ≤ N								
Jumlah Job yang terlambat		3	Li = ci - di > 0 (positif)								
Total Quantity Job yang terlambat		2023	Li = (ci - di > 0) / (Cycle time)								



Gambar 3 Gantt Chart *First Come First Serve* (Kondisi Aktual)

3.2 Penjadwalan Algoritma Greedy

Langkah 1:

Lakukan Pengurutan data berdasarkan *Cycle time* secara *descending* (diurutkan secara menurun). Hasil dari pengurutan data dapat dilihat pada Tabel 3 dibawah ini:

Tabel 3 Hasil Pengurutan data *Cycle Time* secara *Descending*

Data Pesanan Produk 17 Desember 2018 - 24 Desember 2018									
Job	Part Name	Cycle Time	Quantity	Processing Time (Second)	Minute	Hour	Ordering Time	Due Date	Actual Created
6	HK932-425	25	1500	37500	625	10,42	17/12/18	38	1500
2	HK592-400	24	2100	50400	840	14	17/12/18	38	2100
3	HK932-400	24	2000	48000	800	13,33	17/12/18	36	2000
9	HKWN2-400	24	2000	48000	800	13,33	17/12/18	36	2000
10	S4272-400	23	1500	34500	575	9,58	17/12/18	38	1500
11	S4292-400	23	1500	34500	575	9,58	17/12/18	38	1500
12	S4402-400	22	2500	55000	916,67	15,28	17/12/18	40	2500
13	S2463-400	22	2500	55000	916,67	15,28	17/12/18	40	2500
1	HKVB2-425	21	2000	42000	700	11,67	17/12/18	38	2000
7	HK152-400	21	2000	42000	700	11,67	17/12/18	38	2000
4	HKZKL2-425	20	2500	50000	833,33	13,89	17/12/18	39	2500
5	HKWB2-400	20	2500	50000	833,33	13,89	17/12/18	39	2500
8	HK182-400	20	2100	42000	700	11,67	17/12/18	39	732

Langkah 2:

Dengan menggunakan data diatas maka dapat ditentukan himpunan kandidat (C). Himpunan kandidat adalah himpunan yang berisi elemen-elemen pembentuk solusi, dalam hal ini kandidat merupakan himpunan dari seluruh job yang telah diurutkan berdasarkan cycle time secara descending (diurutkan secara menurun). Berikut merupakan himpunan kandidat berdasarkan data pada Tabel 2:

$$C = \{J6[1500], J2[2100], J3[2000], J9[2000], J10[1500], J11[1500], J12[2500], J13[2500], J1[2000], J7[2000], J4[2500], J5[2500], J8[2100]\}$$

Langkah 3:

Fungsi seleksi (Selection function). Memilih kandidat yang paling memungkinkan untuk mencapai solusi optimal. Berikut langkah-langkah yang dilakukan untuk memilih himpunan kandidat:

- Melakukan proses pengecekan apakah himpunan kandidat masih berisi atau sudah habis. Jika himpunan kandidat sudah habis, maka dilanjutkan langkah ke-7.
- Jika himpunan kandidat masih ada, maka ambil himpunan kandidat pada posisi pertama (C1), setelah mengambil himpunan kandidat selanjutnya akan diproses pada fungsi kelayakan. Berikut merupakan himpunan kandidat pada posisi pertama (C1):

Tabel 4 Kandidat Posisi Pertama (C1)

ITERASI KE-0			
Mesin 1	Mesin 2	Mesin 3	Mesin 4
J6[0]	J6[0]	J6[0]	J6[0]
J2[0]	J2[0]	J2[0]	J2[0]
J3[0]	J3[0]	J3[0]	J3[0]
J9[0]	J9[0]	J9[0]	J9[0]
J10[0]	J10[0]	J10[0]	J10[0]
J11[0]	J11[0]	J11[0]	J11[0]
J12[0]	J12[0]	J12[0]	J12[0]
J13[0]	J13[0]	J13[0]	J13[0]
J1[0]	J1[0]	J1[0]	J1[0]
J7[0]	J7[0]	J7[0]	J7[0]
J4[0]	J4[0]	J4[0]	J4[0]
J5[0]	J5[0]	J5[0]	J5[0]
J8[0]	J8[0]	J8[0]	J8[0]
Kumulatif Beban Work Station 1,2,3,4 (Second)			
157725	157725	157725	157725
C =	{J6[1500], J2[2100], J3[2000], J9[2000], J10[1500], J11[1500], J12[2500], J13[2500], J1[2000], J7[2000], J4[2500], J5[2500], J8[2100]}		
S =	{J6[0], J2[0], J3[0], J9[0], J10[0], J11[0], J12[0], J13[0], J1[0], J7[0], J4[0], J5[0], J8[0]}		

