



**PENYELESAIAN PENJADWALAN *FLWSHOP*
DENGAN ALGORITMA *CROSS ENTROPY-GENETIC ALGORITHM*
DAN ALGORITMA *DIFFERENTIAL EVOLUTION PLUS***

SKRIPSI

Oleh
Teddy Prawira Witjaksana
NIM 121810101009

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



**PENYELESAIAN PENJADWALAN *FLWSHOP*
DENGAN ALGORITMA *CROSS ENTROPY-GENETIC ALGORITHM*
DAN ALGORITMA *DIFFERENTIAL EVOLUTION PLUS***

SKRIPSI

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

Oleh
Teddy Prawira Witjaksana
NIM 121810101009

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

2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ibunda Tatik Suratmi dan Ayahanda Nurhadi yang memberikan kasih sayang, motivasi, segala dukungan dan doa yang tiada henti;
2. Kakekku Alfandi dan Nenekku Siti Ainun yang memberikan kasih sayang, motivasi dan doa yang tiada henti;
3. Kakakku Taufiq Hidayat, Farida Malina Budiani, Deddy Prasetyo Nugroho, Septi Ganjar Prima Sari, Darwanto, Tari Astuti, Donny Setiawan, Fitria Hasyim beserta keluarga atas bantuan serta doanya;
4. Calon istriku Dita Aljelia yang memberikan kasih sayang, doa, motivasi dan inspirasinya;
5. seluruh guru sejak sekolah dasar sampai perguruan tinggi yang senantiasa membimbing dan memberikan ilmu yang bermanfaat;
6. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember, SMA Muhammadiyah 3 Jember, SMPN 11 Jember, SDN Karangrejo II Jember;
7. segenap keluarga BATHICS '12 yang kusayangi.

MOTTO

Dan mohonlah pertolongan dengan sabar dan salat; dan sesungguhnya salat itu berat, kecuali atas orang-orang yang khusyu', (yaitu) orang-orang yang meyakini bahwa mereka akan menjumpai Tuhan mereka dan sesungguhnya mereka akan kembali kepada-Nya.

(terjemahan Surat Al – Baqarah Ayat 45 – 46)*)

Maka sesungguhnya beserta kesukaran ada kemudahan, sesungguhnya beserta kesukaran ada kemudahan. Maka apabila engkau telah selesai (dari suatu urusan), maka kerjakanlah (urusan yang lain) dengan sungguh-sungguh, dan hanya kepada Tuhanmu hendaklah engkau berharap.

(terjemahan Surat Al – Insyiraah Ayat 5 – 8)*)

*)Tim Disbintalad. 2008. *Al Quran Terjemah Indonesia*. Jakarta: Suara Agung

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Teddy Prawira Witjaksana

NIM : 121810101009

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul **“Penyelesaian Penjadwalan *Flowshop* dengan Algoritma *Cross Entropy-Genetic Algorithm* dan Algoritma *Differential Evolution Plus*”** adalah benar-benar karya saya 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, Juni 2016

Yang menyatakan,

Teddy Prawira Witjaksana

NIM 121810101009

SKRIPSI

**PENYELESAIAN PENJADWALAN *FLOWSHOP*
DENGAN ALGORITMA *CROSS ENTROPY-GENETIC ALGORITHM*
DAN ALGORITMA *DIFFERENTIAL EVOLUTION PLUS***

Oleh
Teddy Prawira Witjaksana
NIM 121810101009

Pembimbing

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

PENGESAHAN

Skripsi berjudul “**Penyelesaian Penjadwalan *Flowshop* dengan Algoritma *Cross Entropy-Genetic Algorithm* dan Algoritma *Differential Evolution Plus*”** telah diuji dan disahkan oleh Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember pada:

Hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas
Jember.

Tim Penguji

Dosen Pembimbing Utama,

Dosen Pembimbing Anggota,

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

Kusbudiono, S.Si., M.Si
NIP. 197704302005011001

Anggota Tim Penguji

Dosen Penguji I

Dosen Penguji II

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

Prof. Drs. Kusno, DEA., Ph.D
NIP. 196101081986021001

Mengesahkan,
Dekan,

Drs. Sujito, Ph.D
NIP. 196102041987111001

RINGKASAN

Penyelesaian Penjadwalan *Flowshop* dengan Algoritma *Cross Entropy-Genetic Algorithm* dan Algoritma *Differential Evolution Plus*; Teddy Prawira Witjaksana, 121810101009; 2016; 77 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penjadwalan merupakan pengalokasian sumber daya yang terbatas untuk mengerjakan sejumlah pekerjaan. Penjadwalan produksi memiliki tujuan untuk meminimasi *makespan* (total waktu yang dibutuhkan untuk menyelesaikan seluruh pekerjaan). Salah satu jenis penjadwalan produksi yaitu penjadwalan *flowshop*. Penjadwalan *flowshop* merupakan model penjadwalan dimana *job-job* yang akan diproses seluruhnya mengalir pada arah atau jalur produk yang sama.

Pada penelitian ini digunakan data simulasi yang diperoleh dari *website OR-Library*. Data tersebut merupakan data *flowshop* yang diformulasi oleh J. Carlier (1978) dan C. R. Reeves (1995) yang terdiri dari 7 data. Penjadwalan *flowshop* pada 7 data simulasi tersebut menggunakan algoritma *Cross Entropy-Genetic Algorithm* (CEGA) dan algoritma *Differential Evolution Plus* (DE Plus). Setelah dilakukan penjadwalan pada 7 data *flowshop*, dapat diketahui *makespan* minimum, tingkat konvergensi, dan *running time* dari dua algoritma tersebut.

Penerapan algoritma CEGA dan algoritma DE Plus pada 7 data *flowshop* memberikan hasil bahwa 4 dari 7 data *flowshop* yang telah diselesaikan, nilai *makespan* minimum yang dihasilkan kedua algoritma tersebut sama. Sedangkan untuk 3 data yang lain, nilai *makespan* minimum yang dihasilkan algoritma CEGA dan algoritma DE Plus berbeda. Selain itu pada data *car6* (8 Job, 9 Mesin), *car7* (7 Job, 7 Mesin), *car8* (8 Job, 8 Mesin), rata-rata *makespan* yang dihasilkan algoritma CEGA dan DE Plus sama. Sehingga dapat dikatakan bahwa pada data *car6* (8 Job, 9 Mesin), *car7* (7 Job, 7 Mesin), *car8* (8 Job, 8 Mesin) algoritma CEGA dan algoritma DE Plus sama-sama baik. Sedangkan pada data *reC07* (20 Job, 10 Mesin), *reC13* (20 Job, 15 Mesin), *reC19* (30 Job, 10 Mesin), dan *reC27* (30 Job, 15 Mesin) rata-rata *makespan* yang dihasilkan algoritma DE Plus lebih

kecil dibandingkan algoritma CEGA. Sehingga dapat dikatakan bahwa pada data *reC07* (20 *Job*, 10 Mesin), *reC13* (20 *Job*, 15 Mesin), *reC19* (30 *Job*, 10 Mesin), dan *reC27* (30 *Job*, 15 Mesin) algoritma *DE Plus* lebih baik dibandingkan algoritma CEGA.

Pada penelitian tingkat konvergensi algoritma CEGA dan algoritma *DE Plus*, diperoleh hasil bahwa pada kasus data *car6* (8 *Job*, 9 Mesin), *car7* (7 *Job*, 7 Mesin), *car8* (8 *Job*, 8 Mesin) algoritma *DE Plus* lebih cepat konvergen dibandingkan algoritma CEGA. Sedangkan pada kasus data *reC07* (20 *Job*, 10 Mesin), *reC13* (20 *Job*, 15 Mesin), *reC19* (30 *Job*, 10 Mesin), algoritma CEGA lebih cepat konvergen dibandingkan algoritma *DE Plus*. Selain itu, *running time* yang dihasilkan algoritma CEGA untuk menyelesaikan seluruh data lebih cepat dibandingkan algoritma *DE Plus*.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penyelesaian Penjadwalan *Flowshop* dengan Algoritma *Cross Entropy-Genetic Algorithm* dan Algoritma *Differential Evolution Plus*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada jurusan Matematika Fakultas MIPA Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Drs. Sujito, Ph.D. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam yang telah mengesahkan penulisan skripsi ini;
2. Ahmad Kamsyakawuni, S.Si., M.Kom. selaku Dosen Pembimbing Utama dan Kusbudiono, S.Si., M.Si. selaku Dosen Pembimbing Anggota yang telah memberikan bimbingan serta arahan pada penulis sehingga skripsi ini dapat terselesaikan dengan baik;
3. M. Ziaul Arif, S.Si., M.Sc. dan Prof. Drs. Kusno, DEA., Ph.D. selaku Dosen Penguji yang telah memberikan kritik dan saran sehingga skripsi ini menjadi lebih baik lagi;
4. Ibunda Tatik Suratmi, Ayahanda Nurhadi, Bude Endang Sumarni, Pakde Agus Manto, Pakde Mudji Slamet Santoso beserta keluarga besar yang selalu memberikan semangat serta doa;
5. orang-orang yang ikut andil membantu terselesaikannya tugas akhir ini; Tanty Kurnia Sari, Dian Setiya Widodo, Arif Riyanto, Vina F. Martin, dan seluruh rekan-rekan BATHICS '12 yang telah menemani dan memberikan semangat;
6. semua pihak yang tidak dapat saya sebutkan satu per satu.

Penulis juga menerima segala kritik serta saran demi kesempurnaan skripsi ini. Penulis berharap skripsi ini dapat bermanfaat bagi pembaca. Amin.

Jember, Juni 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	iv
HALAMAN PERNYATAAN.....	v
HALAMAN PEMBIMBING	vi
HALAMAN PENGESAHAN.....	vii
RINGKASAN	viii
PRAKATA	x
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	3
1.5 Manfaat	3
BAB 2. TINJAUAN PUSTAKA.....	4
2.1 Definisi Penjadwalan	4
2.2 Penjadwalan <i>Jobshop</i>.....	5
2.3 Penjadwalan <i>Flowshop</i>	6
2.4 Diagram <i>Gantt</i>	7
2.5 Algoritma <i>Cross Entropy-Genetic Algorithm (CEGA)</i>	8
2.6 Algoritma <i>Differential Evolution Plus (DE Plus)</i>.....	17
2.7 Kriteria Kekonvergenan.....	22
BAB 3. METODE PENELITIAN.....	24
3.1 Data Penelitian	24

3.2 Langkah-langkah Penelitian	24
BAB 4. HASIL DAN PEMBAHASAN	27
4.1 Hasil	27
4.1.1 Perhitungan Manual	27
4.1.2 Program Penjadwalan <i>Flowshop</i> menggunakan <i>MATLAB</i>	41
4.2 Pembahasan	46
4.2.1 Penerapan Algoritma CEGA dan Algoritma <i>DE Plus</i> Pada Data <i>Flowshop</i>	46
4.2.2 Tingkat Konvergensi Algoritma CEGA dan Algoritma <i>DE Plus</i>	50
4.2.3 <i>Running Time</i> Algoritma CEGA dan Algoritma <i>DE Plus</i> ..	52
BAB 5. PENUTUP	53
5.1 Kesimpulan	53
5.2 Saran	54
DAFTAR PUSTAKA	55
LAMPIRAN	57

DAFTAR TABEL

	Halaman
2.1 Bagian dari Induk 1 dan Induk 2 yang akan mengalami <i>crossover</i> menggunakan metode <i>2-point order crossover</i>	15
2.2 Calon Anak 1 dan Anak 2 hasil metode <i>2-point order crossover</i>	15
2.3 Anak 1 dan Anak 2 hasil metode <i>2-point order crossover</i>	15
2.4 Contoh <i>Swap Mutation</i>	16
2.5 Contoh <i>Flip Mutation</i>	16
2.6 Contoh <i>Slide Mutation</i>	16
4.1 Waktu proses produksi dengan empat <i>job</i> dan tiga mesin (menit)	27
4.2 Perhitungan <i>Makespan</i> Sampel $X_1 = 3 - 2 - 1 - 4$	28
4.3 Rekapitulasi Perngurutan <i>Makespan</i>	29
4.4 Penentuan <i>Crossover</i>	32
4.5 Populasi Sampel yang Baru	33
4.6 Penentuan Mutasi	34
4.7 Populasi Sampel Setelah Mutasi	35
4.8 Rekapitulasi Perhitungan <i>Makespan</i>	35
4.9 Jumlah <i>Job</i> dan Jumlah Mesin Setiap Data	46
4.10 <i>Makespan</i> Optimal Seluruh Data <i>Flowshop</i>	47
4.11 Rata-rata <i>Makespan</i> Setiap 10 Kali Percobaan Seluruh Data	48
4.12 Rata-rata Iterasi Konvergen Setiap 10 Kali Percobaan Seluruh Data...	50
4.13 Rata-rata <i>Running Time</i> Setiap 10 Kali Percobaan Seluruh Data	52

