

USULAN PENJADWALAN PRODUKSI DENGAN MENGGUNAKAN ALGORITMA ANT COLONY SYSTEM (STUDI KASUS CV. BINA RUBBER SUMEDANG)

PRODUCTION SCHEDULING PROPOSAL USING ANT COLONY SYSTEM ALGORITHM (CASE STUDY OF CV. BINA RUBBER SUMEDANG)

Daniel H. Irawan¹, Victor Suhandi², Vivi Arisandhy³
falconer_120@yahoo.com

Abstrak

CV. Bina Rubber bergerak dalam pembuatan sparepart mesin berbahan dasar karet. Perusahaan merasakan bahwa sistem produksi saat ini tidak terlalu baik, dan masih ada peluang untuk ditingkatkan. Hal ini ditandai dengan besarnya delay mesin dan rendahnya utilisasi sehingga makespan menjadi relatif tinggi.

Pemecahan masalah dilakukan dengan usulan penjadwalan produksi menggunakan metode Ant Colony System Job Shop (ACSJS) atau Active Scheduling. Metode ACSJS akan dihitung dengan software, sedangkan metode Active Scheduling dihitung secara manual.

Dalam memilih antara ACSJS dengan Active Scheduling, dilakukan perbandingan dengan tiga buah contoh kasus jobshop yaitu kasus 3 job 3 mesin, 4 job 3 mesin, 4 job 4 mesin. Pada kasus pertama, ACSJS dan Active Scheduling menghasilkan makespan yang sama. Pada kasus kedua, ACSJS unggul 3 menit, dan pada kasus ketiga, ACSJS unggul 9 menit. Berdasarkan hasil tersebut ACSJS terpilih sebagai metode pemecahan masalah perusahaan.

Penjadwalan dengan software ACSJS menghasilkan output dengan waktu running 40 menit menggunakan program Turbo Pascal for Windows, dengan nilai parameter $\rho=0.01$, $\beta=0.5$, $\alpha=0.5$, $a=1000$. Setelah dilakukan optimasi, didapatkan bahwa parameter optimal $\rho=0.1$, $\beta=0.7$, $\alpha=0.3$, $a=34$, dan menghasilkan output dengan waktu running 1 menit.

Penjadwalan ACSJS untuk pesanan aktual perusahaan (8 pesanan), diperoleh pengurangan makespan 15,2%, pengurangan delay sebesar 8,6%, dan peningkatan utilisasi mesin sebesar 17.9%.

Kesimpulannya, metode ACSJS lebih baik dari metode yang digunakan perusahaan saat ini (metode FIFO) dan perusahaan sebaiknya mengganti metode penjadwalan FIFO dengan ACSJS.

Kata Kunci : Penjadwalan Produksi, Ant Colony Optimization

¹Daniel H. Irawan, mahasiswa Teknik Industri Universitas Kristen Maranatha Bandung

²Victor Suhandi, Dosen Teknik Industri Universitas Kristen Maranatha Bandung

³Vivi Arisandhy, Teknik Industri Universitas Kristen Maranatha Bandung

Abstract

CV Bina Rubber is peripatetic company in the field of making rubber base machine sparepart. Company feels that the production system is not good enough, and still has opportunity to improved. It's marked by high level of machine delay, and low machine utilisation. That makes total time needed to produce goods become relatively high.

Problem solving is by production scheduling proposal to apply one of Ant Colony System Job Shop (ACSJS) method, or Active Scheduling method. ACSJS method will use computerize calculation (software), while Active Scheduling manually calculated.

Three jobshop case comparasion will be use to choose between ACSJS or Active Scheduling. Which is 3-job-3-machine case, 4-job-3-machine case, and 4-job-4-machine case. At first case, ACSJS and Active Scheduling yields the same makespan. For second case, ACSJS yields 3minutes-shorter makespan, and at third case, ACSJS yields 9minutes-shorter makespan. Based on the result, ACSJS then chosed as proposed company trouble-shooting method.

Operation scheduling using ACSJS software generate optimal processing sequences in 40 minutes (with Turbo Pascal for Windows). This output is generates using folowing value of parameters : $\rho=0.01$, $\beta=0.5$, $\alpha=0.5$, $a=1000$,. Afier doing parameter tuning, then known that the optimal parameter value for company case is as folowing : $\rho=0.1$, $\beta=0.7$, $\alpha=0.3$, $a=34$. With this optimal parameter value, the program running time can be reduced to 1 minute.

With ACSJS scheduling method for company actual orders (8 orders), obtained reduction of makespan equal to 15,2%. delay decreases equal to 8,6%. And machine utilisation increasesequals to 17.9%.

The conclusion is that ACSJS method better than existing company method (FIFO) and company should change their scheduling method with ACSJS.

Keyword : Operation Scheduling, Ant Colony Optimization

1. Pendahuluan

CV. Bina Rubber adalah sebuah perusahaan yang bergerak dalam pembuatan *sparepart* mesin berbahan dasar karet. Perusahaan ini terletak di Jl. By Pass Sumedang, dan memiliki konsumen dari berbagai daerah di Indonesia, khususnya di pulau Jawa seperti Yogyakarta, Semarang, Depok, Bekasi, Jakarta, dan Surabaya. Selain itu, perusahaan sedang berencana untuk memperluas pasarnya ke pulau Sumatra dan Kalimantan dengan membuka anak perusahaan. Dengan semakin banyaknya konsumen, maka semakin banyak pula pesanan yang masuk sehingga perusahaan perlu menjaga agar pesanan yang diberikan konsumen dapat dipenuhi dengan baik. Masalah yang timbul di perusahaan adalah proses produksi yang tidak terlalu baik, dan masih terdapat kemungkinan untuk dilakukan perbaikan. Hal ini ditandai oleh besarnya *delay* mesin, dan rendahnya utilisasi yang secara langsung berpengaruh pada waktu proses total yang relatif besar.

