

# PENJADWALAN *FLOWSHOP* DENGAN MENGGUNAKAN *SIMULATED ANNEALING*

Muhammad Firdaus, Ilyas Masudin, Dana Marsetya Utama  
Jurusan Teknik Industri - Universitas Muhammadiyah Malang  
masudin@umm.ac.id

## ABSTRACT

*This article apply a machine scheduling technique, named Simulate Annealing (SA) to schedule 8 jobs and 5 machines to minimize makespan. A flowshop production flow is chosen as a case study to collect data and attempted to reduce jobs' makespan. This article also does a sensitivity analysis to explore the implication of the changes of SA parameters as temperature. The results shows that the completion time of the jobs uses SA algorithm can decrease the completion time of the jobs, about 5 hours lower than the existing method. Moreover, total idle time of the machines is also reduced by 2.18 per cent using SA technique. Based on the sensitivity analysis, it indicates that there is a significant relationship between the changes of temperatures and makespan and computation time.*

**Keywords :** *scheduling, simulated annealing, makespan flow shop.*

## I. PENDAHULUAN

Hampir semua permasalahan utama dari rantai produksi flowshop sebuah perusahaan adalah masalah penjadwalan produksi, terutama dalam menentukan efisiensi urutan pekerjaan (jobs) yang akan dikerjakan. Kegagalan dalam menjadwalkan urutan pengerjaan jobs pada mesin akan mengakibatkan keterlambatan penyelesaian jobs dan rendahnya utilitas mesin yang pada akhirnya akan mengakibatkan biaya produksi yang tinggi. Oleh karena itu, metode penjadwalan mesin bisa menjadi salah satu alternative cara untuk menyelesaikan permasalahan penjadwalan job dan mesin. Salah satu metode penjadwalan job dan mesin yang cukup populer adalah Simulated Annealing (SA). Menurut Ratnasari (2007), masalah penjadwalan produksi *flow shop* terbukti dapat diselesaikan menggunakan algoritma *Simulated Annealing (SA)* dengan tujuan meminimalkan *makespan* dan total *flow time*, selain itu menurut Santoso (2012), algoritma *Simulated Annealing* dapat diterapkan dengan baik dalam penjadwalan produksi yang digolongkan ke dalam *flow shop*. Tidak hanya dapat diterapkan, algoritma ini bahkan sangat efektif karena mampu mengatasi penjadwalan dengan jumlah *job* banyak (15 job) dengan waktu proses yang *relative* cepat dan hasil penjadwalan yang baik, dan juga menurut Damanik (2011), dengan penjadwalan usulan dengan metode algoritma *Simulated Annealing* menghasilkan makespan 13% lebih kecil dari penjadwalan awal dan dengan metode ini penjadwalan *job* dapat terpenuhi, dan tidak terjadi keterlambatan.

Dengan mengacu pada permasalahan diatas dan juga pada penelitian – penelitian sebelumnya, maka artikel ini berusaha memberikan usulan penyelesaian problem penjadwalan jobs pada rantai produksi flowshop dengan Simulated Annealing (SA). Artikel ini berbasis case study yang akan merencanakan penjadwalan produksi agar bisa meminimalkan waktu penyelesaian produksi, sehingga perusahaan dapat mengirimkan barang sesuai dengan waktu yang ditentukan dan dapat meminimalkan biaya yang dikeluarkan karena mesin yang digunakan dapat digunakan dengan *optimal*. Pada penelitian ini, penjadwalan produksi menggunakan metode *simulated annealing (SA)* sedangkan teknik pengacakannya menggunakan yaitu metode *swap* (menukar), *flip* (membalik), dan *slide* (mengeser). dengan menggunakan bantuan *software matlab* diharapkan bisa menyelesaikan permasalahan yang dialami perusahaan.

## II. LANDASAN TEORI

### A. Studi Pustaka

Menurut Baker (1974), pengertian penjadwalan secara umum didefinisikan sebagai proses pengalokasian sumber untuk memilih sekumpulan tugas dalam jangka waktu tertentu. Sedangkan menurut Ginting (2009), penjadwalan adalah pengurutan pembuatan/pengerjaan produk secara menyeluruh yang dikerjakan pada beberapa buah mesin. Dengan demikian masalah *sequencing* senantiasa melibatkan pengerjaan sejumlah komponen yang sering disebut dengan istilah '*job*'. *Job* sendiri masih merupakan komposisi dari sejumlah elemen-elemen dasar yang disebut aktivitas atau operasi ini membutuhkan alokasi sumber daya tertentu selama periode waktu tertentu yang sering disebut dengan waktu proses.

Permasalahan muncul apabila ada tahapan operasi tertentu beberapa atau seluruh pekerjaan itu membutuhkan stasiun kerja yang sama. Dengan dilakukannya pengurutan pekerjaan ini unit-unit produksi (*resource*) dapat dimanfaatkan secara *optimum*. Pemanfaatan ini antara lain dilakukan dengan jalan meningkatkan *utilitas* unit-unit produksi melalui usaha-usaha *mereduksi* waktu menganggur (*idle time*) dari unit-unit yang bersangkutan. Juga dapat dilakukan dengan cara meminimumkan *in-process inventory* melalui *reduksi* terhadap waktu rata – rata pekerjaan yang menunggu dalam baris antrian pada unit – unit produksi.

Menurut Uttari (2008), beberapa istilah umum yang digunakan dalam penjadwalan produksi adalah sebagai berikut :

*Processing time* ( $t_i$ )

Merupakan perkiraan waktu penyelesaian satu pekerjaan. Perkiraan waktu ini meliputi juga perkiraan waktu set-up yang dibutuhkan. Symbol yang digunakan untuk waktu proses pekerjaan I adalah  $t_i$ .

*Makespan* ( $M_s$ )

Adalah jangka waktu penyelesaian suatu penjadwalan yang merupakan jumlah seluruh waktu proses.

$$M_s = \sum_{i=1}^n t_i$$

*Flow time* ( $F_i$ )

Merupakan rentang waktu antara saat pekerjaan tersedia (dapat dimulai) dan saat pekerjaan selesai. Waktu alir sama dengan waktu proses ditambah waktu tunggu sebelum pekerjaan diproses.

$$F_i = t_i + W_i$$

$$F_i = C_i + r_i$$

*Completion time* ( $C_i$ )

Merupakan rentang waktu antara saat pekerjaan dimulai ( $t=0$ ), sampai dengan pekerjaan itu selesai.

$$C_i = F_i + r_i$$

Rata-rata flow time

$$\bar{F} = \frac{1}{n} \sum_{f=1}^n F_j$$

*Due date* ( $d_i$ )

Merupakan waktu maksimal yang dapat diterima untuk menyelesaikan pekerjaan tersebut. Kelebihan waktu dari waktu yang ditetapkan, merupakan suatu kelambatan.

*Lateness* ( $L_i$ )

Merupakan penyimpangan antara waktu penyelesaian pekerjaan dengan batas waktu. Suatu pekerjaan akan mempunyai kelambatan positif jika diselesaikan sesudah batas waktu dan kelambatan negatif jika diselesaikan sebelum batas waktu.

