

Penjadualan Produksi Flowshop dengan *Availability Constraints Genetic Algorithm* dan *Tabu Search* untuk Minimalisasi *Makespan Time*

Wawan Indarto

Jurusan Teknik Informatika, Universitas Islam Indonesia, Yogyakarta
Jalan Kaliurang Km 14 Yogyakarta Indonesia
e-mail: wawan@fti.uui.ac.id

Abstract

This work describes a genetic algorithm and tabu search approach that will be hybridized to CDS algorithm to solve the flowshop scheduling with availability constraint. In such problem, machine are not available to work with, because it must entering the maintenance time. In this paper, two variant of non preemptive with an arbitrary number of machines and arbitrary number of unavailability constraint on each of them considered. Computational experiments are performed on randomly generated instances to show the efficiency of the proposed approaches.

Keywords: *flowshop, multiple objectives, genetic algorithm, availability constraint, tabu search.*

1. Pendahuluan

Setiap perusahaan atau industri memiliki *plan* (perencanaan) dan strategi. Stevenson membagi strategi bisnis atas beberapa elemen, diantaranya: pasar, investasi, strategi pemasaran, strategi keuangan, dan strategi operasi produksi. Strategi operasi produksi merupakan inti dari seluruh kebijakan yang berhubungan dengan produksi, persediaan barang, pendistribusian barang dan pelayanan konsumen.

Penerapan strategi operasi dan produksi dalam sistem manufaktur terdiri atas beberapa fungsi produksi. Fungsi produksi dalam sistem manufaktur dapat digambarkan sebagai sebuah proses yang hirarkis. Pada level teratas, perusahaan harus melakukan peramalan pasar (*forecast demand*) untuk menentukan kapasitas penjualan, sesuai perencanaan awal produksi. *Forecasting* akan memberikan input berupa tingkat produksi dan tingkat kerja yang harus dilakukan terhadap *Aggregate Plan* (rencana induk produksi), yang kemudian akan diterjemahkan ke tingkat produksi di bawahnya sebagai *Master Production Schedule* (MPS). MPS digunakan untuk membuat spesifikasi tujuan produksi dan waktu produksi. *Materials Requirements Planning* (MRP) merupakan tahap setelah MPS, pada level ini akan dilakukan analisis kebutuhan produk dan komponen yang diperlukan untuk memenuhi kebutuhan produksi serta pengaturan tingkat persediaannya. Setelah semua level di atas telah dilakukan, kemudian diterjemahkan ke dalam beberapa set *job* dan batas waktu penyelesaian *job* (*due date*), hal ini terdapat pada level *scheduling*.

Pada penelitian ini akan digunakan penjadualan dengan tipe *flowshop*. Penjadualan dengan tipe *flowshop* merupakan penjadualan proses produksi dengan tipe *flow* dan mempunyai kategori dinamis.

2. Permasalahan

Dalam industri manufaktur yang berorientasi pada produksi massa, kriteria performansi yang ingin dicapai dengan diadakannya kegiatan penjadualan adalah menyelesaikan semua pekerjaan/*job* yang ada dengan *job makespan* seminimal mungkin.

Dengan tercapainya hal ini, maka akan bisa menaikkan produktifitas perusahaan, juga bisa menaikkan tingkat utilitas mesin-mesin produksi yang dipunyai.

Tujuan dari penelitian ini adalah untuk mencari pemecahan masalah penjadwalan produksi flowshop dengan *availability constrain* (PFAC). PFAC dapat dijelaskan sebagai berikut. Satu set job $J = \{J_1, J_2, \dots, J_n\}$ akan diproses pada mesin $M = \{M_1, M_2, \dots, M_m\}$. Setiap job J mempunyai urutan operasi pada mesin M_1, M_2, \dots, M_m . Setiap mesin M_j hanya dapat menjalankan satu operasi dan setiap job hanya dapat dijalankan di satu mesin. Mesin tidak secara kontinyu siap untuk memproses job jika mesin sedang mengalami maintenance. Hal ini yang disebut terjadinya *unavailability period*. Pada studi kasus yang akan dibahas pada setiap proses yang akan dikerjakan oleh mesin bersifat *non preemptive*, setelah setiap proses dimulai untuk dikerjakan di satu proses maka proses tersebut tidak dapat di interupsi oleh proses *maintenance* atau operasi yang lainnya. Tujuan dari pemecahan permasalahan penjadwalan produksi adalah untuk menentukan urutan penjadwalan secara tepat sesuai dengan *constraints* diatas dan mempunyai makespan yang minimal.

