

BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terdahulu

Untuk mendukung perumusan penelitian ini, peneliti menggunakan beberapa penelitian terdahulu sebagai acuan dalam proses penulisan dan analisa lebih lanjut. Peneliti menggunakan beberapa penelitian terdahulu sebagai acuan dalam proses pengolahan data, diantaranya metode *Johnson* untuk mengidentifikasi tipe penjadwalan berdasarkan mesin produksi. Peneliti juga menggunakan penelitian terdahulu terkait algoritma *Heuristic* dalam penjadwalan *Flowshop* seperti metode *Campbell Dudek Smith (CDS)*, *Nawaz Ensore Ham (NEH)*, dan *Palmer*. Berikut beberapa jurnal dan penelitian terdahulu yang digunakan sebagai acuan dalam penelitian ini :

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Metode	Uraian
1	Chamdan Mashuria, Ahmad Heru Mujianto, Hadi Sucipto and Rinaldo Yudianto Arsam (2020)	Penerapan Algoritma <i>Campbell Dudek Smith (CDS)</i> untuk Optimasi Waktu Produksi Pada Penjadwalan Produksi	<i>Johnson</i> dan <i>CDS</i>	Tujuan : Memberi solusi dari masalah yang dialami perusahaan dengan memanfaatkan nilai <i>makespan</i> dengan hasil paling minimum. Hasil Penelitian : Meminimasi <i>makespan</i> dan juga waktu proses produksi bagi perusahaan, karena metode <i>CDS</i> menggunakan perbandingan

				<p>waktu proses pada setiap mesin dengan mendahulukan waktu proses terkecil untuk urutan pemrosesannya, serta dengan melakukan 6 kali iterasi untuk menentukan waktu proses terkecilnya. Dengan mendahulukan waktu proses terkecil maka dihasilkan <i>makespan</i> yang paling minimal. Hasil penerapan <i>CDS</i> pada proses produksi perusahaan dengan jumlah produk jenis wajan yang berjumlah 12 buah menghasilkan 6 iterasi dan diperoleh nilai minimal <i>makespan</i> yaitu 210,12 menit. Aplikasi menunjukkan akurasi sebesar 99,99% untuk waktu pertama dan 99,96% untuk waktu kedua jika dibandingkan dengan</p>
--	--	--	--	---

				perhitungan manual.
2	Andi Sudiarso, Wildanul Isnaini (2018)	Optimasi Penjadwalan Produksi di IKM ED Alumunium Yogyakarta	<i>CDS</i> dan <i>Palmer</i>	<p>Tujuan : Untuk melakukan optimasi penjadwalan produksi dengan <i>makespan</i> yang lebih kecil dari pada sistem penjadwalan sebelumnya dengan metode <i>Palmer</i>, Algoritma <i>Dannenbring</i>, serta Algoritma <i>Campbell Dudek Smith (CDS)</i>.</p> <p>Hasil Penelitian : Penelitian ini menggunakan algoritma <i>Palmer</i>, <i>Dannenbring</i>, dan <i>Campbell Dukdek Smith (CDS)</i> karena sesuai dengan kondisi perusahaan ED Aluminium yaitu <i>flowshop</i>. Produk yang akan dijadwalkan berjumlah 30 buah yang dibagi kedalam kelompok wajan (25 produk) dan Panci Tasik (5 produk).</p>
3	Diah Pramestari, Luviana	Analisis Penjadwalan Produksi Produk	<i>CDS</i>	Tujuan : Mengetahui nilai <i>makespan</i>

