



Optimasi Penjadwalan *Job Shop* dengan Metode Algoritma Greedy

Hery Irwan^{1,2}, MD. Nizam Abd Rahman¹, Zuhriah binti Ebrahim¹, Roberta Simarmata²

¹ Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka,
Hang Tuah Jaya Street, Durian Tunggal 76100, Malacca, Malaysia

²Department of Industrial Engineering, Faculty of Technical, Universitas Riau Kepulauan,
Batu Aji street no. 99 Batam City, Kepulauan Riau, 29422, Indonesia

Email: hery04@gmail.com

Abstraksi

Persoalan penjadwalan timbul apabila beberapa pekerjaan akan dikerjakan secara bersamaan, sedangkan sumber yang dimiliki terbatas. Input dari suatu penjadwalan mencakup jenis dan banyaknya part yang akan diproduksi, urutan proses dan ketergantungan antar operasi, waktu proses untuk masing-masing operasi, serta fasilitas yang dibutuhkan oleh setiap operasi. Sedangkan output dari penjadwalan meliputi *dispatch list*, yaitu daftar yang menyatakan urutan pemrosesan part serta waktu mulai dan selesai dari pemrosesan part. Algoritma Greedy merupakan salah satu metode yang dipergunakan untuk mengoptimalkan penjadwalan *single machine* dengan tujuan minimasi *make span* yang selanjutnya akan diuji pada penjadwalan *multi machine*. Dikakhir penelitian dilakukan pembuktian hasil optimasi yang didapat dengan melakukan penjadwalan dengan bantuan Gantt Chart. Hasil Gantt Chart diperoleh terjadi perbedaan perhitungan kumulatif waktu khususnya untuk mesin yang memiliki alternative, yaitu total waktu dengan algoritma greedy selama 80 jam akan tetapi setelah dilakukan penjadwalan dengan Gantt Chart didapati total waktu yang dihasilkan adalah 86 jam, dikarenakan terjadi perbedaan dari mesin yang memiliki alternative.

Kata Kunci : Penjadwalan, *dispatch list*, Algoritma Greedy

Abstraction

Scheduling problems arise when several jobs will be carried out simultaneously, while the resources that are owned are limited. The input of a scheduling includes the type and number of parts to be produced, the sequence of processes and dependencies between operations, the processing time for each operation and the facilities required by each operation. Meanwhile, the output of scheduling includes a dispatch list, which is a list that states the part processing sequence and the start and finish times of part processing. The Greedy Algorithm is one of the methods used to optimize single machine scheduling with the aim of minimizing the make span which will then be tested in multi machine scheduling. At that time, the research was conducted to prove the optimization results obtained by scheduling with the help of the Gantt Chart. The results of the Gantt Chart show that there is a difference in the cumulative calculation of time, especially for machines that have alternatives, namely the total time with the Greedy algorithm for 80 hours, but after scheduling with the Gantt Chart it is found that the total time generated is 86 hours, due to differences come from machines that have alternatives.

Keywords: Scheduling, dispatch list, Greedy Algorithm

1. Pendahuluan

Dalam dunia industri semua keinginan atau permintaan pelanggan semaksimal mungkin harus dipenuhi. Permintaan dan spesifikasi pelanggan dari setiap produk berubah sangat cepat sehingga sangat penting bagi industri manufaktur untuk dapat mengakomodasi segala bentuk perubahan secepat mungkin sebagai salah satu upaya bersaing di pasaran. Perubahan ini sering menyebabkan konflik pada sistem manufaktur karena seiring meningkatnya variasi produk berakibat menurunnya kecepatan produksi.

Ukuran keberhasilan dari suatu pelaksanaan aktivitas penjadwalan khususnya penjadwalan *job shop* adalah meminimasi kriteria-kriteria keberhasilan sebagai rata-rata waktu alir (*mean flow time*) akan mengurangi persediaan barang setengah jadi, makespan, yaitu total waktu proses yang dibutuhkan untuk menyelesaikan suatu kumpulan job. Dimaksudkan untuk meraih utilisasi yang tinggi dari peralatan dan sumber daya dengan cara menyelesaikan seluruh job secepatnya, rata-rata kelambatan (*mean tardiness*), jumlah *job* yang terlambat, akan meminimasi nilai dari maksimum ukuran kelambatan mesin yang mengkur dan jumlah persediaan.

