

# Ant colony optimization (ACO) pada job shop scheduling problem (JSSP)

Wahidatul Fatin Amanullah <sup>1</sup>, Sapti Wahyuningsih <sup>2</sup>, Lucky Tri Oktoviana <sup>3</sup>  
Universitas Negeri Malang

Email : syahputra.wahida@gmail.com; sapti.wahyuningsih.fmipa@um.ac.id;  
lucky.tri.fmipa@um.ac.id

**ABSTRAK** : *Job Shop Scheduling Problem* (JSSP) merupakan permasalahan dalam menentukan *makespan* yang minimum pada suatu jadwal dengan  $n$  *jobs* dan  $m$  mesin. Salah satu algoritma yang dapat digunakan dalam penyelesaian permasalahan ini adalah *ant colony optimization* (ACO). ACO adalah metode yang terinspirasi oleh perilaku makhluk hidup yaitu perilaku dari sekumpulan semut yang keluar dari sarangnya menuju sumber makanan dengan meninggalkan zat *pheromone*. Dalam algoritma ACO terdapat beberapa tahapan penyelesaian yaitu inisialisasi parameter, aturan transisi status, tahap pembaharuan jejak *pheromone*, dan menemukan solusi terbaik. Parameter yang dibutuhkan yaitu  $m$  (banyaknya mesin),  $\alpha$  (tetapan pengendali intensitas jejak semut),  $\beta$  (tetapan pengendali visibilitas),  $\tau_{ij}(t)$  (intensitas *pheromone*),  $k$  (banyaknya semut),  $\rho$  (evaporasi *pheromone*),  $Q$  (konstanta), dan  $C_{max}$  (banyaknya iterasi) yang digunakan untuk mencari rute dan *makespan*.

**Kata kunci** : *Ant Colony Optimization, Job Shop Scheduling Problem, Makespan.*

**ABSTRACT** : *Job Shop Scheduling Problem* (JSSP) is a problem in determining the minimum *makespan* on a schedule with  $n$  jobs and  $m$  machines. One algorithm that can be used in the settlement of this problem is ant colony optimization (ACO). ACO is a method inspired by the behavior of living things is the behavior of a bunch of ants coming out of the nest to food sources by leaving *pheromone* substance. In the ACO algorithm there are several stages of completion, namely initialization parameters, status transition rules, the *pheromone* trail stage of renewal, and find the best solution. Required parameter  $m$  (number of machines),  $\alpha$  (constant for controlling the intensity of a trail of ants),  $\beta$  (constant controlling visibility),  $\tau_{ij}(t)$  (the intensity of the *pheromone*),  $k$  (the number of ants),  $\rho$  (evaporation *pheromone*),  $Q$  (constant) and  $C_{max}$  (number of iterations) are used to route and *makespan*.

**Key Words :** *Ant Colony Optimization, Job Shop Scheduling Problem, Makespan.*

Masalah penjadwalan muncul ketika terdapat berbagai macam tugas (*job*) atau proses yang harus dilakukan, sedangkan sumber daya (waktu, bahan baku, tenaga kerja, mesin, modal, dan sebagainya) yang dibutuhkan untuk menyelesaikan tugas atau proses tersebut terbatas sehingga diperlukan suatu pengaturan atas pelaksanaan tugas atau proses tersebut. *Job shop scheduling* adalah sistem yang memproses  $n$  banyaknya *jobs* pada  $m$  banyaknya mesin untuk menemukan jadwal yang meminimalkan waktu selesainya *job* (Surekha dan Sumanthi, 2010:95). Karakteristik JSSP menurut Karakteristik penjadwalan *Job-Shop* menurut Pinedo dan Chao (2002) dapat dideskripsikan sebagai berikut:

