



**PENYELESAIAN MASALAH *HYBRID FLOWSHOP SCHEDULING*
DENGAN ALGORITMA *HYBRID TABU SEARCH***

SKRIPSI

Oleh:

**Saiful Fadilah
NIM 121810101027**

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



**PENYELESAIAN MASALAH *HYBRID FLOWSHOP SCHEDULING*
DENGAN ALGORITMA *HYBRID TABU SEARCH***

SKRIPSI

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

Oleh:

Saiful Fadilah
NIM 121810101027

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. (Alm) Ibunda Sri Anggarini terimakasih atas limpahan kasih sayang semasa hidupnya dan memberikan rasa rindu yang berarti;
2. Ayahanda Abdur Rahmad tercinta yang senantiasa memberi doa, inspirasi, semangat dan kasih sayang;
3. Kakakku Suhartini yang memberikan dukungan moril;
4. Kekasihku Devita Arum Seruni yang memberikan dukungan moril serta materil serta semangat yang tulus;
5. Seluruh guru dan dosen sejak taman kanak-kanak hingga perguruan tinggi yang telah memberi banyak ilmu dan membimbing dengan tulus;
6. Almamater Jurusan Matematika FMIPA Universitas Jember, SMA Negeri 2 Situbondo, SMP Negeri 1 Banyuputih dan SD Negeri 4 Sumberwaru;

MOTTO

”Sesuatu mungkin mendatangi mereka yang mau menunggu, namun hanya didapatkan oleh mereka yang bersemangat mengejarnya.”

(Abraham Lincoln)

”Bermimpilah seolah-olah anda hidup selamanya. Hiduplah seakan-akan inilah hari terakhir anda.”

(James Dean)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Saiful Fadilah

NIM : 121810101027

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Penyelesaian Masalah *Hybrid Flowshop Scheduling* dengan Algoritma *Hybrid Tabu Search*" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan dalam institusi manapun dan juga 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,

Saiful Fadilah
NIM 121810101027

SKRIPSI

**PENYELESAIAN MASALAH *HYBRID FLOWSHOP SCHEDULING*
DENGAN ALGORITMA *HYBRID TABU SEARCH***

Oleh

Saiful Fadilah
NIM 121810101027

Pembimbing:

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

Dosen Pembimbing Anggota : M. Ziaul Arif, S.Si., M.Sc.

PENGESAHAN

Skripsi berjudul “Penyelesaian Masalah *Hybrid Flowshop Scheduling* dengan Algoritma *Hybrid Tabu Search*” 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.

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

NIP. 197704302005011001

NIP. 198501112008121002

Dosen Penguji I,

Dosen Penguji II,

Drs. Rusli Hidayat, M.Sc.

Dr. Alfian Futuhul Hadi S.Si., M.Si.

NIP. 196610121993031001

NIP. 198202162006042002

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Sujito, Ph. D.

NIP. 196102041987111001

RINGKASAN

Penyelesaian Masalah *Hybrid Flowshop Scheduling* dengan Algoritma *Hybrid Tabu Search*; Saiful Fadilah, 121810101027; 2016; 48 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penjadwalan adalah suatu proses pengalokasian sumber daya (mesin) yang terbatas untuk menyelesaikan sejumlah pekerjaan (*job*) berbeda. Permasalahan yang muncul pada penjadwalan terjadi apabila pada tahapan operasi tertentu beberapa atau seluruh pekerjaan membutuhkan stasiun kerja yang sama sehingga perlu adanya pengurutan pekerjaan dalam suatu produksi. Salah satu permasalahan dalam penjadwalan adalah penjadwalan *Flowshop*.

Penjadwalan *Flowshop* adalah penjadwalan proses produksi dari masing-masing n *job* yang mempunyai urutan proses produksi dan melalui m mesin yang sama. Permasalahan penjadwalan *Flowshop* sendiri ada beberapa macam sesuai dengan kondisi dan asumsi yang digunakan. Permasalahan yang sering timbul di perusahaan adalah penjadwalan yang memiliki satu operasi atau lebih yang memiliki lebih dari satu mesin. Permasalahan tersebut termasuk dalam kategori *Hybrid Flowshop Scheduling Problem*.

Pada penelitian ini akan diselesaikan permasalahan *Hybrid Flowshop Scheduling* menggunakan algoritma *Hybrid Tabu Search*. Data *Hybrid Flowshop* yang digunakan adalah data sekunder dengan banyak *stage* adalah 2, 5 dan 8 serta banyak *job* adalah 5, 20 dan 50. Data tersebut berisi banyaknya mesin pada setiap *stage*, waktu dan banyaknya mesin yang dibutuhkan untuk menyelesaikan setiap *job*.

Hasil penelitian yang diperoleh setelah melakukan percobaan sebagai berikut. Nilai parameter replika yang semakin besar, menghasilkan solusi yang lebih optimal.

Nilai parameter *threshold* yang paling baik adalah 20, karena nilai *threshold* yang kurang atau lebih dari 20 menghasilkan solusi yang kurang optimum. Tingkat kecepatan kekonvergenan algoritma *Hybrid Tabu Search* tidak dapat dipastikan karena terdapat proses random pada langkah pencarian ketetangaan (*swap*). Waktu komputasi rata-rata yang dibutuhkan untuk menyelesaikan 5 *job* adalah 15,6374 detik (500 iterasi) dan 32,1344 detik (1000 iterasi). Waktu komputasi rata-rata yang dibutuhkan untuk menyelesaikan 10 *job* adalah 38,5006 detik (500 iterasi) dan 73,5739 detik (1000 iterasi). Waktu komputasi rata-rata yang dibutuhkan untuk menyelesaikan 50 *job* adalah 75,8607 detik (500 iterasi) dan 150,2858 detik (1000 iterasi).

Nilai APD_A pada data 2 *Stage* 20 *Job* dan 2 *Stage* 50 *Job* adalah 0 %, yang artinya nilai makespan yang dihasilkan merupakan nilai paling optimal. Sedangkan nilai APD_A terburuk pada hasil penyelesaian data 8 *Stage* 5 *Job* yaitu 11,8056 %. Menurut hasil ini, dapat dikatakan bahwa algoritma *Hybrid Tabu Search* sangat efektif karena nilai yang dihasilkan sama dengan atau mendekati nilai optimal.

PRAKATA

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik dan karuniaNya sehingga skripsi yang berjudul ” Penyelesaian Masalah *Hybrid Flowshop Scheduling* dengan Algoritma *Hybrid Tabu Search*” dapat terselesaikan. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata 1 (S1) di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember. Sholawat dan salam semoga tercurahkan keharibaan Nabi Muhammad SAW yang telah menjadi pembawa rahmatan lil’alamin.

Penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Drs. Sujito, Ph.D. selaku Dekan Fakultas MIPA Universitas Jember dan Kusbudiono S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas MIPA Universitas Jember yang telah memberikan fasilitas-fasilitas dalam tahap penelitian;
2. Kusbudiono S.Si., M.Si. selaku Dosen Pembimbing Utama dan M. Ziaul Arif, S.Si., M.Sc. selaku Dosen Pembimbing Anggota yang telah memberikan bimbingan dan bantuan untuk pengerjaan skripsi ini;
3. Drs. Rusli Hidayat, M.Sc. selaku Dosen Penguji I dan Dr. Alfian Futuhul Hadi S.Si., M.Si. selaku Dosen Penguji II yang telah memberikan kritik dan saran yang membangun untuk penyempurnaan skripsi ini;
4. Drs. Rusli Hidayat, M.Sc. selaku Dosen Pembimbing Akademik yang telah membimbing dalam pemilihan mata kuliah;
5. Seluruh dosen dan karyawan Jurusan Matematika Fakultas MIPA Universitas Jember yang telah memberikan ilmu serta membantu selama proses perkuliahan berlangsung;

6. (Alm) Ibu Sri Anggarini yang selalu memberi doa dan dukungan baik lahir maupun batin selama hidupnya;
7. Bapak Abdur Rahmad yang selalu memberi doa dan dukungan baik lahir maupun batin;
8. Kakak Suhartini yang telah memberi semangat dan dukungan;
9. Devita Arum Seruni yang telah sabar dan memberi semangat, dukungan dan bantuan moral serta materi untuk pengerjaan skripsi ini;
10. Teman-teman Bathics'12 yang selalu memberikan dukungan dalam hal pembelajaran selama kuliah;
11. Kawan UKM PALAPA yang sudah memberikan wawasan dan menjadi teman diskusi;
12. Kawan-kawan CT yang telah memberi semangat dan dukungan moral dalam pengerjaan skripsi ini;
13. Kawan-kawan The Kos 207 yang telah memberi semangat dan dukungan moral dalam pengerjaan skripsi ini;
14. Semua pihak yang membantu terselesaikannya skripsi ini.

Penulis menyadari bahwa dalam menyusun skripsi ini masih terdapat kekurangan baik isi maupun susunannya. Oleh sebab itu, penulis mengharapkan saran dan kritik demi penyempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat memberi manfaat dan sumbangan pengetahuan bagi pembaca.