Berdasarkan Tabell 4 Himpunan kandidat C1 pada iterasi ke-0 (nol) yaitu J6[1500].

Langkah 4:

Fungsi Kelayakan. Pada langkah ini Job pada himpunan kandidat posisi pertama (C1) akan ditentukan layak untuk didistribusikan ke salah satu mesin yang ada. Sebuah job dikatakan layak untuk di distribusikan ke suatu mesin, Mi, jika mesin tersebut memiliki total completion time terkecil atau minimum. Setelah job pada himpunan kandidat C1 dimasukan ke dalam mesin, kandidat pada posisi pertama C1 akan dibuang dari himpunan kandidat dan dimasukan ke dalam himpunan solusi (S) sehingga kandidat pada posisi kedua (C2) menjadi posisi pertama C1. Berikut ini merupakan Tabel 5 setelah semua job pada himpunan kandidat C1 dimasukan ke dalam mesin.

Tabel 5 Hasil himpunan solusi sementara (Iterasi ke-1500)

ITERASI KE-1500			
Mesin 1	Mesin 2	Mesin 3	Mesin 4
J6[375]	J6[375]	J6[375]	J6[375]
J2[0]	J2[0]	J2[0]	J2[0]
J3[0]	J3[0]	J3[0]	J3[0]
J9[0]	J9[0]	J9[0]	J9[0]
J10[0]	J10[0]	J10[0]	J10[0]
J11[0]	J11[0]	J11[0]	J11[0]
J12[0]	J12[0]	J12[0]	J12[0]
J13[0]	J13[0]	J13[0]	J13[0]
J1[0]	J1[0]	J1[0]	J1[0]
J7[0]	J7[0]	J7[0]	J7[0]
J4[0]	J4[0]	J4[0]	J4[0]
J5[0]	J5[0]	J5[0]	J5[0]
J8[0]	J8[0]	J8[0]	J8[0]
Kumulatif Beban Mesin 1,2,3,4 (Second)			
9375	9375	9375	9375
C =	{J6[0], J2[2100], J3[2000], J9[2000], J10[1500], J11[1500], J12[2500], J13[2500], J1[2000], J7[2000], J4[2500], J5[2500], J8[2100]}		
S =	{J6[1500], J2[0], J3[0], J9[0], J10[0], J11[0], J12[0], J13[0], J1[0], J7[0], J4[0], J5[0], J8[0]}		

Berdasarkan Tabel 5 Setelah job yang ada pada himpunan kandidat C1 dimasukan kedalam mesin, kandidat pada posisi pertama C1 yaitu J6[1500] akan dibuang dari himpunan kandidat dan dimasukan kedalam himpunan solusi (S) sehingga kandidat pada posisi kedua C2 yaitu J2[2100] menjadi posisi pertama (C1).

Langkah 5:

Menghitung Kumulatif beban pada setiap mesin:

$$\begin{aligned} \text{Kumulatif Beban Mesin 1} &= \sum_{j=1}^n Q_j \times CT_j \\ &= (375 \text{ unit} \times 25 \text{ detik}) \\ &= 9375 \text{ detik} \end{aligned}$$

$$\begin{aligned} \text{Kumulatif Beban Mesin 2} &= \sum_{j=1}^n Q_j \times CT_j \\ &= (375 \text{ unit} \times 25 \text{ detik}) \\ &= 9375 \text{ detik} \end{aligned}$$

$$\begin{aligned} \text{Kumulatif Beban Mesin 3} &= \sum_{j=1}^n Q_j \times CT_j \\ &= (375 \text{ unit} \times 25 \text{ detik}) \\ &= 9375 \text{ detik} \end{aligned}$$

$$\begin{aligned} \text{Kumulatif Beban Mesin 4} &= \sum_{j=1}^n Q_j \times CT_j \\ &= (375 \text{ unit} \times 25 \text{ detik}) \\ &= 9375 \text{ detik} \end{aligned}$$

Langkah 6:

Ulangi iterasi pada langkah ke-3 (tiga) dan ke-4 (empat) hingga seluruh job pada himpunan kandidat telah habis.