$L_i = C_i - d_i$   
 $L_i < 0$ , saat penyelesaian memenuhi batas akhir  
 $L_i > 0$ , saat penyelesaian melewati batas akhir

Rata-rata *lateness*

$$\bar{L}_s = \sum_{j=1}^n (C_j - d_j)$$

*Tardiness* ( $T_i$ )

Merupakan ukuran kelambatan positif. Jika suatu pekerjaan diselesaikan lebih cepat dari batas waktu yang ditetapkan, maka mempunyai nilai kelambatan negatif tetapi ukuran kelambatan positif. Ukuran ini disimbolkan dengan  $T_i$  dimana  $T_i$  adalah maksimum dari  $(0, L_i)$ .

Rata-rata *tardiness*

$$\bar{T}_s = \frac{1}{n} \sum_{j=1}^n T_j$$

*Slack time* ( $SL_i$ )

Merupakan ukuran yang digunakan untuk melihat selisih waktu antara waktu proses dengan batas waktu yang sudah ditetapkan.

$$S_i = d_i - t_i$$

## B. Simulated Annealing (SA)

Santosa dan Willy (2011), *Annealing* adalah satu teknik yang dikenal dalam bidang *metalurgi*, digunakan dalam mempelajari proses pembentukan *kristal* dalam suatu materi. Agar dapat terbentuk susunan *kristal* yang sempurna, diperlukan pemanasan sampai suatu tingkat tertentu, kemudian dilanjutkan dengan pendinginan yang perlahan-lahan dan terkendali dari materi tersebut. Pemanasan materi di awal proses *annealing*, memberikan kesempatan pada *atom-atom* dalam materi itu untuk bergerak secara bebas, mengingat tingkat energi dalam kondisi panas ini cukup tinggi. Proses pendinginan yang perlahan-lahan memungkinkan *atom-atom* yang tadinya bergerak bebas itu, pada akhirnya menemukan tempat yang *optimum*, di mana energi internal yang dibutuhkan *atom* itu untuk mempertahankan posisinya adalah *minimum*.

*Simulated Annealing* berjalan berdasarkan *analogi* dengan proses *annealing* yang telah dijelaskan diatas. Pada awal proses SA, dipilih suatu solusi awal, yang merepresentasikan kondisi materi sebelum proses dimulai. Gerakan bebas dari *atom-atom* pada materi, direpresentasikan dalam bentuk *modifikasi* terhadap solusi awal/solusi sementara. Pada awal proses SA, saat *parameter* suhu ( $T$ ) diatur tinggi, solusi sementara yang sudah ada diperbolehkan untuk mengalami *modifikasi* secara bebas.

Kebebasan ini secara relatif diukur berdasarkan nilai fungsi tertentu yang mengevaluasi seberapa *optimal* solusi sementara yang telah diperoleh. Bila nilai fungsi *evaluasi* hasil *modifikasi* ini membaik (dalam masalah optimisasi yang berusaha mencari minimum berarti nilainya lebih kecil/*downhill*) solusi hasil *modifikasi* ini akan digunakan sebagai solusi selanjutnya. Bila nilai fungsi *evaluasi* hasil *modifikasi* ini memburuk, pada saat *temperatur annealing* masih tinggi, solusi yang lebih buruk (*uphill*) ini masih mungkin diterima. Dalam tahapan selanjutnya saat *temperatur* sedikit demi sedikit dikurangi, maka kemungkinan untuk menerima langkah *modifikasi* yang tidak memperbaiki nilai fungsi evaluasi semakin berkurang. Sehingga kebebasan untuk *memodifikasi* solusi semakin menyempit, sampai akhirnya diharapkan diperoleh solusi yang mendekati solusi *optimal*.

Metode SA meniru proses pendinginan secara perlahan baja / metal yang mendidih untuk mencapai nilai *minimum* fungsi dalam permasalahan minimasi. Proses pendinginan ini ditiru dengan cara menentukan *parameter* yang serupa dengan suhu lalu mengontrolnya dengan menggunakan dengan menggunakan konsep *distribusi probabilitas Boltzmann*. *Distribusi probabilitas Boltzmann* menyatakan bahwa energi ( $E$ ) dari suatu system dalam keseimbangan panas pada suhu  $T$  terdistribusi secara *probabilistik* mengikuti rumus

$$P(E) = e^{-E/kT}$$

Dimana  $P(E)$  menyatakan peluang mencapai tingkat energi  $E$ ,  $T$  adalah dan  $k$  konstanta Boltzmann. Pada persamaan ini menunjukkan bahwa jika proses pencairan solusi mengikuti *distribusi probabilitas Boltzmann konvergensi* algoritma SA dapat diatur dengan menggunakan *temperatur T*.

Metode yang *mengimplementasikan distribusi probabilistic Boltzmann* dalam *simulated thermodynamic systems*, seperti diusulkan Metropolis dkk yang dikutip dalam (Rao,2009), bisa juga ditetapkan dalam korteks minimasi fungsi. Dalam *minimasi* fungsi, misalkan solusi yang sekarang adalah  $x$  dan nilai fungsinya  $f(x)$ , mirip dengan status energi pada *system termodinamika*, energi  $E_i$  pada status  $x_i$  adalah

$$E_i = f_i = f(x_i)$$

Menurut kriteria Meropolis, *probabilistic* titik solusi berikutnya adalah  $x_{i+1}$  bergantung pada perbedaan status energi atau fungsi tujuan dia dua titik (status) diberikan oleh

$$P[E_{i+1}] = \min \{1, e^{-\Delta E/kT}\},$$

Dimana

$$\Delta E = E_{i+1} - E_i = \Delta f = f_{i+1} - f_i = f(x_{i+1}) - f(x_i)$$

Titik baru bisa ditemukan dengan menggunakan *distribusi probabilitas Boltzmann* tersebut. Untuk faktor Boltzmann,  $k$ , bisa diberi nilai 1. Jika  $\Delta E \leq 0$ , maka  $P[E_{i+1}] = 1$  sehingga titik  $x_{i+1}$  selalu diterima. Ini adalah pilihan yang masuk akal dalam *konteks minimasi* fungsi, yaitu jika  $f(x_{i+1}) \leq f(x_i)$  maka  $x_{i+1}$  pasti diterima. Dilain pihak jika  $\Delta E > 0$ , maka nilai  $f(x_{i+1})$  akan lebih besar (lebih buruk) dari  $f(x_i)$ . dalam *optimasi konvensional* jika ini terjadi maka  $x_{i+1}$  akan ditolak. Tetapi dalam SA ini masih diterima dengan *Probabilitas P*

$$P[E_{i+1}] = e^{-\Delta E/kT}$$

Dari persamaan diatas bisa dilihat bahwa pada temperatur  $T$  tinggi maka peluang menerima  $x_{i+1}$  dengan  $\Delta E$  yang lebi besar akan besar. Dengan menurunnya *temperatur T probabilitas* untuk menerima titik  $x_{i+1}$  yang lebih buruk dari titik sebelumnya akan mengecil. Sehingga jika temperatur semakin rendah (semakin dekat ke titik optimal), peluang suatu solusi  $x_{i+1}$  dengan nilai  $f$  lebih besar dibanding pada titik  $x_i$  akan semakin kecil.