Jember Desember 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN.....	iv
HALAMAN PEMBIMBING	v
PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI.....	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xv
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 Definisi Penjadwalan	4
2.2 Penjadwalan <i>Flowshop</i>	5
2.3 Penjadwalan <i>Hybrid Flowshop</i>	5
2.4 <i>Lower Bound</i>	6

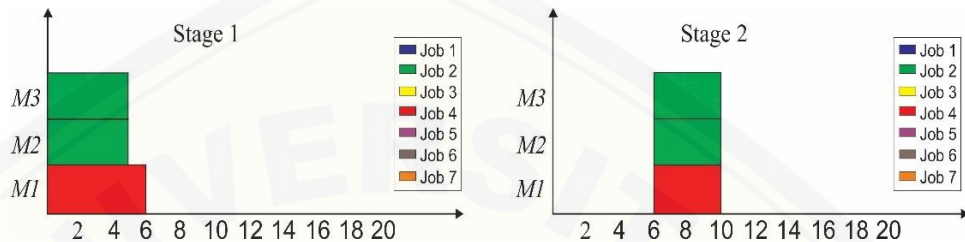
2.5	Diagram Gantt	7
2.6	Algoritma Tabu Search	8
2.7	Algoritma Nawaz-Enscore-Ham (NEH)	9
2.8	Algoritma Hybrid Tabu Search	10
BAB 3.	METODE PENELITIAN	13
3.1	Data Penelitian	13
3.2	Langkah-langkah Penelitian	13
BAB 4.	HASIL DAN PEMBAHASAN	15
4.1	Hasil Penelitian.....	15
4.1.1	Langkah Perhitungan	15
4.1.2	Program	25
4.1.3	Hasil Percobaan.....	28
4.2	Pembahasan	30
BAB 5.	PENUTUP.....	33
5.1	Kesimpulan	33
5.2	Saran	33
DAFTAR PUSTAKA		34
LAMPIRAN.....		36

DAFTAR TABEL

	Halaman
2.1 Contoh data <i>Hybrid Flowshop</i>	7
4.1 Data simulasi, (a) Data <i>stage</i> , (b) Data <i>job</i>	15
4.2 Total waktu proses	16
4.3 Total Waktu proses terurut.....	16
4.4 Urutan parsial 6 <i>job</i>	21
4.5 Urutan parsial 7 <i>job</i>	22
4.6 Pengaruh parameter replika.....	28
4.7 Pengaruh parameter <i>threshold</i>	29
4.8 Rangkuman hasil percobaan	30

- 3) Buat calon urutan parsial baru dari dua *job* dimulai dari *job* yang memiliki waktu proses paling lama, yaitu $J_4 - J_2$ dan $J_2 - J_4$ kemudian hitung *makespan* masing-masing.

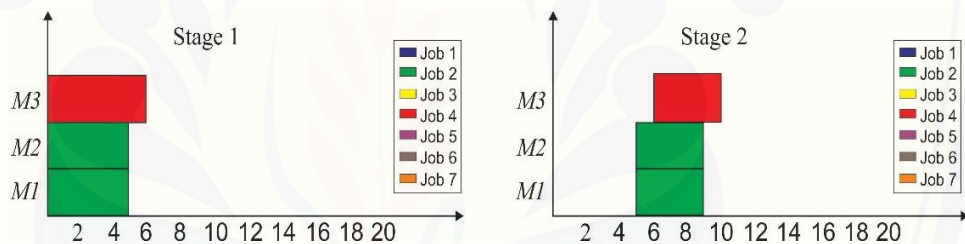
a) Menghitung *makespan* pada calon urutan parsial baru $J_4 - J_2$.



Gambar 4.1 Diagram *Gantt* urutan *job* 4-2

Dihasilkan nilai *makespan* = 10.

b) Menghitung *makespan* pada calon urutan parsial baru $J_2 - J_4$.

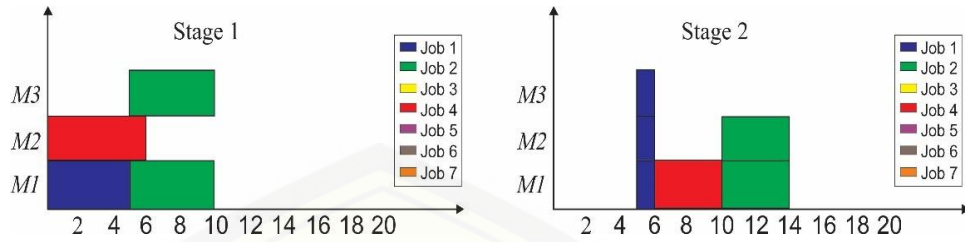


Gambar 4.2 Diagram *Gantt* urutan *job* 2-4

Dihasilkan nilai *makespan* = 10.

- 4) Pilih calon urutan parsial baru pada langkah (3) yang memiliki nilai *makespan* terkecil dan yang berada di urutan pertama, yaitu $J_4 - J_2$ dengan nilai *makespan* 10. Kemudian urutan parsial terpilih digunakan untuk dikombinasikan dengan *job* yang berada pada urutan ketiga yaitu J_1 pada Tabel 4.3. Calon urutan parsial baru tersebut adalah $J_1 - J_4 - J_2$, $J_4 - J_1 - J_2$ dan $J_4 - J_2 - J_1$, kemudian hitung *makespan* setiap calon urutan parsial baru.

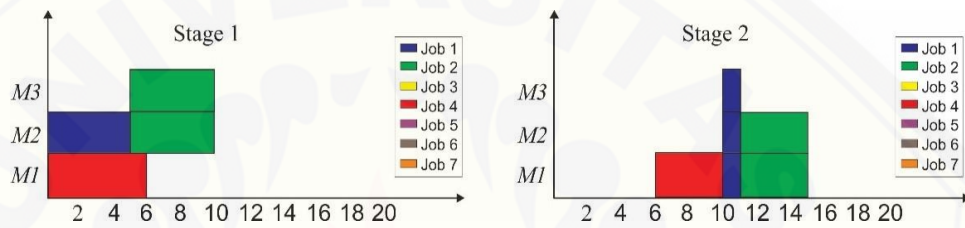
a) Menghitung *makespan* pada calon urutan parsial baru $J_1 - J_4 - J_2$.



Gambar 4.3 Diagram *Gantt* urutan *job* 1-4-2

Dihasilkan nilai *makespan* = 14.

- b) Menghitung *makespan* pada calon urutan parsial baru $J_4 - J_1 - J_2$.



Gambar 4.4 Diagram *Gantt* urutan *job* 4-1-2

Dihasilkan nilai *makespan* = 15.

- c) Menghitung *makespan* pada calon urutan parsial baru $J_4 - J_2 - J_1$.

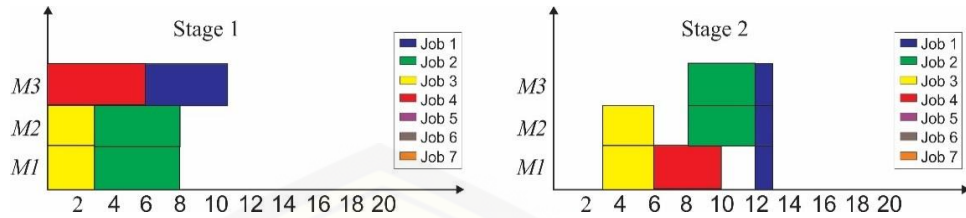


Gambar 4.5 Diagram *Gantt* urutan *job* 4-2-1

Dihasilkan nilai *makespan* = 11.

- 5) Pilih calon urutan parsial baru pada langkah (4) yang memiliki nilai *makespan* terkecil dan yang berada di urutan pertama, yaitu $J_4 - J_2 - J_1$ dengan nilai *makespan* 11. Kemudian urutan parsial terpilih digunakan untuk dikombinasikan dengan *job* yang berada pada urutan keempat yaitu J_3 pada Tabel 4.3. Calon urutan parsial baru tersebut adalah $J_3 - J_4 - J_2 - J_1$, $J_4 - J_3 - J_2 - J_1$, $J_4 - J_2 - J_3 - J_1$ dan $J_4 - J_2 - J_1 - J_3$, kemudian hitung *makespan* setiap calon urutan parsial baru.

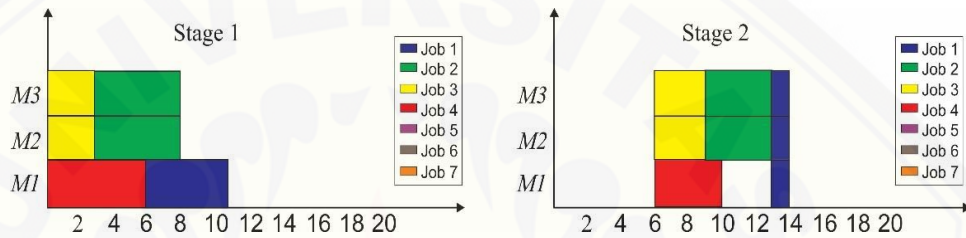
- a) Menghitung *makespan* pada calon urutan parsial baru $J_3 - J_4 - J_2 - J_1$.



Gambar 4.6 Diagram *Gantt* urutan *job* 3-4-2-1

Dihasilkan nilai *makespan* = 13.

b) Menghitung *makespan* pada calon urutan parsial baru $J_4 - J_3 - J_2 - J_1$.



Gambar 4.7 Diagram *Gantt* urutan *job* 4-3-2-1

Dihasilkan nilai *makespan* = 14.

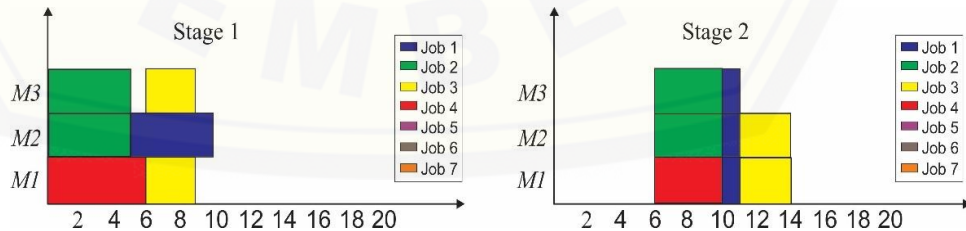
c) Menghitung *makespan* pada calon urutan parsial baru $J_4 - J_2 - J_3 - J_1$.



Gambar 4.8 Diagram *Gantt* urutan *job* 4-2-3-1

Dihasilkan nilai *makespan* = 14.

d) Menghitung *makespan* pada calon urutan parsial baru $J_4 - J_2 - J_1 - J_3$.

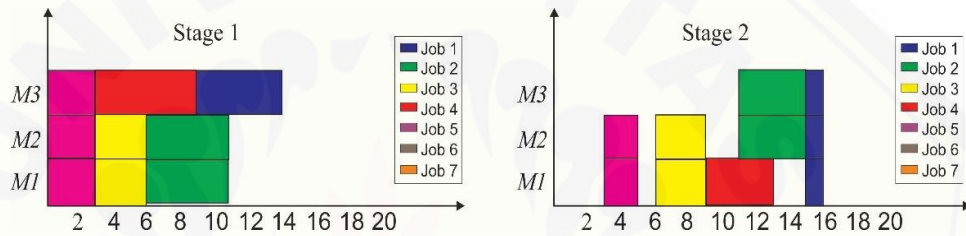


Gambar 4.9 Diagram *Gantt* urutan *job* 4-2-1-3

Dihasilkan nilai *makespan* = 14.

