OPEN ACCESS
MES: *Journal of Mathematics Education and Science*
ISSN: 2579-6550 (online) 2528-4363 (print)
Vol. 6, No. 2, Mei 2021

Situs web:
https://jurnal.uisu.ac.id/index.php/mesuisu

Email: jurnalmes@fkip.uisu.ac.id

MINIMASI MAKESPAN PADA PERSOALAN PENJADWALAN ORDERED FLOWSHOP MENGGUNAKAN PSO

Rizki Habibi *
Akademi Informatika dan Komputer Medicom, Medan, Sumatera Utara, Indonesia, 20153

Arie Candra Panjaitan
Akademi Informatika dan Komputer Medicom, Medan, Sumatera Utara, Indonesia, 20153

M. Huda Firdaus
Akademi Informatika dan Komputer Medicom, Medan, Sumatera Utara, Indonesia, 20153

Abstrak. Persoalan penjadwalan produksi berbentuk flowshop dengan *n* ragam pekerjaan dan *m* ragam mesin yaitu bagaimana mendapatkan urutan jadwal pengalokasian operasi-operasi dalam pekerjaan ke dalam mesin-mesin yang tersedia sedemikian sehingga mendapatkan total waktu minimal selesainya seluruh pekerjaan atau biasa disebut makespan. Penelitian ini mengusulkan pendekatan teknik optimasi dengan algoritma PSO dalam meminimalkan makespan pada masalah penjadwalan ordered flowshop. Kinerja algoritma penjadwalan yang diusulkan dievaluasi dengan melakukan pengujian pada set data benchmark persoalan penjadwalan ordered flowshop sebanyak 240 variasi kombinasi ukuran job dan mesin. Makespan minimum diperoleh sebagai hasil penjadwalan dengan PSO yang prosesnya berhenti pada n iterasi tertentu ketika dalam 10 iterasi terakhir tidak lagi terdapat perubahan nilai makespan yang lebih baik. Kinerja algoritma PSO efisien pada penjadwalan ordered flowshop dengan penggunaan iterasi yang terbanyak adalah 19 iterasi dan waktu eksekusi terlama yaitu selama 28.42 detik atau kurang dari setengah menit yaitu pada penjadwalan instance dengan ukuran jumlah mesin dan pekerjaan yang terbesar. Pada penelitian ini hanya dilakukan analisis minimal makespan yang dihasilkan dan waktu eksekusi, penelitian selanjutnya disarankan untuk dapat diperluas dengan tidak hanya mengukur makespan minimal, seperti mengukur total flowtime, total tardiness, dan lainnya.

Kata Kunci: Penjadwalan Ordered Flowshop, PSO, Flowshop Permutasi.

Abstract. The production scheduling problem is in the kind of flowshop with n jobs and m machines, to get the order of the schedule for allocating operations of the jobs to the available machines so as to get the minimum total time for completion of all job or commonly called makespan. This study uses an optimization technique approach with the PSO algorithm to get minimum makespan on the ordered flowhop scheduling problem. The performance of the scheduling algorithm presented is evaluated by testing on a benchmark data set of 240 variations in the combination number of jobs and machines. The minimum measure is obtained as a result of scheduling with PSO, whose process stops at a certain iteration when in the last 10 iterations there is no change in the value of a better makespan. The performance of the PSO algorithm is efficient at regular flow scheduling with the use of the most iterations of 19 iterations and the longest execution time of 28.42 seconds or less than half a minute, namely scheduling instances with the largest number of machines and jobs. In this research, only the analysis of the resulting minimal forward and the time of execution was carried out. Further research can be extended by not only measuring the minimum makespan, such as measuring total flowtime, total tardiness, and others.

Keywords: Ordered Flowshop Scheduling, PSO, Permutation Flowshop.

