

## Algoritma *ant-lion optimizer* untuk meminimasi emisi karbon pada penjadwalan *flow shop dependent sequence set-up*

### *The ant-lion optimizer algorithm for minimizing carbon emissions in the flow scheduling shop dependent sequence set-up*

Dana Marsetiya Utama\*, Teguh Baroto, Dewi Maharani, Fathiha Raudhatul Jannah, Ricca Andhini Octaria  
Jurusan Teknik Industri, Fakultas Teknik, Universitas Muhammadiyah Malang,  
Jl. Raya Tlogomas No. 246 Malang 61514, Jawa Timur, Indonesia  
\* e-mail: dana@umm.ac.id



#### INFO ARTIKEL

##### Sejarah artikel:

Diterima:  
16 Februari 2019  
Direvisi:  
12 Juni 2019  
Diterbitkan:  
28 Juni 2019

##### Kata kunci:

*ant lion optimization*;  
penjadwalan;  
*flow shop*;  
*dependent sequence  
setup*;  
emisi karbon

##### Keywords:

*ant lion optimization*;  
*scheduling*;  
*flow shop*;  
*dependent sequence  
setup*;  
*carbon emissions*

#### ABSTRAK

Industri manufaktur akhir-akhir ini dituntut untuk memperhatikan isu lingkungan. Pemakaian energi pada produksi umumnya menghasilkan emisi karbon. Emisi karbon ini menjadi permasalahan di lingkungan. Untuk mengurangi pemakaian emisi karbon, penelitian ini menggabungkan metode penjadwalan dan emisi karbon sebagai solusi dalam masalah lingkungan. Kasus pada artikel ini adalah *flow shop dependent sequence set-up*. Jurnal ini mengusulkan algoritma baru *Ant Lion Optimizer* (ALO) yang terinspirasi oleh alam untuk meminimasi emisi karbon. Beberapa percobaan numerik dilakukan untuk mengetahui parameter terbaik dari Algoritma ALO. Untuk menguji keefektifan dari algoritma, Algoritma ALO ini dibandingkan dengan beberapa algoritma populer saat ini. Hasil percobaan numerik menunjukkan algoritma ALO efektif untuk meminimasi emisi karbon.

#### ABSTRACT

*Manufacture industry recently is required to pay attention of enviromental issue. The use of energy in production generally produces carbon emissions. This carbon emission is a problem in the environment. This study combines scheduling methods and carbon emissions as a solution to environmental issues to reduce the use of carbon emissions. The case in this article is the flow shop dependent sequence set-up. This journal proposes a new Ant Lion Optimizer (ALO) algorithm inspired by nature to minimize carbon emissions. Several numerical experiments were conducted to determine the best parameters of the ALO algorithm. This ALO algorithm is compared with several popular algorithms today. The numerical experiment results show that the ALO algorithm is useful for minimizing carbon emissions.*

© 2019 Penulis. Dipublikasikan oleh Baristand Industri Padang. Akses terbuka dibawah lisensi CC BY-NC-SA

## 1. Pendahuluan

Emisi karbon merupakan salah satu permasalahan pada lingkungan (Ma *et al.*, 2019). Emisi karbon memiliki dampak bahaya terhadap lingkungan, seperti perubahan iklim, dan peningkatan suhu global (Du and Lin, 2018). Sektor perindustrian menyumbang sekitar setengah dari total emisi karbon dunia. Hal ini terjadi karena, sebagian besar industri manufaktur melakukan proses produksi menggunakan mesin, dan berbahan bakar fosil (Fang *et al.*, 2011). Dengan demikian,

industri manufaktur dituntut agar memunculkan solusi terhadap permasalahan yang ada. Karena isu lingkungan ini, penelitian ini diharapkan mampu menjadi sebuah solusi bagi industri untuk lingkungan.

Industri manufaktur dapat menerapkan mesin hemat energi untuk mengurangi emisi karbon (Elias *et al.*, 2019). Namun, hal tersebut menimbulkan biaya yang mahal, dan dibutuhkan sumber daya manusia yang sebagian besar tidak dimiliki oleh industri manufaktur kecil dan menengah (Tian *et al.*, 2018b). Sementara itu, beberapa peneliti menggunakan penjadwalan produksi

untuk meminimasi emisi karbon (Zhang *et al.*, 2017, Xu *et al.*, 2018). Penjadwalan merupakan aktivitas pengaturan mesin-mesin untuk menjalankan kumpulan pekerjaan (Utama, 2017, Garside *et al.*, 2018, Harto *et al.*, 2016). Sebagian besar konsumsi energi dihasilkan pada saat mesin dalam keadaan menganggur atau *idle* (Utama *et al.*, 2019). Sehingga, beberapa peneliti mengusulkan untuk mematikan mesin saat menganggur untuk mengurangi konsumsi energi (Tian *et al.*, 2018a). Hal ini memiliki dampak langsung terhadap pengeluaran emisi karbon (Utama, 2019). Masalah dalam penjadwalan umumnya memiliki kinerja minimasi waktu penyelesaian (makespan) (Masudin *et al.*, 2014), keterlambatan, mesin *idle*, konsumsi energi (Utama, 2018). Engin and Güçlü (2018) menyelesaikan permasalahan penjadwalan dengan kriteria meminimalkan waktu penyelesaian dan waktu komputasi.

Dalam beberapa tahun terakhir, algoritma metaheuristik digunakan sebagai teknik untuk mendapatkan solusi yang optimal pada masalah penjadwalan (Pan *et al.*, 2017). Lebih lanjut, teknik ini biasanya menggunakan pendekatan stokhastik dalam menyelesaikan permasalahan. Algoritma metaheuristik biasa digunakan dalam masalah penjadwalan seperti algoritma genetik (GA) (Yu *et al.*, 2018), *Particle Swarm Optimization* (PSO) (Wu *et al.*, 2018), *Ant Colony Optimization* (ACO) (Nguyen *et al.*, 2017), *Simulated Annealing* (SA) (Firdaus *et al.*, 2015, Husen *et al.*, 2015), *Algoritma Immune system* (AIS) (Nasution *et al.*, 2017). Beberapa penelitian penjadwalan untuk meminimasi emisi karbon mulai dilakukan oleh peneliti. Liu *et al.* (2017) menggunakan Algoritma *Fruit Fly* (FF) untuk menyelesaikan kasus *job shop*. Piroozfard *et al.* (2018) menggunakan Algoritma GA untuk menyelesaikan kasus *job shop*. Ding *et al.* (2016) mengusulkan angoritma baru NEH untuk menyelesaikan masalah *flow shop*.

Berdasarkan penelitian terdahulu, penelitian tentang penjadwalan untuk minimasi emisi karbon masih sangat minim. Belum ada peneliti yang menggunakan *Ant Lion Optimizer* (ALO) untuk meminimasi emisi karbon pada penjadwalan *flow shop*. Peneliti mengusulkan algoritma baru yang disebut *Ant Lion Optimizer* (ALO) sebagai pendekatan alternatif untuk memecahkan masalah penjadwalan *flow shop*. Algoritma ALO terinspirasi dari perilaku cerdas *ant-lion* saat berburu di alam (Mirjalili, 2015). ALO ini digunakan untuk meminimasi emisi karbon dalam penjadwalan produksi dengan waktu *setup* mesin tergantung dari urutan pekerjaan. Lebih lanjut, setiap urutan pekerjaan akan memiliki waktu *setup* yang berbeda.