Ada dua model yang dapat digunakan untuk menyelesaikan permasalahan penjadwalan produksi. Model yang pertama disebut *fix maintenance*, posisi dari *unavailability period* dibuat *fix* sehingga tugas utama dari model ini adalah menentukan jadwal semua job yang sehingga tidak bertabrakan dengan *unavailability period time*. Pada model yang kedua disebut *time window*, berdasarkan beberapa observasi yang dilakukan, pada kasus *non preemption* akan menyebabkan terjadinya *idle time* setiap proses akan memasuki kondisi *maintenance*. Jadi untuk meminimalkan kondisi *idle time*, *maintenance period* tidak dibuat secara pasti tetapi ditentukan dengan jangka waktu tertentu dalam setiap penjadwalan. Jadi begitu proses akan mengalami *idle* dan proses tersebut berada pada jangka waktu *maintenance period* maka proses akan segera masuk ke kondisi *maintenance* tanpa kehilangan waktu.

3. Metode Penyelesaian Masalah

Permasalahan optimasi dalam lingkup sebuah sistem manufaktur biasanya mengenai pengalokasian sumber daya yang jumlahnya terbatas dalam beberapa bidang pekerjaan. Pendekatan model matematis dinilai kurang efektif apabila kasus yang diselesaikan sangat kompleks dan rumit. Dalam tugas akhir ini akan dibahas mengenai penyelesaian kasus *scheduling* dalam lingkungan *flowshop* menggunakan teknik pencarian random terstruktur yaitu algoritma genetik.