*Corresponding Author: pakhabibi@gmail.com

Sitasi: **Habibi, R., Panjaitan, A.C., & Firdaus, M.H. 2021. Minimasi Makespan Pada Persoalan Penjadwalan Ordered Flowshop Menggunakan PSO. *MES* (*Journal of Mathematics Education and Science*), 6(2): 40-48.**		
Submit: **08 Maret 2021**	**Revisi:** **23 April 2021**	**Publish:** **02 Mei 2021**

PENDAHULUAN

Pada banyak bidang Industri, salah satu masalah yang banyak ditemui dan sangat serbaguna yaitu persoalan penjadwalan produksi flowshop. Menjadwalkan produksi berbentuk flowshop terdiri dari sebanyak n ragam pekerjaan dan m ragam mesin, dengan proses produk diproses mesin dengan tahap urutan yang tetap dan semua tahap terdapat satu jenis mesin(Davendra, Zelinka, Bialic-Davendra, Senkerik, & Jasek, 2013). Ordered flowshop merupakan penjadwalan yang memiliki ciri khas dengan dua syarat berikut: pertama, apabila suatu pekerjaan memiliki waktu proses yang lebih singkat dari pekerjaan lainnya di sebagian mesin, maka kondisi ini berlaku juga untuk setiap mesin lainnya; dan kedua, apabila suatu waktu pemrosesan suatu pekerjaan di mesin lebih singkat dari waktu proses pada mesin lainnya, maka kondisi ini juga berlaku pada semua pekerjaan lainnya (Khatami et al., 2019). Secara umum, persoalan pada penjadwalan ini adalah bagaimana mendapatkan total waktu minimal selesainya seluruh pekerjaan atau biasa disebut makespan(Fernandez-Viagas, 2017), (Allahverdi el al., 2018),(Assia et al., 2020). Keefektifan suatu penjadwalan merupakan hal penting karena memberikan dampak yang serius untuk mengurangi biaya dan meningkatkan produktivitas (Gupta, 2020) yang juga berdampak pada rantai suplay produksi (Habibi, 2017).

Namun, menyelesaikan persoalan penjadwalan ordered flowshop yang bertujuan mendapatkan nilai makespan minimum memerlukan perhitungan yang rumit dan diketahui sebagai NP-hard (Khatami et al., 2019). Di sisi lain, apabila penjadwalan yang efektif dan tepat tidak dilakukan, hal ini bisa mengakibatkan waktu idle di mesin serta mengurangi produktivitas yang akhirnya bisa mengakibatkan harga produk menjadi naik (Hossain, 2014). Berbagai penelitian sebelumnya yang mempelajari persoalan penjadwalan ordered flowshop diantaranya yaitu membahas masalah ordered machines dimana tenggat jatuh tempo pekerjaan selesai menjadi fungsi tujuannya (Panwalkar & Koulamas, 2012), menyelesaikan persoalan penjadwalan order flowshop yang menggunakan masalah pemesanan fleksibel yang kemudian mendapatkan bahwa desain terbaik pada masalah ini adalah dengan menyusun pekerjaan secara berurutan sesuai aturan short processing time (Panwalkar et al., 2013). Selain itu, algoritma sebelumnya diturunkan kompleksitasnya menjadi O(n log n) dan meningkatkan kinerjanya (Ilić, 2015). Persoalan pemilihan pekerjaan pada penjadwalan flowshop juga telah dipelajari (Koulamas & Panwalkar, 2015)dan telah meminimalkan total waktu penyelesaian dan meminimalkan makespan sebagai penjadwalan permutasi pada sebagian kasus khusus (Panwalkar & Koulamas, 2017).