- 1) Ada sejumlah  $m$  mesin dan  $n$  *job*
- 2) Setiap *job* terdiri dari satu rantai urutan operasi mengikuti rute yang telah ditentukan
- 3) Setiap operasi dalam *job* diproses oleh salah satu mesin yang ada dengan waktu proses yang diasumsikan tetap
- 4) Setiap operasi dapat melalui satu jenis mesin lebih dari satu kali (*recirculaton*)
- 5) Tidak boleh ada *preemption* (penundaan satu *job* oleh *job* yang lain)
- 6) Fungsi tujuan permasalahan penjadwalan model *job shop* adalah untuk mencari satu jadwal yang meminimalkan *makespan*
- 7) Permasalahan penjadwalan *job shop* merupakan permasalahan optimisasi kombinatorial yang kompleks (*np-hard*)

Terdapat metode yang lahir karena terinspirasi oleh tingkah laku makhluk hidup yaitu tingkah laku dari sekumpulan semut yang keluar dari sarangnya menuju sumber makanan dengan meninggalkan zat *pheromone* yang melahirkan algoritma yang disebut *Ant Colony Optimization* (ACO). ACO adalah metaheuristik yang menyatukan konsep dari bidang-bidang seperti *Artificial Intelligence* dan biologi yang terinspirasi pada perilaku kolektif semut. Serangga sosial yang membentuk koloni semut ini, merupakan sistem yang mengatur dirinya sendiri dan desentralisasi yang dianggap sebagai *Swarm Intelligence*.

Berkat kecerdasan yang muncul dari hubungan sederhana antara semut-semut ini, koloni dapat memecahkan masalah yang kompleks di lingkungan mereka, seperti masalah menemukan jalur terpendek antara koloni dan makanan, yang dapat digunakan untuk menemukan solusi terbaik untuk masalah optimasi kombinatorial (Flórez *dkk*, 2013). Setiap semut memulai rutenya melalui sebuah titik yang dipilih secara acak (setiap semut memiliki titik awal yang berbeda). Secara berulang kali, satu persatu titik yang ada dikunjungi oleh semut dengan tujuan untuk menghasilkan sebuah rute.

Zwaan dan Marques (2000:2) pemilihan titik-titik yang akan dilaluinya didasarkan pada fungsi probabilitas, dinamai aturan transisi status (*state transition rule*), dengan mempertimbangkan *visibility* (invers dari jarak) titik tersebut dan jumlah *pheromone* yang terdapat pada ruas yang menghubungkan titik tersebut. Semut mampu mengindra lingkungannya yang kompleks untuk mencari makanan dan kemudian kembali ke sarangnya dengan meninggalkan zat *pheromone* pada rute-rute yang mereka lalui. *Pheromone* adalah zat kimia yang berasal dari kelenjar endokrin dan digunakan oleh makhluk hidup untuk mengenali sesama jenis, individu lain, kelompok, dan untuk membantu proses reproduksi. Berbeda dengan hormon, *pheromone* menyebar ke luar tubuh dan hanya dapat mempengaruhi dan dikenali oleh individu lain yang sejenis (satu *spesies*). Proses peninggalan *pheromone* ini dikenal sebagai *stigmergy*, yaitu sebuah proses memodifikasi lingkungan yang tidak hanya bertujuan untuk mengingat jalan pulang ke sarang, tetapi juga memungkinkan para semut berkomunikasi dengan koloninya. Semut lebih suka bergerak menuju ke titik-titik yang dihubungkan dengan rute dengan jarak yang pendek dan memiliki tingkat *pheromone* yang tinggi.

Algoritma yang dapat digunakan pada *job shop* yaitu algoritma penjadwalan *job shop* alternatif *routing* menggunakan *variabel neighbourhood descent with fixed threshold* (Hartini *dkk*, 2014), algoritma *genetik* (Li dan Chen, 2010), algoritma *Bacterial Foraging Optimization* (BFO) (Narendhar dan Amudha, 2012), algoritma *evolutionary for JSS* (Mesghouni *dkk*, 2004). Sedangkan algoritma ACO dapat digunakan pada permasalahan *Travelling Salesman Problem* (TSP) (Hingrajika *dkk*, 2012), algoritma *ant colony*

*optimization : a solution of load balancing in Cloud* (Mishra dan Jaiswal, 2012), dan algoritma ACO *for Microwave Corrugated Filters Design* (Gaviria dkk, 2013). Berdasarkan jurnal tentang algoritma *ant colony optimization* tersebut, oleh karena itu perlu dibahas mengenai algoritma *ant colony optimization* (ACO) pada *job shop scheduling problem* (JSSP).

