



**PENERAPAN ALGORITMA *CUCKOO SEARCH* DALAM
PERMASALAHAN PENJADWALAN *FLOWSHOP***

SKRIPSI

Oleh

**Reny Andiyani Januari Arti
NIM 131810101052**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017**



**PENERAPAN ALGORITMA *CUCKOO SEARCH* DALAM
PERMASALAHAN PENJADWALAN *FLOWSHOP***

SKRIPSI

disusun guna melengkapi tugas akhir sebagai salah satu syarat
untuk menyelesaikan studi pada Program Studi Matematika (S1)
dan mencapai gelar sarjana Sains

Oleh

**Reny Andiyani Januari Arti
NIM 131810101052**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017**

PERSEMBAHAN

Bismillahirohmannirrohim, Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah dan inayah-Nya kepada saya sehingga saya dapat menyelesaikan tugas akhir ini dengan lancar. Puji syukur pula pada Nabi Muhammad SAW yang telah membimbing umat meninggalkan zaman jahiliyah. Skripsi ini saya persembahkan untuk orang-orang spesial, yaitu:

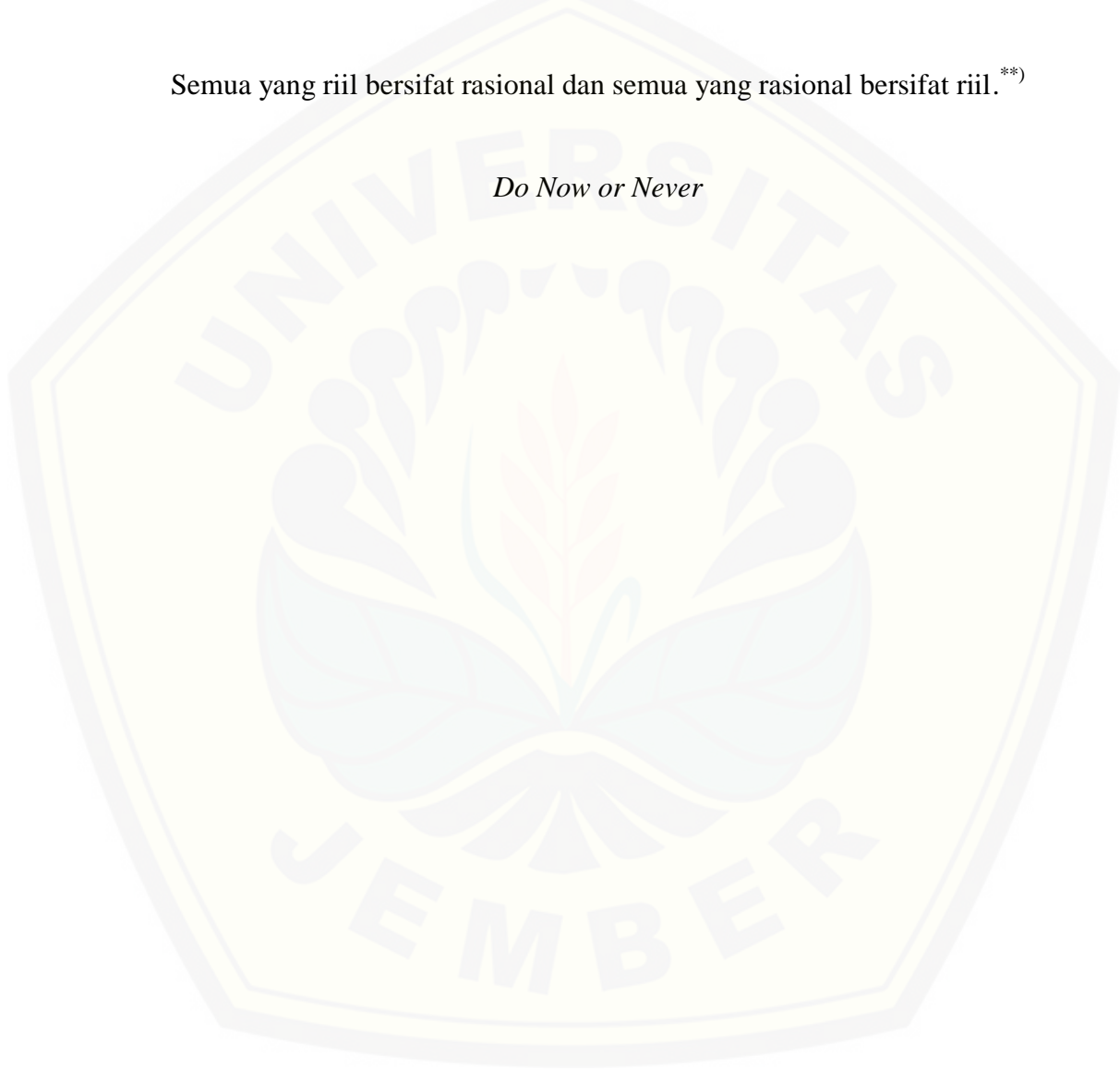
1. Orang tua saya yang selalu menjadi inspirasi dan penyemangat dalam menapaki kehidupan, Ayahanda Sudarsono dan Ibunda Ratmini yang dengan sabar dan penuh kasih sayang membimbing dan memberikan nasehat kepada saya hingga saat ini,
2. Adik saya Dava Kusuma Ardana yang telah memberikan semangat dan motivasi.
3. Pramudha Nur Ramadhana yang telah memberikan bantuan kepada saya baik moril maupun tenaga,
4. Sahabat saya Nita Nurmala dan Selly Acita yang telah memberikan semangat dan masukan untuk menyelesaikan skripsi ini.
5. Semua guru-guru yang pernah membimbing saya mulai dari taman kanak-kanak sampai dengan perguruan tinggi yang telah memberikan ilmu serta pembelajaran hidup yang bermanfaat kepada saya,
6. Teman-teman Atlas dan semua pihak yang selama ini memberikan dukungan pada saya sehingga skripsi ini bisa terselesaikan.
7. Almamater tercinta Jurusan Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTO

Orang-orang yang sukses telah belajar membuat diri mereka melakukan hal yang harus dikerjakan ketika hal itu memang harus dikerjakan, entah mereka menyukainya atau tidak.^{*)}

Semua yang riil bersifat rasional dan semua yang rasional bersifat riil.^{**)}

Do Now or Never



^{*)} Aldus Huxley

^{**)} Hegel

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Reny Andiyani Januari Arti

NIM : 131810101052

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan Algoritma *Cuckoo Search* dalam Permasalahan Penjadwalan *Flowshop*” 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, Juli 2017

Yang menyatakan,

Reny Andiyani Januari Arti
NIM 131810101052

SKRIPSI

**PENERAPAN ALGORITMA *CUCKOO SEARCH* DALAM
PERMASALAHAN PENJADWALAN *FLOWSHOP***

Oleh

Reny Andiyani Januari Arti
NIM 131810101052

Pembimbing:

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

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

PENGESAHAN

Skripsi berjudul “Penerapan Algoritma *Cuckoo Search* dalam Permasalahan Penjadwalan *Flowshop*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji:

Ketua,

Anggota I,

Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP. 197211291998021001

M. Ziaul Arif, S.Si., M.Sc.
NIP. 198501112008121002

Anggota II,

Anggota III,

Prof. Drs. I Made Tirta, M.Sc., Ph.D.
NIP. 195912201985031002

Drs. Rusli Hidayat, M.Sc.
NIP. 196610121993031001

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Sujito, Ph.D.
NIP. 196102041987111001

RINGKASAN

Penerapan Algoritma *Cuckoo Search* dalam Permasalahan Penjadwalan *Flowshop*. Reny Andiyani Januari Arti, 131810101052; 2017; 57 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penjadwalan suatu produksi merupakan hal yang sangat penting dilakukan oleh suatu perusahaan. Penjadwalan merupakan rencana pengaturan urutan kerja serta pengalokasian sumber, baik waktu maupun fasilitas untuk setiap operasi yang harus diselesaikan. Adapun tujuan dari penjadwalan produksi yaitu meminimasi *makespan*. Salah satu bentuk penjadwalan adalah penjadwalan *flowshop*. Penjadwalan *flowshop* merupakan model penjadwalan dimana *job-job* yang akan diproses seluruhnya mengalir pada arah atau jalur produk yang sama.

Penelitian ini bertujuan menemukan solusi masalah penjadwalan *flowshop* dengan algoritma *Cuckoo Search* menggunakan 4 data yang terdiri dari data sekunder dan data simulasi. Data sekunder yang digunakan adalah data produksi kerupuk UD. Samjaya. Sedangkan data simulasi merupakan data yang diambil dari *website* Prof. Éric Tailard, *University of Applied Sciences of Western Switzerland*. Hasil penelitian terhadap data produksi UD. Samjaya akan dibandingkan dengan hasil dari algoritma Genetika dan *Simulated Anaeling* (SA) yang pernah menganalisis data tersebut. Kemudian hasil terhadap data simulasi akan dibandingkan dengan hasil pada algoritma *Iterated Greedy* dan *Social Cognitive Optimization* (SCO). Pebandingan tersebut berdasarkan nilai *makespan* yang diperoleh. Selain itu, penelitian ini juga akan menganalisis pengaruh parameter-parameter terhadap nilai *makespan* yang dihasilkan.

Berdasarkan hasil penelitian yang dilakukan, nilai *makespan* yang dihasilkan algoritma *Cuckoo Search* terhadap data produksi UD. Samjaya sama dengan nilai *makespan* yang diperoleh dari algoritma Genetika dan *Simulated Anaeling* yaitu sebesar 8878. Sedangkan hasil penelitian terhadap data simulasi, algoritma *Cuckoo Search* menghasilkan *makespan* yang lebih minimum dibandingkan dengan nilai *makespan* yang dihasilkan *Iterated Greedy* dan

algoritma *Social Cognitive Optimization* (SCO). Nilai *makepan* yang dihasilkan algoritma *Cuckoo Search* adalah sebesar 1859 menit pada data 2, 2406 menit pada data 3 dan 4018 menit pada data 4. Nilai tersebut lebih minimum jika dibandingkan nilai *makespan* yang dihasilkan algoritma *Social Cognitive Optimization* (SCO) sebesar 1948 menit pada data 2, 2627 menit pada data 3 dan 4367 menit pada data 4. Nilai *makespan* yang dihasilkan algoritma *Cuckoo Search* tersebut juga lebih minimum dibandingkan dengan nilai *makespan* yang dihasilkan oleh algoritma *Iterated Greedy* yaitu sebesar 1882 menit pada data 2, 2411 menit pada data 3 dan 4022 menit pada data 4.