2. Kajian Pustaka

Tujuan penjadwalan menurut Baker (1974) adalah meningkatkan produktifitas mesin, yaitu dengan mengurangi waktu mesin menganggur, mengurangi persediaan barang setengah jadi dengan jalan mengurangi jumlah rata-rata pekerjaan yang menunggu dalam antrian suatu mesin karena mesin

tersebut sibuk, mengurangi keterlambatan suatu pekerjaan. Setiap pekerjaan mempunyai batas waktu (*due date*) penyelesaian, jika pekerjaan tersebut diselesaikan melewati batas waktu yang ditentukan maka pekerjaan tersebut dinyatakan terlambat. Dengan metoda penjadwalan maka keterlambatan ini dapat dikurangi, baik waktu maupun frekuensi. Menurut Narasimhan (1985), penjadwalan yang baik seharusnya simpel, mudah dimengerti dan dapat dilaksanakan oleh pihak manajemen dan oleh siapapun yang menggunakannya. Aturan-aturan penjadwalan seharusnya cukup kuat tetapi mempunyai tujuan yang realistis sehingga cukup *flexible* untuk memecahkan masalah yang tidak terprediksi sebelumnya dan membolehkan satu perencanaan ulang.

Dalam penelitian Harjunkski dan Buer (2016), masalah pada proses *batch multiple-stage* kimia sederhana diatasi dengan cara yang umum dimana sebagian besar telah menyesuaikan waktu pemrosesan sehingga *bottle neck* dapat bergeser tergantung pada kandidat solusi. Solusi optimal tentu saja unik tetapi memiliki variasi terutama dapat memberikan petunjuk jika algoritma heuristik tidak optimal didorong menuju solusi suboptimal dan tidak dapat sepenuhnya menggunakan fleksibilitas yang diinginkan. Dalam penelitian Moh. Husen, Ilyas Masudin, Dana Marsetiya Utama (2015), penjadwalan menggunakan metode *Simulates Annealing* (SA) dengan bantuan Matlab dapat menghasilkan waktu total penyelesaian (*makespan*) lebih cepat dari penjadwalan yang ada di perusahaan. Metode SA mensimulasikan

proses *annealing* pada pembuatan materi yang terdiri dari butiran Kristal atau logam. Tujuan dari proses ini adalah menghasilkan struktur Kristal yang baik dengan menggunakan energy seminimal mungkin. Setiap pesanan yang diterima sangat bervariasi dalam jenis pakaian yang akan diproduksi serta jumlahnya sehingga perlu dilakukan penjadwalan produksi pada setiap pesanan. Dalam penyusunan penjadwalan produksi, Ahmad Juniar (2015) melakukan penelitian untuk penjadwalan produksi *single-stage* pada mesin parallel dan menghasilkan maskespan yang minimal dengan menggunakan algoritma *greedy*.

Algoritma *greedy* membentuk solusi langkah per langkah sebagai berikut:

- 1) Terdapat banyak pilihan yang perlu 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.
- 2) Pendekatan yang digunakan di dalam algoritma *Greedy* adalah membuat pilihan yang terlihat memberikan perolehan terbaik, yaitu dengan membuat pilihan *optimum local* pada setiap langkah yang diharapkan akan mendapatkan solusi *optimum global*.

Algoritma *Greedy* disarankan pada pemindahan *edge* per *edge* dan pada setiap langkah yang diambil tidak memikirkan konsekuensi ke depan, *Greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada serta sebagian masalah *greedy* tidak selalu benar – benar

memberikan *optimum* tapi pasti memberikan solusi yang mendekati nilai *optimum*. Masalah optimasi dalam konteks algoritma *Greedy* disusun oleh elemen- elemen sebagai berikut:

- 1) Himpunan kandidat yang berisi elemen- elemen yang memiliki peluang untuk pembentuk solusi. Pada persoalan lintasan pendek dalam *graph*, himpunan kandidat ini adalah himpunan simpul dari *graph* tersebut.
- 2) Himpunan solusi yang berisi solusi dari permasalahan yang diselesaikan dan elemennya terdiri dari elemen dalam himpunan kandidat, namun tidak semuanya atau dengan kata lain himpunan solusi ini adalah bagian dari himpunan kandidat.
- 3) Fungsi seleksi yang pada setiap langkah memilih kandidat yang paling mungkin untuk menghasilkan solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
- 4) Fungsi kelayakan yang memeriksa apakah suatu kandidat yang telah dipilih (*diseleksi*) dapat memberikan solusi yang layak.
- 5) Fungsi objektif yang memaksimalkan atau meminimumkan nilai solusi. Tujuannya adalah memilih satu saja solusi terbaik dari masing- masing anggota himpunan solusi.

