



**PENERAPAN ALGORITMA *GLOWWORM SWARM OPTIMIZATION* (GSO)
PADA *HYBRID FLOWSHOP SCHEDULING PROBLEM***

SKRIPSI

Oleh:

**Chadli Hakim
NIM 121810101039**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016**



**PENERAPAN ALGORITMA *GLOWWORM SWARM OPTIMIZATION* (GSO)
PADA *HYBRID FLOWSHOP SCHEDULING PROBLEM***

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar sarjana Sains

Oleh:

Chadli Hakim
NIM 121810101039

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Abi Hilal dan Mama Siti Cholisoah yang telah membesarkan dan mendidik saya, menjadi motivasi terbesar saya serta telah memberikan pengorbanan dan kasih sayangnya.
2. Adik-adikku Ahmad Baihaqi, Mahir Habib dan Hasby Ash Shiddiqy yang selalu memberikan rasa rindu tersendiri bagi saya untuk segera menyelesaikan skripsi ini.
3. Keluarga besar Bani Erfan dan Bani Basymel yang memberikan dukungan tersendiri untuk saya.

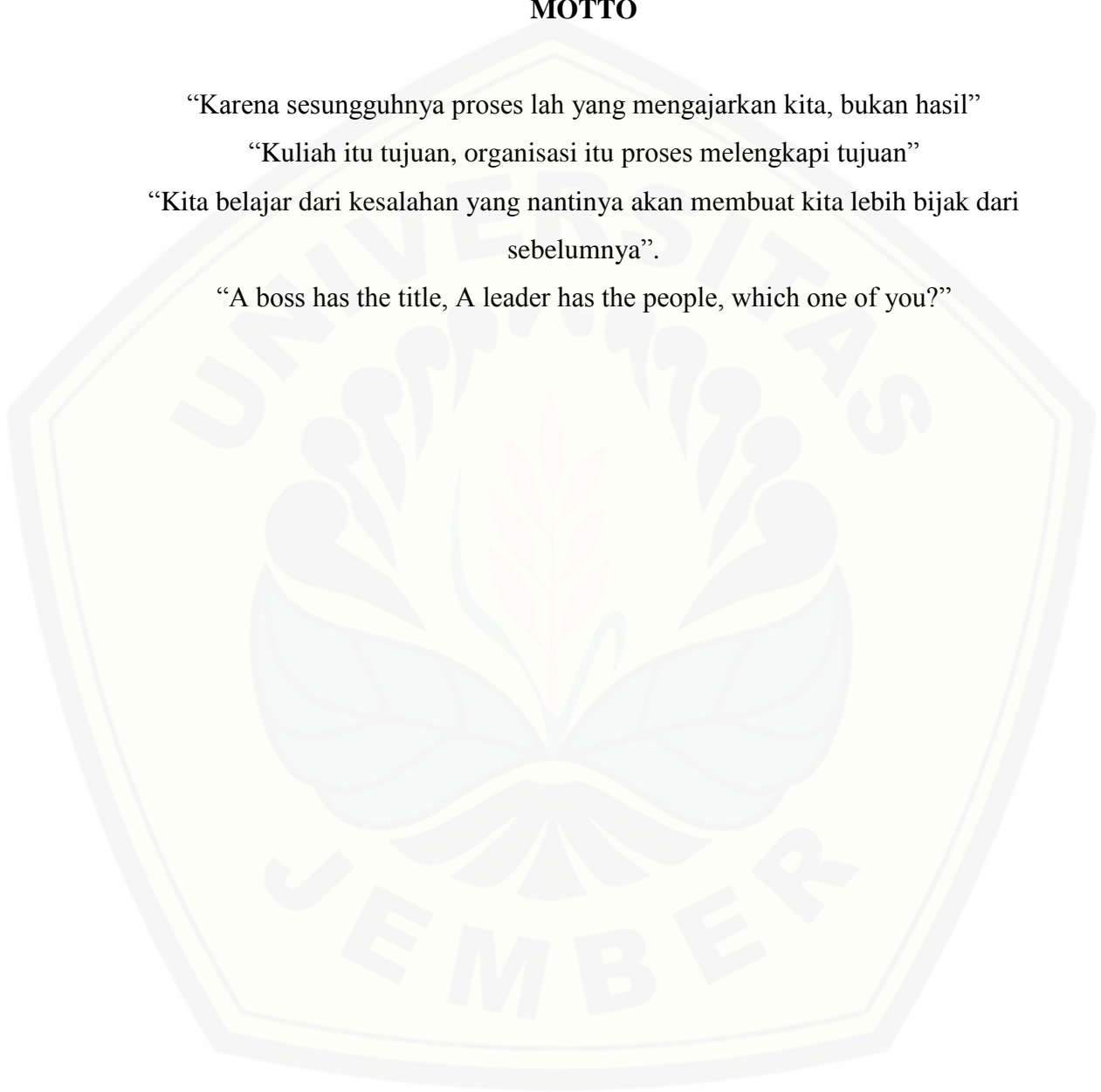
MOTTO

“Karena sesungguhnya proses lah yang mengajarkan kita, bukan hasil”

“Kuliah itu tujuan, organisasi itu proses melengkapi tujuan”

“Kita belajar dari kesalahan yang nantinya akan membuat kita lebih bijak dari sebelumnya”.

“A boss has the title, A leader has the people, which one of you?”



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Chadli Hakim

NIM : 121810101039

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan Algoritma *Glowworm Swarm Optimization* (GSO) pada *Hybrid Flowshop Scheduling Problem*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Desember 2016

Yang menyatakan,

Chadli Hakim

NIM 121810101039

SKRIPSI

**PENERAPAN ALGORITMA *GLOWWORM SWARM OPTIMIZATION* (GSO)
PADA *HYBRID FLOWSHOP SCHEDULING PROBLEM***

Oleh

Chadli Hakim

NIM 121810101039

Pembimbing:

Dosen Pembimbing Utama : Kusbudiono, S.Si., M.Si.

Dosen Pembimbing Anggota : Ahmad Kamsyakawuni, S.Si., M.Kom.

PENGESAHAN

Skripsi berjudul “Penerapan Algoritma *Glowworm Swarm Optimization* (GSO) pada *Hybrid Flowshop Scheduling Problem*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji:

Ketua,

Sekretaris,

Kusbudiono, S.Si., M.Si.

Ahmad Kamsayakawuni, S.Si., M.Kom.

NIP. 197704302005011001

NIP. 197211291998021001

Dosen Penguji I,

Dosen Penguji II,

Kosala Dwidja Purnomo, S.Si., M.Si.

M. Ziaul Arif, S.Si., M.Sc.

NIP. 196908281998021001

NIP. 198501112008121002

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Sujito, Ph. D.

NIP. 196102041987111001

RINGKASAN

Penerapan Algoritma *Glowworm Swarm Optimization* (GSO) pada *Hybrid Flowshop Scheduling Problem*; Chadli Hakim, 121810101039; 2016; 54 halaman; Jurusan Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Pada dunia industri, suatu perusahaan selalu mempunyai banyak masalah. Salah satunya adalah terbatasnya jumlah mesin yang digunakan untuk menyelesaikan banyaknya pekerjaan. Untuk itu perlu dilakukan suatu proses penjadwalan, dimana pada kasus ini disebut penjadwalan *Flowshop*. Masalah penjadwalan *Flowshop* adalah suatu masalah penjadwalan dimana setiap *job* memiliki urutan proses yang sama pada setiap mesin. Penjadwalan *Flowshop* ini bertujuan untuk mencari urutan dari setiap *job* pada mesin yang mengoptimalkan beberapa kriteria, salah satunya meminimumkan *makespan*. *Hybrid Flowshop Scheduling* (HFS) merupakan pengembangan dari penjadwalan *flowshop* yang mempunyai mesin paralel di setiap tahapnya.

Pada penelitian ini akan digunakan data sekunder. Data tersebut merupakan data *Hybrid Flowshop* dengan *multiprocessor tasks*. Data tersebut merupakan data simulasi yang berisi banyaknya *job*, banyaknya *stage* dan waktu yang dibutuhkan untuk menyelesaikan setiap *job*. Penjadwalan *Hybrid Flowshop* pada data tersebut menggunakan algoritma *Glowworm Swarm Optimization* (GSO).

Nilai parameter algoritma GSO sangat berpengaruh terhadap hasil, kecepatan kekonvergenan dan waktu komputasi. Nilai parameter *step-size* $s = 0,05$ mampu menghasilkan *makespan* paling optimal, namun konvergen lebih lama. Nilai parameter jarak antar *Glowworm* r_s yang semakin besar maka waktu komputasi semakin lama. Nilai $r_s = 10$ adalah yang paling optimum. Nilai parameter *dynamic*

decision-range β yang semakin besar, waktu komputasi menjadi lebih cepat. Nilai $\beta = 0,01$ adalah nilai paling optimum. Nilai parameter batas ketetanggan n_t memiliki pengaruh yang sama dengan r_s , namun nilai paling optimumnya adalah 8.

Algoritma *Glowworm Swarm Optimization* dapat digunakan untuk menyelesaikan permasalahan *Hybrid Flowshop Scheduling*. Program penerapan algoritma *Glowworm Swarm Optimization* yang telah dibuat dengan MATLAB membutuhkan waktu komputasi rata-rata 51,5108 detik untuk 10 *job*, 70,374 detik untuk 20 *job* dan 200,5883 detik untuk 50 *job* dengan maksimum iterasi $T_{\max} = 1000$. Algoritma GSO ini konvergen kurang dari 150 iterasi.

PRAKATA

Puji syukur kehadiran Tuhan Yang Maha Esa atas segala kuasa-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Penerapan Algoritma *Glowworm Swarm Optimization* (GSO) pada *Hybrid Flowshop Scheduling Problem*”. Penulisan tugas akhir ini dilakukan guna memenuhi salah satu syarat untuk menyelesaikan Program Studi Matematika (S1) dan mencapai gelar sarjana Sains pada Jurusan Matematika FMIPA Universitas Jember.

Pada kesempatan ini, dengan segala hormat penulis mengucapkan terima kasih kepada:

1. Bapak Kusbudiono, S.Si., M.Si. selaku dosen pembimbing utama dan Bapak Ahmad Kamsyakawuni, S.Si., M.Kom.
2. Keluarga besar PALAPA (Pecinta Alam Mahasiswa MIPA).
3. Keluarga kost NIAS 18 dan segenap rekan-rekan dari IKMBJ (Ikatan Keluarga Mahasiswa Bojonegoro di Jember).
4. Keluarga besar BATHICS'12 jurusan Matematika.
5. Almamater kebanggaan Universitas Jember.