DAFTAR GAMBAR

	Halaman
2.1 Aliran <i>job</i> dapat melompat ke mesin selanjutnya	5
2.2 Aliran <i>job</i> dapat diproses kembali pada mesin yang telah digunakan	6
2.3 Pola aliran <i>pure flowshop</i>	6
2.4 Pola aliran <i>general flowshop</i>	7
2.5 Diagram <i>Gantt</i>	8
2.6 Skema Algoritma <i>Cross Entropy-Genetic Algorithm (CEGA)</i>	17
2.7 Skema Algoritma <i>Differential Evolution Plus (DE Plus)</i>	22
3.1 Skema Langkah-langkah Penelitian	26
4.1 Tampilan Awal Program Penjadwalan <i>Flowshop</i>	42
4.2 Tampilan Tabel Data	43
4.3 Tampilan Input Parameter Kedua Algoritma	44
4.4 Tampilan Output Kedua Algoritma	44
4.5 Tampilan <i>Gantt Chart</i> CEGA Untuk Data <i>car6</i>	45
4.6 Tampilan <i>Gantt Chart</i> DE Plus Untuk Data <i>car6</i>	45

DAFTAR LAMPIRAN

	Halaman
LAMPIRAN A Data Penelitian.....	57
LAMPIRAN B Hasil Percobaan untuk Data <i>car6</i> (2000 Iterasi)	64
LAMPIRAN C Hasil Percobaan untuk Data <i>car7</i> (2000 Iterasi)	66
LAMPIRAN D Hasil Percobaan untuk Data <i>car8</i> (2000 Iterasi)	68
LAMPIRAN E Hasil Percobaan untuk Data <i>reC07</i> (2000 Iterasi).....	70
LAMPIRAN F Hasil Percobaan untuk Data <i>reC13</i> (2000 Iterasi)	72
LAMPIRAN G Hasil Percobaan untuk Data <i>reC19</i> (2000 Iterasi)	74
LAMPIRAN H Hasil Percobaan untuk Data <i>reC27</i> (2000 Iterasi)	76

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Dunia usaha saat ini seperti sektor industri kecil, industri menengah, maupun industri berskala besar memiliki tingkat persaingan yang ketat. Persaingan yang ketat tersebut dibuktikan dengan banyaknya perusahaan yang meningkatkan kinerjanya dengan cara perbaikan struktur organisasi dan manajemen, sumber daya manusia, serta aplikasi di bidang sistem informasi. Selain itu, setiap perusahaan juga memikirkan strategi agar produk yang dihasilkan sesuai harapan konsumen serta setiap pesanan yang diterima dapat diselesaikan tepat waktu. Strategi yang bisa diterapkan setiap perusahaan yaitu melakukan perencanaan dan pengendalian dalam menjalankan usaha dengan memperhatikan sumber daya yang dimiliki seperti fasilitas, tenaga kerja dan bahan baku. Salah satu perencanaan dan pengendalian tersebut dengan melakukan penjadwalan.

Penjadwalan merupakan pengalokasian sumber daya yang terbatas untuk mengerjakan sejumlah pekerjaan. 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. Dengan adanya pengurutan tersebut dapat memanfaatkan unit-unit produksi secara optimum (Ginting, 2009). Penjadwalan produksi memiliki tujuan untuk meminimasi *makespan* (total waktu yang dibutuhkan untuk menyelesaikan seluruh pekerjaan). Penjadwalan produksi dapat dibedakan menjadi penjadwalan *jobshop* dan penjadwalan *flowshop*. Penjadwalan produksi *flowshop* merupakan suatu pergerakan unit-unit yang terus-menerus melalui suatu rangkaian stasiun-stasiun kerja yang disusun berdasarkan produk (Baker, 1974).

Peneliti sebelumnya telah mengembangkan metode metaheuristik untuk menyelesaikan masalah optimasi penjadwalan *flowshop*. Metaheuristik merupakan metode untuk mencari solusi yang memadukan interaksi antara prosedur pencarian lokal dan strategi yang lebih tinggi untuk menciptakan proses yang

mampu keluar dari titik-titik lokal optima dan melakukan pencarian di ruang solusi untuk menemukan solusi global (Santosa dan Willy, 2011). Metode Metaheuristik terdiri dari beberapa algoritma, diantaranya *Genetic Algorithm*, *Simulated Annealing*, *Tabu Search*, *Ant Colony Optimization*, *Particle Swarm Optimization*, *Differential Evolution* dan *Cross Entropy*.

Pada penelitian Widodo (2014) yang menggunakan algoritma *Cross Entropy-Genetic Algorithm* (CEGA) pada penjadwalan *flowshop* (4 job, 4 mesin) menunjukkan bahwa nilai *makespan* dari algoritma *Cross Entropy-Genetic Algorithm* (CEGA) lebih baik dari nilai *makespan* yang dihasilkan menggunakan metode perusahaan serta waktu komputasi yang dihasilkan algoritma *Cross Entropy-Genetic Algorithm* (CEGA) sangat cepat yaitu rata-rata kurang dari satu detik. Widodo (2014) merancang algoritma *Cross Entropy-Genetic Algorithm* (CEGA) dengan cara menggabungkan algoritma *Cross Entropy* (CE) dan *Genetic Algorithm* (GA) yang memiliki mekanisme *crossover* dan mutasi yang dapat menghasilkan kromosom baru. Keunggulan algoritma *Cross Entropy-Genetic Algorithm* (CEGA) yaitu dapat menghindari kemungkinan pencarian solusi terjebak di area lokal optimal sehingga akan didapatkan hasil yang optimal.

Pada penelitian Sari (2015) tentang penjadwalan *flowshop* menunjukkan bahwa algoritma *Differential Evolution Plus* memiliki tingkat kecepatan kekonvergenan yang lebih baik dibandingkan dengan algoritma *Simulated Annealing*. Pada penelitian Dewi (2015) tentang penyelesaian penjadwalan *flowshop* menunjukkan bahwa algoritma *Differential Evolution Plus* lebih efektif dibandingkan dengan algoritma *Artificial Immune System*. Serta pada penelitian Sa'dia (2015) tentang penjadwalan *flowshop* menunjukkan bahwa algoritma *Differential Evolution Plus* memiliki tingkat kecepatan kekonvergenan yang lebih baik dibandingkan dengan algoritma *Artificial Bee Colony*.

Berdasarkan penjelasan diatas, penulis tertarik untuk menerapkan algoritma *Cross Entropy-Genetic Algorithm* dan algoritma *Differential Evolution Plus* pada penjadwalan *flowshop* untuk memperoleh solusi yang optimal dan menentukan algoritma yang memberikan hasil lebih baik berdasarkan aspek *makespan*, *running time* (waktu komputasi) dan tingkat kecepatan kekonvergenan.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas dalam skripsi ini adalah sebagai berikut.

- a. Bagaimana penerapan algoritma *Cross Entropy-Genetic Algorithm* dan algoritma *Differential Evolution Plus* pada penjadwalan *flowshop*?
- b. Bagaimana perbandingan hasil kedua algoritma berdasarkan aspek *makespan*, *running time* (waktu komputasi), dan tingkat kecepatan kekonvergenan pada penjadwalan *flowshop*?

1.3 Batasan Masalah

Batasan-batasan masalah yang terdapat dalam skripsi ini adalah sebagai berikut.

- a. Bahan baku yang digunakan selalu tersedia;
- b. Setiap *job* memiliki *ready time* yang sama;
- c. Mesin berjalan dengan normal (tanpa gangguan);
- d. Waktu membawa *job* ke mesin lainnya diabaikan.

1.4 Tujuan

Skripsi ini memiliki tujuan dalam penulisannya. Adapun tujuan tersebut adalah sebagai berikut.

- a. Menerapkan algoritma *Cross Entropy-Genetic Algorithm* dan algoritma *Differential Evolution Plus* pada penjadwalan *flowshop*;
- b. Mengetahui perbandingan hasil dari kedua algoritma berdasarkan aspek *makespan*, *running time* (waktu komputasi), dan tingkat kecepatan kekonvergenan.

1.5 Manfaat

Adapun manfaat yang dapat diperoleh dari penulisan skripsi ini adalah sebagai berikut.

- a. Mendapatkan solusi yang optimal yaitu nilai *makespan* minimum;
- b. Memperoleh pemahaman mengenai algoritma *Cross Entropy-Genetic Algorithm* dan algoritma *Differential Evolution Plus*.

BAB 2. TINJAUAN PUSTAKA

2.1 Definisi Penjadwalan

Penjadwalan adalah pengalokasian sumber daya yang terbatas untuk mengerjakan sejumlah pekerjaan (Ginting, 2009). Sedangkan menurut Baker (1974), penjadwalan (*scheduling*) didefinisikan sebagai proses pengalokasian sumber daya untuk memilih sekumpulan tugas dalam jangka waktu tertentu. Keputusan yang dibuat dalam penjadwalan meliputi pengurutan pekerjaan (*sequencing*), waktu mulai dan selesai pekerjaan (*timing*), dan urutan operasi untuk suatu pekerjaan (*routing*). Pengurutan pekerjaan (*sequencing*) merupakan suatu urutan kedatangan dari bermacam-macam pekerjaan yang harus diselesaikan dalam jangka waktu tertentu.

Adapun tujuan dari aktivitas penjadwalan adalah sebagai berikut (Ginting, 2009).

- a. Meningkatkan penggunaan sumber daya atau mengurangi waktu tunggu, sehingga total waktu proses dapat berkurang, dan produktivitas dapat meningkat;
- b. Mengurangi persediaan barang setengah jadi atau mengurangi sejumlah pekerjaan yang menunggu dalam antrian ketika sumber daya yang ada masih mengerjakan tugas yang lain;
- c. Mengurangi beberapa keterlambatan pada pekerjaan yang mempunyai batas waktu penyelesaian sehingga akan meminimasi biaya keterlambatan;
- d. Membantu pengambilan keputusan mengenai perencanaan kapasitas pabrik dan jenis kapasitas yang dibutuhkan sehingga penambahan biaya yang mahal dapat dihindarkan.

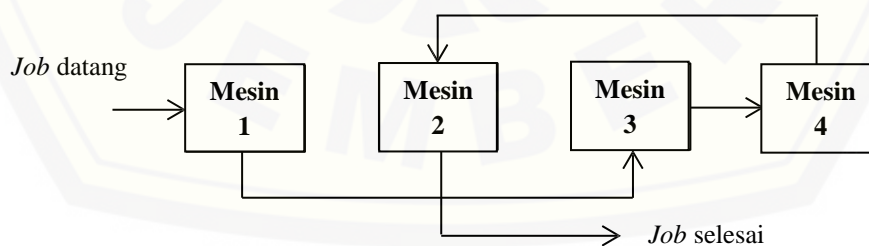
Dengan demikian masalah penjadwalan senantiasa melibatkan pengerjaan sejumlah komponen yang sering disebut dengan istilah *job*. *Job* sendiri masih merupakan komposisi dari sejumlah elemen-elemen dasar yang disebut dengan aktivitas operasi. Tiap aktivitas atau operasi ini membutuhkan alokasi sumber daya tertentu selama periode waktu tertentu yang sering disebut dengan waktu proses.