Tujuan penulisan artikel ini adalah untuk mendeskripsikan langkah-langkah algoritma *ant colony optimization* pada *job shop scheduling problem* dan menerapkan algoritma *ant colony optimization* pada *job shop scheduling problem* untuk 2 mesin dan 3 mesin. Batasan masalah pada artikel ini diasumsikan setiap mesin dapat melakukan semua *job* dan selesai dari perhitungan untuk penerapan algoritma digunakan program Ms. Excel 2007 dengan rumus yang dikembangkan sendiri.

### **Karakteristik Ant Colony Optimization (ACO)**

Florez dkk (2013) pada bagian ini akan menjelaskan mengenai karakteristik *Ant Colony Optimization*. Terdapat empat langkah dari ACO, yaitu :

Langkah 1 : Inisialisasi Parameter dan Nilai *Pheromone* Awal

Inisialisasi parameter terdiri dari : Intensitas *pheromone*  $\tau_{ij}(t)$ ; Tetapan pengendali intensitas jejak semut ( $\alpha$ ); Tetapan pengendali visibilitas ( $\beta$ ); Tingkat evaporasi *pheromone*  $\rho$ ; Banyaknya semut ( $k$ ).

Langkah 2 : Aturan Transisi Status

Aturan transisi status dinamai probability yang ditunjukkan oleh persamaan (2.1).

Menurut Ventressa dan Ombuki  $P_{ij}$  merupakan probabilitas dari  $k$  semut pada titik  $i$  yang memilih untuk menuju ke titik  $j$

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in \text{tabu}_k(t)} [\tau_{ik}(t)]^\alpha [\eta_{ik}]^\beta} & \text{jika } (i, j) \in \text{tabu}_k \\ 0 & \text{untuk yang lainnya} \end{cases} \quad (2.1)$$

Dimana  $\text{Tabu}_k$  adalah tabu list untuk semut  $k$ .  $\text{Tabu}_k(i)$  adalah elemen ke  $i$  dari  $\text{Tabu}_k$ , yaitu titik ke  $i$  yang dikunjungi semut  $k$  pada saat perjalanan. Sedangkan Lequizamón dan Schutz (2002)

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{h \in S_k(t)} [\tau_{ih}(t)]^\alpha [\eta_{ih}]^\beta} & \text{jika } j \in S_k(t) \\ 0 & \text{sebaliknya} \end{cases} \quad (2.1)$$

di mana  $S_k(t)$  adalah himpunan operasi schedulable yang masih belum dijadwalkan oleh  $k$  semut pada  $t$  waktu.  $\tau_{ij}$  adalah jumlah *pheromone* yang terdapat pada sisi antara titik  $i$  dan titik  $j$ .  $\eta_{ij} = \frac{1}{d_{ij}}$  adalah desirability (invers dari jarak  $d_{ij}$ ) yaitu bobot fungsi yang dapat dipilih. Nilai yang diberikan pada bobot fungsi ini biasanya disebut informasi heuristik. Jika semakin tinggi nilai  $\eta_{ij}(t)$  maka akan diperoleh probabilitas yang tinggi. Parameter  $\alpha$  dan  $\beta$  menentukan hubungan antara informasi *pheromone* dan informasi heuristik.  $\alpha$  adalah sebuah parameter yang mengontrol bobot relatif *pheromone* ( $0 \leq \alpha \leq 1$ ).  $\beta$  adalah parameter pengendali jarak ( $0 \leq \beta \leq 1$ ).

Langkah 3 : Update *pheromone* trail

Setelah semua semut menyelesaikan perjalanannya masing-masing kemudian *pheromone* diupdate. Maniezzo dkk (2009) aturan pembaruan *pheromone* global diimplementasikan pada persamaan ( 2.2 ) sebagai berikut.

