ISSN: 1412-3339

METODE META-HEURISTIC UNTUK MENYELESAIKAN PERMASALAHAN PENJADWALAN PRODUKSI: CROW SEARCH ALGORITHM

[1]Antono Adhi, [2]M. Riza Radyanto, [3]Gilang Rachmat Pratama
[1,2,3]Program Studi Teknik Industri, Universitas Stikubank (UNISBANK) Semarang
antonoadhi@edu.unisbank.ac.id
rizaradyanto@edu.unisbank.ac.id

Abstrak

Penjadwalan adalah bagian dari proses produksi. Proses penjadwalan dilakukan sebagai persiapan proses produksi. Pada pelaksanaannya, penjadwalan ulang juga dapat dilakukan jika terjadi perubahan keputusan pada proses produksi. Proses penjadwalan adalah menyelesaikan setiap tugas yang biasa disebut job pada sumber daya mesin yang tersedia. Proses ini adalah menentukan job mana yang terlebih dahulu akan dikerjakan.

Setiap proses produksi mempunyai tujuan. Tujuan yang biasa diambil adalah diperolehnya waktu total proses produksi seminimal mungkin atau makespan. Tujuan lainnya adalah diperolehnya biaya keterlambatan (lateness) dan terlalu cepatnya suatu job dikerjakan (earliness), sekecil mungkin. Tujuan penjadwalan produksi adalah agar proses produksi dapat diselesaikan dengan seoptimal mungkin, yaitu dengan waktu dan biaya sekecil-kecilnya.

Pada kenyataannya, penjadwalan adalah termasuk sebagai proses NP-hard dalam penyelesaiannya. Proses ini sulit akan dikerjakan dengan metode eksak biasa jika memiliki jumlah job yang besar. Dalam hal ini proses akan membutuhkan waktu yang sangat panjang. Oleh karena itu proses penjadwalan hanya dapat diselesaikan metode meta-heuristic. Pada metode ini, hasil optimum dapat diterima pada waktu pemrosesan yang tidak terlalu lama. Salah satu metode meta-heuristic yang digunakan dalam penelitian ini adalah menggunakan algoritma crow search algorithm (CSA). Algorima ini cukup sederhana namun mampu memberikan hasil yang baik untuk menyelesaikan permasalahan penjadwalan produksi.

Kata kunci: Penjadwalan Produksi, Optimum, Meta-Heuristic, Crow Search, Algorithm

Abstract

Scheduling is part of the production process. The scheduling process is done in preparation for the production process. In practice, rescheduling can also be done if there is a change in the production process. The scheduling process is completed for each task which is usually called work on available machine resources. This process determines which work will be done first.

Every production process has a purpose. The usual goal taken is the time obtained by the total production process to a minimum or makespan. Another goal is to get late fees (delay) and too fast the work done (beginning), as small as possible. The purpose of production scheduling is so that the production process can be completed optimally, namely with the smallest time and cost.

In the end, scheduling includes the NP-hard process in its completion. This process is difficult to do with the usual exact method if you have a large number of jobs. In this case the process will take a very long time. Therefore the scheduling process can only be completed by the meta-heuristic method. In this method, optimal results can be received at the time of receipt that is not too long. One of the meta-heuristic methods used in this study is to use the crow search algorithm (CSA) algorithm. This algorithm is quite simple but gives good results to complete production scheduling.

Keywords: Production Scheduling, Optimal, Meta-Heuristics, Crow Search, Algorithm

I. PENDAHULUAN

Penjadwalan adalah proses yang berhubungan dengan pengalokasian sumber daya untuk menjalankan suatu tugas dalam suatu periode waktu tertentu (Baker, 1974; Morton & Pentico, 1993; Pinedo, 2008). Bidang usaha yang melakukan aktivitas penjadwalan ini bisa meliputi apa saja baik di bidang produksi maupun jasa. Sumber daya bisa meliputi mesin dalam sebuah lantai produksi, jalur penerbangan suatu bandara, tenaga kerja pada sebuah pekerjaan konstruksi, unit pemroses dalam suatu komputer, dan lainnya (Pinedo, 2008). Sedangkan tugas yang diberikan kepada sumber daya bisa berupa operasi pada proses produksi, proses terbang dan pendaratan pesawat, pekerjaan dalam proses konstruksi, eksekusi proses komputer, dan lainnya (Pinedo, 2008). Mesin-mesin dijadwalkan untuk menjalankan *job* pesanan pelanggan.

ISSN: 1412-3339

Melihat peranan proses penjadwalan tersebut, maka masalah dalam penjadwalan adalah hal yang sangat penting dalam suatu sistem industri (Kaplanoglu, 2016). Bahkan penjadwalan memegang peranan penting pada sebagian besar sistem produksi dan manufaktur (Pinedo, 2008). Untuk memperoleh hasil yang baik dalam penjadwalan, proses tersebut harus mempunyai target yang terarah. Target utama dalam penjadwalan adalah memperoleh nilai optimal pada satu atau lebih tujuan (Pinedo, 2008).