Penjadwalan secara garis besar dapat dibedakan dalam penjadwalan untuk *jobshop* dan *flowshop*. Permasalahan yang membedakan antara *jobshop* dan *flowshop* adalah pola aliran yang tidak memiliki tahapan-tahapan proses yang sama. Untuk dapat melakukan penjadwalan dengan baik maka waktu proses kerja setiap mesin serta jenis pekerjaannya perlu diketahui, waktu tersebut dapat diperoleh melalui pengukuran waktu kerja, jenis serta jumlah pekerjaan diperoleh dengan melakukan pengamatan dari operator pada bagian tertentu. Setelah mengetahui jenis serta waktu kerja tiap mesin yang akan dijadwalkan maka proses penjadwalan baru dapat dilakukan.

2.2 Penjadwalan *Jobshop*

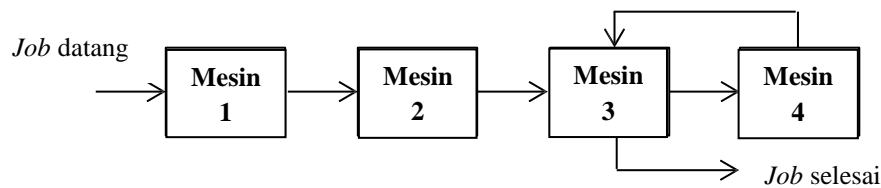
Proses produksi dengan aliran *jobshop* berarti proses produksi dengan pola aliran atau rute proses pada tiap mesin yang spesifik untuk setiap pekerjaan, dan mungkin berbeda untuk tiap *job*. Akibat aliran proses yang tidak searah ini, maka setiap *job* yang dikerjakan di suatu mesin dapat berupa *job* baru atau *job* yang sedang dikerjakan, dan *job* yang akan menjadi produk jadi telah diproses di mesin tersebut. Pola aliran *jobshop* diantaranya akan ditunjukkan pada gambar berikut ini.

- a. *Job* yang datang dapat melompat ke tahap selanjutnya tanpa melewati mesin di depannya, kemudian kembali untuk diproses pada mesin yang telah dilewati tersebut. Ilustrasi tersebut dapat ditunjukkan pada Gambar 2.1.



Gambar 2.1 Aliran *job* dapat melompat ke mesin selanjutnya

- b. *Job* yang datang melewati setiap mesin, tetapi *job* tersebut dapat kembali diproses pada mesin sebelumnya yang telah digunakan untuk memproses *job* tersebut. Ilustrasi tersebut ditunjukkan pada Gambar 2.2



Gambar 2.2 Aliran *job* dapat diproses kembali pada mesin yang telah digunakan

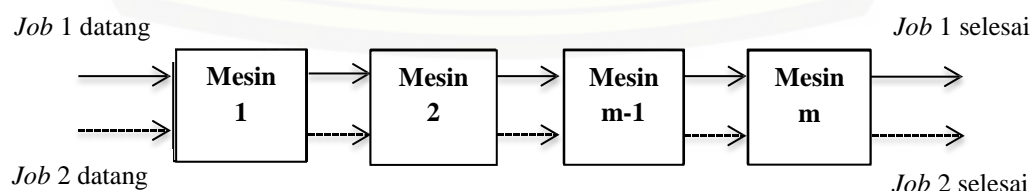
2.3 Penjadwalan *Flowshop*

Penjadwalan *flowshop* merupakan model penjadwalan dimana *job-job* yang akan diproses seluruhnya mengalir pada arah atau jalur produk yang sama. Dengan kata lain, *job-job* yang memiliki urutan kerja yang sama. Umumnya, pada sistem produksi yang bersifat *flowshop*, terdiri dari beberapa mesin (m) dan mempunyai sejumlah *job* yang harus dikerjakan (n) serta waktu proses per unit *job* i pada mesin j , t_{ij} (untuk $i = 1, \dots, n$; $j = 1, \dots, m$). Penjadwalan *flowshop* sering kali diselesaikan dengan mengembangkan permutasi urutan *job* yang akan diurutkan. *Job* bersifat *independent* secara serempak tersedia pada waktu nol, dan urutan mesin dari semua pekerjaan sama. Masing-masing *job* memiliki waktu proses pada masing-masing mesin. Tujuan penjadwalan pada umumnya adalah menemukan suatu urutan *job* yang bertujuan untuk meminimumkan *makespan*.

Flowshop terbagi menjadi dua macam pola yaitu *pure flowshop* dan *general flowshop*.

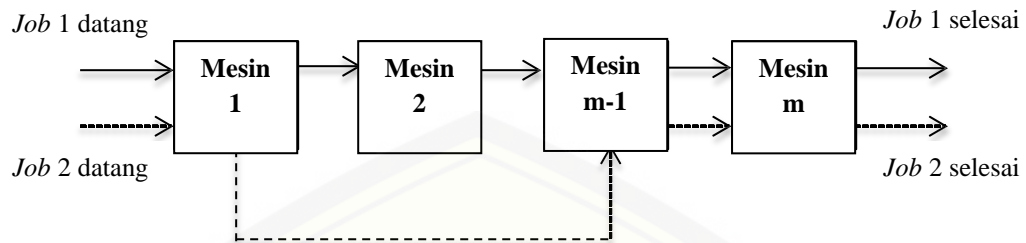
- Pure Flowshop*, berbagai *job* akan mengalir pada lini produksi yang sama atau *job* yang datang dikerjakan di semua mesin. Misalnya, masing-masing *job* melalui mesin 1, kemudian mesin 2, dan seterusnya sampai dengan mesin m .

Gambar 2.3 berikut menunjukkan gambaran tentang *pure flowshop*.



Gambar 2.3 Pola aliran *pure flowshop*

- b. *General Flowshop*, beberapa *job* yang datang tidak harus dikerjakan semua mesin. Gambar 2.4 berikut menunjukkan gambaran tentang *general flowshop*.

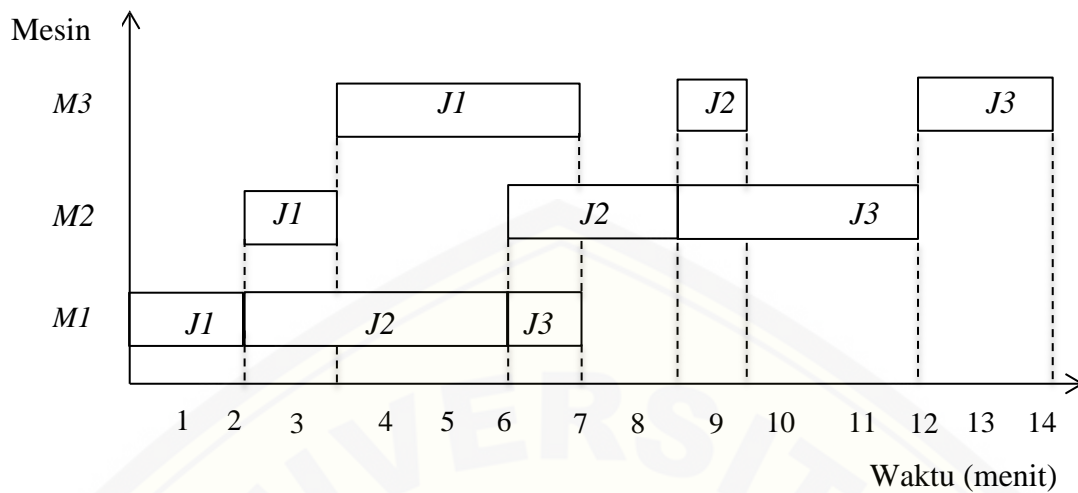


Gambar 2.4 Pola aliran *general flowshop*

Berdasarkan penjelasan diatas, penjadwalan *jobshop* adalah penjadwalan produksi tidak searah dimana tiap *job* yang diproses mengalir pada rute proses yang berbeda. Sedangkan penjadwalan *flowshop* adalah penjadwalan produksi dimana tiap *job* yang diproses mengalir secara searah.

2.4 Diagram Gantt

Masalah penjadwalan sebenarnya masalah murni pengalokasian dan dengan bantuan model matematis akan dapat ditentukan solusi optimal. Model-model penjadwalan akan memberikan rumusan masalah yang sistematis berikut dengan solusi yang diharapkan. Sebagai alat bantu yang digunakan dalam menyelesaikan masalah penjadwalan dikenal suatu model yang sederhana dan umum digunakan secara luas yakni peta Gantt (*ganttt chart*). *Gantt chart* pertama kali diperkenalkan oleh Henry Laurence Gantt pada tahun 1916. *Gantt chart* merupakan grafik hubungan antara alokasi sumber daya dengan waktu. Pada sumbu vertikal digambarkan jenis sumber daya yang digunakan dan sumbu horizontal digambarkan satuan waktu (Ginting, 2009). *Gantt chart* dapat ditunjukkan pada Gambar 2.5.



Gambar 2.5 Diagram *Gantt*

Keuntungan menggunakan diagram *ganttt* sebagai berikut (Ginting, 2009).

- Dalam situasi keterbatasan sumber, penggunaan diagram *ganttt* 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 diperlihatkan secara grafis dalam suatu diagram yang mudah dipahami.

2.5 Algoritma *Cross Entropy-Genetic Algorithm* (CEGA)

Algoritma *Cross Entropy* (CE) termasuk teknik yang cukup baru, tujuannya adalah untuk menghasilkan urutan solusi yang memusat dengan cepat ke arah solusi yang optimal dari hasil iterasi. Algoritma *Cross Entropy* (CE) awalnya diterapkan untuk simulasi kejadian langka (*rare-event*), lalu dikembangkan untuk beberapa kasus seperti optimasi kombinatorial, optimasi kontinyu, *machine learning* dan beberapa kasus lain (Widodo, 2014).

Secara umum langkah-langkah algoritma *Cross Entropy* (CE) adalah sebagai berikut.

- Inisialisasi Parameter

Pada tahap ini ditentukan nilai parameter yang meliputi: jumlah sampel (N), parameter kejarangan (ρ), koefisien penghalusan (α), parameter vektor (v) dan jumlah iterasi.

b. Pembangkitan Sampel

Pada tahap ini dilakukan pembangkitan sampel sebanyak N yaitu X_1, \dots, X_N dengan mekanisme tertentu, memanfaatkan parameter v .

c. Perhitungan Fungsi Tujuan

Pada tahap ini dilakukan evaluasi sampel dengan memasukkan ke dalam fungsi tujuan, kemudian nilai fungsi tujuan diurutkan dari yang terkecil ke yang terbesar (untuk kasus maksimasi dilakukan sebaliknya).

d. Penentuan Sampel Elit

Sampel elit dipilih sebanyak $\rho \times N$ dari jumlah sampel yang dibangkitkan berdasarkan nilai fungsi tujuannya (Basori *et al.*, 2011).

e. Update Parameter Vektor v

Update parameter vektor v bisa dilakukan dengan rumus berikut.

$$v_t = \alpha w_t + (1 - \alpha)v_{t-1} \quad (2.1)$$

dimana w_t adalah parameter vektor yang didapatkan dari solusi. Dalam hal distribusi normal, parameter w_t ini berupa nilai μ dan σ yang dihitung menggunakan sampel elit. Parameter v yang sudah diupdate digunakan untuk membangkitkan sampel baru.

f. Mengecek Iterasi Maksimal

Penentuan iterasi maksimal digunakan untuk memutuskan apakah iterasi selesai atau tidak. Jika iterasi selesai, maka didapatkan solusi optimal. Namun jika tidak kembali pada langkah b.

Widodo (2014) menggabungkan algoritma *Cross Entropy* (CE) diatas dengan *Genetic Algorithm* (GA). *Genetic Algorithm* (GA) merupakan teknik pencarian yang didasarkan pada mekanisme seleksi alam dan genetika alami. *Genetic Algorithm* (GA) berangkat dari himpunan solusi yang dihasilkan secara acak. *Genetic Algorithm* (GA) banyak dipakai dalam penyelesaian masalah kombinatorial seperti TSP, VRP dan *crew scheduling*.