Asumsi bahwa bobot semua *edge* bernilai positif. Algoritma *greedy* untuk mencari lintasan terpendek dapat dirumuskan sebagai berikut:

- 1) Periksa semua *edge* yang langsung bersesuaian dengan mesin awal. Pilih *edge* yang bobotnya terkecil. *Edge* ini menjadikan terpendek pertama, sebut saja sebagai $L(1)$.

- 2) Tentukan lintasan terpendek ke dua dengan cara sebagai berikut: Hitung $d(i) = \text{panjang } L(1) + \text{ bobot } \textit{edge}$ dari mesin akhir $L(1)$ ke mesin lainnya.
- 3) Bandingkan $d(i)$ yang terkecil
- 4) Bandingkan $d(i)$ dengan bobot \textit{edge} , (a,i) lebih kecil daripada $d(i)$, maka

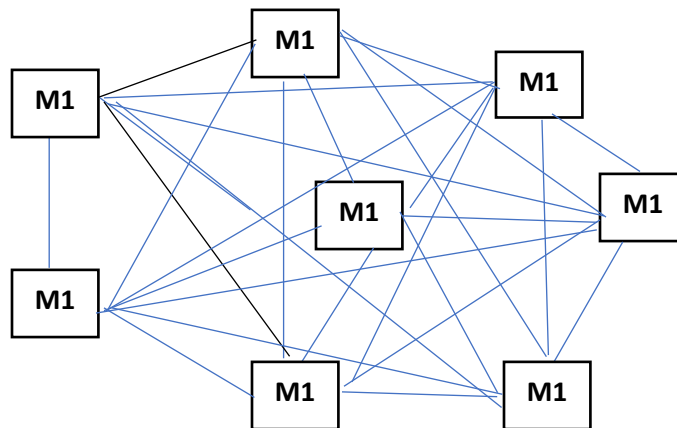
$L(2) = L(1) \cup (\textit{edge} \text{ dari mesin akhir } L(i) \text{ ke mesin } i)$.

- 5) Dengan cara yang sama, ulangi langkah (2) untuk lintasan terpendek berikutnya.

3. Metode Penelitian

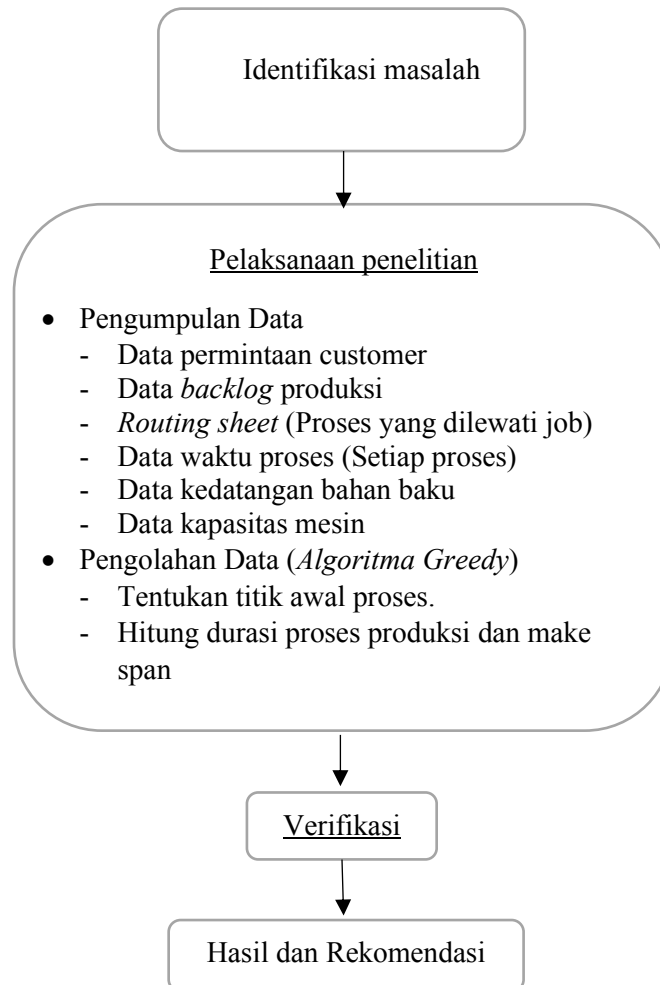
Metode penelitian yang dilakukan adalah metode komperatif yaitu penelitian yang bersifat membandingkan. Penelitian dilakukan dengan membandingkan sistem