Banyak pendekatan yang dapat dilakukan untuk memperoleh hasil penjadwalan *flexible flow shop* yang optimum, baik melalui algoritma eksak, heuristik, maupun meta-heuristik (Lei & Guo, 2016). Namun metode eksak seperti *branch and bound* atau *dynamic programming* hanya cocok dilakukan pada penjadwalan *flexible flow shop* dengan skala kecil (Singh & Mahapatra, 2016). Hal ini disebabkan penjadwalan *flexible flow shop* adalah termasuk masalah *NP-hard* (Lei & Guo, 2016). Dengan problem *NP-hard*, penjadwalan yang komplek, dalam hal ini semakin banyak *job* jumlah komputasinya tidak bertambah secara linear, tapi secara faktorial. Karena jumlah kemungkinan urutan dari suatu permutasi adalah perhitungan faktorial. Metode eksak akan mengalami kegagalan jika skala permasalahan terlalu kompleks karena kebutuhan memori yang besar dan waktu komputasi yang lama (Singh & Mahapatra, 2016).

Metode yang memungkinkan untuk dilakukan pada proses penjadwalan yang *NP-hard* dengan solusi yang mendekati optimum terdekat dan jumlah waktu komputasi lebih pendek adalah dengan metode heuristik atau meta-heuristik (Singh & Mahapatra, 2016; Ding, Song, & Wu, 2016). Salah satu metode meta-heuristik yang baru adalah metode Crow Search Algorithm (CSA). Metode ini terglong dalam metode meta-heuristik kontinyu. Dalam penelitian ini metode CSA akan dicoba untuk menyelesaikan permasalahan penjadwalan. Berdasarkan latar belakang masalah yang telah diuraikan, maka yang menjadi pokok permasalahan dalam penelitian ini adalah bagaimana mengimplementasikan metode CSA dalam permasalahan penjadwalan produksi. Penelitian yang akan dilakukan mempunyai batasan-batasan sebagai berikut: 1). Metode penjadwalan menggunakan metode *meta-heuristic* Crow Search Algorithm (CSA), 2).Proses perhitungan dikerjakan melalui program Visual Basic.

Tujuan yang akan dicapai dari perancangan ini adalah merancang penyelesaian penjadwalan produksi dengan metode *meta-heuristic* Crow Search Algorithm. Manfaat dari penelitian ini adalah untuk memberikan masukan kepada pihak-pihak terkait yaitu sebagai berikut:

1. Menyelesaian permasalahan penjadwalan dengan hasil yang optimum.
2. Dapat dijadikan dasar untuk implementasi di permasalahan yang lain terutama yang berkaitan dengan optimisasi diskret.
3. Menunjang teori di proses pembelajaran di kelas.

II. TELAAH PUSTAKA

Penjadwalan

Secara umum penjadwalan didefinisikan dengan pengalokasian sumber daya untuk melaksanakan suatu pekerjaan dalam lintasan waktu (Baker, 1974). Pengertian ini mengarah kepada dua arti (Baker, 1974). Yang pertama, penjadwalan adalah sebuah fungsi pengambilan keputusan untuk menentukan suatu jadwal (Baker, 1974). Maka penjadwalan dapat diaplikasikan pada pengambilan keputusan yang lain. Yang kedua, penjadwalan adalah bagian dari sebuah teori berkenaan dengan sekumpulan prinsip, model, teknik, dan kesimpulan logis yang ada dalam fungsi penjadwalan (Baker, 1974). Maka penjadwalan dapat diaplikasikan pada teori-teori lain dan mempunyai nilai konsep yang umum.

Permasalahan pengalokasian sumber daya untuk melaksanakan tugasnya dapat terjadi pada berbagai kondisi. Namun begitu, penjadwalan tidak akan dipikirkan hingga beberapa

ISSN: 1412-3339

permasalahan dalam perencanaan dasar dapat diselesaikan (Baker, 1974). Bagi pihak pengambil keputusan, penjadwalan mempunyai tingkat kepentingan kedua (Baker, 1974). Setelah studi pasar dalam pengembangan produk, maka analisis ekonomi akan fokus pada teknologi bagaimana barang akan diproduksi. Setelah permasalahan ini terjawab, maka baru permasalahan penjadwalan akan dipikirkan (Baker, 1974).

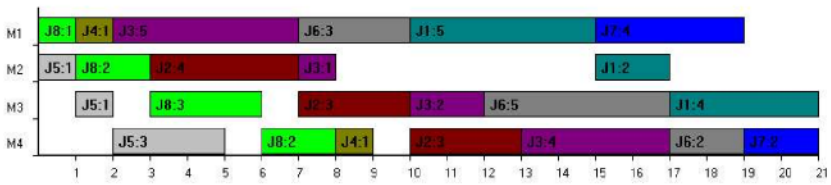
Keputusan manajerial dasar dihadapkan pada pertanyaan, barang atau jasa apa yang akan disediakan, skala apa akan disediakan, dan sumber daya apa yang memungkinkan akan digunakan (Baker, 1974). Definisi tersebut berhubungan dengan fungsi perencanaan. Kebalikannya, bagi fungsi penjadwalan, pertanyaan-pertanyaan tersebut mengansumsikan sudah ada jawabannya. Sehingga fungsi penjadwalan menjadi relevan pada situasi tugas yang akan dijadwal sudah tergambar dan sumber daya sudah ditentukan (Baker, 1974).

Dalam prakteknya tentu fungsi penjadwalan dan perencanaan tidak dapat bekerja sendiri-sendiri (Baker, 1974). Masing-masing saling terkait. Awalnya seorang perencana akan mengidentifikasi tugas yang harus diselesaikan dan batas kapasitas sumber daya yang akan digunakan. Penjadwal menggunakan informasi ini untuk mengalokasikan sumber daya yang ada untuk menyelesaikan tugas. Setelah jadwal sementara terbentuk, dapat mengevaluasinya dan menyampaikan pada perencana. Perencana mungkin tidak puas dengan hasil penjadwalan dan mengubah kapasitas sumber daya atau tugas itu sendiri. Perubahan baru ini menjadi masukan bagi penjadwal.