Secara umum langkah-langkah *Genetic Algorithm* (GA) adalah sebagai berikut.

a. Pembangkitan Populasi

Pada tahap ini, populasi awal atau kromosom-kromosom awal dibangkitkan secara random. Ditentukan pula ukuran populasi, probabilitas *crossover*, probabilitas mutasi dan jumlah iterasi.

b. Evaluasi Solusi

Pada tahap ini, nilai setiap individu didalam populasi dievaluasi menggunakan fungsi *fitness*. Fungsi *fitness* bisa berhubungan langsung dengan fungsi tujuan, atau bisa juga sedikit modifikasi terhadap fungsi tujuan.

c. Elitisme

Tahap elitisme (*elitism*) dalam GA merupakan usaha mempertahankan individu-individu terbaik yang telah diperoleh di suatu generasi ke dalam generasi selanjutnya. Sehingga individu-individu terbaik ini akan tetap muncul di populasi berikutnya.

d. Seleksi

Pada tahap ini dilakukan seleksi kompetitif untuk memilih anggota populasi sebagai induk untuk dilakukan *crossover*. Salah satu teknik seleksi dalam *Genetic Algorithm* adalah *roulette wheel*.

e. *Crossover*

Istilah *crossover* sering disebut kawin silang. Ada bermacam-macam teknik *crossover* yang bisa digunakan dalam GA. *Crossover* dilakukan untuk mendapatkan kombinasi yang lebih baik antara satu individu dengan individu yang lain dalam satu populasi.

f. Mutasi

Mutasi dimaksudkan untuk memunculkan individu baru yang berbeda sama sekali dengan individu yang sudah ada. Dalam konteks optimasi memungkinkan munculnya solusi baru untuk bisa keluar dari *local* optimum.

g. Mengecek Iterasi Maksimal

Penentuan iterasi maksimal digunakan untuk memutuskan apakah iterasi selesai atau tidak. Jika iterasi selesai, maka didapatkan solusi optimal. Namun jika tidak kembali pada langkah b.

Pada penelitian ini, penulis merujuk pada penelitian Widodo (2014) yang menggunakan algoritma *Cross Entropy-Genetic Algorithm* (CEGA). Widodo (2014) merancang algoritma *Cross Entropy-Genetic Algorithm* (CEGA) dengan cara menggabungkan algoritma *Cross Entropy* (CE) dan *Genetic Algorithm* (GA) yang memiliki mekanisme *crossover* dan mutasi yang dapat menghasilkan kromosom baru. Tujuan pengembangan ini adalah memperluas pencarian solusi pada sampel elit *Cross Entropy* (CE) ketika berada di lokal optimal dengan menambahkan mekanisme *crossover* dan mutasi dari *Genetic Algorithm* (GA), hal tersebut berfungsi untuk menghindari kemungkinan pencarian solusi terjebak di area lokal optimal karena mekanisme GA dapat menghasilkan kromosom baru yang memiliki sifat tetap menjaga *nature* kromosom pada populasi awalnya.

Langkah-langkah algoritma *Cross Entropy-Genetic Algorithm* (CEGA) pada penjadwalan *flowshop* adalah sebagai berikut.

a. Inisialisasi Parameter

Pada tahap ini ditentukan nilai parameter inisial yang meliputi: jumlah sampel (N), parameter kejarangan (ρ), koefisien penghalusan (α), parameter *crossover* (P_c), dan jumlah iterasi. Berikut ini penjelasan mengenai penentuan parameter inisial.

- 1) Jumlah sampel (N) pada algoritma ini berupa urutan prioritas dari seluruh *job* yang akan dijadwalkan. Semakin banyak jumlah *job*, maka semakin banyak jumlah sampel awal yang harus dibangkitkan.
- 2) Parameter kejarangan ρ (*rho*), merupakan parameter yang digunakan untuk menentukan banyaknya sampel elit yang akan diambil pada suatu populasi solusi (N). Nilai yang umum digunakan berkisar antara 1%-10% (Kroese, 2009).

- 3) Koefisien penghalusan α (*alpha*), memiliki rentang nilai antara $0 < \alpha < 1$. Namun secara empiris nilai $0,4 < \alpha < 0,9$ merupakan nilai yang paling optimal (Boer *et al.*, 2003).
 - 4) Default nilai inisial untuk parameter *crossover* (P_c) adalah 1 (Budiman, 2010).
- b. Pembangkitan Sampel
- Pada tahap ini dilakukan pembangkitan sampel awal dengan mekanisme random sebanyak N yaitu X_1, \dots, X_N . Sampel tersebut berupa urutan *job* sepanjang n .
- c. Perhitungan Fungsi Tujuan
- Menghitung nilai fungsi tujuan Z (*makespan*), yaitu waktu yang dibutuhkan untuk menyelesaikan seluruh *job* mulai dari *job* 1 sampai ke n dikerjakan sampai selesai. Sehingga untuk menentukan nilai Z adalah sebagai berikut.
- $$Z = \max_{1 \leq j \leq n}(F_j) \quad (2.2)$$
- dimana
- $F_j = \text{Flow Time}$
- d. Mengecek Iterasi Maksimal
- Penentuan iterasi maksimal digunakan untuk memutuskan apakah iterasi selesai atau tidak. Jika iterasi selesai, maka didapatkan solusi optimal. Namun jika tidak maka diteruskan ke langkah e.
- e. Penentuan Sampel Elit
- Untuk penentuan sampel elit maka nilai *makespan* dari semua sampel kemudian diurutkan dari yang terkecil hingga terbesar. Rumus jumlah sampel elit adalah $N \times \rho$.
- f. Pembobotan Sampel Elit
- Pada tahap ini dilakukan pembobotan terhadap sampel elit yang telah diperoleh pada tahap sebelumnya. Bobot yang diberikan sesuai dengan urutan sampel elit. Bobot yang telah diberikan untuk sampel elit nantinya akan dijadikan dasar penentuan sampel yang dielitisme.

g. Perhitungan *Linier Fitness Ranking* (LFR)

Tahap perhitungan nilai LFR ini digunakan untuk pemilihan induk pada proses *crossover*. LFR diperoleh melalui rumus berikut (Nurkhalida *et al.*, 2012).

$$\text{LFR}(I(N-i+1)) = F_{max} - (F_{max} - F_{min}) \left(\frac{(i-1)}{(N-1)} \right) \quad (2.3)$$

dimana

$$F_{max} = \frac{1}{z(1)}$$

$$F_{min} = \frac{1}{z(N)}$$

h. Update Parameter *Crossover*

Update parameter *crossover* pada algoritma CEGA ini merupakan modifikasi dari update parameter vektor pada algoritma *Cross Entropy* (CE). Pada setiap iterasi akan dilakukan update parameter *crossover*. Semakin besar nilai parameter *crossover* maka semakin banyak jumlah sampel yang akan mengalami *crossover*. Rumus update parameter *crossover* adalah sebagai berikut.

$$P_{c(i)} = (1-\alpha) u + P_{c(i-1)} \alpha \quad (2.4)$$

dengan u pada persamaan (2.5) berikut (Puspitasari, 2011).

$$u = \frac{\bar{Z}_e}{2 \times Z_{best}} \quad (2.5)$$

dimana

\bar{Z}_e = rata-rata *makespan* pada sampel elit

Z_{best} = *makespan* terbaik pada setiap iterasi

i. Elitisme

Pada tahap elitisme ini bertujuan untuk menyimpan sampel dengan nilai fungsi tujuan terbaik pada setiap iterasi. Sampel ini nantinya akan muncul kembali pada populasi sampel di tahap berikutnya.

j. Pemilihan Induk *Crossover*

Pemilihan dua buah kromosom sebagai induk yang akan *dicrossover* dilakukan secara proporsional sesuai dengan nilai *fitness*-nya. Dengan

menggunakan mekanisme elitisme, dilakukan pemilihan induk 1 dari sampel elit dan digunakan mekanisme *roulette wheel* untuk memilih induk 2.

1) Pemilihan induk 1

Dalam pemilihan induk 1 ini akan diambil dari nilai sampel elit yang pada perhitungan sebelumnya memiliki nilai fungsi tujuan terbaik.

2) Pemilihan induk 2

Pada pemilihan induk 2 ini akan dipilih berdasarkan nilai evaluasi dari sampel keseluruhan dengan menggunakan nilai LFR dari sampel yang sedang dievaluasi. Apabila nilai perbandingan antara kumulatif LFR dan total LFR lebih besar dari nilai random yang dibangkitkan, maka sampel tersebut menjadi induk 2.

k. *Crossover*

Crossover merupakan tahapan yang menyilangkan 2 induk untuk membentuk kromosom baru sehingga dihasilkan individu baru yang lebih baik. Metode yang digunakan adalah *2-point order crossover*. Jika dalam pembangkitan bilangan random diperoleh bilangan random (R) lebih kecil dari parameter *crossover* (P_c) atau $R < P_c$, maka akan dilakukan *crossover* terhadap dua kromosom induk tersebut. Namun jika pembangkitan bilangan random (R) lebih besar dari parameter *crossover* (P_c) atau $R > P_c$, maka tidak terjadi *crossover* terhadap kedua kromosom induk (Widodo, 2014). Rumus yang digunakan untuk menentukan bagian dari kromosom induk yang mengalami *crossover* yaitu:

$$r_i = \text{ceil}(\text{random} \times n) \quad (2.6)$$

Berikut ini contoh untuk menentukan bagian dari sampel induk yang mengalami *crossover* dengan menerapkan rumus (2.6) diatas.

Misal dibangkitkan bilangan random yaitu 0,43879 dan 0,8491 dimana $n = 9$ merupakan panjang *job*. Maka:

$$r_1 = \text{ceil}(0,4387 \times 9) = 4 \rightarrow p_1 = 4$$

$$r_2 = \text{ceil}(0,8491 \times 9) = 8 \rightarrow p_2 = 8$$

Dari perhitungan diatas, bagian dari kromosom induk yang mengalami *crossover* dapat ditunjukkan pada Tabel 2.1.

Tabel 2.1 Bagian dari Induk 1 dan Induk 2 yang akan mengalami *crossover* menggunakan metode *2-point order crossover*

Kromosom	Gen								
Induk 1	4	5	7	6	1	2	3	8	9
Induk 2	1	9	7	8	4	6	5	2	3

Kemudian dengan menerapkan metode *2-point order crossover*, diperoleh calon anak 1 dan anak 2 seperti yang ditunjukkan pada Tabel 2.2.

Tabel 2.2 Calon Anak 1 dan Anak 2 hasil metode *2-point order crossover*

Kromosom	Gen								
Calon Anak 1	8	4	6	5	2	...
Calon Anak 2	6	1	2	3	8	...

Untuk mengisi titik-titik pada kromosom calon anak 1, maka diperiksa satu persatu gen pada kromosom induk 1. Jika terdapat gen pada kromosom induk 1 yang belum tercantum pada kromosom calon anak 1, maka gen tersebut akan mengisi titik-titik. Hal ini dilakukan secara berurutan. Proses tersebut juga dilakukan untuk calon anak 2. Sehingga kromosom induk setelah dilakukan *crossover* menjadi seperti yang ditunjukkan pada Tabel 2.3.

Tabel 2.3 Anak 1 dan Anak 2 hasil metode *2-point order crossover*

Kromosom	Gen								
Anak 1	7	1	3	8	4	6	5	2	9
Anak 2	9	7	4	6	1	2	3	8	5

1. Mutasi

Mutasi dimaksudkan untuk memunculkan individu baru yang berbeda dengan individu yang sudah ada. Probabilitas mutasi (P_m) akan menentukan kromosom mana yang akan mengalami perubahan gen, semakin besar nilai probabilitas mutasi maka semakin banyak kromosom dalam populasi yang akan mengalami mutasi. Proses mutasi akan dipilih secara random dan gen pada site tersebut akan diubah nilainya. Angka random akan dibangkitkan dengan batasan 0 sampai 1. Jika angka random (R) tersebut lebih kecil dari

parameter mutasi (P_m) maka digit gen akan diganti, dan jika angka random (R) tersebut lebih besar dari parameter mutasi (P_m) maka digit gen tidak akan diganti (Widodo, 2014). Mutasi dilakukan dengan menggunakan tiga mekanisme, yaitu *swap mutation* (menukar), *flip mutation* (membalik), dan *slide mutation* (menggeser). Didalam suatu populasi akan terdapat $P_m \times N$ sampel kromosom yang dimutasi. Nilai parameter mutasi ditentukan dengan rumus sebagai berikut (Hanka *et al.*, 2013).

$$P_m = \frac{P_c}{2} \quad (2.7)$$

Rumus yang digunakan untuk menentukan bagian sampel yang dimutasi yaitu:

$$I, J = \text{ceil}(n \times r_{ij}) \quad (2.8)$$

Rumus yang digunakan untuk menentukan mekanisme mutasi yaitu:

$$k = \text{ceil}(\text{rand } K \times 3) \quad (2.9)$$

Jika nilai $k = 1$ maka dilakukan *Flip Mutation* (membalik), $k = 2$ dilakukan *Swap Mutation* (menukar), $k = 3$ maka dilakukan *Slide Mutation* (menggeser). Ilustrasi dari ketiga mekanisme mutasi tersebut dapat dilihat pada Tabel 2.4, Tabel 2.5 dan Tabel 2.6 (Kachitvichyanukul *et al.*, 2015).