### C. Algoritma Simulated Annealing

Santosa dan Willy (2011), Algoritma SA dapat dijelaskan secara ringkas sebagai berikut. Algoritma *Simulated Annealing* dimulai dengan suatu vektor solusi  $x_1$  (iterasi  $i = 1$ ) dan nilai temperature  $T$  yang cukup tinggi. Bangkitkan vector solusi baru secara random yang dekat dari titik sekarang dan hitung perbedaan nilai fungsi tujuannya :

$$\Delta E = E_{i+1} - E_i = \Delta f = f_{i+1} - f_i = f(x_{i+1}) - f(x_i)$$

Jika  $f_{i+1}$  lebih kecil dari  $f_i$  (dengan nilai  $\Delta f$  negatif), terima titik  $f_{i+1}$  sebagai titik solusi baru. Sebaliknya, jika nilai  $\Delta f$  positif, *probabilitas* menerima  $x_{i+1}$  sebagai solusi baru adalah  $e^{-\Delta f/kT}$ . Untuk menerima atau tidak, perlu dicari pembandingan terhadap nilai *probabilitas* ini. Kita perlu membangkitkan bilangan random antara 0 sampai 1.

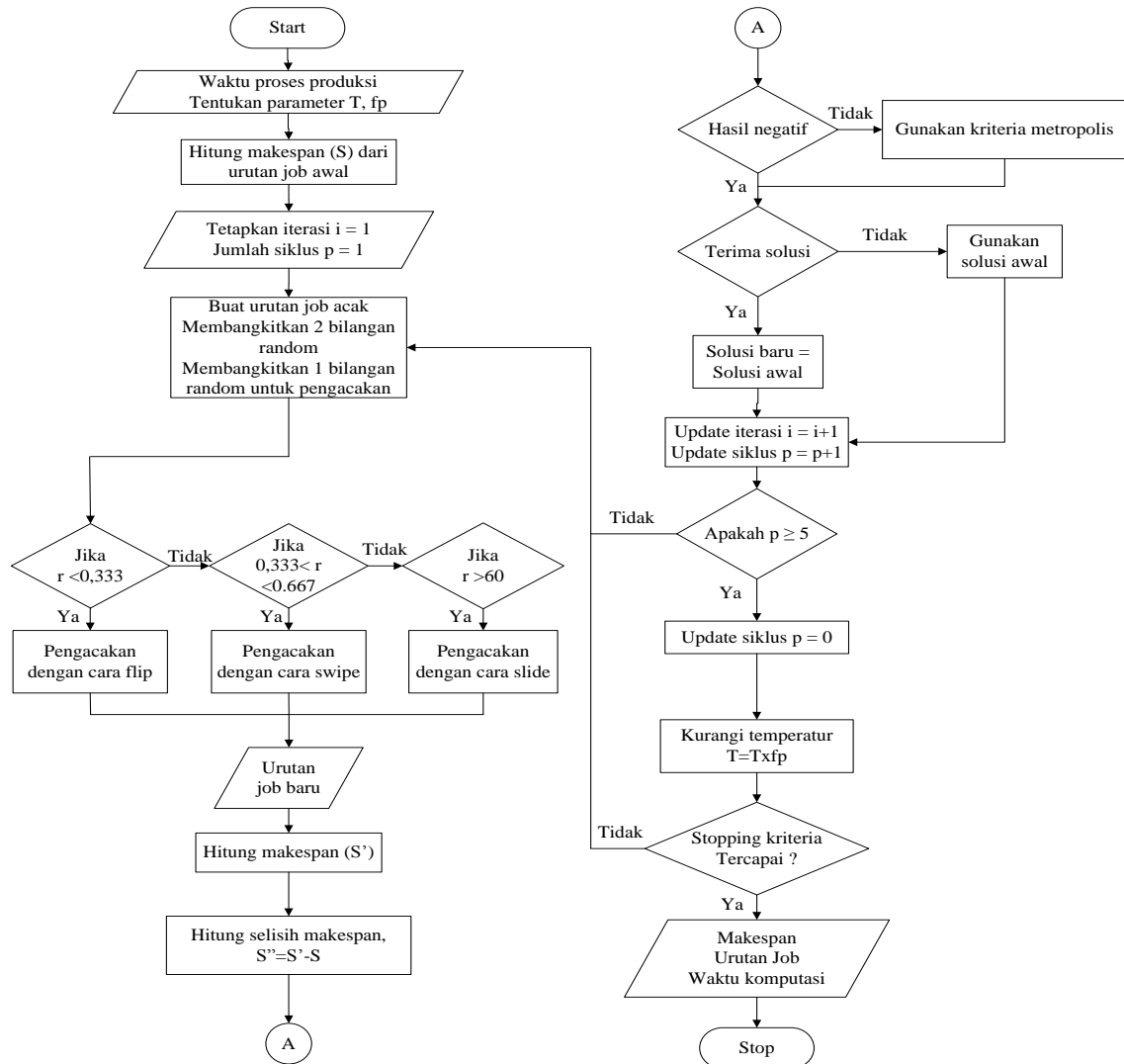
Apabila nilai random yang dibangkitkan ini lebih kecil dari nilai  $e^{-\Delta f/kT}$ , terima titik  $x_{i+1}$ ; Sebaliknya, tolak  $x_{i+1}$ . Sampai disini kita telah menjalani satu iterasi dari algoritma SA. Jika titik  $x_{i+1}$  ditolak, maka dilanjutkan proses pembangkitan nilai baru  $x_{i+1}$  secara *random* dalam area yang berdekatan dengan titik sekarang  $x_i$  dalam batas – batas tertentu, lalu mengevaluasi nilai fungsi tujuan  $f_{i+1}$ , dan memutuskan untuk menerima  $x_{i+1}$  sebagai titik baru, berdasarkan criteria metropolis  $P[E_{i+1}] = e^{-\Delta f/kT}$ . Untuk mensimulasikan pencapaian *equilibrium thermal* pada setiap *temperatur* tertentu T, jika jumlah titik  $x_{i+1}$  yang diuji pada sembarang temperature T melebihi nilai n, temperature T dikurangi dengan proporsi tertentu yaitu c (  $0 < c < 1$  ) dan seluruh proses diulang lagi. *Prosedur* ini diasumsikan akan mengalami *konvergensi* ketika *temperatur* T yang dicapai cukup kecil atau jika perubahan nilai fungsi tujuan ( $\Delta f$ ) sudah sangat kecil.

Pilihan nilai awal *temperatur* T, jumlah *iterasi* k sebelum mengurangi *temperatur*, dan faktor pengurangan *temperatur* c adalah parameter – parameter penting dalam keberhasilan pelaksanaan algoritma SA. Jika misalnya nilai awal *temperatur* terlalu tinggi, maka dibutuhkan lebih banyak pengurangan *temperatur* untuk *konvergen*. Sebaliknya jika nilai awal ini terlalu kecil proses pencairan ini mungkin kurang sempurna sehingga ada titik-titik *potensial* yang bisa menjadi *global optimum* terlewat. *Faktor pereduksi temperatur* c memainkan peran yang sama. Jika terlalu besar (0.8 atau 0.9) akan memerlukan terlalu banyak langkah komputasi. Sebaliknya, jika terlalu kecil nilai c (misal 0.1 atau 0.2) bisa berakibat terlalu cepatnya penurunan *temperatur* sehingga akan banyak titik-titik *potensial* untuk menjadi solusi *global* akan terlewat.