Penulis menyadari bahwa penulisan ini masih jauh dari sempurna. Oleh karena itu, penulis mengharap kritik dan saran demi kesempurnaan penelitian selanjutnya. Semoga tugas akhir ini dapat bermanfaat bagi kita semua.

Jember Desember 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN.....	iv
HALAMAN PEMBIMBINGAN.....	v
SKRIPSI.....	v
PENGESAHAN.....	vi
RINGKASAN	vii
PRAKATA.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xiv
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 <i>Flowshop Scheduling</i>	4
2.2 <i>Hybrid Flowshop Scheduling</i>	4
2.3 <i>Diagram Gantt</i>	6
2.4 <i>Glowworm Swarm Optimization</i>	7

BAB 3. METODE PENELITIAN	10
3.1 Data Penelitian	10
3.2 Langkah-langkah Penelitian	10
BAB 4. HASIL DAN PEMBAHASAN	12
4.1 Langkah Perhitungan	12
4.2 Hasil Penelitian.....	19
4.2.1 Program	19
4.2.2 Hasil Percobaan.....	24
4.3 Pembahasan	28
BAB 5. PENUTUP.....	31
5.1 Kesimpulan	31
5.2 Saran	31
DAFTAR PUSTAKA	32
LAMPIRAN.....	34

DAFTAR TABEL

	Halaman
2.1 Contoh data <i>Hybrid Flowshop</i>	6
4.1 Contoh data 2 <i>Stage 6 Job</i> , (a) <i>Data Stage</i> , (b) <i>Data Job</i>	12
4.2 Pengaruh parameter <i>step size s</i>	25
4.3 Pengaruh parameter jarak antar <i>Glowworm</i> r_s	25
4.4 Pengaruh parameter tingkat update <i>dynamic decision-range</i> β	26
4.5 Pengaruh parameter batas ketetangaan n_t	27

DAFTAR GAMBAR

	Halaman
2.1 Contoh diagram <i>Gantt</i>	7
2.2 <i>Flowchart</i> algoritma <i>Glowworm Swarm Optimization</i>	9
3.1 Skema langkah-langkah penelitian	11
4.1 Diagram <i>gantt</i> untuk <i>Glowworm</i> 1	13
4.2 Diagram <i>gantt</i> untuk <i>Glowworm</i> 2	14
4.3 Diagram <i>gantt</i> untuk <i>Glowworm</i> 3	14
4.4 Tampilan program	19

DAFTAR LAMPIRAN

	Halaman
A.1 Data HFS 2 <i>Stage</i>	34
A.2 Data HFS 5 <i>Stage</i>	35
A.3 Data HFS 8 <i>Stage</i>	38
B.1 Hasil Percobaan Data 2 <i>Stage</i> 10 <i>Job</i>	48
B.2 Hasil Percobaan Data 2 <i>Stage</i> 20 <i>Job</i>	48
B.3 Hasil Percobaan Data 2 <i>Stage</i> 50 <i>Job</i>	49
B.4 Hasil Percobaan Data 5 <i>Stage</i> 10 <i>Job</i>	50
B.5 Hasil Percobaan Data 5 <i>Stage</i> 20 <i>Job</i>	50
B.6 Hasil Percobaan Data 5 <i>Stage</i> 50 <i>Job</i>	51
B.7 Hasil Percobaan Data 8 <i>Stage</i> 20 <i>Job</i>	52
B.8 Hasil Percobaan Data 8 <i>Stage</i> 50 <i>Job</i>	53

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pada dunia industri, suatu perusahaan selalu mempunyai banyak masalah. Salah satunya adalah terbatasnya jumlah mesin yang digunakan untuk menyelesaikan banyaknya pekerjaan. Dalam hal ini, perusahaan harus menentukan pekerjaan mana yang harus diselesaikan terlebih dahulu agar waktu proses seluruh pekerjaan dapat diminimumkan. Untuk itu perlu dilakukan suatu proses penjadwalan, dimana pada kasus ini disebut penjadwalan *Flowshop*.

Masalah penjadwalan *Flowshop* adalah suatu masalah penjadwalan dimana setiap *job* memiliki urutan proses yang sama pada setiap mesin. Penjadwalan *Flowshop* ini bertujuan untuk mencari urutan dari setiap *job* pada mesin yang mengoptimalkan beberapa kriteria, salah satunya meminimumkan *makespan*. Namun pada beberapa perusahaan terdapat beberapa tahap yang memiliki sejumlah mesin yang sama (paralel). Sehingga masalah ini dikategorikan kedalam masalah penjadwalan *Hybrid Flowshop* (HFS).

Penelitian-penelitian terbaru tentang penjadwalan *Hybrid Flowshop* diantaranya adalah Pratiwi *et al.* (2014) yang bertujuan mengoptimalkan mesin sehingga tidak terjadi *idle time* mesin menggunakan *Integer Linear Programming*, Putawama *et al.* (2014) menerapkan algoritma *Modified PSO*, Santosa dan Rofiq (2014) memodifikasi algoritma SA dengan tujuan meminimumkan *makespan* dan total *tardiness* pada HFS, Wang *et al.* (2014) menerapkan algoritma *Branch-and-Bound*. Berdasarkan semua penelitian tersebut, algoritma yang diterapkan dapat digunakan untuk menyelesaikan masalah HFS.

Selain metode yang telah diterapkan dalam penelitian tersebut, terdapat banyak sekali algoritma untuk menyelesaikan masalah optimasi, seperti *Swarm*

Intelligence. *Swarm Intelligence* merupakan metode yang berfokus pada perilaku kolektif yang dihasilkan dari interaksi lokal individu satu sama lain dengan lingkungannya. Salah satu bagian dari *Swarm Intelligence* adalah algoritma *Glowworm Swarm Optimization*.

Algoritma *Glowworm Swarm Optimization* merupakan salah satu metode baru yang didasarkan pada perilaku cacing bercahaya. Algoritma tersebut telah diterapkan pada kasus *Vehicle Routing Problem with Stochastic Demands* oleh Marinaki dan Marinakis (2016) dan pada kasus *Knapsack 0-1 Problem* oleh Gong *et al.* (2011). Kedua penelitian tersebut menghasilkan kesimpulan bahwa algoritma *Glowworm Swarm Optimization* sangat efektif dalam penyelesaian masalah optimasi.

Berdasarkan latar belakang di atas, peneliti ingin menerapkan algoritma *Glowworm Swarm Optimization* pada permasalahan penjadwalan *Hybrid Flowshop* yang diharapkan dapat menghasilkan solusi optimal dengan tingkat konvergensi yang cepat.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas dalam penelitian ini adalah sebagai berikut:

- a. Bagaimana penerapan algoritma *Glowworm Swarm Optimization* pada penjadwalan *Hybrid Flowshop*?
- b. Bagaimana hasil, tingkat konvergensi dan waktu komputasi algoritma *Glowworm Swarm Optimization* pada penjadwalan *Hybrid Flowshop*?

1.3 Batasan Masalah

Batasan-batasan masalah adalah sebagai berikut:

- a. Bahan baku yang dibutuhkan selalu tersedia.
- b. Setiap *job* memiliki *ready time* yang sama.
- c. Mesin berjalan dengan normal.
- d. Waktu membawa *job* ke *stage* selanjutnya diabaikan.

1.4 Tujuan Penelitian

Tujuan dari penulisan penelitian ini adalah sebagai berikut:

- a. Menerapkan algoritma *Glowworm Swarm Optimization* pada penjadwalan *Hybrid Flowshop*.
- b. Mengetahui hasil, tingkat konvergensi dan waktu komputasi algoritma *Glowworm Swarm Optimization* pada penjadwalan *Hybrid Flowshop*.

1.5 Manfaat Penelitian

Adapun manfaat yang diperoleh dari penulisan penelitian ini adalah sebagai berikut

- a. Mendapatkan solusi yang optimal yaitu nilai makespan minimum.
- b. Dapat dijadikan referensi dalam penelitian selanjutnya yang berhubungan tentang algoritma *Glowworm Swarm Optimization* dan penjadwalan *Hybrid Flowshop*.

BAB 2. TINJAUAN PUSTAKA

2.1 *Flowshop Scheduling*

Flowshop Scheduling Problem mempunyai *job* sebanyak n yaitu $\{J_1, \dots, J_n\}$ yang harus diproses pada m mesin yang berbeda $\{M_1, M_2, \dots, M_m\}$. Setiap *job* dibagi menjadi m operasi dan setiap operasi membutuhkan hasil yang berbeda. O_{ij} melambangkan operasi pada mesin i dari *job* j . Setiap operasi memerlukan waktu proses p_{ij} . Dalam *Flowshop*, semua *job* memiliki urutan proses yang sama pada setiap mesin. Tujuan *Flowshop* adalah untuk mencari urutan dari setiap *job* pada mesin yang mengoptimalkan beberapa kriteria. Sejauh ini, kriteria yang paling umum adalah meminimumkan total waktu untuk menyelesaikan semua *job*, yang sering disebut sebagai *makespan* (C_{max}) (Baker, 1974).

Menurut Baker, terdapat beberapa kondisi pada masalah *Flowshop*, sebagai berikut:

- a. Semua operasi bebas dan sudah tersedia.
- b. Semua m mesin selalu siap dan tidak ada gangguan.
- c. Setiap mesin i hanya dapat memroses satu *job* pada satu waktu.
- d. Setiap *job* j diproses hanya pada satu mesin dan satu waktu.
- e. Setiap proses melalui operasi O_{ij} tidak dapat diganggu.
- f. Waktu untuk pengaturan mesin diabaikan.

2.2 *Hybrid Flowshop Scheduling*

Hybrid Flowshop Scheduling (HFS) merupakan pengembangan dari penjadwalan *flowshop* yang mempunyai mesin paralel di setiap tahapnya. Menurut Gómez-gasquet *et al.* (2012), untuk istilah *Hybrid Flowshop* pertama kali dikenalkan oleh Gupta pada tahun 1998, dalam pendeskripsian dari *Flowshop Scheduling* dengan satu mesin pada tahap pertama dan dua mesin pada tahap kedua. Terdapat beberapa

fungsi objektif dari permasalahan penjadwalan *Flowshop*, seperti contoh meminimumkan makespan.

Permasalahan penjadwalan *Flowshop* merupakan permasalahan yang memerlukan waktu lebih untuk mendapatkan solusi yang optimal. Penggunaan metode eksak seperti *Branch and Bound*, *Linear Programming* dan *Langrangian Relaxation* tidaklah cukup efektif dan membutuhkan metode lain yang dapat memberikan hasil dan perhitungan yang efektif.