Langkah 7:

Iterasi selesai dan solusi optimal sudah didapatkan. Berikut merupakan keseluruhan iterasi dalam mencari solusi optimal dengan menggunakan Algoritma Greedy.

3.3 Hasil Penjadwalan menggunakan Algoritma Greedy

Hasil penjadwalan menggunakan Algoritma Greedy dapat dilihat pada Tabel 6 berikut ini:

Tabel 6 Hasil Himpunan Solusi Akhir Algoritma Greedy (n = 26.700)

ITERASI KE-26700			
Mesin 1	Mesin 2	Mesin 3	Mesin 4
J6[375]	J6[375]	J6[375]	J6[375]
J2[525]	J2[525]	J2[525]	J2[525]
J3[500]	J3[500]	J3[500]	J3[500]
J9[500]	J9[500]	J9[500]	J9[500]
J10[375]	J10[375]	J10[375]	J10[375]
J11[375]	J11[375]	J11[375]	J11[375]
J12[625]	J12[625]	J12[625]	J12[625]
J13[625]	J13[625]	J13[625]	J13[625]
J1[500]	J1[500]	J1[500]	J1[500]
J7[500]	J7[500]	J7[500]	J7[500]
J4[625]	J4[625]	J4[625]	J4[625]
J5[625]	J5[625]	J5[625]	J5[625]
J8[525]	J8[525]	J8[525]	J8[525]
Kumulatif Beban Work Station 1,2,3 (Second)			
147225	147225	147225	147225
C =	{J6[0], J2[0], J3[0], J9[0], J10[0], J11[0], J12[0], J13[0], J1[0], J7[0], J4[0], J5[0], J8[0]}		
S =	{J6[1500], J2[2100], J3[2000], J9[2000], J10[1500], J11[1500], J12[2500], J13[2500], J1[2000], J7[2000], J4[2500], J5[2500], J8[2100]}		

$$\begin{aligned}
 \text{Kumulatif Beban Mesin 1} &= \sum_{j=1}^n Q_j \times CT_j \\
 &= (375 \times 25) + \dots + (522 \times 20) \\
 &= 147.225 \text{ Second}
 \end{aligned}$$

$$\begin{aligned}
 \text{Kumulatif Beban Mesin 2} &= \sum_{j=1}^n Q_j \times CT_j \\
 &= (375 \times 25) + \dots + (522 \times 20) \\
 &= 147.225 \text{ Second}
 \end{aligned}$$

$$\begin{aligned}
 \text{Kumulatif Beban Mesin 3} &= \sum_{j=1}^n Q_j \times CT_j \\
 &= (375 \times 25) + \dots + (522 \times 20) \\
 &= 147.225 \text{ Second}
 \end{aligned}$$

$$\begin{aligned}
 \text{Kumulatif Beban Mesin 4} &= \sum_{j=1}^n Q_j \times CT_j \\
 &= (375 \times 25) + \dots + (522 \times 20) \\
 &= 147.225 \text{ Second}
 \end{aligned}$$

Berdasarkan Tabel 6 jumlah iterasi yang dilakukan pada proses penjadwalan menggunakan Algoritma Greedy sebanyak 26.700 iterasi yang juga merupakan iterasi terakhir yang menghasilkan solusi optimal pada penjadwalan menggunakan Algoritma Greedy. Algoritma Greedy untuk kasus ini mempunyai kompleksitas iterasi $O(n)$, n = Total Quantity. Penjadwalan Algoritma Greedy pada kasus ini menghasilkan Makespan sebesar 147.225 detik atau 40,9 jam. Sedangkan untuk hasil dari Tardiness, number of tardy job, not created dan gambar Gantt Chart dapat dilihat pada Tabel 7 dan Gambar 4 dibawah ini:

