

**PENGEMBANGAN PENJADWALAN *RE-ENTRANT FLOWSHOP*
BERDASARKAN ALGORITMA NAWAZ, ENSCORE, DAN HAM (NEH)
DENGAN PENDEKATAN *DISPATCHING RULE*
(Studi Kasus: PT. Sahabat Rubber Industries - Malang)**

**DEVELOPMENT *RE-ENTRANT FLOWSHOP SCHEDULING WITH*
ALGORITHM NAWAZ, ENSCORE, AND HAM (NEH) DISPATCHING
RULE APPROACH
(Case Study: PT. Sahabat Rubber Industries - Malang)**

Achmad Faizal¹, Arif Rahman², Ceria Farela Mada Tantrika³

Jurusan Teknik Industri Universitas Brawijaya

Jalan MT. Haryono 167, Malang, 65145, Indonesia

E-mail : afdalimunthe@gmail.com¹, posku@ub.ac.id², ceria_fmt@ub.ac.id³

Abstrak

Penjadwalan adalah suatu kegiatan perencanaan dalam mengalokasikan sumber daya yang tersedia untuk menyelesaikan seluruh tugas yang ada dalam jangka waktu tertentu. PT. Sahabat merupakan perusahaan manufaktur yang bergerak di bidang industri pembuatan selang LPG (Liquid Petroleum Gas). Sistem alur proses produksi yang dimiliki oleh PT.Sahabat adalah re-entrant flowshop. Permasalahan yang dihadapi oleh PT.Sahabat adalah masih tingginya waktu tunggu karena proses alokasi yang tidak tepat, sehingga menyebabkan tingkat penggunaan sumber daya yang tersedia masih rendah. Oleh karena itu, diperlukan penjadwalan untuk minimasi makespan. Dalam penelitian ini akan dikembangkan model algoritma Nawaz, Enscore dan Ham (NEH) dengan dua pendekatan dispatching rule yaitu Shortest Processing Time (SPT) dan Longest Processing Time (LPT), dengan memperhatikan routing dan precedence. Hasil penelitian ini menunjukkan bahwa dari algoritma Nawaz, Enscore, dan Ham baik dengan pendekatan mendapatkan hasil makespan sebesar 3146,16 menit. Dibandingkan dengan penjadwalan perusahaan yang memiliki makespan 4602,02 menit, maka terjadi penurunan makespan sebesar 46,27%.

Kata kunci : *Re-entrant Flowshop, minimasi makespan, Pengembangan algoritma NEH*

1. Pendahuluan

Pada era kemajuan teknologi saat ini, setiap perusahaan berusaha selalu menekankan pada proses produksi yang efisien dan efektif. Dari suatu kegiatan proses produksi untuk mendapatkan *output* yang optimum, maka seluruh proses aktivitas produksi perlu direncanakan terlebih dahulu dengan baik. Menurut Ginting (2009), Salah satu masalah yang cukup penting dalam perencanaan produksi adalah bagaimana melakukan pengaturan dan penjadwalan produksi. Dengan melakukan penjadwalan yang baik dapat mengurangi waktu mengganggu pada unit-unit produksi dan meminimumkan barang yang sedang dalam proses. Oleh karena itu, dibutuhkan penjadwalan produksi yang tepat agar dapat mengurangi waktu tidak produktif dan meningkatkan waktu produktif dalam proses produksi.

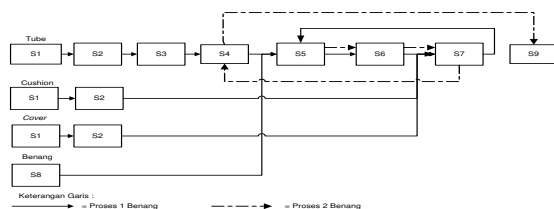
PT. Sahabat Rubber Industries (PT. Sahabat) merupakan perusahaan manufaktur yang

bergerak di bidang industri pembuatan selang LPG (*Liquid Petroleum Gas*). Selang yang diproduksi oleh PT.Sahabat diproduksi sesuai dengan spesifikasi ukuran, jenis selang, dan banyaknya benang yang digunakan. Beberapa jenis produk selang yang diproduksi adalah selang *cushion* hitam polos, selang *cushion* hitam gerigi, selang *simrit*. Sebagai perusahaan dengan kapasitas produksi yang besar dan dengan produk yang sangat bervariasi, maka PT. Sahabat menerapkan strategi produksi *Make to Order* (MTO).

Pada perusahaan yang menerapkan strategi MTO seperti PT. Sahabat, penjadwalan merupakan suatu permasalahan dimana terdapat berbagai macam tujuan dengan keterbatasan sumber daya yang tersedia dalam menyelesaikan suatu kumpulan *job*. PT.Sahabat selama ini belum memiliki penjadwalan dan melakukan proses alokasi berdasarkan jumlah permintaan terbesar akan dikerjakan terlebih dahulu dan melakukan proses tunggu di setiap satu tahapan

hingga semua proses selesai. Selain itu juga, belum adanya dokumentasi waktu proses penyelesaian setiap proses Sehingga menyebabkan timbulnya rata-rata waktu tunggu yang tinggi. Metode konvensional yang dimiliki masih belum efektif yang mengakibatkan terjadinya waktu rata-rata tunggu yang masih tinggi.

PT.Sahabat Rubber Industries memiliki 9 stasiun kerja. Stasiun kerja yaitu stasiun *kneeder*, stasiun giling, stasiun *tubing*, stasiun *oven*, stasiun rajut, stasiun pengeleman, stasiun *extruder*, stasiun pinal, dan stasiun packing. Dimana setiap pesanan produk memiliki alur proses produksi yang berbeda. PT.Sahabat memiliki alur proses produksi yaitu *re-entrant flowshop*. *Re-entrant flowshop* adalah tipe *flowshop* yang memiliki karakteristik pekerjaan dapat mengunjungi stasiun kerja atau mesin lebih dari satu kali (Chen & Lin,2009). Salah satu contoh alur proses produksi di PT. Sahabat ditunjukkan di Gambar 1.



Gambar 1. Alur Proses Produksi di PT.Sahabat

Permasalahan yang dihadapi oleh PT.Sahabat saat ini adalah proses alokasi sumber daya yang tersedia belum efektif. Sehingga menyebabkan waktu tunggu antar proses satu sama lain sangat tinggi. Secara tidak langsung berdampak pada pekerja akan melakukan lembur kerja akibat waktu nonproduktif tersebut.. Oleh karena itu diperlukan penjadwalan yang tepat untuk alokasi sumber daya yang tersedia. Penjadwalan merupakan proses pengambilan keputusan dengan melibatkan beragam sumber daya yang tersedia secara terbatas untuk menyelesaikan sekumpulan tugas dalam jangka waktu tertentu (Bedworth & Bailey,1987). Dengan melakukan penjadwalan produksi yang tepat, maka perusahaan dapat melakukan pengambilan keputusan atas waktu penyelesaian pesanan.

Berdasarkan permasalahan tersebut maka perlu dilakukannya penjadwalan produksi di PT.Sahabat. Hal ini diperlukan untuk mengurangi adanya waktu nonproduktif dari sumber daya yang tersedia. Alur proses

produksi yang dimiliki perusahaan ini termasuk klasifikasi *NP-Hard (Non-deterministic Polynomial -time Hard)*. Klasifikasi *NP-Hard* dalam masalah penjadwalan *flowshop* adalah penjadwalan yang terdiri dari lebih dari dua mesin (Laha & Sapkar,2011). Untuk memperoleh hasil optimal dalam permasalahan *NP-Hard* sebaiknya menggunakan algoritma heuristik.

Maka penelitian ini akan melakukan pengembangan algoritma heuristik yaitu algoritma Nawaz, Enscore, dan Ham (NEH) dengan pendekatan *dispatching rule* yaitu *Shortest Processing Time (SPT)* dan *Longest Processing Time (LPT)*. Selama ini belum ada yang melakukan pengembangan algoritma Nawaz, Enscore, dan Ham dengan pendekatan *dispatching rule* dalam alur proses produksi *re-entrant flowshop*. Tujuan dari penelitian ini adalah minimasi *makespan*. Dalam penelitian ini juga mempertimbangkan memisahkan waktu *setup* dengan waktu proses. Karena ketika dalam asumsinya waktu *setup* diabaikan maka tidak akan memberikan solusi yang baik dalam kondisi nyata penjadwalan proses produksi (Allahverdi, Chen & Kovalyov,2006).

2. Metode Penelitian

Dalam penelitian ini akan dikembangkan algoritma Nawaz, Enscore, dan Ham (NEH) dengan dua pendekatan *dispatching rule* yaitu *Shortest Processing Times (SPT)* dan *Longest Processing Time (LPT)*. Proses penjadwalan dilakukan dengan pendekatan maju (*forward approach*) yang dimulai dari titik waktu ke nol. Pemilihan posisi alokasi didasarkan pada langkah-langkah dalam algoritma Nawaz, Enscore, dan Ham dengan memperhatikan terpenuhinya semua urutan proses (*routing*) dan hubungan aktivitas pendahulu (*precedence*) yang ada diantara setiap operasi. Selain itu, urutan operasi *job* juga akan dipertimbangkan dalam menentukan waktu *setup*, waktu *loading*, dan waktu *unloading*. Ukuran optimasi yang digunakan penjadwalan ini adalah minimasi *makespan*.