Keputusan perencanaan merupakan keputusan jangka panjang, seperti perancangan atau ekspansi fasilitas, pengadaan dan instalasi peralatan, dan menetapkan tenaga kerja (Baker, 1974). Keputusan yang diambil akan mempertimbangkan bagaimana penjadwalannya dapat dilakukan (Baker, 1974). Sehingga proses penjadwalan sering muncul pada situasi di mana ketersediaan sumber daya sudah ditetapkan pada komitmen jangka panjang fungsi perencanaan (Baker, 1974).

Dengan latar belakang seperti tersebut, maka keputusan dalam penjadwalan dapat diperoleh dengan pendekatan sistem (Baker, 1974). Pendekatan sistem memiliki struktur formal. Empat tahap utama dalam pendekatan sistem adalah formulasi, analisis, sintesis, dan evaluasi (Baker, 1974). Pada tahap formulasi, masalah diidentifikasi dan kriteria untuk pengambilan keputusan penyelesaiannya ditetapkan (Baker, 1974). Analisis adalah proses rinci untuk menguji elemen permasalahan dan keterkaitannya (Baker, 1974). Tahap sintesis membangun solusi alternatif untuk menyelesaikan masalah (Baker, 1974). Tahap evaluasi membandingkan alternatif yang memungkinkan dan memilih tindakan yang akan dilakukan berdasarkan kriteria yang telah ditetapkan (Baker, 1974).

Model formal seperti Gantt *chart* dapat digunakan untuk membantu mengambil keputusan dalam masalah penjadwalan (Baker, 1974). Bentuk dasar Gantt *chart* menggambarkan alokasi sumber daya pada lintasan waktu. Gambar 3.1. menunjukkan contoh Gantt chart penjadwalan 8 *job* pada 4 mesin secara *flowshop* dengan *stage skipping*.



Gambar 3.1. Gantt chart

Teori penjadwalan terutama bersangkutan dengan model matematika pada fungsi penjadwalan dan pengembangan model dan teknik (Baker, 1974). Perspektif teorinya terutama berhubungan dengan pendekatan kuantitatif (Baker, 1974). Pendekatan kuantitatif

ISSN: 1412-3339

dimulai dari menerjemahkan tujuan keputusan dalam fungsi tujuan dan batasan pengambilan keputusan.

Idealnya, fungsi tujuan harus mengakomodasi seluruh biaya dalam sistem yang muncul pada pengambilan keputusan penjadwalan (Baker, 1974). Pada kenyataannya, hal seperti biaya sulit untuk diukur atau diidentifikasi secara lengkap (Baker, 1974). Namun begitu, tiga tujuan pengambilan keputusan lazin digunakan dalam penjadwalan, yaitu efisiansi penggunaan sumber daya, kecepatan respon pada permintaan, dan kesesuaian terdekat pada pemenuhan batas waktu (Baker, 1974). Untuk total biaya sistem, biaya yang berhubungan dengan kinerja sistem seperti waktu *idle* mesin, waktu tunggu *job*, atau keterlambatan *job*, dapat digunakan atau dimanfaatkan sebagai perhitungan tambahan (Baker, 1974).

Dua batasan yang umum ditemukan dalam masalah penjadwalan adalah batas kapasitas sumber daya yang tersedia dan teknologi suatu tugas dapat dilaksanakan (Baker, 1974). Sehingga pemecahan masalah penjadwalan akan menjawab pertanyaan: sumber daya mana yang akan dialokasikan untuk menjalankan tugas dan kapan setiap tugas tersebut dijalankan (Baker, 1974). Dengan demikian, inti dari penjadwalan adalah (Baker, 1974): keputusan pengalokasian (*allocation decision*) dan keputusan pengurutan (*sequencing decision*). Literatur-literatur yang meneliti penjadwalan dipenuhi dengan usaha penyelesaian dengan model matematis dalam rangka untuk menyelesaikan dua masalah pengambilan keputusan tersebut (Baker, 1974). Pada dasarnya, permasalahan penjadwalan dapat dipandang sebagai permasalahan optimisasi batasan-batasan, khususnya masalah alokasi dan pengurutan (Baker, 1974).

Elemen yang vital dalam model penjadwalan adalah sumber daya dan tugas. Sumber daya digambarkan dengan kapasistas kualitatif dan kuantitatifnya sehingga model menggambarkan jenis dan jumlah sumber daya (Baker, 1974). Tugas individu digambarkan sebagai informasi yang memerlukan sumber daya apa, durasi waktu, kapan mulai, dan batas waktunya (Baker, 1974).

Teori penjadwalan juga termasuk teknik-teknik yang berguna untuk menyelesaikan permasalahan penjadwalan (Baker, 1974). Tentu wilayah penjadwalan adalah titik pusat untuk pengembangan, aplikasi, dan evaluasi dari kombinasi algoritma, teknik simulasi, metode jaringan kerja, dan pendekatan solusi heuristik (Baker, 1974). Pemilihan teknik yang sesuai tergantung pada kompleksitas masalah, dasar dari model, dan pemilihan kriteria sebagai faktor lainnya (Baker, 1974). Pada beberapa kasus pendekatannya mempertimbangkan beberapa teknik alternatif (Baker, 1974). Untuk alasan ini, teori penjadwalan mungkin studi metodologinya banyak ke studi pemodelan (Baker, 1974).