Data hasil pengujian 6 parameter *Cuckoo Search*, diantaranya parameter *Lévy Fligth* yaitu α , β , γ dan δ , populasi, selang pembakitan bilangan acak (*Lower Bound* dan *Uper Bond*), parameter peluang (P_a), *Stepsize* (s) dan iterasi maksimal menunjukkan bahwa semakin besar atau lebar selang yang dipilih maka nilai *makespan* yang dihasilkan akan semakin kecil. Begitu juga dengan populasi, semakin besar populasi maka semakin kecil nilai *makespan* yang didapatkan. Semakin besar nilai P_a dan iterasi maksimal yang dipilih maka semakin kecil *makespan* yang dihasilkan. Namun pada parameter *stepsize*, nilai *makespan* yang terkecil dihasilkan oleh nilai *stepsize* terkecil. Nilai *makespan* terkecil didapatkan menggunakan nilai-nilai parameter sebagai berikut: selang $-4 < x < 4$, populasi sebesar 200, nilai parameter *Lévy* sebesar $\alpha=1,4$, $\beta=0,9$, $\gamma=1$ dan $\delta=-0,5$, nilai *stepsize* sebesar 0,1, nilai P_a sebesar 0,75 dan iterasi sebanyak 75.000 kali. Setiap nilai minimum yang diperoleh melalui masing-masing 10 kali *running* untuk parameter yang sama.

Dari penelitian yang dilakukan diperoleh hasil yang menunjukkan bahwa algoritma *Cuckoo Search* lebih baik dari algoritma *Social Cognitive Optimization* (SCO) dan *Iterated Greedy* karena menghasilkan nilai *makespan* yang lebih minimum. Namun pada data yang berukuran besar algoritma *Cuckoo Search* memiliki kelemahan yaitu membutuhkan iterasi maksimal yang besar sehingga mengakibatkan lamanya proses komputasi.

PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan kuasa-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Penerapan Algoritma *Cuckoo Search* dalam Permasalahan Penjadwalan *Flowshop*”. 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 Ahmad Kamsyakawuni, S.Si., M.Kom. selaku dosen pembimbing utama dan Bapak M. Ziaul Arif, S.Si., M.Sc. selaku dosen pembimbing anggota.
2. Bapak Prof. Drs. I Made Tirta, M.Sc., Ph.D. selaku dosen penguji I dan Bapak Drs. Rusli Hidayat, M.Sc. selaku dosen penguji II.
3. Seluruh dosen dan karyawan Jurusan Matematika FMIPA Universitas Jember.
4. Keluarga yang telah memberikan motivasi dan doa tulus ikhlas penuh kasih sayangnya.
5. Teman-teman dan semua pihak yang telah membantu dan memberikan semangat.

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, Juli 2017

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN.....	ii
HALAMAN MOTO	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PEMBIMBINGAN.....	v
HALAMAN PENGESAHAN.....	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN	xiv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1 Penjadwalan Produksi.....	5
2.1.1 Elemen Penjadwalan.....	5
2.1.2 Tujuan Penjadwalan.....	6
2.2 Penjadwalan <i>Flowshop</i>	7
2.3 Diagram <i>Gantt</i>	8
2.4 Algoritma <i>Cuckoo Search</i>.....	10
2.4.1 Perilaku Burung <i>Cuckoo</i>	11
2.4.2 Karakteristik <i>Cuckoo Search</i>	12
2.4.3 Mekanisme Algoritma <i>Cuckoo Search</i>	13

BAB 3. METODE PENELITIAN	15
3.1 Data	15
3.2 Langkah-langkah Penelitian	15
BAB 4. HASIL DAN PEMBAHASAN	19
4.1 Hasil Penelitian	19
4.1.1 Langkah Perhitungan	19
4.1.2 Hasil Program	25
4.1.3 Percobaan dan Uji Parameter.....	30
4.2 Pembahasan	34
BAB 5. KESIMPULAN DAN SARAN	40
5.1 Kesimpulan	40
5.2 Saran	40
DAFTAR PUSTAKA	42
LAMPIRAN	44

DAFTAR GAMBAR

	Halaman
2.1 Pola Aliran <i>Pure Flowshop</i>	7
2.2 Pola Aliran <i>General Flowshop</i>	8
2.3 Diagram <i>Gantt</i>	9
2.4 Skema Algoritma <i>Cuckoo Search</i>	14
3.1 Skema Langkah Penelitian.....	17
4.1 Tampilan Awal Program.....	27
4.2 Tampilan Setelah Data Diinput.....	28
4.3 Program Ketika Diproses	29
4.4 Tampilan Hasil Program	29
4.5 Tampilan Diagram <i>Gantt</i>	30

DAFTAR TABEL

	Halaman
2.1 Perbandingan Hasil antara Algoritma GA, PSO dan CS	10
4.1 Data Perhitungan.....	19
4.2 Nilai Awal	20
4.3 Nilai Awal Urut.....	20
4.4 Matriks Hasil Diskritisasi.....	21
4.5 Perhitungan Nilai <i>Makespan</i> Baris 1.....	21
4.6 Nilai <i>Best-So-Far</i>	22
4.7 Nilai Random <i>Lêvy</i>	22
4.8 <i>Cuckoo i</i>	23
4.9 <i>Cuckoo i</i> Ururt	23
4.10 Perhitungan Nilai <i>Makespan Cuckoo i</i>	23
4.11 <i>Cuckoo j</i>	23
4.12 Tabel Solusi.....	24
4.13 Nilai Random <i>Lêvy</i>	25
4.14 Kandidat Solusi Baru	25
4.15 Hasil Uji Parameter <i>Lower Bound (Lb)</i> dan <i>Uper Bound (Ub)</i>	31
4.16 Hasil Uji Parameter Populasi	31
4.17 Hasil Uji Parameter <i>Lêvy</i>	32
4.18 Hasil Uji Parameter <i>Stepsize</i>	33
4.19 Hasil Uji Parameter P_a	33
4.20 Hasil Uji Parameter Iterasi Maksimal	34

DAFTAR LAMPIRAN

	Halaman
A.1 Data Produksi Kerupuk UD. Samjaya	44
A.2 Data Simulasi 20 Job 15 Mesin.....	45
A.3 Data Simulasi 30 Job 15 Mesin.....	46
A.4 Data Simulasi 50 Job 20 Mesin.....	47
B.1 Uji Parameter <i>Lower Bound (Lb)</i> dan <i>Uper Bound (Ub)</i> ($-4 < x < 4$)	50
B.2 Uji Parameter <i>Lower Bound (Lb)</i> dan <i>Uper Bound (Ub)</i> ($0 < x < 4$)	50
B.3 Uji Parameter <i>Lower Bound (Lb)</i> dan <i>Uper Bound (Ub)</i> ($-1 < x < 1$)	50
B.4 Uji Parameter <i>Lower Bound (Lb)</i> dan <i>Uper Bound (Ub)</i> ($0 < x < 1$)	51
C.1 Uji Parameter Populasi (pop=20)	51
C.2 Uji Parameter Populasi (pop=50)	51
C.3 Uji Parameter Populasi (pop=100)	52
C.4 Uji Parameter Populasi (pop=200)	52
D.1 Uji Parameter <i>Lêvy</i> ($\alpha, \beta, \gamma, \delta$) (1,4, 0,8, 1,3, 0,7)	52
D.2 Uji Parameter <i>Lêvy</i> ($\alpha, \beta, \gamma, \delta$) (1,4, 0,9, 1, -0,5)	53
D.3 Uji Parameter <i>Lêvy</i> ($\alpha, \beta, \gamma, \delta$) (0,1, 0, 1,3, 0,7)	53
D.4 Uji Parameter <i>Lêvy</i> ($\alpha, \beta, \gamma, \delta$) (0,001, 0,8, 1,3, 0,7)	53
E.1 Uji Parameter <i>Stepsize</i> ($s = 0,1$)	54
E.2 Uji Parameter <i>Stepsize</i> ($s = 0,25$)	54
E.3 Uji Parameter <i>Stepsize</i> ($s = 1$)	54
E.4 Uji Parameter <i>Stepsize</i> ($s = 2$)	55
F.1 Uji Parameter P_a ($P_a = 0,25$)	55
F.2 Uji Parameter P_a ($P_a = 0,5$)	55
F.3 Uji Parameter P_a ($P_a = 0,75$).....	56
G.1 Uji Parameter Iterasi Maksimal ($MaxGen = 5000$)	56
G.2 Uji Parameter Iterasi Maksimal ($MaxGen = 10000$)	56
G.3 Uji Parameter Iterasi Maksimal ($MaxGen = 50000$)	57
G.4 Uji Parameter Iterasi Maksimal ($MaxGen = 75000$)	57

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Sektor industri pada masa ini berkembang semakin pesat seiring dengan kemajuan teknologi. Perkembangan tersebut ditandai dengan semakin ketatnya persaingan antara suatu perusahaan dengan perusahaan yang lain. Dengan persaingan yang semakin ketat, perusahaan dituntut untuk semakin memaksimalkan kinerjanya di berbagai bidang. Kinerja suatu perusahaan akan maksimal jika perusahaan dapat meningkatkan sumber daya manusia yang dimiliki, perbaikan manajemen perusahaan dan peningkatan teknologi yang digunakan. Selain itu perusahaan juga perlu meningkatkan strategi pelayanan konsumen, diantaranya dengan meningkatkan kualitas produk sesuai dengan permintaan konsumen serta agar barang tersebut dapat diterima konsumen tepat waktu.

Proses mengolah input menjadi produk pada suatu industri tidak lepas dari peralatan dan mesin untuk mempermudah proses pengolahan serta diperlukan juga tenaga manusia sebagai operatornya. Proses pengolahan tersebut membutuhkan pengaturan agar mesin maupun operator dapat bekerja secara optimum dan efisien terhadap waktu. Pengaturan operator maupun mesin dapat dilakukan dengan penjadwalan. Penjadwalan berfungsi untuk mengoptimalkan waktu penyelesaian pekerjaan pada mesin agar dapat memenuhi target produksi dan juga berfungsi sebagai alat pengambil keputusan.

Penjadwalan adalah kegiatan pengalokasian sumber-sumber atau mesin-mesin yang ada untuk menjalankan sekumpulan tugas dalam jangka waktu tertentu. Penjadwalan juga dapat dikatakan sebagai proses pengambilan keputusan yang memiliki peranan penting dalam dunia industri. Penjadwalan produksi dapat dibedakan menjadi penjadwalan *flowshop* dan *jobshop*. Penjadwalan *flowshop* adalah suatu pergerakan unit yang berlangsung secara terus menerus melalui suatu rangkaian stasiun-stasiun kerja yang disusun berdasarkan produk (Baker,1974).

Dalam pengembangan ilmu Matematika, metode optimasi untuk menyelesaikan permasalahan *flowshop* semakin berkembang. Beberapa algoritma

yang dapat digunakan dalam penyelesaian penjadwalan *flowshop* diantaranya algoritma *Tabu Search* (TS), *Harmony Search* (HS), *Cross Entropy* (CE), Genetika (AG), *Differential Evolution Plus*, *Iterated Greedy*, *Social Cognitive Optimization* (SCO) serta dengan banyak metode-metode metaheuristik lain. Menurut Santosa dan Willy (2011), metaheuristik merupakan metode untuk mencari solusi dengan memadukan interaksi antara pencarian lokal dan strategi yang lebih tinggi untuk menciptakan proses yang mampu keluar dari titik-titik lokal optimum dan melakukan pencarian di ruang solusi untuk menemukan solusi global.