2.1 Langkah – langkah Penelitian

Langkah – langkah yang dilakukan dalam melakukan penelitian ini adalah sebagai berikut.
 1. Tahap Pendahuluan

Pada tahap pendahuluan, langkah-langkah yang dilakukan adalah sebagai berikut:

a. Studi Lapangan

- Studi lapangan dimulai dengan melakukan wawancara secara langsung dengan pihak internal perusahaan untuk memperoleh informasi yang diperlukan secara akurat terkait dengan proses produksi, termasuk jenis produk yang di produksi, alur proses produksi dan permasalahan dalam pelaksanaannya.
- b. Studi Pustaka
Studi pustaka digunakan untuk mendapatkan informasi yang berkaitan dengan *re-entrant flowshop scheduling problem* dan algoritma Nawaz, Enscore, dan Ham (NEH) dengan tujuan minimasi *makespan*.
 - c. Identifikasi dan Perumusan Masalah
Mengidentifikasi pokok permasalahan yang muncul dari hasil pengamatan pada objek penelitian. Setelah mengidentifikasi masalah, maka merumuskan masalah yang akan dijadikan fokus pembahasan dalam penelitian ini.
 - d. Penentuan tujuan penelitian
Setelah merumuskan masalah dan mempelajari literatur yang ada, selanjutnya akan menentukan tujuan penelitian agar arah yang jelas dalam pemecahan masalah yang dihadapi PT. Sahabat
2. Tahap Pengumpulan dan Pengolahan Data
Pada tahap pengumpulan dan pengolahan data, langkah-langkah yang dilakukan adalah sebagai berikut:
 - a. Pengumpulan data
Proses pengumpulan data pada penelitian ini dilakukan dengan pengamatan, dokumentasi dan wawancara terkait dengan penelitian yang dilakukan di PT. Sahabat.
 - b. Pengolahan data
Setelah mendapatkan data yang dibutuhkan, kemudian dilakukan pengolahan data dengan metode yang relevan dengan permasalahan yang dihadapi.
 - 1) Perhitungan waktu baku operasi per stasiun kerja dengan *stopwatch time study*.
 - 2) Mengidentifikasi definisi dari notasi variabel dan parameter yang dijadwalkan.
 - 3) Perancangan dan pengembangan algoritma Nawaz, Enscore, dan Ham (NEH) dengan pendekatan dispatching rule yaitu *Shortest Processing Time (SPT)* dan *Longest Processing Time (LPT)* untuk penjadwalan produksi di PT.Sahabat.
 - 4) *Scheduling*
Proses penjadwalan (*scheduling*) merupakan pengalokasian sejumlah sumberdaya produksi untuk melakukan serangkaian aktivitas operasi pada satu periode tertentu.
 - a) *Allocation / loading*
 - b) *Sequencing*
 - c) Interpretasi hasil kedalam peta penjadwalan (*Gantt-Chart*).
 - 5) Verifikasi dan validasi model algoritma
Untuk memastikan model bekerja sesuai dengan tujuan yang diharapkan, maka dilakukan proses verifikasi dan validasi model.
 3. Tahap Analisis dan Pembahasan
Pada tahap analisis dan pembahasan, langkah-langkah yang dilakukan adalah sebagai berikut:
 - a. Melakukan pengujian performansi menggunakan *Efficiency Index* dan *Relative Error* antara penjadwalan eksisting dengan pengembangan penjadwalan algoritma Nawaz, Enscore dan Ham yang dikembangkan dengan SPT dan LPT..
 - b. Melakukan analisa perbandingan antara tahap inisialisasi awal penjadwalan dengan *Shortest Processing Time (SPT)*, *Longest Processing Time (LPT)*, dan penjadwalan eksisting dalam algoritma Nawaz, Enscore dan Ham (NEH).
 - c. Melakukan pemilihan alternatif algoritma terbaik yang memiliki nilai *makespan* minimum optimal.
 4. Kesimpulan dan Saran
Tahap penarikan kesimpulan dan saran merupakan tahap terakhir dari penelitian ini yang berisi kesimpulan yang diperoleh dari hasil analisis data yang telah dilakukan sehingga dapat menjawab tujuan penelitian.
3. **Penjadwalan *Re-entrant Flowshop***
PT. Sahabat Rubber Industries memiliki pola aliran *re-entrant flowshop*. *Re-entrant flowshop* adalah semua pekerjaan memiliki aliran produksi yang sama di setiap stasiun kerja dan membutuhkan waktu proses yang sama di setiap mesin, Namun dalam beberapa stasiun, dimungkinkan ada pekerjaan yang kembali untuk melakukan proses yang berbeda sebanyak satu kali atau lebih dari satu kali sebelum menyelesaikan pekerjaan (Graves, Meal, & Stefek,1983). Dalam penelitian ini dilakukan pengembangan penjadwalan *re-entrant flowshop* berdasarkan algoritma Nawaz,

Enscore, dan Ham dengan pendekatan *dispatching rule* baik dengan *Shortest Processing Time* (SPT) dan *Longest Processing Time* (LPT) dengan fungsi tujuan minimasi *makespan*.

3.1 Algoritma Nawaz, Ensore, dan Ham (NEH)

Algoritma Nawaz, Ensore, dan Ham dikembangkan oleh Nawaz, Ensore dan Ham (NEH) pada tahun 1983. Adapun langkah-langkah dari algoritma Nawaz, Ensore, dan Ham sebagai berikut.

- Jumlahkan waktu proses setiap *job*.
- Urutkan *job-job* menurut jumlah waktu prosesnya (w) dimulai dari yang terbesar hingga yang terkecil.
- Ambil ($w=2$) dari i yang memiliki *index* pengurutan paling atas.
- Buat w alternatif calon urutan parsial baru dan pilih yang memiliki *makespan* parsial yang terkecil. Apabila nilai *makespan* memiliki nilai yang sama maka ke Langkah 5. Jika tidak ke Langkah 6.
- Dari w alternatif calon urutan parsial sebelumnya memiliki nilai *makespan* yang sama, pilih yang memiliki nilai *mean flowtime* parsial yang lebih kecil. Apabila memiliki nilai *mean flowtime* yang sama, Maka pilihlah calon urutan parsial baru tadi secara acak.
- Calon urutan parsial baru yang terpilih menjadi urutan parsial baru.
- Coret *job-job* dari *item* i yang diambil tadi dari daftar pengurutan *job*.
- Periksa apakah $w = i$ (dimana i adalah jumlah *job item* yang ada). Jika ya, lanjutkan ke Langkah 8. Jika tidak, maka ulangi ke langkah 3 dan jumlahkan ($w=w+1$).
- Urutan parsial baru menjadi urutan final.

3.2 Metode Priority Dispatching Rule

Priority dispatching rule adalah aturan penjadwalan yang mengatur *job* dimana pada suatu antrian *job* pada suatu mesin yang harus diproses terlebih dahulu berdasarkan prioritas-prioritas tertentu. Penelitian ini menggunakan dua pendekatan *dispatching rule* yaitu *Shortest Processing Times* (SPT) dan *Longest Processing Times* (LPT).

3.2.1 Shortest Processing Times (SPT)

Metode *Shortest Processing Time* adalah metode penjadwalan yang memberikan prioritas tertinggi pada waktu penyelesaian *job* paling cepat diselesaikan. Adapun algoritma SPT sebagai berikut (Baker & Trietsch, 2009):

- Dimulai dengan urutan *job* yang tidak mengacu pada aturan SPT.
- Alokasikan dua pekerjaan sebagai *job i* dan *j*, dimana i mengikuti j dengan syarat $p_i < p_j$.
- Ubah urutan antara *job i* dan j .
- Kembali ke langkah 2. sampai urutan SPT terbentuk

3.2.2 Longest Processing Times (LPT)

Metode *Longest Processing Time* merupakan metode penjadwalan yang

memberikan prioritas tertinggi pada waktu penyelesaian *job* paling lama diselesaikan. Adapun algoritma LPT sebagai berikut (Ginting, 2009):

- Urutkan semua tugas menurut waktu proses terpanjang (*Longest Processing Time*). Pekerjaan yang memiliki waktu yang terpanjang dapatkan pada urutan pertama.
- Urutkan masing-masing tugas sesuai dengan waktu proses terpanjang pada masing-masing mesin sesuai dengan aturan waktu proses terpanjang.
- Setelah semua tugas-tugas selesai diurutkan, balikkan urutannya pada masing-masing mesin sesuai dengan aturan waktu proses terpanjang.