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2.2)$$

variabel  $\tau_{ij}(t)$  menunjukkan intensitas *pheromone* pada koneksi (i,j) pada waktu  $t$  dengan  $0 \leq \rho \leq 1$  adalah parameter tingkat evaporasi *pheromone* atau koefisien penguapan *pheromone*.  $\Delta\tau_{ij}(t) = \sum_{k=1}^a \Delta\tau_{ij}^k(t)$ , dimana  $a$  adalah jumlah sebenarnya semut di koloni dan  $\Delta\tau_{ij}^k(t)$  adalah jumlah per unit panjang jejak substansi (*pheromone* untuk semut nyata) diletakkan pada koneksi (i,j) oleh semut ke- $k$  pada saat  $t$  waktu dan diberikan oleh rumus berikut:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & \text{jika } (i,j) \in \text{perjalanan yang dilakukan oleh semut } k \\ 0 & \text{untuk yang lainnya} \end{cases} \quad (2.3)$$

Dimana  $Q$  adalah parameter konstan,  $L_k$  *makespan* yang ditemukan oleh semut ke- $k$  yaitu panjang perjalanan yang dilalui oleh semut. Cara alternatif untuk mendapatkan  $\Delta\tau_{ij}(t)$  yaitu hanya mempertimbangkan semut terbaik dalam siklus saat ini bukan semua semut dalam koloni. Dengan pendekatan elitis ini, hanya koneksi yang terlibat dalam solusi terbaik yang akan terpengaruh mengenai

penguatan tingkat *pheromone*. Dalam hal ini,  $\Delta\tau_{ij}(t) = \Delta\tau_{ij}^k(t)$  di mana  $k \in \{1, \dots, a\}$  adalah indeks dari semut yang menemukan solusi terbaik dalam siklus saat algoritma (Lequizamón dan Schutz, 2002).

Untuk memastikan bahwa semut mengunjungi  $n$  titik yang berbeda, diberikan sebuah memori yang dinamai tabu list pada masing-masing semut, yaitu sebuah struktur data yang menyimpan titik-titik yang telah dikunjungi semut dan melarang semut mengunjungi kembali titik-titik tersebut sebelum mereka menyelesaikan sebuah perjalanan dengan kata lain tabu list menghubungkan setiap semut dalam menghindari urutan sehingga semut menjadwalkan operasi yang sudah dijadwalkan. List tabu <sub>$k$</sub> ( $t$ ) mempertahankan seperangkat operasi terjadwal hingga  $t$  waktu untuk semut ke- $k$ . Ketika solusi selesai, tabu list digunakan untuk menghitung solusi yang ditemukan semut pada perjalanan tersebut. Tabu list kemudian dikosongkan dan semut kembali bebas memilih titik tujuannya pada perjalanan berikutnya atau memilih permutasi alternatif dari operasi untuk siklus berikutnya (Ventresca dan Ombuki, 2004).

Langkah 4 : Menemukan Solusi terbaik

Menemukan solusi terbaik oleh semut dilakukan dengan mencari lintasan yang tingkat evaporasinya kecil, tingkat *pheromone* yang dilewati tinggi dan jalur lintasan dari semut ke makanan memiliki jarak yang optimum. Dengan hal ini, semut akan memperoleh makanan dengan melalui lintasan yang lebih dekat dengan sarangnya.

## PEMBAHASAN

Tujuan penyelesaian permasalahan JSSP pada ACO adalah untuk meminimumkan hasil makespan. Langkah-langkah untuk menyelesaikan permasalahan adalah sebagai berikut.