Untuk itu akan dilakukan usulan penjadwalan untuk memperbaiki pejadwlan yang sudah ada saat ini dengan tujuan untuk meminimasi delay, yang secara langsung mengurangi makespan dan meningkatkan utilisasi mesin.

2. Kajian Pustaka

2.1 Penjadwalan

Menurut Kenneth R. Baker, Morton dan Pentico, Richard W Conway, dan Vollman, [Astrid,2007], pengertian penjadwalan adalah sebagai berikut. Kenneth R. Baker : Proses pengalokasian sumber-sumber untuk memilih sekumpulan tugas dalam jangka waktu tertentu. Morton dan Pentico : Proses pengorganisasian, pemilihan dan penentuan waktu penggunaan sumber untuk memenuhi seluruh aktifitas yang dibutuhkan untuk menghasilkan output yang diharapkan pada waktu yang telah ditentukan. Richard W Conway : Proses pengurutan pembuatan produk secara menyeluruh pada sejumlah mesin tertentu dan pengurutan didefinisikan sebagai pembuatan produk pada satu mesin dalam jangka waktu tertentu. Vollman : Suatu proses persiapan jadwal yang berkaitan dengan penentuan saat suatu item harus dimulai dan saat akan selesai, atau penentuan susunan aktifitas atau pekerjaan yang akan diselesaikan selama kurun waktu tertentu oleh suatu departemen atau *work center*.

2.2 Tujuan Penjadwalan

Tujuan penjadwalan perlu diketahui terlebih dahulu agar pemilihan teknik penjadwalan dapat dilakukan dengan sebaik-baiknya Terdapat berbagai macam tujuan penjadwalan yang pada garis besarnya dapat dikelompokkan ke dalam tiga bagian [Kusuma, 2004], yaitu:

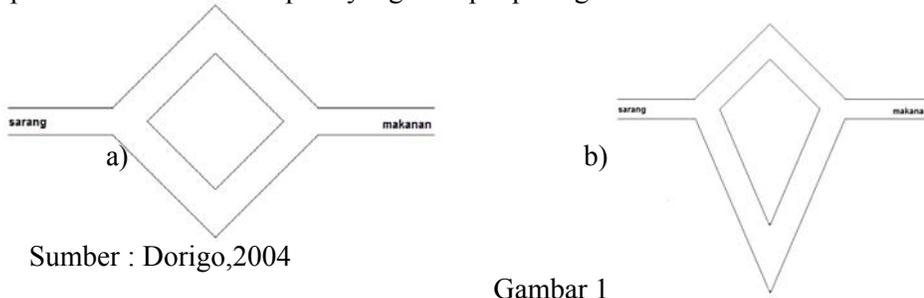
- Peningkatan utilisasi peralatan/sumber daya dengan cara menekan waktu menganggur sumber daya tersebut. Untuk sejumlah pekerjaan telah diketahui bahwa maksimasi utilisasi sumber daya berbanding terbalik dengan waktu yang diperlukan untuk menyelesaikan seluruh pekerjaan (*makespan*). Dengan demikian sasaran penjadwalan yang terutama adalah menekan waktu penyelesaian produk secara keseluruhan.
- Sasaran lain yang mungkin dicapai ialah minimasi jumlah persediaan barang dalam proses. Tujuan ini dicapai dengan cara meminimasi jumlah pekerjaan yang menunggu dalam antrian untuk diproses. Indikator jumlah antrian pekerjaan ini dinyatakan dengan besaran waktu alir rata-rata.
- Tujuan penjadwalan lainnya ialah menekan kelambatan. Dalam banyak hal sejumlah pekerjaan memiliki batas waktu penyelesaian pekerjaan (*due date*), dan apabila pekerjaan selesai setelah *due date* maka perusahaan dikenai penalti. Terdapat beberapa tujuan penjadwalan berkenaan dengan kelambatan ini. Tujuan penjadwalan dapat berupa minimasi kelambatan, atau minimasi jumlah pekerjaan yang terlambat, atau minimasi kelambatan rata-rata.

2.2 Ant Colony System

Ant colony sistem dikembangkan oleh Marco Dorigo pada tahun 1991. Algoritma ini adalah algoritma yang meniru cara kerja atau sifat alami koloni semut dalam bekerjasama untuk menemukan jalur terpendek dari sarang mereka menuju makanan. Semut adalah organisme yang bekerja secara kelompok. Mereka berkomunikasi satu sama lain dengan menggunakan *pheromone* atau parfum serangga, karena mereka buta. Saat seekor semut berjalan, ia akan meninggalkan air

liur (*pheromone*) dalam jumlah tertentu. *Pheromone* yang ditinggalkan ini selanjutnya menjadi tanda bagi semut lain yang berjalan di belakangnya. Semut cenderung untuk berjalan mengikuti jalur dengan intensitas *pheromone* yang lebih besar. Jalur dengan intensitas *pheromone* yang besar menandakan bahwa jalur itu sering dilalui oleh semut, jalur yang sering dilalui ini akan dianggap sebagai jalur yang paling baik untuk mencari makanan (jalur dengan jarak terpendek) [Dorigo,2004].

Untuk membuktikan hal tersebut, Goss *et al* pada tahun 1989 membuat percobaan sederhana seperti yang terdapat pada gambar 1.