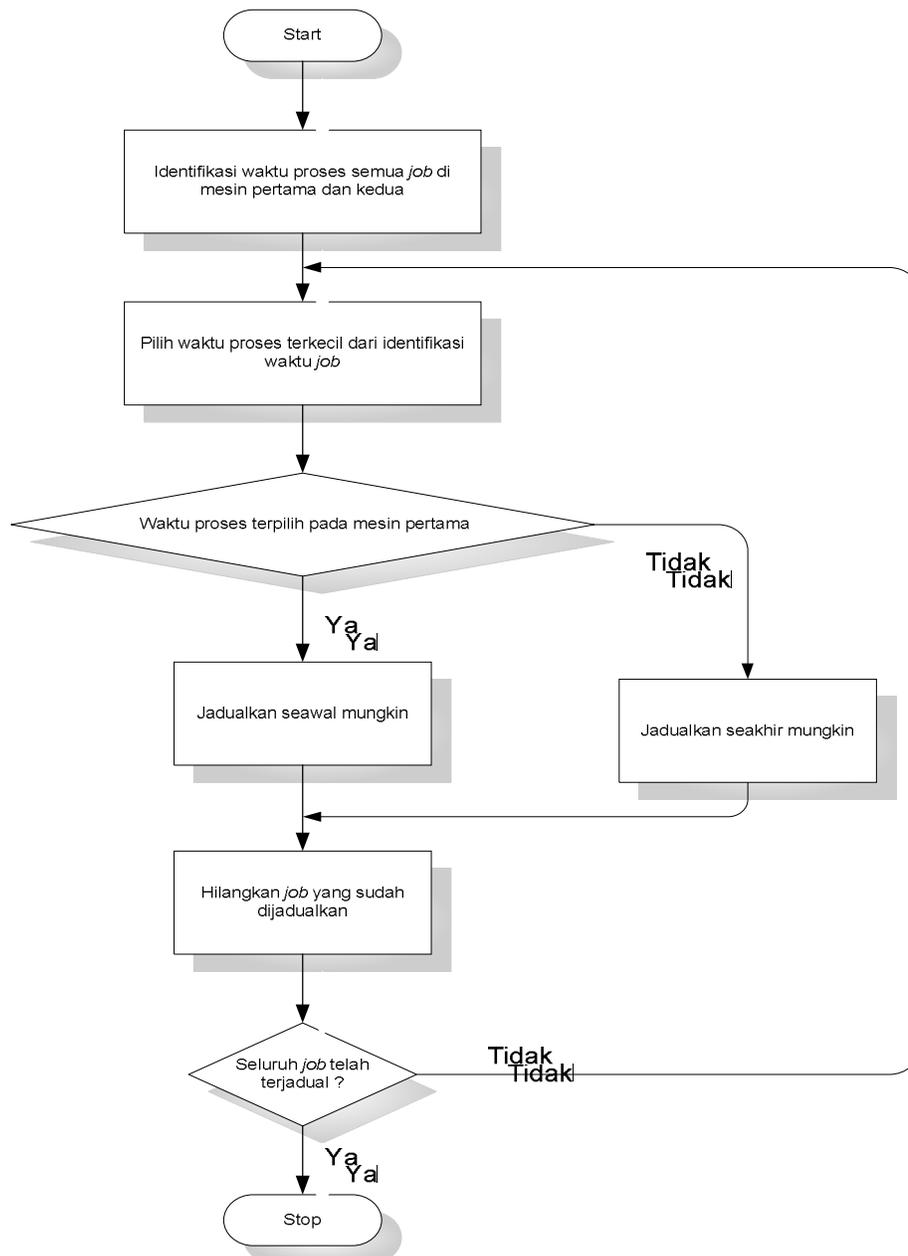
Masalah penjadwalan *flowshop* adalah menjadwalkan proses produksi dari masing-masing n job yang mempunyai urutan proses produksi dan melalui m mesin yang sama. Beberapa penelitian yang telah dilakukan antara lain: Campbell et al. (1970), Dannenbring (1977), Nawaz et al. (1983), Ogbu & Smith (1990), serta Widner & Hertz (1989). Salah satu algoritma yang bisa digunakan untuk menyelesaikan permasalahan-permasalahan penjadwalan dengan kriteria di atas, adalah algoritma CDS. Algoritma CDS biasanya digunakan untuk menyelesaikan permasalahan penjadwalan pada lingkungan *flowshop* dengan n job dan m mesin yang tidak bisa diselesaikan dengan menggunakan aturan Johnson biasa. Dalam langkah awal, algoritma CDS berusaha untuk meminipulasi jumlah mesin yang dipunyai menjadi beberapa pasangan *pseudo-workstation*, yang kemudian baru diselesaikan dengan menggunakan aturan Johnson. Sedangkan penetapan solusi optimalnya adalah dengan cara memilih solusi yang dihasilkan dari langkah awal tadi yang mempunyai *makespan* yang paling kecil.

Tujuan penjadwalan secara umum adalah sebagai berikut:

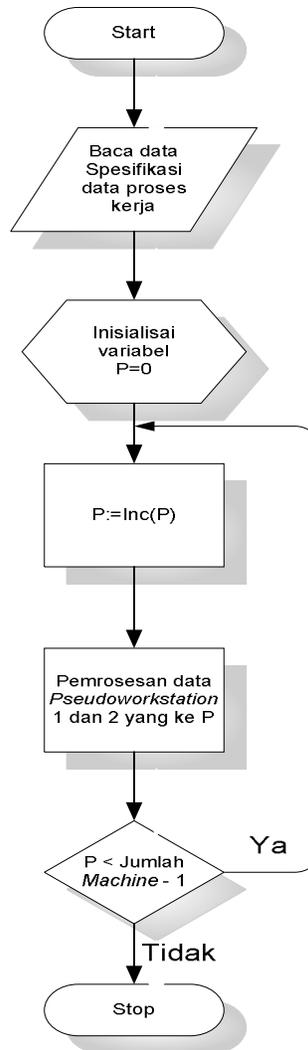
- a. Meningkatkan produktifitas mesin, yaitu dengan mengurangi *idle time* (waktu dimana *resource* tidak dalam kondisi “bekerja”) *machine*.
- b. Mengurangi persediaan barang setengah jadi (*work in process inventory*) dengan jalan mengurangi jumlah rata-rata pekerjaan yang menunggu dalam antrian suatu mesin karena mesin tersebut sibuk.