4. Hasil dan Pembahasan

4.1 Waktu Baku, Waktu Setup, dan Waktu Loading

Selama ini PT. Sahabat belum memiliki dokumentasi tentang waktu proses produksi masing-masing stasiun kerja. Sehingga ketika melakukan penjadwalan dari sumber daya yang tersedia pada sistem produksi. Oleh karena itu dilakukan pengambilan sampel waktu proses kerja masing-masing stasiun kerja dengan melakukan pengamatan secara langsung. Metode untuk menentukan waktu baku masing-masing stasiun kerja menggunakan *stopwatch time study*. Waktu proses yang dilakukan pengambilan sampel waktu kerja adalah stasiun giling (S2), stasiun *tubing* (S3), stasiun pengeleman (S6), stasiun *extruder* (S7), stasiun pital (S8), dan stasiun *packaging* (S9). Sedangkan stasiun *kneeder* (S1), stasiun *oven* (S4), dan stasiun rajut (S5) tidak melakukan pengambilan sampel, karena mesin melakukan proses kerja secara otomatis dan operator hanya melakukan pengawasan terhadap mesin tersebut.

Selain waktu proses yang menjadi pertimbangan dalam masukan (*input*) penjadwalan produksi, ada juga waktu setup dan waktu *loading* yang dipertimbangkan dalam masukan (*input*) penjadwalan. Waktu *setup* adalah waktu persiapan mesin yaitu menghidupkan mesin, memanaskan mesin, mempersiapkan alat-alat untuk mempersiapkan proses dari mesin yang akan digunakan. Sedangkan waktu *loading* adalah waktu pemindahan material dari satu stasiun kerja ke stasiun kerja berikutnya.

Dari hasil pengolahan data yang dilakukan dengan *stopwatch time study* yang mempertimbangkan *performace rating* dan *allowance* operator. Kemudian juga waktu proses mesin yang konstan, waktu setup mesin,

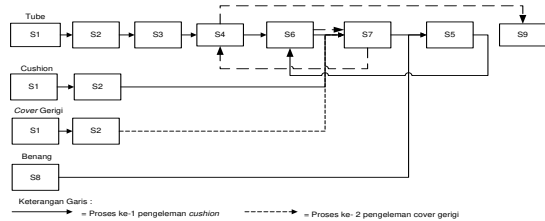
dan waktu *loading* akan direkapitulasi pada Tabel 1.

Tabel 1. Waktu Proses, Waktu *Setup*, Waktu *Loading*

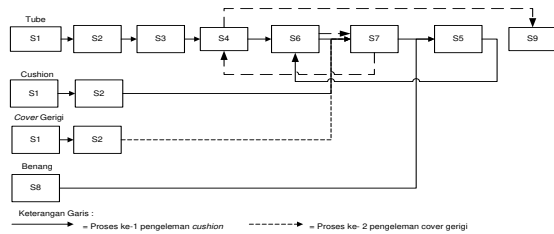
Stasiun Kerja	Waktu Proses	Waktu <i>Setup</i>	Waktu <i>Loading</i>
S1	25,00	5,00	-
S2	20,52	-	-
S3	4,91	30,00	-
S4	90,00	90,00	0,5
S5	120,00	5,00	-
S6	6,31	-	-
S7 _{cushion}	4,02	30,00	-
S7 _{cover}	5,14	30,00	-
S8	11,40	-	-
S9	7,95	-	-

4.2 Aliran Produksi

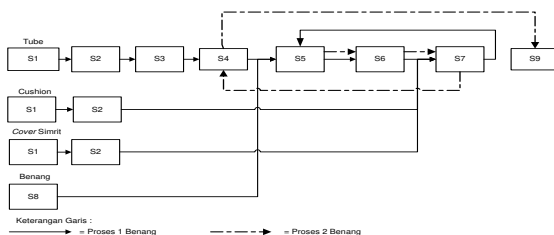
PT.Sahabat memiliki pola aliran *re-entrant flowshop*. Dalam pola aliran produksi setiap produk ada yang memiliki pola produksi yang sama dan ada juga pola aliran produksi yang berbeda. Pola aliran produksi dipengaruhi oleh spesifikasi yaitu jumlah benang dan jenis selang. Gambar 2 sampai Gambar 5 menunjukkan pola proses produksi pada empat item (CHG 1B, Simrit 2B, CHP 1B, dan HP 1B) yang dijadwalkan penelitian ini.



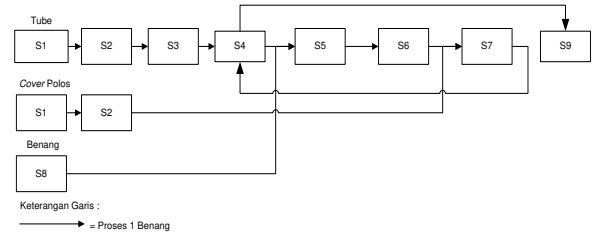
Gambar 2. Pola aliran CHG 1B



Gambar 3. Pola aliran CHP 1B



Gambar 4. Pola aliran Simrit 2B



Gambar 5. Pola aliran HP 1B

4.2.1 Skenario Pengembangan Model

Penelitian ini mengembangkan model algoritma Nawaz, Enscore, dan Ham (NEH) dengan pendekatan *priority dispatching rule* yang berorientasi pada minimasi *makespan* dalam kondisi statis, *priority dispatching rule* yang dikembangkan melalui pendekatan *Shortest Processing Times (SPT)* dan *Longest Processing Times (LPT)*. Model penjadwalan dalam penelitian ini diasumsikan bahwa semua item (*job*) harus dikerjakan tersedia pada awal periode perencanaan ($t = 0$) tanpa mempertimbangkan kedatangan item baru di tengah horizon perencanaan.

4.2.2 Notasi dan Definisi

Dalam pemodelan permasalahan penjadwalan pada sistem produksi *re-entrant flowshop* statis yang memproduksi multi item berstruktur maka diperlukan penyusunan notasi-notasi sebagai berikut.

1. *Subscript*

- i : menyatakan jenis *item* i yang akan diproduksi, sehingga $i = 1, 2, \dots, q$ dimana q menyatakan banyaknya *item* yang akan diproduksi
- j : menyatakan komponen ke- j dari c_{i0} jenis komponen untuk membuat *item* i , sehingga $j = 1, 2, \dots, c_{i0}$
- k : menyatakan urutan stasiun kerja ke k dari G_{ij} proses yang harus dilalui didalam pembuatan komponen ke- j dari *item* i , sehingga $k = 1, 2, \dots, G_{ij}$.
- l : menyatakan *level* ke- l dari e_{i0} dari *item* i , sehingga $l = 0, 1, 2, \dots, e_{i0}$.
- m : menyatakan jenis mesin ke- m dari y mesin yang tersedia, sehingga $m = 1, 2, 3, \dots, y$.
- n : menyatakan jenis mesin identik (sama) ke- n dari jumlah mesin z pada setiap tahapan proses atau kelompok mesin, sehingga $n = 1, 2, 3, \dots, z$.
- o : menyatakan urutan proses operasi dari *job* G_{ij} dari *item* i di stasiun kerja ke k , sehingga $o = 1, 2, \dots, p_{ij}$.
- ϑ : menyatakan aliran proses pengulangan (*looping*) untuk jenis mesin ke- m dari y mesin yang tersedia, sehingga $\vartheta = 0, 1, 2, 3, \dots, \vartheta_x$.
- w : menyatakan urutan total waktu proses penyelesaian *item* i ($\sum C_{ijklmno\vartheta}$), sehingga $w = 1, 2, 3, \dots, \sum C_{ijklmno\vartheta}$.

2. Parameter

- c_{io} : menyatakan jumlah jenis komponen yang diperlukan untuk membuat *item* *i* sampai selesai, dimana c_{io} tidak menyatakan jumlah *level*.
- e_{io} : menyatakan jumlah *level* yang dimiliki oleh *item* *i* didalam struktur produknya.
- p_{io} : menyatakan *item* *i* yang akan diproduksi ; $i = 1,2,\dots,q$
- p_{ij} : menyatakan komponen ke-*j* dari *item* *i*.
- n_{io} : menyatakan jumlah kuantitas permintaan *item* *i* yang akan diproduksi.
- n_{ij} : menyatakan jumlah kuantitas komponen ke-*j* dari *item* *i*.
- n_{ijk} : menyatakan jumlah komponen ke-*j* dari *item* *i* untuk melakukan proses *batch* stasiun *k*.
- A_{ijko} : jumlah komponen ke-*j* yang belum dilakukan proses di stasiun *k* pada operasi *o*
- $Z(p_{ij})$: set dari induk-induk komponen *j* dari *item* *i* berdasarkan struktur produknya.
- L_m : menyatakan jumlah kapasitas yang terdapat pada mesin ke-*m*.
- y : menyatakan jumlah jenis mesin yang tersedia didalam sistem produksi.
- z : menyatakan jumlah jenis mesin identik yang tersedia untuk setiap mesin ke-*m*.
- M_j : menyatakan kuantitas jumlah yang dapat dihasilkan oleh dalam satu bahan baku untuk mengerjakan komponen ke-*j*.
- G_{ij} : menyatakan banyak proses yang dilalui untuk pengerjaan dari keseluruhan item *i* pada komponen *j* (n_{ij}).
- $O_{iok0mno\theta}$: operasi ke-*o* untuk *item* *i* yang berada pada *level* ke-0 yang dilakukan di mesin *m* ke *n* pada stasiun ke-*k*, dimana $m = 1,2,\dots,y$; $n = 1,2,\dots,z$.
- $O_{ijklmno\theta}$: operasi ke-*o* untuk komponen *p_{ij}* yang berada pada *level* ke-*l* yang dilakukan di mesin *m* ke *n* pada stasiun ke-*k*, dimana $m = 1,2,\dots,y$; $n = 1,2,\dots,z$.
- $t_{iok0mno\theta}$: waktu proses operasi $O_{iok0mno\theta}$ untuk setiap unit p_{oj} .
- $t_{ijklmno\theta}$: waktu proses operasi $O_{ijklmno\theta}$ untuk setiap unit p_{ij} .
- $S_{ijklmno\theta}$: waktu *setup* semua mesin *m* untuk proses operasi $O_{ijklmno\theta}$
- $\mu_{ijklmno\theta}$: waktu untuk menunggu proses pengeringan/ pendinginan dari mesin ke-*m* pada operasi $O_{ijklmno\theta}$.
- $b_{ijklmno\theta}$: waktu untuk proses *loading* dan *unloading* material di stasiun *k* dari proses operasi sebelumnya ke- $O_{ijklmno\theta}$ ke proses operasi selanjutnya ke- $O_{ijklmno\theta}$.
- $A_{ijklmno\theta}$: waktu tunggu dan waktu proses untuk menyelesaikan sisa proses komponen dari stasiun kerja sebelumnya yang memiliki waktu proses lebih lama dibanding waktu proses stasiun kerja selanjutnya pada operasi ke-*o*, stasiun ke-*k*.