- 6) Pilih calon urutan parsial baru pada langkah (5) yang memiliki nilai *makespan* terkecil dan yang berada di urutan pertama, yaitu $J_3 - J_4 - J_2 - J_1$ dengan nilai *makespan* 13. Kemudian urutan parsial terpilih digunakan untuk dikombinasikan dengan *job* yang berada pada urutan kelima yaitu J_5 pada Tabel 4.3. Calon urutan parsial baru tersebut adalah $J_5 - J_3 - J_4 - J_2 - J_1$, $J_3 - J_5 - J_4 - J_2 - J_1$, $J_3 - J_4 - J_5 - J_2 - J_1$, $J_3 - J_4 - J_2 - J_5 - J_1$ dan $J_3 - J_4 - J_2 - J_1 - J_5$, kemudian hitung *makespan* setiap calon urutan parsial baru.

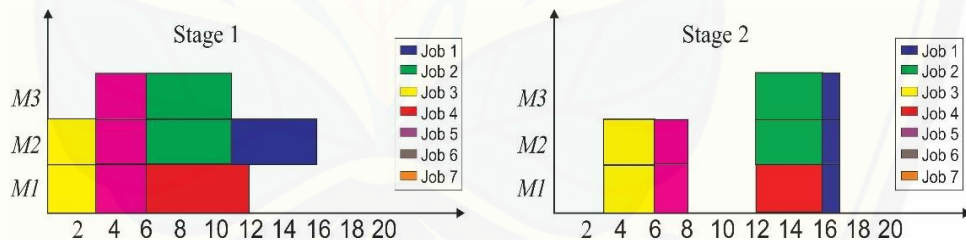
- a) Menghitung *makespan* pada calon urutan parsial baru $J_5 - J_3 - J_4 - J_2 - J_1$.



Gambar 4.10 Diagram Gantt urutan $job\ 5-3-4-2-1$

Dihasilkan nilai *makespan* = 16.

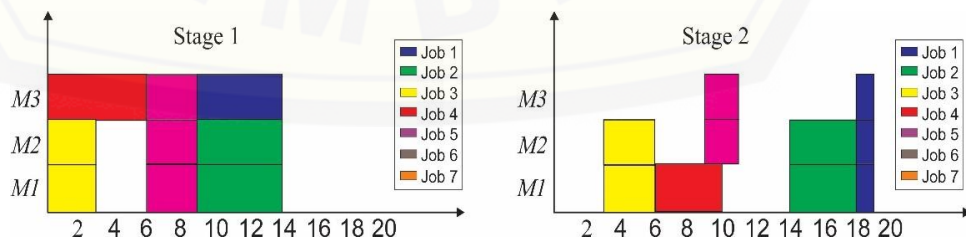
- b) Menghitung *makespan* pada calon urutan parsial baru $J_3 - J_5 - J_4 - J_2 - J_1$.



Gambar 4.11 Diagram Gantt urutan $job\ 3-5-4-2-1$

Dihasilkan nilai *makespan* = 17.

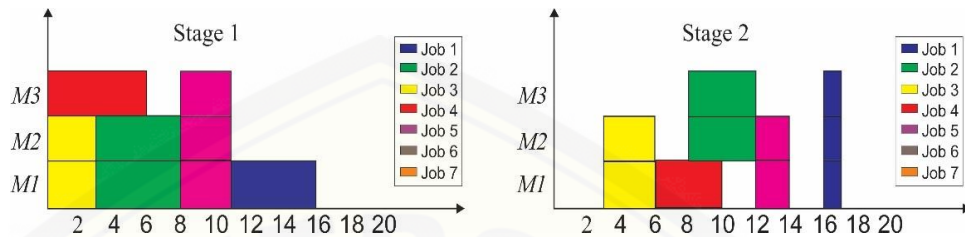
- c) Menghitung *makespan* pada calon urutan parsial baru $J_3 - J_4 - J_5 - J_2 - J_1$



Gambar 4.12 Diagram Gantt urutan $job\ 3-4-5-2-1$

Dihasilkan nilai *makespan* = 19.

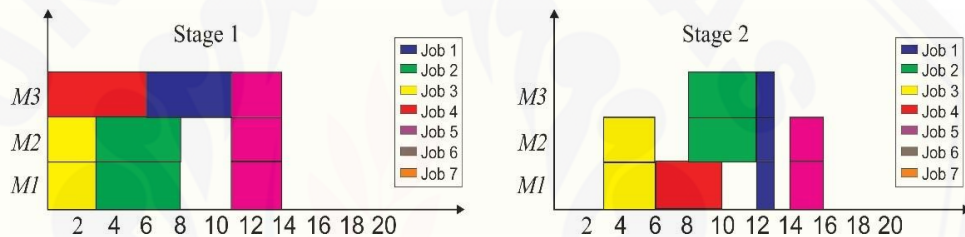
d) Menghitung *makespan* pada calon urutan parsial baru $J_3 - J_4 - J_2 - J_5 - J_1$.



Gambar 4.13 Diagram *Gantt* urutan *job* 3-4-2-5-1

Dihasilkan nilai *makespan* = 17.

e) Menghitung *makespan* pada calon urutan parsial baru $J_3 - J_4 - J_2 - J_1 - J_5$.



Gambar 4.14 Diagram *Gantt* urutan *job* 3-4-2-1-5

Dihasilkan nilai *makespan* = 16.

7) Pilih calon urutan parsial baru pada langkah (6) yang memiliki nilai *makespan* terkecil dan yang berada di urutan pertama, yaitu $J_5 - J_3 - J_4 - J_2 - J_1$ dengan nilai *makespan* 16. Kemudian urutan parsial terpilih digunakan untuk dikombinasikan dengan *job* yang berada pada urutan keenam yaitu J_6 pada Tabel 4.3. Kemudian hitung *makespan* setiap calon urutan parsial baru:

Tabel 4.4 Urutan parsial 6 *job*

No	Urutan Parsial	<i>Makespan</i>
1	$J_6 - J_5 - J_3 - J_4 - J_2 - J_1$,	18
2	$J_5 - J_6 - J_3 - J_4 - J_2 - J_1$,	18
3	$J_5 - J_3 - J_6 - J_4 - J_2 - J_1$,	18
4	$J_5 - J_3 - J_4 - J_6 - J_2 - J_1$,	19
5	$J_5 - J_3 - J_4 - J_2 - J_6 - J_1$,	17
6	$J_5 - J_3 - J_4 - J_2 - J_1 - J_6$,	18

8) Pilih calon urutan parsial baru pada langkah (7) yang memiliki nilai *makespan* terkecil dan yang berada di urutan pertama, yaitu $J_5 - J_3 - J_4 - J_2 - J_6 - J_1$

dengan nilai *makespan* 17. Kemudian urutan parsial terpilih digunakan untuk dikombinasikan dengan *job* yang berada pada urutan ketujuh yaitu J_7 pada Tabel 4.3. Kemudian hitung *makespan* setiap calon urutan parsial baru:

Tabel 4.5 Urutan parsial 7 *job*

No	Urutan Parsial	Makespan
1	$J_7 - J_5 - J_3 - J_4 - J_2 - J_6 - J_1$,	19
2	$J_5 - J_7 - J_3 - J_4 - J_2 - J_6 - J_1$,	18
3	$J_5 - J_3 - J_7 - J_4 - J_2 - J_6 - J_1$,	18
4	$J_5 - J_3 - J_4 - J_7 - J_2 - J_6 - J_1$,	20
5	$J_5 - J_3 - J_4 - J_2 - J_7 - J_6 - J_1$,	20
6	$J_5 - J_3 - J_4 - J_2 - J_6 - J_7 - J_1$,	19
7	$J_5 - J_3 - J_4 - J_2 - J_6 - J_1 - J_7$,	19

- 9) Pilih calon urutan parsial baru pada langkah (8) yang memiliki nilai *makespan* terkecil dan yang berada di urutan pertama, yaitu $J_5 - J_7 - J_3 - J_4 - J_2 - J_6 - J_1$ dengan nilai *makespan* 18. Karena semua *job* telah masuk dalam urutan parsial, maka urutan parsial tersebut yang terpilih sebagai solusi awal.

b. Pencarian *neighborhood*

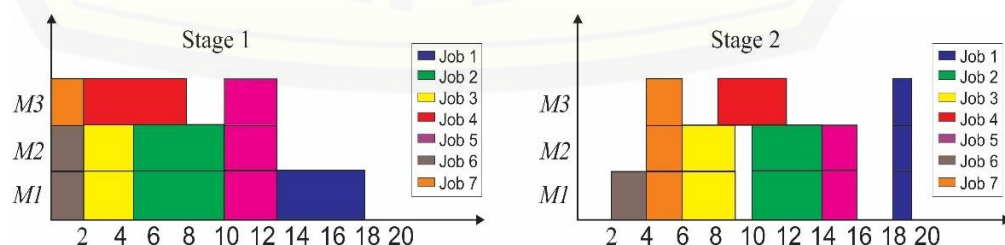
Solusi yang dihasilkan algoritma NEH digunakan sebagai solusi awal pencarian *neighborhood*, digunakan sebagai solusi *best-so-far* dan disimpan dalam *tabu list*. Pencarian *neighborhood* dilakukan sebanyak replika kali dan dengan cara *swap*.