Untuk memisahkan model penjadwalan utamanya, perlu untuk melihat karakter konfigurasi sumber daya dan tugas-tugasnya (Baker, 1974). Misalnya model dapat berisi tipe sumber daya atau beberapa tipe darinya (Baker, 1974). Jika hanya terdapat satu tipe sumber daya, tugas mungkin ada pada *stage* tunggal, namun jika ada banyak model sumber daya biasanya melibatkan banyak tugas, dan dalam kasus yang sama, sumber daya bisa lebih dari satu atau paralel (Baker, 1974). Jika sekumpulan tugas dalam penjadwalan tidak berubah sepanjang waktu, sistem disebut statis, sedangkan ada kasus di mana tugas baru muncul setiap saat dan sistem ini dinamakan sistem dinamis (Baker, 1974). Model statis lebih mudah ditangani daripada model dinamis (Baker, 1974). Model statis dapat dipandang sebagai model dasar sistem dinamis yang lebih kompleks dan analisis masalah statis sering mengatasi pengertian penting dan prinsip-prinsip heuristik yang berguna pada situasi umum (Baker, 1974).

Banyak pengembangan awal wilayah penjadwalan didorong karena ditemukannya permasalahan pada manufaktur (Baker, 1974). Oleh karena itu, permasalahan penjadwalan lebih banyak menggunakan istilah-istilah dalam manufaktur (Baker, 1974). Meskipun saat ini pekerjaan dalam penjadwalan sudah dipertimbangkan dalam banyak bidang di luar

ISSN: 1412-3339

manufaktur, istilah-istilah dalam manufaktur masih digunakan. Seperti sumber daya yang biasa disebut dengan mesin. Modul tugas dasarnya disebut dengan *job* (Baker, 1974). Lebih luas lagi, tugas yang harus diselesaikan disebut *job*, *project*, atau *assignment* (Morton, 1993). Penyelesaian disebut *activity* atau operasi (Morton, 1993). Sumber daya disebut mesin, *cell*, transport, *delay* (Morton, 1993).

Permasalahan sebenarnya dalam pengurutan adalah permasalahan penjadwalan khusus di mana sebuah deretan *job* ditetapkan sebagai jadwal (Baker, 1974). Bahkan masalah penjadwalan ada pada permasalahan sederhana yang hanya dengan satu mesin tunggal. Namun hal ini tetaplah penting karena proses pembelajaran dari masalah mesin tunggal dapat menggambarkan variasi topik-topik penjadwalan pada model yang dapat dikerjakan (Baker, 1974). Masalah penjadwalan pada mesin tunggal mempunyai karakteristik sebagai berikut (Baker, 1974):

1. sejumlah *n job* independen dengan operasi tunggal memungkinkan untuk diproses dengan waktu nol,
2. waktu setup setiap *job* adalah independen tidak tergantung urutan dan dapat dimasukkan pada waktu proses,
3. deskripsi *job* diketahui terlebih dahulu,
4. satu mesin secara terus menerus dapat digunakan dan tidak diam saat ada pekerjaan yang menunggu,
5. sekali proses dimulai pada suatu *job*, proses akan diselesaikan tanpa interupsi.

Dalam kondisi yang ada pada karakteristik mesin tunggal, terdapat korespondensi satu-satu antara urutan *n job* dan permutasi *job* 1, 2, ..., *n* (Baker, 1974). Jumlah kemungkinan solusi yang berbeda dari masalah mesin tunggal adalah *n*!, yaitu sejumlah permutasi dari *n* elemen (Baker, 1974). Saat sebuah jadwal dapat diselesaikan dengan karakteristik sebuah permutasi bilangan bulat, maka hal ini disebut penjadwalan permutasi, yang dikembangkan dari kasus mesin tunggal (Baker, 1974).

Masalah Penjadwalan *Flow Shop*

Pada masalah penjadwalan *flow shop*, sejumlah job $J_1;...; J_n$ diproses pada mesin $M_1;...;M_m$ (Benavides & Ritt, 2016). Setiap job harus diproses pada seluruh mesin ada urutan yang telah ditetapkan. Setiap mesin M_i dijalankan tanpa interupsi pada waktu p_{ij}, dan setiap *job* dapat diproses pada satu mesin pada waktu yang telah ditentukan. *Makespan* dari suatu jadwal diperoleh dari waktu *job* terakhir pada mesin terakhir. Minimisasi merupakan tujuan sebagian besar penjadwalan yang ada. Masalah ini dinotasikan dengan $F\|C_{max}$ dan akan menjadi masalah *NP-hard* untuk setiap $\min\{n, m\} \geq 3$ (Benavides et al., 2016).