Model matematika dari permasalahan *Hybrid Flowshop Scheduling* dikutip dari Liao *et al.* (2012) sebagai berikut:

$$\text{Min } Z = C_{\max} \quad (2.1)$$

Permasalahan penjadwalan *Hybrid Flowshop* dideskripsikan: Terdapat s stage di setiap sistem dan di setiap stage i terdiri dari m_i mesin identik. Semua pekerjaan (*job*) harus diproses melalui stage 1, stage 2, seterusnya sampai stage k . Job j harus diproses secara simultan pada $size_{ij}$ mesin paralel identik (*processor*) pada stage i untuk p_{js} satuan waktu. Setiap *job* tidak boleh diganggu selama proses pengerjaan. Tujuannya adalah untuk menentukan urutan proses setiap *job* (π) yang meminimumkan nilai *makespan* (C_{\max}) atau memaksimalkan waktu penyelesaian (Blazewicz *et al.*, 2001, Oguz dan Ercan, 2005).

2.3 Lower Bound

Evaluasi performansi dilakukan dengan mengujikan algoritma yang dihasilkan pada *benchmark problems*. Ketiadaan informasi mengenai nilai solusi optimal diakomodasi dengan penggunaan *Lower Bound* (LB), yaitu batas bawah (nilai minimum). *Lower Bound* dapat dicapai secara teoritis dari solusi untuk sebuah kasus. Kriteria yang umum digunakan pada masalah *Hybrid Flowshop Scheduling* adalah *makespan*. Nilai *Lower Bound* dihitung menggunakan persamaan (2.1).

$$LB = \max_{i=1, \dots, k} \{ \min_{j \in J} \{ \sum_{l=1}^{i-1} t_{lj} \} + \max \{ M_1(i), M_2(i) \} + \min_{j \in J} \{ \sum_{l=i+1}^k t_{lj} \} \} \quad (2.2)$$

dimana,

$$M_1(i) = \left\lceil \frac{1}{m_i} \sum_{j \in J} t_{ij} \cdot size_{ij} \right\rceil, \quad M_2(i) = \sum_{j \in A_i} t_{ij} + \left\lceil \frac{1}{m_i} \sum_{j \in B_i} t_{ij} \cdot size_{ij} \right\rceil$$

$$A_i = \left\{ j \mid size_{ij} > \frac{m_i}{2} \right\}, \quad B_i = \left\{ j \mid size_{ij} = \frac{m_i}{2} \right\}$$

Perbandingan antara *makespan* dan *Lower Bound* dilakukan dengan menghitung persentase deviasi atau *Percentage Deviation Algorithm* (PD_A). Perhitungan persentase deviasi adalah sebagai berikut:

$$PD_A(l) = 100 \times \frac{C_{\max}(l) - LB}{LB} \quad (2.3)$$

dimana l adalah percobaan yang dilakukan. Kemudian setelah dilakukan percobaan sebanyak L kali, dihitung *Average Percentage Deviation* (APD_A) menggunakan rumus:

$$APD_A = \frac{1}{L} \sum_{l=1}^L PD_A(l) \quad (2.4)$$

(Serifoglu dan Ulusoy, 2004).

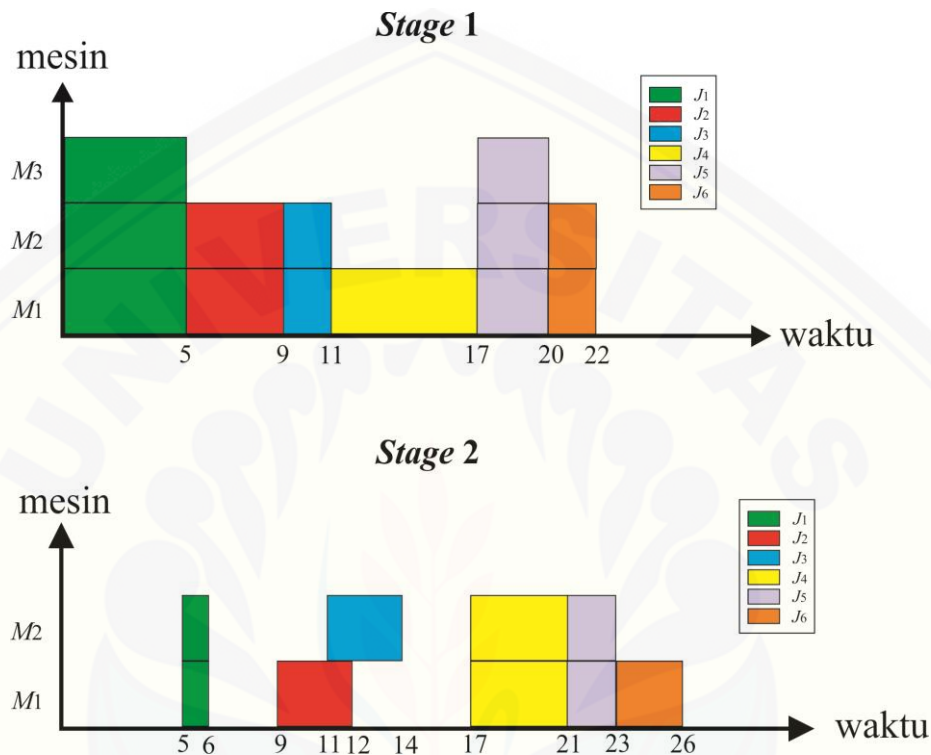
2.4 Diagram Gantt

Diagram *Gantt* merupakan diagram yang menggambarkan jenis sumber daya (sumbu vertikal) dan satuan waktu (sumbu horizontal). Diagram ini digunakan sebagai alat bantu dalam menyelesaikan masalah penjadwalan dengan set model yang sederhana dan umum digunakan secara luas. Diagram *Gantt* diperkenalkan pertama kali pada tahun 1916 oleh Henry Laurence Gantt. Misalkan diberikan data pada Tabel 2.1.

Tabel 2.1 Contoh data *Hybrid Flowshop*

Stage	Job					
	1	2	3	4	5	6
t_{lj}	5	4	2	6	3	2
$size_{lj}$	3	2	2	1	3	2

t_{2j}	1	3	3	4	2	3
$size_{2j}$	2	1	1	2	2	1



Gambar 2.1 Contoh diagram Gantt

2.5 Glowworm Swarm Optimization

Algoritma *Glowworm Swarm Optimization* (GSO) merupakan salah satu jenis metode baru dari *Swarm Intelligence*. Para agen (*Glowworm*) di GSO membawa kuantitas dari cahaya (*luminescent*) yang disebut *luciferin*. Cahaya pada setiap *Glowworm* digunakan untuk menarik tetangganya. *Glowworm* yang memiliki cahaya redup akan tertarik pada tetangganya yang bercahaya paling terang dan akan bergerak menuju *Glowworm* tetangganya tersebut. Perpindahan *Glowworm* hanya pada daerah lokal saja sehingga dari sebuah kawanan akan terbentuk beberapa kelompok (Gong *et al.*, 2011).

Pada algoritma GSO, nilai *luciferin* setiap *Glowworm* dipengaruhi oleh nilai *fitness* yang bergantung pada fungsi tujuan dari masalah optimasi. Setiap *Glowworm* menggunakan *radial sensor range* untuk mengidentifikasi tetangganya dan menghitung pergerakannya. Kemudian setiap *Glowworm* menggunakan mekanisme probabilistic untuk memilih tetangga yang memiliki nilai *luciferin* lebih tinggi dari nilainya sendiri dan selanjutnya berpindah ke tetangga tersebut.

Pengkodean adalah kunci untuk menyelesaikan masalah pada GSO untuk masalah optimasi. Prosesnya sebagai berikut: \mathbf{x}_i^t adalah lokasi dari *Glowworm* i pada iterasi t dimana $\mathbf{x}_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{id}^t]$ (d adalah banyaknya dimensi) dan \mathbf{y}_i^t adalah solusi *Glowworm* dengan $\mathbf{y}_i^t = [y_{i1}^t, y_{i2}^t, \dots, y_{id}^t]$ yang elemennya tergantung pada solusi dari masalah.

Langkah – langkah algoritma GSO adalah sebagai berikut:

- Inisialisasi parameter – parameter GSO antara lain: banyaknya *Glowworm* (n), nilai awal *luciferin* ($l_0 > 0$), nilai awal *adaptive decision-range* ($r_0 > 0$), jarak antar *Glowworm* ($r_s > 0$), tingkat *update* nilai *luciferin* ($\gamma > 0$), tingkat *update dynamic decision-range* ($\beta > 0$), batas ketetanggaan ($n_t > 0$), *step-size* ($s > 0$), tingkat disipasi nilai *luciferin* ($0 < \rho < 1$), iterasi (t), nilai maksimum iterasi. Bangkitkan *Glowworm* sebanyak n secara random.
- Hitung nilai *fitness*.

$$fitness_i = \begin{cases} f(\mathbf{y}_i), & \text{untuk kasus maksimasi} \\ \frac{1}{f(\mathbf{y}_i)}, & \text{untuk kasus minimasi} \end{cases} \quad (2.5)$$

- Perbarui nilai *luciferin*, nilai *adaptive decision-range*, dan lokasi *Glowworm* sebagai berikut:

$$l_i(t+1) = (1 - \rho)l_i(t) + \gamma(fitness_i) \quad (2.6)$$

$$N_i(t) = \{j: |l_j(t) - l_i(t)| < r_d^i(t); l_i(t) < l_j(t)\} \quad (2.7)$$

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (2.8)$$

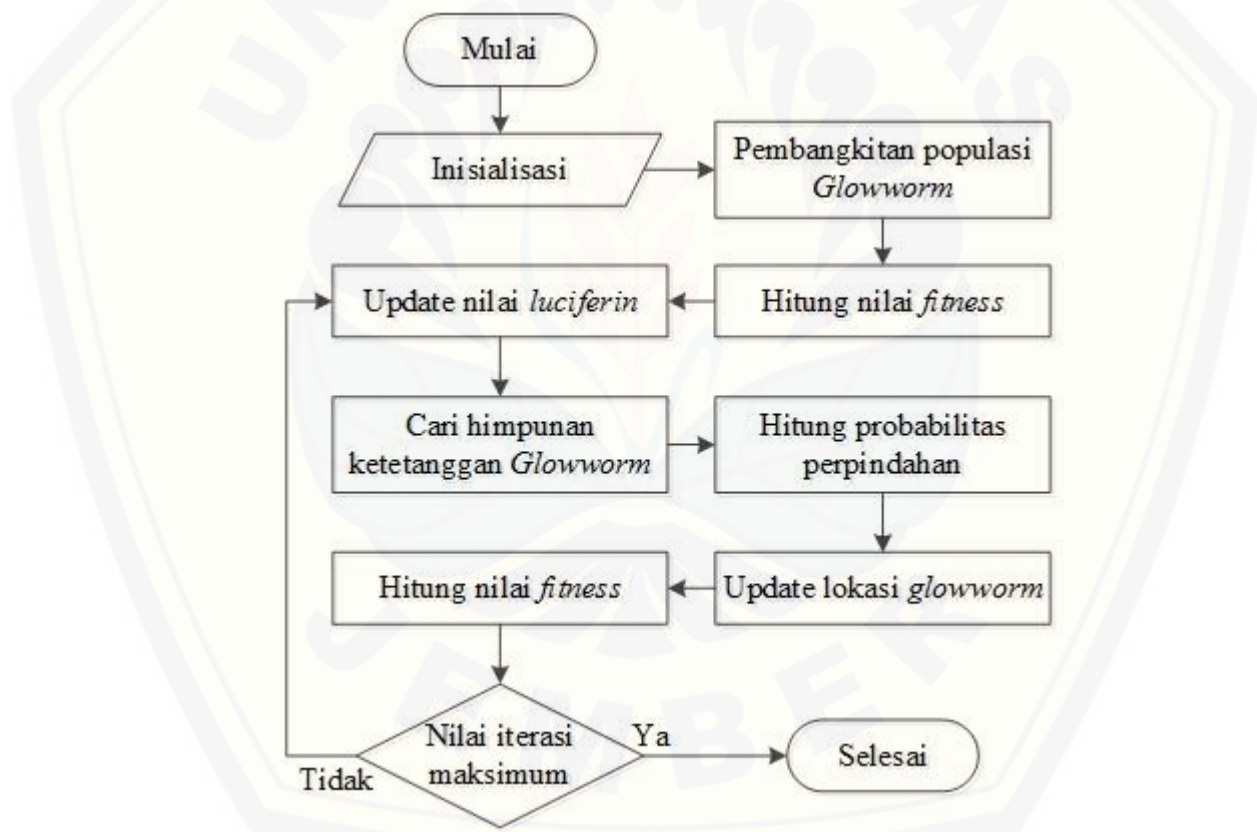
Setiap *Glowworm* i memilih satu tetangga j dengan maksimum kemungkinan $p_{ij}(t)$ dan berpindah ke tetangga j

$$x_i(t+1) = x_i(t) + s \left(\frac{x_j(t) - x_i(t)}{|x_j(t) - x_i(t)|} \right) \quad (2.9)$$

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\} \quad (2.10)$$

- d. Jika nilai maksimum iterasi terpenuhi maka iterasi dihentikan, jika belum maka kembali ke langkah b.

Adapun bagan langkah-langkah algoritma *Glowworm Swarm Optimization* akan digambarkan pada *Flowchart* sebagai berikut:



Gambar 2.2 *Flowchart* algoritma *Glowworm Swarm Optimization*

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Pada penelitian ini akan digunakan data sekunder. Data diambil dari *website* <http://home.ku.edu.tr/~coguz/research.htm>. Data tersebut merupakan data *Hybrid Flowshop* dengan *multiprocessor tasks*. Data lengkap dapat dilihat pada Lampiran A. Data tersebut merupakan data simulasi yang berisi banyaknya *job*, banyaknya *stage* dan waktu yang dibutuhkan untuk menyelesaikan setiap *job*.

3.2 Langkah-langkah Penelitian

Adapun langkah-langkah yang akan dilakukan pada penelitian ini untuk menyelesaikan permasalahan pada subbab 1.2 sebagai berikut:

a. Studi Literatur

Langkah pertama dalam penelitian ini adalah pengumpulan literatur tentang algoritma *Glowworm Swarm Optimization* dan permasalahan penjadwalan *Hybrid Flowshop*. Kemudian dilakukan studi terhadap literatur tersebut.

b. Pengambilan Data

Pada tahap ini, data *Hybrid Flowshop* yang terdapat pada *website* <http://home.ku.edu.tr/~coguz/research.htm> dalam *website* OR-Library diambil.

c. Penerapan Algoritma *Glowworm Swarm Optimization*

Data yang telah dikumpulkan pada langkah b kemudian diolah dan diselesaikan menggunakan algoritma GSO dengan langkah-langkah sebagai berikut:

1) Inisialisasi parameter

Parameter yang digunakan berupa jumlah *job*, jumlah *stage*, jumlah mesin, jumlah *Glowworm*, iterasi maksimal dan parameter lain dalam GSO.

2) Menerapkan algoritma GSO dengan langkah-langkah pada subbab 2.5 dengan memperhatikan bentuk masalah dan solusi pada subbab 2.2.

d. Pembuatan Program

Pada tahap ini, peneliti akan membuat program penerapan *Glowworm Swarm Optimization* untuk menyelesaikan penjadwalan *Hybrid Flowshop* menggunakan *software* MATLAB. Program dibuat dalam bentuk *Graphical User Interface* (GUI).

e. Simulasi Program

Program yang telah dibuat pada langkah d akan disimulasikan untuk menyelesaikan permasalahan dengan menggunakan data yang didapat pada langkah b.

f. Kesimpulan

Hasil dari simulasi program akan ditarik kesimpulan untuk menjawab rumusan masalah.

Skema langkah-langkah penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Skema langkah-langkah penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan dapat disimpulkan bahwa algoritma *Glowworm Swarm Optimization* dapat digunakan untuk penyelesaian *Hybrid Flowshop Scheduling Problem*. Selain itu, berdasarkan percobaan diperoleh beberapa hal di bawah ini:

- a. Nilai – nilai parameter yang diperoleh adalah *step size* s yaitu 0,05, jarak antar *Glowworm* r_s yaitu 10, tingkat update *dynamic decision-range* β yaitu 0,01, batas ketetangaan n_t yaitu 8. Nilai – nilai tersebut merupakan nilai yang paling optimal karena mampu menghasilkan makespan yang paling minimum. Algoritma *Glowworm Swarm Optimization* cukup efektif dilihat berdasarkan *Average Percentage Deviation*.
- b. Tingkat kecepatan kekonvergenan tidak dapat dipastikan. Waktu komputasi yang dibutuhkan untuk menyelesaikan 8 data dari 2 stage 10 *job* sampai 8 stage 50 *job* adalah sekitar 49 sampai 333 detik.

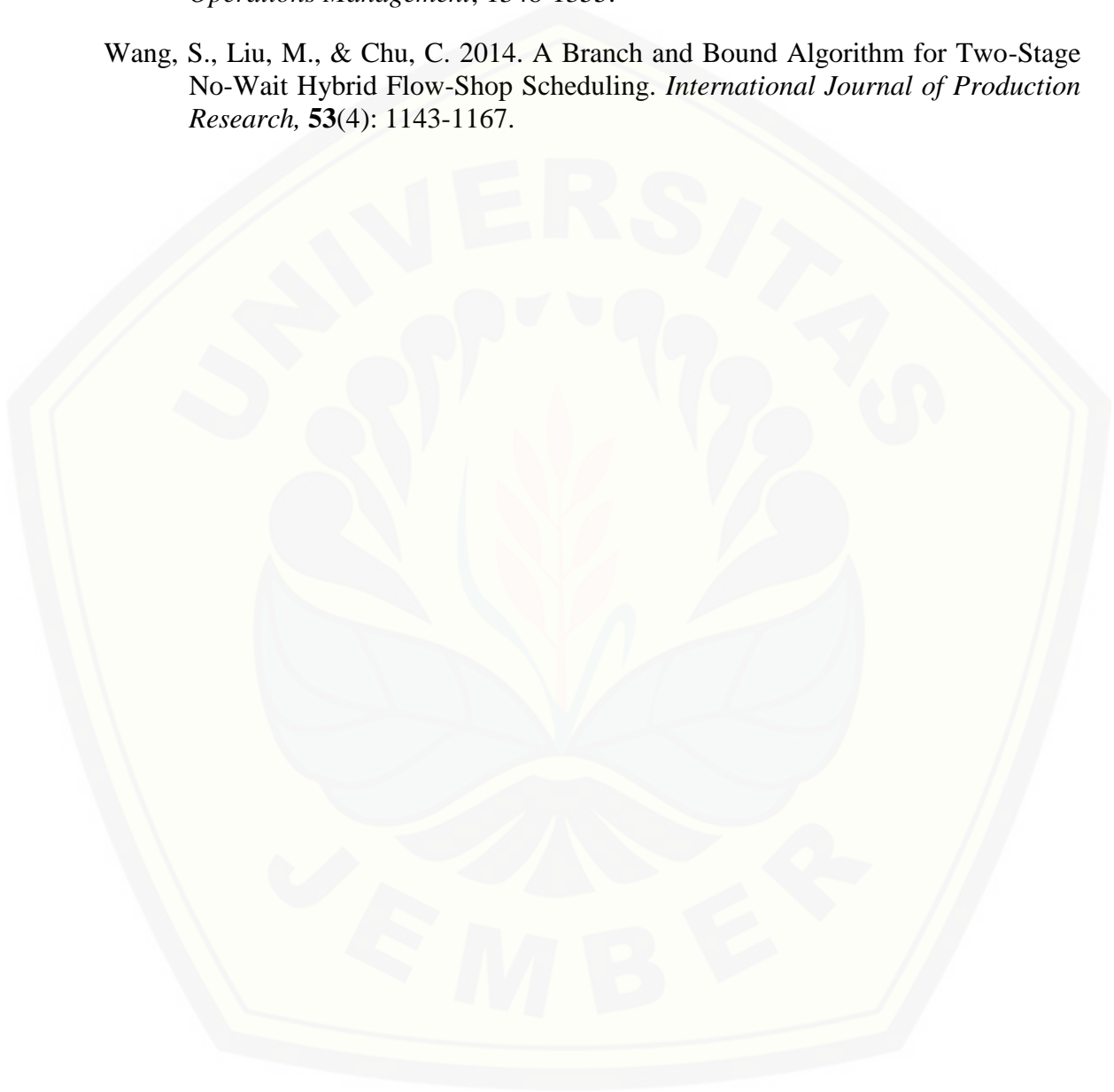
5.2 Saran

Berdasarkan kesimpulan, penulis menyarankan kepada peneliti lain yang akan meneliti lebih lanjut tentang algoritma *Glowworm Swarm Optimization* (GSO) untuk mengkombinasikan atau meng-*hybrid* algoritma ini dengan algoritma lain agar didapatkan solusi yang lebih optimum lagi.