## 2. Metode

### 2.1. Asumsi masalah

Permasalahan penjadwalan *Flow Shop Sequence Dependent Setup* (FSSDS) ini memiliki waktu proses  $t_{ij}$ , dan  $n_{job}$  ( $n=1,2,3,\dots,i$ ) yang dikerjakan pada mesin  $m$  ( $m=1,2,3,\dots,j$ ). Beberapa asumsi FSSDS adalah (1) urutan  $n$  *job* ( $n = 1,2,3,\dots,j$ ) yang dikerjakan pada tiap  $m$  mesin ( $m = 1,2,3,\dots,i$ ) adalah sama. (2) waktu proses untuk tiap *job* adalah  $P_{ij}$ .  $P_{ij}$  menunjukkan waktu proses *job* ke-

$i$  dan dikerjakan di mesin ke- $j$ . (3) Semua mesin tersedia pada saat  $t=0$ . (4) Waktu *setup* dependent terhadap urutan pekerjaan dan terpisah dengan waktu proses. (5) waktu *setup job*  $i-1$  ke *job*  $i$  pada mesin  $j$  adalah  $S_{(i-1,i)}$ .

Lebih lanjut,  $S_i$  menunjukkan waktu *setup* untuk *job*  $i$  jika *job*  $i$  tersebut adalah *job* urutan pertama. (6) Tiap *job* ketika mulai diproses sampai selesai harus sesuai urutan tidak boleh disela. (7) Tiap mesin dimulai di waktu = 0 dan selesai ketika pekerjaan terakhir pada tiap mesin selesai (tiap mesin yang berhenti *independent* dengan mesin yang lain). Notasi yang digunakan dalam penjadwaan minimasi emisi karbon adalah sebagai berikut:

- $i$  : indeks dari job,  $i = 1,2 \dots,n$
- $j$  : indeks dari mesin,  $j = 1,2 \dots,m$
- $n$  : total jobs.
- $m$  : total mesin
- $S_i$  : Waktu *setup job*  $i$  pada urutan 1 (kilo watt)
- $S_{(i-1,i)}$  : Waktu *setup* untuk *job*  $i-1$  dan urutan *job* selanjutnya  $i$ (jam)
- $ST_i$  : Total waktu *setup*
- $P_{i,j}$  : Waktu proses *job* urutan  $i$  pada mesin  $j$  (jam)
- $Pe_j$  : konsumsi energi pemrosesan pada mesin  $j$  (kilo watt)
- $Se_j$  : konsumsi energi *setup* pada mesin  $j$  (kilo watt)
- $Ie_j$  : konsumsi energi menganggur pada mesin  $j$  (kilo watt)
- $C_{i,j}$  : Waktu penyelesaian *job* urutan  $i$  pada mesin  $j$
- $T_j$  : Waktu penyelesaian pada mesin  $j$
- $B_j$  : Total waktu sibuk pada mesin  $j$
- $I_j$  : Total waktu menganggur pada mesin  $j$
- $CE_j$  : Emisi karbon yang dikeluarkan pada mesin  $j$  (kg/kilo wall)
- TCE : total *emisi karbon* (kg)

Tujuan dari model ini adalah untuk meminimalkan emisi karbon (TCE) (Galinato and Yoder, 2010); Berikut adalah formula FSSDS untuk meminimasi emisi karbon :

$$C_{1,1} = P_{1,1} + S_1 \tag{1}$$

$$C_{1,j} = \max(C_{1,j-1}, S_1) + P_{1,j}, j = 2 \dots m \tag{2}$$

$$C_{i,1} = C_{i-1,1} + S_{i-1,i} + P_{i,1}, i = 2 \dots n \tag{3}$$

$$C_{i,j} = \max(C_{i,j-1}, S_{i-1,i} + C_{i-1,j}) + P_{i,j}, i = 2 \dots n, j = 2 \dots m \tag{4}$$

$$B_j = \sum_{i=1}^n P_{ij} \tag{5}$$

$$ST_j = S_i + \sum_{i=2}^m S_{i-1,i} \tag{6}$$

$$I_j = T_j - (B_j + ST_j) \tag{7}$$

$$TCE = \sum_{j=1}^n (B_j \cdot Pe_j \cdot CE_j + ST_j \cdot Se_j \cdot CE_j + I_j \cdot Ie_j \cdot CE_j) \tag{8}$$

Persamaan (1) menggambarkan formula waktu penyelesaian *job* urutan 1 di mesin 1. Persamaan (2) menunjukkan waktu penyelesaian *job* urutan 1 di mesin 2 sampai  $j$ . Persamaan (3) menjelaskan waktu penyelesaian *job* urutan  $i$  di mesin 1. Persamaan (4) menggambarkan waktu penyelesaian *job* urutan  $i$  mesin  $j$ . Persamaan (5) menjelaskan waktu sibuk mesin  $j$ . Persamaan (6) menggambarkan total waktu *setup* di mesin

j. Persamaan (7) menunjukkan total waktu *idle* pada mesin  
 j. Persamaan (8) menggambarkan formula untuk menghitung emisi karbon total.

**2.2. Definisi masalah**

Permasalahan pada penelitian ini adalah untuk meminimasi emisi karbon yang ditimbulkan oleh penjadwalan *flow shop* dengan *dependent setup time*. Waktu *setup* tergantung dari urutan pekerjaan dan berbeda dengan waktu proses. Terdapat 3 urutan pekerjaan yaitu J1-J2-J3 dan J3-J2-J1. Data contoh permasalahan dapat dilihat pada tabel 1, tabel 2, tabel 3 dan tabel 4. Gambar 1 a menunjukkan waktu penyelesaian setiap pekerjaan di tiap mesin. Jika urutan J1-J2-J3, waktu penyelesaian untuk mesin 1 adalah 16, 17 untuk mesin 2, dan 18 untuk mesin 3. Gambar 1 b menggambarkan waktu penyelesaian setiap pekerjaan dengan urutan J3-J2-J1. Waktu penyelesaiannya adalah 15 untuk mesin 1, 16 untuk mesin 2, dan 18 untuk mesin 3. Dari urutan pekerjaan pada Gambar 1, urutan *job* tersebut memiliki *completion time* (makespan) yang sama. *Completion time* pada Gambar 1 a dan b adalah 18. Pada gambar 1 a, mesin 1 membutuhkan waktu proses sebesar 6, waktu *setup* sebesar 9, dan waktu *idle* sebesar 0. Lebih lanjut, konsumsi energi waktu proses, *setup*, dan *idle* yang dibutuhkan untuk pekerjaan itu adalah 30, 18, dan 0. Oleh karena itu, emisi karbon yang dihasilkan pada mesin 1 adalah 39,10. Melalui proses yang sama, hasil perhitungan total emisi karbon yang dihasilkan untuk urutan J1-J2-J3 adalah sebesar 101,14 kg. Lebih lanjut, pada urutan J3-J2-J1 menghasilkan total emisi karbon sebesar 72,69 kg. Meskipun memiliki *completion time* sama, gambar 1 b menghasilkan emisi karbon lebih efisien dibandingkan gambar 1 a.