Sebaliknya jika terlalu kecil proses pencarian mungkin kurang sempurna sehingga ada titik – titik *potensial* yang bisa menjadi *global optimum* terlewat. Begitu juga jika jumlah *iterasi* k terlalu besar akan membantu mencapai keseimbangan *termal* pada setiap *temperatur* tetapi dengan tambahan jumlah *komputasi*. Jika jumlah n sedikit, maka bisa terjadi *konvergensi* yang terlalu cepat atau menuju kesolusi tetapi *local optimum*. Sayangnya tidak ada nilai T, n, dan c yang unik yang akan sesuai untuk semua problem. Tetapi ada beberapa cara untuk mendapatkan nilai yang cukup baik. Misalnya untuk T bisa dipilih dari nilai rata-rata dari fungsi tujuan yang dihitung pada sejumlah solusi awal yang dipilih secara *random*. Jumlah iterasi k bisa dipilih antara 50 dan 100 berdasarkan pada akurasi solusi yang diinginkan. *Faktor pereduksi temperatur* c bisa dipilih antara 0.4 dan 0.6 untuk strategi pengurangan *temperatur* yang masuk akal (*cooling schedule*). Gambar 1 menunjukkan langkah-langkah dari *Algoritma simulated annealing* (SA). Dalam SA, perlu diperhatikan adalah adanya langkah khusus untuk keluar dari *local optimum*. Langkah ini berupa penerimaan suatu titik  $x_{i+1}$  dengan peluang  $e^{-\Delta f/kT}$  walaupun nilai fungsi pada titik ini tidak lebih dari titik sebelumnya  $x_i$  ini dilakukan dengan harapan pada langkah selanjutnya akan dicapai suatu titik dengan nilai fungsi yang lebih baik lagi. Jika langkah khusus ini tidak ada, maka SA akan seperti teknik *optimasi konvensional* biasanya tidak bisa keluar dari *local search trap*. Hal-hal yang bisa dicatat dalam SA antara lain, pertama, solusi akhir tidak dipengaruhi solusi awal yang dimasukkan. Tetapi jika titik awal yang dimasukkan tidak cukup baik maka diperlukan waktu lebih lama untuk mencapai solusi akhirnya dan kedua, *konvergensi* tidak dipengaruhi *konveksitas* dari ruang pencarian yang (*feasible space*).

**III. METODOLOGI PENELITIAN**

Tahap pertama metodologi penelitian adalah menentukan teknik pengumpulan data, dimana dalam artikel ini beberapa data utama dikumpulkan meliputi data proses pembuatan produk yang dinyatakan sebagai job.



Gambar 1 Flowchart flow shop simulated annealing

Beberapa data yang diperlukan antara lain data mesin-mesin yang dipakai yang meliputi data mesin, jumlah mesin, spesifikasi mesin, yang berisi kapasitas mesin dan juga urutan proses pengerjaan serta waktu proses permesinan tiap job. Beberapa data tersebut didapatkan dengan langsung melakukan pengamatan atau observasi seperti data waktu permesinan tiap job dan urutan proses pengerjaan job di tiap mesin sementara beberapa data diantaranya merupakan sekunder yang disediakan oleh perusahaan seperti data kapasitas mesin dan spesifikasi mesin.

Tahap berikutnya melakukan penjadwalan dengan algoritma Simulated Annealing, dimana metode ini bertujuan untuk meminimalkan makespan dengan cara menentukan suhu awal dan penurunan suhu untuk selanjutnya dilakukan urutan job acak dengan cara membangkitkan 2 bilangan random untuk urutan mana saja yang nanti akan dirubah. Detail algoritma simulated annealing dapat dijelaskan dalam flowchart di diatas.

Pada tahap selanjutnya, dilakukan coding menggunakan software Matlab untuk dilakukan analisa sensitivitas untuk melihat pengaruh perubahan atau penurunan suhu yang dilakukan dengan suhu awal (T) 100, 200, 300, 500 1000, faktor penurunan suhu (fp) 0.2, 0.5, 0.9 dengan kriteria pemberhentian suhu 0.0000001 dan jumlah siklus dalam 1 penurunan suhu sebanyak 5 kali terhadap waktu komputasi, jumlah iterasi dan makespan.

#### IV. HASIL DAN PEMBAHASAN

Pada dasarnya keunggulan metode *Simulated Annealing* (SA) yaitu dapat mendapatkan urutan job yang lebih *optimal* karena hasil *makespan* dari perusahaan dijadikan *inputan* untuk kemudian diproses dalam penurunan suhu dimana dalam 1 *siklus* penurunan suhu tersebut didapatkan solusi baru yang nantinya akan dibandingkan dengan solusi yang lama apakah lebih baik atau tidak, jika misalkan kurang baikpun maka hasil *makespan* dari SA ini akan sama dengan *makespan* awal. Berdasarkan hasil perhitungan SA manual maupun *output* program MATLAB berbasis SA pada kasus *flow shop* telah menghasilkan solusi *optimal*, akan tetapi dalam melakukan perhitungan SA manual akan membutuhkan waktu yang sangat lama karena harus dilakukan perhitungan secara bertahap, oleh karena itu dibutuhkan program matlab untuk membantu melakukan penjadwalan produksi. Keunggulan penggunaan SA dengan program Matlab yaitu dapat mempermudah dan mempercepat dalam melakukan penyelesaian pencarian solusi yang *optimal*, dari hasil *running* Matlab didapatkan waktu *komputasi* yang sangat cepat. Sehingga dari hasil tersebut diharapkan perusahaan dapat dengan cepat mengetahui kinerja penjadwalannya.