DAFTAR PUSTAKA

- Baker, K. R. 1974. *Introduction to Sequencing And Scheduling*. New York: John Wiley And Sons, Inc.
- Błażewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. 2001. *Scheduling Computer and Manufacturing Processes* (2nd ed.). Berlin: Springer-Verlag.
- Gómez-Gasquet, P., Andrés, C., & Lario, F. C. 2012. An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan. *Expert Systems with Applications*, **39**(9): 8095-8107.
- Gong, Q., Zhou, Y., & Yang, Y. 2011. Artificial Glowworm Swarm Optimization Algorithm for Solving 0-1 Knapsack Problem. *Advanced Materials Research* **143-144**: 166-171.
- Liao, C. J., Tjandradjaja, E., & Chung, T. P. 2012. An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem. *Applied Soft Computing*, **12**(6): 1755-1764.
- Marinaki, M., & Marinakis, Y. 2016. A Glowworm Swarm Optimization algorithm for the Vehicle Routing Problem with Stochastic Demands. *Expert Systems With Applications*, **46**: 145–163.
- Oğuz, C., & Ercan, M. F. 2005. A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *Journal of Scheduling*, **8**(4): 323–351.
- Pratiwi, F. R., Rahman, A., & Tantrika, C. F. 2014. Penjadwalan Hybrid Flowshop dengan Integer Linear Programming untuk Meminimasi Makespan (Studi Kasus: Pt. Dwisutra Setia Agung Surabaya). *Jurnal Rekayasa dan Manajemen Sitem Industri*, **2**(5): 940-951.
- Putamawa, F., Santosa, B., & Siswanto, N. 2014. *Penyelesaian Permasalahan Multi-Objective Hybrid Flow Shop Scheduling dengan Algoritma Modified Particle Swarm Optimization*. Surabaya: ITS.

- Santosa, B., & Rofiq, A. 2014. The Development of Simulated Annealing Algorithm for Hybrid Flow Shop Scheduling Problem to Minimize Makespan and Total Tardiness. *International Conference on Industrial Engineering and Operations Management*, 1348-1355.
- Wang, S., Liu, M., & Chu, C. 2014. A Branch and Bound Algorithm for Two-Stage No-Wait Hybrid Flow-Shop Scheduling. *International Journal of Production Research*, **53**(4): 1143-1167.



LAMPIRAN

Lampiran A. Data Penelitian

A.1 Data HFS 2 Stage

10 Job (P10S2T05), Mesin: 1 4

Stage	Job									
	1	2	3	4	5	6	7	8	9	10
t_{1j}	94	93	89	19	30	47	91	4	26	18
size $_{1j}$	1	1	1	1	1	1	1	1	1	1
t_{2j}	30	15	73	63	75	51	78	72	34	59
size $_{2j}$	1	4	4	3	2	3	3	1	4	3

20 Job (P20S2T02), Mesin: 4 4

Stage	Job																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
t_{1j}	32	49	14	21	30	26	98	81	58	62	69	3	11	36	67	80	98	76	52	84
size $_{1j}$	3	1	1	3	2	2	3	3	3	1	1	2	1	1	2	1	1	2	3	4
t_{2j}	43	19	46	44	22	26	7	2	34	3	39	11	100	82	46	24	77	83	78	19
size $_{2j}$	4	4	4	4	3	4	3	4	2	1	2	4	1	1	1	2	2	2	2	1

50 Job (P50S2T05), Mesin: 4 3

Stage	Job																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
t_{1j}	24	7	66	65	23	61	100	5	26	76	40	47	29	56	4	29	47	76	34	61
size $_{1j}$	4	1	2	2	1	2	3	1	4	2	4	3	2	2	2	2	3	3	4	4
t_{2j}	44	74	47	32	16	1	34	66	12	77	86	87	26	87	28	100	45	9	21	73
size $_{2j}$	1	2	3	3	2	3	3	1	2	1	2	1	3	1	1	1	1	3	3	1

Stage	Job																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
t_{1j}	63	74	94	39	12	61	21	25	52	93	14	1	90	92	72	60	34	49	84	58
size $_{1j}$	4	4	4	2	1	2	3	2	3	2	3	1	2	2	1	2	4	1	1	4
t_{2j}	20	46	50	83	58	87	61	45	71	90	81	64	35	18	85	42	97	26	56	2
size $_{2j}$	1	3	3	1	3	1	2	3	2	3	1	3	2	1	2	3	2	3	3	3

Stage	Job									
	41	42	43	44	45	46	47	48	49	50
t_{1j}	14	92	92	55	11	60	99	95	29	76
size $_{1j}$	3	3	2	1	4	4	3	1	2	3
t_{2j}	96	56	93	89	97	49	54	24	43	30
size $_{2j}$	2	1	1	2	2	1	2	1	2	2

A.2 Data HFS 5 Stage

10 Job (P10S5T02), Mesin: 2 1 5 3 3

Stage	Job									
	1	2	3	4	5	6	7	8	9	10
t_{1j}	10	33	51	39	67	16	72	41	63	34
size $_{1j}$	2	2	1	2	1	2	2	1	2	1

t_{2j}	98	96	12	45	49	98	3	40	75	63
size $_{2j}$	1	1	1	1	1	1	1	1	1	1
t_{3j}	15	69	91	39	10	37	28	92	73	56
size $_{3j}$	3	2	3	4	2	4	3	5	1	3
t_{4j}	77	31	89	91	2	69	35	1	3	81
size $_{4j}$	1	2	2	2	2	1	2	3	3	2
t_{5j}	34	88	14	40	30	79	11	51	76	17
size $_{5j}$	2	1	1	1	2	1	1	3	3	3

20 Job (P20S5T02), Mesin: 4 3 3 1 3

Stage	Job																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
t_{1j}	100	16	31	5	29	5	39	29	59	49	70	85	37	97	27	41	82	81	1	17
size $_{1j}$	2	4	4	3	2	1	2	2	2	4	1	4	3	3	4	4	1	2	4	4
t_{2j}	25	4	100	89	33	47	6	52	77	90	22	53	91	10	100	86	19	38	34	71
size $_{2j}$	1	3	2	3	2	1	3	3	2	2	1	3	1	1	1	2	1	3	3	2
t_{3j}	98	15	75	58	71	93	55	14	56	51	62	96	2	92	34	17	76	94	87	76
size $_{3j}$	3	1	1	3	2	2	2	3	3	1	3	1	2	2	1	3	1	2	3	3
t_{4j}	11	5	19	39	80	24	55	80	21	5	86	46	89	61	76	77	78	44	24	20
size $_{4j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{5j}	84	2	12	39	25	15	37	50	78	35	82	28	56	48	20	11	4	20	28	26
size $_{5j}$	1	3	2	1	3	3	3	1	3	3	1	3	2	1	1	3	2	1	2	3

50 Job (P50S5T08), Mesin: 1 5 2 2 5

Stage	Job																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
t_{1j}	42	82	80	58	42	46	86	45	21	1	32	88	30	76	89	42	96	95	40	38
size $_{1j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{2j}	72	30	83	70	41	82	28	19	37	58	23	50	58	50	87	12	82	33	36	58
size $_{2j}$	2	5	2	4	1	1	3	5	5	5	1	1	5	5	2	2	2	1	4	4

t_{3j}	16	10	92	8	56	96	38	36	38	10	27	77	52	23	53	9	67	21	12	37
size _{3j}	1	2	2	1	2	1	1	1	1	2	2	1	2	1	1	2	2	1	1	2
t_{4j}	64	27	17	78	61	72	23	99	5	78	85	81	39	91	99	20	2	66	46	11
size _{4j}	2	2	1	2	1	1	2	1	2	1	2	2	2	2	1	2	1	2	2	2
t_{5j}	22	50	34	21	25	48	71	48	60	83	94	21	53	46	9	44	77	72	12	25
size _{5j}	3	5	4	2	1	2	4	3	4	1	5	1	5	2	5	2	4	4	5	4

Stage	Job																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
t_{1j}	58	70	35	48	95	58	39	19	40	53	53	33	33	3	27	94	83	32	72	68
size _{1j}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{2j}	47	59	71	85	5	15	32	12	54	20	80	52	89	13	93	62	40	6	53	83
size _{2j}	3	5	5	3	1	2	5	3	1	3	1	2	1	3	5	1	4	4	5	2
t_{3j}	60	98	43	86	98	17	73	7	72	60	36	89	75	91	35	47	33	85	38	89
size _{3j}	1	2	2	2	2	1	2	2	1	2	1	2	2	1	2	1	1	1	1	1
t_{4j}	25	70	85	96	27	36	99	41	93	77	70	7	76	21	30	15	95	50	55	5
size _{4j}	1	1	1	2	2	1	1	2	1	1	1	1	1	1	2	2	1	1	2	1
t_{5j}	91	65	84	51	99	63	86	15	50	73	22	97	39	52	55	96	13	77	86	22
size _{5j}	1	3	5	4	1	2	5	5	2	3	5	5	3	5	1	5	4	1	3	5

Stage	Job									
	41	42	43	44	45	46	47	48	49	50
t_{1j}	92	27	62	4	68	69	67	85	34	95
size _{1j}	1	1	1	1	1	1	1	1	1	1
t_{2j}	34	26	28	63	68	64	74	41	50	11
size _{2j}	4	3	1	1	2	5	3	5	4	1
t_{3j}	51	20	85	62	78	1	15	80	49	93
size _{3j}	2	2	1	1	1	2	2	1	2	1

t_{4j}	57	11	32	79	16	89	93	38	43	66
size $_{4j}$	1	2	2	1	1	2	1	2	1	2
t_{5j}	98	83	18	56	24	26	20	44	35	2
size $_{5j}$	3	1	2	1	3	2	4	1	4	5

A.3 Data HFS 8 Stage

20 Job (P20S8T01), Mesin: 1 2 4 5 1 1 4 5

Stage	Job																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
t_{1j}	73	3	42	31	16	81	40	94	56	61	53	10	59	30	44	11	79	28	36	75
size $_{1j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{2j}	37	58	94	76	16	75	96	19	40	7	60	68	91	15	80	32	11	63	25	6
size $_{2j}$	2	1	2	2	1	1	1	2	2	2	2	2	1	2	2	2	1	2	1	1
t_{3j}	90	39	51	35	53	99	66	32	75	89	43	86	82	83	94	98	67	69	100	93
size $_{3j}$	2	3	1	2	3	4	4	3	3	2	3	4	4	3	2	3	4	3	1	3
t_{4j}	22	62	25	83	33	79	75	77	87	79	21	17	70	95	52	41	28	46	62	93
size $_{4j}$	5	2	5	2	4	5	4	5	3	1	1	2	2	1	3	4	1	2	3	4
t_{5j}	69	38	38	58	46	37	45	31	55	82	34	56	36	81	47	76	71	15	17	76
size $_{5j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{6j}	19	66	47	69	60	51	54	29	16	97	52	2	8	91	66	23	64	94	27	85
size $_{6j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{7j}	11	55	43	95	84	93	56	6	24	17	33	84	86	47	62	66	88	55	72	51
size $_{7j}$	2	4	3	4	4	3	4	1	2	3	2	2	3	2	1	3	3	2	4	4
t_{8j}	31	96	18	8	43	97	78	48	92	42	29	81	24	61	55	94	99	5	63	68
size $_{8j}$	5	3	1	2	4	5	5	4	4	2	3	5	2	1	5	2	1	1	5	1