Aplikasi algoritma heuristic telah banyak dipelajari peneliti sebelumnya pada masalah penjadwalan ini, seperti penerapan Ant Colony Optimization (ACO) untuk mendapatkan susunan jadwal yang paling baik, yaitu susunan jadwal yang menghasilkan makespan paling kecil (Widyawati, 2018) dan Particle Swarm Optimization (PSO) yang juga telah diterapkan untuk menyelesaikan masalah susunan jadwal (Muharniet al., 2019). Maka pada makalah ini, diusulkan suatu teknik solusi penjadwalan yang meminimumkan makespan pada persoalan penjadwalan ordered flowshop dengan menggunakan algoritma heuristic berbasis kecerdasan kawanan, yaitu algoritma PSO. PSO juga telah diterapkan dalam penyelesaian persoalan pernjadwalan jalur bergerak yang menghasilkan jadwal yang efisien pada node agen selular untuk persoalan wireless sensor networks (Gao et al., 2019). Set data benchmark terkait

persoalan penjadwalan ordered flowshop memang cukup terbatas, Studi ini menerapkan PSO dalam mencari susunan jadwal dengan makespan minimum pada persoalan penjadwalan ordered flowshop dengan melakukan pengujian menggunakan data benchmark masalah penjadwalan ordered flowshop yang dibuat oleh Khatami et al. (2019).

METODE

1. Masalah Penjadwalan ordered flowshop

Diambil satu set mesin yang berbeda$M = \{1, \dots, m\}$ serta satu set pekerjaan $J = \{1, \dots, n\}$, dengan p_{rj} melambangkan waktu eksekusi pekerjaan $j$di mesin r, dimana$r \in$ M dan $j \in$ J. Persoalan*ordered flowshop*memiliki ciriyaitu adanya dua kondisi yaitu[2]:Apabila terdapat dua pekerjaan *j*, *k*∈*J*, dengan $p_{r,j} < p_{r,k}$, r ∈ M, maka $p_{q,j} \leq p_{q,k}$, ∀*q*∈*M*; dan,Apabila dua mesin *r*, *q*∈*M*, dan $p_{r,k} < p_{q,k}$, *k*∈*J*, maka $p_{r,j} \leq p_{q,j}$, ∀*j*∈*J*. Diasumsikan eksekusi pekerjaan pada setiap mesin sebagai penjadwlan permutasi, dan setiap pekerjaan mulai dikerjakan pada waktu nol dengan tidak mengizinkan *preemption* pekerjaan, yaitu begitueksekusi pekerjaan diproses, pekerjaan tersebut tidak boleh disela oleh pekerjaan lainnya, dansemua mesin hanya bisamengerjakanhanya satu pekerjaan pada satu waktu. Kegiatan penjadwalan ditujukan untuk memperoleh urutan jadwal terbaik dalam menyelesaikan pekerjaan di mesin (permutasi) dimana fungsi tujuannyameminimumkan *makespan* yang disimbolkanoleh C_{max}, yaitu total waktu untuk menyelesaikan hingga pekerjaan terakhir padasuatu penjadwalan.

Set data yang digunakan dalam ujicoba pada penelitian ini yaitu set data *benchmark* untuk persoalan penjadwalan *ordered flowshop* yang disusun olehKhatami et al. (2019), yaituset data *benchmark Vallada Small*(selanjutnya disebut Vallada Smallinstances) yang disusun berdasarkan set data benchmark *permutation flowshop 240* ***S****mall instances* yang sebelumnya disusun oleh Vallada et al. (2017).Data Vallada *Small Benchmark* terdiri dari 240 data berukuran kecil, dengan ukuran pekerjaan dipilih dari himpunan {10, 20, 30, 40, 50, 60}, dan ukuran mesin dipilih dari himpunan himpunan {5, 10, 15, 20}. Total data ada sebanyak 6 × 4 = 24 kombinasi n dan m. Setiap kombinasi terdiri dari 10 data, sehingga terdapat 240 data. Untuk menjadwalkan dari sebanyak m mesin dan n pekerjaan, data waktu proses disajikan dalam bentuk matriks n x m.Pekerjaan ke-*j* ditunjukkan oleh baris ke-*j*, mesin ke-*r* ditunjukkan oleh kolom ke-*r*, dan $p_{r,j}$ menyatakan isi sel yang merupakan waktu proses penyelesaian pekerjaan *j*pada mesin *r*. Adapun, data bisa diperoleh di url:https://data.mendeley.com/datasets/cd2rv7hyyj/1.