	Firlyanti Darmawan (2018)	<i>Oxygen</i> sensor dengan metode <i>Heuristic GUPTA</i> dan <i>Campbell Dudek Smith</i> di PT. DENSO INDONESIA		<p>menggunakan metode <i>Campbell Dudek Smith (CDS)</i> dan <i>Algoritma Gupta</i> untuk membuktikan metode yang memiliki efisiensi penjadwalan yang terbaik dan juga untuk menentukan <i>dispatch list</i> yang optimal.</p> <p>Hasil Penelitian : Penjadwalan produksi produk <i>oxygen sensor</i> di PT. DENSO INDONESIA dengan menggunakan metode <i>CDS</i> dan Metode <i>Algoritma Gupta</i> memperoleh hasil penggunaan metode <i>CDS</i> yang paling baik, karena memiliki <i>makespan</i> terkecil dari metode <i>Gupta</i> dan dari <i>makespan</i> yang diterapkan perusahaan saat ini.</p>
4	Dana Marsetya Utama dan Febrianto Susastro (2014)	Penjadwalan <i>Flowshop</i> Menggunakan <i>Algoritma Nawaz Ensore Ham</i>	<i>NEH</i>	Tujuan : Mengurangi <i>makespan</i> dan total biaya produksi sebagai salah satu

				<p>dampak dari penjadwalan mesin.</p> <p>Hasil Penelitian : Penerapan metode <i>NEH</i> dalam menjadwalkan 10 job pada 5 mesin bertujuan untuk mengurangi <i>makespan</i>. Dari hasil simulasi didapatkan bahwa nilai <i>makespan</i> dengan metode <i>NEH</i> memberikan nilai yang lebih kecil dari metode perusahaan yang menerapkan penjadwalan mesin dengan basis <i>first come first service (FCFS)</i>. Kemudian terdapat pengurangan total biaya dengan diterapkannya penjadwalan dengan metode <i>NEH</i> sebesar Rp 137.648, sedangkan dari <i>machine's utility</i> didapatkan pengoptimalan waktu <i>idle</i> mesin sebesar 582 menit.</p>
--	--	--	--	--

II.2 Definisi Penjadwalan

Penjadwalan adalah proses pengurutan pengerjaan sebuah produk secara *long term* dan menyeluruh pada mesin produksi. Dalam penjadwalan terdapat proses pengerjaan sejumlah komponen yang sering disebut dengan istilah *job*. *Job* sendiri merupakan gabungan dari beberapa aktivitas atau operasi dalam proses produksi. Tiap aktivitas atau operasi ini membutuhkan alokasi daya tertentu yang sering disebut dengan waktu proses (Ginting, 2009).

Pada penjadwalan, pengalokasian sumber daya memiliki tujuan penting yang akan dicapai. Menurut Bedworth (2002:72), terdapat dua target yang ingin dicapai melalui penjadwalan, yaitu jumlah *output* yang dihasilkan dan batas waktu penyelesaian yang telah ditetapkan (*due date*). Kedua target ini dinyatakan melalui kriteria penjadwalan seperti *minimum makespan* (waktu proses terpanjang dalam proses produksi), *minimum mean flow time* (rata-rata waktu proses produksi), *minimum mean lateness* (rata-rata keterlambatan), *minimum tardiness* (keterlambatan), *minimum mean tardiness* (rata-rata keterlambatan), minimasi *number of tardy* (jumlah keterlambatan) dan sebagainya.

Input pertama dari permasalahan dalam penjadwalan adalah menentukan urutan prioritas dalam mengambil keputusan untuk mengalokasikan tugas ke dalam sumber daya yang dimiliki. Terdapat bermacam kompleksitas masalah dalam proses ini, terutama karena terdapat beberapa pekerjaan yang harus diproses dengan sumber daya yang terbatas.

Beberapa opsi keputusan yang bisa diambil dalam penjadwalan yaitu :

- a) Pengurutan seluruh pekerjaan (*Sequencing*).
- b) Waktu mulai dan waktu selesai pekerjaan (*timing*).
- c) Urutan operasi setiap pekerjaan (*Routing*).

II.3 Tujuan Penjadwalan

Tujuan penjadwalan adalah untuk mengurangi waktu keterlambatan dari batas waktu yang ditentukan agar dapat memenuhi batas waktu yang telah disetujui dengan konsumen, penjadwalan juga dapat meningkatkan produktifitas mesin dan mengurangi waktu menganggur. Jika produktifitas mesin meningkat maka waktu menganggur akan berkurang. Dengan kata lain, perusahaan dapat mengurangi biaya produksi. Semakin baik suatu penjadwalan, maka akan semakin menguntungkan juga bagi perusahaan dan dapat menjadi dasar peningkatkan *profit* dan strategi pemenuhan kepuasan pelanggan.