50 Job (P20S8T05), Mesin: 2 1 2 5 2 3 2 3

Stage	Job																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
t_{1j}	67	12	10	58	82	7	18	10	78	17	27	28	44	56	22	42	98	66	59	19
size $_{1j}$	1	1	1	1	1	1	1	1	1	2	1	1	2	2	2	2	1	1	2	1
t_{2j}	47	77	4	67	14	75	35	41	25	41	44	65	9	67	38	28	91	83	100	87
size $_{2j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{3j}	89	19	78	32	15	12	6	58	84	18	11	74	21	98	31	42	31	77	14	88
size $_{3j}$	2	2	2	2	2	1	1	2	2	1	1	1	1	1	2	1	1	1	1	2
t_{4j}	77	91	85	9	3	91	44	31	54	95	79	53	61	70	10	68	53	91	83	63
size $_{4j}$	4	4	4	4	1	1	5	3	5	2	2	4	1	5	4	1	3	4	5	1
t_{5j}	55	79	34	54	67	50	5	60	93	2	94	78	81	68	72	32	84	42	30	34
size $_{5j}$	2	2	1	2	1	2	2	1	2	2	1	1	2	1	2	2	1	2	1	1
t_{6j}	93	52	67	91	87	13	46	78	5	52	96	55	5	17	28	54	76	48	98	15
size $_{6j}$	2	3	1	2	1	1	1	1	1	2	2	2	3	1	2	3	1	3	3	3
t_{7j}	85	90	83	71	65	53	17	34	95	59	4	39	56	46	15	68	11	14	79	94
size $_{7j}$	1	1	1	1	1	2	1	1	2	2	2	1	2	1	2	2	1	2	2	2
t_{8j}	52	84	82	86	2	30	51	69	7	76	25	22	95	1	22	12	77	40	59	12
size $_{8j}$	1	1	3	3	2	2	2	3	1	3	1	2	1	1	2	3	1	1	2	3

Stage	Job																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
t_{1j}	46	56	15	95	33	59	12	48	41	23	1	32	77	74	61	47	6	32	77	92
size $_{1j}$	2	1	2	2	2	1	2	2	2	2	1	1	1	2	2	1	1	2	1	2
t_{2j}	80	25	64	13	19	79	51	73	6	45	88	76	31	25	2	45	3	37	85	1
size $_{2j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{3j}	64	26	30	6	59	61	15	21	62	81	31	71	13	74	65	30	9	74	55	91
size $_{3j}$	1	2	1	2	1	1	1	2	2	2	2	1	2	2	2	2	1	1	2	2
t_{4j}	24	58	62	49	42	99	47	16	77	47	67	49	99	48	86	49	73	34	16	53
size $_{4j}$	5	5	1	2	1	2	1	1	4	5	3	4	5	3	3	3	2	4	2	4

t_{5j}	74	13	3	75	67	86	13	16	29	66	72	45	7	43	25	68	8	88	14	40
size $_{5j}$	1	1	2	2	1	1	1	1	2	1	2	1	2	2	1	1	1	1	2	1
t_{6j}	81	15	95	20	19	81	55	65	93	91	85	5	14	5	99	62	91	62	45	63
size $_{6j}$	2	3	2	3	3	3	1	3	2	2	3	2	2	2	1	1	1	2	2	3
t_{7j}	94	53	50	15	17	36	55	94	17	13	28	79	68	18	95	72	69	26	58	40
size $_{7j}$	1	1	2	2	1	2	2	1	2	2	1	1	2	2	1	2	1	2	2	2
t_{8j}	74	23	78	57	75	21	11	98	10	71	35	98	78	81	53	58	80	2	40	47
size $_{8j}$	1	3	1	2	1	3	3	2	1	1	1	2	3	2	3	2	2	3	2	1

Stage	Job									
	41	42	43	44	45	46	47	48	49	50
t_{1j}	48	1	98	66	11	15	34	72	50	80
size $_{1j}$	2	2	2	2	1	1	2	1	2	2
t_{2j}	71	54	43	83	69	46	47	92	71	6
size $_{2j}$	1	1	1	1	1	1	1	1	1	1
t_{3j}	31	67	35	24	55	7	48	5	43	87
size $_{3j}$	2	2	1	1	2	2	1	1	1	1
t_{4j}	44	35	28	100	19	31	37	13	87	71
size $_{4j}$	2	5	5	5	1	1	3	3	4	5
t_{5j}	11	87	62	26	21	94	55	71	40	34
size $_{5j}$	1	2	1	1	2	2	1	2	1	1
t_{6j}	79	98	46	27	61	64	41	21	63	11
size $_{6j}$	3	1	1	1	1	3	3	1	3	2
t_{7j}	16	22	47	77	26	69	43	42	80	12
size $_{7j}$	1	1	1	1	2	1	2	2	2	2
t_{8j}	17	69	7	31	44	90	76	47	3	38
size $_{8j}$	1	3	2	3	1	3	2	2	2	1

Lampiran B. Hasil Percobaan

B.1 Hasil percobaan pengaruh parameter *step-size s*

Data	Nilai <i>s</i>	<i>Makepsan</i>	Iterasi konvergen	Waktu komputasi (detik)
<i>Stage</i> 2 10 <i>Job</i>	0.01	526	1	65.4952
	0.05	526	1	74.5362
	0.1	526	1	61.8134
	0.2	526	1	61.8612
	0.3	526	1	60.8728
	0.5	526	1	62.9816
	1	526	1	61.7105
	2	526	1	62.1158
<i>Stage</i> 2 20 <i>Job</i>	0.01	648	1	64.314
	0.05	606	112	65.2261
	0.1	657	36	73.7823
	0.2	646	34	75.7798
	0.3	649	21	80.0545
	0.5	663	9	71.0277
	1	657	1	62.1163
	2	648	3	70.9285
<i>Stage</i> 2 50 <i>Job</i>	0.01	2022	55	65.7728
	0.05	2067	13	63.1219
	0.1	2114	1	71.522
	0.2	2086	17	73.4908
	0.3	2071	13	74.9886
	0.5	2060	7	78.5211
	1	2043	4	90.1867
	2	2089	1	90.7367
<i>Stage</i> 5 10 <i>Job</i>	0.01	759	1	77.4564
	0.05	750	31	76.1284
	0.1	745	25	75.6372
	0.2	749	11	75.4672
	0.3	759	1	74.8104
	0.5	759	1	82.4521
	1	759	1	77.0626
	2	759	1	79.7768
<i>Stage</i> 5 20 <i>Job</i>	0.01	1182	26	75.935
	0.05	1187	34	76.7947
	0.1	1182	3	69.3223
	0.2	1165	2	78.8837

Data	Nilai s	<i>Makepsan</i>	Iterasi konvergen	Waktu komputasi (detik)
	0.3	1188	3	69.5969
	0.5	1193	4	76.0191
	1	1222	1	75.903
	2	1232	1	69.6506
	0.01	2925	31	78.0366
	0.05	2925	6	77.9982
5	0.1	2925	3	76.1409
<i>Stage</i>	0.2	2917	2	72.4888
50	0.3	2917	12	70.4508
<i>Job</i>	0.5	2917	1	73.1393
	1	2917	2	73.3984
	2	2932	2	124.2773
	0.01	1710	1	77.2217
	0.05	1687	29	76.629
8	0.1	1692	2	72.1852
<i>Stage</i>	0.2	1692	1	70.3822
20	0.3	1656	9	69.6152
<i>Job</i>	0.5	1674	2	72.2406
	1	1703	3	73.4693
	2	1708	3	97.0753
	0.01	3056	1	75.6533
	0.05	2961	8	78.8547
8	0.1	3048	2	74.7994
<i>Stage</i>	0.2	2995	5	76.5525
50	0.3	3056	1	72.5308
<i>Job</i>	0.5	3021	1	109.6598
	1	3026	3	72.212
	2	2991	2	72.682

B.2 Hasil percobaan pengaruh parameter jarak antar *Glowworm* r_s

Data	Nilai r_s	<i>Makepsan</i>	Iterasi konvergen	Waktu komputasi (detik)
	1	526	1	66.2331
	2	526	1	66.2554
2	3	526	1	66.3326
<i>Stage</i>	4	526	1	66.3942
10	5	526	1	66.7885
<i>Job</i>	6	526	1	66.9882
	8	526	1	67.1822

Data	Nilai r_s	Makepsan	Iterasi konvergen	Waktu komputasi (detik)
	10	526	1	67.1943
2 Stage 20 Job	1	684	1	67.2201
	2	628	63	70.2814
	3	606	112	70.3298
	4	606	112	70.7452
	5	606	112	70.8648
	6	606	112	71.1361
	8	606	112	71.2939
	10	606	112	71.7571
2 Stage 50 Job	1	2114	1	67.7542
	2	2114	1	68.7937
	3	2114	1	69.7688
	4	2067	13	70.8589
	5	2067	13	71.4243
	6	2067	13	71.8432
	8	2067	13	72.1956
	10	2067	13	72.5891
5 Stage 10 Job	1	759	1	68.0724
	2	759	1	68.8886
	3	759	1	69.2369
	4	750	31	69.2444
	5	750	31	69.585
	6	750	31	69.6592
	8	750	31	69.9926
	10	750	31	70.1602
5 Stage 20 Job	1	1187	14	70.2726
	2	1187	14	70.6794
	3	1187	14	70.8312
	4	1187	14	71.0628
	5	1187	14	71.1746
	6	1187	14	71.5978
	8	1187	14	71.9975
	10	1164	58	72.227
5 Stage 50 Job	1	2931	2	74.1861
	2	2931	2	74.7597
	3	2925	6	75.064
	4	2925	6	75.51
	5	2925	6	75.8697
	6	2925	6	76.0753