Model Wilson untuk masalah penjadwalan *flowshop* dapat disajikan dengan formulasi berikut ini(Khatami et al., 2019):

$$z = \min C_{max} = \min\left(s_{m,n} + \sum_{j=1}^{n} p_{m,j} z_{j,n}\right) \quad (1)$$

$$\sum_{j=1}^{n} z_{j,i} = 1, \qquad 1 \leq i \leq n, \quad (2)$$

$$\sum_{j=1}^{n} z_{j,i} = 1, \qquad 1 \leq j \leq n, \quad (3)$$

$$s_{1,1} = 0 \quad (4)$$

$$s_{1,i} + \sum_{j=1}^{n} p_{1,j} z_{j,i} = s_{1,i+1} \, 1 \leq i \leq n-1, \quad (5)$$

$$s_{r,1} + \sum_{j=1}^{n} p_{r,j} z_{j,1} = s_{r+1,1} 1 \le r \le m-1, \quad (6)$$

$$s_{r,i} + \sum_{j=1}^{n} p_{r,j} z_{j,i} = s_{r+1,i} 1 \le r \le m-1, \qquad 2 \le i \le n, \quad (7)$$

$$s_{r,i} + \sum_{j=1}^{n} p_{r,j} z_{j,i} = s_{r,i+1} 2 \le r \le m, \qquad 1 \le i \le n-1, \quad (8)$$

$$z_{j,i} \in \{0,1\}, \quad 1 \le j \le n, \quad 1 \le i \le n, \quad (9)$$

$$s_{r,i} \ge 0, \quad 1 \le r \le m, \quad 1 \le i \le n, \quad (10)$$

Fungsi tujuan (1) meminimalkan total waktu penyelesaian seluruh pekerjaan hingga pekerjaan di urutan terakhir pada mesin terakhir, yaitu *makespan*. Kendala (2) dan (3) merupakanpengontrol penugasan, dan memastikan bahwa tepat satu pekerjaan ditugaskan untuk setiap tahapan atau mesin, dan tepat satu mesin ditugaskan untuk setiap pekerjaan. Batasan (4) mengatur waktu mulai penjadwalan yaitu dimulai dari nol. Batasan (5) dan (6) memastikan bahwa tidak ada waktu *idle* pada mesin pertama, dan tidak ada penundaan pengerjaan pekerjaan pertama dalam urutan diproses pada setiap mesin *m*. Batasan (7) menunjukkan bahwa waktu mulai setiap pekerjaan pada mesin $i+1$ tidak lebih awal dari waktu penyelesaiannya pada mesin r. Batasan (8) memastikan bahwa pekerjaan di posisi $i+1$ tidak dimulai pada mesin r sampai pekerjaan di posisi i telah menyelesaikan prosesnya pada mesin itu. Kendala (9) dan (10) menyatakan bahwa domain variabel keputusan adalah biner dan *non-negatif*.

2.Pencarian Solusi Susunan Jadwal menggunakan algoritma PSO

Didefenisikan bahwa ruang pencarian PSO adalah berdimensi-p dan kemudian partikel ke-i kawanannya adalah $x_i = (x_{i1}, x_{i2}, \ldots, x_{is})$, kecepatan partikelnya adalah $v_i = (v_{i1}, v_{i2}, \ldots, v_{is})$, dimana x_{ij}, $v_{ij} \in$ [a,b], $j = (1,2, \ldots, s)$. Partikel terbaik lokal dilambangkan sebagai $pbest_i = (pbest_{i1}, pbest_{i2}, \ldots, pbest_{is})$, dan partikel terbaik global dilambangkan sebagai $gbest_i = (gbest_{i1}, gbest_{i2}, \ldots, gbest_{is})$. Diagram alir dari PSO dapat dilihat pada Gambar 1. Ketika individu (partikel) telah mendapatkan dua nilai terbaik diatas, maka pembaruan kecepatan dan posisinya dilakukan dengan menggunakan persamaan (Tuegehet al., 2009):