Langkah 1 : Inisialisasi Parameter dan Nilai *Pheromone* Awal

Input untuk ACO pada JSSP yaitu:

1. Waktu proses tiap operasi
2. Intensitas *pheromone*  $\tau_{ij}(t)$  (*pheromone* awal) dengan  $\tau_{ij}(t) > 0$
3. Tetapan pengendali intensitas jejak semut ( $\alpha$ ) dengan  $0 \leq \alpha \leq 1$

4. Tetapan pengendali visibilitas ( $\beta$ ) dengan  $0 \leq \beta \leq 1$
5. Banyaknya semut ( $k$ )
6.  $\rho$ , dengan  $0 \leq \rho \leq 1$
7.  $Q$  suatu konstanta
8.  $C_{max}$  (banyaknya iterasi)

#### Langkah 2 : Aturan Transisi Status

Proses yang dilakukan dalam algoritma ACO pada JSSP melalui aturan transisi status dengan rumus probabilitas  $P_{ij}^k(t)$  yang ditunjukkan semut untuk mencari kemungkinan rute atau mencari solusi. Mencari rute yang akan dilalui selanjutnya, akan berhubungan dengan nilai  $\tau_{ij}(t), \eta_{ij}(t), \alpha, \beta, S_k, Tabu_k$  untuk menghitung probabilitas terpilihnya jalan yang akan dilalui semut. Dengan menggunakan probabilitas  $P_{ij}^k(t)$  pada persamaan (2.1) akan ditemukan nilai-nilai probabilitas. Probabilitas dihitung untuk setiap *job* pada masing-masing mesin, selanjutnya pilih nilai probabilitas yang tertinggi pada setiap mesin. Perlu diperhatikan bahwa nilai heuristik yang berbeda dari yang dipertimbangkan oleh Dorigo, Maniezzo, Colomi, Zwaan dan Marques dihitung sebagai  $\eta_{ij}(t) = 1/(C_{time_j} - I_{time_j})$ .  $C_{time_j}$  dan  $I_{time_j}$  masing-masing mewakili selesainya operasi  $j$  dan periode waktu yang tertentu mesin tetap siaga karena penugasan operasi  $j$ .  $I_{time_j}$  dimisalkan sama dengan nol. Alasan dibalik nilai heuristik ini adalah bahwa operasi terjadwal yang menyelesaikan lebih awal kemungkinan dan menghindari kemalasan dari mesin tertentu dengan lebih baik. Dengan demikian, semakin kecil  $C_{time_j}$  dan  $I_{time_j}$ , semakin tinggi nilai heuristik yang terkait dengan operasi  $j$ .

#### Langkah 3 : Update *Pheromone Trail*

Meng-*update pheromone trail* terdiri dari *update pheromone* lokal dan *update pheromone* global. *Update pheromone* lokal dihitung berdasarkan perolehan nilai dari perhitungan pada langkah 2. Menghitung waktu yang optimal diperoleh dari proses pencarian rute di berbagai mesin yang dipilih dari *pheromone* lokal yang tertinggi. Dengan rumus  $\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$  akan diperoleh *update pheromone* lokal dan *update pheromone* global

#### Langkah 4 : Menemukan Solusi Terbaik

Menemukan solusi yang terbaik atau output dalam algoritma ACO pada JSSP yaitu rute yang dilewati semut, *makespan* yang minimum, dan nilai *pheromone* lokal dan global dari rute yang telah diperoleh dari langkah-langkah sebelumnya, sehingga akan dihasilkan jadwal pada permasalahan JSSP yang lebih optimum.

### Penerapan Algoritma *Ant Colony Optimization* (ACO) pada *Job Shop Scheduling Problem* (JSSP)

Dalam proses penerapan algoritma digunakan dua contoh permasalahan dengan contoh 1 diperoleh dengan memilih waktu proses untuk setiap *job* pada setiap mesin. Contoh 2 diperoleh dari jurnal oleh Florez *dkk*, (2013). Data dari contoh 1 dan contoh 2 tersebut dipilih karena memiliki data yang diperlukan yaitu data mesin, *job*, dan waktu proses. Kedua contoh akan diselesaikan dengan bantuan aplikasi Ms.Excel 2007. Untuk ilustrasi pengerjaan dapat diamati pada Gambar 1 berikut.