Tabel 1 Perbandingan Hasil Penjadwalan *Komputasi* Matlab

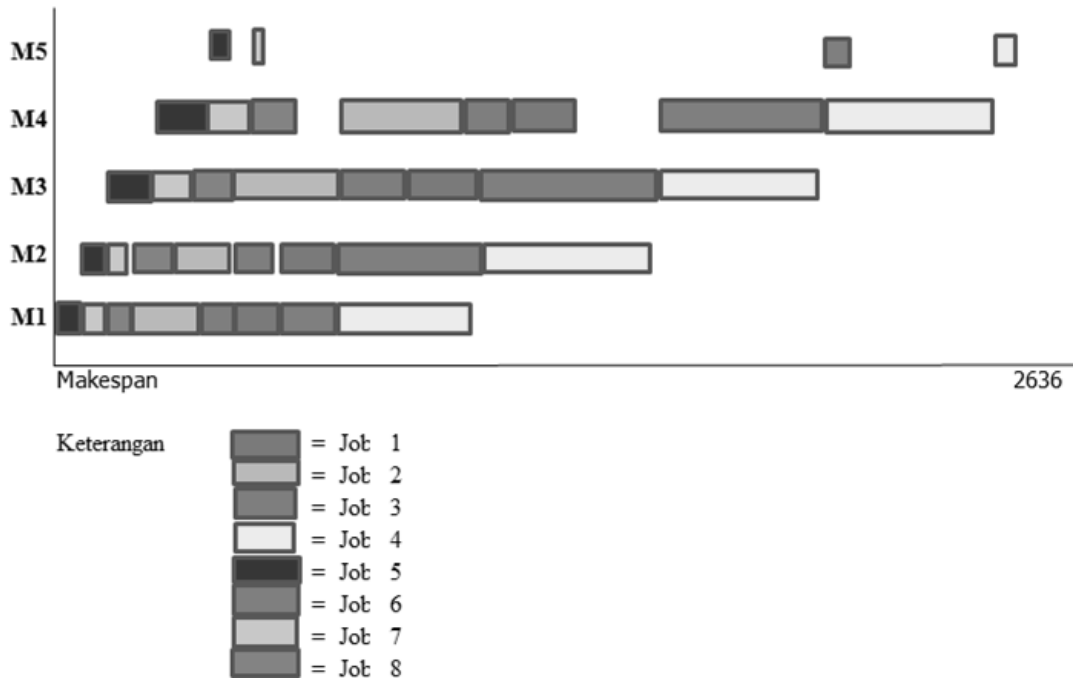
<i>Parameter</i>	<i>Makespan</i>	<i>Urutan Job</i>	<i>Iterasi</i>	<i>Waktu Komputasi</i>	
<b>T = 100</b>	fp = 0.2	2732	3-2-5-6-1-8-4-7	66	0.0156
	fp = 0.5	2636	5-7-8-2-3-1-6-4	151	0.0156
	fp = 0.9	2642	5-8-3-1-2-6-4-7	986	0.1404
<b>T = 200</b>	fp = 0.2	2702	5-2-3-8-6-4-1-7	71	0.0312
	fp = 0.5	2636	5-7-2-3-1-8-6-4	156	0.0312
	fp = 0.9	2642	5-2-8-1-3-6-4-7	1021	0.1248
<b>T = 500</b>	fp = 0.2	2710	5-2-1-7-6-4-8-3	71	0.0156
	fp = 0.5	2636	5-3-7-2-8-1-6-4	166	0.0468
	fp = 0.9	2636	5-8-3-2-7-1-6-4	1061	0.1404
<b>T = 1000</b>	fp = 0.2	2672	3-5-1-2-6-8-4-7	76	0.0312
	fp = 0.5	2636	5-3-1-8-7-2-6-4	171	0.0468
	fp = 0.9	2636	5-7-3-8-2-1-6-4	1096	0.1404

Penjadwalan SA yang terdiri dari *job1, job2, job3, job4, job5, job6, job7, job8* digunakan *parameter-parameter* seperti suhu awal (T) dan faktor *pereduksi* suhu (fp) untuk pencarian solusi yang *optimal*. Menurut Santosa dan Willy (2011), Jika misalnya nilai awal *temperatur* terlalu tinggi, maka dibutuhkan lebih banyak pengurangan *temperature* untuk *konvergen*. Sebaliknya jika nilai awal ini terlalu kecil proses pencairan ini mungkin kurang sempurna sehingga ada titik-titik *potensial* yang bisa menjadi *global optimum* terlewat.

Faktor *pereduksi temperatur*  $c$  memainkan peran yang sama. Jika terlalu besar (0.8 atau 0.9) akan memerlukan terlalu banyak langkah komputasi. Sebaliknya, jika terlalu kecil nilai (misal 0.1 atau 0.2) bisa berakibat terlalu cepatnya penurunan *temperatur* sehingga akan banyak titik-titik *potensial* untuk menjadi solusi *global* akan terlewat. Dengan mengacu pada penelitian-penelitian sebelumnya yang sudah membahas tentang penjadwalan dengan metode SA yaitu dengan menggunakan T100, T200, T500 dan T1000. Dari keempat parameter T tadi setiap parameter T dilakukan uji coba faktor *pereduksi suhu* ( $fp$ ) 0.2, 0.5, dan 0.9 didapat *makespan* seperti pada tabe 11.

Penjadwalan dengan metode SA ini dilakukan dengan bantuan *komputasi* matlab yang menggunakan beberapa *parameter* suhu  $T = 100, T = 200, T = 500, T = 1000$  dan pada tiap parameter T digunakan  $fp$  sebesar 0.2, 0.5, 0.9 didapat *makespan* terkecil yaitu sebesar 2636.

Dari perbandingan *parameter* diatas yang bisa dijadikan sebagai solusi usulan adalah pada suhu T 100 dengan urutan proses produksi 5-7-8-2-3-1-6-4 karena dengan suhu yang rendah dibandingkan yang lain pada suhu T 100 ini sudah mendapat solusi yang *optimum* yaitu menghasilkan *makespan* 2636 menit  $\approx 44$ jam yang hanya membutuhkan *iterasi* sebanyak 151 kali dan waktu komputasi yang dihasilkan paling cepat yaitu 0.0468 detik. Untuk *Gantt Chart* bisa dilihat pada gambar dibawah ini.



Gambar 2 Gambar *Gantt Chart* Penjadwalan *Simulated Annealing*

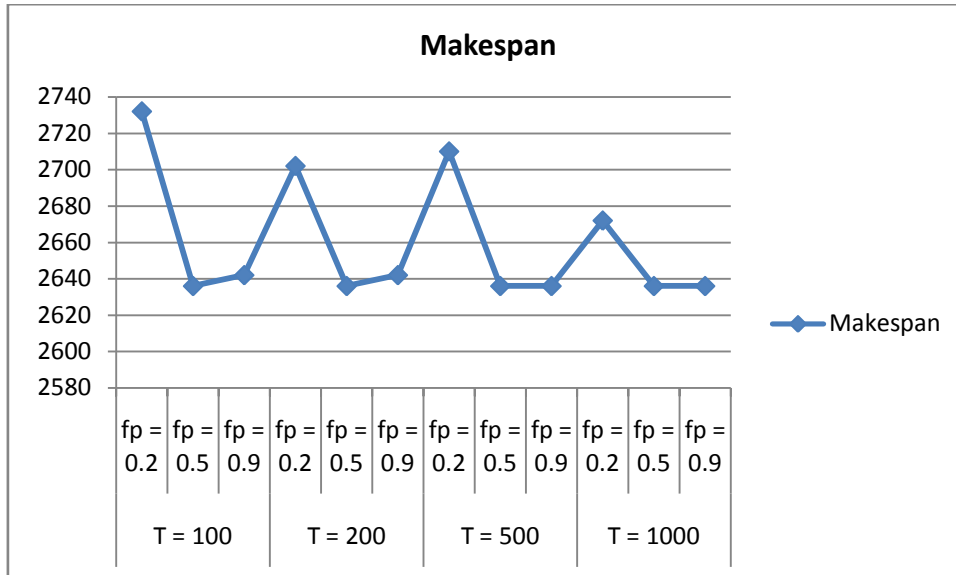
Dari Gambar *Gantt Chart* diatas dapat dihitung total *idle time* dari penjadwalan *Simulated Annealing* pada mesin 1 sebesar = 1558 menit  $\approx 26$  jam, mesin 2 = 1281  $\approx 21$ jam, mesin 3 = 726  $\approx 12$  jam, mesin 4 = 621  $\approx 10$ jam, mesin 5 = 2180  $\approx 36$  jam.