Pada penelitian sebelumnya yang telah dilakukan Efendi et al. (2016) masalah penjadwalan *flowshop* telah berhasil diselesaikan menggunakan algoritma *Iterated Greedy* dan algoritma *Social Cognitive Optimization* (SCO). Selain itu, penelitian mengenai masalah penjadwalan *flowshop* lain dilakukan oleh Nurjanah et al. (2015) yang menggunakan algoritma Genetika dan algoritma *Simulated Annealing* (SA) serta Witjaksana et al. (2016) yang menyelesaikan permasalahan *flowshop* menggunakan *Cross Entropy-Genetic Algorithm* dan algoritma *Differential Evolution Plus*.

Dari berbagai algoritma metaheuristik yang telah dikembangkan, algoritma *Cuckoo Search* (CS) merupakan salah satu dari *Nature-Inspired Algorithms* yang memiliki performa mengesankan pada permasalahan optimasi untuk kasus bilangan real. *Cuckoo Search* (CS) pun menjanjikan performa yang baik untuk kasus permasalahan diskret, seperti masalah kombinatorial. Seperti yang dipaparkan pada penelitian Bayu et al. (2012) mengenai implementasi *Cuckoo Search algorithm* pada *Travelling Salesman Problem* menggunakan *lêvy flight*. Penelitian lain yang menerapkan algoritma *Cuckoo Search* (CS) adalah penelitian yang dilakukan oleh Baihaki et al. (2016) pada permasalahan *Multiple Constraints Knapsack* 0-1. Hasilnya menunjukkan bahwa permasalahan knapsack 0-1 dapat dioptimalkan menggunakan algoritma *Cuckoo Search*. Pada tahun 2013 hasil penelitian Yildiz menunjukkan bahwa algoritma *Cuckoo Search* lebih baik dari beberapa algoritma optimasi populer seperti *Ant Colony Algorithm*, *Immune Algorithm*, *Hybrid Immune Algorithm*, *Hybrid Particle Swarm Algorithm*, *Genetic*

Algorithm Dan *Feasible Direction Method* untuk menguji tiga parameter yaitu harga jual, waktu pengerjaan serta keuntungan dalam permasalahan *Milling Operation*. Pada tahun 2009, dalam penelitiannya Yang dan Deb juga dibuktikan bahwa algoritma *Cuckoo Search* memiliki persentase peluang yang lebih besar untuk menemukan global optima dari pada algoritma *Particle Swarm Optimization* dan Algoritma Genetika pada hasil uji 12 fungsi stokastik yang berbeda.

Berbagai algoritma telah berhasil menemukan solusi optimal dari permasalahan penjadwalan *flowshop*. Beberapa permasalahan pun telah berhasil dioptimasi menggunakan algoritma *Cuckoo Search*. Dari berbagai penelitian tersebut, dapat diketahui bahwa algoritma *Cuckoo Search* lebih baik dalam menemukan solusi dari sebuah permasalahan dibandingkan dengan beberapa algoritma lain. Berkaitan dengan hal tersebut, penulis tertarik untuk menerapkan algoritma *Cuckoo Search* pada penjadwalan *flowshop* untuk menguji keoptimalan algoritma *Cuckoo Search* dan memperoleh solusi optimal berdasarkan aspek *makespan* terkecil.

1.2 Rumusan Masalah

Berdasarkan latar belakang, diperoleh permasalahan sebagai berikut:

- a. Bagaimana menerapkan algoritma *Cuckoo Search* pada penjadwalan *flowshop*?
- b. Bagaimana pengaruh parameter algoritma *Cuckoo Search* dalam optimasi penjadwalan *flowshop*?

1.3 Batasan Masalah

Agar tidak memperluas permasalahan, peneliti mendefinisikan asumsi-asumsi sebagai berikut:

- a. Bahan baku yang akan digunakan selalu tersedia.
- b. Setiap pekerjaan memiliki *ready time* yang sama.
- c. Mesin berjalan dengan normal tanpa gangguan.
- d. Waktu pendistribusian produk dari satu mesin ke mesin lainnya diabaikan.

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

- a. Menerapkan solusi permasalahan penjadwalan *flowshop* dengan menggunakan algoritma *Cuckoo Search* (CS).
- b. Mengetahui pengaruh parameter dalam algoritma *Cuckoo Search* (CS) dalam optimasi penjadwalan *flowshop*.

1.5 Manfaat

Manfaat yang diharapkan dalam penelitian ini adalah sebagai berikut:

- a. Dapat memberikan informasi kepada peneliti selanjutnya tentang algoritma *Cuckoo Search* dan penjadwalan *flowshop*.
- b. Dapat membantu perusahaan-perusahaan dalam melakukan penjadwalan sehingga diharapkan dapat memperoleh keuntungan yang lebih maksimal.

BAB 2. TINJAUAN PUSTAKA

2.1 Penjadwalan Produksi

Penjadwalan didefinisikan sebagai rencana pengaturan urutan kerja serta pengalokasian sumber, baik waktu maupun fasilitas untuk setiap operasi yang harus diselesaikan (Vollman, 1998). Penjadwalan dapat dikatakan suatu proses pengambilan keputusan yang memainkan peranan penting dalam kebanyakan bidang manufaktur dan pelayanan industri. Penjadwalan digunakan dalam pengadaan bahan produksi dalam bidang transportasi dan distribusi serta dalam proses informasi dan komunikasi. Hal tersebut juga dapat dijadikan sebagai alat ukur yang berguna dalam kegiatan produksi dengan jangka yang pendek pada satu periode. Suatu perusahaan memerlukan suatu penjadwalan untuk memaksimalkan pelayanan, yaitu dengan memproduksi produk sesuai dengan permintaan dan dapat diterima konsumen tepat waktu (Nurjanah *et al.*, 2015).

2.1.1 Elemen Penjadwalan

Menurut Baker (1974), dalam proses produksi terdapat tiga elemen penjadwalan sebagai berikut:

a. *Job*

Job dapat didefinisikan sebagai suatu pekerjaan yang harus diselesaikan untuk mendapatkan suatu produk. *Job* biasanya terdiri dari beberapa operasi yang harus dikerjakan (minimal 1 operasi). Manajemen melalui perencanaan yang telah dibuat atau berdasarkan pesanan dari pelanggan, memberikan *job* kepada bagian operator untuk dikerjakan. Informasi yang dimiliki oleh suatu *job* ketika datang ke bagian operator biasanya adalah operasi-operasi yang harus dilakukan didalamnya.

b. Operasi

Operasi adalah bagian proses dari *job* untuk menyelesaikan suatu *job*. Operasi-operasi dalam *job* diurutkan dalam suatu urutan pengerjaan tertentu. Urutan tersebut ditentukan pada saat perencanaan proses. Suatu operasi baru dapat dikerjakan apabila operasi atau proses yang mendahuluinya sudah dikerjakan

terlebih dahulu. Waktu proses operasi adalah waktu pengerjaan yang diperlukan untuk melakukan operasi tersebut. Waktu proses operasi untuk suatu *job* biasanya telah diketahui sebelumnya dan mempunyai besar tertentu. Waktu proses operasi ditampilkan juga dalam bentuk tabel yang dikenal dengan tabel waktu operasi.

c. Mesin

Mesin adalah sumber daya yang diperlukan untuk mengerjakan proses penyelesaian suatu *job*. Setiap mesin hanya dapat memproses satu tugas pada satu waktu.

2.1.2 Tujuan Penjadwalan

Menurut Ginting (2009), dalam melakukan penjadwalan terdapat berbagai tujuan yang ingin dicapai dengan berbagai keterbatasan sumber daya yang dimiliki diantaranya:

- a. Menekan waktu tunggu suatu sumber daya sehingga waktu total pemrosesan bisa berkurang dan produktivitas meningkat.
- b. Meminimumkan nilai *makespan* dengan cara meminimasi *job* yang ada dalam daftar tunggu ketika sedang berlangsung proses lain. Indikator yang digunakan dinyatakan dengan besaran waktu alir rata-rata.
- c. Mengurangi waktu keterlambatan dengan menjamin pemenuhan batas waktu yang telah ditetapkan dalam setiap *order*. Sejumlah pekerjaan memiliki batas waktu penyelesaian, bila waktu penyelesaiannya melewati batas maka akan ada biaya tambahan yang harus dikeluarkan.

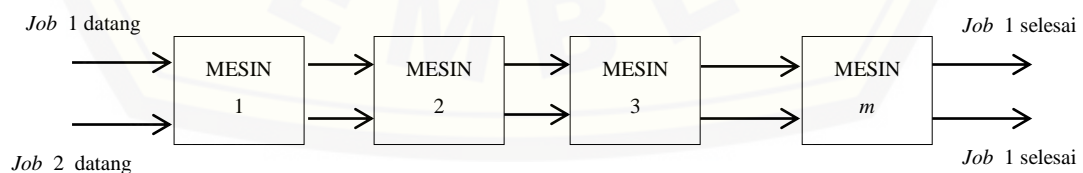
Masalah penjadwalan memiliki beberapa jenis model yang setiap modelnya memiliki karakteristik tertentu. Karakteristik penjadwalan tersebut dibedakan dalam beberapa hal diantaranya jumlah mesin, pola kedatangan *job*, sistem informasi yang berhubungan dengan karakteristik *job* dan pola aliran proses. Berdasarkan pola alirannya penjadwalan dibagi menjadi tiga, yaitu *pure flowshop*, *general flowshop*, dan *jobshop*. *Flowshop* adalah penjadwalan produksi yang searah sedangkan *jobshop* tidak searah (Conway *et al.*, 1967).

2.2 Penjadwalan *Flowshop*

Proses produksi dengan aliran *flowshop* merupakan model penjadwalan dimana *job-job* yang akan diproses seluruhnya mengalir pada arah atau jalur produk yang sama. Dengan kata lain, setiap *job* dapat diproses dengan urutan yang sama pada paling tidak satu mesin dan satu mesin dapat memproses paling banyak satu *job*. Umumnya, pada sistem produksi yang menggunakan aliran *flowshop*, terdapat sejumlah mesin yaitu $(i,1)$, $(i,2)$, ..., (i,m) yang digunakan untuk mengerjakan sejumlah pekerjaan yaitu $(j,1)$, $(j,2)$, ..., (j,n) dan membutuhkan waktu pengerjaan (t_{ij}) . Penjadwalan *flowshop* dapat diselesaikan dengan mengembangkan permutasi urutan *job*. Dalam *flowshop*, *job* bersifat *independent*, memiliki *ready time* yang sama pada waktu ke nol, dan urutan mesin dari semua pekerjaan adalah sama dengan masing-masing waktu yang diperlukan. Tujuan dari proses penjadwalan ini secara umum adalah untuk menghitung nilai *makespan* yang minimum atau total waktu proses yang dibutuhkan untuk menyelesaikan sekumpulan *job* (Nawas, 1983).

Aliran pada penjadwalan *flowshop* dibagi menjadi 2, yaitu *pure flowshop* dan *general flowshop* (Nawas, 1983).