Data	Nilai r_s	Makepsan	Iterasi konvergen	Waktu komputasi (detik)
	8	2925	6	76.697
	10	2925	6	76.9647
	1	1704	14	71.8439
	2	1704	14	71.9784
8	3	1704	14	72.0889
Stage	4	1687	29	72.4933
20	5	1687	29	72.5336
Job	6	1687	29	72.9788
	8	1687	29	73.5416
	10	1687	29	73.7337
	1	3034	3	75.7281
	2	3034	3	75.8479
8	3	3024	3	75.9595
Stage	4	2961	8	77.1243
50	5	2961	8	77.3076
Job	6	2961	8	77.8055
	8	2961	8	78.7316
	10	2961	8	79.6392

B.3 Hasil percobaan pengaruh parameter *dynamic-decision range* β

Data	Nilai β	Makepsan	Iterasi konvergen	Waktu komputasi (detik)
	0.01	526	1	67.3345
	0.05	526	1	73.5664
2	0.1	526	1	77.4261
Stage	0.2	526	1	71.146
10	0.3	526	1	70.945
Job	0.5	526	1	71.578
	1	526	1	72.0104
	2	526	1	70.5966
	0.01	627	99	102.5602
	0.05	638	114	83.3421
2	0.1	606	113	87.4494
Stage	0.2	606	112	77.6376
20	0.3	606	112	76.1781
Job	0.5	628	63	72.1205
	1	628	63	75.5232
	2	684	1	70.4747

Data	Nilai β	Makepsan	Iterasi konvergen	Waktu komputasi (detik)
<i>2</i> <i>Stage</i> <i>50</i> <i>Job</i>	0.01	2022	41	110.626
	0.05	1968	129	95.0056
	0.1	1978	131	107.4699
	0.2	2067	13	71.7595
	0.3	2067	13	74.8685
	0.5	2067	13	75.6661
	1	2114	1	70.9995
	2	2114	1	72.4114
<i>5</i> <i>Stage</i> <i>10</i> <i>Job</i>	0.01	725	58	55.9569
	0.05	745	46	53.0469
	0.1	745	46	56.4529
	0.2	745	46	56.36
	0.3	750	31	55.4864
	0.5	759	1	51.1955
	1	759	1	56.5564
	2	759	1	51.238
<i>5</i> <i>Stage</i> <i>20</i> <i>Job</i>	0.01	1120	70	84.3605
	0.05	1190	26	63.6569
	0.1	1190	26	59.6685
	0.2	1164	58	69.3407
	0.3	1164	58	67.761
	0.5	1223	12	53.141
	1	1231	1	50.7915
	2	1231	1	53.01
<i>5</i> <i>Stage</i> <i>50</i> <i>Job</i>	0.01	2905	118	86.9878
	0.05	2907	12	62.021
	0.1	2925	6	64.342
	0.2	2925	6	61.9796
	0.3	2925	6	62.1523
	0.5	2925	6	55.7612
	1	2943	1	53.222
	2	2943	1	50.6687
<i>8</i> <i>Stage</i> <i>20</i> <i>Job</i>	0.01	1626	69	81.6542
	0.05	1692	4	61.6821
	0.1	1692	4	63.5528
	0.2	1687	29	55.4224
	0.3	1687	29	56.9529
	0.5	1687	29	81.1879
	1	1704	14	81.1767

Data	Nilai β	Makepsan	Iterasi konvergen	Waktu komputasi (detik)
	2	1704	14	76.5842
	0.01	2853	23	206.8313
	0.05	2906	122	212.7489
8	0.1	2961	8	93.7085
Stage	0.2	2941	120	126.3642
50	0.3	2961	8	77.8543
Job	0.5	3034	3	78.0177
	1	3056	1	71.6296
	2	3056	1	82.8214

B.4 Hasil percobaan pengaruh parameter batas ketetanggaan n_t

Data	Nilai n_t	Makepsan	Iterasi konvergen	Waktu komputasi (detik)
	1	526	1	76.3438
	2	526	1	87.3387
2	3	526	1	70.5194
Stage	4	526	1	75.3412
10	5	526	1	74.1971
Job	6	526	1	73.8553
	8	526	1	73.8672
	10	526	1	75.0461
	1	643	48	85.9743
	2	643	48	85.8209
2	3	643	48	103.5064
Stage	4	629	85	86.8117
20	5	627	99	107.6044
Job	6	627	99	108.4788
	8	629	80	90.7451
	10	629	80	93.1031
	1	2040	21	82.2886
	2	2040	21	84.673
2	3	2040	21	86.2159
Stage	4	1994	137	157.5554
50	5	2022	41	110.4523
Job	6	2000	107	160.7264
	8	1969	144	246.5466
	10	1981	134	296.299
5	1	720	76	76.611
Stage	2	725	58	75.7189

Data	Nilai n_t	Makepsan	Iterasi konvergen	Waktu komputasi (detik)	
10 <i>Job</i>	3	725	58	85.5774	
	4	725	58	86.5189	
	5	725	58	79.704	
	6	725	58	80.2974	
	8	725	58	80.1761	
	10	725	58	83.064	
5 <i>Stage</i> 20 <i>Job</i>	1	1120	70	91.8567	
	2	1120	70	86.0788	
	3	1120	70	87.2667	
	4	1120	70	107.7844	
	5	1120	70	109.6572	
	6	1103	80	80.1505	
5 <i>Stage</i> 50 <i>Job</i>	8	1103	80	91.9317	
	10	1103	80	84.531	
	1	2907	12	90.7508	
	2	2907	12	97.1599	
	3	2907	12	99.2198	
	4	2907	12	133.2993	
20 <i>Job</i>	5	2905	118	117.3645	
	6	2905	109	124.6034	
	8	2875	88	138.4221	
	10	2875	88	173.7047	
	8 <i>Stage</i> 20 <i>Job</i>	1	1583	96	76.1849
		2	1626	69	80.1088
3		1626	69	82.6889	
4		1626	69	92.2128	
5		1626	69	106.2669	
6		1626	69	109.92	
50 <i>Job</i>	8	1581	91	86.2645	
	10	1569	114	93.5464	
	1	2853	23	104.319	
	2	2853	23	119.5041	
	3	2853	23	124.6665	
	4	2853	23	208.1228	
8 <i>Stage</i> 50 <i>Job</i>	5	2853	23	212.1925	
	6	2853	23	363.116	
	8	2853	23	477.8407	
	10	2853	23	580.66	

B.5 Hasil Percobaan Data 2 Stage 10 Job

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
1	8-4-5-7-9-10-6-1-3-2	526	1	47,4516	0
2	4-7-8-3-9-10-6-1-5-2	526	1	47,7747	0
3	5-4-8-10-6-9-7-1-3-2	526	1	46,9762	0
4	10-8-4-7-6-3-9-5-1-2	526	2	49,1002	0
5	8-4-10-3-6-5-9-1-7-2	526	1	47,1935	0
6	5-3-9-7-8-10-1-4-6-2	526	10	49,7342	0
7	6-5-9-7-8-3-4-1-10-2	526	51	54,2773	0
8	4-7-8-6-3-5-9-1-10-2	526	17	52,2522	0
9	5-4-10-8-3-1-6-9-7-2	526	1	51,5985	0
10	4-3-9-10-6-8-5-7-1-2	526	1	51,703	0

B.6 Hasil Percobaan Data 2 Stage 20 Job

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
1	12-15-18-3-4-17-19-6-1-2-14-20-11-5-9-16-13-7-8-10	613	78	78,317	11,6576
2	6-18-5-12-13-9-10-17-11-19-4-2-3-15-1-14-20-7-16-8	627	105	73,7233	14,2077
3	18-14-2-4-10-9-12-17-19-3-7-11-13-5-6-1-16-15-20-8	618	106	79,4028	12,5683
4	3-1-14-19-6-15-12-18-17-5-11-13-9-2-16-4-7-20-8-10	605	89	66,3232	10,2004
5	19-13-14-17-9-5-1-10-16-4-11-3-6-12-15-18-2-8-20-7	627	104	65,3219	14,2077
6	17-1-6-8-10-12-2-9-5-15-16-4-13-14-19-3-7-11-18-20	639	107	104,314	16,3934
7	3-1-16-5-10-14-19-9-2-4-15-18-13-8-17-12-6-11-7-20	612	64	70,7132	11,4754
8	4-1-11-3-7-13-10-14-17-16-12-9-2-19-20-18-15-5-6-8	610	124	70,5962	11,1111
9	6-11-3-19-13-12-20-18-15-4-1-2-7-14-17-10-9-16-8-5	660	131	127,9762	9,2896
10	11-14-19-3-15-13-18-17-8-1-20-6-12-16-9-2-4-7-10-5	622	68	86,5082	13,2969

B.7 Hasil Percobaan Data 2 Stage 50 Job

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
1	11-33-3-15-2-43-29-26-10-13-41-47-14-24-49-19-42-31-16-5-27-4-8-46-7-35-17-25-36-6-1-37-9-32-44-12-21-28-39-30-23-20-45-22-40-50-34-48-18-38	1945	143	181,0873	14,2773
2	8-12-39-32-30-25-24-44-37-29-11-33-43-47-41-14-49-17-15-2-23-18-28-20-16-21-1-42-9-35-36-7-13-4-45-6-40-22-38-26-5-19-31-27-46-48-10-50-34-3	2015	119	177,0347	18,3901
3	29-37-24-26-45-21-39-4-17-43-35-33-36-8-16-5-50-2-23-6-28-25-30-14-46-41-10-22-47-11-38-7-31-27-32-49-15-3-40-20-12-9-13-18-48-19-44-42-1-34	1963	137	186,1524	15,3349
4	28-14-2-41-20-22-5-25-12-3-4-23-26-35-40-36-6-31-19-15-47-11-13-45-7-44-46-39-49-33-48-32-30-9-8-21-29-37-24-17-27-10-43-16-42-18-38-1-50-34	1988	110	167,3641	16,8038
5	36-26-41-30-33-31-28-3-46-43-39-11-16-2-22-32-23-6-4-20-14-47-45-50-10-5-24-29-38-19-18-15-8-9-27-21-42-37-25-34-49-13-48-44-40-1-17-12-35-7	1971	139	174,4248	15,8049
6	49-13-2-31-44-26-22-50-27-29-38-42-48-46-39-17-7-45-11-1-40-24-15-30-28-36-25-23-3-14-12-10-47-33-43-18-16-35-37-20-9-32-41-8-34-19-6-4-5-21	1972	136	154,9883	15,8637
7	36-2-30-23-9-24-3-15-46-25-7-44-10-6-14-26-34-17-39-28-21-37-20-48-49-18-8-45-38-12-35-40-29-1-22-41-27-19-5-42-50-32-43-33-47-31-13-4-16-11	2021	127	149,186	18,7427
8	45-21-8-10-33-9-32-13-39-15-41-26-24-20-42-2-11-14-5-49-38-34-29-25-22-37-43-30-36-3-40-7-19-23-48-31-17-18-46-12-35-28-4-6-44-1-27-16-47-50	1915	128	170,9993	12,5147
9	22-1-27-16-45-35-12-9-31-38-26-11-29-42-19-39-18-7-37-21-2-8-40-3-25-5-4-30-28-24-46-50-44-34-43-13-20-17-36-15-14-41-6-23-32-49-48-47-10-33	1944	119	197,6148	14,2186