- c. Mengurangi keterlambatan karena batas waktu (*due date*) telah dilewati dengan cara mengurangi maksimum keterlambatan maupun dengan mengurangi jumlah pekerjaan yang terlambat.
- d. Meminimalkan biaya produksi.

Algoritma yang digunakan akan dihibridisasi dengan algoritma CDS untuk mendapatkan solusi yang optimal dari kasus penjadualan yang dihadapi. Algoritma CDS digunakan untuk mengeksploitasi secara lokal kromosom-kromosom dalam sebuah generasi, sedangkan algoritma genetik digunakan untuk mengeksploitasi secara global kromosom-kromosom dalam semua generasi. Dalam penelitian ini, akan digunakan sebuah sistem kecerdasan buatan, dengan menggunakan algoritma genetika dan tabu search yang kemudian hasilnya akan dibandingkan, untuk menyelesaikan kasus penjadualan produksi pada lingkungan *flowshop* untuk mencari solusi penjadualan dengan *makespan* yang minimum.



Gambar 1. Proses pengurutan Johnson



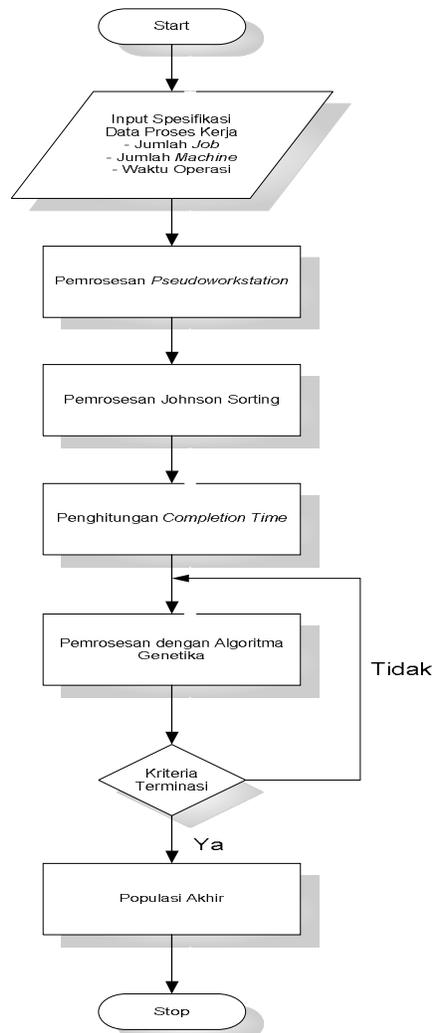
Gambar 2. Pemrosesan algoritma CDS

3.1 Algoritma Genetika

Algoritma genetika merupakan sebuah algoritma pencarian yang dikembangkan berdasarkan mekanika seleksi alami dan genetika alami oleh Holland dan dilanjutkan oleh Goldberg.

Algoritma genetika bekerja menggunakan kode dari parameter yang menjadi permasalahan. Parameter problem tersebut dikodekan menjadi sebuah kromosom. Setiap kromosom terdiri dari bagian-bagian yang disebut gen. Gen - gen ini dapat berupa angka biner ('1' atau '0'), sehingga kromosom yang terbentuk merupakan deretan string biner. Pencarian solusi pada algoritma ini melibatkan sejumlah populasi dari titik-titik pada ruang parameter. Setiap titik tersebut disebut individu.

Variabel dan parameter yang digunakan yaitu *fitness* (nilai hasil evaluasi *objective function*) untuk menentukan tingkat kesesuaian suatu individu dengan kriteria yang ingin dicapai, populasi jumlah individu per generasi, probabilitas terjadinya *crossover* pada suatu generasi, probabilitas terjadinya mutasi, serta banyaknya generasi yang dilibatkan. Algoritma akan dihentikan setelah mencapai 1000 generasi.