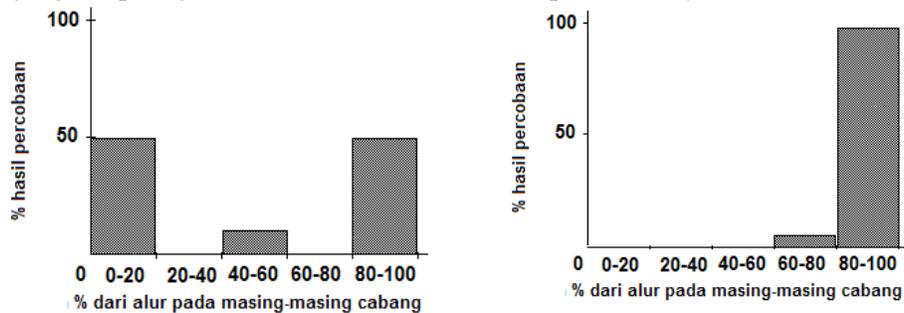


Sumber : Dorigo,2004

Gambar 1

Percobaan Goss *et al*

Pada percobaan a, dikondisikan 2 jalur dengan jarak yang sama. Sedangkan pada percobaan b, dikondisikan 2 jalur dengan panjang berbeda dimana jalur 2 lebih panjang daripada jalur 1. Berikut ini adalah hasil percobaannya.



Sumber : Marco Dorigo 2004

Gambar 2

Hasil percobaan Goss *et al*

Berikut ini adalah keterangan untuk masing-masing percobaan

→ Percobaan a

Pada percobaan ini dibuat 2 jalur dengan panjang yang sama dari sarang menuju makanan. Saat semut pertama dimasukkan ke dalam percobaan, dan menemui percabangan jalur, semut akan memilih jalur sepenuhnya secara random. Hal ini dikarenakan belum adanya *pheromone* pada jalur yang semut lalui sehingga probabilitas terpilihnya kedua jalur adalah sama. Setelah semut pertama memilih jalur, maka ia akan meninggalkan air liur di jalur tersebut. Semut kedua, ketiga, keempat, dan seterusnya akan cenderung memilih jalur yang sama seperti jalur yang dilalui oleh semut pertama, karena semut-semut tersebut mencium air liur yang

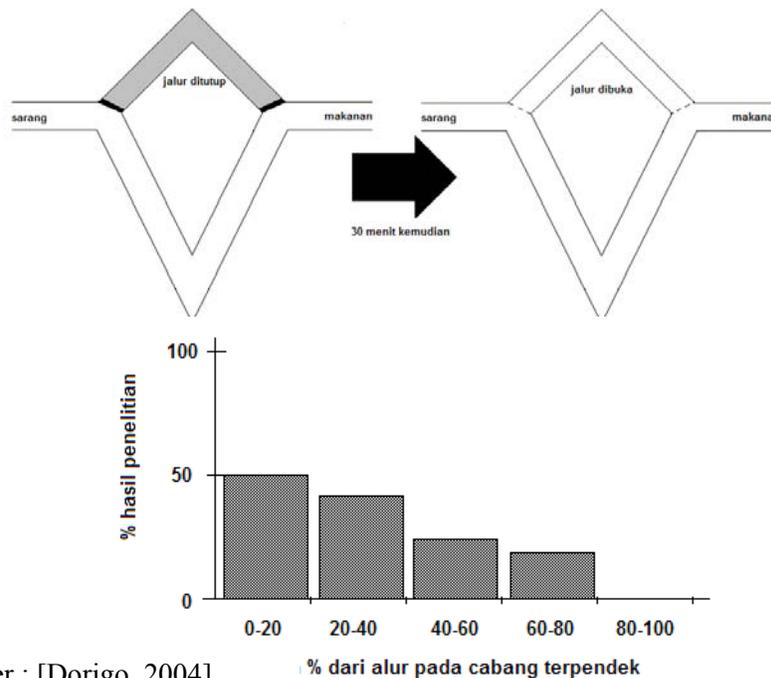
ditinggalkan semut pertama. namun terdapat kemungkinan bahwa semut akan memilih cabang lain selain cabang yang sudah dilalui semut sebelumnya. Setelah semut memilih jalur baru, maka *pheromone* di jalur lama akan menguap, dan terus menguap sampai jalur tersebut dipilih lagi. Hal ini menyebabkan probabilitas terpilihnya kedua jalur menjadi sama besar.

➔ Percobaan b

Pada percobaan b, terdapat 2 jalur yang tidak sama panjang dimana jalur 2 lebih panjang dibandingkan jalur 1. Percobaan b ini akan menunjukkan bagaimana koloni semut bekerjasama untuk mencari jalur terpendek menuju makanan. Pada awalnya, semut pertama memilih jalur 1 (jalur pendek), semut kedua, ketiga, keempat, dan seterusnya juga memilih jalur pendek karena mencium *pheromone* yang ditinggalkan oleh semut pertama.

Dengan semakin banyaknya *pheromone* yang ditinggalkan semut di jalur 1, maka semut berikutnya menjadi ragu untuk memilih jalur 1 sebagai jalur yang baik. Hal ini menyebabkan ada kemungkinan semut untuk memilih jalur 2 (jalur panjang). Namun kemungkinan terpilihnya jalur panjang ini lebih kecil karena perbedaan waktu tempuh dari kedua jalur akan menyebabkan intensitas penguapan *pheromone* yang berbeda. Intensitas *pheromone* di jalur panjang akan lebih cepat menguap karena untuk menempuh jalur panjang membutuhkan waktu lebih lama dibandingkan dengan waktu untuk menempuh jalur pendek.

Yang menjadi pokok pembahasan dalam percobaan diatas adalah kapan semut akan memilih jalur terpanjang. Karena itu dibuat percobaan lain seperti yang tampak pada gambar 3 berikut ini.



Sumber : [Dorigo, 2004]

Gambar 3
Percobaan lanjutan

Hasil percobaan lanjutan diatas memperlihatkan bahwa semut cenderung untuk memilih cabang pendek setelah 30 menit. Untuk kasus ini, koloni semut lebih memilih cabang panjang karena pada cabang terpanjang terdapat penambahan *pheromone*, dan sedikit penguapan *pheromone*. Penguapan ini perlahan-lahan akan membuat semut memilih jalur terpanjang sebagai jalur baru bagi mereka karena semut tidak dapat mencium atau lupa pada jalur terbaik mereka.