- $r_{ijklmno\theta}$: variabel yang menyatakan saat dimulai operasi $O_{ijklmno\theta}$.
- $C_{iok0mno\theta}$: variabel yang menyatakan saat selesainya operasi $O_{iok0mno\theta}$
- $C_{ijklmno\theta}$: variabel yang menyatakan saat selesainya operasi $O_{ijklmno\theta}$.

4.2.3 Model Penjadwalan

Dalam penelitian ini algoritma penjadwalan Nawaz, Enscore, dan Ham akan dilakukan pengembangan berdasarkan sistem yang terdapat dalam studi kasus ini. Algoritma NEH yang lebih menekankan pada urutan ke-*w* setiap *job* akan mempertimbangkan adanya proses pengulangan (θ) untuk jenis mesin *m* urutan ke-*y* dalam pengerjaan *job* tersebut. Dalam Algoritma Nawaz, Enscore, dan Ham untuk menentukan urutan (*w*) dari *job* diambilkan berdasarkan jumlah total waktu proses ($\sum C_{ijklmno\theta}$) yang memiliki nilai paling tinggi dari dua teratas yaitu urutan pertama ($w=1$) dan urutan kedua ($w=2$) dalam melakukan penyeleksian sesuai dengan peraturan *Longest Processing Time* (LPT). Maka akan diubah urutan menjadi yang memiliki nilai dua terbawah sesuai dengan peraturan *Shortest Processing Time* (SPT). Untuk penentuan operasi dari berbagai macam *item* yang harus dikerjakan dahulu di stasiun ke-*k* akan mempertimbangkan waktu proses yang lebih selesai dari stasiun sebelumnya atau *First Come First Served* (FCFS).

Setiap mesin memiliki waktu proses waktu *setup* $S_{ijklmno\theta}$, dan waktu *loading* $B_{ijklmno\theta}$ waktu proses $t_{ijklmno\theta}$ dimana waktu proses operasi sama untuk semua *n* mesin identik. Ketika terjadi pengulangan dalam proses maka waktu proses (pengulangan) $t_{ijklmno\theta}$ ke-*m* akan dijumlahkan dengan waktu proses akhir $t_{ijklmno\theta}$ ke-*m* terakhir (dengan $\theta = 0$ atau belum dilakukan pengulangan) dari proses mesin yang dilalui dengan indeks $m = 1,2,\dots,y$. Kedatangan pekerjaan bersifat statis yang artinya semua pekerjaan sudah siap tersedia dalam awal perencanaan horizon penjadwalan dan tidak adanya penyisipan *item* atau penambahan order *item* dalam proses produksi selama perencanaan horizon penjadwalan.

Dalam beberapa mesin pengerjaan *job* dapat dilakukan secara *batch* atau dikerjakan secara bersamaan. Mesin yang dapat melakukan pekerjaan secara bersamaan adalah mesin *kneeder*, mesin giling dan mesin *oven*. Khusus untuk proses *oven* dan proses rajut diperlukan menunggu pemindahan material ($b_{ijklmno\theta}$)

3. Variabel

- $r_{iok0mno\theta}$: variabel yang menyatakan saat dimulai operasi $O_{iok0mno\theta}$

untuk melakukan proses secara *batch*. Dimana dalam melakukan proses *batch* maka n_{ijk} akan dikalikan dengan waktu proses pemindahan material ($b_{ijklmno\theta}$). Kemudian untuk kasus menunggu pengeringan atau pendinginan ($\mu_{ijklmno\theta}$) setelah dikerjakan pada proses *kneeder* dan proses pengeleman diperlukan pemisahan dalam visualisasi dalam bentuk *Gantt Chart* namun untuk perhitungan waktu proses menunggu tersebut digabung dengan proses baik di *kneeder* dan pengeleman. Dalam kondisi tertentu, ada beberapa stasiun ke- k pada operasi ke- o (*successor*) yang harus menunggu penyelesaian sisa komponen (A_{ijko}) dari stasiun ($k-1$) pada operasi ($o-1$) (*predecessor*), kondisi ini terjadi apabila pada kapasitas pada stasiun ke- k sudah penuh. Oleh karena itu, total penyelesaian stasiun (k) akan mempertimbangkan waktu tunggu dari sisa komponen tersebut. $A_{ijklmno\theta}$ adalah waktu tunggu untuk menyelesaikan proses sisa komponen pada proses *batch* di pada operasi ke- o , stasiun ke- k .

Secara keseluruhan, proses produksi PT.Sahabat Rubber Industries memiliki sembilan stasiun kerja ($k = 1,2,\dots,9$). Dimana setiap proses produk setiap *item* memiliki alur proses produksi yang berbeda walaupun ada beberapa *item* yang memiliki alur proses produksi sama. Dari setiap produk memiliki tahapan operasi yang berbeda-beda sesuai dengan spesifikasi jumlah benang dan jenis dari *item* tersebut. Waktu *setup* bersifat *independent* terhadap urutan *job* yang diproses, dimana waktu *setup* tidak akan digabung dengan waktu proses dan waktu *setup* semua komponen adalah sama di jenis mesin ke- m . Waktu *setup* dalam penelitian ini akan dialokasikan sebelum dimulainya aktivitas selanjutnya (*successor*) akan dialokasikan. Waktu *loading* dan *unloading* akan dipertimbangkan dalam penjadwalan ini yang dialokasikan antara sebelum memulai aktivitas *successor* dengan setelah aktivitas *predecessor* selesai dikerjakan. Setiap *job* dalam penjadwalan ini dikerjakan secara berkaitan (*dependent*), artinya waktu mulai proses di jenis mesin ke- m akan bergantung pada waktu penyelesaian di jenis mesin- m . Dalam penjadwalan sistem ini tidak adanya interupsi operasi di tengah perencanaan penjadwalan, artinya semua *job* mulai dari titik nol dan tidak adanya *job item* sisipan.

Permasalahan penjadwalan dapat dideskripsikan ke dalam fungsi *triplet* $\alpha|\beta|\gamma$ (Pinedo,2011). Fungsi α menjelaskan aliran mesin yang terdapat pada studi kasus, Fungsi β menjelaskan karakteristik proses dan kendala-kendala yang dapat ditemui dalam aliran produksi, dan Fungsi γ menjelaskan fungsi tujuan dalam melakukan penjadwalan. Fungsi triplet dalam permasalahan dalam studi kasus di PT.Sahabat Rubber Industries yaitu $RFS|prec, s_i, batch(b)|\sum_{i=1}^r C_i$ dimana fungsi tersebut menjelaskan bahwa sistem yang dijadwalkan mengadopsi *re-entrant flowshop system flowshop*. Dimana sistem tersebut memiliki karakteristik proses dan batasan sebagai berikut : *precedence*, *independent setup times*, dan *batch processing*. Kemudian untuk fungsi tujuan dalam penelitian ini untuk meminimasi *makespan*.

4.2.4 Asumsi Model

Asumsi yang digunakan dalam pengembangan algoritma adalah sebagai berikut:

1. Keseluruhan mesin siap memproses *job* di titik waktu ke nol.
2. Waktu *setup* bersifat *independent* (setiap komponen memiliki waktu *setup* mesin yang sama di masing-masing stasiun kerja).
3. Ketika di titik tertentu ada mesin paralel di suatu tahap operasi hanya dapat memproses satu *job*.
4. Ketika terjadi pengulangan (*looping*) proses dalam suatu sistem ke suatu mesin, maka proses dianggap berbeda.
5. Setiap *job* dapat mulai diproses di mesin yang tersedia (mesin tersebut tidak sedang melakukan *job*).
6. Setiap *job* yang sedang diproses di suatu mesin tidak boleh diinterupsi.
7. Urutan (*Sequencing*) penjadwalan *job* di awal setiap tahap masing-masing proses tidak harus sama.
8. Waktu setiap proses adalah deterministik.