Menurut Bedworth (1987), proses penjadwalan memiliki beberapa tujuan dari aktivitas penjadwalan sebagai berikut :

1. Meningkatkan efektivitas penggunaan sumber daya atau mengurangi waktu tunggunya, sehingga dapat mengurangi total waktu proses dan meningkatkan produktivitas.
2. Mengurangi persediaan barang setengah jadi atau mengurangi antrian pekerjaan yang menunggu ketika sumber daya sedang mengerjakan tugas lain.
3. Mengurangi keterlambatan pada pekerjaan yang mempunyai batas waktu penyelesaian (*due date*) sehingga akan meminimasi denda keterlambatan (*penalty*).
4. Membantu proses pengambilan keputusan perencanaan kapasitas pabrik dan jenis kapasitas yang dibutuhkan, sehingga dapat menghindari biaya yang keluar dengan percuma.

Baker (2009:30) menjelaskan jika *makespan* suatu penjadwalan bersifat konstan, maka urutan kerja dapat menurunkan *mean flowtime* dan juga menurunkan *WIP (Work In Process)*. Tujuan akhir dari proses penjadwalan adalah pemenuhan *due date*. Jika terjadi keterlambatan dalam pemenuhan *due date* yang telah ditetapkan dapat dikenakan suatu denda.

Untuk mengurangi suatu denda keterlambatan, digunakan sebuah aturan prioritas. Aturan prioritas berisi panduan urutan pengerjaan yang diutamakan. Aturan prioritas bertujuan untuk mengurangi *completion time* (waktu penyelesaian), jumlah pekerjaan, dan keterlambatan kerja sementara.

Eddy (2008:321) mengatakan bahwa terdapat beberapa cara dalam menentukan prioritas untuk menetapkan sebuah pedoman *dispatching* terbaik. Beberapa aturan prioritas yang umum adalah :

A. FCFS (*First Come First Serve*)

Pekerjaan yang datang lebih awal akan dikerjakan lebih dahulu. Aturan ini banyak digunakan pada bank, dan supermarket.

B. SPT (*Shortest Processing Time*)

Pekerjaan yang memiliki waktu proses terpendek akan mendapat prioritas pertama untuk dikerjakan. Pedoman ini biasa diterapkan bagi perusahaan yang bergerak dalam bidang perakitan atau jasa.

C. EDD (*Earliest Due Date*)

Pekerjaan dengan *due date* terdekat dikerjakan lebih dahulu. Cara ini seringkali digunakan pada perusahaan yang bergerak di bidang konveksi dan tekstil.

II.4 Istilah Dalam Penjadwalan

Baker (2009) dalam pembahasan mengenai masalah penjadwalan menjelaskan mengenai beberapa istilah yang sering digunakan, yaitu :

1. *Ready time* (r_j), yaitu waktu yang menunjukkan saat *job* siap untuk dikerjakan.
2. *Waiting time* (W_j), yaitu waktu yang dilalui suatu pekerjaan (menunggu) sebelum mulai diproses.
3. *Flow Time* (F_j), yaitu waktu antara *job* ke- j siap dikerjakan hingga *job* tersebut diselesaikan.

$$F_j = C_j - r_j$$

4. *Completion Time* (C_i), yaitu waktu yang dibutuhkan untuk menyelesaikan suatu operasi dari pekerjaan j pada mesin i . Waktu proses juga mencakup waktu yang dibutuhkan dalam tahap persiapan dan pengaturan operasi. (*set up time*).

$$C_j = F_j + r_j$$

5. *Process time* ($t_{i,j}$), yaitu waktu yang diperlukan untuk menyelesaikan suatu operasi atau proses ke- i dari *job* ke- j .
6. *Due Date* (d_j), yaitu batas waktu penyelesaian yang ditentukan untuk penyelesaian produk.
7. *Lateness* (L_j), yaitu besarnya simpangan waktu penyelesaian *job* j terhadap *due date* yang telah ditentukan untuk *job* tersebut.