Tabel 1.  
Waktu Proses (t)

Job	Mesin		
	1	2	3
1	1	1	3
2	3	1	1
3	2	1	2

Tabel 2.  
Waktu *setup* pada pekerjaan pertama (s1)

Job	Set Up
1	1
2	2
3	2

Tabel 3.  
Waktu *setup* dari job j ke job k (s2)

Job (j)	Job (k)		
	1	2	3
1	3	5	2
2	4	2	4
3	3	3	4

Tabel 4.  
Energi per satuan waktu

Konsumsi	Mesin		
	1	2	3
Energi <i>setup</i>	3	3	3
Energi <i>idle</i>	1	1	1
Energi proses	5	5	5
Emisi karbon	0.81472	0.90579	0.12699

**2.3. Algoritma usulan**

Algoritma ALO meniru interaksi antara *antlions* dan semut dalam suatu perangkat (Umamaheswari *et al.*, 2017). Untuk memodelkan interaksi tersebut, semut diharuskan bergerak di atas ruang pencarian, dan antlion diizinkan untuk memburunya. semut mencari makan dengan berpindah-pindah. Pergerakan semut ini diberikan model pada persamaan (9).

$$X(t) = [0, \text{cumsum}(2r(t1) - 1), \text{cumsum}(2r(tn) - 1)] \tag{9}$$

*Cumsum* merupakan perhitungan dari kumulatif sum. Selanjutnya, *n* merupakan nilai maksimum dari angka iterasi. *t* menunjukkan langkah dari bilangan acak. *r(t)* adalah fungsi stokastik yang diformulasikan pada persamaan (10).

$$r(t) = \begin{cases} 1, & \text{if } rand > 0,5 \\ 0, & \text{if } rand \leq 0,5 \end{cases} \tag{10}$$

Posisi untuk semut disajikan dengan matrix  $M_{Ant}$  (persamaan (11)).

$$M_{Ant} = \begin{bmatrix} A1.1 & A1.2 & \dots & A1,d \\ A2.1 & A2.2 & \dots & A2,d \\ \vdots & \vdots & \ddots & \vdots \\ An.1 & An.2 & \dots & An,d \end{bmatrix} \tag{11}$$

$A_{ij}$  menggambarkan nilai dari variabel *j*-th. Lebih lanjut, *i*-th merupakan nomor dari semut. Fungsi fitness untuk semut dijadikan matrix  $M_{OA}$  (persamaan 12):

$$M_{OA} = \begin{bmatrix} f([A1.1, A1.2, \dots, A1,d]) \\ f([A2.1, A2.2, \dots, A2,d]) \\ \vdots \\ f([An,1, An,2, \dots, An,d]) \end{bmatrix} \tag{12}$$

$A_{(n,d)}$  menggambarkan *n*, untuk mesin. *d* menggambarkan nomor operator. Untuk posisi dari antlion dimasukkan kedalam matrix  $M_{Antlion}$  (persamaan (13)). Untuk fungsi fitness dari antlion dimasukkan ke matrix  $M_{OAL}$  (persamaan (14)):

$$M_{Antlion} = \begin{bmatrix} AL1,1 & AL1,2 & \dots & AL1,d \\ AL2,1 & AL2,2 & \dots & AL2,d \\ \vdots & \vdots & \ddots & \vdots \\ ALn,1 & ALn,2 & \dots & ALn,d \end{bmatrix} \tag{13}$$

$$M_{OAL} = \begin{bmatrix} f([AL1,1 . AL1,2 , \dots , AL1, d]) \\ f([AL2,1 . AL2,2 , \dots , AL2, d]) \\ \vdots \\ f([ALn, 1 . ALn, 2 , \dots , ALn, d]) \end{bmatrix} \quad (14)$$

$AL_{(n,d)}$  menunjukkan untuk nomor dari antlion,  $d$  menunjukkan variabel dan  $f$  adalah fungsi objektif. Antlion berjalan dengan posisi acak di setiap langkah. Persamaan (15) menjelaskan formula untuk memastikan langkah perjalanan dari antlion ini tetap berada di lingkup pencarian. Formula ini menggunakan perhitungan min-max normalisasi [21].

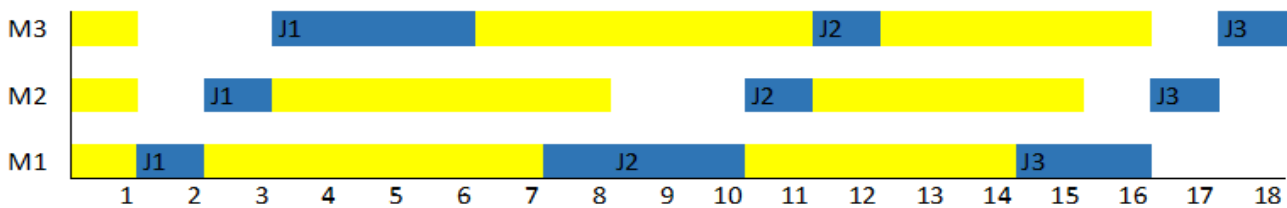
$$X_i^t = \frac{(X_i^t - a_i) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i, \quad (15)$$

$a_i$  merupakan nilai minimum dari jalannya antlion.  $c_i$  merupakan nilai minimum dari iterasi.  $d_i^t$  merupakan nilai maksimum dari iterasi. Perangkap semut pada antlion dimasukkan kedalam model matematika sebagai berikut (Jain *et al.*, 2018) (persamaan (16)):

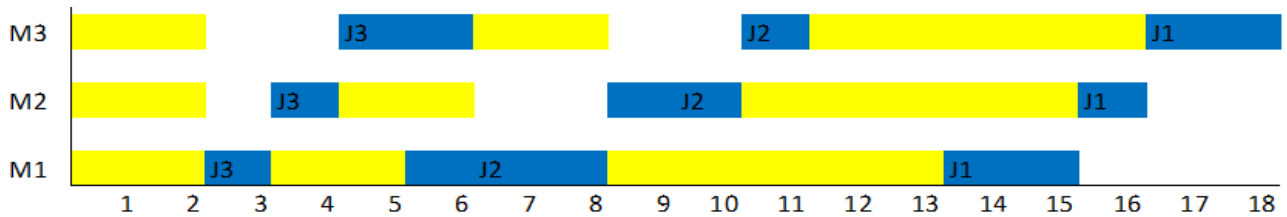
$$c_i^t = Antlion_j^t + c^t, \quad d_i^t = Antlion_j^t + d^t \quad (16)$$

$c^t$  merupakan nilai minimum dari iterasi.  $d^t$  merupakan nilai maksimum dari iterasi. Pemburuan Antlion dimodelkan dengan *roulette wheel*. Model pemberuan tersebut adalah sebagai berikut (persamaan 17):

$$c^t = \frac{c^t}{I}, \quad d^t = \frac{d^t}{I}, \quad I = 10^w \cdot \frac{t}{T} \quad (17)$$