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
10	8-26-39-18-15-5-12-35-47-44-1-27-24-37-21-22-31-34-49-16-38-25-45-20-23-36-50-9-40-3-28-6-33-13-2-43-30-32-4-14-41-46-19-10-17-48-11-42-29-7	1977	116	158,5474	16,1575

B.8 Hasil Percobaan Data 5 Stage 10 Job

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
1	6-10-3-9-2-5-7-8-4-1	721	19	51,9549	14,2631
2	6-8-2-7-4-10-9-3-1-5	714	77	53,1454	13,1537
3	6-8-5-4-2-3-7-9-10-1	721	67	53,702	14,2631
4	1-3-6-8-9-4-2-10-5-7	722	88	54,3095	14,4216
5	6-3-8-10-2-1-9-4-5-7	721	32	52,3792	14,2631
6	2-3-8-4-6-9-7-10-1-5	719	112	52,746	13,9461
7	10-9-3-8-4-2-6-1-5-7	717	86	52,7283	13,6292
8	6-2-3-5-8-4-9-7-10-1	721	57	52,5945	14,2631
9	6-4-9-3-10-2-7-5-8-1	721	56	53,4394	14,2631
10	6-8-4-9-10-2-3-1-5-7	699	59	55,157	10,7765

B.9 Hasil Percobaan Data 5 Stage 20 Job

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
1	19-8-7-4-17-10-2-11-5-1-13-9-3-15-16-6-14-12-18-20	1107	118	61,1561	13,306
2	5-7-16-17-11-6-13-2-18-3-1-4-20-8-9-15-10-12-14-19	1127	110	58,5053	15,3531
3	6-17-5-4-7-13-8-1-19-16-15-2-20-9-10-14-11-3-12-18	1119	107	56,4718	14,5343

4	13-6-16-19-10-15-17-8-11-12-14-7-2-9-18-20-5-3-1-4	1161	34	62,6908	18,8332
5	7-11-5-8-19-4-14-16-6-12-3-15-2-17-10-9-13-1-18-20	1106	93	62,3081	13,2037
6	19-5-13-9-8-7-20-11-15-17-10-2-4-1-3-14-6-16-18-12	1137	123	74,78	16,3767
7	8-5-11-13-19-9-7-17-10-15-16-1-20-4-18-14-12-6-3-2	1103	90	74,9716	12,8966
8	6-13-5-19-14-17-12-10-7-20-9-15-11-4-8-1-3-2-16-18	1151	84	57,746	17,8096
9	8-11-13-20-1-4-7-17-19-5-16-9-2-15-6-3-14-18-10-12	1134	85	64,1944	16,0696
10	7-5-20-2-16-11-13-1-12-14-15-3-17-18-9-4-8-19-10-6	1108	110	57,8796	13,4084

B.10 Hasil Percobaan Data 5 Stage 50 Job

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
1	30-24-15-40-12-34-3-16-42-35-23-28-32-47-5-4-22-33-29-14-43-49-1-38-26-13-46-50-10-17-27-45-6-2-11-8-21-44-36-9-25-37-19-18-39-41-7-31-48-20	2910	44	128,4264	3,1915
2	15-28-3-22-10-8-49-44-17-11-34-36-4-39-21-18-41-24-13-6-43-25-23-35-7-5-16-19-30-32-45-42-26-14-12-29-27-9-1-46-50-47-33-40-38-37-2-31-48-20	2910	74	163,9998	3,1915
3	39-42-34-1-45-40-21-9-6-28-2-25-33-18-8-49-32-37-23-24-29-19-26-30-47-15-31-50-35-12-4-41-22-44-36-11-27-10-14-16-38-3-17-20-46-5-7-13-48-43	2908	100	90,143	3,1206
4	50-27-18-46-36-39-3-30-1-10-28-11-31-49-23-38-9-16-35-32-5-25-17-15-33-13-34-4-45-19-6-14-22-42-24-12-47-48-21-41-44-2-20-26-40-8-29-37-7-43	2908	112	91,4345	3,1206
5	35-46-1-32-11-48-38-22-5-49-44-17-30-25-8-39-28-34-29-24-40-31-47-6-19-10-3-2-16-4-45-42-15-43-33-41-37-26-23-27-9-12-13-20-14-36-50-18-21-7	2905	40	85,2587	3,0142
6	30-37-35-10-18-21-41-46-14-13-25-8-39-1-6-33-20-23-44-4-9-49-34-28-7-45-31-11-42-22-32-12-24-27-3-16-29-48-38-40-5-36-17-50-15-47-2-26-19-43	2908	44	89,7149	3,1206

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
7	27-14-36-31-7-18-40-11-43-42-39-1-16-23-12-41-24-47-6-9-3-25-35-5-13-8-22-44-15-46-49-34-33-32-29-21-17-19-28-20-37-4-2-10-30-48-38-26-45-50	2917	19	73,0594	3,4397
8	35-27-3-46-11-16-14-34-1-7-6-36-30-28-38-10-45-5-37-32-20-24-29-33-41-39-8-2-42-9-26-19-17-15-49-47-13-4-22-21-50-31-23-40-18-44-25-12-48-43	2908	89	94,4343	3,1206
9	17-27-6-48-20-3-11-45-33-35-24-36-43-28-39-4-38-49-15-34-12-7-46-30-5-10-32-37-42-26-23-13-18-2-16-19-14-22-40-1-47-44-21-31-9-8-25-29-41-50	2917	17	72,3455	3,4397
10	38-31-15-34-45-44-1-23-32-35-14-18-29-26-36-4-12-49-37-47-13-2-17-22-11-41-5-3-33-39-21-9-8-10-30-42-27-6-25-19-24-20-46-40-28-48-7-16-50-43	2908	14	84,5518	3,1206

B.11 Hasil Percobaan Data 8 Stage 20 Job

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
1	19-14-18-12-4-10-6-2-17-20-16-3-8-7-13-15-5-1-11-9	1613	98	60,1216	21,9199
2	10-17-16-19-9-14-6-1-15-11-12-5-7-3-2-4-20-13-18-8	1594	87	64,4923	20,4923
3	5-10-2-7-17-19-14-9-18-4-15-8-6-12-20-16-13-11-1-3	1552	111	62,8826	17,3091
4	5-14-15-6-20-12-8-3-17-7-2-13-1-10-9-19-4-16-18-11	1587	108	72,5446	19,9546
5	14-9-10-4-16-19-11-12-20-8-6-17-15-2-1-5-7-18-13-3	1638	114	63,6489	23,8095
6	12-4-19-14-13-5-15-10-20-6-7-2-3-9-17-16-1-8-18-11	1586	118	62,5378	19,8791
7	17-11-19-20-14-16-8-3-5-7-4-9-2-10-1-6-15-13-18-12	1618	58	76,2645	22,2978
8	14-16-2-19-5-8-18-12-20-3-10-1-7-13-9-15-17-11-6-4	1606	31	75,4029	21,3908
9	16-4-17-5-2-12-15-10-3-19-18-11-9-6-7-20-14-8-13-1	1544	100	59,6446	16,7045
10	14-7-19-2-4-9-20-8-5-17-10-15-11-6-1-3-18-12-16-13	1576	92	59,78	19,1232

B.12 Hasil Percobaan Data 8 Stage 50 Job

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
1	46-12-13-8-6-29-42-20-16-4-21-44-3-10-40-1-24-41-47-26-43-30-38-17-9-35-37-22-15-5-19-18-14-48-34-23-33-31-50-39-36-45-2-28-49-27-7-32-11-25	2920	123	328,9645	10,6899
2	46-19-29-2-33-18-5-42-10-39-35-11-25-13-1-23-36-21-45-50-41-26-40-9-47-17-3-24-12-6-27-37-7-30-8-4-34-14-20-22-15-31-44-48-28-32-16-43-38-49	2872	88	241,4877	8,8704
3	30-2-15-46-50-9-7-36-19-47-13-27-24-11-16-34-38-18-20-21-3-49-25-17-44-43-28-22-29-5-33-48-10-40-42-45-39-37-35-8-1-4-26-12-32-6-14-23-31-41	2889	131	307,4405	9,5148
4	21-31-9-29-39-16-42-11-8-37-18-35-24-32-12-10-46-28-4-5-14-22-34-7-27-48-50-43-1-25-3-13-49-19-23-40-33-41-6-36-2-45-30-17-38-20-47-15-26-44	2905	109	333,1824	10,1213
5	3-7-42-39-15-36-35-49-32-5-11-10-1-9-30-24-19-37-23-22-4-25-40-38-43-2-27-14-41-46-16-6-50-13-29-45-28-26-33-21-34-18-47-8-44-48-31-20-17-12	2817	112	294,5095	6,7854
6	11-46-17-49-8-33-4-37-21-22-35-24-27-38-23-42-15-34-9-16-47-28-36-18-3-32-30-6-26-25-13-2-10-50-39-19-7-31-1-48-29-43-40-20-44-12-5-45-14-41	2850	124	343,1205	8,0364
7	46-36-38-8-32-13-9-39-31-2-25-49-42-35-29-30-1-3-21-17-4-37-10-45-20-5-27-24-7-19-22-47-50-6-41-11-28-12-33-43-14-44-15-40-23-26-34-48-18-16	2859	114	402,3203	8,3776
8	19-28-13-17-38-40-45-11-47-5-27-1-15-25-37-20-35-42-39-10-41-2-4-6-29-16-8-24-9-7-32-23-22-30-3-26-34-18-46-44-43-50-36-48-33-21-49-31-12-14	2911	39	401,4654	10,3487
9	3-8-15-19-16-28-29-20-21-2-33-38-40-22-4-37-9-27-35-10-25-41-26-44-17-14-5-30-1-6-50-46-13-42-23-7-24-47-49-31-34-48-43-12-18-32-11-45-36-39	2880	125	281,3065	9,1736

Percobaan	Urutan Job	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PDA (%)
10	31-47-42-43-11-9-21-33-22-48-15-50-4-38-30-44-16-3-8-23-36-19-46-34-37-5-35-20-10-1-7-49-25-39-24-12-14-41-27-13-17-29-28-18-6-26-40-2-32-45	2906	113	293,0847	10,1592