1) Replika 1

Dibangkitkan 2 angka secara random sebagai posisi *job* dalam struktur solusi.

posisi1 = 1 dan posisi2 = 6

Urutan <i>Job</i> Awal	5	7	3	4	2	6	1
Urutan <i>Job</i> Baru	6	7	3	4	2	5	1



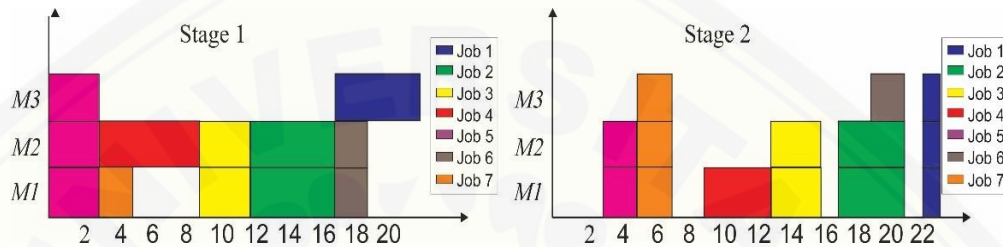
Gambar 4.15 Diagram *Gantt* urutan 6-7-3-4-2-5-1

Didapatkan nilai *makespan* = 19.

2) Replika 2

posisi1 = 3 dan posisi2 = 4

Urutan <i>Job</i> Awal	5	7	3	4	2	6	1
Urutan <i>Job</i> Baru	5	7	4	3	2	6	1



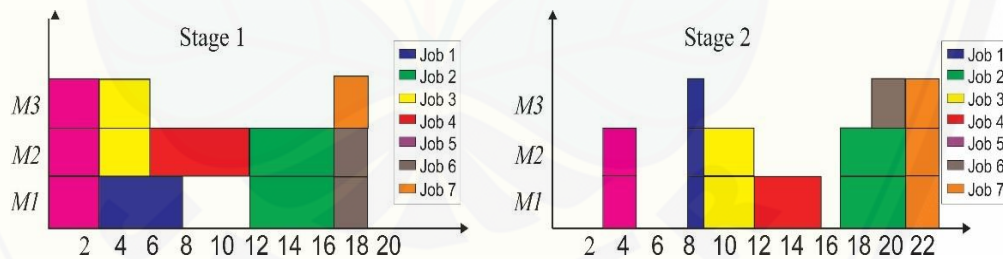
Gambar 4.16 Diagram *Gantt* urutan 5-7-4-3-2-6-1

Didapatkan nilai *makespan* = 23.

3) Replika 3

posisi1 = 2 dan posisi2 = 7

Urutan <i>Job</i> Awal	5	7	3	4	2	6	1
Urutan <i>Job</i> Baru	5	1	3	4	2	6	7



Gambar 4.17 Diagram *Gantt* urutan 5-1-3-4-2-6-7

Didapatkan nilai *makespan* = 23.

c. Pemilihan solusi terbaik

Dari ketiga replika dipilih solusi dengan nilai *makespan* terkecil, yaitu replika 1 dengan nilai *makespan* = 19.

d. Penyimpanan dalam *tabu list*

Solusi terbaik yang terpilih dalam pencarian *neighborhood* langsung disimpan dalam *tabu list* jika *tabu list* belum penuh. Karena ukuran *tabu list* = 1 dan telah terisi oleh solusi awal. Maka solusi terbaik yang terpilih dibandingkan dengan solusi dalam *tabu list* tersebut. Jika solusi baru lebih baik maka akan menggantikan. Nilai *makespan* solusi baru = 19 dan nilai *makespan* solusi dalam *tabu list* = 18, yang artinya solusi baru lebih buruk, sehingga solusi dalam *tabu list* tetap dipertahankan.

e. Penyimpanan solusi *best-so-far*

Jika solusi terbaik dalam *neighborhood search* lebih baik dari solusi *best-so-far* saat ini, maka menggantikan solusi *best-so-far* saat ini dan mereset jumlah *non improvement* ke nilai nol. Namun, jika solusi terbaik dalam *neighborhood search* lebih buruk atau sama dengan solusi *best-so-far* saat ini, maka nilai *non improvement* akan ditambah satu. Karena nilai *makespan* solusi *neighborhood search* lebih besar dari nilai *makespan* solusi *best-so-far*, yang artinya lebih buruk, maka solusi *best-so-far* tetap dan nilai *non improvement* ditambah satu. Nilai *non improvement* = 1.

f. Metode HTS

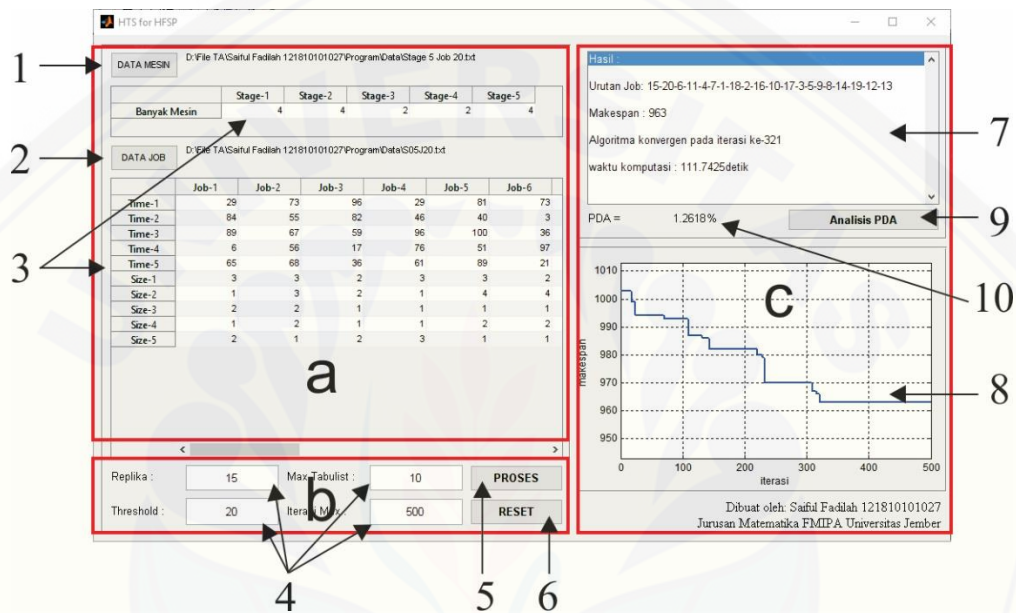
Pada langkah ini nilai *non improvement* dicek apakah melebihi nilai *threshold*. Jika telah melebihi maka solusi *best-so-far* digantikan dengan solusi hasil algoritma NEH dan nilai *non improvement* direset menjadi nol (0). Karena nilai *non improvement* = 1 yang artinya kurang dari nilai *threshold*, maka langsung dilanjutkan langkah berikutnya.

g. Pengecekan iterasi maksimal

Jika iterasi telah mencapai iterasi maksimal, maka proses dihentikan dan solusi terbaik dalam *tabu list* menjadi solusi akhir. Karena iterasi = 1 dan belum mencapai iterasi maksimal, maka proses dilakukan kembali mulai langkah b (pencarian *neighborhood*).

4.1.2 Program

Pada skripsi ini dihasilkan sebuah program penyelesaian *Hybrid Flowshop* yang dibuat menggunakan *software* MATLAB. Tujuan dari pembuatan program ini adalah untuk menyelesaikan permasalahan *Hybrid Flowshop* menggunakan algoritma *Hybrid Tabu Search*. Program ini memiliki tampilan utama pada Gambar



Gambar 4.18 Tampilan program *Hybrid Flowshop*

Tampilan dari program ini mempunyai beberapa komponen sebagai berikut:

a. Input

- 1) DATA MESIN untuk membuka *file* jumlah mesin pada setiap *stage*.
- 2) DATA JOB untuk membuka *file* data *job* yang berisi *time* dan *size*.
- 3) Table sebagai tempat untuk menampilkan data mesin dan data *job*.

b. Proses

- 1) Edit Text digunakan untuk memasukkan parameter algoritma antara lain: Replika, Threshold, Max Tabulist dan Iterasi Max.
- 2) PROSES untuk memulai penyelesaian data dengan algoritma *Hybrid Tabu Search*.
- 3) RESET untuk mengembalikan tampilan ke tampilan awal.

c. Hasil

- 1) List Box untuk menampilkan hasil dari algoritma antara lain: urutan job, makespan, kekonvergenan dan waktu komputasi.
- 2) Pushbutton Analisis PDA untuk menghitung *Lower Bound* dari data kemudian menghitung *Percentage Deviation Algorithm* (PDA).
- 3) Static Text sebagai tempat untuk menampilkan hasil perhitungan PDA.
- 4) Axes untuk menampilkan grafik kekonvergenan algoritma.

Penyelesaian penjadwalan *Hybrid Flowshop Scheduling* dengan algoritma *Hybrid Tabu Search* diawali dengan membangkitkan solusi awal dengan algoritma NEH. Selanjutnya inisialisasi parameter replika, *threshold*, max tabulist dan iterasi max dan menjalankan program sesuai proses *Hybrid Flowshop*. Prosedur program secara ringkas ditunjukkan sebagai berikut:

Prosedur Algoritma *Hybrid Tabu Search*

```

input data mesin, data job;
inisialisasi parameter;
ss=size(data,1);
time=data(1:ss/2,:);
Size=data(ss/2+1:ss,:);
[stage,job]=size(Size);

bangkitkan solusi awal dengan algoritma NEH;
simpan solusi awal dalam tabulist
hybrid=0;
for iter=1:replika
    for rep=1:replika
        pencarian ketetanggan
        hitung nilai makespan ketetanggan
    end
    pilih tetangga terbaik=minimum makespan
    if length tabulist<max tabulist
        tabulist(length(tabulist)+1)=tetangga terbaik;
    else
        pilih tabulist terburuk
        if tabulist terburuk>tetangga terbaik
            tabulist terburuk=tetangga terbaik
        else
            hybrid=hybrid+1;
            if hybrid=maxhybrid
                flow=solusi awal;
                hybrid=0;
            end
        end
    end

```

```

    end
  end
end
pemilihan solusi akhir=solusi terbaik dalam tabulist;

```

a. Membangkitkan solusi awal

Untuk membangkitkan solusi awal dalam algoritma *Hybrid Tabu Search* digunakan algoritma NEH. Langkah awal dari algoritma ini adalah mengurutkan masing-masing *job* menurut total waktu. Kemudian dicari calon urutan parsial *job* yang akan digunakan sebagai solusi awal. *Makespan* terkecil akan ditetapkan sebagai solusi awal. Prosedur algoritma NEH sebagai berikut:

Prosedur Algoritma NEH

```

input: Size(stage×job), time(stage×job);
output: solusi awal(1×job), makespan;
proses: job=ukuran kolom matriks time;
for i=1:job
    jumlahkan waktu setiap job (waktu)
end
urutkan waktu job "descending"=[waktu_urut,indeks]
sol=indeks(1)
for j=1:job-1
    for i=1:j+1
        solusi(i,:)= [sol(1:i-1) indeks(length(sol)+1)
                     sol(i:length(sol))];
        hitung nilai makespan
    end
    urutkan makespan=[makespan_urut,idx]
    sol=solusi(idx,:);
    pilih makespan terkecil
end
solusi awal=sol

```

b. Inisialisasi parameter

Parameter-parameter yang digunakan pada penyelesaian penjadwalan *Hybrid Flowshop Scheduling* dengan algoritma *Hybrid Tabu Search* adalah replika, *threshold*, max tabulist dan iterasi max.

c. Pencarian ketetangaan

Solusi dari algoritma NEH akan digunakan sebagai solusi awal. Solusi awal tersebut akan dicari ketetanggaannya dengan cara *swap* sebanyak replika kali. Prosedur pencarian ketetangaan sebagai berikut:

Prosedur Pencarian Ketetangaan

```
posisi_swap=[(random(1,2)×job)];
new_flow(rep,:) = flow;
new_flow(rep, posisi_swap) = flow([posisi_swap(2) posisi_swap(1)]);
```

d. Menghitung *makespan*

Proses selanjutnya adalah menghitung *makespan* dari urutan job. Urutan job yang terbentuk dari algoritma NEH maupun urutan job dari proses *swap* akan diproses. Tujuan dari langkah ini untuk menghitung total waktu yang dibutuhkan untuk menyelesaikan semua *job* pada semua *stage* sesuai urutan.

4.1.3 Hasil Percobaan

Program penerapan algoritma *Hybrid Tabu Search* yang telah dibuat diujikan pada data yang telah dikumpulkan. Pada penelitian ini, terlebih dahulu dicari pengaruh nilai parameter algoritma sebelum diterapkan pada semua data. Parameter yang diuji adalah parameter replika dan *threshold*.

a. Pengaruh parameter replika

Untuk mengetahui pengaruh parameter replika digunakan replika yang berbeda-beda dan nilai parameter lain tetap. *Threshold* = 20; Iterasi Maks = 500.

Tabel 4.6 Pengaruh parameter replika

Replika	<i>Makespan</i>	Iterasi Konvergen	Waktu Komputasi (detik)	Rata-rata <i>Makespan</i>
3	1011	104	19,6935	1002,6667
	1000	144	19,5166	
	997	215	19,2490	
4	999	329	21,6437	996,6667
	994	297	22,1531	
	997	311	22,3733	
5	995	290	23,9682	994,3333
	994	197	23,9953	
	994	149	23,9609	
6	994	95	25,2107	992,3333
	994	128	25,4483	
	989	158	25,1243	
8	990	71	31,5270	989,3333
	990	40	31,0653	
	988	106	31,1002	

Replika	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	Rata-rata Makespan
10	987	157	35,0030	986
	984	40	34,0337	
	987	212	34,8337	
12	984	108	39,2650	980,3333
	981	146	38,5062	
	976	86	39,2126	
15	974	151	47,4971	971,3333
	969	150	46,4639	
	971	302	47,1861	

Berdasarkan hasil penelitian pengaruh parameter replika (lihat Tabel 4.6), nilai replika yang menghasilkan rata-rata *makespan* paling minimum adalah 15.

b. Pengaruh parameter *threshold*

Untuk mengetahui pengaruh parameter replika digunakan replika yang berbeda-beda dan nilai parameter lain tetap. Replika = 15; Iterasi Maks = 500.

Tabel 4.7 Pengaruh parameter *threshold*

Threshold	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	Rata-rata Makespan
5	994	4	47,2718	996,3333
	1000	1	47,1907	
	995	67	47,1273	
10	997	77	46,4664	994,6667
	994	65	47,0102	
	993	25	46,8121	
15	994	78	48,7291	990,6667
	993	43	46,8751	
	985	25	47,0187	
20	976	68	46,6875	973
	974	232	46,6414	
	969	150	46,9004	
25	979	120	46,8162	977
	976	58	51,4622	
	976	7	48,6628	
30	979	56	48,2973	984,3333
	985	66	50,1380	
	989	81	50,2430	

Berdasarkan hasil penelitian pengaruh parameter *threshold* (lihat Tabel 4.7), nilai replika yang menghasilkan rata-rata *makespan* paling minimum adalah 20.

c. Rangkuman hasil percobaan

Setelah didapatkan nilai parameter paling optimal, selanjutnya nilai parameter tersebut digunakan untuk menyelesaikan semua data yang pada setiap iterasi dilakukan sebanyak lima kali. Hasil penelitian lengkap dapat dilihat pada Lampiran B. Rangkuman dari semua hasil penelitian dapat dilihat pada Tabel 4.8.

Tabel 4.8 Rangkuman hasil percobaan

Data	Iterasi	Minimum Makespan	Rata-rata Makespan	Rata-rata Iterasi Konvergen	Rata-rata Waktu Komputasi (detik)	APD _A (%)
2 Stage	500	306	306	1	15,6922	8,8968
5 Job	1000	306	306	1	32,0733	8,8968
2 Stage	500	807	807	1	22,3221	0
20 Job	1000	807	807	1	46,9455	0
2 Stage	500	2593	2593	1	67,7447	0
50 Job	1000	2593	2593	1	145,2051	0
5 Stage	500	451	451	1	15,6092	4,8837
5 Job	1000	451	451	1	33,1004	4,8837
5 Stage	500	974	979,4	362	53,6409	2,9863
20 Job	1000	969	975,4	546	95,0864	2,5657
5 Stage	500	2630	2630	1	52,8193	0,9985
50 Job	1000	2630	2630	1	101,4533	0,9985
8 Stage	500	644	644	1	15,6107	11,8056
5 Job	1000	644	644	1	31,2296	11,8056
8 Stage	500	1493	1557	10	39,5389	4,2866
20 Job	1000	1493	1551	15	78,6898	3,8848
8 Stage	500	2661	2661	94	107,0182	0,8719
50 Job	1000	2653	2658,6	376	204,1911	0,7809

4.2 Pembahasan

Algoritma *Hybrid Tabu Search* dipengaruhi oleh parameter replika dan *threshold*. Pada parameter replika sangat berpengaruh pada hasil *makespan* dan waktu komputasi (lihat Tabel 4.6). Nilai replika yang semakin besar dapat menghasilkan solusi yang lebih optimal yang artinya nilai *makespan* lebih minimum. Hal ini dikarenakan algoritma menyediakan kemungkinan solusi yang lebih banyak sehingga mencakup semua ketetangaan. Namun, waktu komputasi yang dibutuhkan oleh

algoritma menjadi lebih lama karena algoritma harus melakukan pencarian ketetangaan dan perhitungan *makespan* lebih banyak pula.

Hasil percobaan yang dilakukan untuk mengetahui pengaruh parameter *threshold* dapat dilihat pada Tabel 4.7. Nilai *threshold* yang menghasilkan *makespan* paling minimum adalah 20. Nilai *threshold* yang lebih kecil dan lebih besar dari 20 menghasilkan *makespan* yang kurang minimum. Nilai *threshold* yang kecil menyebabkan algoritma tidak dapat mengeksplorasi ketetangaan, karena ketika tidak dapat menemukan solusi yang lebih baik, solusi kembali pada hasil algoritma NEH. Sedangkan nilai *threshold* yang besar menyebabkan algoritma mudah terjebak lokal optimum, karena ketika algoritma terjebak tidak segera diperbarui solusinya.

Setelah itu program diuji untuk menyelesaikan 9 data *Hybrid Flowshop* dengan menggunakan parameter terbaik yang telah diujikan pada salah satu data untuk mencari parameter terbaik yaitu replika = 15 dan *threshold* = 20. Setiap data diuji sebanyak lima kali dengan 500 iterasi dan 1000 iterasi. Rangkuman hasil percobaan yang telah dilakukan dapat dilihat pada Tabel 4.8. Berdasarkan hasil tersebut semakin banyak iterasi yang digunakan, memungkinkan algoritma untuk menemukan solusi yang lebih optimal. Hasil yang didapatkan dari percobaan sebagai berikut:

- a. Data 2 Stage 5 Job, nilai *makespan* paling minimum adalah 306 dan rata-rata *makespan* adalah 306.
- b. Data 2 Stage 20 Job, nilai *makespan* paling minimum adalah 807 dan rata-rata *makespan* adalah 807.
- c. Data 2 Stage 50 Job, nilai *makespan* paling minimum adalah 2593 dan rata-rata *makespan* adalah 2593.
- d. Data 5 Stage 5 Job, nilai *makespan* paling minimum adalah 451 dan rata-rata *makespan* adalah 451.
- e. Data 5 Stage 20 Job, nilai *makespan* paling minimum adalah 969 dan rata-rata *makespan* adalah 977,4.

- f. Data 5 *Stage 50 Job*, nilai *makespan* paling minimum adalah 2630 dan rata-rata *makespan* adalah 2630.
- g. Data 8 *Stage 5 Job*, nilai *makespan* paling minimum adalah 644 dan rata-rata *makespan* adalah 644.
- h. Data 8 *Stage 20 Job*, nilai *makespan* paling minimum adalah 1493 dan rata-rata *makespan* adalah 1554.
- i. Data 8 *Stage 50 Job*, nilai *makespan* paling minimum adalah 2653 dan rata-rata *makespan* adalah 2659,8.

Tingkat kecepatan kekonvergenan algoritma *Hybrid Tabu Search* tidak dapat dipastikan karena terdapat proses random pada langkah pencarian ketetangaan (*swap*). Sedangkan waktu komputasi rata-rata yang dibutuhkan untuk menyelesaikan 5 *job* adalah 15,6374 detik (500 iterasi) dan 32,1344 detik (1000 iterasi). Waktu komputasi rata-rata yang dibutuhkan untuk menyelesaikan 10 *job* adalah 38,5006 detik (500 iterasi) dan 73,5739 detik (1000 iterasi). Waktu komputasi rata-rata yang dibutuhkan untuk menyelesaikan 50 *job* adalah 75,8607 detik (500 iterasi) dan 150,2858 detik (1000 iterasi).