#### A. Analisa Sensitifitas

Analisis *sensitivitas* merupakan analisis yang berkaitan dengan perubahan diskrit parameter untuk melihat berapa besar perubahan dapat ditolerir sebelum solusi optimum mulai kehilangan optimalitasnya. Jika suatu perubahan kecil dalam parameter menyebabkan perubahan drastis dalam solusi, dikatakan bahwa solusi sangat *sensitive* terhadap nilai *parameter* tersebut. Sebaliknya, jika perubahan *parameter* tidak mempunyai pengaruh besar terhadap solusi dikatakan solusi *relative insensitive* terhadap nilai parameter itu.



Dalam penelitian ini *parameter-parameter* yang digunakan adalah *parameter* suhu (T) dan *parameter* faktor *pereduksi* suhu (fp). Berikut grafik perbandingan T 100, T 200, T 500, dan T 1000. Dimana pada setiap suhu (T) digunakan faktor penurunan suhu (fp) 0.2, 0.5, dan 0.9 yang nantinya *parameter-parameter* ini akan dilihat seberapa besar perubahannya dengan *makespan*, jumlah *iterasi*, dan waktu *komputasi*.



Gambar 3  
Grafik Perbandingan T 100, T 200, T 500, dan T 1000 dengan *makespan*

Pada grafik diatas untuk suhu (T) 100 dengan menggunakan fp 0.2 menghasilkan *makespan* yang sangat tinggi yaitu 2732 menit. Hal ini dikarenakan penurunan suhu yang sangat cepat sehingga banyak pencarian solusi *optimum* yang terlewati. Untuk T 100 dengan menggunakan fp 0.5 menghasilkan *makespan* yang rendah yaitu 2636 menit. Hal ini menunjukkan bahwa dengan menaikkan faktor *pereduksi* maka penurunan suhunya tidak terlalu cepat dan pencarian solusi *optimalnya* bisa lebih banyak lagi sehingga menghasilkan *makespan* yang lebih kecil dari yang sebelumnya. Pada T 100 dengan menggunakan fp 0.9 didapat *makespan* sebesar 2642 menit. *Makespan* ini lebih besar dari yang sebelumnya yang hanya menghasilkan 2636 menit. Hal ini dikarenakan dari bilangan *random* yang dibangkitkan pada saat pencarian solusi *optimal* juga berpengaruh pada hasil *makespan* dari setiap *parameter* suhu (T) dan faktor penurunan/*pereduksi* suhu (fp).

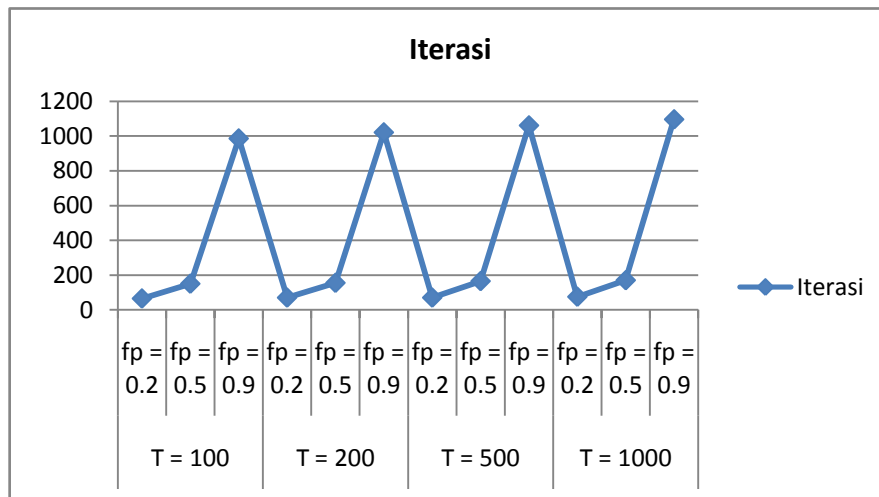
Pada suhu (T) 200 dengan menggunakan fp 0.2 menghasilkan *makespan* yang cukup tinggi yaitu 2702 menit. Hal ini dikarenakan penurunan suhu yang sangat cepat sehingga banyak pencarian solusi *optimum* yang terlewati. Untuk T 200 dengan menggunakan fp 0.5 menghasilkan *makespan* yang rendah yaitu 2636 menit. Hal ini menunjukkan bahwa kenaikan faktor *pereduksi* maka penurunan suhunya tidak terlalu cepat dan pencarian solusi *optimalnya* bisa lebih banyak lagi sehingga menghasilkan *makespan* yang lebih kecil dari yang sebelumnya. Pada T 200 dengan menggunakan fp 0.9 didapat *makespan* sebesar 2642 menit. *Makespan* ini lebih besar dari yang sebelumnya yang hanya menghasilkan 2636 menit. Hal ini dikarenakan dari bilangan *random* yang dibangkitkan pada saat pencarian solusi *optimal* juga berpengaruh pada hasil *makespan* dari setiap *parameter* suhu (T) dan faktor penurunan/*pereduksi* suhu (fp).

Pada suhu (T) 500 dengan menggunakan fp 0.2 menghasilkan *makespan* yang cukup tinggi yaitu 2710 menit. Ini lebih besar dari pada T sebelumnya karena meskipun

T dinaikkan sedangkan fp menggunakan 0.2 penurunan suhu yang cepat akan mempersempit hasil pencarian solusi *optimum*. Untuk T 500 dengan menggunakan fp 0.5 menghasilkan *makespan* 2636 menit. Hal ini menunjukkan bahwa kenaikan faktor *pereduksi* maka penurunan suhunya tidak terlalu cepat dan pencarian solusi *optimalnya* bisa lebih banyak lagi sehingga menghasilkan *makespan* yang lebih kecil dari yang sebelumnya. Juga pada T 500 dengan menggunakan fp 0.9 menghasilkan *makespan* yang sama yaitu 2636 menit.

Sedangkan pada suhu (T) 1000 dengan menggunakan fp 0.2 menghasilkan *makespan* sebesar 2672 menit. *Makespan* ini tidak terlalu besar jika dibandingkan pada fp 0.2 di T sebelumnya karena untuk T 1000 hasil pencarian solusi *optimum* lebih banyak dari pada sebelumnya. T 1000 dengan menggunakan fp 0.5 menghasilkan *makespan* 2636 menit. Hal ini menunjukkan bahwa fp 0.5 menunjukkan kestabilannya karena penurunan suhunya tidak terlalu banyak juga tidak terlalu rendah. Begitu juga pada T 1000 dengan menggunakan fp 0.9 menghasilkan *makespan* yang sama yaitu 2636 menit karena hasil pencarian solusi bisa lebih banyak.