Percobaan ini menunjukkan karakteristik semut yang selalu berusaha menemukan jalur terpendek untuk mencari makanan. Karakteristik inilah yang diadopsi oleh *Ant Colony System* (ACS) untuk melakukan optimasi di berbagai bidang. Perbedaannya adalah :sistem alami semut memiliki *pheromone* awal sebesar 0 (tidak ada *pheromone* saat semut pertama jalan), sedangkan pada ACS *pheromone* awalnya ditentukan terlebih dahulu. Sehingga “semut” dalam ACS akan memilih jalur berdasarkan probabilitas (yang tergantung dari jarak dan intensitas *pheromone*) dan bilangan random.

ACS adalah algoritma heuristik yang serba guna untuk memecahkan masalah optimasi. Algoritma ini memiliki karakteristik sebagai berikut [Dorigo,1996]:

- Serbaguna. Dapat memecahkan banyak masalah optimasi
- Sempurna. Dapat diterapkan untuk memecahkan banyak masalah dengan hanya mengubah sedikit dari algoritma asalnya.
- Berbasis populasi.

2.3 Permasalahan Dalam *Ant Colony Sistem*

Berikut ini adalah contoh masalah yang dapat dipecahkan dengan menggunakan ACS beserta referensi utama dari permasalahan tersebut.

Tabel 1
Tabel Permasalahan Dalam *Ant colony*

Problem name	Problem Type	Main References	Problem name	Problem Type	Main References
Routing	Traveling Salesman	Dorigo, Maniezzo & Colomi (1991a, b, 1996) Dorigo (1992) Gambardella & Dorigo (1995) Dorigo & Gambardella (1997a, b) Stutzle & Hoos (1997, 2000)	Subset	Multiple knapsack	Leguizamón & Michalewicz (1999)
		Bullnheimer, Hartl & Strauss (1999c) Cordon, de Viana, Herrera & Moreno (2000)		Max independent set	Leguizamón & Michalewicz (2000)
	Vehicle routing	Bullnheimer, Hartl & Strauss (1999a, b) Gambardella, Taillard & Agazzi (1999) Reimann, Stummer & Doerner (2002) Gambardella & Dorigo (1997, 2000)		Redundancy allocation	Liang & Smith (1999)
	Sequential Ordering	Gambardella & Dorigo (1997, 2000)		Set Covering	Leguizamón & Michalewicz (2000)
Assignment	Quadratic assignment	Maniezzo, Colomi & Dorigo (1994) Stutzle (1997b) Maniezzo & Colomi (1999) Maniezzo (1999) Stutzle & Hoos (2000)	Other	Weight Constrained graph tree partition	Hadji, Rahoual, Talbi & Bachelet (2000)
	Graph Coloring	Costa & Hertz (1997)		Arc-weighted-cardinality tree	Cordone & Maffioli (2001)
	Generalized assignment	Lourenco & Serra (1990, 2002)		Maximum clique	Blum & Blesa (2003)
	Frequency assignment	Maniezzo & Carbonaro (2000)		Shortest common supersequence	Michel & Middendorf (1998, 1999)
Scheduling	Job shop	Colomi, Dorigo, Maniezzo & Trubian (1994)	Constraint satisfaction	Solnon (2000, 2002)	
	Open shop	Pfahringer (1996)	2D-HP protein folding	Shmygelska, Aguirre-Hernandez & Hoos (2002)	
	Flow shop	Stutzle (1998a)	Bin packing	Lewne & Ducatelle (2003)	
	Total tardiness	Bauer, Bullnheimer, Hartl & Strauss (2000)	Machine Learning	Classification rules	Parpenelli, Lopes & Freitas (2002b)
Total weighted tardiness	den Besten, Stutzle & Dorigo (2000) Merkle & Middendorf (2000, 2003a) Gagne, Price & Gravel (2002) Merkle, Middendorf & Schemeck (2000a, 2002) Blum (2002a, 2003a)	Bayesian networks		de Campos, Gamez & Puerta (2002b)	
Project scheduling		Fuzzy systems		Casillas, Cordon & Herrera (2000)	
Group shop		Connection-oriented network routing		Stoondenwoerd, Holland, Bruten & Rothkrantz (1996) Stoondenwoerd, Holland & Bruten (1997) White, Pagurek & Opacher (1990) Di Caro & Dorigo (1998) Bonabeau, Henay, Guerin, Snyers & Theraulaz (1998)	
Subset	Multiple knapsack	Leguizamón & Michalewicz (1999)	Connectionless network routing	Di Caro & Dorigo (1997, 1998c,f) Subramanian, Druschel & Chen (1997) Heusse, Snyers, Guerin & Kuntz (1998) van der Put (1998)	
	Max independent set	Leguizamón & Michalewicz (2000)	Optical network routing	Navarro Varela & Sinclair (1999)	
	Redundancy allocation	Liang & Smith (1999)			
	Set Covering	Leguizamón & Michalewicz (2000) Hadji, Rahoual, Talbi & Bachelet (2000)			

Sumber : [Dorigo, 2004]

2.4 Ant colony Job shop

Dalam melakukan pengolahan secara manual untuk metode ACSJS, ada beberapa data dan notasi yang digunakan. Berikut adalah notasi-notasi yang digunakan dalam pengolahan data :

Pheromone : air liur semut

Ω : job yang belum dijadwalkan

α : kepekaan terhadap jejak ($0 < \alpha < 1$)

β : kepekaan terhadap desirability ($0 < \beta < 1$)

ρ : evaporasi / penguapan jejak *pheromone* ($0 < \rho < 1$)

NC: jumlah siklus

NC max: jumlah siklus maksimal

n: jumlah job

m: jumlah mesin

a: jumlah semut

σ : *tabu list* / proses yang sudah dijadwalkan

d_{ij} : jarak heuristik dari kota i ke j (waktu proses di j)

τ_{ij} : jumlah *pheromone* yang menghubungkan kota i ke kota j

$\tau_{ij}(t)$: intensitas *pheromone* yang menghubungkan kota i ke kota j pada saat t

$\Delta\tau_{ij}(t)$: penambahan jejak *pheromone* pada saat t

$\Delta\tau_{ij}(t + 1)$: intensitas *pheromone* pada saat $t + 1$

$P_{ij}(t)$: probabilitas semut ke- k pergi dari kota i ke kota j pada saat t

berikut ini adalah penjelasan untuk tiap langkah pengerjaan :

Langkah 1 :

Mengidentifikasi input yang diperlukan dalam pengolahan data. Input yang diperlukan adalah $\alpha, \beta, \rho, NC_{max}, P_j, a, n$, matriks proses, dan matriks *routing*.