Berdasarkan hasil percobaan, nilai APD_A pada data 2 *Stage 20 Job* dan 2 *Stage 50 Job* adalah 0 %, yang artinya nilai *makespan* yang dihasilkan merupakan nilai paling optimal. Sedangkan nilai APD_A terburuk pada hasil penyelesaian data 8 *Stage 5 Job* yaitu 11,8056 %. Menurut hasil ini, dapat dikatakan bahwa algoritma *Hybrid Tabu Search* sangat efektif karena nilai yang dihasilkan sama dengan atau mendekati nilai optimal.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan dapat ditarik kesimpulan beberapa hal sebagai berikut:

- a. Algoritma *Hybrid Tabu Search* dapat menyelesaikan permasalahan *Hybrid Flowshop Scheduling*. Solusi diperoleh dengan input berupa data waktu (*time*) dan *size*, data mesin, replika, *threshold*, Max tabulist dan iterasi. Algoritma *Hybrid Tabu Search* sangat efektif karena hasil yang diperoleh sama dengan atau mendekati nilai optimal berdasarkan *Average Percentage Deviation*.
- b. Berdasarkan yang penulis teliti tingkat kecepatan kekonvergenan tidak dapat dipastikan pada setiap kasus. Waktu komputasi yang dibutuhkan untuk menyelesaikan data 5 *job* pada 2, 5 dan 8 *stage* sekitar 15 sampai 32 detik, untuk data 20 *job* pada 2, 5 dan 8 *stage* sekitar 22 sampai 95 detik dan untuk data 50 *job* pada 2, 5 dan 8 *stage* sekitar 52 sampai 204 detik pada parameter iterasi 500 dan 1000.

5.2 Saran

Penelitian lebih lanjut dapat menerapkan algoritma *Hybrid Tabu Search* untuk menyelesaikan permasalahan *Multi-Objective Flowshop/Hybrid Flowshop*. Selain itu juga dapat menerapkan pada permasalahan optimasi lain seperti *Travelling Salesman Problem*, *Vehicle Routing Problem*, *Knapsack* dan sebagainya.

DAFTAR PUSTAKA

- Baker, K. 1997. *Introduction to Sequencing And Scheduling*. New York: John Wiley And Sons, Inc.
- Berlianty, I. dan Arifin, M. 2010. *Teknik-teknik Optimalisasi Heuristik*. Yogyakarta: Graha Ilmu.
- Chen, J. S., Pan, J. C. H. dan Wu, C. K. 2008. Hybrid Tabu Search for Re-Entrant Permutation Flow-Shop Scheduling Problem. *Expert Systems with Applications*. **34** (3): 1924-1930.
- Engin, O. dan Doyen, A. 2004. A New Approach to Solve Hybrid Flowshop Scheduling Problems by Artificial Immune System. *Future Generation Computer Systems*. **20**: 1083-1095.
- Ginting, R. 2009. *Penjadwalan Mesin*. Yogyakarta: Graha Ilmu.
- Glover, F. 1986. Future Path for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*. **13** (5): 533-549.
- Gupta, J. N. D. 1988. Two Stage, Hybrid Flowshop Scheduling Problem. *The Journal of The Operational Research Society*. **39** (4): 359-364.
- Hejazi, S. R. dan Saghafian, S. 2005. Flowshop-Scheduling Problems with Makespan Criterion: A Review. *International Journal of Production Research*. **43** (14): 2895-2929.
- Janiak, A., Kozan, E., Lichtenstein, M. dan Oguz, C. 2007. Metaheuristic Approaches to The Hybrid Flow Shop Scheduling Problem with A Cost-Related Criterion. *International Journal of Production Economics*. **105**: 407-424.
- Nowicki, E, dan C Smutnicki. 1996. A fast taboo search algorithm for jobshop problem. *Management Science*, **42** (6): 797-813.
- Nowicki, E, dan C Smutnicki. 1996. A fast taboo search algorithm for the permutation flowshop problem. *European Journal of Operation Research*, **91** (1): 160-175.
- Nowicki, E, dan C Smutnicki. 1998. Flowshop with parallel machines: A tabu search approach. *European Journal of Operation Research*, **106** (2-3): 226-253.

- Pinedo, M. 2002. *Scheduling Theory, Algorithms, and Systems*. 2nd. New Jersey: Prentice Hall.
- Pratiwi, F. R., Rahman, A. dan Tantrika, C. F. M. 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. dan Siswanto, N. 2014. *Penyelesaian Permasalahan Multi-Objective Hybrid Flow Shop Scheduling dengan Algoritma Modified Particle Swarm Optimization*. Surabaya: ITS.
- Santosa, B. dan 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.
- Serifoglu, F.S. dan Ulusoy, G. 2004. Multiprocessor Task Scheduling in Multistage Hybrid Flow Shop: A Genetic Algorithm Approach. *Journal of Operational Research Society* **55**: 504-512.
- Sitorus, H. M., Juwono, C. dan Dwikaragus, K. P. 2013. *Pengembangan Algoritma Viral Systems untuk Masalah Penjadwalan Hybrid Flow Shop untuk Meminimasi Makespan*. Surakarta: Universitas Muhammadiyah Surakarta.
- Tseng, C. dan Liao, C. 2008. A Particle Swarm Optimization Algorithm for Hybrid Flow-Shop Scheduling with Multiprocessor Tasks. *International Journal of Production Research*. **46** (17): 4655-4670.
- Uetake, T., Tsubone, H. dan Ohba, M. 1995. A Production Scheduling System in A Hybrid Flow Shop. *International Journal of Production Economics*. **41**: 395-398.
- Wang, S., Liu, M. dan 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.
- Ying, K. dan Lin, S. 2006. Multiprocessor Task Scheduling in Multi stage Hybrid Flowshops : An Ant Colony System Approach. *International Journal of Production Research*. **44** (16): 3161-3177.

LAMPIRAN

Lampiran A. Data Penelitian

A.1 Data *Hybrid Flowshop 2 Stage 5 Job*

Mesin: 3 4

Stage	Job				
	1	2	3	4	5
t_{1j}	60	29	78	38	37
$size_{1j}$	3	2	1	1	1
t_{2j}	81	30	100	87	24
$size_{2j}$	3	2	3	2	2

A.2 Data *Hybrid Flowshop 2 Stage 20 Job*

Mesin: 5 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}	48	53	3	66	9	95	45	21	57	20	34	40	86	67	100	87	8	63	40	88
$size_{1j}$	1	4	4	4	5	5	2	4	3	1	4	2	4	2	4	1	1	2	3	2
t_{2j}	73	87	71	85	32	84	5	5	65	94	36	28	37	53	56	70	21	2	28	15
$size_{2j}$	3	1	3	3	3	3	3	2	2	3	3	2	3	2	1	2	3	3	2	2

A.3 Data Hybrid Flowshop 2 Stage 50 Job

Mesin: 4 1

Stage	Job																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
t_{1j}	35	96	18	86	13	90	27	14	12	18	69	87	80	82	1	31	99	98	36	67	18	30	58	24	68
size _{1j}	1	1	4	1	1	4	1	2	3	2	4	3	1	2	1	3	4	3	3	1	4	3	4	4	1
t_{2j}	38	43	26	57	63	16	100	62	10	24	94	86	72	72	61	24	96	49	4	38	18	35	94	68	61
size _{2j}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Stage	Job																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
t_{1j}	66	82	49	53	12	35	28	52	40	29	16	11	38	45	78	20	28	96	89	96	87	92	49	39	99
size _{1j}	1	4	2	3	3	4	2	1	2	1	4	4	1	3	1	2	1	3	3	3	3	3	3	4	3
t_{2j}	43	47	87	73	100	4	97	14	7	32	54	22	65	61	49	26	78	89	94	77	17	12	29	9	95
size _{2j}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

A.4 Data Hybrid Flowshop 5 Stage 5 Job

Mesin: 5 1 1 2 4

Stage	Job				
	1	2	3	4	5
t_{1j}	33	37	79	69	42
size _{1j}	5	5	3	4	4
t_{2j}	27	20	73	28	4
size _{2j}	1	1	1	1	1
t_{3j}	98	30	54	6	15
size _{3j}	1	1	1	1	1
t_{4j}	18	50	35	22	60
size _{4j}	2	1	2	1	2
t_{5j}	86	83	58	76	64
size _{5j}	3	4	1	3	4

A.5 Data Hybrid Flowshop 2 Stage 20 Job

Mesin: 4 4 2 2 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}	29	73	96	29	81	73	81	81	95	10	29	73	80	61	15	36	47	51	20	72
size $_{1j}$	3	3	2	3	3	2	1	2	2	1	4	4	3	3	2	3	2	1	1	2
t_{2j}	84	55	82	46	40	3	2	30	67	91	28	44	54	84	29	24	69	38	84	22
size $_{2j}$	1	3	2	1	4	4	2	3	1	4	2	1	3	4	1	1	2	1	4	3
t_{3j}	89	67	59	96	100	36	81	4	84	80	48	79	71	54	97	16	36	19	61	82
size $_{3j}$	2	2	1	1	1	1	1	2	2	1	1	1	1	2	2	2	1	2	2	1
t_{4j}	6	56	17	76	51	97	46	80	43	33	47	1	31	11	77	77	11	53	14	5
size $_{4j}$	1	2	1	1	2	2	1	2	1	1	1	2	1	2	2	1	2	1	1	1
t_{5j}	65	68	36	61	89	21	44	34	61	33	61	47	1	77	28	79	55	11	35	92
size $_{5j}$	2	1	2	3	1	1	3	3	3	2	4	1	1	1	2	2	3	2	1	3