penjawalan yang sudah ada di perusahaan dengan penjawalan menggunakan algoritma *greedy* dengan model penelitian seperti berikut:



Gambar 1. *Job shop process*

Tahapan penelitian digambarkan seperti dibawah ini:



Gambar 2. Tahapan Penelitian

Algoritma greedy untuk mencari lintasan terpendek dapat dirumuskan sebagai berikut:

- 1) Periksa semua *edge* yang langsung bersesuaian dengan mesin awal. Pilih *edge* yang bobotnya terkecil. *Edge* ini menjadikan terpendek pertama, sebut saja sebagai $L(1)$.
- 2) Tentukan lintasan terpendek ke dua dengan cara sebagai berikut: Hitung $d(i) = \text{panjang } L(1) +$
- 3) bobot *edge* dari mesin akhir $L(1)$ ke mesin lainnya.
- 4) Bandingkan $d(i)$ yang terkecil
- 5) Bandingkan $d(i)$ dengan bobot *edge*, (a,i) lebih kecil daripada $d(i)$, maka $L(2) = L(1) \cup (\text{edge dari mesin akhir } L(i) \text{ ke mesin } i)$.
- 6) Dengan cara yang sama, ulangi langkah (2) untuk lintasan terpendek berikutnya.
- 7) Bandingkan hasil penjadwalan terhadap jadwal produksi tanpa



penerapan penjadwalan dengan metode algoritma greedy.

4. Pembahasan

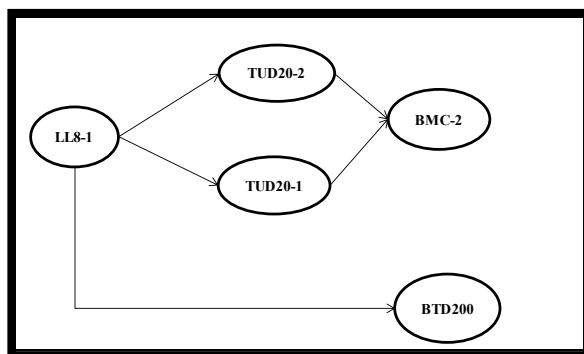
Tabel 1. Produk dan *Sequences*

No	Produk	Kode Job	Qty Job	Sequences		
				Opt#1	Opt#2	Opt#3
1	Casing Spool Body	Job#01	2	LL8-1	TUD20-1 / TUD20-2	BMC-2
2	Casing Spool Body	Job#02	2	LL8-1	TUD20-1 / TUD20-2	BMC-2
3	Casing Spool Body	Job#03	2	LL8-1	TUD20-1 / TUD20-2	BMC-2
4	Casing Spool Body	Job#04	2	LL8-1	TUD20-1 / TUD20-2	BMC-2
5	Casing Spool Body	Job#05	2	LL8-1	TUD20-1 / TUD20-2	BMC-2
6	Casing Head	Job#06	2	LL8-1	-	BTD200
7	Casing Head	Job#07	2	LL8-1	-	BTD200
8	Casing Head	Job#08	2	LL8-1	-	BTD200
9	Casing Head	Job#09	2	LL8-1	-	BTD200
10	Casing Head	Job#10	2	LL8-1	-	BTD200