Tabel 2.4 Contoh *Swap Mutation*

<i>Parent</i>	1	9	5	2	8	4	6	3	7
<i>Offspring</i>	1	9	4	2	8	5	6	3	7

Tabel 2.5 Contoh *Flip Mutation*

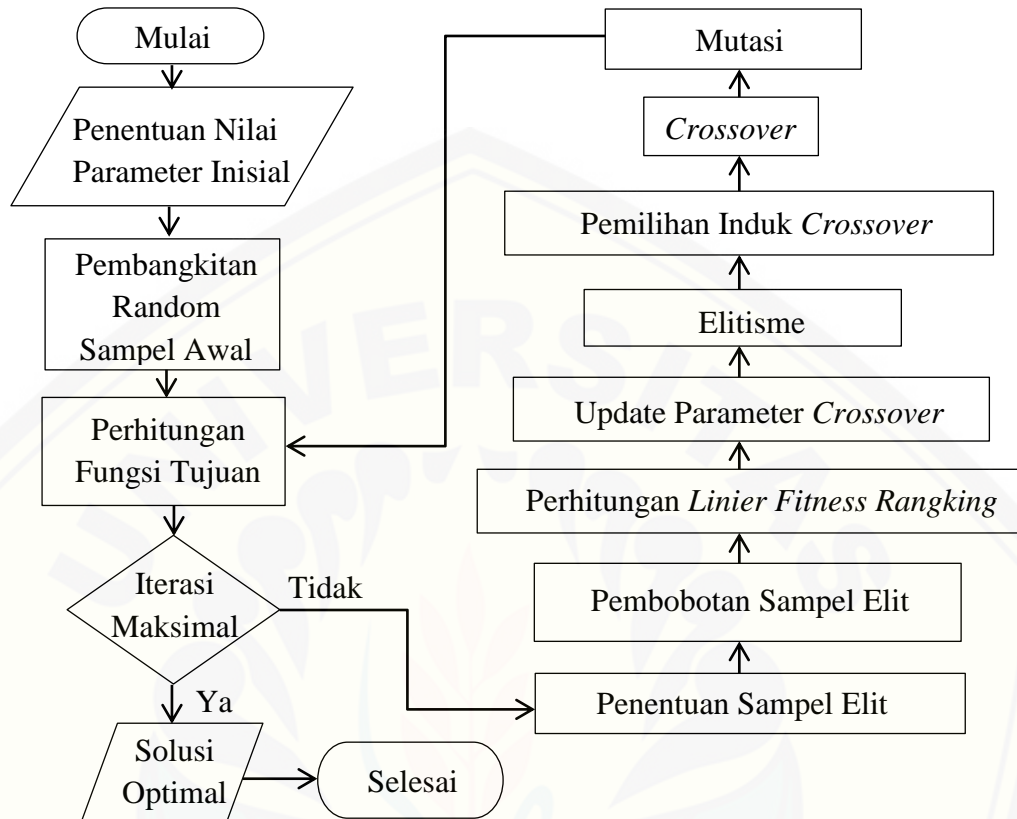
<i>Parent</i>	1	9	5	2	8	4	6	3	7
<i>Offspring</i>	1	9	4	8	2	5	6	3	7

Tabel 2.6 Contoh *Slide Mutation*

<i>Parent</i>	1	9	5	2	8	4	6	3	7
<i>Offspring</i>	1	9	4	5	2	8	6	3	7

Setelah dilakukan proses mutasi maka akan diperoleh populasi sampel dengan anggota yang baru, kemudian kembali ke langkah c.

Secara umum prosedur algoritma *Cross Entropy-Genetic Algorithm* (CEGA) dapat dilihat pada Gambar 2.6.



Gambar 2.6 Skema Algoritma *Cross Entropy-Genetic Algorithm* (CEGA)

2.6 Algoritma *Differential Evolution Plus* (DE Plus)

Differential Evolution (DE) adalah suatu metode metaheuristik yang pemakaiannya cukup luas dalam bidang rekayasa. *Differential Evolution* (DE) termasuk metode pencarian stokastik yang berdasarkan populasi (*population based search*). DE memiliki kesamaan dengan *evolutionary algorithms* (EA) yang lain, tetapi berbeda dalam hal informasi jarak dan arah dari populasi yang sekarang digunakan untuk memandu proses pencarian solusi yang lebih baik. Pada awalnya DE digunakan dalam kasus optimasi kontinyus. Sejak diusulkan 1995, DE mendapatkan reputasi yang bagus sebagai global optimizer yang efektif (Santosa dan Willy, 2011).

Seperti halnya semua anggota yang termasuk dalam *Evolutionary Algorithm* (EA), DE juga didasarkan pada pembangkitan populasi titik-titik untuk

mencapai minimum suatu fungsi. DE berbeda dengan algoritma EA dalam hal berikut ini.

- a. Mutasi diterapkan dulu untuk membangkitkan vektor percobaan (*trial vector*), yang kemudian akan digunakan dalam proses *crossover* untuk menghasilkan satu turunan (*offspring*);
- b. *Step Size* dalam mutasi tidak disampel dari distribusi populasi yang sudah diketahui (Santosa dan Willy, 2011).

Secara umum algoritma *Differential Evolution* memiliki tahapan prosedur yaitu inialisasi populasi, mutasi, *crossover*, dan seleksi (Santosa dan Willy, 2011). Algoritma DE *Plus* memiliki perbedaan dengan DE murni, yaitu dengan melakukan tiga modifikasi. Pertama, memodifikasi pengendalian nilai parameter F dan Cr (*adaptive parameters*) dengan cara menghitung parameter F dan Cr pada tiap generasi menggunakan formula tertentu sehingga nilai parameter F dan Cr pada tiap generasi berubah-ubah. Kedua, menambahkan prosedur *local search* pada DE untuk meningkatkan kualitas solusi yang dihasilkan. Ketiga, *crossover* pada mulanya merupakan langkah untuk menyilangkan populasi target dengan populasi mutan, dimodifikasi menjadi langkah untuk memilih populasi target yang akan dimutasi sehingga dilakukan sebelum proses mutasi (Wiratno *et al.*, 2012).

Langkah-langkah algoritma *Differential Evolution Plus* adalah sebagai berikut.

- a. Inialisasi populasi

Pada tahap inialisasi populasi, DE murni dan algoritma DE *Plus* memiliki proses yang sama. Sebelum melakukan inialisasi terhadap titik populasi maka perlu dilakukan penentuan batas atas (ub) dan batas bawah (lb). Untuk pembangkitan nilai awal generasi $g = 0$, variabel ke- j dan vektor ke- i bisa diwakili dengan persamaan (2.10).

$$x_{j,i,0} = lb_j + rand_j(0,1)(ub_j - lb_j) \quad (2.10)$$

dimana

x = individu target

j = variabel job j

i = populasi ke- i

lb = batas bawah

ub = batas atas

Bilangan random dibangkitkan dengan fungsi $\text{rand}()$, dimana bilangan yang dihasilkan terletak antara $(0,1)$. Solusi awal diperoleh dengan cara mengurutkan populasi menggunakan prosedur SPV (*Smallest Position Value*). SPV merupakan prosedur pengurutan nilai x_i tiap populasi dari nilai terkecil ke nilai terbesar.

b. *Crossover*

Langkah *crossover* pada DE murni dilakukan dengan menyilangkan setiap vektor $x_{i,g}$, dengan vektor mutan $v_{i,g}$, untuk membentuk vektor hasil persilangan $u_{i,g}$. Langkah tersebut ditunjukkan oleh persamaan (2.11).

$$u_{i,g} = u_{i,i,g} = \begin{cases} v_{j,i,g}, & \text{jika } j(\text{rand}_j(0,1) \leq Cr, \text{ atau } j = j_{\text{rand}} \\ x_{j,i,g}, & \text{jika yang lain} \end{cases} \quad (2.11)$$

Pada algoritma DE *Plus*, *crossover* dilakukan sebelum mutasi untuk memilih populasi yang akan dimutasi. Konsep mutasi yaitu apabila bilangan random lebih kecil atau sama dengan Cr maka populasi tersebut akan dimutasi, bila bilangan random lebih besar dari Cr maka populasi itu tidak dimutasi (Wiratno *et al.*, 2012). Probabilitas *crossover*, $Cr \in (0,1)$ adalah nilai yang didefinisikan untuk mengendalikan fraksi nilai parameter yang disalin dari mutan. Probabilitas *crossover* untuk tiap generasi akan ditentukan dengan persamaan Mingyong dan Erbao yaitu pada persamaan (2.12) (Wiratno *et al.*, 2012).

$$Cr = Cr_{\min} + G \cdot \frac{Cr_{\max} - Cr_{\min}}{MAXGEN} \quad (2.12)$$

dimana

Cr = probabilitas *crossover*

Cr_{\min} = nilai terkecil dari probabilitas *crossover*

Cr_{\max} = nilai terbesar dari probabilitas *crossover*

G = iterasi pada saat waktu *running time*

$MAXGEN$ = jumlah maksimum iterasi

Tujuan dari penentuan nilai Cr adalah untuk meningkatkan keragaman vektor yang akan mengalami *crossover* dan menghindari dari local optimal.

c. Mutasi

Mutasi pada algoritma DE murni dan algoritma DE *Plus* dilakukan dengan menggunakan formula yang sama. Algoritma DE *Plus* melakukan mutasi dan kombinasi terhadap populasi yang telah dipilih pada langkah *crossover*. Mutasi dilakukan dengan cara menambahkan perbedaan dua vektor terhadap vektor ketiga secara acak. Formulasinya dapat dilihat pada persamaan (2.13).

$$v_{i,g} = x_{r0,g} + F(x_{r1,g} - x_{r2,g}) \quad (2.13)$$

dimana

F = parameter mutasi

$x_{r0}, x_{r1,g}, x_{r2,g}$ = vektor acak

Faktor skala $F \in (0,1)$ adalah bilangan real positif yang mengendalikan tingkat pertumbuhan populasi. Dalam hal ini nilai parameter F tiap generasi akan berubah-ubah dengan menghitung nilai parameter pada tiap generasi dengan formula pada persamaan (2.14).

$$F = \begin{cases} \max(F_{min}, 1 - \left| \frac{f_{max}}{f_{min}} \right|) & \text{jika } \left| \frac{f_{max}}{f_{min}} \right| < 1 \\ \max(F_{min}, 1 - \left| \frac{f_{min}}{f_{max}} \right|) & \text{jika yang lain} \end{cases} \quad (2.14)$$

dimana

f_{min} = nilai minimum fungsi objektif

f_{max} = nilai maksimum fungsi objektif

F_{min} = input parameter yang memastikan $F \in [F_{min}, 1]$

d. Seleksi

Tahap seleksi pada algoritma DE murni dan algoritma DE *Plus* akan diperoleh solusi terbaik. Seleksi tersebut dapat ditunjukkan pada persamaan (2.15).

$$x_{i,g+1} = \begin{cases} v_{i,g}, & \text{jika } f(v_{i,g}) \leq f(x_{i,g}) \\ x_{i,g}, & \text{jika sebaliknya} \end{cases} \quad (2.15)$$

e. *Local Search*

Hasil dari seleksi akan dikenai prosedur *insert-based local search* yang cenderung mengarahkan pencarian ke daerah solusi yang menjanjikan dalam waktu relatif singkat. Prosedur dari *insert-based local search* ditunjukkan pada tahap berikut ini.