Menurut Benavides et al. (2016) jika x_{ij} adalah adalah waktu pengerjaan *job* J_j pada mesin M_i dan variabel $y_{ijj'} \in \{0,1\}$ mengindikasikan bahwa *job* J_j dimulai sebelum job $J_{j'}$ pada mesin M_i, sehingga program linear untuk masalah *flow shop* adalah sebagai berikut:

$$\textbf{min } C_{max}, \tag{1}$$

$$\text{s.t. } x_{mj} + p_{mj} \leq C_{max}, \qquad \forall j \in [n], \tag{2}$$

$$x_{ij} + p_{ij} \leq x_{i+1j}, \qquad \forall i \in [m-1], \forall j \in [n], \tag{3}$$

$$x_{ij} + p_{ij} \leq x_{ij'} + M(1 - y_{ijj'}), \qquad \forall i \in [m], j \neq j' \in [n], \tag{4}$$

$$y_{ijj'} + y_{ij'j} = 1, \qquad \forall i \in [m], j \neq j' \in [n], \tag{5}$$

$$x_{ij} \geq 0, \qquad \forall i \in [m], j \in [n], \tag{6}$$

$$y_{ijj'} \in \{0, 1\}, \qquad \forall i \in [m], j \neq j' \in [n], \tag{7}$$

konstrain (2) mendefinisikan makespan, konstrain (3) mensyaratkan sebuah *job* harus berakhir pada suatu mesin sebelum memulai job berikutnya, konstrain (4) menjamin *job* diproses pada urutan yang didefinisikan pada variabel *y*, dan konstrain (5) memaksa urutan linear *job* pada semua mesin. Konstanta M memiliki nilai cukup besar, misalnya $M = \Sigma_{i\in[m]} \Sigma_{j\in[n]} p_{ij}$.

ISSN: 1412-3339

Optimasi Tujuan Penjadwalan

Tujuan dari minimisasi adalah fungsi waktu penyelesaian *job-job* (Pinedo, 2008). Pinedo (2008) menotasikan waktu untuk menyelesaikan operasi pada *job j* pada mesin *i* dengan C_{ij}. Waktu *job j* pada sebuah sistem adalah C_j. Maka waktu keterlambatan *lateness* suatu *job j* dibandingkan dengan batas waktu d_j didefinisikan sebagai berikut:

$$L_j = C_j - d_j$$

yang bernilai positif saat *job j* diselesaikan terlambat dan negatif saat diselesaikan lebih awal. Sedangkan keterlambatan *tardiness* dari *job j* bernilai positif saat terlambat dan negatif saat diselesaikan lebih awal. Definisi *tardiness job j* adalah sebagai berikut:

$$T_j = \max(C_j - d_j, 0) = \max(L_j, 0)$$

Perbedaan antara *lateness* dan *tardiness* adalah bahwa *tardiness* tidak pernah bernilai negatif. *Unit penalty* dari *job j* didefinisikan sebagai:

$$U_j = \begin{cases} 1 & if \quad C_j > d_j \\ 0 & otherwise \end{cases}$$

Makespan atau total waktu penyelesaian seluruh *job* pada operasi dinotasikan dengan C_{max}. Definisi *makespan* $\max(C_1, ..., C_n)$ ekuivalen dengan penyelesaian *job* terakhir pada sistem. *Earliness* didefinisikan sebagai kebalikan dari *lateness*. *Earliness* didefinisikan sebagai:

$$E_j = \max(d_j - C_j, 0)$$

Tujuan dari penjadwalan bisa merupakan penjumlahan dari total *earliness* dan total *tardiness* yang didefinisikan sebagai berikut:

$$\sum_{j=}^{n} E_j + \sum_{j=}^{n} T_j$$

Tujuan penjadwalan ini dapat dikembangkan dengan membedakan tingkat kepentingan *earliness* dan *tardiness* dengan memberikan bobot pada masing-masing. Sehingga definisinya adalah sebagai berikut:

$$\sum_{j=1}^{n} w' E_j + \sum_{j=1}^{n} w'' T_j$$

Bobot yang berhubungan dengan *earliness job j* (w') bisa berbeda dengan bobot yang berhubungan dengan *tardiness job j* (w'').

Metode *Meta-heuristic*

Penjadwalan adalah menjadwal sejumlah *job* untuk diproses pada mesin-mesin yang tersedia. Tindakan untuk mencari urutan *job* akan diproses adalah permutasi. Permutasi pada masalah penjadwalan *flow shop* sangatlah penting dalam sistem manufaktur (Shao & Pi, 2016). Dia digunakan secara luas pada proses-proses produksi. Didefinisikan sebagai proses pengurutan *n jobs* pada sejumlah *m* mesin untuk meminimalkan total waktu proses atau memenuhi tujuan lainnya (Shao et al., 2016).

Permutasi dengan jumlah *job* yang semakin besar akan menambah tingkat kompleksitas perhitungan penjadwalan. Hal tersebut juga ditambah dengan model-model produksi yang mempunyai proses yang rumit sehingga menambah tingkat kompleksitas perhitungan penjadwalan. Konsekuensinya adalah waktu perhitungan yang lama. Oleh sebab itu permasalahan penjadwalan merupakan permasalahan *NP-hard* yang hanya dapat diselesaikan dengan bantuan metode meta-heuristik yang mampu memberikan hasil mendekati optimum pada waktu yang dapat diterima (Lei & Guo, 2016; Singh & Mahapatra, 2016; Ding et al., 2016).

Setiap algoritma metode *meta-heuristic* mempunyai karakteristik perhitungan yang berbeda-beda dengan hasil hasil optimisasi yang berbeda-beda pula. Oleh karena itu dalam kasus perhitungan penjadwalan *flexible flow shop* dengan *sequence-dependent setup time*

ISSN: 1412-3339

akan dibandingkan hasil perhitungan pengembangan algoritma baru dengan algoritma metode meta-heuristik lainnya.