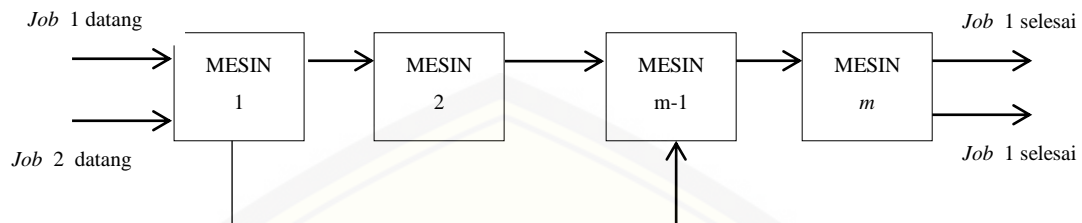
- a. *Pure flowshop*, pola aliran prosesnya identik. Setiap *job* melewati seluruh mesin yang bekerja dari proses awal hingga proses akhir sesuai dengan urutan. Masing-masing *job* melalui mesin 1, kemudian mesin 2, dan seterusnya hingga mesin ke m . Gambar 2.1 berikut menunjukkan gambaran tentang *Pure flowshop*.



Gambar 2.1 Pola aliran *Pure flowshop*.

- b. *General flowshop*, pola aliran prosesnya tidak identik. Masing-masing *job* tidak selalu melewati seluruh mesin yang bekerja. *Job* yang datang tidak harus

dikerjakan oleh semua mesin. Gambar 2.2 berikut menunjukkan gambaran tentang *General flowshop*.



Gambar 2.2 Pola aliran *General flowshop*.

Penjadwalan *flowshop* memiliki korespondensi satu-satu antara urutan dari pekerjaan dan permutasi dari indeks pekerjaan. Jumlah total solusi yang berbeda untuk masalah mesin tunggal adalah yang merupakan jumlah urutan yang berbeda dari *job*. Dalam proses penjadwalan, perlu diketahui jenis dan jumlah masing-masing sumber daya sehingga bisa ditentukan kapan tugas dapat dicapai. Ketika hal tersebut sudah ditentukan maka secara efektif batas penjadwalan dapat ditentukan. Secara umum durasi tugas tidak pasti, akan tetapi ketidakpastian dapat ditekan dengan menentukan batas akhir atau jatuh tempo *job*. Salah satu model yang paling sederhana dan paling banyak digunakan adalah *Gantt chart* (representasi dari jadwal).

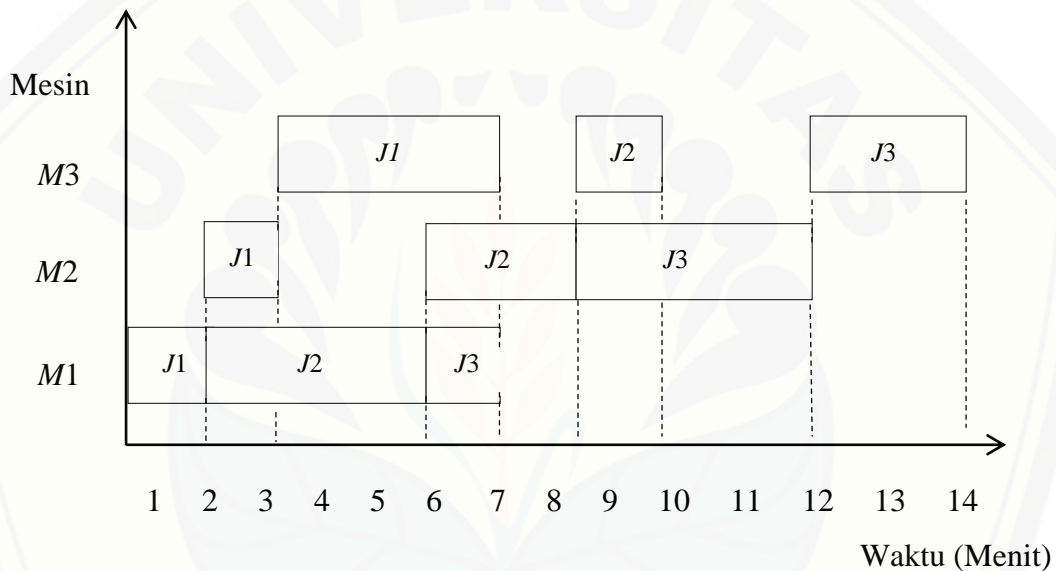
2.3 Diagram *Gantt*

Berbagai model penjadwalan akan memberikan rumusan masalah yang sistematis berikut dengan solusi yang diharapkan. Alat bantu yang digunakan untuk menyelesaikan masalah tersebut adalah diagram atau peta *Gantt*. Menurut Ginting (2009), *Gantt chart* adalah grafik hubungan antara alokasi sumber daya yang dimiliki dengan waktu pemrosesan yang dibutuhkan. Dalam bentuk dasarnya, diagram *Gantt* menampilkan alokasi sumber daya dari waktu ke waktu ditunjukkan sepanjang sumbu vertikal dan skala waktu ditampilkan di sepanjang sumbu horisontal.

Keuntungan menggunakan diagram antara lain:

- Dengan keadaan sumber yang terbatas, memungkinkan evaluasi lebih awal mengenai penggunaan sumber daya seperti yang telah direncanakan;
- Kemajuan pekerjaan mudah diperiksa pada setiap waktu karena sudah tergambar dengan jelas;
- Semua pekerjaan diperhatikan secara grafis dalam suatu diagram yang mudah dipahami.

Gambar 2.3 menunjukkan gambaran secara sederhana dari bagan diagram *Gantt*.



Gambar 2.3 Diagram *Gantt*

Gantt chart mengasumsikan bahwa waktu pengolahan yang diketahui dengan pasti, seperti pada Gambar 2.3. Sebuah grafik seperti Gambar 2.3 membantu untuk memvisualisasikan jadwal karena sumber daya dan tugas dapat terlihat dengan jelas. Dengan *Gantt chart* bisa ditemukan informasi tentang jadwal yang diberikan dengan menganalisis hubungan geometris. Selain itu, kita dapat mengatur ulang tugas pada bagan untuk memperoleh informasi tentang jadwal alternatif. Dengan cara ini, *Gantt chart* berfungsi sebagai bantuan untuk mengukur kinerja dan membandingkan jadwal serta untuk memvisualisasikan masalah dalam tempat pertama.

2.4 Algoritma *Cuckoo Search*

Menurut Yang dan Deb (2009), algoritma *Cuckoo Search* merupakan algoritma metaheuristik modern yang didasarkan pada perilaku parasit burung *Cuckoo* dalam menempatkan telurnya. Burung *Cuckoo* yang bertelur tidak mengerami telurnya itu, melainkan menaruh telurnya di sarang burung yang lain dan telur yang lain dalam sarang itu dibuang untuk menambah kemungkinan menetas keturunannya. Jika sang burung asli mengetahui keberadaan telur yang asing, ada kemungkinan untuk membuang telur tersebut atau membuat sarang baru.

Lévy Flight (LF) merupakan *random walk* yang terdistribusi oleh *Lévy*. LF terbukti dapat memetakan lalat buah dalam mencari makan. Lalat buah dalam mencari makan akan berkonsentrasi pada satu titik kemudian jika lalat buah tersebut merasa makanan disana sudah habis, maka akan mencari ke tempat lain. Pada penelitian tersebut juga dijelaskan bahwa algoritma LF membantu algoritma CS dalam pencarian karena langkah pencariannya semakin lebar. Algoritma CS yang menggunakan LF memiliki ketepatan yang lebih baik dari algoritma optimasi yang lain dalam menentukan titik optimum.

Berdasarkan hasil penelitian Yang dan Deb (2009), perbandingan yang dilakukan antara algoritma *Cuckoo Search*, PSO, dan Genetika, dengan menggunakan beberapa fungsi pengetesan berbeda dan uji coba sebanyak 100 kali. Dari data tersebut dapat dilihat, *Cuckoo Search* memiliki persentase peluang untuk menemukan global optima lebih besar daripada algoritma *Particle Swarm Optimization* dan Algoritma Genetika. Pada Tabel 2.1 dapat dilihat perbandingan ketiga algoritma tersebut pada dua belas fungsi stokastik berbeda.

Tabel 2.1. Perbandingan hasil antara algoritma Genetik, PSO dan CS

Fungsi	GA	PSO	CS
Michaelwicz	89325 ± 7914 (95%)	6922 ± 537 (98%)	3221 ± 519 (100%)
Rosenbrock	55723 ± 8901 (90%)	32756 ± 5325 (98%)	5923 ± 1937 (100%)
De Jong	15232 ± 1270 (100%)	10079 ± 970 (100%)	3015 ± 540 (100%)

Fungsi	GA	PSO	CS
Schwefel	23790 ± 6523 (95%)	92411 ± 1163 (97%)	4710 ± 592 (100%)
Ackley	32720 ± 3327 (90%)	23407 ± 4325 (92%)	4936 ± 903 (100%)
Rastrigin	110523 ± 5199 (77%)	79491 ± 3715 (90%)	10354 ± 3755(100%)
Easom	19239 ± 3307 (92%)	17273 ± 2929 (90%)	6751 ± 1902 (100%)
Griewangk	70925 ± 7652 (90%)	55970 ± 4223 (92%)	10912 ± 4050 (100%)
Stokastik pertama Yang	79025 ± 6312 (49%)	34056 ± 4470 (90%)	11254 ± 2733 (99%)
Stokastik kedua Yang	35072 ± 3730 (54%)	22360 ± 2649 (92%)	8559 ± 3480 (98%)
Robsenrockdengan komponen stokastik	63268 ± 5091 (40%)	49152 ± 6505 (89%)	10564 ± 4297 (99%)
Stokasik De Jong	24164 ± 4923 (68%)	11780 ± 4912 (94%)	7723 ± 2504 (100%)

2.4.1 Perilaku Burung *Cuckoo*

Yang and Deb (2013) menjelaskan lebih lanjut tentang perilaku unik burung *Cuckoo* sebagai burung yang memiliki perilaku parasit. Burung *Cuckoo* memiliki strategi reproduksi yang unik, dimana burung *Cuckoo* tersebut akan menaruh telurnya di sarang burung lain, baik burung yang memiliki jenis yang berbeda maupun pada sarang burung *Cuckoo* lainnya. Setiap burung *Cuckoo* hanya menaruh satu butir telurnya di sarang burung lain yang dipilih secara acak untuk meningkatkan kemungkinan menetas telur tersebut. Mungkin saja terjadi konflik antara burung *Cuckoo* dengan burung asli, sehingga pada saat setelah burung *Cuckoo* meletakkan telurnya pada sarang lain, burung asli (inang) akan membuang telur burung *Cuckoo* tersebut atau meninggalkan sarang yang telah diambil alih dan membuat sarang yang baru. Sarang burung yang menghasilkan generasi *Cuckoo* terbaik akan melanjutkan proses ke generasi berikutnya. Setiap pergantian generasi, jumlah dari pemilik sarang burung asli akan diatur kembali dan pemilik asli sarang ini mempunyai peluang untuk mengenali telur burung *Cuckoo* yang ditaruh di sarangnya. Jika terjadi kasus anak *Cuckoo* pendatang

lebih dahulu menetas, maka telur burung yang belum menetas akan dibuang dari sarang agar anak burung *Cuckoo* pendatang tersebut mendapat lebih banyak makanan. Dalam hal ini, bila pemilik asli sarang menemukan telur burung *Cuckoo*, maka pemilik itu dapat memilih untuk membuang telur tersebut ataupun meninggalkan sarangnya.