$$L_j = C_j - d_j \leq 0, \text{ artinya saat penyelesaian memenuhi batas akhir.}$$

$$L_j = C_j - d_j \geq 0, \text{ artinya saat penyelesaian melewati batas akhir.}$$

8. *Tardiness* (T_j), yaitu besarnya keterlambatan dari *job* j . *Tardiness* adalah *Lateness* yang bernilai positif.
9. *Earliness* (e_j), yaitu *Lateness* yang bernilai negatif.
10. *Slack* (S_i), yaitu waktu sisa yang tersedia bagi suatu pekerjaan.

$$S_i = d_i - t_i$$

11. *Makespan* (M), yaitu jangka waktu penyelesaian terpanjang suatu pekerjaan merupakan penjumlahan dari seluruh waktu proses suatu mesin.

II.5 Jenis Penjadwalan

Klasifikasi penjadwalan menurut Pinedo (2012), berdasarkan jumlah mesin yang digunakan dibagi menjadi:

1) Penjadwalan mesin tunggal (*single machine*)

Penjadwalan mesin tunggal merupakan penjadwalan satu mesin yang digunakan untuk memproses seluruh pekerjaan (n -jobs) untuk satu jenis proses produksi.

2) Penjadwalan Paralel (mesin majemuk)

Penjadwalan paralel dibagi menjadi:

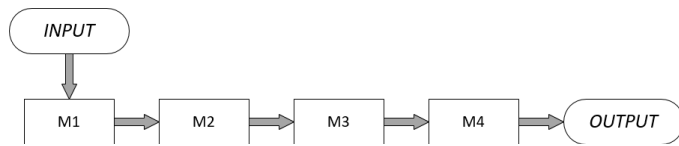
a) Penjadwalan N job pada mesin paralel identik (*identical machines parallel*). Ciri khas penjadwalan paralel identik adalah pengalokasian beban ke mesin yang lebih dahulu *idle*/kosong.

b) Penjadwalan N job pada mesin paralel non identik, dimana setiap mesin mempunyai fungsi yang sama dengan waktu proses yang berbeda-beda. Alternatif waktu terpendek dari setiap *job* tidak selalu menjadi keputusan alokasi terbaik pada mesin.

c) Penjadwalan N job pada mesin *paralel unrelated* perluasan dari paralel non identik. Terdapat m mesin paralel, dimana mesin i untuk memproses *job* j maka kecepatan mesin adalah v_{ij} .

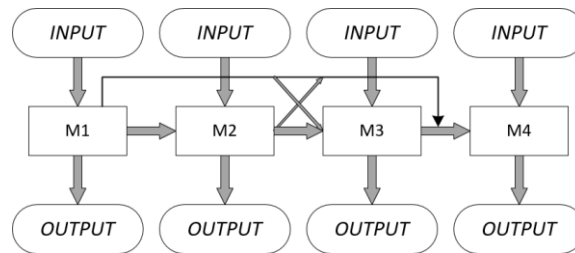
d) Penjadwalan *Flow Shop* dan *Flexible Flow shop*, terdapat m mesin serial dimana masing-masing *job* harus diproses di setiap mesin. Semua *job* harus mengikuti rute yang sama. Setelah proses selesai di satu mesin maka akan dilanjutkan proses pada mesin selanjutnya. Aliran proses, berdasarkan aliran proses dibagi menjadi tiga bagian yaitu *pure flowshop* dan *general flowshop*.

i. *Pure flowshop*, pola aliran prosesnya identik. Setiap *job* melewati seluruh mesin yang bekerja dari proses awal hingga akhir sesuai urutan mesin.



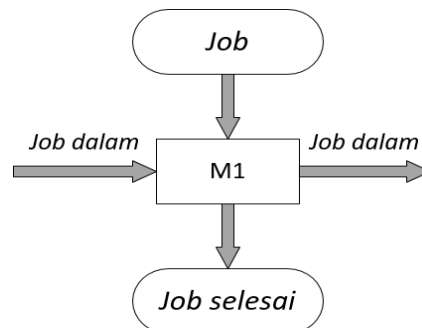
Gambar 2.1 Pola Aliran *Pure Flowshop*

ii. *General flowshop, flowshop* dengan aliran proses yang tidak identik. Setiap *job* tidak selalu melewati seluruh mesin yang bekerja.