4.2.5 Pengembangan Model Algoritma Nawaz, Enscore, dan Ham

1. Pengembangan algoritma Nawaz, Enscore, dan Ham (NEH)
 - a. Jumlahkan waktu proses ($t_{ijklmno\theta}$) awal sampai selesai ($C_{ijklmno\theta}$) setiap *job* (i) dari seluruh *item* yang dijadwalkan.
 - b. Urutkan *job-job* menurut jumlah waktu proses ($C_{ijklmno\theta}$) dimulai dari yang terkecil hingga yang terbesar (SPT) atau dimulai dari terbesar hingga yang

terkecil (LPT). Hasil pengurutan mulai dari Urutan pertama adalah *job* dari *i* yang memiliki nilai $\sum C_{ijklmno\theta}$ sesuai dengan aturan dari SPT atau LPT.

- c. Ambil ($w=2$) dari *i* yang memiliki *index* pengurutan paling atas .
- d. Buat *w* alternatif calon urutan parsial baru dan pilih yang memiliki *makespan* parsial yang terkecil, Apabila nilai *makespan* memiliki nilai yang sama maka ke Langkah 5. Jika tidak ke Langkah 6.
- e. Dari *w* alternatif calon urutan parsial sebelumnya memiliki nilai *makespan* yang sama, pilih yang memiliki nilai *mean flowtime* parsial yang lebih kecil. Apabila memiliki nilai *mean flowtime* yang sama, Maka pilihlah calon urutan parsial baru tadi secara acak.
- f. Calon urutan parsial baru yang terpilih menjadi urutan parsial baru.
- g. Coret *job-job* dari *item* *i* yang diambil tadi dari daftar pengurutan *job*.
- h. Periksa apakah $w = i$ (dimana *i* adalah jumlah *job item* yang ada). Jika ya, lanjutkan ke Langkah 8. Jika tidak, maka ulangi ke langkah 3 dan jumlahkan ($w=w+1$).
- i. Urutan parsial baru menjadi urutan final.

2. Allocation Model

- a. Tentukan item (*i*) yang akan diproduksi dengan indeks $i = 1,2,3,\dots, q$.
- b. Buat struktur dari item *i* bersangkutan dan tentukan level dari produk akhir hingga komponen-komponen penyusunnya
- c. Tentukan subscript *j,k,l,m,n,o*, dan θ . untuk semua *item i*
- d. Untuk setiap produk akhir (p_{io}) tentukan n_{io} , $O_{iokom.}$, $t_{iokom.}$, dan $C_{iokom.}$. Ketika adanya tahapan pengulangan (θ) maka pada posisi *o* terakhir akan bertambah satu ($o+1$) dengan nilai *ready time* sesuai dengan waktu penyelesaian stasiun terakhir (*k*) atau operasi terakhir (*o*) hingga pada proses akhir.
- e. Untuk setiap komponen (p_{ij}) penyusun produk akhir (p_{io}), tentukan induk komponen tersebut hingga menjadi *item* atau produk akhir dan banyaknya komponen yang dibutuhkan dalam menyusun satu unit induk langsung (n_{ij}). Untuk menentukan n_{ij} yang harus dibuat dapat dirumuskan:

$$n_{ij} = \frac{n_{io}}{M_j} \quad (\text{pers.1})$$

- f. Untuk menentukan jumlah proses yang dilakukan dalam mengerjakan seluruh permintaan sesuai dengan kapasitas ditentukan dengan rumus sebagai berikut:

$$G_{ij} = \frac{n_{io}}{L_m \times z} \quad (\text{pers. 2})$$

- g. Untuk menentukan jumlah yang belum dilakukan pemrosesan (A_{ij}) Dapat diketahui dengan rumus sebagai berikut:

$$A_{ij} = \left(\frac{n_{io} - (G_{ij(k-1)} \times L_m^{(k-1)} \times z^{(k-1)})}{L_m} \right) \quad (\text{pers.3})$$

- h. Ketika sudah mengetahui waktu penyelesaian *item* ($C_{iokom.}$) maka urutkan (*w*) ambil dua nilai ($C_{iokom.}$) terkecil (SPT) atau terbesar (LPT). Setiap urutan akan bertambah satu ($w+1$). Urutkan *item-item* tersebut dari urutan paling awal hingga paling akhir pada urutan. Urutan *w* akan berhenti hingga sesuai dengan jumlah *item* yang akan dijadwalkan ($w=i$).

3. Sequencing Model

Sequencing Algorithm

0. Tentukan *p* (p_{io}) yaitu *set* dari *item-item* yang akan diproduksi, dan nyatakan *item* sebagai *job* dengan indeks $i = 1,2,\dots,r$.
1. Untuk masing-masing mesin, tentukan *O* ($O_{ijklmno\theta}$); $\forall i, j, l, m, \theta; k = 1$ yaitu *set* operasi-operasi dari komponen-komponen p_{ij} pada mesin yang melakukan proses terhadap komponen tersebut. Indeks mesin $m = 1,2,3,\dots,y$.
2. Urutan penempatan pekerjaan *item* paling awal menggunakan pendekatan (Sub-Algoritma Nawaz, Enscore dan Ham pendekatan SPT atau Sub-Algoritma Nawaz, Enscore, dan Ham pendekatan LPT),

Oleh karena itu , diperlukan perhitungan waktu proses terlebih dahulu dari masing-masing *item* yang akan dijadwalkan (Langkah 3).

3. Urutkan operasi-operasi sesuai dengan urutan aliran proses produksi ($O_{ijklmno\theta}$) mulai dari mesin 1 ($O_{11111110}$). Penempatan pekerjaan dilakukan pada salah satu mesin yang melakukan proses pengerjaan dari komponen (p_{11}) pertama dari satu *item* pertama yang dialokasikan..

4. Apakah pada saat *ready time* tertentu ($r_{ijklmno\theta} = 0$) terdapat operasi yang sudah dialokasikan?

- a. Jika iya, maka lanjutkan ke Langkah 5.
- b. Jika tidak, maka tidak, maka alokasikan operasi (*o*), beri subscript $w = 1$, dan hitung dengan model matematis :

$$C_{mn(1)} = S_{klmno\theta(1)} + t_{klmno\theta(1)} + r_{klmno\theta(1)} + b_{klmno\theta(1)} \quad (\text{pers.4})$$

Namun karena dalam penjadwalan ini bersifat status maka pada awal diasumsikan nilai *ready time* ($r_{ijklmno\theta}$) = 0 dan waktu *loading* ($b_{ijklmno\theta}$) = 0, maka model rumus matematisnya adalah:

$$C_{mn(1)} = S_{klmno\theta(1)} + t_{klmno\theta(1)} \quad (\text{pers.5})$$

Untuk penentuan *ready time* ($r_{ijklmno\theta}$) dari setiap item (*i*) ditentukan berdasarkan kedatangan *First Come First Served* (FCFS) dan utilitas mesin sebelum dan sesudahnya.

$$\sum C_{ijklmno(\theta-1)} = r_{klmno\theta}; \forall m_{(\theta-1)} = m_{(\theta)} \quad (\text{pers.6})$$

5. Apakah waktu *setup* di stasiun *k* yang perlu dialokasikan pada operasi ke-*o*?

- a. Jika iya, maka waktu *setup* akan diurutkan sebelum operasi ke-*o* selanjutnya. (*successor*) dimulai maka model matematisnya adalah:

$$Z = \sum C_{(klmno\theta-1)} - S_{ijklmno\theta} \quad (\text{pers.7})$$

Keterangan: *Z* adalah waktu urutan dimulainya *setup* terhadap penempatannya dalam *Gantt Chart*. Maka *ready time* dari aktivitas setelahnya akan disesuaikan waktu selesai dari waktu *setup*.

Namun dalam kondisi tertentu, apabila kondisinya waktu paling akhir dari *predecessor* adalah operasi *cover* atau *cushion* dan *successor*nya adalah *cover* atau *cushion* Maka, untuk waktu *ready time* ($r_{i37lmno\theta}$) atau ($r_{i27lmno\theta}$) dari operasi *cover* ($O_{i37lmno\theta}$) atau operasi *cushion* ($O_{i27lmno\theta}$) waktu *setup* akan dijumlahkan dengan waktu operasi *cushion* terakhir ($\sum C_{i27lmno\theta}$) atau operasi *cover* terakhir ($\sum C_{i37lmno\theta}$) karena memerlukan penggantian komponen pada mesin untuk melakukan proses tersebut. Maka model matematika proses nya adalah sebagai berikut.

$$Z = r_{klmno\theta(1)} + S_{klmno\theta(1)} \quad (\text{pers.8})$$

- b. Jika tidak, lanjutkan ke langkah 6.