2.4.2 Karakteristik algoritma *Cuckoo Search*

Sesuai dengan uraian sebelumnya, penemu algoritma *Cuckoo search* Yang dan Deb dalam penelitiannya pada tahun 2009 juga mengungkapkan bahwa telah ditemukan beberapa aturan yang harus dipenuhi dalam penggunaan algoritma ini, diantaranya:

- a. Setiap burung meletakkan suatu telur pada satu waktu dan kemudian membuang telur pada sarang yang dipilih dengan acak;
- b. Sarang burung yang dianggap paling baik akan dilanjutkan untuk generasi selanjutnya;
- c. Jumlah sarang burung dalam satu koloni adalah tetap;
- d. Peluang untuk mengenali telur burung *Cuckoo* (P_a) yang ditaruh di sarang pemilik aslinya adalah 0 sampai 1.

Aturan terakhir dapat didekati dengan parameter P_a untuk menentukan solusi terburuk dari n sarang yang akan diganti dengan sarang baru secara acak. Untuk masalah maksimasi, guna mempermudah dapat digunakan representasi sederhana bahwa setiap telur dalam sarang menggambarkan sebuah solusi, dan telurnya merepresentasikan sebuah solusi baru. Tujuannya adalah untuk menggunakan solusi yang baru dan berpotensi lebih baik untuk menggantikan solusi pada sarang. Kemudian telur-telur pada sarang tersebut akan diseleksi dan dievolusikan dengan membuang telur-telur yang dianggap kurang baik. Pada beberapa keadaan, sarang induk asli bisa saja memiliki dua telur, sehingga dengan kata lain suatu sarang dapat berisikan lebih dari satu solusi. Namun untuk menyederhanakan permasalahan, suatu sarang hanya menyimpan satu solusi saja (Yang, 2010).

2.4.3 Mekanisme algoritma *Cuckoo Search*

Ketika sedang menghasilkan generasi baru $x^{(t+1)}$, maka akan digunakan proses pengacakan langkah dengan *Lévy Flights* yang dapat dilihat sebagai berikut

$$x_i^{(t+1)} = x_i^t + s \oplus \text{Lévy}(\alpha, \beta, \gamma, \delta) \quad (2.1)$$

di mana $x_i^{(t+1)}$ merupakan solusi baru, x_i^t adalah solusi lama, $s > 0$ merupakan ukuran langkah yang dikaitkan dengan tingkatan masalah yang dikerjakan dan tanda \oplus memiliki arti kali atau dikalikan. *Lévy flight* atau yang disebut *Lévy Stable Distribution* pada matlab memiliki beberapa paramter diantaranya:

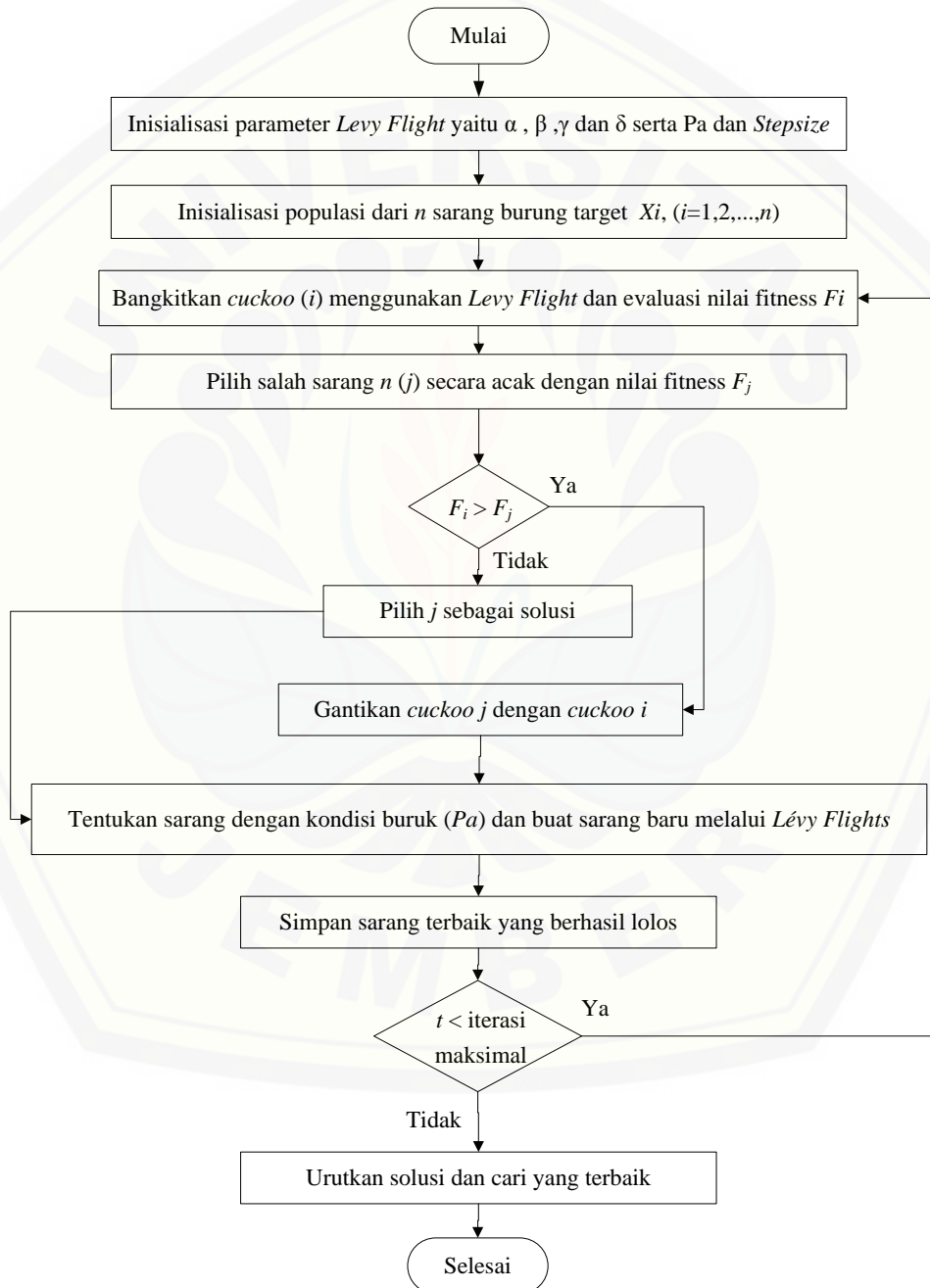
- Alpha(α), merupakan parameter bentuk pertama dengan selang $0 < \alpha \leq 2$.
- Betha(β), merupakan parameter bentuk kedua dengan selang $-1 \leq \beta \leq 1$.
- Gamma(γ), merupakan parameter skala yang memiliki selang $0 < \gamma < \infty$.
- Delta(δ), merupakan parameter lokasi yang memiliki selang $-\infty < \delta < \infty$.

Langkah-langkah menjalankan algoritma *Cuckoo Search* adalah sebagai berikut:

- Inisialisasi parameter *Lévy Flight* yaitu α , β , γ dan δ . Kemudian inisialisasi jumlah populasi dan banyaknya *job* dengan menggunakan nilai vektor acak, nilai *stepsiz*e yang akan digunakan, serta nilai parameter peluang (P_a).
- Pembangkitan solusi awal
 - Bangkitkan x secara acak menggunakan distribusi uniform
 - Lakukan proses diskritisasi (indeks)
 - Hitung nilai *makespan* pada setiap baris
 - Hitung nilai *fitness*
 - Solusi dengan *fitness* terbesar ditetapkan sebagai *Best-So-Far*
- Bangkitkan *Cuckoo* (i) secara acak dengan *Lévy Flight* berdasarkan *Best-So-Far* dan evaluasi nilai *fitness* F_i .
- Pilih salah sarang (j) secara acak dengan nilai fungsi *fitness* F_j .
- Jika dipenuhi $F_i > F_j$ gantikan *Cuckoo* j dengan *Cuckoo* i .
- Tentukan solusi terburuk dengan menggunakan parameter peluang (P_a) dan digantikan dengan solusi baru menggunakan *random walk* dari *Lévy Flight*.

- g. Simpan sarang terbaik yang berhasil lolos. Jika lebih baik dari *Best-So-Far*, maka akan menggantikan *Best-So-Far*.
- h. Jika generasi *Cuckoo* (t) mencapai generasi maksimal, maka *Cuckoo* dengan *fitness* terbesar adalah solusi akhir

Langkah-langkah diatas akan dijelaskan melalui Gambar 2.4



Gambar 2.4 Skema Algoritma *Cuckoo Search*

BAB 3. METODE PENELITIAN

3.1 Data

Data yang digunakan dalam penelitian ini adalah data sekunder yang diambil dari data penelitian Nurjanah et al. (2015) dan data simulasi. Data sekunder dari penelitian Nurjannah et al. (2015) adalah data dari industri pembuatan kerupuk “UD. Samjaya” yang terletak di desa Klatakan, kecamatan Tanggul Kabupaten Jember. Data tersebut terdiri dari 10 *job* dan 9 mesin. Sedangkan data simulasi merupakan data yang diambil dari penelitian Efendi et al. (2016) yang diperoleh dari *website* Prof. Éric Tailard, *University of Applied Sciences of Western Switzerland*. Data simulasi yang digunakan adalah 3 buah data, masing-masing terdiri dari 20 *job* 15 mesin, 30 *job* 15 mesin dan 50 *job* 20 mesin. Data-data tersebut tersaji dalam lampiran.

3.2 Langkah-langkah Penelitian

Pada penelitian ini, dilakukan langkah-langkah sistematis dalam pengerjaannya sehingga penelitian dapat dijalankan secara runtut dan terstruktur. Langkah-langkah yang akan dilakukan dalam penelitian tentang penerapan algoritma *Cuckoo Search* pada penjadwalan *flowshop*, secara skematik dapat dilihat pada Gambar 3.1.