Gambar 2.2 Pola Aliran *General Flowshop*

e) Penjadwalan *Job Shop* dan *Flexible Job Shop* adalah kondisi dimana terdapat m mesin dengan setiap *job* memiliki rute produksi yang harus diikuti. *Job* dapat diproses lebih dari satu kali pada mesin yang sama. Kondisi ini yang sering disebut dengan *recirculation*. *Flexible job shop* adalah perluasan dari *job shop* dan mesin paralel. Pada kasus ini terdapat m mesin seri dengan c stasiun kerja dengan sejumlah mesin identik untuk setiap stasiun kerja.



Gambar 2.3 Pola Aliran *Jobshop*

II.6 Ukuran Performansi Penjadwalan

Menurut Baker & Trietsch (2009), ukuran performansi adalah salah satu indikator keberhasilan dari penjadwalan terbaik yang diinginkan. Kriteria penjadwalan urutan *job* adalah sebagai berikut :

1. Kriteria performansi penjadwalan berdasarkan atribut tugas.

- a) *Flow time* (F_i).
- b) *Completion Time* (C_i).
- c) *Mean flow time* .
- d) *Mean weight flow time*.
- e) *Maximum Lateness*.
- f) *Tardiness*.
- g) *Total tardiness*.
- h) *Mean tardiness*.

2. Kriteria berdasarkan atribut pabrik

- a) Utilitas mesin, merupakan perbandingan dari jumlah *mesin* yang dibebankan pada mesin dengan rentan waktu untuk menyelesaikan seluruh tugas pada setiap mesin.

$$U = \frac{\sum_{j=1}^n (U_j)}{m} \text{ dimana } U_m = \frac{\sum_{j=1}^n (T_i)}{F_{max}}$$

- b) Minimasi *makespan*, yaitu optimalisasi waktu penyelesaian seluruh *job* terpanjang pada suatu proses produksi yang merupakan total dari seluruh waktu proses.

$$Ms = \sum_{t=1}^n t_i$$

- c) Pemenuhan *due date*, merupakan penyelesaian pekerjaan sesuai dengan batas waktu yang ditentukan oleh pelanggan untuk penyelesaian suatu produk yang diinginkan.

II.7 Algoritma dalam Penjadwalan *Flowshop*

Dalam penjadwalan produksi tipe *flowshop* terdapat metode-metode yang dapat digunakan untuk menyelesaikan masalah penjadwalan. Pendekatan yang paling sering digunakan adalah menggunakan pendekatan *heuristic*. Langkah awal dalam mengolah data pokok produksi yaitu mengidentifikasi tipe penjadwalan yang diterapkan perusahaan. Setelah dilakukan identifikasi, langkah selanjutnya adalah mengolah data pokok berupa waktu proses produksi sebuah produk. Selanjutnya data waktu proses akan diolah masing-masing dengan metode *heuristic* klasik. Beberapa metode *heuristic* klasik yang biasa digunakan adalah metode *Campbell Dudek Smith (CDS)*, metode *Nawaz Enscore Ham (NEH)*, dan metode *Palmer*.

II.7.1 Algoritma *Johnson*

Salah satu syarat pada algoritma *Johnson* adalah setiap pekerjaan yang akan diselesaikan harus melewati setiap mesin. Setiap mesin bekerja sesuai dengan jadwal urutan proses produksi. Algoritma *Johnson* merupakan suatu aturan meminimalkan *makespan* dari

proses produksi 2 mesin yang disusun seri. Permasalahan *Johnson* diformulasikan dengan j job yang diproses pada 2 mesin dengan $t_{j,1}$ adalah waktu proses pada mesin 1 dan $t_{j,2}$ adalah waktu proses pada mesin 2. Tahapan perhitungan dengan algoritma *Johnson* adalah :