- Input α dan β ($0 < \alpha, \beta < 1$) tidak dapat dipastikan karena nilai $\alpha + \beta = 1$, maka nilai α dan β harus dicari melalui proses *trial* dan *error*.
- Input ρ yang paling optimal adalah 0,01. [Dorigo,2004]
- Input NC_{max} , diambil nilai yang sebesar-besarnya. Semakin banyak siklus operasi (iterasi), maka semakin banyak jalur yang dihasilkan, dan akan menghasilkan nilai *makespan* yang beragam. Keragaman nilai *makespan* ini akan memperbesar kemungkinan untuk mendapatkan nilai *makespan* yang optimal (yang paling kecil).
- Input d_{ij} , diperoleh dari pengumpulan data di perusahaan.
- Input a (jumlah semut), digunakan nilai yang sebesar-besarnya. Semakin banyak jumlah semut, maka semakin beragam nilai *makespan* yang dihasilkan.
- Input n (jumlah job) didapatkan dari pengumpulan data di perusahaan.
- Input matriks proses dibuat berdasarkan input d_{ij} .
- Input matriks *routing*, dibuat berdasarkan penggunaan mesin di masing-masing operasi.

Setelah semua input didapatkan, maka selanjutnya diset $t=0$ dan $NC=0$. kemudian semua operasi diterjemahkan ke dalam grafik untuk perhitungan probabilitas.

Lalu ditetapkan *pheromone* awal adalah mencapai nilai maksimal (1). *Pheromone* awal ditetapkan=1 karena pada awal proses, belum ada penguapan jejak *pheromone*.

Kemudian dibuat matriks *pheromone* dari satu operasi ke operasi lainnya. Disini *tabulist* masih kosong karena belum ada pekerjaan yang dijadwalkan

Langkah 2:

Pertama, set $k=1$, lalu set k sampai sebanyak a (jumlah semut).Lalu tempatkan sebanyak a semut pada *node* 0.Untuk $k=1$ (*tabulist* $\{0,.. \}$) maka buat himpunan untuk operasi yang akan dijadwalkan (Ω) dengan melihat grafik.

Langkah 3:

Hitung probabilitas setiap operasi yang terpilih (Ω) dengan menggunakan persamaan berikut :

$$P_{ij} = \left\{ \frac{[\tau_{ij}(t)]^{\alpha} \cdot \left[\frac{1}{d_{ij}} \right]^{\beta}}{\sum_{k \in \Omega} k \varepsilon \Omega [\tau_{ij}(t)]^{\alpha} \cdot \left[\frac{1}{d_{ij}} \right]^{\beta}} \right\}$$

Melalui bilangan random *uniform*, maka pilih operasi dengan probabilitas yang paling mendekati bilangan random, lalu hitung kembali probabilitas dengan langkah yang sama sampai semua operasi dijadwalkan. Setelah semua operasi $k=1$ dijadwalkan, hitung probabilitas untuk semut ke-2, semutke-3, dan seterusnya dengan langkah yang sama seperti perhitungan probabilitas untuk semut pertama diatas.

Langkah 4:

Setelah semua semut dijadwalkan, maka dihitung *makespan* untuk masing-masing semut dan dipilih penjadwalan yang memiliki *makespan* terkecil.

Langkah 5:

- Hitung penambahan *pheromone* $\Delta\tau_{ij}$ berdasarkan nilai *makespan* terkecil menggunakan persamaan berikut :