(a)



(b)

Gambar 1. Gant chart untuk urutan J1-J2-J3 (a), Gant chart untuk urutan J3-J2-J1 (b)

$t$  merupakan nilai iterasi yang terpilih.  $T$  adalah nilai maksimum dari iterasi.  $w$  merupakan nilai iterasi kontan yang digunakan berdasarkan :  $w = 2$ , if  $t > 0.1T$ ,  $w = 3$ , if  $t > 0.5T$ ,  $w = 5$  if  $t > 0.9T$ ,  $w = 6$  if  $t > 0.95T$ . Antlion menangkap mangsa dan membuat kembali lubangnya untuk mendorong semut agar masuk ke dalam pasir. Sifat tersebut dimodel sebagai berikut (persamaan (18)):

$$Antlion_j^t = Ant_i^t, \text{ if } f(Ant_i^t) > f(Antlion_j^t) \quad (18)$$

$t$  menunjukkan iterasi saat ini.  $Antlion_j^t$  menunjukkan posisi dari antlion pada iterasi terpilih. Setiap semut berjalan secara acak di sekitar antlion dan dengan posisi dimodelkan sebagai berikut (19):

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (19)$$

Dimana  $R_A^t$  menggambarkan posisi semut ketika berjalan disekitar antlion dengan menggunakan iterasi roulette wheel ke- $t$ .  $R_E^t$  menunjukkan posisi dimana semut berjalan di sekitar elite antlion pada iterasi ke- $t$ .

Kode pseudo dari algoritma ALO adalah sebagai berikut:

```

Inisialisasi populasi dan random permutasi pertama dari
semut dan antlions secara acak
Hitung nilai fitness dari semut dan antlions menggunakan
persamaan (8)
Temukan antlion terbaik dan asumsi sebagai elite (yang
paling optimum)
while kondisi akhir tidak terpenuhi
for semua semut
Pilih antlion menggunakan Roulette Wheel
Update c dan d Persamaan (17)
Buatlah jalan secara acak dan normalisasi
menggunakan persamaan (9) dan (10)
Update posisi dari semut menggunakan persamaan (19)
end for
Hitung nilai fitness dari semua semut
Ganti antlion dengan semut yang sesuai jika itu terlihat
lebih baik Persamaan (18)
Update elite jika antlion lebih baik dibandingkan dengan
yang elite
end while
Return elite
    
```

## 2.4. Prosedur percobaan

Penelitian ini menggunakan 7 variasi job. Selain itu, penelitian ini menggunakan 3 variasi jumlah mesin. Total lingkungan percobaan pada penelitian ini adalah 21. Waktu proses, *set up*, dibangkitkan dari bilangan acak distribusi uniform. Konsumsi energi operasi, *set up*, dan *idle* dibangkitkan dari bilangan acak distribusi uniform. Selanjutnya, emisi karbon juga dihasilkan bilangan acak distribusi uniform. Waktu proses dibangkitkan dengan nilai parameter uniform (10,50). Waktu *set up* pada urutan pertama dibangkitkan dengan nilai parameter uniform (1,10). Waktu *setup* untuk urutan dari job  $i - 1$

ke job  $i$  diperoleh dengan nilai parameter uniform (1,10). Konsumsi energi proses diperoleh dengan nilai parameter uniform (5,10). Konsumsi energi *setup* diperoleh dengan nilai parameter uniform (1,3). Konsumsi energi mesin menganggur diperoleh dengan nilai parameter uniform (1,2). Emisi karbon setiap mesin diperoleh dengan nilai parameter uniform (0,1).

Penelitian ini juga menguji parameter dari ALO. Pada percobaan parameter, parameter yang digunakan dalam percobaan adalah populasi dan iterasi. Faktor populasi terdiri 3 level yaitu populasi 10, 50, dan 100. Faktor Iterasi terdiri 5 level yaitu iterasi 10, 50, 100, 200, dan 500. Untuk 1 jenis atribut percobaan dilakukan 15 kali perlakuan. Penelitian memiliki 21 lingkungan percobaan yang berbeda. Sehingga, percobaan yang dilakukan sebanyak 315 kali perlakuan.

Untuk menguji keefektifan dari algoritma, penelitian juga melakukan perbandingan dengan beberapa algoritma. Algoritma yang dipilih untuk perbandingan adalah *simulated annealing* dan algoritma genetika. Algoritma-algoritma tersebut merupakan algoritma yang populer saat ini. Parameter ALO yang digunakan adalah 100 populasi dan 500 iterasi. Uji perbandingan menggunakan permasalahan 100 job dan 16 mesin. Permasalahan tersebut dilakukan perulangan 20 kali. Uji perbandingan menggunakan *independent sample t test*.

## 3. Hasil dan pembahasan

Pada jurnal ini, tes numerik dilakukan dalam perangkat lunak Matlab R16 pada windows 8.1 prosesor AMD 4 x86-64 4 GB. Berbagai variasi populasi, job, mesin dan jumlah iterasi digunakan untuk mencari kemungkinan paling optimal untuk meminimasi emisi karbon.

## 3.1. Pengujian parameter ALO terhadap karbon emisi

Gambar 2 hingga gambar 4 merupakan rekapitulasi perhitungan emisi karbon. Berdasarkan hasil perhitungan emisi karbon pada berbagai percobaan (gambar 2, gambar 3 dan gambar 4), terdapat beberapa temuan dari hasil percobaan. Semakin besar job dan ketika jumlah mesin sama, maka jumlah emisi karbon yang dihasilkan semakin besar. Semakin besar jumlah mesin dan ketika jumlah job sama, maka nilai emisi karbon yang dihasilkan akan semakin besar. Semakin besar populasi yang digunakan, maka nilai emisi karbon yang dihasilkan semakin kecil. Sehingga, emisi karbon dipengaruhi oleh jumlah populasi. Apabila iterasi yang digunakan semakin besar, maka nilai carbon emisinya akan semakin kecil. Pada masalah job kecil, parameter yang sebaiknya digunakan adalah populasi kecil dan iterasi kecil. Lebih lanjut, pada masalah job besar, parameter yang sebaiknya digunakan adalah populasi besar dan iterasi besar.

## 3.2. Pengujian parameter ALO terhadap waktu komputasi

Pada gambar 5, gambar 6 dan gambar 7 merupakan rekapitulasi waktu komputasi untuk berbagai kemungkinan populasi, job, mesin, dan jumlah iterasi. Temuan dari beberapa percobaan adalah sebagai berikut. Semakin besar job yang digunakan maka waktu komputasi yang dibutuhkan semakin besar. Semakin besar jumlah mesin yang digunakan, maka waktu komputasi yang diperlukan semakin besar. Semakin besar jumlah populasinya, maka waktu komputasi semakin besar. Sehingga, waktu komputasi dipengaruhi oleh jumlah populasi. Jika iterasi semakin besar, maka waktu komputasi juga semakin besar. Sehingga, waktu komputasi dipengaruhi oleh jumlah iterasinya.