- 1) Tentukan nilai $(t_{j,1}, t_{j,2})$.
- 2) Jika waktu proses minimal terdapat pada mesin pertama (misal $t_{j,1}$), tempatkan *job* tersebut pada awal deret penjadwalan.
- 3) Bila waktu proses minimal didapat pada mesin kedua (misal $t_{j,2}$), *job* tersebut ditempatkan pada posisi akhir deret penjadwalan.
- 4) Pindahkan *job-job* tersebut dari daftarnya dan susun dalam bentuk deret penjadwalan. Total waktu $t_{1,1}$ yaitu waktu proses *job* 1 pada mesin 1. Total waktu $t_{1,2}$ yaitu $t_{1,1} + t_{1,2}$. Total waktu $t_{2,1}$ yaitu $t_{1,1} + t_{2,1}$. Total waktu $t_{2,2}$ yaitu $\max \{ t_{1,2}, t_{2,1} \} + t_{2,2}$ dan seterusnya. Jika masih ada *job* yang tersisa ulangi kembali langkah 1, sebaliknya jika tidak ada lagi *job* yang tersisa berarti penjadwalan sudah selesai.

II.7.2 Campbell Dudek Smith (CDS)

Pada metode *Campbell Dudek Smith* (1970:633) proses penjadwalan atau penugasan kerja dilakukan berdasarkan atas waktu kerja yang terkecil yang digunakan dalam melakukan produksi. Dalam permasalahan ini, digunakan n job dan m mesin. Metode *CDS* ini adalah metode yang pertama kali ditemukan oleh Campbell, Dudek, dan Smith pada tahun 1965, yang dilakukan untuk mengurutkan n pekerjaan terhadap m mesin. *Campbell Dudek Smith* (1970:631) memutuskan untuk urutan yang pertama yaitu :

$$t_{j,1}^k = t_{j,1} \text{ dan } t_{j,2}^k = t_{j,2}.$$

sebagai waktu proses pada mesin pertama dan mesin terakhir. Untuk urutan yang kedua dirumuskan dengan:

$$t_{j,1}^k = t_{j,1} + t_{j,2}$$

$$t_{j,2}^k = t_{j,m} + t_{j,m-1}$$

sebagai waktu proses pada dua mesin pertama dan dua mesin yang terakhir.

untuk urutan ke-k:

$$t_{j,1}^k = \sum_{i=1}^m t_{j,i}$$

$$t_{j,2}^k = \sum_{i=m+1-k}^m t_{j,i}$$

Keterangan :

j = job

i = mesin

m = jumlah mesin yang bekerja (awal-akhir)

k = 1,2,3,...,(m-1)

Berikut ini merupakan tabel iterasi dalam dua mesin.

Tabel 2.2 Iterasi Dua Mesin

k	$t_{j,1}^k$ (total mesin pertama)	$t_{j,2}^k$ (total mesin kedua)
1	$t_{j,1}$	t_{m+1-1}
2	$t_{j,1} + t_{j,2}$	$t_{m+1-1} + t_{m+1-2}$
3	$t_{j,1} + t_{j,2} + t_{j,3}$	$t_{m+1-1} + t_{m+1-2} + t_{m+1-3}$
...
m-1	$t_{j,1} + t_{j,2} + t_{j,3} + \dots + t_{j,(m-1)}$	$t_{m+1-1} + t_{m+1-2} + t_{m+1-3} + t_{m+1-(m-1)}$

Dari penyusunan atau penjadwalan yang ada diharapkan akan mengurangi waktu menganggur dari mesin karena pengaturan yang kurang tepat.

Perhitungan ini berlangsung terus dengan ketentuan $k =$

1,2,3,...,(m-1), artinya harga perhitungan k mulai dari 1 sampai dengan m , bentuk perhitungan melalui tabel-tabel iterasi (k) dari 1 s/d m tersebut dan setiap tabel memiliki urutan *job* tersendiri dan terdapat p dimana $p = m-1$.

“Campbell, Dudek, dan Smith mencoba metode mereka dan menguji *performance*-nya pada beberapa masalah, mereka menemukan bahwa metode *Campbell Dudek Smith* lebih efektif, baik untuk masalah kecil maupun masalah besar.”