- 1) Ubah nilai individual $x_i(t)$ menjadi permutasi *job* π_{i_0} ;
- 2) Pilih secara acak u dan v , dimana $u \neq v$; $\pi_i = \text{insert}(\pi_{i_0}, u, v)$;
- 3) *Set loop* = 1

Pilih secara acak u dan v , dimana $u \neq v$;

$\pi_{i_1} = \text{insert}(\pi_i, u, v)$;

Jika $f(\pi_{i_1}) < f(\pi_i)$, maka $\pi_i = \pi_{i_1}$;

loop⁺⁺;

while loop < ($n \times (n - 1)$).

- 4) Jika $f(\pi_i) < f(\pi_{i_0})$, maka $\pi_{i_0} = \pi_i$.

dimana

$\pi_{i_0}, \pi_i, \pi_{i_1}$ = permutasi *job*

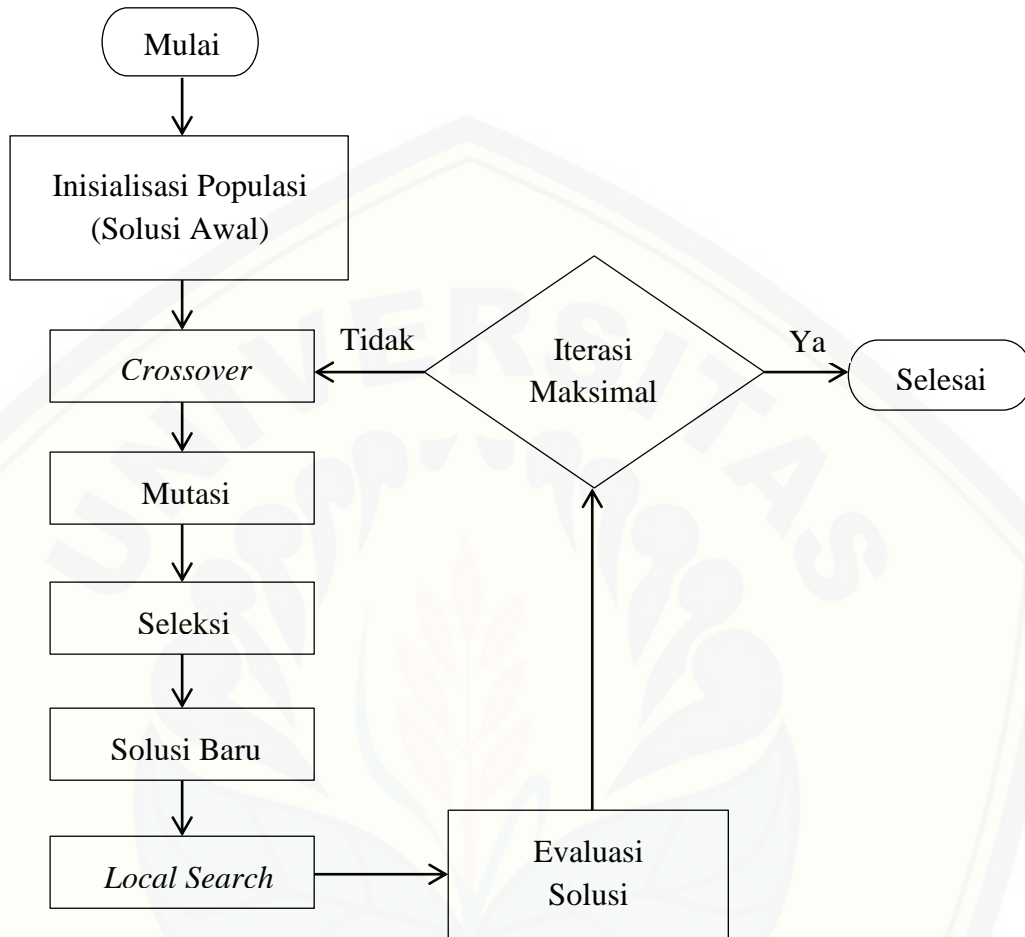
u = *job* yang dipilih secara acak

v = posisi

f. Mengecek Iterasi Maksimal

Penentuan iterasi maksimal digunakan untuk memutuskan apakah iterasi selesai atau tidak. Jika iterasi selesai, maka didapatkan solusi optimal. Namun jika tidak kembali pada langkah b.

Secara umum prosedur algoritma *Differential Evolution Plus* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Skema Algoritma *Differential Evolution Plus* (DE Plus)

2.7 Kriteria Kekonvergenan

Menurut Sivanandam & Deepa (2008) pemberhentian atau kriteria kekonvergenan suatu algoritma dapat dituliskan sebagai berikut.

a. Iterasi Maksimum

Algoritma akan berhenti ketika iterasi maksimum sudah tercapai.

b. Batas waktu maksimum

Proses algoritma akan berhenti ketika batas waktu maksimum telah terlampaui. Jika iterasi maksimum sudah tercapai sebelum batas waktunya terpenuhi maka prosesnya akan berhenti.

- c. Nilai *fitness*/fungsi objektif tidak terjadi perubahan

Proses algoritma akan berhenti jika tidak ada perubahan pada nilai *fitness* atau nilai fungsi objektif terbaiknya.



BAB 3. METODE PENELITIAN

3.1 Data Penelitian

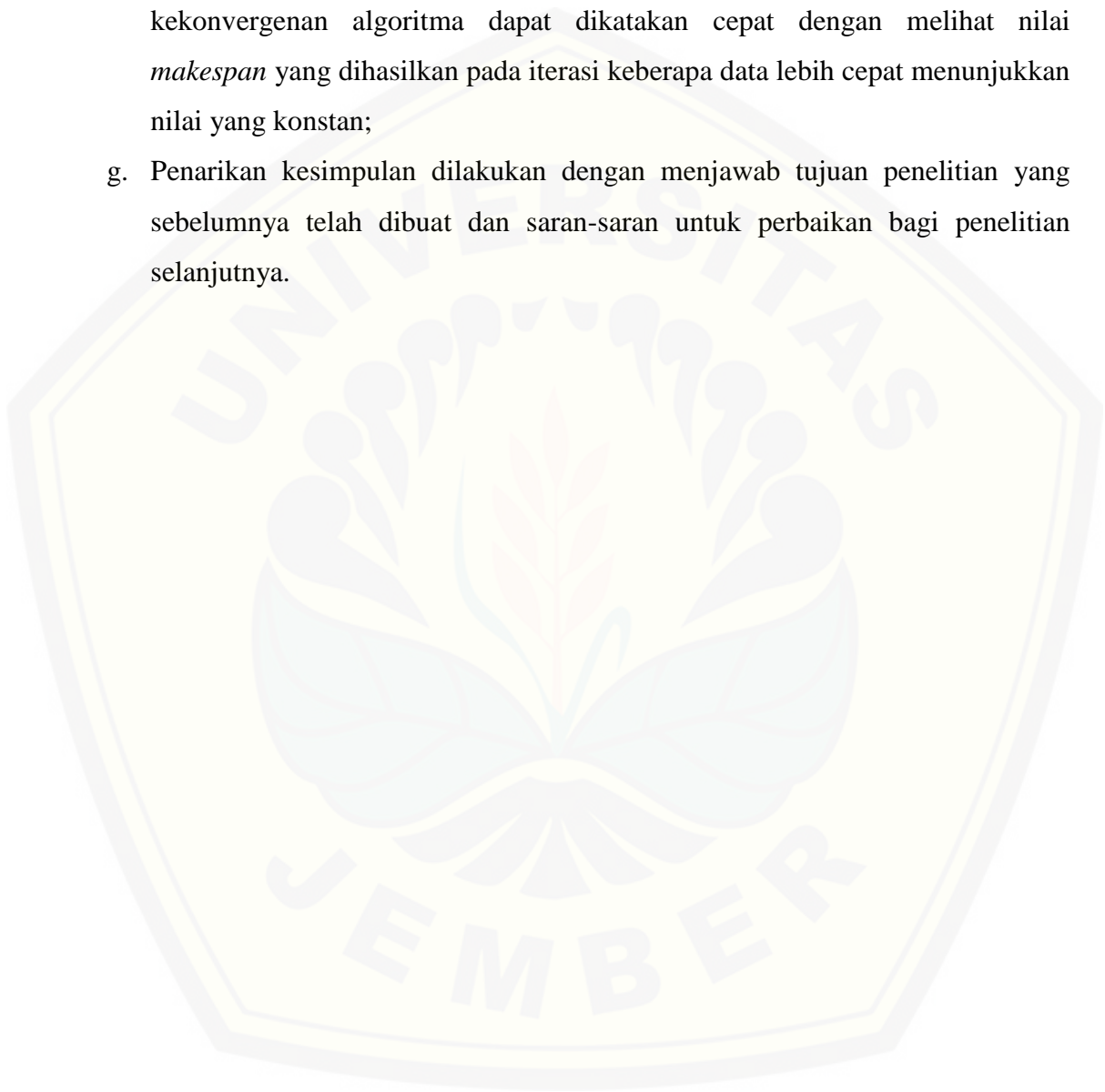
Data yang digunakan dalam penelitian ini adalah data sekunder yang diambil dari *website OR-Library*. Data-data tersebut merupakan data *flowshop* yang diformulasi oleh J. Carlier (1978) dan C.R. Reeves (1995) (lihat lampiran A).

3.2 Langkah-langkah Penelitian

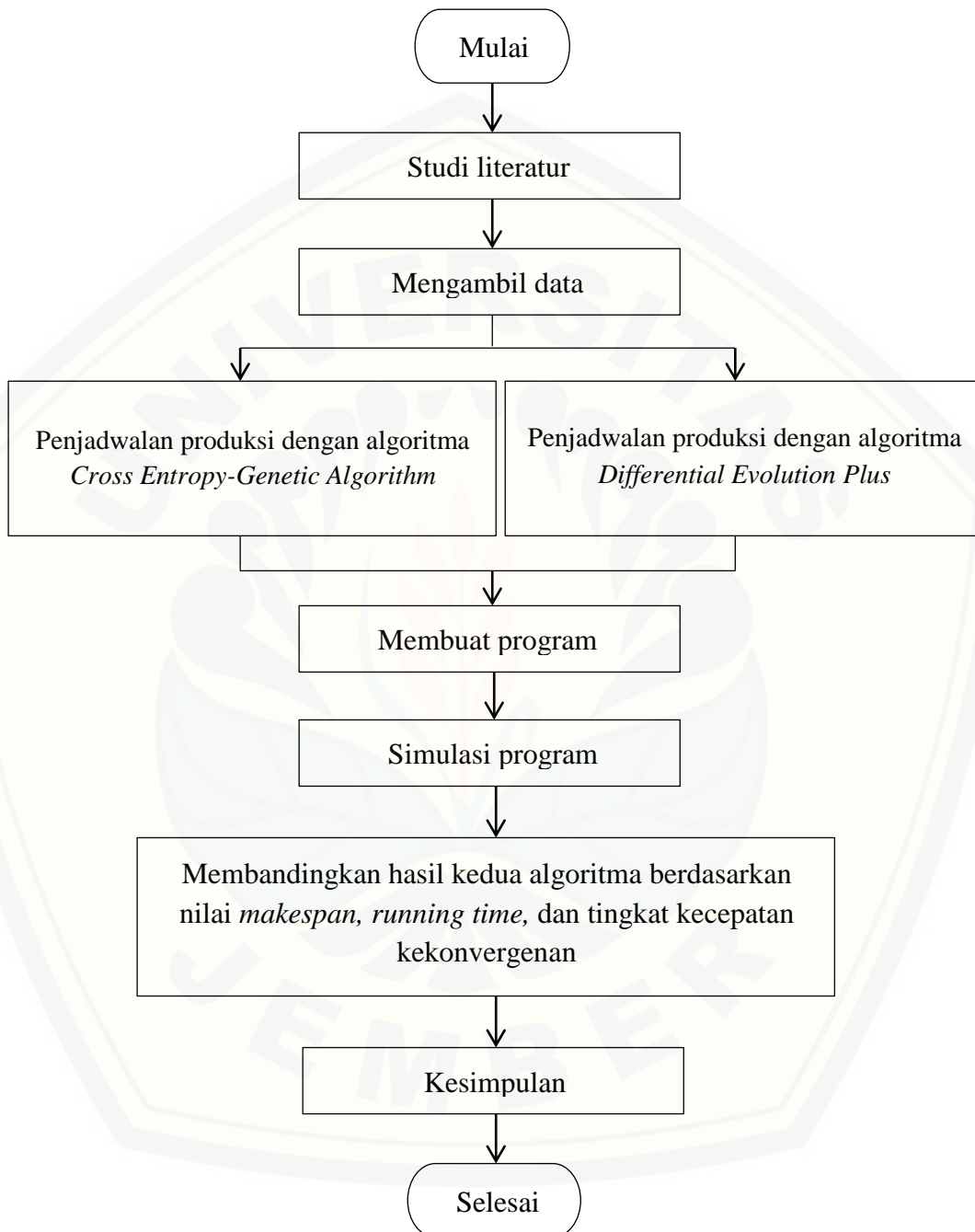
Dalam melakukan penelitian, maka terlebih dahulu dibuat langkah-langkah sistematis yang menjadi proses pengerjaan sehingga penelitian dapat dijalankan secara lebih terencana dan terstruktur. Langkah-langkah penelitian yang akan dilakukan dalam penyelesaian permasalahan penjadwalan *flowshop* pada tugas akhir ini adalah sebagai berikut.