Tabel 7 Hasil Penjadwalan Algoritma Greedy

Mesin	Job	Processing Time (Second)	Completion Time (Second)	Due date (Second)	Lateness (Second)	Lateness (Hour)	Quantity of Tardy Job (Unit)
1	J6[375]	9375	9375	136800	-127425	-35,39583333	-
	J2[525]	12600	21975	136800	-114825	-31,89583333	-
	J3[500]	12000	33975	129600	-95625	-26,5625	-
	J9[500]	12000	45975	129600	-83625	-23,22916667	-
	J10[375]	8625	54600	136800	-82200	-22,83333333	-
	J11[375]	8625	63225	136800	-73575	-20,4375	-
	J12[625]	13750	76975	144000	-67025	-18,61805556	-
	J13[625]	13750	90725	144000	-53275	-14,79861111	-
	J1[500]	10500	101225	136800	-35575	-9,881944444	-
	J7[500]	10500	111725	136800	-25075	-6,965277778	-
	J4[625]	12500	124225	140400	-16175	-4,493055556	-
	J5[625]	12500	136725	140400	-3675	-1,020833333	-
	J8[525]	10500	147225	140400	6825	1,895833333	342
Completion Time (Hour)	40,90	Completion Time = $\sum t$					
Tardiness (Hour)	1,90	Tardy Job = $\{0, L_i\}$: hanya melihat L yang > 0, dengan $1 \leq i \leq N$					
Number of Tardy Job	1	$L_i = c_i - d_i > 0$ (positif)					
Quantity of Tardy Job (Unit)	342	$L_i = (c_i - d_i > 0) / (\text{Cycle time})$					
Mesin	Job	Processing Time (Second)	Completion Time (Second)	Due date (Second)	Lateness (Second)	Lateness (Hour)	Quantity of Tardy Job (Unit)
2	J6[375]	9375	9375	136800	-127425	-35,39583333	-
	J2[525]	12600	21975	136800	-114825	-31,89583333	-
	J3[500]	12000	33975	129600	-95625	-26,5625	-
	J9[500]	12000	45975	129600	-83625	-23,22916667	-
	J10[375]	8625	54600	136800	-82200	-22,83333333	-
	J11[375]	8625	63225	136800	-73575	-20,4375	-
	J12[625]	13750	76975	144000	-67025	-18,61805556	-
	J13[625]	13750	90725	144000	-53275	-14,79861111	-
	J1[500]	10500	101225	136800	-35575	-9,881944444	-
	J7[500]	10500	111725	136800	-25075	-6,965277778	-
	J4[625]	12500	124225	140400	-16175	-4,493055556	-
	J5[625]	12500	136725	140400	-3675	-1,020833333	-
	J8[525]	10500	147225	140400	6825	1,895833333	342
Completion Time (Hour)	40,90	Completion Time = $\sum t$					
Tardiness (Hour)	1,90	Tardy Job = $\{0, L_i\}$: hanya melihat L yang > 0, dengan $1 \leq i \leq N$					
Number of Tardy Job	1	$L_i = c_i - d_i > 0$ (positif)					
Quantity of Tardy Job (Unit)	342	$L_i = (c_i - d_i > 0) / (\text{Cycle time})$					
Mesin	Job	Processing Time (Second)	Completion Time (Second)	Due date (Second)	Lateness (Second)	Lateness (Hour)	Quantity of Tardy Job (Unit)
3	J6[375]	9375	9375	136800	-127425	-35,39583333	-
	J2[525]	12600	21975	136800	-114825	-31,89583333	-
	J3[500]	12000	33975	129600	-95625	-26,5625	-
	J9[500]	12000	45975	129600	-83625	-23,22916667	-
	J10[375]	8625	54600	136800	-82200	-22,83333333	-
	J11[375]	8625	63225	136800	-73575	-20,4375	-
	J12[625]	13750	76975	144000	-67025	-18,61805556	-
	J13[625]	13750	90725	144000	-53275	-14,79861111	-
	J1[500]	10500	101225	136800	-35575	-9,881944444	-
	J7[500]	10500	111725	136800	-25075	-6,965277778	-
	J4[625]	12500	124225	140400	-16175	-4,493055556	-
	J5[625]	12500	136725	140400	-3675	-1,020833333	-
	J8[525]	10500	147225	140400	6825	1,895833333	342
Completion Time (Hour)	40,90	Completion Time = $\sum t$					
Tardiness (Hour)	1,90	Tardy Job = $\{0, L_i\}$: hanya melihat L yang > 0, dengan $1 \leq i \leq N$					
Number of Tardy Job	1	$L_i = c_i - d_i > 0$ (positif)					
Quantity of Tardy Job (Unit)	342	$L_i = (c_i - d_i > 0) / (\text{Cycle time})$					