$$v_{ij}^{k+1} = \omega_k * v_{ij}^k + c_1 * rand * \left(pbest_{ij}^k - x_{ij}^k\right) + c_2 * rand * \left(gbest_{ij}^k - x_{ij}^k\right) \quad (11)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (12)$$

dimana:

v_{ij}^{k+1}	: kecepatan individu atau partikel
x_{ij}^k	: posisi individu (partikel) pada suatu iterasi (solusi)
$pbest_{ij}^k$	: nilai terbaik milik individu di sekitarnya
$gbest_{ij}^k$	: nilai terbaik secara keseluruhan
$rand( )$	: adalah nomor acak antara (0,1)
ω_k	: bobot inertia
c_1	: bobot kognitif
c_2	: bobot sosial

ω_k, c_1, dan c_2 adalah faktor yang mempengaruhi kecepatan perpindahan partikel, biasanya $c_1 = c_2 = 2$

Gambar 1. Diagram Alir PSO

HASIL DAN PEMBAHASAN

Pengujian ini bertujuan untukmendapatkan makespan terbaik, yaitu makespan minimum yang dapat diperoleh dari suatu susunan jadwal, dimana susunan jadwal tersebut dibangun dengan menggunakan algoritma PSO. Proses Penyusunan jadwal dilakukan dalam 3 tahap; pertama, inisialisasi parameter PSO dan membangun susunan awal jadwal secara random; kemudian kedua, dilakukan penyusunan jadwal menggunakan algoritma PSO untuk mendapatkan makespan minimal pada jumlah iterasi tertentu. Untuk menjalankan algoritma PSO pada penelitian ini, dilakukan penentuan parameter yang ditentukan di awal, yaitu:

Jumlah partikel : 50 partikel
bobot inertia (ω_k) : 2
bobot kognitif (c_1) : 2
bobot sosial (c_2) : 2

Parameter-parameter PSO, yaitu bobot inertia(ω_k)digunakan sebagai parameter yang mengontrol pengaruhdari kecepatan partikel sebelumnya. Jika nilai inertia terlalu besar, kecepatan akan terus meningkat sehingga partikel akan divergen. Jarakpartikel terhadap nilai optimumnya akan terusmeningkat tiap iterasi.Parameter lain dalam algoritma PSO adalah parameter bobot kognitif (c_1) dan parameter bobot sosial (c_2), yaitu konstanta akselerasi yang mempunyai pengaruh terhadap kecepatankonvergensi (Rachman et al., 2012).Nilai ω_k, c_1, dan c_2 yang digunakan pada pengujian penelitian ini didasarkan pada parameter yang biasa digunakan dalam penerapan PSO, seperti padaMansur (2014)yang menggunakan nilai

yang sama yaitu 2. Jumlah partikel yang digunakan pada pengujian penelitian ini adalah sebanyak 50 partikel.

Pada tahap awal, dilakukan penyusunan jadwal secara random terlebih dahulu. Susunan awal jadwal dilakukan dengan membangun bilangan random antara 0 dan 1 untuk setiap job pada masing-masing mesin dan kemudian mengurutkan nilai random sebagai urutan jadwal pada setiap mesin. Susunan jadwal dibangkitkan sebanyak 50 partikel awal yang merepresentasikan jadwal urutan job pada setiap mesin. Setiap partikel mewakili satu susunan jadwal. Kemudian, setiap partikel dievaluasi dengan menghitung makespan dari susunan jadwal awal tersebut. Selanjutnya posisi partikel diupdate untuk iterasi berikutnya dengan menggunakan algoritma PSO sampai tercapai kriteria berhenti, yaitu apabila dalam 10 iterasi berturut-turut tidak diperoleh nilai makespan yang lebih kecil.