Perhitungan metode *Campbell Dudek and Smith* (1970:631) dilakukan dengan tahapan-tahapan berikut:

- 1) Ambil urutan pertama $k = 1$. Untuk seluruh tugas yang ada, cari harga $t_{j,1}^k$ dan $t_{j,2}^k$ yang minimal yang merupakan waktu proses mesin pertama dan kedua pada iterasi ke- k .
- 2) Jika waktu minimum didapat pada mesin pertama (misal $t_{j,1}^k$) selanjutnya tempatkan tugas tersebut pada urutan awal, bila waktu minimal didapat pada mesin kedua (misal $t_{j,2}^k$) tugas tersebut ditempatkan pada urutan terakhir.
- 3) Pindahkan tugas-tugas tersebut dari daftarnya dan urutkan. Total waktu $t_{1,1}$ yaitu waktu proses *job* 1 pada mesin 1. Total waktu yaitu $t_{1,1}$. Total waktu $t_{1,2}$ yaitu $t_{1,1} + t_{1,2}$. Total waktu $t_{2,1}$ yaitu $t_{1,1} + t_{2,1}$. Total waktu $t_{2,2}$ yaitu $\max \{ t_{1,2}, t_{2,1} \} + t_{2,2}$ dan seterusnya. Jika masih ada tugas yang tersisa ulangi kembali langkah 1, sebaliknya jika tidak ada lagi tugas yang tersisa, berarti pengurutan telah selesai.

II.7.3 *Nawaz Ensore Ham (NEH)*

Metode Nawaz Ensore Ham (NEH) ini dikembangkan oleh Muhammad Nawaz, E. Emory Ensore Jr, dan Inyong Ham pada tahun 1983. “*In a general flowshop, where all the jobs must*

past through all the machines in the same order, certain heuristic algorithms propose that the jobs with higher total process time should be given higher priority than the jobs with less total process time” yang artinya dalam penjadwalan *flowshop* secara umum, dimana semua *job* harus melewati semua mesin pada order yang sama. Algoritma *heuristic* ini mengusulkan bahwa *job* dengan total waktu proses yang lebih besar seharusnya diberikan prioritas yang lebih besar dari pada *job* dengan total waktu proses yang lebih kecil (Nawaz, 1983).

NEH merupakan salah satu algoritma yang bersifat *constructive heuristic*. Algoritma NEH mengasumsikan *job* yang memiliki total waktu proses untuk semua mesin yang lebih besar harus didahulukan dibanding *job* dengan total waktu proses yang lebih kecil. NEH menginisialisasikan urutan *job* secara *descending* berdasarkan total waktu proses tiap *job*nya. Kemudian dilakukan proses *partial sequence*, yaitu menentukan urutan terbaik dari setiap posisi *job* yang mungkin.

Beberapa penelitian penjadwalan produksi *flowshop* menggunakan algoritma NEH antara lain oleh Satriawan (2010). Metode penjadwalan yang digunakan dalam penelitian ini adalah algoritma genetika dan algoritma *Nawaz Enscore Ham* (NEH) untuk menentukan solusi optimum global. Penelitian lain pada penjadwalan *flexible flowshop* dengan kriteria minimasi *mean tardiness* dilakukan oleh Supriyanto (2008) dengan mengembangkan *Nawas Enscore Ham* pada *flexible flowshop*. Dalam penelitian ini prioritas yang digunakan menggunakan *dispatching rules* yaitu *earliest due date* (EDD), *first come first served* (FCFS), dan *Nawaz Enscore Ham* (NEH). Penggunaan ketiga prioritas ini didasarkan aturan prioritas (*priority rule*) yang berisi panduan mengurutkan pekerjaan yang harus dilakukan.

Untuk penjadwalan n *job* terhadap mesin dilakukan dengan

algoritma *NEH* dengan langkah-langkah :

Langkah 1

- a. Jumlah waktu prosese setiap *job*.
- b. Urutkan semua *job* menurut jumlah waktu prosesnya dimulai dari yang terbesar hingga yang terkecil.
- c. Hasil urutan ini disebut dengan daftar pengurutan semua *job*.