Mesin	Job	Processing Time (Second)	Completion Time (Second)	Due date (Second)	Lateness (Second)	Lateness (Hour)	Quantity of Tardy Job (Unit)
4	J6[375]	9375	9375	136800	-127425	-35,39583333	-
	J2[525]	12600	21975	136800	-114825	-31,89583333	-
	J3[500]	12000	33975	129600	-95625	-26,5625	-
	J9[500]	12000	45975	129600	-83625	-23,22916667	-
	J10[375]	8625	54600	136800	-82200	-22,83333333	-
	J11[375]	8625	63225	136800	-73575	-20,4375	-
	J12[625]	13750	76975	144000	-67025	-18,61805556	-
	J13[625]	13750	90725	144000	-53275	-14,79861111	-
	J1[500]	10500	101225	136800	-35575	-9,881944444	-
	J7[500]	10500	111725	136800	-25075	-6,965277778	-
	J4[625]	12500	124225	140400	-16175	-4,493055556	-
	J5[625]	12500	136725	140400	-3675	-1,020833333	-
	J8[525]	10500	147225	140400	6825	1,895833333	342
Completion Time (Hour)	40,90	Completion Time = $\sum t$					
Tardiness (Hour)	1,90	Tardy Job = $\{0, Li\}$: hanya melihat L yang > 0, dengan $i \leq i \leq N$					
Number of Tardy Job	1	$Li = ci - di > 0$ (positif)					
Quantity of Tardy Job (Unit)	342	$Li = (ci - di > 0) / (\text{Cycle time})$					

3.4 Implementasi Penjadwalan Produksi dengan Bahasa Go programming language (Golang).

Penjadwalan *single-stage identical parallel machine* menggunakan Algoritma *Greedy* diimplementasikan dalam bahasa *Go Programming Language* (Golang). *Output* program dapat dilihat pada Gambar 5 dibawah ini:

```

Microsoft Windows [Version 10.0.17134.885]
(c) 2018 Microsoft Corporation. All rights reserved.

D:\TELKOM UNIVERSITY\SEMESTER 7\TUGAS AKHIR\Scheduling\Tugas Akhir\Seminar Proposal\Program\main.exe
Make Span: 40H 53M 45s
Tardy Job: 7H 36M 0s

***** Stations *****
Machine #1:
  Completion Time: 40H 53M 45s
  Jobs: HK932-425(375) HK592-400(525) HK932-400(500) HKWN2-400(500) S4272-400(375) S4292-400(375) S4402-400(625)
  S2463-400(625) HKVB2-425(500) HK152-400(500) HKZKL2-425(625) HKWB2-400(625) HK182-400(525)
Machine #2:
  Completion Time: 40H 53M 45s
  Jobs: HK932-425(375) HK592-400(525) HK932-400(500) HKWN2-400(500) S4272-400(375) S4292-400(375) S4402-400(625)
  S2463-400(625) HKVB2-425(500) HK152-400(500) HKZKL2-425(625) HKWB2-400(625) HK182-400(525)
Machine #3:
  Completion Time: 40H 53M 45s
  Jobs: HK932-425(375) HK592-400(525) HK932-400(500) HKWN2-400(500) S4272-400(375) S4292-400(375) S4402-400(625)
  S2463-400(625) HKVB2-425(500) HK152-400(500) HKZKL2-425(625) HKWB2-400(625) HK182-400(525)
Machine #4:
  Completion Time: 40H 53M 45s
  Jobs: HK932-425(375) HK592-400(525) HK932-400(500) HKWN2-400(500) S4272-400(375) S4292-400(375) S4402-400(625)
  S2463-400(625) HKVB2-425(500) HK152-400(500) HKZKL2-425(625) HKWB2-400(625) HK182-400(525)

***** Jobs *****
Job  PartName      Cycle Time      Quantity      Lateness      Actual Created
-----
1  HKVB2-425        21              2000          0              2000
2  HK592-400        24              2100          0              2100
3  HK932-400        24              2000          0              2000
4  HKZKL2-425      20              2500          0              2500
5  HKWB2-400        20              2500          0              2500
6  HK932-425        25              1500          0              1500
7  HK152-400        21              2000          0              2000
8  HK182-400        20              2100          1368           732
9  HKWN2-400        24              2000          0              2000
10 S4272-400        23              1500          0              1500
11 S4292-400        23              1500          0              1500
12 S4402-400        22              2500          0              2500
13 S2463-400        22              2500          0              2500