$$\Delta\tau_{ij}(t) = \frac{1}{\text{makespan}_{\text{terkecil}}}$$

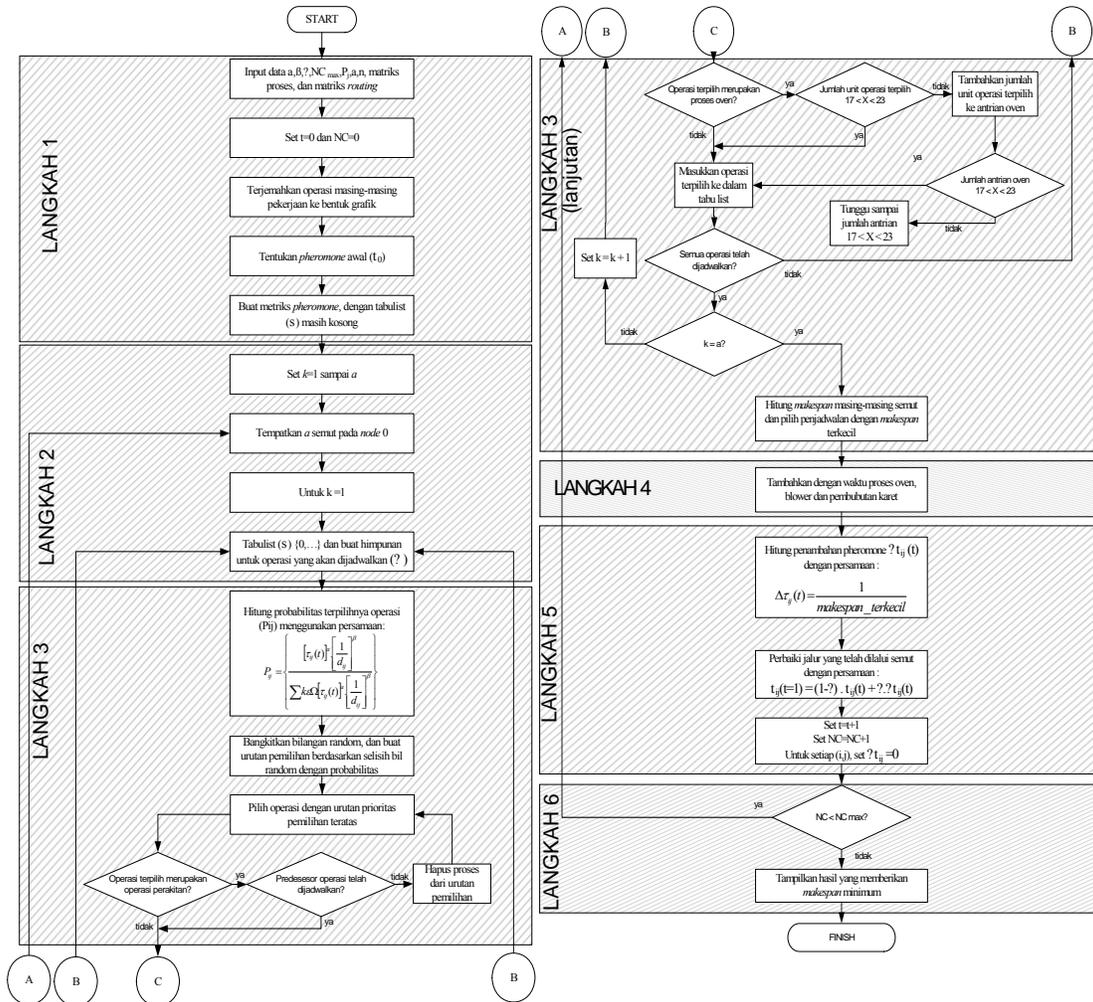
- Lalu perbaiki jalur yang dilalui semut menggunakan persamaan berikut : $\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}(t)$
- Jalur terpendek memiliki persamaan $\Delta\tau_{ij}$ dari hasil persamaan diatas, sedangkan alur lainnya memiliki nilai $\Delta\tau_{ij}=0$.
- Lalu set $t = t + 1$, dan $NC = NC + 1$.

Langkah 6:

Kosongkan semua *tabulist*. Jika $NC < NC_{\text{max}}$, maka kembali ke langkah 2. Jika tidak, maka tampilkan hasil *makespan* terkecil.

Untuk penelitian kasus CV. Bina Rubber, perlu dilakukan modifikasi metode Ant Colony System (ACS). Modifikasi ini bertujuan untuk menyesuaikan metode ACS agar sesuai dengan kondisi, dan kebutuhan perusahaan.

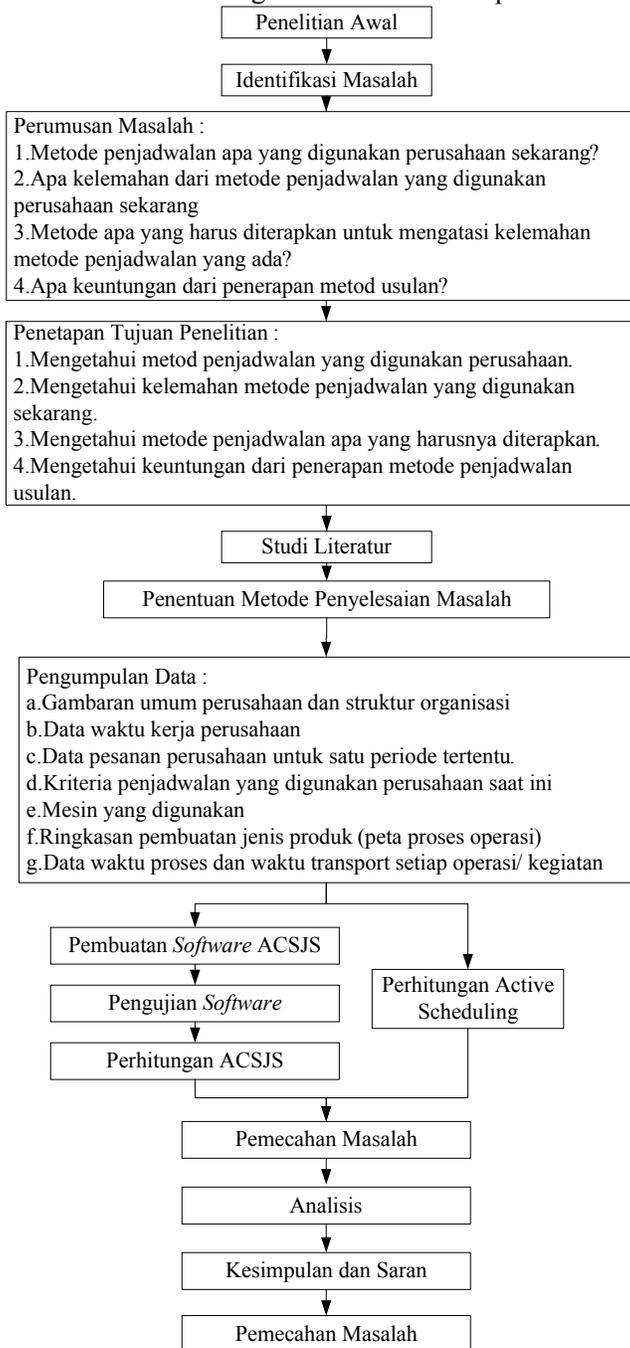
Setelah dilakukan penyesuaian dengan kondisi perusahaan, maka langkah pengerjaan ACS menjadi seperti tampak pada gambar 4



Gambar 4
Bagan ACS modifikasi

3. Metodologi Penelitian

Berikut ini adalah metodologi dalam melakukan penelitian ini.