Langkah 1 : $t_{ij}=0,1$ , Alfa =1, betha = 1, k=2, ro=0,1 , Q=1, Cmax=2														
Langkah 2 : Probability= $(t_{ij})^1 * (visibility)^1$														
Langkah 3 : Update pheromone														
semut 1,														
	M1	M2	visib	M1	M2	atas	M1	M2	Prob	M1	M2	lok.M1	lok.M2	global
J1	13	3	J1	0,08	0,33	J1	0,008	0,033	J1	0,038	0,283			
J2	2	5	J2	0,5	0,2	J2	0,05	0,02	J2	0,247	0,17			
J3	1	3	J3	1	0,33	J3	0,1	0,033	J3	0,493	0,283	0,583		0,561
J4	4	6	J4	0,25	0,17	J4	0,025	0,017	J4	0,123	0,142			
J5	5	7	J5	0,2	0,14	J5	0,02	0,014	J5	0,099	0,121			
						max	0,203	0,118	max	0,493	0,283			
										0-j3	0-j1			
J1	13	3	J1	0,08	0,33	J1	0,008	0	J1	0,075	0	0,577		0,555
J2	2	5	J2	0,5	0,2	J2	0,05	0,02	J2	0,487	0,237			
J3	1	3	J3	1	0,33	J3	0	0,033	J3	0	0,395	0,577		0,485
J4	4	6	J4	0,25	0,17	J4	0,025	0,017	J4	0,243	0,198			
J5	5	7	J5	0,2	0,14	J5	0,02	0,014	J5	0,195	0,169			
						max	0,103	0,084	max	0,487	0,395			
										j3-j2	j1-j3			
J1	13	3	J1	0,08	0,33	J1	0,008	0	J1	0,146	0			
J2	2	5	J2	0,5	0,2	J2	0	0,02	J2	0	0,393			0,483
J3	1	3	J3	1	0,33	J3	0	0	J3	0	0	0,564		0,544
J4	4	6	J4	0,25	0,17	J4	0,025	0,017	J4	0,474	0,327			
J5	5	7	J5	0,2	0,14	J5	0,02	0,014	J5	0,38	0,28			
						max	0,053	0,051	max	0,474	0,393			
										j2-j4	j3-j2			
J1	13	3	J1	0,08	0,33	J1	0,008	0	J1	0,278	0			
J2	2	5	J2	0,5	0,2	J2	0	0	J2	0	0			
J3	1	3	J3	1	0,33	J3	0	0	J3	0	0			
J4	4	6	J4	0,25	0,17	J4	0	0,017	J4	0	0,538			0,628
J5	5	7	J5	0,2	0,14	J5	0,02	0,014	J5	0,722	0,462	0,812		0,767
						max	0,028	0,031	max	0,722	0,538			
										j4-j5	j2-j4			
										1	1	1,09	1,09	1,017

Gambar 1 Ilustrasi

Gambar tersebut merupakan tabel data dan pengerjaan dengan Ms.Excel untuk contoh 1. Terlihat bahwa langkah 1 telah terpenuhi terdiri dari inisialisasi parameter dan pheromone awal. Pada langkah 2 dilakukan perhitungan probabilitas, untuk semut 1 seperti pada Gambar 1 dicari nilai probabilitas yang tertinggi (bertanda kuning pada gambar) untuk masing-masing mesin hingga masing-masing mesin memiliki rutenya. Rute yang diperoleh bisa digunakan untuk menghitung makespan (waktu total) dari semua job yang dapat diselesaikan. Langkah 3 (berwarna hijau) dapat diperoleh dengan meng-update jejak pheromone lokal yang diperoleh dengan rumus update jejak berdasarkan probabilitas yang tertinggi dan pheromone global diperoleh dari pemilihan pheromone lokal yang tertinggi setelah dilakukan iterasi seperti pada Tabel 1 berikut.