6. Apakah bentuk proses adalah single processing (100 m)?

- a. Jika ya, maka bentuk model matematis untuk perhitungan waktu proses adalah.

$$C_{ijklmno\theta} = t_{ijklmno\theta} + b_{ijklmno\theta} + r_{ijklmno\theta} \quad (\text{pers.9})$$

Kemudian untuk melakukan perhitungan total waktu proses dengan model matematis sebagai berikut.

$$\sum C_{ijklmno\theta} = t_{ijklmno\theta} + t_{ijklmno\theta} \times (G_{ij}) + b_{ijklmno\theta} \quad (\text{pers.10})$$

Untuk kondisi tertentu perlu waktu tunggu di stasiun ke-*k* operasi ke-*o* sebab komponen pada stasiun sebelumnya (*k-1*) belum selesai diproses akibat waktu proses mesin sebelumnya lebih lama dibanding stasiun setelahnya. Maka total waktu dilakukan pemodelan matematis sebagai berikut:

$$\sum C_{ijklmno\theta} = (\sum C_{ijklmno\theta-1}) + (t_{ijklmno\theta} \times (A_{ij})) + b_{ijklmno\theta} \quad (\text{pers.11})$$

- b. Jika tidak, maka lanjutkan ke langkah 7.

7. Apakah bentuk proses adalah *batch* (melakukan proses secara bersamaan)?

- a. Jika ya, apabila terjadi pada stasiun kerja *batch* maka akan dilakukan perhitungan dengan model matematis untuk *ready time*

$$r_{ijklmno\vartheta} = t_{ijklmno\vartheta-1} \times \left(\frac{\text{Kapasitas mesin } m}{\text{kapasitas mesin } (m-1)} \right) + r_{ijklmno\vartheta-1} + b_{ijklmno\vartheta} \quad (\text{pers.12})$$

Untuk perhitungan waktu penyelesaian proses maka model matematisnya:

$$C_{ijklmno\vartheta} = t_{ijklmno\vartheta} + r_{ijklmno\vartheta} + (n_{ijk} \times b_{ijklmno\vartheta} \times 2) \quad (\text{pers.13})$$

Namun apabila tidak ada proses *loading* maka ($b_{ijklmno\vartheta}$) dianggap nol. Setelah dilakukan perhitungan waktu proses. Maka akan dilakukan perhitungan total waktu penyelesaian dengan model matematis sebagai berikut:

$$\sum C_{ijklmno\vartheta} = t_{ijklmno\vartheta} \times (G_{ij}) + r_{ijklmno\vartheta} + (n_{ijk} \times (b_{ijklmno\vartheta} \times 2 \times H_{ij})) \quad (\text{pers.14})$$

Untuk kondisi tertentu perlu waktu tunggu di stasiun ke-*k* operasi ke-*o* sebab komponen pada stasiun sebelumnya (*k-1*) belum selesai diproses akibat waktu proses mesin sebelumnya lebih lama dibanding stasiun setelahnya. Maka total waktu penyelesaian dilakukan pemodelan matematis:

$$\sum C_{ijklmno\vartheta} = (\sum C_{ijklmno\vartheta-1}) + (t_{ijklmno\vartheta} \times (A_{ij})) + b_{ijklmno\vartheta} \quad (\text{pers.15})$$

- b. Jika tidak, maka lanjutkan ke langkah 8.
 8. Periksa apakah ada proses yang melakukan pengeringan atau pendinginan $\mu_{ijklmno\vartheta}$ pada proses $O_{ijklmno\vartheta}$ = ?

- a. Jika ya, maka akan dilakukan perhitungan dengan model matematis waktu penyelesaian.

$$C_{ijklmno\vartheta} = t_{ijklmno\vartheta} + r_{ijklmno\vartheta} + b_{ijlmo\vartheta} \text{ atau } C_{ijklmno\vartheta} = \mu_{ijklmno\vartheta} + r_{ijklmno\vartheta} + b_{ijlmo\vartheta} \quad (\text{pers.16})$$

Untuk melakukan perhitungan total waktu penyelesaian maka model matematis sebagai berikut.

$$\sum C_{ijklmno\vartheta} = t_{ijklmno\vartheta} \times (G_{ij}) + r_{ijklmno\vartheta} + b_{ijlmo\vartheta} \quad (\text{pers.17})$$

Proses pengeringan dinyatakan dalam proses operasi $O_{(ijklmno\vartheta+1)}$ setelah proses masak maupun pengeleman.

- b. Jika tidak, maka lanjutkan ke langkah 9.
 9. Periksa apakah ada proses *looping* pada operasi ke *k* (pengulangan proses)?
 a. Jika ya
 b. Jika tidak, maka lanjutkan ke Langkah 11.
 10. Periksa apakah pada proses *looping* tersebut, sudah dilakukan alokasi kepada stasiun kerja ke- *k* pada mesin ke-*m*?

- a. Jika ya, lanjut ke Langkah 11.
 b. Jika tidak, maka alokasikan dan periksa dahulu waktu *ready time* dari *looping* mesin tersebut yang dapat dihitung dengan model matematis:

$$r_{(\vartheta+1)ijklmno} = \sum C_{(\vartheta)ijklmno} ; \forall i, j, k, l, m, n, o \quad (\text{pers.18})$$

Dimana waktu dimulainya proses *looping* dapat dilanjutkan ketika penyelesaian proses operasi *predecessors* ($\sum C_{ijklmno\vartheta}$) selesai. Operasi ini harus menunggu terlebih dahulu untuk melakukan proses.

Untuk perhitungan waktu proses dari *single processing* dapat ke Langkah 6 dan *batch processing* dapat ke Langkah 7.

11. Periksa apakah ada mesin ke-*m* dalam mesin *n* identik yang masih tersedia?

- a. Jika mesin tersedia maka alokasikan jadwal operasi *o* pada salah satu mesin *m* dan *n* identik, masukkan operasi *o* ke dalam *set* operasi ($O_{ijklmno\vartheta}$) dan lanjutkan ke Langkah 12.
 b. Jika tidak tersedia, maka lanjutkan ke Langkah 15.

12. Jadwalkan operasi *o+1* dibelakang mengikuti operasi yang terjadwal pada mesin *m* ke-*n* tersebut, dengan syarat: $\min(C_{mn} \forall m, n)$.

Dimana:

$$\sum C_{m(n=1)} = s_{klmno\vartheta(1)} + t_{klmno\vartheta(1)} \times G_{ij} + b_{klmno\vartheta(1)}$$

$$\sum C_{m(n=2)} = s_{klmno\vartheta(2)} + t_{klmno\vartheta(2)} \times G_{ij} + b_{klmno\vartheta(2)}$$

$$\sum C_{m(n=z)} = s_{klmno\vartheta(z)} + t_{klmno\vartheta(z)} \times G_{ij}$$

$$+ b_{klmno\vartheta(z)} \quad (\text{pers.19})$$

13. Set $m = m+1$ dan set $x = x+1$?

- a. Jika $m < y$ dan $x < O$ kembali ke Langkah 3

- b. Jika $m \geq y$ dan $x \geq O$ lanjutkan ke Langkah 14.

Keterangan :

x = Urutan operasi yang akan dijadwalkan pada mesin *m* ke *n* masukkan semua operasi yang telah terjadwal ke dalam *set* operasi ($O_{ijklmno\vartheta}$).

14. Gambarkan ke *Gantt Chart* untuk jadwal akhir sesuai dengan modifikasi algoritma Nawaz, Enscore, dan Ham yang terbentuk dengan memperhatikan susunan *item* serta ketersediaan mesin. Selesai.

4.2.6 Verifikasi dan Validasi Model

Data yang digunakan dalam verifikasi dan validasi model terdiri dari empat item produk. Dimana di setiap *stage* dapat memiliki lebih dari satu jenis mesin identik. Kondisi ini, dapat digunakan untuk menjamin bahwa algoritma yang dikembangkan dapat menyelesaikan permasalahan di PT. Sahabat atau pada lingkungan *re-entrant flowshop* lainnya yang memproduksi multi item berstruktur multi *level* dengan mempertimbangkan waktu *setup*, waktu *loading* dan waktu *unloading*. Berikut ini ditampilkan contoh verifikasi dari algoritma *sequencing* yang telah dirancang dalam penelitian ini

Berikut ini akan disajikan contoh dari langkah-langkah untuk melakukan perhitungan waktu makespan dari item Cushion Hitam Gerigi 1B berdasarkan algoritma *sequencing* yang telah dirancang.