Crow Search Algorithm (CSA)

Gagak adalah burung yang paling cerdas. Mereka memiliki volume otak relatif lebih besar dibandingkan dengan besar tubuhnya (Askarzadeh, 2016). Dibandingkan dengan rasio kepala dan tubuh, otak mereka sedikit lebih kecil dibandingkan dengan manusia. Gagak mampu mengenali wajah dan memberi peringatan saat ada gangguan datang. Beberapa prinsip dari CSA menurut Askarzadeh (2016) adalah sebagai berikut:

1. Gagak hidup berkelompok,
2. Gagak mengingat posisi tempat persembunyiannya,
3. Gagak saling ikut satu sama lain saat mencuri makanan,
4. Gagak menjaga kemungkinan penyerobotan tempat makanannya.

Diasumsikan ada dimensi lingkungan d, ada sejumlah gagak. Jumlah gagak sebanyak N dan posisi gagak setiap iterasi *iter* dinotasikan dengan vektor $x^{i,iter}$ (i=1, 2, ..., N; $iter$ = 1, 2, ..., $iter_{max}$). $x^{i,iter} = [x_1^{i,iter}, x_2^{i,iter}, ..., x_d^{i,iter}]$ dan $iter_{max}$ adalah maksimum iterasi. Setiap gagak memiliki memori lokasi persembunyiannya. Pada setiap iterasi *iter*, posisi tempat persembunyiannya dinotasikan dengan $m^{i,iter}$. Ini adalah posisi terbaik gagak ke i. Di lingkungannya, setiap gagak akan mencari posisi terbaiknya untuk dijadikan tempat persembunyian untuk menyimpan makanan.

Asumsikan pada suatu iterasi *iter*, seekor gagak j akan mengunjungi tempat persembunyiannya, $m^{j,iter}$. Kemudian gagak i mengikuti gagak j di lokasi persembunyian gagak j. Kondisi yang mungkin terjadi adalah sebagai berikut:

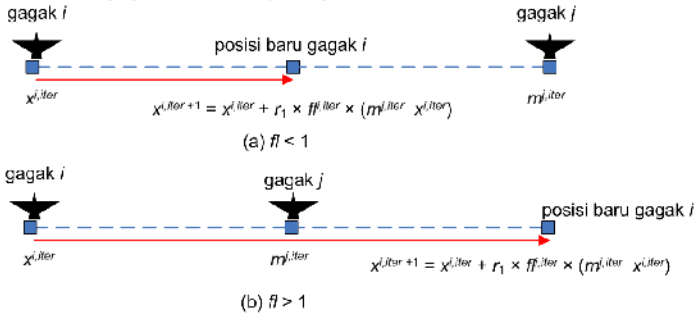
1. Gagak j tidak tahu bahwa gagak i mengikutinya. Sehingga gagak i mendekati tempat persembunyian gagak j. Posisi baru gagak i adalah sebagai berikut:

   $x^{i,iter+1} = x^{i,iter} + r_1 \times fl^{i,iter} \times (m^{j,iter} - x^{i,iter})$

   r_1 adalah bilangan random antara 0 dan 1. $fl^{i,iter}$ adalah panjang terbang gagak i pada iterasi ke *iter*.

   Gambar 3.2 menunjukkan skema kondisi 1 dan pengaruh fl pada kemampuan pencarian. Bilangan kecil fl mengarahkan gagak i ke pencarian lokal (sekitar $x^{i,iter}$) dan bilangan besarnya akan menyebabkan ke pencarian global (jauh dari $x^{i,iter}$).

   Gambar 3.2. (a) menunjukkan kondisi jika fl bernilai kurang dari 1. Posisi gagak i berikutnya berada antara $x^{i,iter}$ dan $m^{i,iter}$. Gambar 3.2. (b) menunjukkan jika fl bernilai lebih dari 1. Posisi gagak i berikutnya di jalur di luar $m^{i,iter}$.



Gambar 3.2. Perpindahan gagak i

2. Gagak j tahu jika gagak i mengikutinya. Maka untuk melindungi tempat penyimpanan makanannya agar tidak diserobot, gagak j akan mengelabuhi dengan berpindah ke posisi baru di ruang pencarian. Maka kondisi 1 dan 2 dapat diekspresikan dengan perhitungan sebagai berikut:

ISSN: 1412-3339

$$x^{i,iter+1} = \begin{cases} x^{i,iter} + r_i \times fl^{i,iter} \times \left(m^{j,iter} - x^{i,iter}\right) & r_j \geq AP^{j,iter} \\ posisi \quad acak, \quad lainnya \end{cases}$$

III. METODE PENELITIAN

Metode penelitian dilakukan mulai dari studi lanjut, kemudian dilanjutkan dengan pengembangan teori, pengembangan aplikasi, pembangkitan data, pemrosesan data, analisis, dan penyimpulan.

Studi Literatur

Pada tahap studi literatur, pelaksanaan penelitian dilakukan melalui studi pada teori dasar. Teori dasar dipelajari terhadap teori tentang penjadwalan, metode *meta-heuristic* dan algoritma CSA.

Pengembangan Teori Dasar

Permasalahan penjadwalan dibagi dalam beberapa jenis. Model penjadwalan yang sering diimplementasikan dan diteliti adalah *flow-shop*, *job-shop*, dan *batch-shop*. Setiap model pun dikembangkan dengan permasalahan yang lebih detil lagi.

Model penjadwalan yang digunakan dalam penelitian ini adalah model *hybrid flow-shop*. Pada model ini setiap *job* yang dijadwal untuk dikerjakan dalam suatu mesin akan melalui tahapan yang sama. Tahapan tersebut disebut dengan *stage*. Namun ada juga *job* yang tidak melewati *stage* atau disebut *stage skipping*.