Untuk mengukur kinerja pendekatan PSO dalam menemukan susunan jadwal yang menghasilkan makespan minimum, scenario pengujian menggunakan data benchmark sebanyak 240 jadwal dengan variasi ukuran dimensi. Hasil minimasi makespan pada penjadwalan sebagian set data (1 instance mewakili setiap variasi ukuran dimensi) menggunakan algoritma PSO disajikan pada Tabel 1.

Tabel 1. Hasil Minimasi Makespan

Nama Data	Jumlah Mesin	Jumlah Job	Makespan	Iterasi	Waktu (s)
S_10_5_1	10	5	565	11	0.26
S_10_10_1	10	10	1123	14	0.69
S_10_15_1	10	15	567	12	0.29
S_10_20_1	10	20	1913	13	1.48
S_20_5_1	20	5	1273	15	0.57
S_20_10_1	20	10	1726	15	1.60
S_20_15_1	20	15	3035	13	4.36
S_20_20_1	20	20	2935	15	3.88
S_30_5_1	30	5	1896	15	1.12
S_30_10_1	30	10	2378	16	2.78
S_30_15_1	30	15	3220	10	3.53
S_30_20_1	30	20	3567	13	6.70
S_40_5_1	40	5	2579	17	1.91
S_40_10_1	40	10	3264	16	4.37
S_40_15_1	40	15	3849	17	8.74
S_40_20_1	40	20	4615	16	13.18
S_50_5_1	50	5	3241	19	2.91
S_50_10_1	50	10	3724	14	6.26
S_50_15_1	50	15	4620	13	10.35
S_50_20_1	50	20	5133	14	17.83
S_60_5_1	60	5	3587	18	3.65
S_60_10_1	60	10	4415	16	7.99
S_60_15_1	60	15	5349	10	10.80
S_60_20_1	60	20	6096	16	28.42

Pada Tabel 1 dapat dilihat 24 instances yang mewaliki setiap ukuran variasi dimensi penjadwalan yaitu variasi banyak mesin dan banyak pekerjaan. Masing-masing ukuran variasi dimensi penjadwalan mewakili 10 instances yang memiliki ukuran dimensi yang

sama. Nama Data menunjukkan nama instances yang dijadwalkan, sesuai dengan nama yang ada pada data benchmark. Makespan yang diperoleh merupakan makespan minimum hasil penjadwalan dengan PSO dengan jumlah iterasi yang digunakan dan waktu proses yang digunakan. Nilai makespan bergantung pada waktu proses yang ada pada masing-masing instance yang dijadwalkan, yang mana PSO mencari susunan jadwal terbaik yang menghasilkan makespan minimum. Iterasi yang digunakan menunjukkan proses PSO berhenti pada iterasi tertentu ketika dalam 10 iterasi terakhir tidak lagi terdapat perubahan nilai makespan yang lebih baik. Iterasi yang terbanyak ada pada penjadwalan 50 mesin dan 5 pekerjaan, yaitu sebanyak 19 iterasi dengan makespan minimum sebesar 3241.

Namun waktu eksekusinya tergolong cepat, yaitu hanya 2.91 detik. Sedangkan iterasi paling sedikit ada pada penjadwalan 30 mesin dan 15 pekerjaan serta pada penjadwalan 60 mesin dan 15 pekerjaan dengan waktu proses masing-masing 3.53 detik dan 10.80 detik. Waktu eksekusi tercepat ada pada penjadwalan 10 mesin dan 5 pekerjaan yang diselesaikan dengan 11 iterasi dengan waktu proses selama 0.26 detik atau kurang dari setengah menit. Dan ini merupakan ukuran jumlah mesin dan pekerjaan yang paling sedikit. Waktu eksekusi terlama ada pada penjadwalan 60 mesin dan 20 pekerjaan yang diselesaikan dengan 16 iterasi yaitu selama 28.42 detik atau kurang dari setengah menit. Dan ini merupakan ukuran jumlah mesin dan pekerjaan yang paling banyak/besar.Waktu operasi penjadwalan algoritma PSO pada set data benchmark, ditunjukkan pada grafik perbandingan waktu operasi dari variasi ukuran jadwal pada Grafik 1.