Sumber : [Hasil pengolahan data penulis, 2008]

Gambar 5
Metodologi Penelitian

4. Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil data umum perusahaan, dan data yang digunakan untuk melakukan penjadwalan (umumnya data dari bagian produksi)

Data umum perusahaan meliputi :

- Sejarah singkat perusahaan
- Struktur Organisasi
- Job description

Data penjadwalan meliputi :

- Jam kerja
- Data Mesin
- Data pesanan
- Peta Proses Operasi
- Data Waktu Proses

5. Pengolahan Data dan Analisis

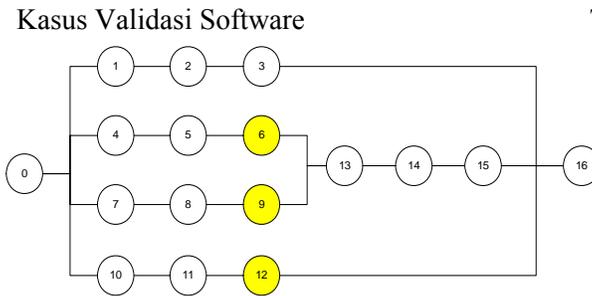
5.1 Penjadwalan FIFO

Dengan menggunakan penjadwalan yang ada saat ini (FIFO) maka pesanan tgl 11-16 Agustus 2008 (8 pesanan) dapat diselesaikan dalam 11 hari kerja (dari tanggal 16-08-08 sampai 27-08-08). Prioritas pengerjaan penjadwalan ini berdasarkan diterimanya pesanan. Waktu proses dan urutan proses tidak dipertimbangkan. Hal ini merupakan kelemahan dari FIFO.

Tabel 2
Hasil Penjadwalan FIFO

Job	Tanggal Siap Produksi	Waktu (menit)		Selesai	
		Operasi	Transport	Menit	Tanggal
RS	16-08-08	2108.3	200	1857	22-08-08
RP	16-08-08	3048	240	2037	22-08-08
RB	16-08-08	3999.3	280	2188	23-08-08
RI	16-08-08	3565.3	280	2548	25-08-08
RM	16-08-08	2550	216	2706	25-08-08
RE	16-08-08	1838.3	180	2838	26-08-08
JR	16-08-08	6066	360	3439	27-08-08
RH	16-08-08	3815	252	3691	27-08-08

5.2 Validasi Software



Gambar 6

Graph Validasi Software

Operasi 6, 9, dan 12 (kuning) adalah operasi dummy

Hasil Validasi Software

Tabel 3
Tabel Matriks Proses dan waktu
Matriks waktu dan Proses

Matriks Waktu					
Job	Waktu Proses (menit)				
1	75	200	275		
2	90	186	120	108	264
	90	30			
3	140	315			

Matriks Proses					
Job	Mesin				
1	2	3	7		
2	2	3	6	3	7
	4	5			
3	1	7			

Tabel 4
Hasil Validasi Software

Metode	Urutan	Makespan
manual	10-11-4-1-7-8-5-6-2-3-9-13-12-14-15	2450
software	10-11-4-1-7-8-5-6-2-3-9-13-12-14-15	2450

Dari hasil diatas dapat diketahui bahwa hasil dari metode ACSJS manual sudah sama dengan hasil dari ACSJS dengan menggunakan bantuan *software*. Maka dapat disimpulkan bahwa *software* yang dibuat sudah valid dan dapat digunakan sebagai pengganti perhitungan manual ACSJS. Untuk perhitungan penjadwalan berikutnya, digunakan *software* ACSJS untuk memudahkan penelitian.

5.3 ACSJS Vs Active Scheduling

Dari hasil disamping dapat disimpulkan bahwa metode ACSJS memberikan nilai *makespan* yang lebih singkat di contoh kasus 2 dan 3. Karena itu metode ACSJS akan digunakan dalam memecahkan masalah penjadwalan perusahaan.

Tabel 5
Perbandingan ACSJS Vs Active Scheduling

Kasus	Metode	Makespan
1	ACSJS	81Menit
	<i>Active Scheduling</i>	81Menit
2	ACSJS	13 Menit
	<i>Active Scheduling</i>	15 Menit
3	ACSJS	18 Menit
	<i>Active Scheduling</i>	27 Menit

5.4 Penjadwalan ACSJS

Dari segi *makespan*, metode FIFO menghasilkan *makespan* sebesar 3691 menit, sedangkan metode ACSJS menghasilkan *makespan* sebesar 3130 menit. Metode ACSJS dapat menyelesaikan semua pekerjaan 561 menit (15,2%) lebih cepat dibandingkan metode FIFO.

Dari segi *delay* dan utilisasi, metode ACSJS menghasilkan *delay* sebesar 1256 menit (8,6% lebih baik dari FIFO), dan utilisasi sebesar 71,86%. (meningkat 17,92% dibandingkan dengan FIFO)

Hasil perbandingan diatas menunjukkan bahwa metode ACSJS menghasilkan solusi penjadwalan yang lebih baik dibandingkan metode FIFO, kerana menghasilkan *makespan* 15,2% lebih singkat, *delay* 8,6% lebih kecil, dan utilisasi 17,92% lebih besar.