Hybrid flow-shop menunjukkan bahwa proses *flow-shop* mempunyai beberapa mesin paralel di setiap tahapannya. Pada proses penjadwalan, permasalahan akan dihadapkan pada mesin mana saja yang dapat digunakan untuk memproses suatu *job*.

Proses penjadwalan *sequence dependent time* juga mempertimbangkan waktu *setup* yang berbeda-beda untuk setiap *job* dan berbeda-beda juga untuk *job* yang berbeda yang mengawali pemrosesan suatu *job*. Proses pengembangan ini akan membutuhkan tingkat kesulitan perhitungan penjadwalan yang lebih kompleks dan waktu yang lebih banyak.

Algoritma CSA adalah algoritma *meta-heuristic* yang dikembangkan dengan meniru perilaku gagak. Algoritma ini termasuk algoritma penyelesaian masalah kontinyu. Oleh karena itu algoritma ini perlu dimodifikasi untuk dapat menyelesaikan permasalahan penjadwalan yang diskrit. Proses modifikasi dengan mengambil urutan setiap *job* atau melalui proses *short*.

Pengembangan Aplikasi

Aplikasi dikembangkan untuk membantu mempermudah proses entri data, penjadwalan, dan analisis perhitungan. Aplikasi dikembangkan melalui bahasa pemrograman Visual Basic. Dengan demikian proses interaksi antara *user* dan komputer dapat dilakukan dengan baik dan mudah.

Data-data yang diperlukan dalam pemrosesan disimpan dalam format text. Data diambil pada awal aplikasi dijalankan dan disimpan dalam format *array*. Dan akan disimpan setiap kali ada perubahan data. Hal ini dapat dilakukan dengan baik karena data tidak terlalu besar sehingga tidak perlu disimpan dalam *database engine* tertentu yang justru dapat memperlambat pemrosesan.

Dengan disimpannya data dalam format text tersebut mengakibatkan proses instalasi dapat dilakukan dengan cepat dan sederhana.

Pembangkitan data

Data-data yang diperlukan untuk mengaplikasikan algoritma CSA ke dalam penjadwalan dibangkitkan dari proses pengacakan dengan range nilai tertentu. Sebelum mengbangkitkan data, beberapa data perlu ditentukan nilainya terlebih dahulu, yaitu data: 1). Jumlah *stage* dan *job*. 2). Jumlah mesin paralel untuk setiap *stage*. Data yang dibangkitkan adalah:

1. Stage yang dilewati oleh setiap *job*

ISSN: 1412-3339

2. Waktu proses *job* dalam setiap mesin
3. Waktu setup *job* yang berbeda untuk *job* berbeda yang memulai
4. *Due-date job*

Pemrosesan Data

Tahap pemrosesan data adalah tahap penyelesaian masalah penjadwalan. Bebera hal harus diseting terlebih dahulu seperti parameter algoritma CSA. Parameter tersebut adalah jumlah populasi burung gagak (*pop*), jumlah maksimal iterasi perhitungan (*itr*), *awarness probability* (*AP*) atau pengontrol tingkat intensifikasi, *flight length* (*fl*) atau panjang lintasan suatu gagak, *lower bound* (*LB*) atau batas nilai bawah, dan *upper bound* (*UB*) atau batas nilai atas.

Tujuan dari penjadwalan mempunyai pilihan apakah biaya *tardiness* terkecil, biaya *earliness* terkecil, atau *makespan*. Biaya *tardiness* dan *earliness* dapat diperbandingkan. Sedangkan jika *makespan* dipilih, maka jika 2 kombinasi mempunyai biaya yang sama, akan dibandingkan mana yang mempunyai *makespan* terkecil.

Analisis dan Kesimpulan

Percobaan penyelesaian permasalahan penjadwalan dengan algoritma CSA dilakukan sebanyak 30 kali. Percobaan akan memberikan informasi rata-rata waktu penyelesaian dan pencapaian tercepat nilai optimum. Nilai optimum akan dibandingkan dengan nilai global optimum yang diperoleh melalui algoritma *Generate and Test*.

IV. HASIL DAN PEMBAHASAN

Data Hasil Perhitungan

Data hasil percobaan yaitu penyelesaian masalah penjadwalan dengan algoritma CSA adalah seperti pada tabel 5.1. *Fitness* yang diukur adalah total waktu penyelesaian seluruh *job* (*makespan*). Nilai *fitness* terkecil yang dihitung melalui algoritma *Generate and Test* adalah 46. *Iterasi* adalah iterasi tercepat yang diperoleh untuk menghasilkan nilai *fitness* terkecil. *Waktu* adalah waktu tercepat hingga diperoleh nilai *fitness* terkecil. Hasil rekapitulasi perhitungan ditunjukkan seperti pada Tabel 5.2.