Langkah 2

- a. Set $K = 2$.
- b. Ambil *job* yang menempati urutan pertama dan kedua pada daftar pengurutan semua *job*.
- c. Buat dua alternatif calon urutan parsial baru.
- d. Hitung setiap *makespan* parsial dan *mean time partial* dari calon urutan partial baru.
- e. Pilih calon urutan parsial baru yang memiliki *makespan* dengan parsial yang terkecil. Jika ada calon urutan parsial baru yang akan memiliki *makespan* partial terkecil yang sama, pilihlah calon urutan parsial baru tadi yang memiliki *mean flow time partial* yang lebih kecil. Jika sama juga pilihlah calon urutan parsial baru tadi secara acak.
- f. Calon urutan parsial baru yang terpilih menjadi urutan parsial baru.
- g. Coret semua *job* yang diambil tadi dari daftar pengurutan semua *job* h . Periksa apakah $k = n$ (dimana n adalah jumlah *job* yang ada). Jika ya, lanjutkan ke langkah 4. Jika tidak lanjutkan ke langkah 3.

Langkah 3

- a. Set $k = k + 1$.
- b. Ambil *job* yang menempati urutan pertama dari daftar pengurutan semua *job*.
- c. Hasilkan sebanyak k calon urutan parsial baru dengan memasukkan *job* yang diambil ke dalam setiap slot urutan parsial sebelumnya.
- d. Hitungan setiap *makespan* parsial dan *mean flow time partial* dari calon urutan parsial baru.
- e. Pilih calon urutan parsial baru yang memiliki *makespan* yang parsial yang terkecil. Jika ada calon urutan parsial baru yang memiliki *makespan* parsial terkecil yang sama, pilihlah calon urutan parsial baru tadi yang memiliki *mean flow time partial* yang lebih kecil. Jika sama juga pilihlah calon urutan untuk parsial baru tadi secara acak.
- f. Calon untuk parsial baru yang terpilih menjadi urutan parsial baru.
- g. Coret semua *job-job* yang diambil tadi dari daftar pengurutan semua *job h*. Periksa apakah $k = n$ (dimana n adalah jumlah *job* yang ada). Jika ya, lanjutkan ke langkah 4. Jika tidak lanjutkan langkah 3.

Palmer

Pada penyelesaian masalah dengan menggunakan metode ini merupakan proses yang perhitungannya memiliki *slope* indeks dari rumus palmer menurut jumlah mesin yang ada, dengan demikian *job* yang memiliki *slope* indeks terbesar akan dijadwalkan lebih awal. *Slope* indeks merupakan bobot yang

diberikan untuk setiap *job*. Pada keadaan menggunakan mesin berjumlah ganjil, maka variabel bobot yang digunakan adalah $-(m-1), \dots, 0, \dots, (m-1)$ sedangkan untuk mesin berjumlah genap menggunakan variabel bobot $-(m-1), \dots, -(m-3), (m-3), \dots, (m-1)$ dengan interval antar bobotnya berjarak 2. Kemudian bobot tersebut dikalikan dengan nilai waktu proses masing-masing *job*. Nilai bobot (w) pada *job*-j merupakan total penjumlahan hasil kali nilai bobot mesin m dengan *job*-j pada mesin tersebut.

Setelah diperoleh nilai-nilai bobot tersebut, maka selanjutnya akan didapat *dispatch list* sementara yang berasal dari *job* dengan nilai bobot terbesar hingga terkecil. Kemudian *dispatch list* ini akan dijadikan acuan dalam melakukan penjumlahan waktu proses untuk mencari nilai *makespan*.

Persamaan rumus untuk bentuk umum penjadwalan dengan metode Palmer masing-masing *job* adalah :

$$S_i = - \sum_{j=1}^m (n - (2j - 1)) t_{ij}$$

Dimana $i = 1, 2, 3, \dots, n$ dan $j = 1, 2, 3, \dots, m$

Keterangan :

S_i = Nilai slope indeksnya.

m = Jumlah mesin yang dipakai.

j = Mesin yang digunakan untuk proses *job* i .

i = *Job* yang diproses.

t_{ij} = Waktu proses suatu *job* ke- i dan mesin ke- j .