1. Langkah 1: ($O_{ijklmno\vartheta}$) ; $\forall i, j, l, m, n, o, \vartheta; k = 1$.
2. Langkah 2: Untuk melakukan pengurutan menggunakan algoritma NEH, maka ke Langkah 3. Untuk menghitung *makespan* dari awal sampai akhir.
3. Langkah 3: Urutan dimulai dari stasiun kneeder O ($O_{111111110}$).
4. Langkah 4: Tidak, karena pada stasiun *kneeder* tidak memiliki aktivitas sebelumnya (*precedence*) maka *ready time* ($r_{ijklmno\vartheta}$) adalah nol.
5. Langkah 5: Ya, ada waktu *setup* pada stasiun *kneeder* yang terdapat di proses 1 (*o-1*), karena pada operasi ke-1 maka waktu *setup*nya dijumlahkan dengan waktu proses dan waktu *loading* (perhitungan model matematis ada pada Langkah 7).
6. Langkah 6: Tidak ada *single processing*.
7. Langkah 7: Ya, stasiun *kneeder* melakukan proses *batch*:

$$C_{111111110} = t_{ijklmno\vartheta} + r_{ijklmno\vartheta} + b_{ijlmo\vartheta} \times s_{ijklmno\vartheta}$$

$$C_{111111110} = 25,00 + 0 + 0 + 5,00$$

$$C_{111111110} = 30,00$$

Untuk perhitungan total waktu proses pada operasi pertama di stasiun *kneeder* adalah sebagai berikut:

$$\sum C_{ijklmno\vartheta} = t_{ijklmno\vartheta} (G_{ij}) + r_{ijklmno\vartheta} + b_{ijklmno\vartheta} + s_{ijklmno\vartheta}$$

$$\sum C_{11111110} = 25,00 (7) + 0 + 0 + 5,00$$

$$\sum C_{1111110} = 180$$

8. Langkah 8: Tidak memiliki proses pengeringan.
9. Langkah 9: Tidak memiliki proses *looping*, maka ke Langkah 11.
10. Langkah 10: (Lanjut ke langkah 11).
11. Langkah 11: Tidak tersedia mesin identik karena mesin *kneeder* hanya ada 1 buah, maka ke Langkah 13.
12. Langkah 12: (Lanjut ke langkah 13).
13. Langkah 13: *Set* $m = 0, m = m+1$; $m(1) = y (9)$, Jumlah operasi adalah 21. maka nilai $x = 0, set x = x + 1$; $x = 1$, maka $x (1) < O (21)$ dan $m (1) < y (9)$, maka kembali ke Langkah 3 , Lakukan setiap langkah sampai O ke-21 maka $o (21) = x (21)$. Langkah akan terhenti ketika semua operasi sudah terpenuhi.

Tabel 2 menunjukkan seluruh rekapitulasi waktu proses berdasarkan algoritma *sequencing* dari empat *item*.

Tabel 2. Jumlah Waktu Proses Tiap Item

Item	Total Waktu Proses (menit)
CHG 1B	673,43
Simrit 2B	880,40
HP 1B	1612,84
CHP 1B	1836,71

4.3 Model Algoritma Nawaz, Enscore, dan Ham dengan SPT

Model algoritma Nawaz, Enscore, dan Ham dengan SPT (*Shortest Processing Time*) dimulai dari iterasi 0 sampai iterasi akhir.

a. Iterasi 0 ($w = 2$)

1. Urutkan *job-job* dari yang memiliki waktu total proses terkecil sampai terbesar. Tabel 3 menunjukkan pengurutan dengan SPT.

Tabel 3. Pengurutan Shortest Processing Times (SPT)

Item	Total Waktu Proses (menit)	Urutan
CHG 1B	673,43	1
Simrit 2B	880,40	2
HP 1B	1612,84	3
CHP 1B	1836,71	4

2. Ambil ($w=2$) dari i yang memiliki *index* pengurutan paling atas. Tabel 4 akan menunjukkan i yang memiliki *index* paling teratas.

Tabel 4. Index Teratas

Item	Total Waktu Proses (menit)	Urutan
CHG 1B	673,43	1
Simrit 2B	880,40	2

3. Buat dua calon urutan parsial baru pilih yang memiliki *makespan* parsial yang terkecil. Ada 2 calon urutan parsial yang akan diseleksi yaitu 1-2 dan 2-1.
4. Dari dua calon alternatif yang terbentuk, maka terpilih calon urutan parsial 1-2 karena memiliki nilai *makespan* lebih kecil. Tabel 5 menunjukkan hasil perhitungan calon urutan parsial 1-2 dan 2-1.

Tabel 5. Calon Urutan Parsial Sementara ($w = 2$)

Calon urutan parsial	Total Waktu Proses (menit)
1-2	1086,21
2-1	1126,40

5. Urutan Parsial sementara 2-1

6. Coret *job* dari *item* 1 dan *job* dari *item* 2 dari daftar tabel.
7. $w (2) \neq i (4)$, maka ulangi ke Langkah 3 , dan jumlahkan ($w = 2+ 1= 3$). (Langkah 3).

b. Iterasi 1 ($w = 3$)

3. Maka saat ini nilai w adalah 3. Maka ada tiga kemungkinan calon urutan parsial baru yaitu 1-2-3, 1-3-2, dan 3-1-2.
4. Dari tiga calon alternatif, maka terpilih calon urutan parsial 3-1-2 karena memiliki nilai *makespan* lebih kecil dibanding calon urutan parsial 1-3-2 dan 1-2-3. Tabel 6 menunjukkan hasil rekapitulasi *makespan* calon urutan parsial 3-1-2, 1-3-2, 1-2-3.

Tabel 6. Calon Urutan Parsial Sementara ($w = 2$)

Calon urutan parsial	Total Waktu Proses (menit)
3-1-2	2066,16
1-3-2	2067,77
1-2-3	2140,11

5. Urutan parsial sementara 3-1-2.
6. Coret urutan 1, urutan 2, dan urutan 3 dari daftar tabel.
7. $w (3) \neq i (4)$, maka ulangi ke Langkah 3 , dan jumlahkan ($w = 3 + 1= 4$). (Langkah 3).

c. Iterasi 2 ($w = 4$)

3. Maka saat ini nilai w adalah 4. Maka ada empat kemungkinan calon urutan parsial baru yaitu 4-3-1-2, 3-4-1-2, 3-1-4-2, dan 3-1-2-4.
4. Dari empat calon alternatif, maka terpilih calon urutan parsial 3-4-1-2 karena memiliki nilai *makespan* lebih kecil dibanding calon urutan parsial 3-4-1-2, 3-1-4-2, dan 3-1-2-4. Tabel 7 menunjukkan hasil rekapitulasi *makespan* calon urutan parsial 4-3-1-2, 3-4-1-2, 3-1-4-2, dan 3-1-2-4.

Tabel 7. Calon Urutan Parsial Sementara ($w = 4$)

Calon urutan parsial	Total Waktu Proses (menit)
4-3-1-2	3216,04
3-4-1-2	3146,16
3-1-4-2	3219,50
3-1-2-4	3321,69

5. Urutan parsial sementara 3-4-1-2.
6. Coret urutan 1, urutan 2, urutan 3, dan urutan 4 dari daftar tabel.
7. $w (4)=i (4)$ maka ke langkah 8.
8. Urutan final berdasarkan nilai *makespan* minimum adalah 3-4-1-2. Dengan nilai *makespan* sebesar 3146,16 menit.

4.4 Model Algoritma Nawaz, Enscore, dan Ham dengan LPT

Model algoritma Nawaz, Enscore, dan Ham dengan LPT (*Longest Processing Time*) dimulai dari iterasi 0 sampai iterasi akhir.

a. Iterasi 0 ($w = 2$)

1. Urutkan *job-job* dari yang memiliki waktu total proses terbesar sampai terkecil. Tabel 8 menunjukkan pengurutan dengan LPT.

Tabel 8. Pengurutan Longest Processing Times (LPT)

Item	Total Waktu Proses (menit)	Urutan
CHG 1B	1836,71	1
Simrit 2B	1612,84	2
HP 1B	880,40	3
CHP 1B	673,43	4

1. Ambil ($w=2$) dari i yang memiliki *index* pengurutan paling atas. Tabel 9 menunjukkan i yang memiliki *index* paling teratas.

Tabel 9. *Index Teratas*

Item	Total Waktu Proses (menit)	Urutan
CHG 1B	1836,71	1
Simrit 2B	1612,84	2

2. Buat dua calon urutan parsial baru pilih yang memiliki *makespan* parsial yang terkecil. Ada 2 calon urutan parsial yang akan diseleksi yaitu 1-2 dan 2-1.
3. Dari dua calon alternatif yang terbentuk, maka terpilih calon urutan parsial 2-1 karena memiliki nilai *makespan* lebih kecil. Tabel 10 menunjukkan hasil perhitungan calon urutan parsial 1-2 dan 2-1.

Tabel 10. Calon Urutan Parsial Sementara ($w = 2$)

Calon urutan parsial	Total Waktu Proses (menit)
1-2	2767,86
2-1	2721,69

4. Urutan Parsial sementara 2-1
5. Coret *job* dari *item* 1 dan *job* dari *item* 2 dari daftar tabel.
6. $w(2) \neq i(4)$, maka ulangi ke Langkah 3, dan jumlahkan ($w = 2 + 1 = 3$). (Langkah 3).
- b. Iterasi 1 ($w = 3$)**
3. Maka saat ini nilai w adalah 3. Maka ada tiga kemungkinan calon urutan parsial baru yaitu 3-2-1, 2-3-1, dan 2-1-3.
4. Dari tiga calon alternatif, maka terpilih calon urutan parsial 2-1-3 karena memiliki nilai *makespan* lebih kecil dibanding calon urutan parsial 1-3-2 dan 1-2-3. Tabel 11 menunjukkan hasil rekapitulasi *makespan* calon urutan parsial 3-1-2, 2-3-1, 1-2-3.