A.6 Data Hybrid Flowshop 5 Stage 50 Job

Mesin: 3 1 1 3 5

Stage	Job																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
t_{1j}	16	38	38	14	32	45	9	56	69	84	95	46	25	25	30	84	88	15	35	12	22	85	79	30	25
size $_{1j}$	3	1	3	1	2	3	2	2	3	2	3	2	1	2	2	3	3	2	2	3	3	2	1	1	3
t_{2j}	81	84	89	4	67	67	40	81	94	40	92	82	37	40	90	42	10	33	86	16	91	13	8	26	3
size $_{2j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{3j}	16	96	92	21	56	74	32	49	95	39	73	9	74	77	54	77	6	27	34	19	41	16	72	42	1
size $_{3j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{4j}	38	36	93	22	88	12	51	81	78	50	85	48	67	59	77	32	6	91	28	76	98	45	18	100	72
size $_{4j}$	3	1	1	2	3	3	1	1	3	1	1	2	2	3	3	1	1	3	3	3	3	3	1	1	2

Stage	Job																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
t_{1j}	2	56	18	65	99	59	52	47	25	91	28	76	99	62	34	94	18	6	91	2	67	27	40	47	100
$size_{1j}$	2	5	3	2	2	3	1	2	5	1	5	1	4	3	3	4	1	1	1	1	1	3	1	2	1

Stage	Job																													
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50					
t_{1j}	86	13	56	30	92	67	58	30	28	21	42	22	49	30	50	73	11	5	60	49	24	1	38	35	44					
$size_{1j}$	3	1	1	2	3	2	1	3	1	3	1	2	1	2	1	2	3	3	3	3	3	2	2	1	3					
t_{2j}	45	92	72	39	21	20	60	52	20	70	53	99	60	38	22	69	4	13	91	83	96	40	23	34	41					
$size_{2j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
t_{3j}	12	39	71	100	11	75	8	64	72	23	89	33	72	18	35	16	82	12	7	97	52	26	82	63	5					
$size_{3j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
t_{4j}	66	100	40	83	59	20	20	27	2	14	39	53	26	79	93	25	44	25	66	69	11	54	88	96	19					
$size_{4j}$	3	3	3	3	3	1	3	2	1	3	3	3	2	2	3	3	2	3	3	3	2	3	3	1	2					
t_{5j}	41	20	93	30	30	66	75	46	40	67	43	27	30	9	1	52	74	88	18	93	36	59	6	9	97					
$size_{5j}$	5	1	4	4	3	2	1	5	4	5	2	2	1	4	5	2	5	3	4	1	4	2	2	3	3					

A.7 Data Hybrid Flowshop 8 Stage 5 Job

Mesin: 4 5 5 4 3 4 1 2

Stage	Job				
	1	2	3	4	5
t_{1j}	12	70	23	10	32
$size_{1j}$	4	4	1	1	1
t_{2j}	21	33	28	59	33
$size_{2j}$	1	4	2	5	3
t_{3j}	17	87	30	49	54
$size_{3j}$	3	3	3	1	5

<i>Stage</i>	<i>Job</i>				
	1	2	3	4	5
<i>t_{4j}</i>	2	15	86	77	28
<i>size_{4j}</i>	1	3	3	1	4
<i>t_{5j}</i>	42	2	41	43	56
<i>size_{5j}</i>	3	1	1	1	3
<i>t_{6j}</i>	3	66	77	99	48
<i>size_{6j}</i>	4	1	1	4	2
<i>t_{7j}</i>	49	85	9	90	49
<i>size_{7j}</i>	1	1	1	1	1
<i>t_{8j}</i>	6	96	65	47	50
<i>size_{8j}</i>	1	1	2	1	1

A.8 Data Hybrid Flowshop 8 Stage 20 Job

Mesin: 1 5 4 2 5 1 5 3

<i>Stage</i>	<i>Job</i>																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
<i>t_{1j}</i>	33	83	21	51	94	61	46	88	60	71	65	67	71	2	6	62	94	95	86	91
<i>size_{1j}</i>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<i>t_{2j}</i>	19	11	10	9	98	54	65	66	93	82	41	84	81	70	63	40	18	52	7	17
<i>size_{2j}</i>	5	4	2	5	4	3	2	2	5	1	1	4	4	5	5	4	3	3	2	3
<i>t_{3j}</i>	74	3	51	88	97	60	87	9	77	32	27	18	17	27	25	14	39	3	9	80
<i>size_{3j}</i>	4	2	3	2	1	3	4	4	4	3	2	3	1	4	2	2	3	1	2	2
<i>t_{4j}</i>	100	37	15	91	43	8	16	82	37	82	45	45	60	14	68	37	97	37	63	31
<i>size_{4j}</i>	1	2	1	2	2	2	2	1	2	1	2	2	1	1	2	1	2	2	1	1
<i>t_{5j}</i>	27	79	50	65	59	38	73	85	61	82	16	34	96	24	95	76	3	19	99	37
<i>size_{5j}</i>	2	4	3	1	2	1	5	1	2	5	1	5	1	5	2	3	3	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_{6j}	26	53	80	27	39	55	22	23	15	48	29	19	24	21	38	73	9	41	85	29
size $_{6j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{7j}	95	14	80	24	3	62	92	6	52	14	10	71	42	13	6	37	48	49	28	13
size $_{7j}$	4	5	4	4	2	3	1	3	5	1	4	3	1	5	5	4	4	3	2	5
t_{8j}	89	84	97	51	2	41	32	79	47	72	95	59	58	92	85	71	59	84	39	39
size $_{8j}$	3	2	2	1	3	2	1	3	2	1	1	1	2	3	2	3	2	3	1	3

A.9 Data Hybrid Flowshop 2 Stage 20 Job

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	21	22	23	24	25
t_{1j}	67	12	10	58	82	7	18	10	78	17	27	28	44	56	22	42	98	66	59	19	46	56	15	95	33
size $_{1j}$	1	1	1	1	1	1	1	1	1	2	1	1	2	2	2	2	1	1	2	1	2	1	2	2	2
t_{2j}	47	77	4	67	14	75	35	41	25	41	44	65	9	67	38	28	91	83	100	87	80	25	64	13	19
size $_{2j}$	1	1	1	1	1	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	64	26	30	6	59
size $_{3j}$	2	2	2	2	2	1	1	2	2	1	1	1	1	1	2	1	1	1	1	2	1	2	1	2	1
t_{4j}	77	91	85	9	3	91	44	31	54	95	79	53	61	70	10	68	53	91	83	63	24	58	62	49	42
size $_{4j}$	4	4	4	4	1	1	5	3	5	2	2	4	1	5	4	1	3	4	5	1	5	5	1	2	1
t_{5j}	55	79	34	54	67	50	5	60	93	2	94	78	81	68	72	32	84	42	30	34	74	13	3	75	67
size $_{5j}$	2	2	1	2	1	2	2	1	2	2	1	1	2	1	2	2	1	2	1	1	1	1	2	2	1
t_{6j}	93	52	67	91	87	13	46	78	5	52	96	55	5	17	28	54	76	48	98	15	81	15	95	20	19
size $_{6j}$	2	3	1	2	1	1	1	1	1	2	2	2	3	1	2	3	1	3	3	3	2	3	2	3	3
t_{7j}	85	90	83	71	65	53	17	34	95	59	4	39	56	46	15	68	11	14	79	94	94	53	50	15	17
size $_{7j}$	1	1	1	1	1	2	1	1	2	2	1	2	1	2	2	1	2	2	2	2	1	1	2	2	1
t_{8j}	52	84	82	86	2	30	51	69	7	76	25	22	95	1	22	12	77	40	59	12	74	23	78	57	75
size $_{8j}$	1	1	3	3	2	2	2	3	1	3	1	2	1	1	2	3	1	1	2	3	1	3	1	2	1

Stage	Job																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
t_{1j}	59	12	48	41	23	1	32	77	74	61	47	6	32	77	92	48	1	98	66	11	15	34	72	50	80
size $_{1j}$	1	2	2	2	1	1	1	2	2	1	1	1	2	1	2	2	2	2	2	1	1	2	1	2	2
t_{2j}	79	51	73	6	45	88	76	31	25	2	45	3	37	85	1	71	54	43	83	69	46	47	92	71	6
size $_{2j}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
t_{3j}	61	15	21	62	81	31	71	13	74	65	30	9	74	55	91	31	67	35	24	55	7	48	5	43	87
size $_{3j}$	1	1	2	2	2	2	1	2	2	2	2	1	1	2	2	2	2	1	1	2	2	1	1	1	1
t_{4j}	99	47	16	77	47	67	49	99	48	86	49	73	34	16	53	44	35	28	100	19	31	37	13	87	71
size $_{4j}$	2	1	1	4	5	3	4	5	3	3	3	2	4	2	4	2	5	5	5	1	1	3	3	4	5
t_{5j}	86	13	16	29	66	72	45	7	43	25	68	8	88	14	40	11	87	62	26	21	94	55	71	41	34
size $_{5j}$	1	1	1	2	1	2	1	2	2	1	1	1	1	2	1	1	2	1	1	2	2	1	2	1	1
t_{6j}	81	55	65	93	91	85	5	14	5	99	62	91	62	45	63	79	98	46	27	61	64	41	21	63	11
size $_{6j}$	3	1	3	2	2	3	2	2	2	1	1	1	2	2	3	3	1	1	1	1	3	3	1	3	2
t_{7j}	36	55	94	17	13	28	79	68	18	95	72	69	26	58	40	16	22	47	77	26	69	43	42	80	12
size $_{7j}$	2	2	1	2	2	1	1	2	2	1	2	1	2	2	2	1	1	1	1	2	1	2	2	2	2
t_{8j}	21	11	98	10	71	35	98	78	81	53	58	80	2	40	47	17	69	7	31	44	90	76	47	3	38
size $_{8j}$	3	3	2	1	1	1	2	3	2	3	2	2	3	2	1	1	3	2	3	1	3	2	2	2	1

Lampiran B. Hasil Percobaan

B.1 Hasil Percobaan Data 2 Stage 5 Job

Iterasi	Percobaan	Urutan	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PD _A (%)
500	1	5-4-2-3-1	306	1	15,6043	8,8968
	2	5-4-2-3-1	306	1	15,5994	8,8968
	3	5-4-2-3-1	306	1	15,7340	8,8968
	4	5-4-2-3-1	306	1	15,6730	8,8968
	5	5-4-2-3-1	306	1	15,8501	8,8968
1000	1	5-4-2-3-1	306	1	31,2167	8,8968
	2	5-4-2-3-1	306	1	32,7521	8,8968
	3	5-4-2-3-1	306	1	32,5315	8,8968
	4	5-4-2-3-1	306	1	32,6662	8,8968
	5	5-4-2-3-1	306	1	31,1998	8,8968