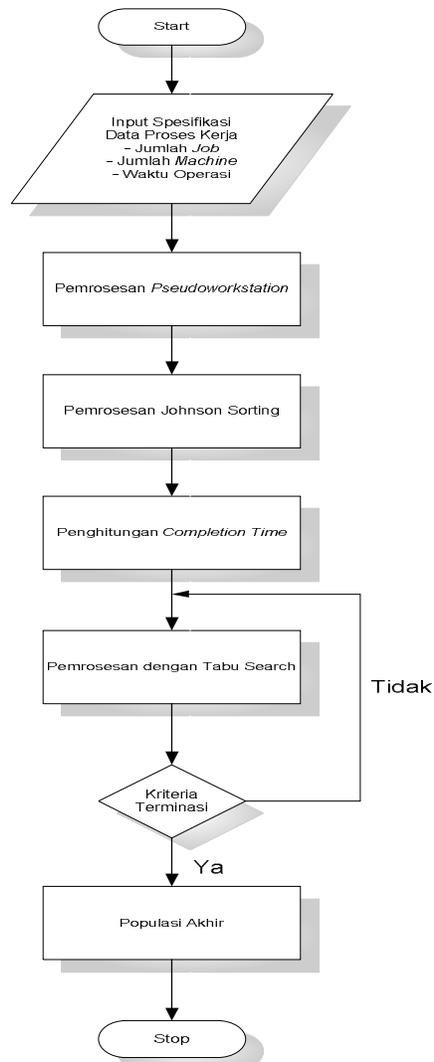


Gambar 3. Primary Hybrid GA [7]

3.2 Tabu search

Tabu search (TS) digunakan untuk menyelesaikan permasalahan kombinatorial untuk problem optimasi. Tujuan dari metode ini adalah untuk mendapatkan ruang solusi tanpa terjebak dalam *local optima* dan tanpa berputar-putar terlalu lama dengan algoritma. TS menggunakan pendekatan iterasi yang dimulai dari *initial solution* (*feasible* atau tidak) dan mencoba untuk memperbaikinya sampai tercapai kondisi stop tertentu. Pada setiap iterasi, hasil terbaik dari nilai tetangga (*neighborhood*) dipilih untuk mendapatkan nilai solusi berikutnya. Atribut dari nilai terbaik disimpan di list tabu T, dan sisanya disimpan pada sisa |T|, dengan |T| adalah ukuran list tabu. Informasi yang didapat dari list tabu digunakan untuk melakukan restriksi pengulangan tertentu. Kajian lebih jauh tentang hal ini bisa dilihat di [5] dan [6].

Pada penelitian ini, tabu search di hibridkan ke algoritma CDS, sama dengan yang dilakukan pada algoritma genetika. Urutan dari nilai tetangga (*neighborhood*) N didapat dengan melakukan permutasi posisi dari pasangan job J_i dan J_j , yang berada di posisi i dan j . Pasangan (J_i, i) dan (J_j, j) diinputkan ke tabu list T sehingga job J_i (berurut J_j) tidak akan kembali ke posisi i (berurut j) selama iterasi |T|. Ukuran dari list tabu di samakan dengan jumlah job. Tabu search dihentikan setelah 1000 iterasi.



Gambar 4. *Primary hybrid tabu search*

4. Hasil Percobaan dan Pembahasan

Untuk dapat mengevaluasi perbandingan performansi dari algoritma genetika dan tabu search, akan dilakukan pembangkitan data secara acak. Pembangkitan data di buat mendekati kasus nyata dalam dunia industri. Sebagai contoh Taillard [11] melakukan pemrosesan data job sebanyak 100, suatu hal yang akan jarang dijumpai pada kasus nyata di industri. Kemudian data tersebut akan diproses menggunakan algoritma CDS, yang hasilnya akan di umpankan ke algoritma genetika dan tabu search sebagai pembanding untuk mendapatkan hasil terbaik.