Keterangan dari skema pada Gambar 3.1 untuk memperoleh hasil yang diinginkan adalah sebagai berikut:

- a. Studi literatur dengan mengumpulkan teori-teori yang berkaitan dengan algoritma *Cuckoo Search* melalui buku ataupun referensi jurnal dan artikel. Hal ini bertujuan agar lebih memahami berbagai teori yang digunakan dalam penelitian ini serta sebagai pedoman dalam pengerjaannya. Setelah melakukan studi literatur maka dilakukan pengambilan data untuk penjadwalan *flowshop*. Data yang digunakan adalah data sekunder dari penelitian Nurjannah et al. (2015) dan data simulasi dari penelitian Efendi et al. (2016) yang diperoleh dari *website* Prof. Éric Tailard, *University of Applied Sciences of Western Switzerland*.

b. Melakukan perhitungan secara manual dengan menjadwalkan produksi *flowshop* menggunakan algoritma *Cuckoo Search* berdasarkan data yang diperoleh dengan mencari nilai *makespan* terkecil. Adapun langkah-langkah penyelesaian *flowshop* menggunakan algoritma *Cuckoo Search* adalah sebagai berikut:

- 1) Inisialisasi beberapa parameter diantaranya parameter *Lévy* yaitu α , β , γ dan δ (yang selanjutnya disebut *Lévy* (α , β , γ , δ)), parameter *stepsize* dan parameter peluang (P_a).
- 2) Inisialisasi jumlah populasi dan banyaknya *job* dengan menggunakan nilai vektor acak. Dimana memiliki ketentuan jumlah populasi sebagai jumlah baris dan jumlah *job* sebagai jumlah kolom.
 - a) Bangkitkan x secara acak menggunakan distribusi uniform
 - b) Lakukan proses diskritisasi (indeks)
 - c) Hitung nilai *makespan* pada setiap baris
 - d) Hitung nilai *fitness*
 - e) Solusi dengan *fitness* terbesar ditetapkan sebagai *Best-So-Far*.

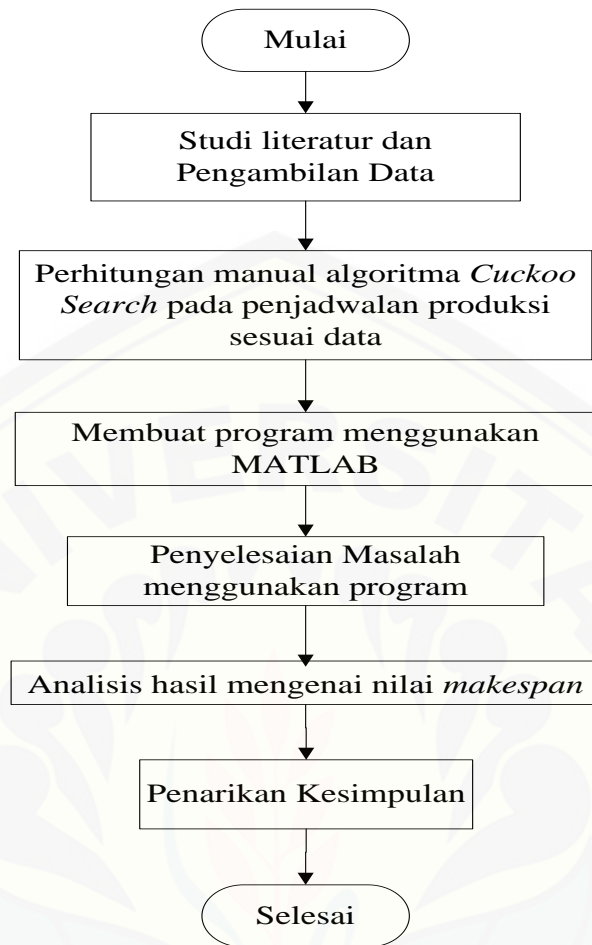
$$fitness = \frac{1}{makespan} \quad (3.1)$$

- 3) Bangkitkan *Cuckoo* i secara acak dan pilih sarang j secara acak, dengan ketentuan j tidak sama dengan i
 - a) Lakukan proses diskritisasi pada masing-masing *Cuckoo* dan sarang
 - b) Hitung nilai *makespan* pada setiap baris pada masing-masing *Cuckoo* dan sarang
 - c) Hitung nilai *fitness* (fungsi tujuan) pada masing-masing *Cuckoo* dan sarang
- 4) Bandingkan nilai fungsi *fitness* *Cuckoo* i dan *Cuckoo* j . Jika *fitness* *Cuckoo* i lebih dari *fitness* *Cuckoo* j , maka *Cuckoo* j sama dengan *Cuckoo* i (pergantian dilakukan pada setiap nilai x bilangan acak).
- 5) Urutkan nilai *fitness* dari yang terbesar (terbaik).
- 6) W sebagai banyaknya solusi terburuk ditentukan melalui persamaan

$$W = Pa * n \quad (3.2)$$

- 7) Solusi terburuk adalah sejumlah W yang dimulai dari urutan terbawah
 - 8) Lakukan perbaikan populasi dengan membangkitkan $L\hat{e}vy$ ($\alpha, \beta, \gamma, \delta$) secara acak. Gantikan solusi terburuk untuk memperoleh x yang baru dengan menggunakan persamaan (2.1)
 - 9) Evaluasi *fitness* dari masing-masing solusi yang diperoleh
 - 10) Jika generasi *Cuckoo* (t) mencapai generasi maksimal, maka *cuckoo* dengan *fitness* terbesar adalah solusi akhir.
- c. Pembuatan program menggunakan *software* Matlab sesuai dengan algoritma yang digunakan. Pada langkah ini penulis akan membuat skrip program dan desain tampilan pada program menggunakan GUI Matlab.
 - d. Penyelesaian masalah penjadwalan menggunakan program yang dilakukan dengan menerapkan data yang telah didapat pada langkah dalam program yang telah dibuat.
 - e. Menganalisis hasil yang telah dijalankan oleh program berdasarkan nilai *makespan* terkecil yaitu dengan melihat nilai *makespan* yang dihasilkan pada iterasi beberapa yang datanya menunjukkan nilai yang konstan.
 - f. Penarikan kesimpulan dengan menjawab rumusan masalah yang telah dibuat sebelumnya serta membuat saran-saran untuk penelitian selanjutnya.

Langkah-langkah penelitian permasalahan *flowshop* dengan menggunakan algoritma *Cuckoo Search* dapat dilihat pada gambar 3.1.



Gambar 3.1 Skema Langkah Penelitian

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan data hasil penelitian yang telah dilakukan diperoleh hasil sebagai berikut:

- a. Dari hasil pengujian empat data menggunakan algoritma *Cuckoo Search* diperoleh nilai *makespan* paling minimum adalah sebesar 8878 pada data 1, 1859 menit pada data 2, 2406 menit pada data 3 dan 4018 menit pada data 4. Hasil yang diperoleh menunjukkan bahwa algoritma *Cuckoo Search* lebih baik atau menghasilkan nilai *makespan* yang lebih minimum dibandingkan beberapa algoritma pembandingan yaitu algoritma Genetika, *Simulated Annaeling*, *Social Cognitive Optimization* dan *Iterated Greedy*.
- b. Data pengujian 6 parameter *Cuckoo Search* yaitu selang, populasi, *Lêvy*, *stepsize*, P_a (parameter peluang) dan iterasi maksimal menunjukkan hasil sebagai berikut:
 - 1) Semakin besar atau lebar selang yang dipilih maka nilai *makespan* yang dihasilkan akan semakin kecil.
 - 2) Semakin besar populasi yang dipilih maka nilai *makespan* yang dihasilkan akan semakin kecil.
 - 3) Semakin besar pula nilai P_a (parameter peluang) dan iterasi maksimal yang dipilih maka semakin kecil *makespan* yang dihasilkan.
 - 4) Nilai *makespan* yang terkecil dihasilkan oleh nilai *stepsize* terkecil.

Berdasarkan hasil penelitian yang dilakukan, pada data yang berukuran besar algoritma *Cuckoo Search* memiliki kelemahan yaitu membutuhkan iterasi maksimal yang besar sehingga mengakibatkan lamanya proses komputasi.

5.2 Saran

Peneliti selanjutnya diharapkan dapat melakukan pengembangan atau improvisasi algoritma *Cuckoo Search* dengan menambahkan metode heuristik

NEH yang diharapkan mampu memperoleh nilai *makespan* yang lebih optimum dalam menyelesaikan masalah penjadwalan. Selain itu, masih terbuka untuk menerapkan algoritma perilaku hewan lainnya.



DAFTAR PUSTAKA

- Baihaki, A., M. Z. Arif, dan A. Kamsyakawuni. 2016. Penerapan Algoritma Cuckoo Search (CS) Pada Permasalahan Multiple Constraints Knapsack 0-1. *Unej Jurnal*.**1**(1):1-6.
- Baker, K. R. 1974. *Introduction to Sequencing and Scheduling*. New York: John Wiley and sons.
- Bayu, H., Suryanto dan Saepudin, D. 2012. *Implementasi Cuckoo Search Algorithm Pada Travelling Salesman Problem Menggunakan Levy Flight*. Bandung: Fakultas Informatika Universitas Telkom.
- Conway, R.W., Maxwell, W.L dan Miller, L.W. 1967. *Theory of Scheduling*. USA : Addison-Wesley Publishing Company.
- Efendi, I., Kusbudiono, dan A. Kamsyakawuni. 2016. Penyelesaian Masalah Penjadwalan Flowshop dengan Algoritma Iterated Greedy dan Algoritma Social Cognitive Optimization. *Unej Jurnal*.**1**(1):1-6.
- Fister, I., Dusan, F., & Iztok, F. Jr. 2013. A Comprehensive Review Of Cuckoo Search: Variants and Hybrids. *Journal Mathematical Modelling and Numerical Optimisation*. **4**: 387-409.
- Gherboudj, A., Layeb, A., & Chikhi, S. 2012. Solving 0-1 knapsack problems by a discrete binary version of cuckoo search algorithm. *Journal of Bio-Inspired Computation*. **4**: 4.
- Ginting, R. 2009. *Penjadwalan Mesin*. Yogyakarta: Graha Ilmu.
- Liang, Y., & W. Chen, 2013. A survey on computing Levy stable distribution and new MATLAB toolbox. *Journal of signal processing*. **93**: 242–251.
- Nawas, M. 1983. A heuristic algorithm for the m -machine, n -job flow-shop sequencing problem. *Journal Omega*. **11**: 91-95 .
- Nurjanah, W. D., K. A. Santoso, dan A. Kamsyakawuni. 2015. Penerapan Algoritma Genetika dan Algoritma Simulated Annaeling Pada Penjadwalan Flowshop. *Unej Jurnal*.**1**(1):1-5.
- Santosa, B & Willy, P. 2011. *Metoda Metaheuristik: Konsep dan Implementasi*. Surabaya : Guna Widya.
- Vollman, T. E., Whybark, dan W. Lea Berry. 1998. *Manufacturing Planning & Control System*. 4thEdition. New York: McGraw-Hill Trade.

- Witjaksana, T. P., A. Kamsyakawuni, dan Kusbudiono. 2016. Penyelesaian Penjadwalan Flowshop Dengan Algoritma Coss Entropy-Genetic Algorithm dan Algoritma Difrential Evolution Plus. *Unej Jurnal*.1(1):1-7.
- Yang, X. S., & S. Deb. 2009. Cuckoo search via *Lévy flights*. *Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*. December 2009. *IEEE Publications*: 210-214.
- Yang, X. S., & S. Deb. 2013. Multiobjective cuckoo search for design optimization. *Computers & Operations Research*. **40**: 1616–1624.
- Yang, X. S., 2010, *Nature-Inspire Metaheuristic Algorithm*. Edisi 2. United Kingdom: Luniver Press
- Yildiz, A. R. 2013. Cuckoo Search Algorithm For The Selection Of Optimal Machining Parameters in Milling Operations. *Journal Adv Manuf Technol*. **64**: 55–61.