Tabel 6
Hasil Penjadwalan ACSJS

Job	Tanggal Siap Produksi	Waktu (menit)		Selesai	
		Operasi	Transport	Menit	Tanggal
RS	16-08-08	2108.3	200	2307	23-08-08
RP	16-08-08	3048	240	2936	26-08-08
RB	16-08-08	3999.3	280	3130	26-08-08
RI	16-08-08	3565.3	280	2470	25-08-08
RM	16-08-08	2550	216	1774	22-08-08
RE	16-08-08	1838.3	180	1418	21-08-08
JR	16-08-08	6066	360	2710	25-08-08
RH	16-08-08	3815	252	1605	21-08-08

5.5 Parameter Optimal

Tabel 8
Optimasi Parameter Alpha dan Beta

Alpha	Beta	Siklus sampai optimal
0	1	409
0.1	0.9	103
0.2	0.8	2
0.3	0.7	1
0.4	0.6	72
0.5	0.5	160
0.6	0.4	368
0.7	0.3	399
0.8	0.2	>1000
0.9	0.1	>1000
1	0	>1000

Tabel 9
Optimasi Parameter a

Parameter a	Siklus sampai optimal
25	218
26	210
27	202
28	>1000
29	>1000
30	182
31	176
32	170
33	165
34	160
35	272

Tabel 10
Optimasi Parameter ρ

ρ	Mencapai optimal di (siklus)
0.1	271
0.01	778
0.001	>1000

Dari hasil penentuan parameter optimal dapat diketahui bahwa parameter optimal untuk penjadwalan perusahaan adalah : $\alpha = 0.3$, $\beta = 0.7$, $\rho = 0.1$, dan $a = 34$

6. Kesimpulan dan Saran

6.1 Kesimpulan

Dari hasil pengumpulan dan pengolahan data serta analisis, penulis dapat menyimpulkan sebagai berikut :

1. Metode penjadwalan perusahaan saat ini (FIFO)

- Metode penjadwalan perusahaan (FIFO) memiliki urutan pengerjaan berdasarkan urutan diterimanya *job* yaitu 1-2-3-4-5-6-7.
- Dengan metode ini, maka pesanan dapat diselesaikan seluruhnya dalam waktu 3691 menit (setara dengan 10 hari kerja).
- Utilisasi mesin dengan metode ini sebesar 60.94%.
- Delay mesin dengan metode ini sebesar 1374 menit.
- Kelemahan dari metode ini adalah hanya menggunakan patokan waktu ACC pesanan.

2. Metode usulan (ACSJS)

- Metode penjadwalan usulan diperoleh dengan membandingkan antara ACSJS dengan *Active Scheduling*. Dari hasil perbandingan disimpulkan bahwa metode ACSJS lebih baik dari *Active Scheduling* karena metode ACSJS memberikan nilai *makespan* yang lebih kecil dibandingkan metode *Active Scheduling*
- Metode penjadwalan usulan lebih baik dibandingkan metode penjadwalan perusahaan (FIFO). Dengan metode penjadwalan usulan yaitu metode ACSJS (*Ant Colony System – Job Shop*), semua pekerjaan dapat diselesaikan dalam waktu 3130 menit (15,2% lebih cepat dari metode FIFO).
- Utilisasi mesin dengan metode ini sebesar 71.86%.
- Delay mesin dengan metode ini sebesar 1256 menit.
- Setting parameter optimal untuk meminimasi jumlah iterasi yang dibutuhkan dalam mencapai solusi optimal ACSJS adalah sebagai berikut :
 1. Nilai ρ optimal untuk penjadwalan perusahaan adalah 0.1
 2. Nilai a optimal untuk penjadwalan perusahaan adalah 34
 3. Nilai α optimal untuk penjadwalan perusahaan adalah 0.3
 4. Nilai β optimal untuk penjadwalan perusahaan adalah 0.7

3. Manfaat metode penjadwalan usulan (ACSJS)

- *Makespan* lebih singkat 15,2%.
- Delay berkurang 8.6%
- Utilisasi meningkat 17.92%
- Kepuasan konsumen diharapkan meningkat dengan berkurangnya *makespan*.

6.2 Saran

Setelah menyelesaikan semua proses dalam penelitian ini, maka penulis dapat memberikan saran kepada pihak perusahaan untuk memperbaiki penjadwalan produksi yang ada saat ini. Usulan tersebut adalah sebagai berikut :

- Perusahaan sebaiknya mengganti metode penjadwalan yang ada saat ini yaitu metode FIFO dengan metode ACSJS.
- Untuk memudahkan perhitungan penjadwalan dengan ACSJS, sebaiknya perhitungan ACSJS diterjemahkan kedalam bahasa pemrograman yang lebih baru dan memiliki *user interface design* yang lebih baik, dan yang terpenting dapat menghasilkan bilangan random murni.

DAFTAR PUSTAKA

1. A. Colorni; M. Dorigo; V. Maniezzo; M. Trubian; "**Ant System For Job Shop Scheduling**"; JORBELL, London, 2004.
2. Dorigo, Marco.; Stützle, Thomas; "**Ant Colony Optimization**"; Massachusetts Institute of Thecnology, London, 2004.
3. *Job Shop Scheduling Benchmark*, OR-library, <http://mscmga.ms.ic.ac.uk/jeb/orlib/jobshopinfo.html>.
4. Kusuma, Hendra, Ir.; "**Perencanaan dan Pengendalian Produksi I**"; Universitas Kristen Maranatha, Bandung; 1992.
5. Nasution, Arman Hakim Ir.; "**Perencanaan dan Pengendalian Produksi**", Alfabeta, Bandung, 1991.
6. Sugiyono, DR.; "**Metode Penelitian Administrasi**", Penerbit ALFABETA, Bandung, 2003.
7. Sutalaksana., Iftikar Z.; Anggawisastra, Ruhana; Tjakraatmadja, Jhon H.; "**Teknik Tata Cara Kerja**", Jurusan Teknik Industri Institut Teknologi Bandung, 1979.
8. Van Der Zwaan, Sjoerd.; Marques, Carlos.; "**Ant Colony Optimisation for Job Shop Scheduling**"; Instituto de Sistemas e Robótica, Lisboa, 2006.
e-mail : sjoerd@isr.ist.utl.pt; cmarques@isr.ist.utl.pt