```

Gambar 5 Implementasi Penjadwalan Produksi dengan Bahasa Pemrograman Golang

4. Kesimpulan

Hasil perbandingan penjadwalan pada kondisi awal dengan kondisi usulan didapatkan kondisi awal menghasilkan Makespan sebesar 50,42 jam atau 50 jam 25 menit sedangkan penjadwalan usulan menghasilkan Makespan sebesar 47,78 jam atau 40 jam 53 menit 45 detik. Total Tardiness pada kondisi awal sebesar 12,24 jam atau 12 jam 26 menit 40 detik sedangkan total Tardiness pada kondisi usulan sebesar 7,58 jam atau 7 jam 35 menit. Number of tardy job pada kondisi aktual sebanyak 3 buah job yaitu job 9, job 12, dan job 13, sedangkan Number of tardy job pada kondisi usulan sebanyak 1 (satu) buah job saja yaitu job 8. Penjadwalan kondisi aktual terdapat 2023 unit yang belum dapat diselesaikan sedangkan penjadwalan kondisi usulan terdapat 1368 unit yang belum dapat diselesaikan.

5. Saran

Adapun saran untuk penelitian selanjutnya adalah:

1. Penelitian dapat dilakukan dengan membahas masalah penjadwalan untuk beberapa produk yang lain.
2. Penyelesaian persoalan penjadwalan produksi multi-stage baik berjenis flowshop ataupun jobshop dengan Algoritma Greedy.
3. Membandingkan pengoptimalan penjadwalan menggunakan metode Algoritma Greedy dan metode lainnya
4. Penelitian dapat memperhitungkan bila terjadi job sisipan di tengah-tengah proses produksi.

Daftar Pustaka

- [1] Baker, Kenneth R, 1984. Introduction to Sequencing and Scheduling, John Wiley & Sons, New York.
- [2] Ginting, Rosnani. 2009. Penjadwalan Mesin, Edisi Pertama. Yogyakarta: Graha Ilmu
- [3] Juniar, Ahmad. 2015. Penerapan Algoritma Greedy pada Penjadwalan Produksi *Single-Stage* dengan *Parallel Machine* di Industri Konveksi. Jurnal Sifo Mikroskil. 16(2): 175-164.
- [4] Gozali, William., dan Alham Fikri Aji. 2014. Pemrograman Kompetitif Dasar. Jakarta: Ikatan Alumni Tim Olimpiade Komputer Indonesia
- [5] Baroto, Teguh. 2020. Perencanaan dan Pengendalian Produksi. Jakarta: Ghalia Indonesia
- [6] Rasjidin, Roesfiansjah dan Iman Hidayat. 2006. Penjadwalan Produksi *Injection Molding* Pada PT. Duta Flow Plastic Machinery. Jurnal Inovasi. 5(2): 52-57
- [7] Yoson, Bagus R. 2014. Penjadwalan Produksi Dengan Menggunakan Metode FCFS, EDD, SPT dan LPT Untuk Meningkatkan Produktivitas Kerja. Jurnal Ilmiah PASTI. 6(1): 97-107.
- [8] Purnomo, A.C., Yuliana, M., Prasetyaningrum, I., 2014, Implementasi algoritma Greedy pada layanan call center taksi wisata berbasis web, EEPIS-Online Politeknik Elektronika Negeri, Surabaya: <http://www.pens.ac.id/post/20130813144934-1292>
- [9] Munir, R, 2009, Diklat Kuliah: Strategi Algoritma, Penerbit ITB, Bandung
- [10] Bedworth D.D. and Bailey J.E., Integrated Production Control System, 1987, John Wiley & Sons, New York.