Tabel 11. Calon Urutan Parsial Sementara ($w = 3$)

Calon urutan parsial	Total Waktu Proses (menit)
3-1-2	3022,59
2-3-1	3121,62
2-1-3	2993,09

5. Urutan parsial sementara 2-1-3.
6. Coret urutan 1, urutan 2, dan urutan 3 dari daftar tabel.
7. $w(3) \neq i(4)$, maka ulangi ke Langkah 3, dan jumlahkan ($w = 3 + 1 = 4$). (Langkah 3).
- c. Iterasi 2 ($w = 4$)**
3. Maka saat ini nilai w adalah 4. Maka ada empat kemungkinan calon urutan parsial baru yaitu 4-2-1-3, 2-4-1-3, 2-1-4-3, dan 2-1-3-4. Berikut ini ditunjukkan pengurutan calon urutan parsial pada Tabel 4.40.
4. Dari empat calon alternatif, maka terpilih calon urutan parsial 2-1-4-3 karena memiliki nilai *makespan* lebih kecil dibanding calon urutan parsial 4-2-1-3, 2-4-1-3, dan 2-1-3-4. Tabel 12 menunjukkan hasil rekapitulasi *makespan* calon urutan parsial 4-2-1-3, 2-4-1-3, 2-1-4-3, dan 2-1-3-4.

Tabel 12. Calon Urutan Parsial Sementara ($w = 4$)

Calon urutan parsial	Total Waktu Proses (menit)
4-2-1-3	3260,49
2-4-1-3	3219,50
2-1-4-3	3146,16
2-1-3-4	3244,09

5. Urutan parsial sementara 2-1-4-3.
6. Coret urutan 1, urutan 2, urutan 3, dan urutan 4 dari daftar tabel.
7. $w(4) = i(4)$ maka ke langkah 8.
8. Urutan final berdasarkan nilai *makespan* minimum adalah 2-1-4-3. Dengan nilai *makespan* sebesar 3146,16 menit.

4.5 Uji Performansi Penjadwalan

Dalam penjadwalan dengan menggunakan algoritma Nawaz, Enscore, dan Ham (NEH) baik menggunakan LPT dan SPT menghasilkan *makespan* sebesar 3146,16 menit dengan urutan pekerjaan Selang Simrit 2B- Selang CHG 1B – Selang HP 1B – Selang CHP 1B. Kemudian dibandingkan dengan proses alokasi penjadwalan yang dilakukan perusahaan dengan jumlah permintaan terbesar dengan urutan Selang CHG 1B- Selang Simrit 2B- Selang HP 1B- Selang CHP 1B dengan *makespan* 4602,02 menit. Maka dapat dilakukan pengujian *Efficiency Index* sebagai berikut.

$$EI = \frac{F_{\max \text{ metode Perusahaan}}}{F_{\max \text{ heuristic}}} = \frac{4602,02}{3146,16} = 1,46$$

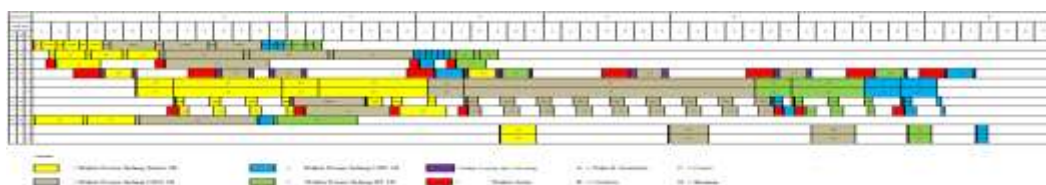
Nilai EI (1,46) lebih besar daripada 1. Menunjukkan bahwa penjadwalan NEH baik dengan pendekatan SPT dan LPT lebih baik dibanding milik perusahaan. Kemudian untuk selisih dari *makespan* yang dibuat dapat dilakukan dengan menghitung *Relative Error* (RE) sebagai berikut.

$$RE = \frac{F_{\max \text{ metode perusahaan}} - F_{\max \text{ heuristic}}}{F_{\max \text{ heuristic}}} \times 100 \%$$

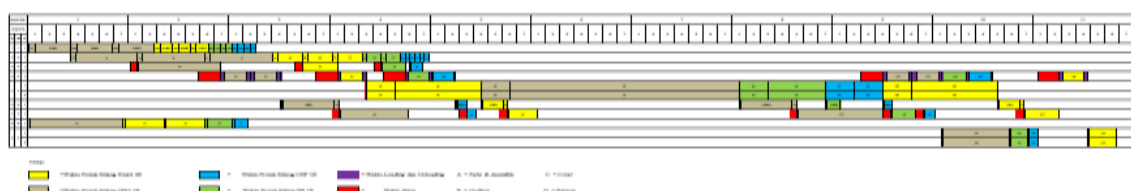
$$RE = \frac{4602,02 - 3146,16}{3146,16} \times 100 \% = 46,27$$

4.6 Analisa dan Pembahasan

Dalam Gambar 6 dan Gambar 7 akan ditampilkan *Gantt Chart* baik yang dirancang dalam penelitian ini dengan *Gantt Chart* yang dimiliki perusahaan.



Gambar 6. *Gantt Chart* Algoritma NEH SPT dan LPT



Gambar 7. *Gantt Chart* Eksisting

Dari hasil dalam *Gantt Chart* menunjukkan perbandingan antara algoritma NEH pendekatan SPT dan LPT dengan urutan pengerjaan (Simrit 2B- CHG 1B- CHP 1B- HP 1B) memiliki nilai makespan sama yaitu 3146,16 menit lebih baik dibandingkan dengan *Gantt Chart* milik perusahaan dengan urutan pengerjaan (CHG 1B- Simrit 2B- HP 1B- CHP 1B) memiliki makespan sebesar 4602,02 menit. Dari hasil *Efficiency Index* dan *Relative Error* dengan nilai 1,46 dan 46,27 % menunjukkan bahwa penjadwalan yang dirancang menggunakan algoritma Nawaz, Enscore, dan Ham lebih baik dibandingkan eksisting. Hal ini terlihat dari nilai *makespan* menurun 1455,86 menit.

Maka rekomendasi dari penelitian ini adalah dalam melakukan prioritas *job* yang dikerjakan terlebih dahulu berdasarkan jumlah benang kemudian jumlah operasi dan yang terakhir berdasarkan jumlah permintaan apabila terjadi datangnya *order job* secara bersamaan. Kemudian dalam hal melakukan alokasi proses produksi sebaiknya PT. Sahabat tidak menunggu semua proses selesai terlebih dahulu baru di alokasikan ke stasiun selanjutnya, sebaiknya ketika sudah memenuhi kapasitas dari proses selanjutnya maka harus langsung di proses agar tidak terjadi penumpukan di satu stasiun kerja.

5. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini berdasarkan rumusan masalah adalah sebagai berikut:

1. Dalam penelitian ini diketahui bahwa ketika mengerjakan pesanan sebanyak 7000 meter selang CHG 1B memiliki waktu pengerjaan sebesar 1836,71 menit, 1000 meter selang CHP 1B memiliki waktu pengerjaan sebesar 673,83 menit, 3000 meter Simrit 2B memiliki waktu pengerjaan sebesar 1612,84 menit, dan 2000 meter selang HP 1B memiliki waktu pengerjaan sebesar 880,40 menit.
2. Dari penelitian ini didapatkan minimasi waktu *makespan* yang sama baik menggunakan inisialisasi awal dengan LPT dan SPT mendapatkan hasil 3146,16 menit dengan urutan proses pengerjaan dari Selang Simrit 2B- Selang CHG 1B – Selang CHP 1B – HP 1B.
3. Dari perbandingan antara antara penjadwalan eksisting (konvensional) dengan algoritma Nawaz, Enscore, dan Ham

(NEH) dengan LPT dan SPT didapatkan nilai *Efficiency Index* (EI) sebesar 1,46 yang berarti model algoritma NEH yang digunakan dalam penelitian ini lebih baik dibandingkan dengan model penjadwalan eksisting (konvensional) di PT. Sahabat Rubber Industries. Dari nilai *Relative Error* (RE) memiliki selisih sebesar 46,27% yang dibandingkan antara model algoritma NEH dalam penelitian ini dengan model penjadwalan eksisting.

Daftar Pustaka

- Allahverdi, Ali, Cheng, Ng. T.C.E., & Kovalyov, Mikhail. (2006). *A Survey Scheduling Problems With Setup Times or Cost*. Minsk : European Journal of Operational Research
- Bedworth, David & Bailey, James. (1987). *Integrated Production Control System, Management Analysis, Design*. Edisi ke-2. New York: John Wiley & Sons, Inc.
- Baker, K.R. & Trietsch. (2009). *Principles of Sequencing and Scheduling*. New York: John Wiley & Sons, inc.
- Chen, J.S, Pan, J.C.H. & Lin, C.M. (2009). *Solving the Re-entrant Permutation Flowshop Scheduling Problem with a Hybrid Genetic Algorithm*. Taipei : International Journal of Industrial Engineering.
- Ginting, Rosnani. (2009). *Penjadwalan Mesin*. Yogyakarta: Graha Ilmu.
- Graves, S.C, Meal, Harlan C., Stefek, Daniel. (1983). *Scheduling of Re-entrant Flowshops*. Massachusetts: Journal of Operations Management Massachusetts Institute of Technology.
- Laha, Dipak & Sapkal, Sagar. (2011). *An Efficient Heuristic Algorithm for m-Machine No-Wait Flowshop*. Hongkong: Proceeding of International MultiConference of Engineers and Computer Scientist 2011 Vol. 1.
- Pinedo M. (2008). *Scheduling*. New York: Springer.