## 3.3. Pengujian ALO terhadap algoritma lain

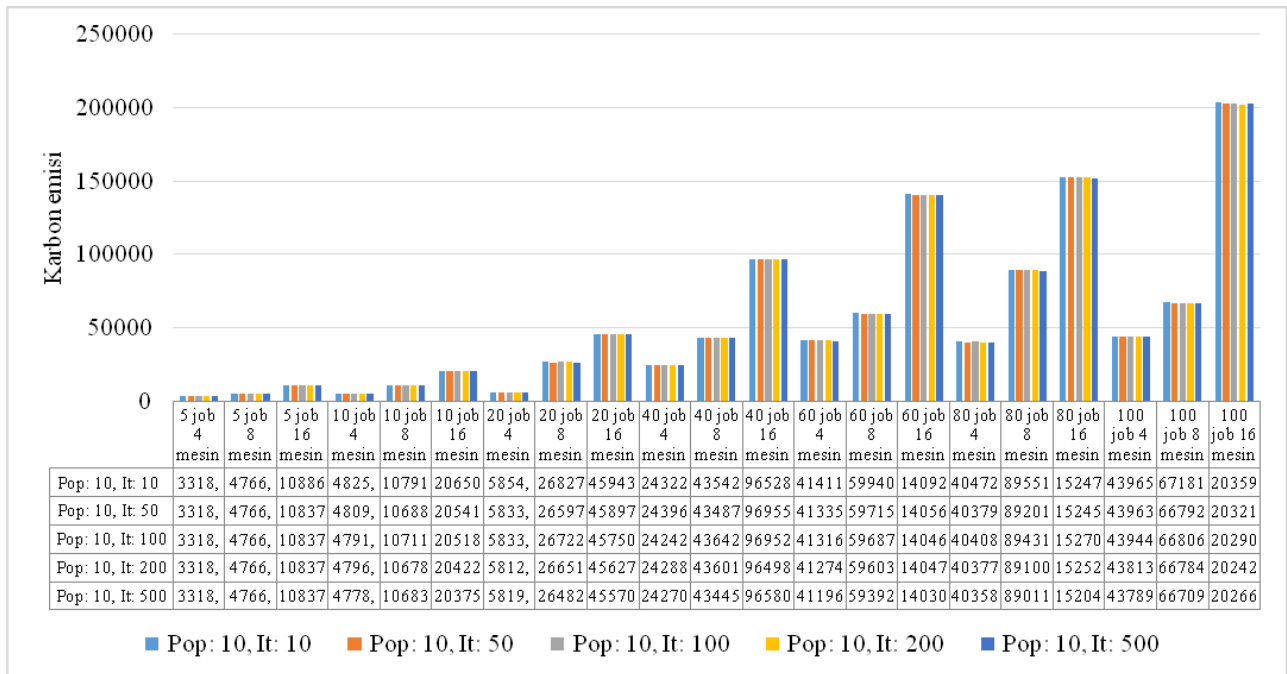
Berdasarkan uji *independent sample t-test* (tabel 5 dan tabel 6), Algoritma ALO memiliki kinerja yang lebih baik dibandingkan algoritma GA dan SA. Nilai rata-rata emisi karbon ALO yang dihasilkan dalam percobaan adalah 200088,90. Lebih lanjut, algoritma GA dan SA menghasilkan emisi karbon 205950,09 dan 207566,97. Dari uji statistik *independent t test* menunjukkan algoritma ALO memiliki perbedaan yang signifikan terhadap algoritma GA dan SA.

Tabel 5.  
Ringkasan deskriptif dari perbandingan algoritma

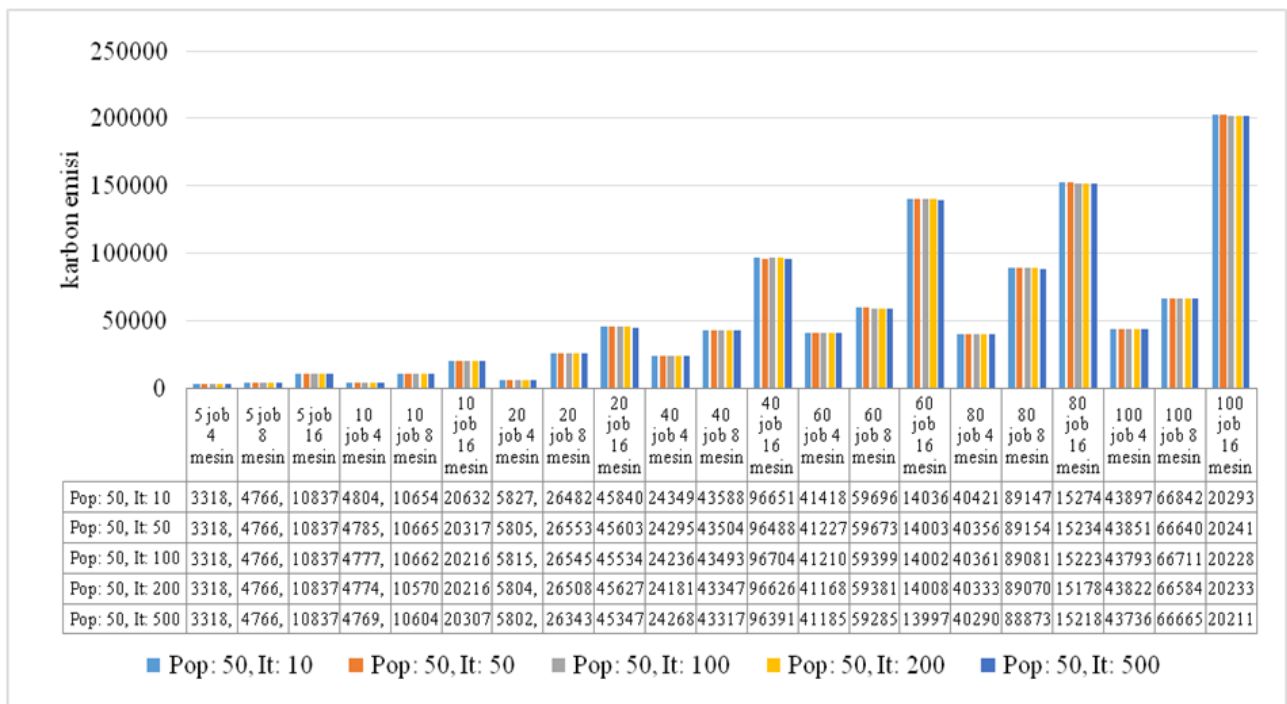
Algoritma	Jumlah percobaan	Rata-rata emisi karbon	Standard deviation	Standard error mean
ALO	20	200088,90	1854,68	414,72
GA	20	205950,09	4036,87	902,67
SA	20	207566,97	3537,30	790,96