Data sample akan dibagi menjadi tiga kategori, setiap kategori mempunyai 5 sample. Jumlah job dari tiap sample adalah 10, 10 dan 20 secara berurutan. Jumlah mesin 5 untuk kategori sample pertama dan 10 mesin untuk kategori yang lain. Waktu pemrosesan job diambil secara acak dari set $\{0,1,2,\dots, 20\}$ dengan ketelitian dua digit di belakang koma yang juga diacak dan waktu pemrosesan lebih dari 0. Pada setiap mesin terdapat dua jenis *maintenance*. Dilakukan pengacakan untuk menentukan *maintenanceperiod* tetapi dengan menggunakan suatu pola, untuk setiap mesin setelah terjadi *maintenance* yang pertama minimal harus ada satu operasi baru kemudian *maintenance* yang kedua bisa dilakukan. Sebaliknya untuk *maintenance* yang kedua maksimal berada pada urutan sebelum operasi

yang terakhir. Waktu rata-rata *maintenance* pada sebuah mesin didapat dari waktu rata-rata operasi pada mesin tersebut.

Pada tabel pertama terdapat didapat *initial* yang merupakan waktu awal setelah diproses menggunakan algoritma CDS tetapi sebelum di hibridkan ke algoritma genetika, *best* merupakan waktu terbaik dan *mean* yang merupakan waktu rata-rata yang didapat dari hasil *running* menggunakan algoritma genetika. Terdapat dua group pada setiap kategori, kategori pertama merupakan hasil pemrosesan dengan algoritma genetika dan dengan menggunakan penjadualan *maintenance* bersifat *fix* atau tetap (GAF). Untuk tipe kedua menggunakan *time window* atau penentuan menggunakan jangka waktu tertentu untuk menghindari *idle time* (GAD).

Tabel 1. Hasil dari pencarian makespan menggunakan algoritma genetika

Problem	GAF			GAD		
	Initial	Best	Mean	Initial	Best	Mean
AC1	67,45	65,39	66,23	60,68	55,23	57,45
AC2	60,56	58,38	58,97	55,34	50,34	53,33
AC3	70,78	65,57	67,44	66,67	59,34	63,24
AC4	50,79	47,45	48,33	45,53	40,53	42,32
AC5	60,57	55,34	56,53	54,36	52,34	53,43
AC6	138,67	130,87	131,43	137,22	128,22	127,66
AC7	120,03	115,79	118,82	115,88	114,73	115,34
AC8	130,24	125,67	127,35	133,68	132,04	132,97
AC9	112,25	109,25	110,83	103,27	99,87	100,78
AC10	133,67	129,78	133,98	122,89	120,76	121,27
AC11	229,08	220,33	225,44	210,35	206,23	208,21
AC12	210,34	207,56	208,23	197,37	190,32	194,43
AC13	209,87	205,34	207,53	195,38	189,32	192,31
AC14	230,84	227,34	228,56	220,48	212,23	216,23
AC115	190,34	187,29	188,31	180,38	169,32	176,32

Pada tabel pertama terlihat bahwa terjadi perbaikan makespan time pada kolom *intial* GAD jika dibandingkan dengan kolom *intial* GAF. Terlihat terjadi perubahan secara signifikan pada makespan yang dihasilkan pada *time window* jika dibandingkan dengan *fix period*.

Pada tabel kedua terdapat didapat *initial* yang merupakan waktu awal setelah diproses menggunakan algoritma CDS sebelum dihibridkan ke algoritma genetika, *best* merupakan waktu terbaik dan *mean* yang merupakan waktu rata-rata yang didapat dari hasil *running* menggunakan tabu search. Terdapat dua group pada setiap kategori, kategori pertama merupakan hasil pemrosesan dengan tabu search dan dengan menggunakan penjadualan *maintenance* bersifat *fix* atau tetap (TSF). Untuk tipe kedua menggunakan *time window* atau penentuan menggunakan jangka waktu tertentu untuk menghindari *idle time* (TSD).