Tabel 2. Ruting Proses

No	Kode Job	Ruting Proses				
		(LL8-1)	Alternatif		BTMC-2	BTD200
			(TUD20-1)	(TUD20-2)		
1	Job#01	8	22	20	10	-
2	Job#02	8	22	20	10	-
3	Job#03	8	22	20	10	-
4	Job#04	8	22	20	10	-
5	Job#05	8	22	20	10	-
6	Job#06	8	-	-	-	4
7	Job#07	8	-	-	-	4
8	Job#08	8	-	-	-	4
9	Job#09	8	-	-	-	4
10	Job#10	8	-	-	-	4

Pola aliran *job shop* dari table 1 untuk pengerjaan produk dapat digambarkan sebagai berikut:



Gambar 3. Pola Aliran Proses

5. Penyelesaian Permasalahan Penjadwalan produksi dengan Algoritma Greedy

Berikut ini ini adalah penyelesaian permasalahan penjadwalan produksi dengan algoritma greedy. Kondisi saat ini saat inialisasi:

$C = \{\text{Job1, job2, job3, job4, job5, job6, job7, job8, job9, job10}\}$

$S = \{ \}$ // Himpunan solusi.

Tabel 4. Iterasi -1

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-

Kumulatif beban stasiun kerja 1,2,3,4,5 = 8,0,0,0,0

$C = \{\text{Job1, job2, job3, job4, job5, job6, job7, job8, job9, job10}\}$

Tabel 5 Iterasi -2

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-

Kumulatif beban stasiun kerja 1,2,3,4,5 = 16,22,0,0,0

$C = \{\text{Job1, job2, job3, job4, job5, job6, job7, job8, job9, job10}\}$.



Tabel 6. Iterasi -3

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-
Job3	-	Job2	Job1	-
Kumulatif beban stasiun kerja 1,2,3,4,5 = 24,22,20,10,0				

$C = \{\text{job2, job3, job4, job5, job6, job7, job8, job9, job10}\}$.

Tabel 7. Iterasi -4

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-
Job3	-	Job2	Job1	-
Job4	-	Job3	Job2	-
Kumulatif beban stasiun kerja 1,2,3,4,5 = 32,22,40,20,0				

$C = \{\text{job3, job4, job5, job6, job7, job8, job9, job10}\}$

Tabel 8. Iterasi -5

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-
Job3	-	Job2	Job1	-
Job4	-	Job3	Job2	-
Job5	Job4	-	Job3	-
Kumulatif beban stasiun kerja 1,2,3,4,5 = 40,44,40,30,0				

$C = \{\text{job4, job5, job6, job7, job8, job9, job10}\}$



Tabel 9. Iterasi -6

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-
Job3	-	Job2	Job1	-
Job4	-	Job3	Job2	-
Job5	Job4	-	Job3	
Job6	-	Job5	Job4	

Kumulatif beban stasiun kerja 1, 2,3,4,5 = 48,44,60,40,0

$C = \{job5, job6, job7, job8, job9, job10\}$

Tabel 10. Iterasi -7

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-
Job3	-	Job2	Job1	-
Job4	-	Job3	Job2	-
Job5	Job4	-	Job3	
Job6	-	Job5	Job4	
Job7	-		Job5	Job6

Kumulatif beban stasiun kerja 1,2,3,4,5 = 56,44,60,50,4

$C = \{job6, job7, job8, job9, job10\}$

Tabel 11. Iterasi -8

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-
Job3	-	Job2	Job1	-
Job4	-	Job3	Job2	-
Job5	Job4	-	Job3	
Job6	-	Job5	Job4	
Job7	-	-	Job5	Job6
Job8	-	-	-	Job7

Kumulatif beban stasiun kerja 1,2,3,4,5 = 64,44,60,50,8

$C = \{job7, job8, job9, job10\}$



Tabel 12. Iterasi -9

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-
Job3	-	Job2	Job1	-
Job4	-	Job3	Job2	-
Job5	Job4	-	Job3	
Job6	-	Job5	Job4	
Job7	-	-	Job5	Job6
Job8	-	-	-	Job7
Job9	-	-	-	Job8
Kumulatif beban stasiun kerja 1,2,3,4,5 = 72,44,60,50,12				

$C = \{job8, job9, job10\}$

Tabel 13. Iterasi -10

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-
Job3	-	Job2	Job1	-
Job4	-	Job3	Job2	-
Job5	Job4	-	Job3	
Job6	-	Job5	Job4	
Job7	-	-	Job5	Job6
Job8	-	-	-	Job7
Job9	-	-	-	Job8
Job10	-	-	-	Job9
Kumulatif beban stasiun kerja 1,2,3,4,5 = 80,44,60,50,16				