Tabel 1 Iterasi contoh 1

Iterasi	Semut	Rute yang diperoleh	Makespan	Pheromone	
				Lokal	Global
1	1	0-J <sub>3</sub> - J <sub>2</sub> - J <sub>4</sub> - J <sub>5</sub> - J <sub>1</sub>	28	3,627	3,443
	1	0-J <sub>1</sub> - J <sub>3</sub> - J <sub>2</sub> - J <sub>4</sub> - J <sub>5</sub>	30	3,06	
	2	0-J <sub>2</sub> - J <sub>3</sub> - J <sub>4</sub> - J <sub>5</sub> - J <sub>1</sub>	28	3,548	
	2	0-J <sub>3</sub> - J <sub>1</sub> - J <sub>2</sub> - J <sub>4</sub> - J <sub>5</sub>	30	3,06	
2	<b>1</b>	<b>0-J<sub>3</sub>- J<sub>2</sub>- J<sub>4</sub>- J<sub>5</sub>- J<sub>1</sub></b>	<b>28</b>	<b>4,965</b>	<b>4,647</b>
	1	0-J <sub>3</sub> - J <sub>2</sub> - J <sub>4</sub> - J <sub>5</sub> - J <sub>1</sub>	39	4,268	
	2	0-J <sub>2</sub> - J <sub>3</sub> - J <sub>4</sub> - J <sub>5</sub> - J <sub>1</sub>	28	4,243	
	2	0-J <sub>1</sub> - J <sub>3</sub> - J <sub>2</sub> - J <sub>5</sub> - J <sub>4</sub>	28	3,203	

Melalui Tabel 1 diperoleh makespan yang memiliki pheromone tertinggi saat iterasi 2 yaitu saat rute 0-J<sub>3</sub>- J<sub>2</sub>- J<sub>4</sub>- J<sub>5</sub>- J<sub>1</sub> makespan diperoleh 28.

### Hasil Analisis

Analisis yang dilakukan adalah dengan mengubah-ubah nilai parameter yaitu  $\alpha = 1$ ,  $\beta = 1$ ,  $\tau_{ij}(t) = 0.1$ ,  $k = 2$ ,  $\rho = 0.1$ ,  $Q = 1$  dan  $C_{max} = 2$ . Parameter yang diubah pertama adalah  $\alpha$  menjadi  $\alpha = 0$ ,  $\alpha = 0.5$ , hasil yang diperoleh yaitu rute 0-J<sub>3</sub>-J<sub>2</sub>-J<sub>4</sub>-J<sub>5</sub>-J<sub>1</sub> dan makespan 28. Demikian juga saat parameter  $\beta$ ,  $\tau_{ij}(t)$ ,  $k$ ,  $\rho$ ,  $Q$  dan  $C_{max}$  diubah menjadi  $\beta = 0, 0.5$ ,  $\tau_{ij}(t) = 0.5, 1$ ,  $k = 4$ ,  $\rho = 0, 1$ ,  $Q = 4$  dan  $C_{max} = 4$  hasil yang diperoleh adalah rute dan makespan yang sama. Jadi perubahan parameter tersebut tidak mempengaruhi hasil rute dan makespan.

### KESIMPULAN DAN SARAN

Dapat disimpulkan bahwa pada artikel ini, setelah diterapkan algoritma *ant colony optimization* pada *job shop scheduling problem* untuk 2 mesin dan 3 mesin yang dibagi menjadi 8 kasus, kasus 1 yaitu  $\alpha = 1; \beta = 1; \tau_{ij}(t) = 0,1, k = 2; \rho = 0,1; Q = 1; \text{ dan } C_{\max} = 2$ . Hasil analisis diperoleh bahwa pada kedua contoh menghasilkan rute dan makespan yang sama. Jadi perubahan nilai parameter tersebut tidak mempengaruhi hasil rute dan makespan. Permasalahan pada artikel ini telah dilakukan perhitungan dengan bantuan aplikasi Ms. Excel yang telah dikembangkan sendiri. Dari hasil pada worksheet Ms. Excel, penulis selanjutnya bisa meneruskan program yang telah dibuat atau bisa mengembangkan program tersebut dengan menggunakan data job, mesin, semut yang lebih banyak.