Tabel 5.1. Hasil Perhitungan

NO	FITNESS	ITERASI	WAKTU
1	46	803	256
2	46	282	117
3	47	6	52
4	46	282	160
5	47	3	98
6	47	8	21
7	46	363	153
8	46	201	130
9	46	381	212
10	46	414	146
11	46	627	245
12	47	9	77
13	46	59	114
14	46	3	19
15	46	103	68
16	46	306	147
17	46	78	98
18	47	12	98

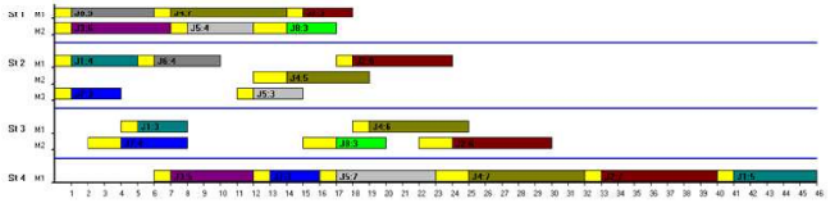
ISSN: 1412-3339

19	46	276	113
20	46	375	172
21	46	335	202
22	46	65	42
23	46	262	131
24	46	327	174
25	47	28	107
26	46	720	251
27	46	902	328
28	46	597	251
29	47	5	89
30	46	125	131

Tabel 5.2. Hasil Rekapitulasi

REKAPITULASI	FITNESS	ITERASI	WAKTU
AVERAGE	46,23333	265,2333	140,0667
MAX	47	902	328
MIN	46	3	19
RANGE	1	899	309

Hasil dari salah satu perhitungan dengan urutan eksekusi *job* 6, 3, 4, 7, 5, 2, 1, 8 menghasilkan *fitness makespan* sebesar 46 satuan periode waktu. Urutan pengerjaan *job* dalam mesin tersebut ditunjukkan seperti *gannit chart* pada Gambar 5.1. Gambar balok dengan warna kuning adalah waktu *setup* yang memulai proses pengerjaan *job*.



Gambar 5.1. Gantt chart hasil perhitungan

Analisis

Dari hasil 30 perhitungan penjadwalan dengan metode CSA, hasil *fitness* dapat dicapai mendekati nilai optimum global dengan besar 46,23 satuan periode waktu. Hasil ini cukup baik mengingat hasil paling optimum yang dapat diperoleh adalah 46. Waktu yang diperoleh untuk menghasilkan nilai *fitness* terkecil cukup baik yang rata-rata diperoleh 140 detik. Bahkan ada perhitungan yang hanya membutuhkan waktu 19 detik pada iterasi ke 3 untuk memperoleh *fitness terkecil*.

Beberapa perhitungan mempunyai nilai *fitness* sebesar 47 satuan waktu. Ini tidak lebih dari besar bedanya dengan nilai *fitness* terkecil yang dapat dicapai. Namun *fitness* bernilai 47 ini juga lebih banyak diperoleh pada waktu dan iterasi yang kecil. Hal ini juga menunjukkan algoritma CSA masih mungkin untuk terjebak pada nilai optimum lokal.

V. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil pengembangan aplikasi dan implementasi algoritma CSA untuk menyelesaikan permasalahan dalam proses penjadwalan produksi, dapat disimpulkan sebagai berikut:

ISSN: 1412-3339

1. Algoritma CSA mempunyai kinerja yang baik untuk menyelesaikan proses permasalahan optimisasi pada penjadwalan.
2. Kinerja CSA yang baik tersebut dihasilkan dari nilai *fitness* yang diperoleh dan kecepatan waktu proses untuk menghasilkan nilai optimum.

Saran

Dari penelitian yang telah dikembangkan untuk mengimplementasikan algoritma CSA pada permasalahan penjadwalan, diperoleh saran untuk memperbaikinya sebagai berikut:

1. Algoritma CSA memungkinkan dipergunakan untuk menyelesaikan permasalahan optimisasi lain baik diskrit maupun kontinyu.
2. Algoritma CSA dapat dinilai tingkat efektifnya untuk menyelesaikan permasalahan optimisasi jika dapat dibandingkan dengan algoritma *meta-heuristic* lainnya.

DAFTAR PUSTAKA

Askarzadeh, Alireza, (2016). "*A Novel Metaheuristic Method for Solving Constrained Engineering optimization problems: Crow Search Algorithm, Computers and Structures*", 169, p. 1–12.

Baker, Kenneth R., (1974). "*Introduction to Sequencing and Scheduling*", John Wiley & Sons, New York.

Benavides, Alexander J., Ritt, Marcus, (2016). "*Two Simple and Effective Heuristics for Minimizing the Makespan in Non-Permutation Flow Shops, Computers & Operations Research*", 66, p.160–169.

Ding, Jian-Ya, Song, Shiji, Wu, Cheng, (2016). "*Carbon-Efficient Scheduling of Flow Shops by Multi-Objective Optimization, European Journal of Operational Research*", 248, p. 758–771.

Kaplanoglu, Vahit, (2016). "*An Object-Oriented Approach For Multi-Objective Flexible Job-Shop Scheduling Problem, Expert Systems With Applications*", 45, p. 71–84.

Lei, Deming, Guo, Xiuping, (2016). "*Hybrid Flow Shop Scheduling with Not-All-Machines Options via Local Search with Controlled Deterioration, Computers & Operations Research*", 65, p. 76–82.

Morton, Thomas E., Pentico, David W., (1993). "*Heuristic Scheduling System: with Applications to Production Systems and Project Management*", John Wiley & Sons, Inc, New York.

Pinedo, Michael L., (2008). "*Scheduling, Springer*", Third Edition.

Shao, Weishi, Pi, Dechang, (2016). "*A Self-Guided Differential Evolution with Neighborhood Search for Permutation Flow Shop Scheduling, Expert Systems With Applications*", 51, p. 161–176.

Singh, Manas Ranjan, Mahapatra, S.S., (2016). "*A Quantum Behaved Particle Swarm Optimization for Flexible Job Shop Scheduling*", Computers & Industrial Engineering, 93, p. 36–44.