$C = \{job9, job10\}$



Tabel 14. Iterasi -11

LL8-1	TUD20-1	TUD20-2	BMC	TUD200
Step#1	Step#2	Step#3	Step#4	Step#5
Job1	-	-	-	-
Job2	Job1	-	-	-
Job3	-	Job2	Job1	-
Job4	-	Job3	Job2	-
Job5	Job4	-	Job3	
Job6	-	Job5	Job4	
Job7	-	-	Job5	Job6
Job8	-	-	-	Job7
Job9	-	-	-	Job8
Job10	-	-	-	Job9
-	-	-	-	Job10
Kumulatif beban stasiun kerja 1,2,3,4,5 = 80,44,60,50,20				

$C = \{ \}$ // Himpunan kandidat telah kosong, iterasi berakhir

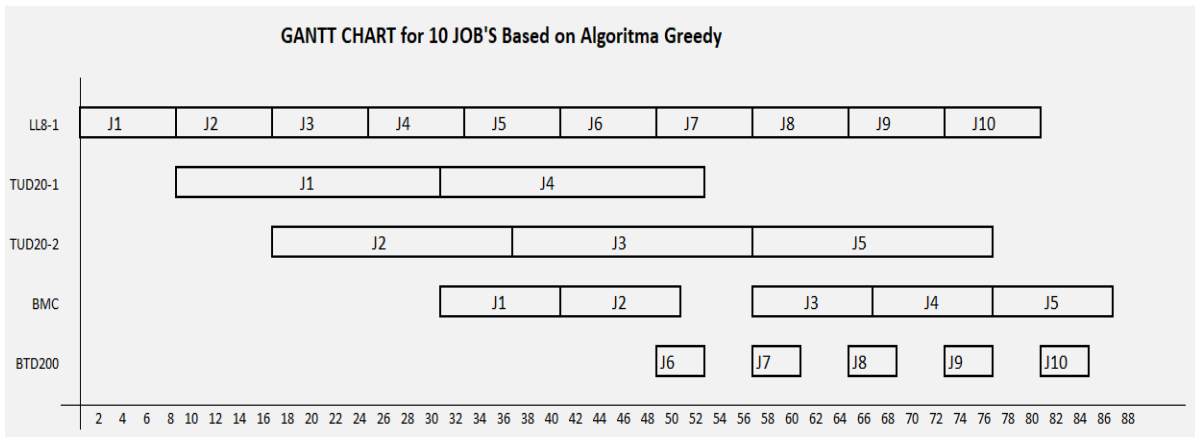
Tabel 15. Iterasi -12

LL8-1	Idle (Jam)	TUD20-1	Idle (Jam)	TUD20-2	Idle (Jam)	BMC	Idle (Jam)	BTD200	Idle (Jam)
Step#1	Step#2		Step#3		Step#4		Step#5		
Job1	-	-	8	-	8	-	8	-	8
Job2	-	Job1	-	-	8	-	22	-	8
Job3	-	-	-	Job2	-	Job1	-	-	8
Job4	-	-	-	Job3	-	Job2	-	-	8
Job5	-	Job4	-	-	-	Job3	-	-	8
Job6	-	-	-	Job5	-	Job4	-	-	8
Job7	-	-	-	-	-	Job5	-	Job6	
Job8	-	-	-	-	-	-	-	Job7	4
Job9	-	-	-	-	-	-	-	Job8	4
Job10	-	-	-	-	-	-	-	Job9	4
-								Job10	4
Kumulatif beban stasiun kerja 1,2,3,4,5 = 80,52,76,80,84									

$C = \{ \}$ // Himpunan kandidat telah kosong, iterasi berakhir

Degan algoritma greedy maka dihasilkan penjadwalan produksi dengan total waktu penyelesaian pekerjaan = max (80,52,76,80,88). Algoritma untuk kasus ini mempunyai kompleksitas waktu $O(n)$. Hasil akhir dari greedy ditunjukkan oleh langkah 12.