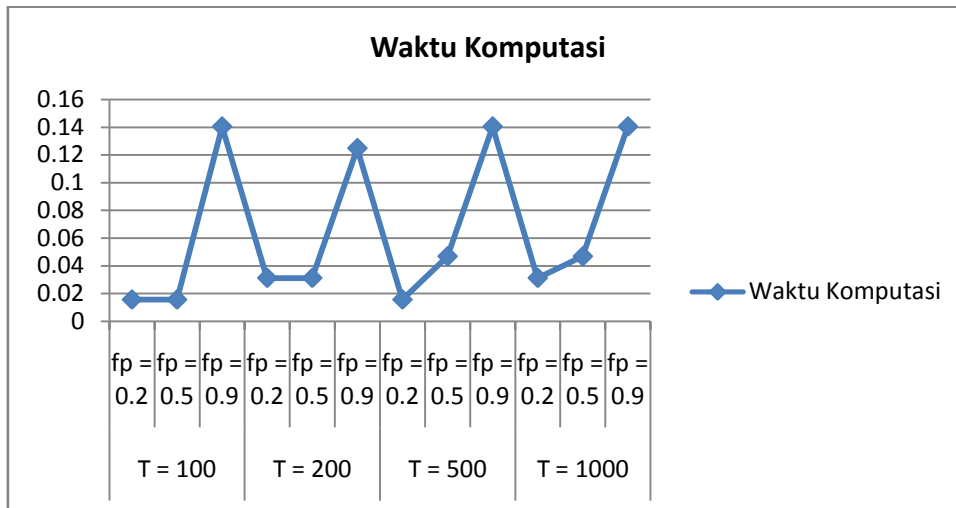
Dari analisa diatas fp sangat berpengaruh pada jumlah *makespan* yang dihasilkan. Jika suhu rendah dan digunakan fp rendah juga maka hasil dari *makespan* juga tinggi karena banyak titik optimal yang terlewati, jika suhu tinggi digunakan fp rendah juga masih belum bisa menemukan titik yang paling *optimal*. Sedangkan jika menggunakan T 0.5 suhu yang diturunkan tidak terlalu besar juga tidak terlalu tinggi, untuk semu T menghasilkan *makespan* yang sama yaitu 2636 menit. Dan untuk penggunaan fp tinggi yang berpengaruh hanya pada yang cukup tinggi pula yaitu pada T 500 dan 1000. Hal ini juga dipengaruhi oleh bilangan random yang dibangkitkan ketika pencarian solusi baru. Sehingga untuk solusi terbaik dilihat dari *makespannya* didapat pada T 100, 200, 500, dan 1000 pada fp 0.5 juga pada T 500, dan T 1000 pada fp 0.9 dengan *makespan* sama yaitu 2636 menit.



Gambar 4  
Grafik Perbandingan T 100, T 200, T 500, dan T 1000 dengan Jumlah Iterasi

Pada grafik diatas untuk suhu (T) 100 dengan menggunakan fp 0.2 menghasilkan *iterasi* sebanyak 66 iterasi. Hal ini dikarenakan penurunan suhu yang sangat cepat sehingga pada *iterasi* tersebut sudah mencapai *steady state*. Untuk T 100 dengan menggunakan fp 0.5 menghasilkan 151. Hal ini menunjukkan bahwa dengan menaikkan faktor *pereduksi* maka *iterasi* yang dihasilkan juga semakin banyak dari pada sebelumnya. Pada T 100 dengan menggunakan fp 0.9 didapat *iterasi* sebanyak 986. *Iterasi* ini lebih besar dari

yang sebelumnya karena jumlah *iterasi* sangat berpengaruh pada setiap *parameter* suhu (T) dan faktor penurunan/*pereduksi* suhu (fp) yang digunakan. Begitu juga pada T 200, T 500 dan T100, setiap terjadi kenaikan pada fp maka jumlah *iterasi* yang dihasilkan juga semakin banyak. Untuk bilangan *random* yang dibangkitkan pada pencarian solusi tidak berpengaruh karena T, fp dan *iterasi* berbanding lurus. Bisa dilihat pada grafik 5.3 setiap T yang dinaikkan dan fp yang dinaikkan maka jumlah *iterasi* yang dihasilkan juga semakin banyak.



Gambar 5

Grafik Perbandingan T 100, T 200, T 500, dan T 1000 dengan Waktu *Komputasi*

Pada grafik diatas untuk suhu (T) 100 dengan menggunakan fp 0.2 menghasilkan waktu *komputasi* 0.0156 detik. Hal ini dikarenakan penurunan suhu yang sangat cepat dan proses pencarian solusi juga akan cepat selesai. Untuk T 100 dengan menggunakan fp 0.5 menghasilkan waktu *komputasi* sama 0.0156 dalam hal ini pembangkitan dari bilangan *random* untuk penggunaan aturan pengacakan juga berpengaruh karena jika pengacakan semakin sulit maka akan dibutuhkan waktu yang lama sesuai dengan aturan pengacakan yang sudah dibahas pada bab sebelumnya. Pada T 100 dengan menggunakan fp 0.9 didapat wktu *komputasi* yang lebih lama dari fp sebelumnya yaitu 0.1404 detik karena pencarian solusi juga semakin banyak, oleh karena itu setiap faktor penurunan/*pereduksi* suhu (fp) yang digunakan juga mempengaruhi waktu *komputasi* yang dihasilkan.

Pada suhu (T) 200 dengan menggunakan fp 0.2 menghasilkan waktu *komputasi* 0.0312 detik. Waktu *komputasi* ini juga sama untuk T 200 dengan menggunakan fp 0.5 yang juga menghasilkan waktu *komputasi* 0.0312 detik, dalam hal ini pembangkitan dari bilangan *random* untuk penggunaan aturan pengacakan juga berpengaruh karena jika pengacakan semakin sulit maka akan dibutuhkan waktu yang lama sesuai dengan aturan pengacakan yang sudah dibahas pada bab sebelumnya. Pada T 200 dengan menggunakan fp 0.9 didapat waktu *komputasi* yang lebih lama dari fp sebelumnya yaitu 0.1248 detik karena pencarian solusi juga semakin banyak, tapi jika dibandingkan dengan fp 0.9 pada T sebelumnya hasilnya lebih kecil karena pembangkitan dari bilangan *random* untuk penggunaan aturan pengacakan juga berpengaruh karena jika pengacakan semakin sulit maka akan dibutuhkan waktu yang lama sesuai dengan aturan pengacakan yang sudah dibahas pada bab sebelumnya.

Pada suhu (T) 500 dengan menggunakan fp 0.2 menghasilkan waktu *komputasi* 0.0156 detik. Untuk waktu *komputasi* pada T 500 dengan menggunakan fp 0.5 menghasilkan waktu *komputasi* 0.0412 detik, lebih besar dari fp sebelumnya. Pada T

500 dengan menggunakan fp 0.9 didapat waktu *komputasi* yang lebih lama dari fp sebelumnya yaitu 0.1404, dengan ini pada T 500 seiring dengan peningkatan fp maka waktu *komputasi* yang dihasilkan juga semakin tinggi.

Sedangkan pada suhu (T) 1000 Pada dengan menggunakan fp 0.2 menghasilkan waktu *komputasi* 0.0312 detik. Untuk waktu *komputasi* pada T 1000 dengan menggunakan fp 0.5 menghasilkan waktu *komputasi* 0.0468 detik, lebih besar dari fp sebelumnya. Pada T 200 dengan menggunakan fp 0.9 didapat waktu *komputasi* yang lebih lama dari fp sebelumnya yaitu 0.1248 dan lebih kecil dari pada fp 0.9 pada T sebelumnya, hal ini dikarenakan pembangkitan dari bilangan *random* untuk penggunaan aturan pengacakan juga berpengaruh karena jika pengacakan semakin sulit maka akan dibutuhkan waktu yang lama sesuai dengan aturan pengacakan yang sudah dibahas pada bab sebelumnya. Dengan ini pada T 1000 seiring dengan peningkatan fp maka waktu *komputasi* yang dihasilkan juga semakin tinggi.