Tabel 6.  
Hasil uji independent *t test*

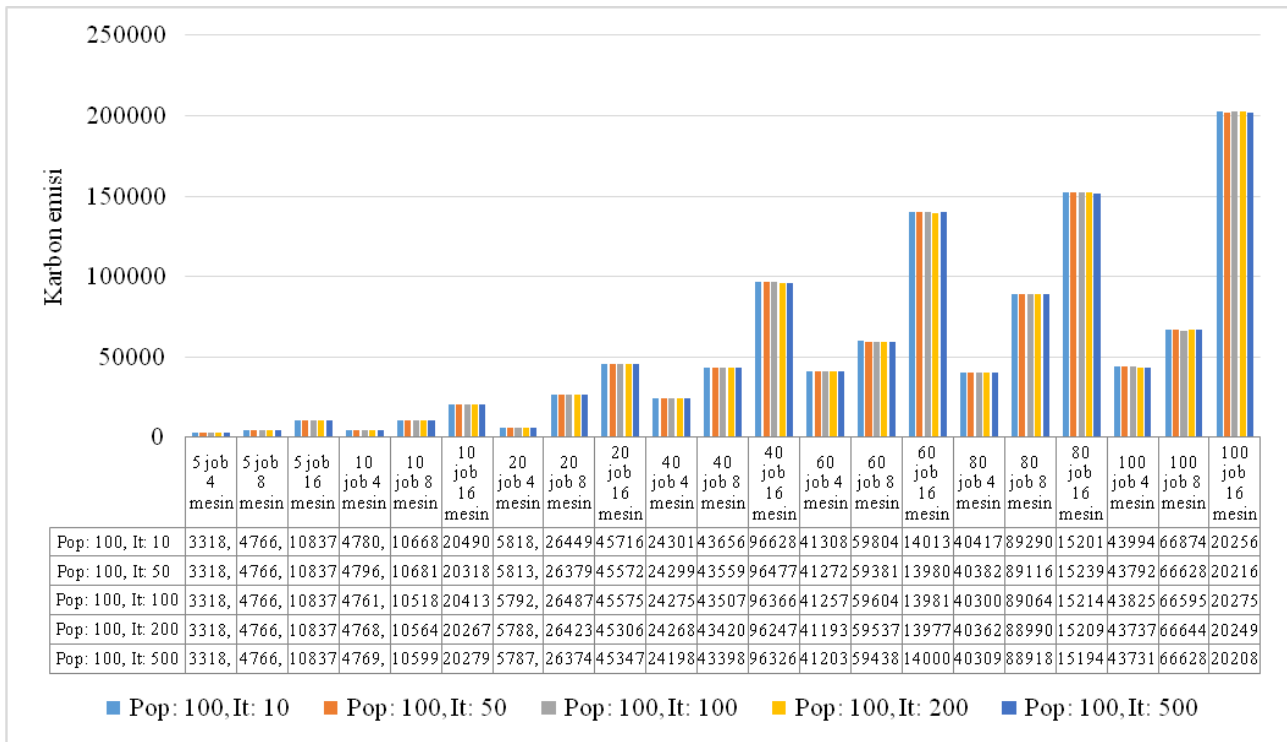
Algoritma	nilai t hitung	Significant. (2-tailed)	Kesimpulan
ALO-GA	-5,900	0,000	Terdapat perbedaan
ALO-SA	-8,373	0,000	Terdapat perbedaan



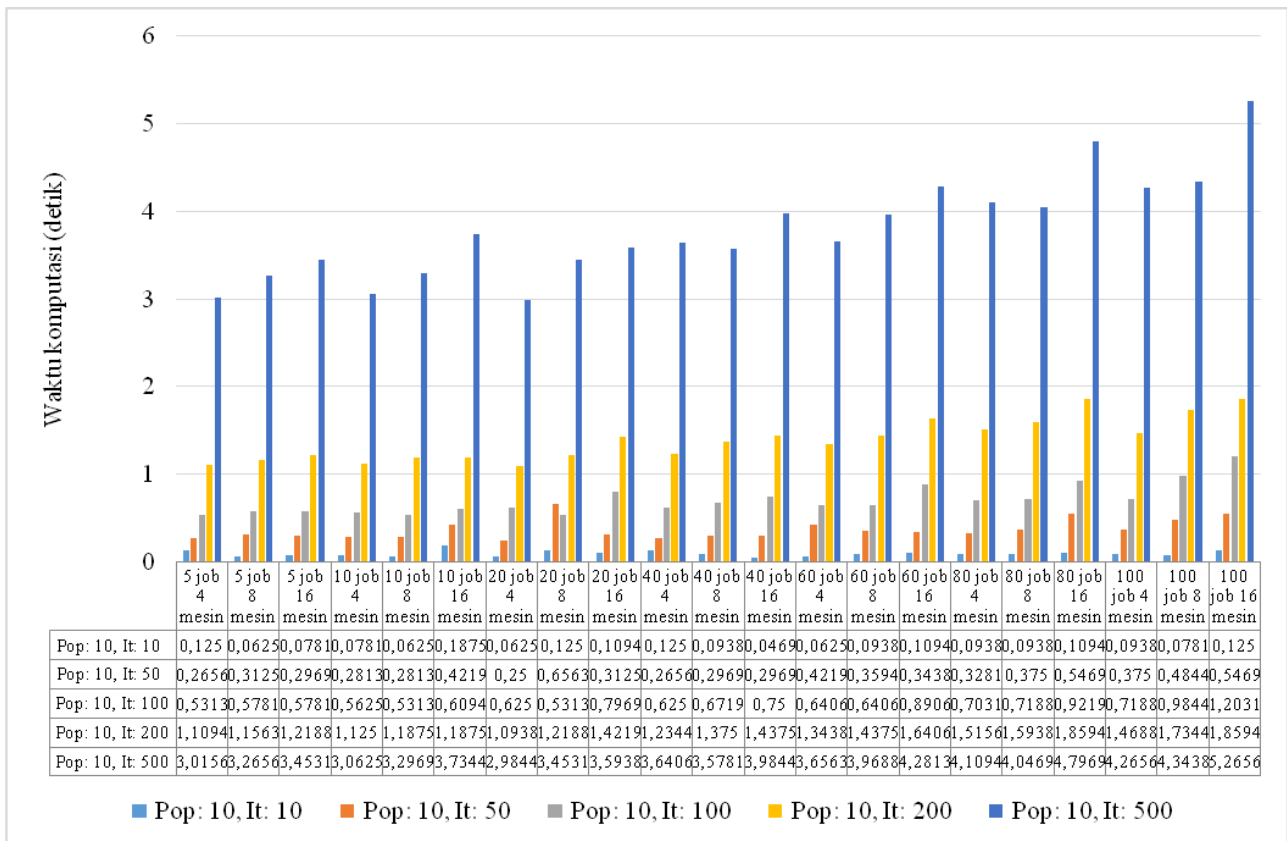
Gambar 2. Perhitungan emisi karbon pada populasi 10