LAMPIRAN

Lampiran A. Data Penelitian

A.1 Data Produksi Kerupuk UD. Samjaya

Tabel 3.1 Data Waktu Pembuatan Kerupuk (per menit)

Mesin Job	M1	M2	M3	M4	M5	M6	M7	M8	M9
J1	15	160	15	30	30	1440	150	180	30
J2	15	160	15	30	30	1440	180	360	30
J3	15	160	15	30	30	360	85	180	30
J4	15	60	30	10	3	360	10	360	30
J5	15	60	30	10	3	300	30	180	30
J6	15	60	30	10	3	300	10	360	30
J7	15	60	30	10	3	720	30	180	30
J8	15	45	15	35	30	720	72	360	10
J9	15	45	15	35	30	1440	168	180	15
J10	15	75	36	25	17	1440	45	360	25

Keterangan :

J1 = Kerupuk impala panjang

J2 = kerupuk impala pendek

J3 = Kerupuk ranjang

J4 = Kerupuk mawar 2 udang besar

J5 = Kerupuk mawar 2 udang kecil

J6 = Kerupuk mawar SI besar

J7 = Kerupuk mawar SI kecil

J8 = Kerupuk kasandra besar

J9 = Kerupuk kasandra besar

J10= Kerupuk obar-abir

M1= Persiapan bahan

M2= Pengadonan

M3= Mesin mulen (pelembutan)

M4= Pencetakan

M5= Pengukusan

M6= Pendinginan

M7= Pemotongan (pembetulan)

M8= Penjemuran

M9= Pengemasan

A.2 Data Simulasi 20 Job 15 Mesin

JOB	MESIN														
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
J1	25	75	75	76	38	62	38	59	14	13	46	31	57	92	3
J2	67	5	11	11	40	34	77	42	35	96	22	55	21	29	16
J3	22	98	8	35	59	31	13	46	52	22	18	19	64	29	70
J4	99	42	2	35	11	92	88	97	21	56	17	43	27	19	23
J5	50	5	59	71	47	39	82	35	12	2	39	42	52	65	35
J6	48	57	5	2	60	64	86	3	51	26	34	39	45	63	54
J7	40	43	50	71	46	99	67	34	6	95	67	54	29	30	60
J8	59	3	85	6	46	49	5	82	18	71	48	79	62	65	76
J9	65	55	81	15	32	52	97	69	82	89	69	87	22	71	63
J10	70	74	52	94	14	81	24	14	32	39	67	59	18	77	50
J11	18	6	96	53	35	99	39	18	14	90	64	81	89	48	80
J12	44	75	12	13	74	59	71	75	30	93	26	30	84	91	93
J13	39	56	13	29	55	69	26	7	55	48	22	46	50	96	17
J14	57	14	8	13	95	53	78	24	92	90	68	87	43	75	94
J15	93	92	18	28	27	40	56	83	51	15	97	48	53	78	39
J16	47	34	42	28	11	11	30	14	10	4	20	92	19	59	28
J17	69	82	64	40	27	82	27	43	56	17	18	20	98	43	68
J18	84	26	87	61	95	23	88	89	49	84	12	51	3	44	20
J19	43	54	18	72	70	28	20	22	59	36	85	13	73	29	45
J20	7	97	4	22	74	45	62	95	66	14	40	23	79	34	8

A.3 Data Simulasi 30 Job 15 Mesin

JOB	MESIN														
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
J1	99	43	6	99	23	98	84	24	30	53	34	95	50	48	38
J2	19	24	65	16	94	9	60	32	59	85	9	36	22	25	5
J3	54	62	93	78	59	71	49	88	40	13	17	88	47	30	56
J4	60	16	79	84	84	42	59	14	74	60	98	17	42	31	19
J5	49	52	46	50	1	14	2	56	64	51	75	28	9	37	6
J6	59	65	85	40	23	39	99	46	17	94	6	67	69	86	8
J7	10	7	22	36	31	75	57	49	44	21	77	70	64	46	69
J8	53	74	93	26	54	89	82	66	37	63	71	17	58	4	46
J9	76	72	42	17	27	56	78	5	72	19	90	46	43	56	17
J10	18	79	93	71	48	23	20	90	94	87	6	36	84	25	83
J11	52	61	45	60	15	74	49	26	94	54	1	58	56	54	72
J12	63	73	82	84	15	54	52	52	36	21	45	41	21	97	50
J13	90	90	77	33	31	26	14	75	92	70	55	56	39	49	23
J14	87	45	58	34	29	83	24	48	97	89	84	82	53	99	10
J15	35	32	30	93	58	28	88	16	98	4	82	98	26	29	77
J16	18	92	62	59	3	94	34	56	24	18	66	53	30	41	10
J17	2	26	17	18	60	39	23	95	81	56	34	8	47	72	56
J18	6	79	65	58	94	45	80	3	29	80	27	60	94	14	76
J19	31	79	87	79	57	48	33	42	93	86	54	32	8	16	63
J20	96	1	75	42	45	51	10	58	71	92	23	18	63	27	63
J21	84	82	16	61	43	75	28	15	19	93	22	1	62	9	5
J22	46	29	50	12	72	18	79	73	23	1	58	1	95	25	71

JOB	MESIN														
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
J23	10	39	49	56	71	40	90	28	89	42	9	92	52	6	20
J24	70	63	68	97	86	81	38	7	53	48	43	59	88	29	87
J25	81	97	65	60	15	29	9	80	78	85	95	85	91	28	92
J26	39	6	59	34	34	32	12	7	35	4	53	69	89	3	40
J27	98	85	51	9	24	7	59	98	50	98	64	31	31	29	1
J28	59	68	3	8	2	9	69	14	72	84	69	54	45	59	7
J29	92	21	53	64	59	79	52	14	61	86	82	98	83	24	87
J30	51	70	94	80	35	56	8	94	11	3	60	73	26	21	45

A.4 Data Simulasi 30 Job 20 Mesin

JOB	MESIN																			
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20
J1	48	40	54	71	52	70	41	76	52	24	5	43	68	10	49	9	81	30	93	17
J2	85	18	54	42	41	71	68	82	54	49	21	1	58	1	69	58	40	59	66	29
J3	33	34	77	42	95	2	71	73	19	25	45	88	19	40	42	17	81	72	70	67
J4	51	41	74	97	26	4	25	12	17	76	6	79	49	39	1	27	44	75	1	18
J5	22	99	7	7	72	24	19	81	23	72	50	95	31	67	67	22	12	28	68	88
J6	52	51	44	38	64	11	62	20	54	15	83	79	55	48	38	37	42	81	89	60
J7	82	43	57	1	89	11	41	50	68	2	4	65	20	56	46	36	33	56	13	50
J8	45	11	63	59	69	39	44	61	67	72	74	59	16	26	90	66	56	47	95	39
J9	92	2	88	90	45	88	90	94	34	1	81	64	70	55	7	33	21	35	62	61
J10	89	21	61	18	77	20	42	59	79	12	56	14	21	43	89	31	71	92	47	71
J11	61	84	3	73	35	36	79	88	54	96	22	70	10	4	76	40	85	84	93	65
J12	68	72	74	97	63	33	96	4	63	31	1	98	39	65	72	20	7	63	33	26

JOB	MESIN																			
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20
J13	41	65	34	71	19	49	87	61	79	61	29	22	74	68	60	23	82	33	94	42
J14	17	40	40	28	6	62	83	95	44	91	79	39	68	79	1	20	96	62	62	70
J15	39	89	37	7	84	60	61	73	3	75	3	75	3	48	74	67	39	32	69	25
J16	9	83	30	3	31	93	86	49	34	91	56	80	33	37	35	63	72	46	22	73
J17	21	46	33	54	22	64	20	76	77	97	28	54	81	95	81	72	80	75	18	81
J18	52	30	38	70	22	15	66	26	55	34	13	65	87	38	85	89	77	22	67	44
J19	63	95	18	94	73	51	35	57	38	65	69	60	90	68	32	40	11	75	97	51
J20	68	37	90	13	76	77	6	6	53	41	72	71	46	24	46	50	12	39	92	54
J21	93	95	8	27	53	75	3	42	5	24	73	88	57	20	99	39	74	75	44	24
J22	83	14	66	96	11	96	20	5	72	38	79	10	27	27	90	8	83	10	61	69
J23	22	56	54	50	51	9	15	36	20	79	51	84	40	59	48	27	65	44	40	83
J24	5	75	43	17	10	92	22	36	7	71	77	70	10	24	78	77	56	42	16	48
J25	37	96	81	12	92	86	63	88	28	57	58	23	4	95	80	12	82	53	5	57
J26	58	59	65	78	68	50	38	97	72	94	59	42	5	19	27	54	69	2	56	51
J27	4	7	36	35	80	95	51	59	93	5	61	4	43	30	93	76	42	99	30	46
J28	88	75	81	40	61	94	78	24	19	44	96	23	90	94	80	97	24	44	64	52
J29	5	99	60	87	64	36	78	32	4	18	26	87	74	26	90	45	35	54	27	23
J30	93	95	11	14	99	86	41	26	50	74	21	6	67	87	46	84	11	89	89	66
J31	50	71	71	5	60	29	17	29	98	61	87	58	6	60	84	92	23	25	23	57
J32	75	60	77	48	87	52	98	8	55	97	55	68	59	90	50	98	57	43	72	35
J33	46	22	11	49	34	30	79	72	77	47	55	63	58	89	71	94	95	13	97	46
J34	25	98	71	68	8	72	57	39	83	17	90	31	81	6	97	98	82	82	52	82
J35	42	77	71	19	80	31	66	90	18	15	76	58	92	34	66	8	65	67	84	42
J36	41	42	69	81	95	16	45	52	48	35	72	80	81	4	3	4	96	53	14	80
J37	6	6	12	86	26	52	70	93	81	31	89	99	99	71	74	7	43	86	1	93

JOB	MESIN																			
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20
J38	44	54	36	40	68	49	45	58	44	65	72	65	53	48	90	98	60	7	27	48
J39	9	16	56	27	50	57	55	87	44	47	29	82	80	43	75	10	70	38	28	2
J40	29	91	85	51	86	34	73	12	14	51	1	38	74	92	60	43	36	23	82	30
J41	21	97	4	85	21	55	34	62	78	11	34	17	3	43	38	44	45	17	3	83
J42	29	6	45	15	60	29	97	91	13	8	50	46	72	86	7	30	28	13	27	42
J43	38	10	93	6	72	38	73	88	44	66	79	47	61	6	64	18	2	6	91	37
J44	21	20	51	96	51	42	52	37	85	18	44	60	68	3	6	20	81	96	30	9
J45	16	54	53	57	46	84	1	76	26	7	69	88	29	73	32	51	4	74	75	75
J46	27	54	90	25	97	68	41	54	29	14	8	1	60	13	16	41	81	35	18	79
J47	56	7	31	55	85	35	82	63	35	54	52	77	82	94	81	25	24	56	23	79
J48	33	50	22	70	59	51	80	84	47	88	27	18	34	47	4	41	56	42	26	66
J49	31	83	9	34	62	83	61	41	58	96	87	18	56	2	95	21	51	13	31	96
J50	62	95	8	3	27	19	36	97	87	62	86	21	37	11	11	67	84	34	48	97