Jika hasil dari algoritma greedy dibuat dalam gantt chart diperoleh hasil sebagai berikut :



Gambar 4. Gantt Chart – Algoritma Greedy

6. Hasil dan Saran

Dari hasil pembahasan diketahui terdapat perbedaan berkaitan dengan kumulatif beban stasiun kerja khususnya mesin BMC. Dengan algoritma Greedy mesin BMC memiliki kumulatif sebanyak 80 jam, sedangkan berdasarkan Gantt Chart diketahui untuk mesin BMC kumulatif waktu sebesar 86 Jam. Terjadi selisih 6 jam di saat perpindahan J2 ke J3.

Algoritma Greedy masih memiliki kelemahan untuk menghitung waktu tunggu secara manual karena adanya perbedaan waktu proses mesin dan terjadinya pemilihan alternative mesin. Untuk mendapatkan akurasi waktu tunggu dapat menggunakan Gantt

Chart sehingga diperoleh perhitungan *make span* yang lebih baik.

Daftar Pustaka

- Andy, Ham (2017). *Flexible job shop scheduling problem for parallel batch processing machine with compatible job families (Elsevier Journal)*. Liberty University, USA.
- Baker, Kenneth R.(1974), *Introduction To Sequencing and Scheduling* , Jhon Willey and Sons, Inc. New York.
- Bierwirth, C., Kuhpfasl,J.(2017). *Extended GRASP for the job shop scheduling problem with Total weighted tardiness objective*. Halle University, Germany.



- Bouazza, W, Sallez, W, Beldjilali, B (2017). *A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect (Elsevier Journal)*. Oran University, Algeria.
- Conway (1967). *Theory of Scheduling*. Devor Publication, INC. Mineola, New York.
- Dios, Manuel, F.V, Victor, M.FJose (2018). *Efficient Heuristics for the hybrid Flow shop scheduling problem with missing Operations*. University of Seville, Spain.
- Fogarty, Robin (1991). *How to Integrate the Curricula*. Palatine, Illinois: IRI/Skylight Publishing, Inc.
- G. Brassard, C. Crépeau and M. Sántha(1996), "Oblivious transfers and intersecting codes", *IEEE Transactions on Information Theory*.
- Giliang Gong, dkk., (2017). *A new double flexible job-shop scheduling problem integrating processing time, green production and human factor indicators*, *Journal of Clear production*, doi: 10.1016/j.jclepro.2017.10.188
- Gu, Manzhan, dkk. (2016). *Single-machine scheduling problems with machine aging effect and an optional maintenace activity*. Shanghai University, China.
- Hamaz, Idir (2017). *Cycle job shop problem with varying processing times (Elsevier Journal)*. Toulouse University, France.
- Harjunkaoski and Reinhard Buer (2016). *Industial scheduling based on flexible heuristics (Elsevier Journal)*. Ladenburg, Germany.
- Jamili, Amin (2017). *Robust Job Shob scheduling problem: Mathematical Models, Exact and heuristics Algorithms*. Tehran University, Iran.
- Juniar, Ahmad (2015). *Penerapan Algoritma Greedy pada Penjadwalan Produksi Single- stage dengan Paralel Machine di Industri konveksi*. Kementrian Perindustrian. Jakarta.
- Kashif, Mahmood (2017). *Performance Analysis of Flexible Manufacturing system (FMS) (Elsevier Journal)*. Tallin Univerityof Technology, Estonia.
- Moh. Husen, Ilyas Masudin, Dana Marsetiya Utama (2015). *Penjadwalan Job Shop Statik dengan Metode Simulated Annealing untuk Meminimasi Waktu Makespan*. Universitas Muhammadiyah, Malang.
- Rawat, G.S 92016). *Productivity Measurement of Manufacturing System (Elsevier Journal)*. Uttarakhand, India.
- Sreekara, MBS, dkk. (2017). *An effective hybrid multi objective evolutionary algorithm for solving ral time event in flexible job shop scheduling problem*. K L University, India.
- Visa, Charan MVN (2017). *A review on facility layout Design of An automated Guided Vechile In Flexible Manufacturing system (Elsevier Journal)*. Bhopal, India.
- Xixing Li, dkk. (2017). *Hybrid artificial bee colony algorithm with a rescheduling strategy for solving flexible job shop scheduling problems*. Wuhan University, China.