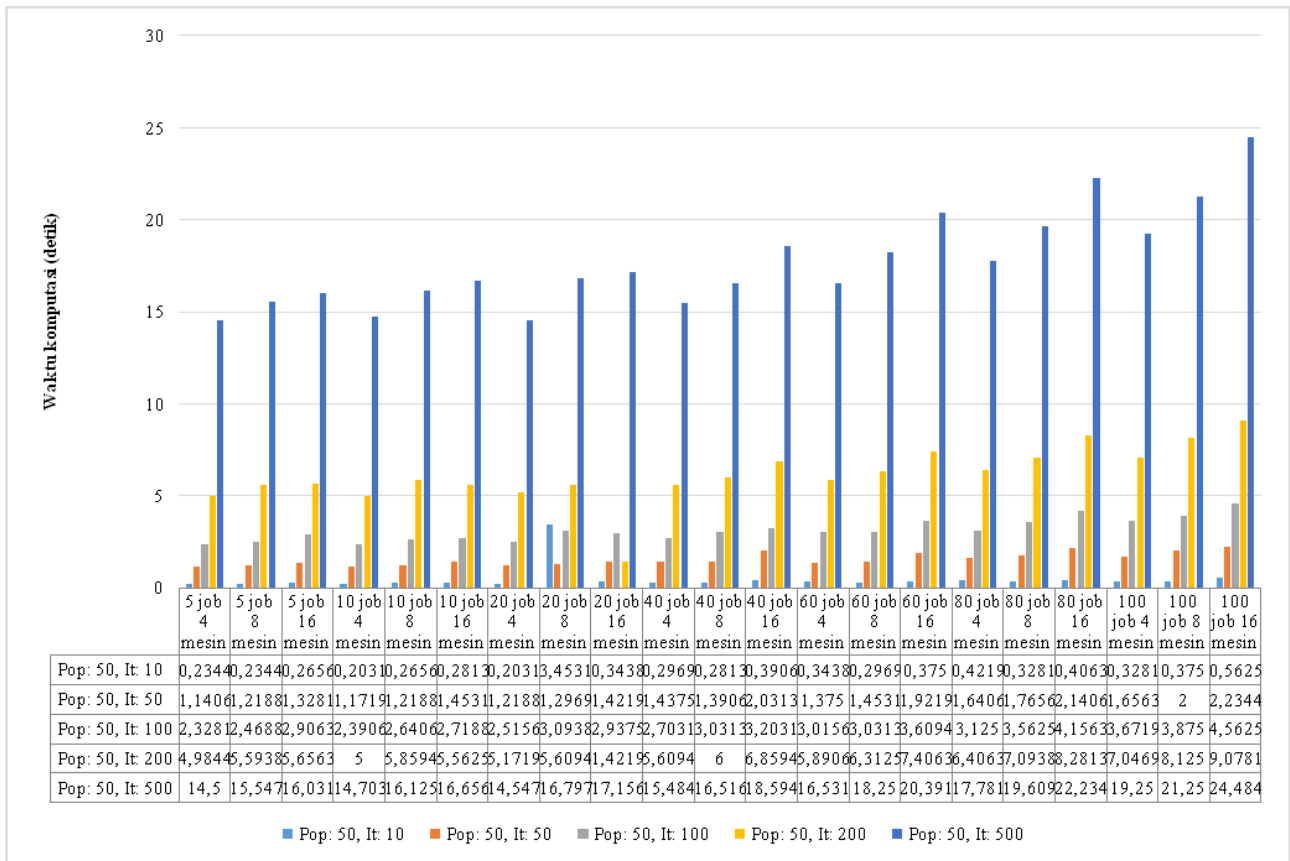
Gambar 3. Perhitungan emisi karbon pada populasi 50



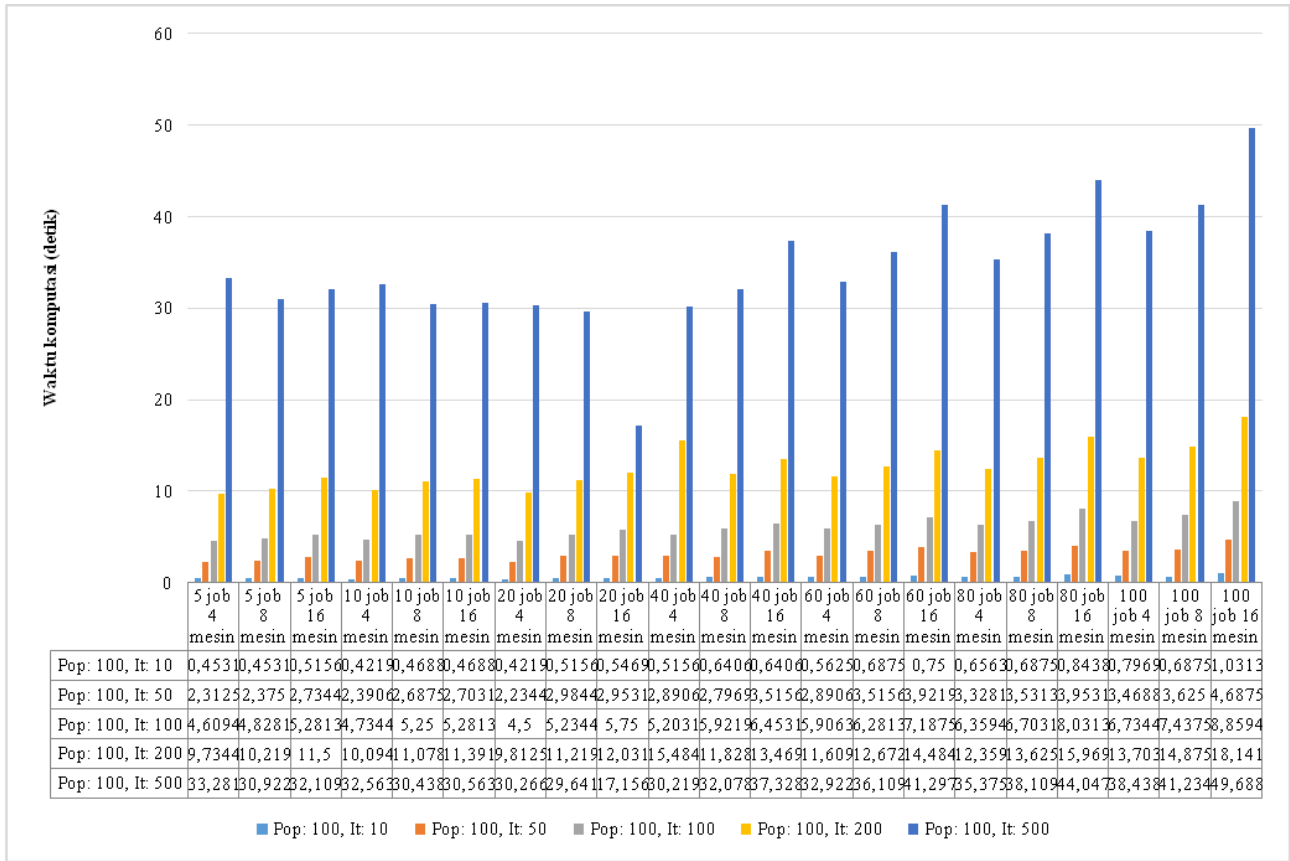
Gambar 4. Perhitungan emisi karbon pada populasi 100



Gambar 5. Waktu komputasi pada populasi 10



Gambar 6. Waktu komputasi pada populasi 50



Gambar 7. Waktu komputasi pada populasi 100



#### 4. Kesimpulan

Studi ini mengusulkan algoritma *Ant Lion Optimizer* untuk meminimalkan emisi karbon. Hasil percobaan menunjukkan algoritma ALO efektif untuk meminimasi emisi karbon pada masalah *flow shop*. Dari uji parameter ALO menunjukkan semakin besar populasi, maka emisi karbon yang dihasilkan semakin kecil. Apabila iterasinya semakin besar maka nilai carbon emisinya semakin kecil. Saran untuk penelitian selanjutnya adalah mengimprove algoritma dengan memberikan tahapan baru yang lebih efektif untuk mengurangi waktu komputasi.