Lampiran B. Hasil Uji Parameter *Lower Bound*(Lb) dan *Uper Bond*(Ub)B.1 Uji Parameter *Lower Bound*(Lb) dan *Uper Bond*(Ub) ($-4 < x < 4$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1907	2513	4185
2	8878	1910	2554	4199
3	8878	1906	2537	4194
4	8878	1920	2548	4182
5	8878	1924	2527	4193
6	8878	1928	2545	4190
7	8878	1915	2535	4184
8	8878	1907	2534	4192
9	8878	1887	2538	4175
10	8878	1924	2541	4195

B.2 Uji Parameter *Lower Bound*(Lb) dan *Uper Bond*(Ub) ($0 < x < 4$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1894	2542	4207
2	8878	1921	2538	4197
3	8878	1910	2544	4192
4	8878	1925	2546	4204
5	8878	1919	2547	4194
6	8878	1924	2534	4186
7	8878	1908	2524	4180
8	8878	1921	2548	4195
9	8878	1917	2539	4185
10	8878	1916	2538	4189

B.3 Uji Parameter *Lower Bound*(Lb) dan *Uper Bond*(Ub) ($-1 < x < 1$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1916	2553	4191
2	8878	1912	2549	4178
3	8878	1922	2543	4164
4	8878	1906	2551	4188
5	8878	1924	2539	4185
6	8878	1903	2543	4191
7	8878	1913	2549	4193
8	8878	1899	2540	4202
9	8878	1909	2542	4185
10	8878	1917	2546	4204

B.4 Uji Parameter *Lower Bound*(Lb) dan *Uper Bond*(Ub) ($0 < x < 1$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1908	2545	4195
2	8878	1927	2536	4196
3	8878	1921	2554	4191
4	8878	1908	2539	4201
5	8878	1920	2535	4193
6	8878	1908	2539	4197
7	8878	1912	2542	4189
8	8878	1913	2537	4191
9	8878	1907	2523	4181
10	8878	1925	2538	4194

Lampiran C. Hasil Uji Parameter Populasi

C.1 Uji Parameter Populasi (Pop=20)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1912	2562	4208
2	8878	1940	2561	4206
3	8878	1914	2555	4229
4	8878	1929	2553	4203
5	8878	1917	2536	4215
6	8878	1921	2574	4216
7	8878	1910	2565	4184
8	8878	1922	2582	4192
9	8878	1919	2556	4197
10	8878	1930	2546	4193

C.2 Uji Parameter Populasi (Pop=50)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1917	2564	4178
2	8878	1911	2556	4201
3	8878	1908	2547	4197
4	8878	1936	2545	4214
5	8878	1911	2552	4202
6	8878	1915	2541	4218
7	8878	1918	2563	4205
8	8878	1920	2543	4221
9	8878	1912	2550	4211
10	8878	1916	2532	4189

C.3 Uji Parameter Populasi (Pop=100)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1913	2539	4198
2	8878	1930	2550	4195
3	8878	1917	2545	4197
4	8878	1920	2550	4194
5	8878	1913	2555	4197
6	8878	1922	2533	4162
7	8878	1914	2552	4189
8	8878	1916	2530	4171
9	8878	1903	2558	4219
10	8878	1924	2533	4199

C.4 Uji Parameter Populasi (Pop=200)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1915	2544	4189
2	8878	1912	2546	4191
3	8878	1919	2550	4199
4	8878	1906	2544	4193
5	8878	1902	2543	4199
6	8878	1917	2532	4186
7	8878	1906	2540	4202
8	8878	1920	2541	4183
9	8878	1920	2556	4181
10	8878	1904	2523	4172

Lampiran D. Hasil Uji Parameter Levy

D.1 Uji Parameter Levy ($\alpha \beta \gamma \delta$) (1,4, 0,8, 1,3, 0,7)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1925	2538	4182
2	8878	1912	2550	4173
3	8878	1901	2550	4184
4	8878	1927	2544	4177
5	8878	1915	2544	4184
6	8878	1925	2549	4198
7	8878	1909	2548	4197
8	8878	1921	2543	4172
9	8878	1914	2552	4193
10	8878	1932	2543	4198

D.2 Uji Parameter Levy ($\alpha \beta \gamma \delta$) (1,4, 0,9, 1,-0,5)

Percobaan ke	Makespan			
	Data 1	Data 2	Data 3	Data 4
1	8878	1918	2541	4199
2	8878	1914	2515	4195
3	8878	1916	2538	4194
4	8878	1914	2540	4206
5	8878	1895	2543	4199
6	8878	1903	2545	4187
7	8878	1914	2545	4184
8	8878	1911	2553	4165
9	8878	1922	2533	4196
10	8878	1919	2546	4184

D.3 Uji Parameter Levy ($\alpha \beta \gamma \delta$) (0,1, 0, 1,3, 0,7)

Percobaan ke	Makespan			
	Data 1	Data 2	Data 3	Data 4
1	8878	1926	2522	4171
2	8878	1919	2525	4192
3	8878	1911	2521	4189
4	8878	1934	2531	4182
5	8878	1922	2540	4206
6	8878	1909	2541	4181
7	8878	1909	2541	4184
8	8878	1916	2545	4183
9	8878	1922	2542	4183
10	8878	1915	2538	4203

D.4 Uji Parameter Levy ($\alpha \beta \gamma \delta$) (0,001, 0,8, 1,3, 0,7)

Percobaan ke	Makespan			
	Data 1	Data 2	Data 3	Data 4
1	8878	1936	2526	4372
2	8878	1949	2665	4359
3	8878	1950	2665	4392
4	8878	1946	2664	4352
5	8878	1951	2647	4376
6	8878	1938	2632	4335
7	8878	1948	2652	4382
8	8878	1938	2667	4359
9	8878	1957	2633	4365
10	8878	1951	2606	4410

Lampiran E. Hasil Uji Parameter *Stepsize*E.1 Uji Parameter *Stepsize* ($s = 0,1$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1916	2527	4191
2	8878	1906	2547	4174
3	8878	1894	2536	4180
4	8878	1920	2548	4181
5	8878	1924	2545	4196
6	8878	1903	2537	4183
7	8878	1910	2521	4198
8	8878	1909	2542	4164
9	8878	1908	2526	4185
10	8878	1908	2538	4187

E.2 Uji Parameter *Stepsize* ($s = 0,25$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1926	2522	4171
2	8878	1919	2525	4192
3	8878	1911	2521	4189
4	8878	1934	2531	4182
5	8878	1922	2540	4206
6	8878	1909	2541	4181
7	8878	1909	2541	4184
8	8878	1916	2545	4183
9	8878	1922	2542	4183
10	8878	1915	2538	4203

E.3 Uji Parameter *Stepsize* ($s = 1$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1908	2533	4170
2	8878	1915	2540	4185
3	8878	1911	2543	4195
4	8878	1908	2534	4159
5	8878	1921	2539	4193
6	8878	1907	2539	4189
7	8878	1918	2507	4197
8	8878	1919	2540	4196
9	8878	1907	2527	4189
10	8878	1911	2535	4195

E.4 Uji Parameter *Stepsize* ($s = 2$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1927	2546	4197
2	8878	1920	2535	4197
3	8878	1918	2538	4197
4	8878	1924	2545	4188
5	8878	1933	2533	4188
6	8878	1923	2552	4193
7	8878	1922	2542	4194
8	8878	1905	2541	4179
9	8878	1922	2540	4189
10	8878	1915	2541	4196

Lampiran F. Hasil Uji Parameter P_a F.1 Uji Parameter P_a ($P_a = 0,25$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1911	2565	4243
2	8878	1921	2573	4245
3	8878	1930	2570	4244
4	8878	1916	2568	4251
5	8878	1907	2582	4251
6	8878	1932	2581	4259
7	8878	1919	2582	4258
8	8878	1933	2580	4248
9	8878	1901	2579	4216
10	8878	1937	2554	4250

F.2 Uji Parameter P_a ($P_a = 0,5$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1916	2527	4191
2	8878	1906	2547	4174
3	8878	1894	2536	4180
4	8878	1920	2548	4181
5	8878	1924	2545	4196
6	8878	1903	2537	4183
7	8878	1910	2521	4198
8	8878	1909	2542	4164
9	8878	1908	2526	4185
10	8878	1908	2538	4187

F.3 Uji Parameter P_a ($P_a = 0,75$)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1881	2513	4149
2	8878	1889	2499	4151
3	8878	1904	2514	4148
4	8878	1887	2505	4143
5	8878	1882	2495	4150
6	8878	1884	2507	4159
7	8878	1876	2489	4150
8	8878	1878	2515	4141
9	8878	1872	2509	4134
10	8878	1887	2507	4149

Lampiran G. Hasil Uji Parameter Iterasi Maksimal

G.1 Uji Parameter Iterasi Maksimal (MaxGen = 5000)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1883	2463	4071
2	8878	1868	2470	4078
3	8878	1868	2453	4076
4	8878	1862	2472	4066
5	8878	1864	2474	4072
6	8878	1876	2465	4071
7	8878	1881	2464	4073
8	8878	1867	2460	4063
9	8878	1885	2473	4072
10	8878	1870	2485	4076

G.2 Uji Parameter Iterasi Maksimal (MaxGen = 10000)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1882	2467	4048
2	8878	1869	2464	4057
3	8878	1866	2473	4052
4	8878	1868	2466	4052
5	8878	1862	2467	4064
6	8878	1864	2464	4062
7	8878	1872	2461	4055
8	8878	1879	2449	4057
9	8878	1877	2442	4060
10	8878	1859	2453	4057

G.3 Uji Parameter Iterasi Maksimal (MaxGen = 50000)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1872	2406	4055
2	8878	1862	2447	4060
3	8878	1864	2406	4044
4	8878	1859	2447	4018
5	8878	1864	2434	4060
6	8878	1866	2458	4020
7	8878	1860	2432	4032
8	8878	1872	2441	4030
9	8878	1862	2430	4024
10	8878	1864	2450	4056

G.4 Uji Parameter Iterasi Maksimal (MaxGen = 75000)

Percobaan ke	<i>Makespan</i>			
	Data 1	Data 2	Data 3	Data 4
1	8878	1860	2447	4018
2	8878	1879	2406	4028
3	8878	1866	2458	4032
4	8878	1872	2411	4056
5	8878	1864	2432	4026
6	8878	1868	2425	4024
7	8878	1862	2410	4032
8	8878	1872	2431	4021
9	8878	1859	2445	4020
10	8878	1878	2421	4034