## DAFTAR RUJUKAN

- Florez, E., Gomez, W dan Bautista, L. 2013. An Ant Colony Optimization Algorithm for Job Shop Scheduling Problem. *International Journal of Artificial Intelligence & Applications*, (Online), 4(4):53-66, (<https://arxiv.org/abs/1309.5110>), diakses 1 Maret 2015.
- Gaviria, I., Morcillo, A dan Tejedor, J. 2013. An Ant Colony Optimization Algorithm for Microwave Corrugates Filters Design. *Journal of Computational Engineering*, (Online), 2013:1-9, (<http://www.hindawi.com/journals/jcengi/2013/942126/>), diakses 13 April 2016.
- Hartini, S., Zaini, E. dan Imran, A. 2014. Algoritma Penjadwalan Job Shop Alternatif Rounting menggunakan Variabel Neighbourhood Descent with Fixed Threshold untuk Meminimalisasi Makespan. *Jurnal Online Institut Teknologi Nasional*, (Online), 1(4):269-280, (<http://jurnalonline.itenas.ac.id/index.php/rekaintegra/article/viewFile/324/539>), diakses 4 Februari 2016.
- Hingrajika, K., Gupta, R., Chandel, C. 2012. An Ant Colony Optimization Algorithm for Solving Travelling Salesman Problem. *International Journal of Scientific and Research Publications*, (Online), 2(8): 1-6, (<http://jurnalonline.itenas.ac.id/index.php/rekaintegra/article/viewFile/324/539>), diakses 13 April 2016.
- Leguizamon, G dan Schutz, M. 2002. An Improved Ant Colony Algorithm for the Job Shop Scheduling Problem. (Online), ([http://Documento\\_completo.pdf](http://Documento_completo.pdf)), diakses 1 Februari 2016.
- Li, Y and Chen, Y. 2010. A Genetic Algorithm for Job Shop Scheduling. *Journal of Software*, (Online), 5(3): 269-274, (<http://mat.uab.cat/~alseda/MasterOpt/p11-31.pdf>), diakses 21 April 2016.
- Mishra, R dan Jaiswal, A. 2012. Ant Colony Optimization: A Solution of Load Balancing in Cloud. *International Journal of Web & Semantic Technology*,

- (Online), 3(2): 33-50, (<http://airccse.org/journal/ijwest/papers/3212ijwest03.pdf>)
- Narendhar, S dan Amudha, T. 2012. A Hybrid Bacterial Foraging Algorithm for Solving Job Shop Scheduling Problem. *International Journal of Programming Languages and Applications*, (Online), 2(4) : 1-11, (<http://arxiv.org/pdf/1211.4971>), diakses 21 April 2016.
- Pinedo, M dan Chao, X. 2002. *Operations Scheduling with Applications in Manufacturing dan Services*. New York : McGraw-Hill, 5(1):95-96.
- Shop Scheduling*. Jurnal Online, (<http://www.zbc.uz.zgora.pl/Content/2570/11mesg.pdf>), diakses 2 Januari 2016.
- Surekha, P dan Sumanthi, S. 2010. Solving Fuzzy based Job Shop Scheduling Problem Using GA and ACO. *Journal of Emerging Trends in Computing and Information Sciences*, (Online), 1(2): 95-102, ([http://www.cisjournal.org/archive/vol1no1/vol1no1\\_13.pdf](http://www.cisjournal.org/archive/vol1no1/vol1no1_13.pdf)), diakses 13 April 2016.
- Ventresca, M. dan Ombuki, B.,M. 2004. *Ant Colony Optimization for Job Shop Scheduling Problem*. Canada : Department of Computer Science, (Online), (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.3586&rep=rep1&type=pdf>), diakses 2 Januari 2016.
- Zwaan, S and Marques, C. 2000. *Ant Colony Optimization For Job Job Scheduling*. Jurnal (Online), (<https://www.cosc.brocku.ca/sites/all/files/downloads/research/cs0404.pdf>), diakses 2 Januari 2016.