#### Daftar pustaka

- Ding, J.-Y., Song, S. & Wu, C. 2016. Carbon-efficient scheduling of flow shops by multi-objective optimization. *European Journal of Operational Research*, 248, 758-771.
- Du, Z. & Lin, B. 2018. Analysis of carbon emissions reduction of China's metallurgical industry. *Journal of Cleaner Production*, 176, 1177-1184.
- Elias, R., Yuan, M., Wahab, M. & Patel, N. 2019. Quantifying saving and carbon emissions reduction by upgrading residential furnaces in Canada. *Journal of Cleaner Production*, 211, 1453-1462.
- Engin, O. & Güçlü, A. 2018. A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems. *Applied Soft Computing*, 72, 166-176.
- Fang, K., Uhan, N., Zhao, F. & Sutherland, J. W. 2011. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems*, 30, 234-240.
- Firdaus, M., Masudin, I. & Utama, D. M. 2015. Penjadwalan *flow shop* dengan menggunakan simulated annealing. *Spektrum Industri*, 13.
- Galinato, G. I. & Yoder, J. K. 2010. An integrated tax-subsidy policy for carbon emission reduction. *Resource and Energy Economics*, 32, 310-326.
- Garside, A. K., Utama, D. M. & Arifin, M. R. 2018. Penjadwalan produksi *flow shop* menggunakan algoritma branch and bound untuk meminimasi mean tardiness. Prosiding SENTRA (Seminar Teknologi dan Rekayasa), 2018.
- Harto, S., Garside, A. K. & Utama, D. M. 2016. Penjadwalan produksi menggunakan algoritma jadwal non delay untuk meminimalkan makespan studi kasus di cv. Bima mebel. *Spektrum Industri*, 14.
- Husen, M., Masudin, I. & Utama, D. M. 2015. Penjadwalan *job shop* statik dengan metode simulated annealing untuk meminimasi waktu makespan. *Spektrum Industri*, 13.
- Jain, M., Singh, V. & Rani, A. 2018. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm and Evolutionary Computation*.
- Liu, Q., Zhan, M., Chekem, F. O., Shao, X., Ying, B. & Sutherland, J. W. 2017. A hybrid fruit fly algorithm for solving flexible job-shop scheduling to reduce manufacturing carbon footprint. *Journal of Cleaner Production*, 168, 668-678.
- Ma, X., Wang, C., Dong, B., Gu, G., Chen, R., Li, Y., Zou, H., Zhang, W. & Li, Q. 2019. Carbon emissions from energy consumption in China: Its measurement and driving factors. *Science of the total environment*, 648, 1411-1420.
- Masudin, I., Utama, D. M. & Susastro, F. 2014. Penjadwalan *flow shop* menggunakan algoritma nawaz enscore ham.
- Mirjalili, S. 2015. The ant lion optimizer. *Advances in Engineering Software*, 83, 80-98.
- Nasution, R., Garside, A. K. & Utama, D. M. 2017. Penjadwalan *job shop* dengan pendekatan algoritma artificial immune system. *Jurnal Teknik Industri*, 18, 29-42.
- Nguyen, D. C. H., Ascough Ii, J. C., Maier, H. R., Dandy, G. C. & Andales, A. A. 2017. Optimization of irrigation scheduling using ant colony algorithms and an advanced cropping system model. *Environmental Modelling & Software*, 97, 32-45.
- Pan, Q.-K., Gao, L., Li, X.-Y. & Gao, K.-Z. 2017. Effective metaheuristics for scheduling a hybrid flow shop with sequence-dependent setup times. *Applied Mathematics and Computation*, 303, 89-112.
- Piroozfard, H., Wong, K. Y. & Wong, W. P. 2018. Minimizing total carbon footprint and total late work criterion in flexible *job shop* scheduling by using an improved multi-objective genetic algorithm. *Resources, Conservation and Recycling*, 128, 267-283.
- Tian, W., He, M., Guo, W., Huang, W., Shi, X., Shang, M., Toosi, A. N. & Buyya, R. 2018a. On minimizing total energy consumption in the scheduling of virtual machine reservations. *Journal of Network and Computer Applications*, 113, 64-74.
- Tian, Y., Xiong, S., Ma, X. & Ji, J. 2018b. Structural path decomposition of carbon emission: A study of China's manufacturing industry. *Journal of Cleaner Production*, 193, 563-574.
- Umamaheswari, E., Ganesan, S., Abirami, M. & Subramanian, S. 2017. Cost Effective Integrated Maintenance Scheduling in Power Systems using Ant Lion Optimizer. *Energy Procedia*, 117, 501-508.
- Utama, D. M. 2017. Analisa perbandingan penggunaan aturan prioritas penjadwalan pada penjadwalan non delay n *job 5 machine*. *Research Report*, 1.
- Utama, D. M. 2018. Pengembangan algoritma NEH Dan CDS Untuk meminimasi consumption energy pada penjadwalan *flow shop* Prosiding SENTRA (Seminar Teknologi dan Rekayasa), 2018 Malang. Malang: Universitas Muhammadiyah Malang.
- Utama, D. M. 2019. An effective hybrid sine cosine algorithm to minimize carbon emission on flow-shop scheduling sequence dependent setup. *2019*, 20, 10.
- Utama, D. M., Widodo, D. S., Wicaksono, W. & Ardiansyah, L. R. 2019. A new hybrid metaheuristics algorithm for minimizing energy consumption in the flow shop scheduling problem. *International Journal of Technology*, 10, 320-331.
- Wu, C.-C., Chen, J.-Y., Lin, W.-C., Lai, K., Liu, S.-C. & Yu, P.-W. 2018. A two-stage three-machine assembly flow shop scheduling with learning consideration to minimize the flowtime by six

- hybrids of particle swarm optimization. *Swarm and Evolutionary Computation*.
- Xu, J., Wang, F., Lv, C. & Xie, H. 2018. Carbon emission reduction and reliable power supply equilibrium based daily scheduling towards hydro-thermal-wind generation system: A perspective from China. *Energy Conversion and Management*, 164, 1-14.
- Yu, C., Semeraro, Q. & Matta, A. 2018. A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Computers & Operations Research*, 100, 211-229.
- Zhang, Y., Liu, Q., Zhou, Y. & Ying, B. 2017. Integrated optimization of cutting parameters and scheduling for reducing carbon emissions. *Journal of Cleaner Production*, 149, 886-895.