Tabel 2. Hasil dari pencarian makespan menggunakan tabu search

Problem	TSF			TSD		
	Initial	Best	Mean	Initial	Best	Mean
AC1	93,34	89,34	90,43	89,34	75,43	80,65
AC2	88,23	82,34	84,23	83,28	74,19	78,95
AC3	92,32	80,43	85,34	84,83	70,34	77,75
AC4	70,89	64,43	68,43	68,39	44,82	50,48
AC5	80,28	77,32	78,43	77,27	60,86	67,74
AC6	132,23	128,33	130,34	128,39	119,14	123,58
AC7	118,03	110,31	113,36	108,34	100,38	103,29
AC8	133,49	125,48	129,32	125,38	106,27	116,34
AC9	108,29	100,32	103,58	103,77	90,28	97,29
AC10	138,237	128,42	130,43	130,23	119,27	124,28
AC11	240,39	228,38	236,58	228,27	213,37	219,22
AC12	250,29	234,34	241,34	238,21	210,28	229,25
AC13	244,27	230,45	235,98	228,31	204,48	209,34
AC14	289,12	270,37	278,64	268,12	247,46	255,76
AC115	260,29	245,47	255,34	245,12	224,47	238,84

Dapat kita lihat dari tabel 2 tabu search terlihat sedikit lebih stabil jika dibandingkan dengan algoritma genetika. Pada kasus ini TSD terlihat mengalami perubahan optimalisasi makespan jika dibandingkan TSF.

5. Kesimpulan

Penelitian ini mencoba mencari cara terbaik untuk mengoptimalkan makespan time dari proses penjadwalan produksi jika penjadwalan harus mengalami *maintenance*. Pada kasus *non-preemptive* diujicobakan dua model penanganan masalah *maintenance*, yaitu *fix period* dan *window time* dengan menggunakan algoritma CDS yang di hibridkan ke algoritma genetika dan tabu search. Berdasarkan proses *benchmarking*, yang dilakukan terhadap data yang acak dengan suatu pola tertentu dengan menggunakan dua algoritma tersebut maka didapatkan hasil bahwa penggunaan *time window* untuk kasus *availability constraint* pada penjadwalan produksi flowshop ternyata memberikan peningkatan yang cukup signifikan.

Daftar Pustaka

- [1] Aggune, Riad, *Minimizing the Makespan for the Flow Shop Scheduling Problem with Availability Constraints*, ORP, Paris, September 2001.
- [2] Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.
- [3] Gen, Mitsuo, Runwei Cheng, *Genetic Algorithms and Engginering Design*, John Wiley & Sons, New York, 1997.
- [4] Gen, Mitsuo, Yasuhiro Tsujimura, dan Yuichiro Mafune, *Effects of Symbiotic Evolution in Genetic Algorithms for Job-Shop Scheduling*,
- [5] Glover, Taillard and D. Werra, *A User's Guide to Tabu Search*, Annals of Operation Research, Vol 41, pp. 3-28, 1993.
- [6] Glover, *Tabu Search Fundamental and Uses*, Graduate School of Business, University of Colorado, Bolder, 1995
- [7] Indarto, Wawan, *Penjadwalan Produksi Flowshop dengan Steady State Multiple Pbjective Algorithm (MOEGA)*, Media Informatika, 2003
- [8] Manongga, D., Handok, *Penggunaan Algoritma Genetika dalam Estimasi Fungsi Non Linier*, Proceeding SITIA 200, Surabaya 2000.
- [9] Purnomo, Hari dan M. R. Andi P, *Penerapan Algoritma Genetik dalam Penyelesaian Kasus Penjadwalan Produksi*, Proceeding Seminar Nasional TIMP, Surabaya, 2001.
- [10] Saito, Osami, et. Al, *Unjuk Kinerja Algoritma Genetika Berbasis Floating-Point*, Prosiding Kecerdasan Komputasional, UI, Jakarta 1999.
- [11] Taillard, *Bench-marks for basic schedulling instances*, European Journal of Operational Research, Vol. 64, pp. 278-285, 1993.
- [12] Wahjudi, Didik, dan Tessa Vani19480723na Soetanto, *Aplikasi Algoritma Genetika untuk Penjadwalan Produksi Flowshop*, Prosiding Kecerdasan Komputasional, UI, Jakarta 1999.
- [13] Wall, Matthew Bartschi, *A Genetic Algorithm for Resource-Constrained Scheduling – A Desertation*, Departement of Mechanical Engginering Massachusetts Institute of Technology, 1996.