Grafik 1. Waktu Operasi Penjadwalan algoritma PSO

Pada Grafik 1, dapat dilihat bahwa waktu proses eksekusi penjadwalan menggunakan algoritma PSO, meningkat seiring bertambahnya ukuran dimensi (banyak pekerjaan dan banyak mesin) suatu instance yang dijadwalkan. Dapat dilihat juga bahwa waktu proses eksekusi penjadwalan menggunakan algoritma PSO lebih dipengaruhi oleh banyak pekerjaan yang dijadwalkan, dimana terlihat waktu proses eksekusi penjadwalan semakin meningkat ketika meningkatnya ukuran banyak pekerjaan. Ketika ukuran banyak pekerjaan berkurang, walaupun dengan jumlah mesin yang meningkat, waktu proses eksekusi menjadi lebih cepat.

KESIMPULAN

Penelitian ini mengkontribusikan pengembangan teknik optimasi dalam meminimalkan *makespan* pada masalah penjadwalan *ordered flowshop*dengan algoritmaPSO. Kinerja algoritma penjadwalan yang diusulkan dievaluasi dengan melakukan pengujian pada

set data persoalan penjadwalan *ordered flowshop* sebanyak 240 variasi kombinasi ukuran job dan mesin.Makespan minimum diperoleh sebagai hasil penjadwalan dengan PSO yang prosesnya berhenti pada iterasi tertentu ketika dalam 10 iterasi terakhir tidak lagi terdapat perubahan nilai makespan yang lebih baik.Penjadwalan ordered flowshop dengan algoritma PSO cukup efisien dengan iterasi yang terbanyak yaitu 19 iterasi waktu eksekusi terlama yaitu selama 28.42 detik atau kurang dari setengah menit yaitu pada penjadwalam dengan ukuran jumlah mesin dan pekerjaan yang paling besar.Pada penelitian ini hanya dilakukan analisis minimal makespan yang dihasilkan dan waktu eksekusi, penelitian selanjutnya disarankan untuk dapat diperluas dengan tidak hanya mengukur makespan minimal, seperti mengukur total flowtime, total tardiness, dan lainnya.

DAFTAR PUSTAKA

Allahverdi, A., Aydilek, H. & Aydilek, A., (2018). No-Wait Flowshop Scheduling Problem With Two Criteria; Total Tardiness And Makespan.*European Journal of Operational Research*, Vol. 269(2), pp. 590–601.

Assia, S., El-Abbassi, I., El-Barkany, A., Darcherif, M., & El-Biyaali, A., (2020). Green Scheduling of Jobs and Flexible Periods of Maintenance in a Two-Machine Flowshop to Minimize Makespan, a Measure of Service Level and Total Energy Consumption. *Advances in Operations Research*, pp. 1–9.

Davendra, D., Zelinka, I., Bialic-Davendra, M., Senkerik, R., & Jasek, R., (2013). Discrete Self-Organising Migrating Algorithm For Flow-Shop Scheduling With No-Wait Makespan. *Mathematical and Computer Modelling*, Vol. 57, pp. 100–110.

Fernandez-Viagas, V., Ruiz, R. & Framinan, J. M., (2017). A new vision of approximate methods for the permutation flowshop to minimise makespan: State-of-the-art and computational evaluation.*European Journal of Operational Research*, Vol. 257(3), pp. 707–721.