Pada suhu (T) dengan peningkatan fp maka waktu *komputasi* yang dihasilkan cenderung lebih besar meskipun masih ada yang dalam peningkatan fp tersebut waktu *komputasinya* tetap. Begitu juga pada peningkatan suhu (T) waktu *komputasi* yang dihasilkan juga semakin banyak meskipun juga ada salah satu yang lebih kecil tapi keseluruhan cenderung waktu komputasinya naik. Hal ini dikarenakan pembangkitan dari bilangan *random* untuk penggunaan aturan pengacakan juga berpengaruh karena jika pengacakan semakin sulit maka akan dibutuhkan waktu yang lama sesuai dengan aturan pengacakan yang sudah dibahas pada bab sebelumnya.

#### D. Perbandingan Penjadwalan Perusahaan Dengan Penjadwalan *Simulated Annealing*

Untuk mengetahui apakah metode usulan SA lebih baik dari metode yang diterapkan perusahaan saat ini maka akan dilakukan pengukuran terhadap parameter *performance* yang dilihat dari perhitungan nilai *performance* dan efisien. Untuk mengetahui *performance* parameter yang digunakan dengan menentukan metode yang lebih baik digunakan pendekatan dengan mengukur nilai *efisiensi*. *Efisiensi* digunakan untuk mengetahui seberapa besar *efisiensi* dari perbedaan *makespan* yang dihasilkan oleh kedua metode. Untuk mengetahui nilai *efisiensi* menggunakan rumus:

$$Efisiensi = \frac{Z_{Makespan\ Perusahaan} - Z_{Makespan\ SA}}{Z_{Makespan\ Perusahaan}} \times 100\%$$

Dalam perhitungan *makespan* sebelumnya telah didapatkan bahwa total *makespan* metode usulan SA sebesar 2636 menit, sedangkan metode perusahaan didapatkan total *makespan* sebesar 2917 menit. Maka besar parameter *performance efisiensi* adalah sebagai berikut:

$$Efisiensi = \frac{2917 - 2636}{2636} \times 100\% \\ Efisiensi = 10,66\%$$

Dalam perhitungan *makespan* sebelumnya telah didapatkan bahwa total *makespan* metode usulan SA sebesar 2636 menit, sedangkan metode perusahaan didapatkan total *makespan* sebesar 2917 detik. Maka selisih total *makespan* sebesar 281 detik atau berkisar 5 jam. Berdasarkan hasil perhitungan nilai *efisiensi*, didapatkan nilai sebesar 10,66% yang artinya bahwa metode SA menunjukkan lebih *efisien* dan lebih baik dibanding dengan metode yang diterapkan oleh perusahaan saat ini berdasarkan nilai *makespan*nya. Berikut ini tabel hasil perbandingan metode usulan SA dengan metode yang dipakai perusahaan.

Selain makespan juga didapat *idle time* setiap mesin pada penjadwalan awal perusahaan dan penjadwalan dengan *simulated annealing* (SA). Berikut hasil perhitungan efisiensi dari *idle time* tiap mesin.

Tabel 2 Perbandingan *Idle Time* Metode SA dengan kondisi awal

Mesin	<i>Idle time</i> awal	<i>Idle time</i> SA	Selisih	<i>Efisiensi</i>
1	1839	1558	281	2.81 %
2	1562	1281	281	2.81 %
3	1007	726	281	2.81 %
4	902	621	281	2.81 %
5	2461	2180	281	2.81 %

Berdasarkan hasil perhitungan nilai *efisiensi*, didapatkan nilai sebesar 22,07% yang artinya bahwa metode SA menunjukkan lebih *efisien* dan lebih baik dibanding dengan metode yang diterapkan oleh perusahaan saat ini berdasarkan total *idle timenya*. Berikut ini tabel hasil perbandingan metode usulan SA dengan kondisi awal.

Tabel 3 Perbandingan Metode SA Dengan kondisi awal

Perbandingan Metode SA Dengan kondisi awal			
NO.	Parameter	SA	Kondisi awal
1	Urutan <i>job</i> optimal	5-7-2-1-3-8-6-4	1-2-3-4-5-6-7-8
2	<i>Makespan</i>	49 jam	44 jam
3	Selisih waktu	5 jam	
4	<i>Efisiensi makespan</i>	10,66 %	
5	<i>Efisiensi idle time</i>	2.81 % tiap mesin	

## V. KESIMPULAN

Dari hasil pengolahan data dan analisa sensitivitas yang telah dilakukan, maka kesimpulan bahwa Penjadwalan produksi dengan metode *Simulated Annealing* (SA) dari analisa pada pembahasan sebelumnya didapat solusi pada *parameter* suhu (T) 100 dan *parameter* faktor penurunan suhu (fp) 0.5 menghasilkan *makespan* sebesar 2.636 menit  $\approx$  44 jam.

Dari hasil perbandingan pada pembahasan sebelumnya penjadwalan yang dilakukan oleh perusahaan menghasilkan *makespan* sebesar 2917 menit  $\approx$  49jam. Maka hasil *makespan* penjadwalan dengan metode SA lebih baik dengan selisih waktu 5 jam hampir mendekati 1 hari jam kerja kondisi sebelumnya. Selain *makespan* juga didapat *idle time* dari metode SA lebih baik yaitu 2.81 % pada tiap mesin.

## VI. DAFTAR PUSTAKA

- [1] Baker, Kenneth R. 1974. *Principles Of Sequencing and Scheduling*. America : John Wiley and Son Inc.
- [2] Ginting, Rosnani. 2009. *Penjadwalan Mesin*. Yogyakarta : Graha Ilmu
- [3] Uttari, Saria. Skripsi, 2008. "Usulan penjadwalan produksi produk manframe pada mesin punch exentrik". Fakultas Teknik, Universitas Pembangunan Nasional "Veteran". Jakarta
- [4] Nasution, Arman Hakim. 1999. *Perencanaan dan Pengendalian Produksi*. Surabaya : Guna Widya
- [5] Santosa, Budi dan Willy, Paul. 2011. *Metoda Metaheuristik Konsep dan Implementasi*. Surabaya : Guna Widya

- [6] Damanik, Dian Amru. Skripsi, 2011. “Penjadwalan Produksi dengan metode Simulated Annealing Pada Unit Produksi Daun Pintu di PT. Mahogany Lestari”. Fakultas Teknik, Universitas Sumatera Utara. Medan
- [7] Ratnasari, Aditya. 2007. “Implementasi Algoritma Simulated Annealing Dalam Penyelesaian Masalah Penjadwalan Produksi Jenis Flowshop”. Fakultas Teknik Universitas Kristen Duta Wacana
- [8] Rofiq, Ainur 2013. Skripsi, 2013. “Pengembangan Algoritma Simulated Annealing Pada Permasalahan Hybrid Flowshop Scheduling Untuk Meminimasi Makespan dan Total Tardines”. Fakultas Teknik Universitas Sepuluh November Surabaya.