- a. Studi literatur dengan mengumpulkan berbagai macam teori yang berkaitan dengan algoritma *Cross Entropy-Genetic Algorithm* dan algoritma *Differential Evolution Plus* yang didapat dari buku-buku referensi, artikel, maupun jurnal. Studi literatur dilakukan sebagai pedoman pengerjaan laporan tugas akhir baik untuk pengolahan data dan analisis yang akan dilakukan. Studi literatur juga bertujuan agar lebih memahami teori-teori yang dipakai dan proses pengerjaan laporan yang ada;
- b. Pengambilan data penjadwalan *flowshop* yang terdapat pada *website OR-Library*;
- c. Menjadwalkan produksi *flowshop* menggunakan algoritma *Cross Entropy-Genetic Algorithm* dan algoritma *Differential Evolution Plus* pada data kecil yang meminimumkan *makespan* secara manual;
- d. Pembuatan program menggunakan *software MATLAB*, pada langkah ini dibuat desain dan *script* program sesuai dengan langkah kerja kedua algoritma yang digunakan;
- e. Simulasi program dilakukan dengan menggunakan data yang didapat pada langkah b;

- f. Membandingkan hasil dari kedua algoritma dengan menggunakan program yang telah dibuat berdasarkan nilai *makespan*, *running time* (waktu komputasi) dan tingkat kecepatan kekonvergenan. Nilai *makespan* dan *running time* yang dipilih adalah nilai yang paling kecil. Sedangkan kekonvergenan algoritma dapat dikatakan cepat dengan melihat nilai *makespan* yang dihasilkan pada iterasi beberapa data lebih cepat menunjukkan nilai yang konstan;
- g. Penarikan kesimpulan dilakukan dengan menjawab tujuan penelitian yang sebelumnya telah dibuat dan saran-saran untuk perbaikan bagi penelitian selanjutnya.



Gambar 3.1 berikut menunjukkan langkah-langkah penelitian yang akan dilakukan dalam tugas akhir ini.



Gambar 3.1 Skema Langkah-langkah Penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil percobaan sebanyak sepuluh kali untuk setiap parameter α , Cr_{min} , dan Cr_{max} yang berbeda dengan 2000 iterasi pada data *car6* (8 Job, 9 Mesin), *car7* (7 Job, 7 Mesin), *car8* (8 Job, 8 Mesin), *reC07* (20 Job, 10 Mesin), *reC13* (20 Job, 15 Mesin), *reC19* (30 Job, 10 Mesin), dan *reC27* (30 Job, 15 Mesin) dapat diambil kesimpulan sebagai berikut.

- a. Penerapan algoritma CEGA dan algoritma DE *Plus* pada tujuh data *flowshop* yaitu *car6* (8 Job, 9 Mesin), *car7* (7 Job, 7 Mesin), *car8* (8 Job, 8 Mesin), *reC07* (20 Job, 10 Mesin), *reC13* (20 Job, 15 Mesin), *reC19* (30 Job, 10 Mesin), dan *reC27* (30 Job, 15 Mesin) menunjukkan bahwa pada kasus data *car6* (8 Job, 9 Mesin), *car7* (7 Job, 7 Mesin), *car8* (8 Job, 8 Mesin), dan *reC07* (20 Job, 10 Mesin) nilai *makespan* optimal yang dihasilkan kedua algoritma sama yaitu berturut-turut 8505 satuan waktu, 6590 satuan waktu, 8366 satuan waktu, dan 1566 satuan waktu. Selanjutnya pada kasus data *reC13* (20 Job, 15 Mesin), *reC19* (30 Job, 10 Mesin), *reC27* (30 Job, 15 Mesin) nilai *makespan* optimal yang dihasilkan oleh algoritma CEGA yaitu berturut-turut 1938 satuan waktu, 2141 satuan waktu, dan 2396 satuan waktu. Sedangkan nilai *makespan* optimal yang dihasilkan oleh algoritma DE *Plus* yaitu berturut-turut 1943 satuan waktu, 2136 satuan waktu, dan 2426 satuan waktu.
- b. Pada kasus data *car6* (8 Job, 9 Mesin), *car7* (7 Job, 7 Mesin), dan *car8* (8 Job, 8 Mesin) rata-rata *makespan* yang dihasilkan algoritma CEGA dan algoritma DE *Plus* sama. Sedangkan pada kasus data *reC07* (20 Job, 10 Mesin), *reC13* (20 Job, 15 Mesin), *reC19* (30 Job, 10 Mesin), dan *reC27* (30 Job, 15 Mesin) rata-rata *makespan* yang dihasilkan algoritma DE *Plus* lebih kecil dibandingkan algoritma CEGA. Pada tingkat konvergensi dengan batas toleransi $\varepsilon = 10^{-5}$, untuk data *car6* (8 Job, 9 Mesin), *car7* (7 Job, 7 Mesin), dan *car8* (8 Job, 8 Mesin) algoritma DE *Plus* lebih cepat konvergen dibandingkan algoritma CEGA. Sedangkan pada kasus data *reC07* (20 Job,

10 Mesin), *reC13* (20 *Job*, 15 Mesin), *reC19* (30 *Job*, 10 Mesin), dan *reC27* (30 *Job*, 15 Mesin) algoritma CEGA lebih cepat konvergen dibandingkan algoritma DE *Plus*. Serta *Running time* yang dihasilkan algoritma CEGA untuk menyelesaikan seluruh data lebih cepat dibandingkan algoritma DE *Plus*.

5.2 Saran

Penelitian lebih lanjut dapat dilakukan untuk mengembangkan algoritma *Cross Entropy-Genetic Algorithm* dan algoritma *Differential Evolution Plus* dalam menyelesaikan masalah penjadwalan yaitu tentang minimisasi *mean flow time*, *mean weight flow time*, maksimum *lateness*, *mean tardiness*, dan *mean weight tardiness*. Selain itu, masih terbuka untuk menerapkan algoritma *Cross Entropy-Genetic Algorithm* dan algoritma *Differential Evolution Plus* pada masalah penjadwalan dengan multiobyektif.

DAFTAR PUSTAKA

- Baker, K. 1974. *Introduction to Sequencing and Scheduling*. New York: John Wiley and Sons, Inc.
- Budiman, M.A. 2010. “Pendekatan Cross Entropy-Genetic Algorithm Untuk Permasalahan Penjadwalan Job Shop Tanpa Waktu Tunggu Pada Banyak Mesin”. Tidak diterbitkan. Skripsi. Surabaya: Institut Teknologi Sepuluh Nopember.
- Basori, Y & Suparno. 2011. *Pendekatan Cross Entropy Untuk Minimasi Bikriteria Makespan Dan Total Tradiness Pada Penjadwalan Produksi Flowshop Dengan Mesin Paralel*. Surabaya: Jurusan Teknik Industri Institut Teknologi Sepuluh Nopember.
- Boer, D., Pleter-Tjerk., Kroese, D.P., Mannor, S., Rubinstein, R.Y. 2003. *A tutorial on the Cross-Entropy Method*. Haifa: Department of Industrial Engineering, Technion-Israel Institute of Technology.
- Dewi, A.R. 2015. “Penyelesaian Penjadwalan Flowshop Dengan Algoritma Artificial Immune System dan Algoritma Differential Evolution Plus”. Tidak diterbitkan. Skripsi. Jember: FMIPA Universitas Jember.
- Ginting, R. 2009. *Penjadwalan Mesin*. Yogyakarta : Graha Ilmu.
- Hanka, M.K.R., Santosa, B. 2013. “Pengembangan Algoritma Hybrid Cross Entropy-Genetic Algorithm Pada Permasalahan Multiobjective Job Shop Scheduling Untuk Minimasi Makespan Dan Mean Flow Time”. Tugas Akhir. Surabaya: Institut Teknologi Sepuluh Nopember.
- Kachitvichyanukul, V., Sethanan, K., Golinska-Dawson, P. 2015. *Toward Sustainable Operations of Supply Chain and Logistic Systems*. New York: Springer.
- Kroese, D.P. 2009. *Cross-Entropy Method*. Brisbane: Mathematics Department, University of Queensland.
- Mattfeld, Dirk C. & Vaessens, Rob J. M. 2004, *OR-Library : FSP Test Instances*. [serial online] <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/flowshop1.txt> [8 April 2016].
- Nurkhalida, L., Santosa, B. 2012. *Pendekatan Cross Entropy-Genetic Algorithm Pada Permasalahan Multi Objective Job Shop Scheduling*. Surabaya: Jurusan Teknik Industri Institut Teknologi Sepuluh Nopember.

- Puspitasari, P.D., Santosa, B. 2011. *Penjadwalan Truk Pada Sistem Cross Docking Dengan Penyimpanan Sementara Dengan Algoritma Hybrid Cross Entropy-Genetic Algorithm*. Surabaya: Jurusan Teknik Industri Institut Teknologi Sepuluh Nopember.
- Sa'dia, N. 2015. "Perbandingan Algoritma Artificial Bee Colony dan Algoritma Differential Evolution Plus Untuk Penjadwalan Flowshop". Tidak diterbitkan. Skripsi. Jember: FMIPA Universitas Jember.
- Santosa, B & Willy, P. 2011. *Metoda Metaheuristik: Konsep dan Implementasi*. Surabaya : Guna Widya.
- Sari, T.K. 2015. "Perbandingan Algoritma Simulated Annealing dan Algoritma Differential Evolution Plus Untuk Penjadwalan Flowshop". Tidak diterbitkan. Skripsi. Jember: FMIPA Universitas Jember.
- Sivanandam, S.N. & Deepa, S.N. 2008. *Introduction to Genetic Algorithm*. India: PSG College of Technology Coimbatore.
- Widodo, D.S. 2014. "Pendekatan Algoritma Cross Entropy-Genetic Algorithm Untuk Menurunkan Makespan Pada Penjadwalan Flowshop". *JEMIS*. 2(1) : 35-42.
- Wiratno, S.E., Nurdiansyah, R., Santosa, B. 2012. "Algoritma Differential Evolution Untuk Penjadwalan Flowshop Banyak Mesin Dengan Multi Obyektif". *Jurnal Teknik Industri Institut Teknologi Sepuluh Nopember*. 13(1) : 1-6.