B.2 Hasil Percobaan Data 2 Stage 20 Job

Iterasi	Percobaan	Urutan	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PD _A (%)
500	1	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	22,5346	0
	2	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	22,5743	0
	3	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	22,4081	0
	4	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	22,2150	0
	5	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	21,8787	0
1000	1	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	47,5273	0
	2	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	44,7010	0
	3	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	47,7189	0
	4	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	46,2145	0
	5	3-8-17-5-18-19-11-10-7-20-1-14-9-2-12-4-6-15-16-13	807	1	48,5659	0

B.3 Hasil Percobaan Data 2 Stage 50 Job

Iterasi	Percobaan	Urutan	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PD _A (%)
500	1	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	66,2217	0
	2	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	67,8710	0
	3	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	72,3583	0
	4	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	67,8287	0
	5	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	64,4436	0
1000	1	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	148,0905	0
	2	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	135,5999	0
	3	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	146,3570	0
	4	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	151,1789	0
	5	15-9-37-10-21-3-41-49-16-19-35-34-30-31-22-33-36-1-8-5-48-24-38-46-20-42-7-39-26-47-6-32-29-40-28-25-27-23-11-2-4-18-13-44-17-50-43-12-45-14	2593	1	144,7991	0

B.4 Hasil Percobaan Data 5 Stage 5 Job

Iterasi	Percobaan	Urutan	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PD _A (%)
500	1	2-5-1-3-4	451	1	15,6055	4,8837
	2	2-5-1-3-4	451	1	15,6154	4,8837
	3	2-5-1-3-4	451	1	15,6156	4,8837
	4	2-5-1-3-4	451	1	15,6104	4,8837
	5	2-5-1-3-4	451	1	15,5990	4,8837
1000	1	2-5-1-3-4	451	1	31,2308	4,8837
	2	2-5-1-3-4	451	1	32,7912	4,8837
	3	2-5-1-3-4	451	1	34,6376	4,8837
	4	2-5-1-3-4	451	1	34,0725	4,8837
	5	2-5-1-3-4	451	1	32,7700	4,8837

B.5 Hasil Percobaan Data 5 Stage 20 Job

Iterasi	Percobaan	Urutan	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PD _A (%)
500	1	15-18-1-16-11-10-6-20-4-7-5-3-9-2-14-8-19-17-12-13	976	348	50,3609	2,6288
	2	15-19-9-6-17-11-4-7-5-10-16-18-14-12-20-8-2-1-3-13	974	195	49,0926	2,4185
	3	15-1-17-20-6-19-16-11-4-5-10-7-3-18-14-8-9-2-12-13	981	466	55,5619	3,1546
	4	15-1-8-6-20-19-16-2-7-4-11-10-5-3-9-14-18-17-12-13	991	438	55,4370	4,2061
	5	15-7-4-16-19-18-9-8-20-6-17-11-10-5-12-1-2-14-3-13	975	362	57,7523	2,5237
1000	1	15-1-11-6-20-10-4-7-2-16-18-5-17-3-19-8-9-14-12-13	976	535	94,1118	2,6288
	2	15-16-18-4-7-1-2-11-10-6-3-17-5-20-19-14-8-9-12-13	979	487	93,7060	2,9443
	3	15-19-18-16-1-20-17-6-3-7-4-11-5-10-2-14-8-9-12-13	969	424	94,1750	1,8927
	4	15-19-16-9-8-11-10-7-5-20-6-14-4-17-3-18-1-2-12-13	974	676	99,4013	2,4185
	5	15-1-16-11-10-6-4-7-14-8-9-18-5-20-3-17-19-2-12-13	979	607	94,0379	2,9443

B.6 Hasil Percobaan Data 5 Stage 50 Job

Iterasi	Percobaan	Urutan	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PD _A (%)
500	1	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	53,1837	0,9985
	2	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	53,0954	0,9985
	3	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	52,6984	0,9985
	4	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	52,8207	0,9985
	5	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	52,2981	0,9985
1000	1	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	101,3664	0,9985
	2	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	101,5715	0,9985
	3	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	101,4140	0,9985
	4	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	101,5580	0,9985
	5	47-20-43-34-18-39-4-17-7-35-42-37-22-40-23-50-46-30-33-32-25-14-49-41-48-24-27-31-12-29-36-21-13-28-16-45-10-3-9-5-11-8-2-19-15-6-26-38-44-1	2630	1	101,3567	0,9985

B.7 Hasil Percobaan Data 8 Stage 5 Job

Iterasi	Percobaan	Urutan	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PD _A (%)
500	1	2-4-3-1-5	644	1	15,5950	11,8056
	2	2-4-3-1-5	644	1	15,6056	11,8056
	3	2-4-3-1-5	644	1	15,6301	11,8056
	4	2-4-3-1-5	644	1	15,6158	11,8056
	5	2-4-3-1-5	644	1	15,6069	11,8056
1000	1	2-4-3-1-5	644	1	31,2719	11,8056
	2	2-4-3-1-5	644	1	31,2051	11,8056
	3	2-4-3-1-5	644	1	31,2041	11,8056
	4	2-4-3-1-5	644	1	31,2094	11,8056
	5	2-4-3-1-5	644	1	31,2573	11,8056

B.8 Hasil Percobaan Data 8 Stage 20 Job

Iterasi	Percobaan	Urutan	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PD _A (%)
500	1	15-3-9-1-14-10-7-13-4-5-2-11-16-8-19-12-18-6-17-20	1493	44	39,4261	0
	2	14-11-2-17-6-15-18-12-3-4-16-7-1-13-10-8-19-9-5-20	1573	1	39,4782	5,3583
	3	14-11-2-17-6-15-18-12-3-4-16-7-1-13-10-8-19-9-5-20	1573	1	39,6682	5,3583
	4	14-11-2-17-6-15-18-12-3-4-16-7-1-13-10-8-19-9-5-20	1573	1	39,4901	5,3583
	5	14-11-2-17-6-15-18-12-3-4-16-7-1-13-10-8-19-9-5-20	1573	1	39,6322	5,3583
1000	1	2-15-19-1-6-11-5-14-16-8-3-7-12-13-10-18-4-9-17-20	1543	23	78,4384	3,3490
	2	14-11-2-17-6-15-18-12-3-4-16-7-1-13-10-8-19-9-5-20	1573	1	79,0026	5,3583
	3	15-3-9-1-14-10-7-13-4-5-2-11-16-8-19-12-18-6-17-20	1493	51	78,4209	0
	4	14-11-2-17-6-15-18-12-3-4-16-7-1-13-10-8-19-9-5-20	1573	1	78,6357	5,3583
	5	14-11-2-17-6-15-18-12-3-4-16-7-1-13-10-8-19-9-5-20	1573	1	78,9514	5,3583

B.9 Hasil Percobaan Data 8 Stage 50 Job

Iterasi	Percobaan	Urutan	Makespan	Iterasi Konvergen	Waktu Komputasi (detik)	PD _A (%)
500	1	31-29-20-13-22-37-36-2-38-8-35-10-4-9-33-19-23-34-47-6-3-25-24-39-40-30-49-1-28-50-46-16-5-32-42-18-21-43-27-11-26-17-14-12-44-48-45-7-41-15	2661	104	107,9060	0,8719
	2	31-38-16-36-35-12-25-26-46-13-34-22-10-39-1-19-24-23-8-37-42-40-44-9-30-49-2-28-33-4-21-50-48-3-14-18-20-5-32-6-29-17-11-43-47-7-45-27-41-15	2661	10	106,7230	0,8719
	3	31-25-42-6-16-38-8-27-37-47-11-13-10-39-33-34-23-24-29-43-46-36-2-44-40-9-49-4-28-3-21-50-19-35-1-20-30-5-32-17-26-18-12-14-7-45-48-22-41-15	2661	72	106,3689	0,8719
	4	31-25-27-49-16-38-10-13-29-46-11-47-8-39-33-34-23-37-42-14-24-36-2-44-40-4-6-9-28-3-21-50-19-35-1-20-30-5-32-17-26-18-12-43-22-48-45-7-41-15	2661	184	106,6350	0,8719
	5	31-38-16-36-35-12-25-26-46-13-34-22-10-39-1-19-24-23-8-37-42-40-44-9-30-49-2-28-33-4-21-50-48-3-14-18-20-5-32-6-29-17-11-43-47-7-45-27-41-15	2661	101	107,4582	0,8719
1000	1	31-25-6-37-16-38-10-27-29-13-11-47-8-48-33-34-46-24-42-43-23-36-2-44-40-9-49-4-28-3-21-50-19-35-1-20-30-5-32-17-26-18-12-14-22-39-45-7-41-15	2661	372	204,0671	0,8719
	2	31-3-20-43-19-34-16-42-29-13-11-8-47-25-33-38-2-22-48-28-37-32-23-26-10-9-49-35-4-46-24-50-40-21-44-1-5-30-36-17-6-18-12-14-45-7-27-39-41-15	2661	34	205,5649	0,8719
	3	31-25-27-37-7-8-10-16-29-13-11-42-38-39-33-34-23-24-47-22-46-36-2-44-40-9-49-4-28-3-21-50-19-35-1-20-30-5-32-17-26-18-12-14-43-45-6-41-48-15	2657	372	204,6068	0,7202
	4	42-2-11-36-35-16-49-28-40-25-37-31-9-4-3-34-50-47-10-26-5-23-13-44-29-45-6-46-7-33-20-24-19-43-1-8-38-21-32-17-30-18-12-14-39-22-41-48-27-15	2653	984	204,1775	0,5686
	5	31-25-6-37-38-22-10-13-29-20-11-47-8-23-33-34-16-36-42-27-46-24-2-44-40-9-49-4-35-3-21-50-19-28-1-48-30-5-32-17-26-18-12-14-39-45-43-7-41-15	2661	119	202,5394	0,8719