Gao, Y.,Wang, J., Wu, W., Sangaiah, A.K., & Lim, S., (2019). A Hybrid Method For Mobile Agent Moving Trajectory Scheduling Using ACO and PSO in WSNs.*Sensors (Switzerland)*, Vol. 19(3), pp. 1-19.

Gupta, J. N. D., Majumder, A. & Laha, D., (2020). Flowshop Scheduling With Artificial Neural Networks.*Journal of the Operational Research Society*, Vol. 71(10), pp. 1619–1637.

Habibi, R. (2017). Teknik Linierisasi Untuk Menyelesaikan Persoalan Rantai Suplai Lokasi-Inventori. *Jurnal As-Salam*, Vol. 1(1), pp. 44–55.

Hossain, M. S. , Asadujjaman, M. &Bhattacharya, P., (2014). Minimization of Makespan in Flow Shop Scheduling Using Heuristics. *ICMIEE*.

Ilić, A. (2015). On The Variable Common Due Date, Minimal Tardy Jobs Bicriteria Two-Machine Flow Shop Problem With Ordered Machines.*Theoretical Computer Science*, Vol, 582, pp. 70–73.

Khatami, M., Salehipour, A. & Hwang, F. J. (2019). Makespan Minimization For The M-Machine Ordered Flow Shop Scheduling Problem. *Computers and Operations Research*, Vol. 111, pp. 400–414.

Koulamas, C. & Panwalkar, S. S., (2015). Job Selection In Two-Stage Shops With Ordered Machines. *Computers and Industrial Engineering*, Vol. 88, pp. 350–353.

Mansur, (2014). Perancangan Sistem Informasi Penjadwalan Resource Perguruan Tinggi Menggunakan Metode Particle Swarm Optimization (PSO). *Inovtek*, Vol. 4, No. 2, pp. 75 – 86.

Muharni, Y., Febianti, E. & Sofa, N. N., (2019). Minimasi Makespan Pada Penjadwalan Flow Shop Mesin Paralel Produk Steel Bridge B-60 Menggunakan Metode Longest Processing Time Dan Particle Swarm Optimization. *Journal Industrial Servicess*, Vol. 4(2).

Panwalkar, S. S. & Koulamas, C., (2012). An O (N2) Algorithm For The Variable Common Due Date, Minimal Tardy Jobs Bicriteria Two-Machine Flow Shop Problem With Ordered Machines. *European Journal of Operational Research*, Vol. 221, Issue 1, pp. 7-13.

Panwalkar, S. S. & Koulamas, C. (2017). On The Dominance Of Permutation Schedules For Some Ordered And Proportionate Flow Shop Problems. *Computers and Industrial Engineering*, Vol. 107, pp. 105–108.

Panwalkar, S. S., Smith, M. L. &Koulamas, C., (2013). Review Of The Ordered And Proportionate Flow Shop Scheduling Research. *Naval Research Logistics (NRL)*.

Rachman, A.R., Syarif, D., dan Sari, P.R., (2012). Analisa dan Penerapan MetodeParticle Swarm Optimization Pada Optimasi Penjadwalan Kuliah.*Jurnal Teknik Informatika*, Vol 1, No. 2.

Vallada, E., Ruiz, R., Framinan, J.M., (2015). New Hard Benchmark For Flowshop Scheduling Problems Minimising Makespan. *European Journal of Operational Research*, Vol. 240, No. 3, pp. 666–677.

Widyawati, W. (2018). Penerapan Algoritma Ant Colony Optimization (ACO) Pada Job Shop Scheduling Problem (JSSP) di PT. Siemens Indonesia (Cilegon Factory). *Jurnal Sistem Informasi Dan Informatika (SIMIKA),*Vol. 1(01), pp. 35-51.

Tuegeh, M., Soeprijanto & Purnomo, M. H., (2009). Modified Improved Particle Swarm Optimization For Optimal.*Seminar Nasional Aplikasi Teknologi Informassi 2009* (SNATI 2009), pp. 85–90.