LAMPIRAN

LAMPIRAN A. Data Penelitian

A.1 car6

	<i>M1</i>	<i>M2</i>	<i>M3</i>	<i>M4</i>	<i>M5</i>	<i>M6</i>	<i>M7</i>	<i>M8</i>	<i>M9</i>
J1	887	447	234	159	201	555	463	456	753
J2	799	779	567	267	478	444	123	789	21
J3	999	999	852	483	520	120	456	630	427
J4	666	666	140	753	145	142	789	258	520
J5	663	25	222	420	699	578	876	741	142
J6	333	558	558	159	875	965	543	36	534
J7	222	886	965	25	633	412	210	985	157
J8	114	541	412	863	222	25	123	214	896

A.2 car7

	<i>M1</i>	<i>M2</i>	<i>M3</i>	<i>M4</i>	<i>M5</i>	<i>M6</i>	<i>M7</i>
J1	692	310	832	630	258	147	255
J2	581	582	14	214	147	753	806
J3	475	475	785	578	852	2	699
J4	23	196	696	214	586	356	877
J5	158	325	530	785	325	565	412
J6	796	874	214	236	896	898	302

J7 542 205 578 963 325 800 120

A.3 car8

	M1	M2	M3	M4	M5	M6	M7	M8
J1	456	654	852	145	632	425	214	654
J2	789	123	369	678	581	396	123	789
J3	654	123	632	965	475	325	456	654
J4	321	456	581	421	32	147	789	123
J5	456	789	472	365	536	852	654	123
J6	789	654	586	824	325	12	321	456
J7	654	321	320	758	863	452	456	789
J8	789	147	120	639	21	863	789	654

A.4 reC07

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
J1	28	18	38	11	97	23	90	52	79	63
J2	50	30	75	82	38	39	28	84	48	57
J3	75	50	33	58	56	41	51	29	75	97
J4	65	42	66	29	36	29	10	84	14	67
J5	84	68	42	41	86	23	95	30	73	97
J6	33	72	79	85	81	51	72	19	48	48
J7	91	66	87	88	97	36	21	59	61	4
J8	51	23	100	93	48	84	74	7	98	55

J9	58	61	17	54	25	71	52	47	49	86
J10	44	27	40	19	34	33	3	89	39	66
J11	70	94	7	19	31	48	38	48	73	34
J12	60	38	34	55	63	28	70	35	68	88
J13	39	33	53	87	2	6	51	42	93	67
J14	72	35	45	20	84	23	10	34	8	48
J15	100	71	80	89	47	15	90	33	97	26
J16	79	23	57	54	70	99	85	5	9	4
J17	14	23	36	79	4	65	78	51	95	79
J18	3	32	81	26	19	59	80	90	44	33
J19	68	33	94	37	33	74	64	50	22	17
J20	94	17	54	27	55	34	7	56	10	41

A.5 reC13

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
J1	78	37	79	98	100	21	82	27	7	22	9	57	84	91	5
J2	13	81	100	77	45	39	60	87	38	91	17	85	43	81	33
J3	47	9	31	40	86	27	69	50	87	34	13	15	95	96	72
J4	26	61	5	22	26	52	57	97	10	68	8	49	41	16	35
J5	68	41	46	58	37	59	22	43	49	21	42	70	13	2	76
J6	47	60	23	23	57	60	35	56	54	73	81	61	15	70	51
J7	63	21	10	50	12	84	91	7	44	75	60	72	28	83	52

J8	52	100	38	79	90	52	81	54	51	11	76	50	24	12	23
J9	91	21	59	55	39	75	3	11	14	24	36	72	48	69	55
J10	100	54	77	53	90	91	53	68	18	86	28	61	86	36	15
J11	63	58	93	24	18	78	74	57	100	92	51	73	6	12	83
J12	96	60	79	84	86	31	41	79	63	17	31	65	41	77	74
J13	54	55	89	77	40	71	21	43	60	58	13	3	73	44	85
J14	50	33	51	100	46	27	30	80	56	50	97	70	9	22	92
J15	17	7	12	66	98	25	76	25	76	25	69	69	73	45	74
J16	46	76	32	32	28	48	63	92	85	69	24	38	57	21	66
J17	91	42	30	39	97	72	96	78	71	75	17	26	94	3	39
J18	99	93	57	23	52	89	4	74	21	10	53	94	59	100	68
J19	20	84	25	1	72	63	79	63	17	38	21	36	1	73	58
J20	18	4	21	43	55	86	99	38	81	74	34	40	61	76	100

A.6 reC19

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
J1	40	16	50	59	100	78	38	76	9	68
J2	39	3	35	70	65	80	40	49	52	50
J3	31	56	88	71	83	69	48	98	88	96
J4	65	77	58	66	86	93	69	49	85	51
J5	11	5	75	12	56	64	20	6	83	49
J6	48	56	41	55	3	94	11	87	78	48

J7	47	60	16	67	61	36	36	62	13	74
J8	67	100	3	85	70	19	58	87	61	51
J9	32	47	40	47	66	85	99	50	19	45
J10	27	97	84	30	68	28	26	98	88	96
J11	81	12	1	88	63	32	38	82	68	61
J12	58	56	53	88	100	8	57	92	39	45
J13	38	30	81	51	70	28	10	93	53	45
J14	14	13	1	84	97	69	20	68	19	83
J15	70	98	83	22	27	44	93	46	91	45
J16	75	30	45	64	13	47	6	49	57	21
J17	69	41	37	12	3	81	92	25	24	36
J18	47	92	28	4	28	3	32	85	8	94
J19	52	7	97	56	90	60	37	42	19	15
J20	58	11	18	100	47	24	41	48	51	65
J21	61	69	45	17	4	31	83	32	68	5
J22	63	22	5	77	99	19	99	37	92	19
J23	63	25	83	78	89	66	8	57	89	56
J24	42	86	8	83	39	26	99	75	60	67
J25	51	100	42	53	10	66	19	2	24	41
J26	100	90	68	91	46	5	59	11	10	44
J27	42	41	76	76	61	52	44	78	40	57

J28	83	24	14	100	26	41	19	18	21	12
J29	5	84	57	6	60	91	18	83	44	87
J30	69	35	72	62	90	8	44	67	4	77

A.7 reC27

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
J1	94	24	34	45	45	93	28	83	12	73	34	64	57	58	49
J2	73	46	90	94	20	9	2	89	35	38	87	14	15	28	50
J3	55	6	28	38	93	74	81	89	61	38	9	61	90	24	65
J4	53	9	67	19	46	41	68	33	68	46	51	96	74	16	71
J5	89	63	14	77	14	47	3	35	98	96	24	45	68	18	23
J6	41	42	38	67	92	21	100	39	85	42	37	10	41	16	67
J7	80	35	48	72	77	19	75	37	81	48	64	2	1	65	4
J8	63	83	93	11	87	67	38	33	43	40	46	3	86	1	68
J9	84	78	16	19	31	56	56	18	77	65	82	14	96	20	6
J10	10	94	30	17	37	50	14	50	34	80	88	52	100	37	65
J11	36	32	59	18	56	39	65	77	91	21	58	90	65	33	65
J12	66	98	49	11	61	85	65	8	46	56	39	29	63	42	4
J13	79	42	65	49	77	12	38	75	80	76	65	50	8	9	33
J14	90	52	18	17	66	7	83	24	37	48	15	92	72	20	2
J15	64	40	19	49	100	74	12	71	80	9	31	1	55	18	9
J16	34	69	36	47	71	44	71	20	5	96	15	83	93	36	95

J17	25	6	13	9	46	80	37	52	76	95	56	84	17	9	51
J18	40	62	98	3	23	69	1	3	43	68	91	37	92	84	48
J19	6	38	74	83	21	14	84	41	100	88	58	59	36	99	68
J20	17	79	95	70	70	40	53	5	86	57	59	37	40	33	89
J21	55	15	11	12	85	40	11	26	8	97	38	28	3	59	30
J22	84	59	92	29	89	89	56	48	32	41	29	9	58	84	58
J23	44	76	34	98	87	99	19	89	43	8	55	55	87	22	97
J24	92	97	2	59	47	69	47	85	66	69	19	85	32	86	65
J25	71	90	69	8	47	89	53	86	97	93	56	58	65	37	94
J26	20	94	66	75	18	14	32	66	2	1	10	9	8	28	66
J27	48	56	18	96	53	5	29	22	41	83	57	14	61	90	4
J28	48	46	80	26	42	68	81	11	1	41	34	59	33	43	78
J29	91	5	10	66	64	57	13	38	26	52	89	38	93	93	50
J30	57	78	60	30	20	3	68	83	45	42	94	26	32	47	94

LAMPIRAN B. Hasil Percobaan untuk Data *car6* (2000 Iterasi)

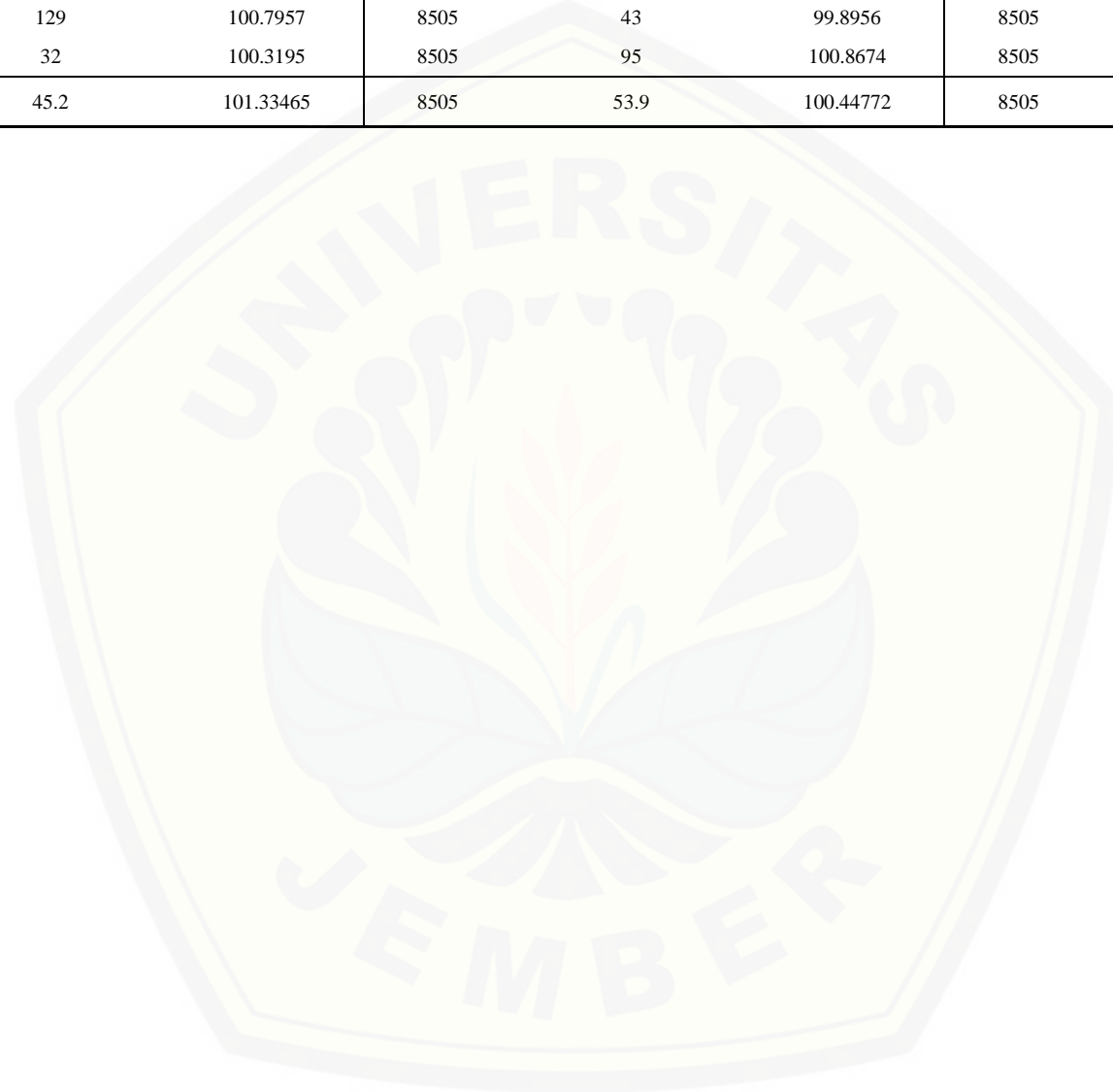
B.1 Algoritma CEGA

No	N=30, rho=0.1, alfa=0.5, Pc=1			N=30, rho=0.1, alfa=0.6, Pc=1			N=30, rho=0.1, alfa=0.7, Pc=1			N=30, rho=0.1, alfa=0.8, Pc=1		
	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time
1	8505	80	10.5962	8505	169	10.712	8505	491	10.5235	8505	267	10.9221
2	8505	120	10.604	8505	781	10.5498	8505	81	10.6797	8505	191	10.8462
3	8505	80	10.6702	8505	129	10.5648	8505	575	10.7699	8505	438	10.7135
4	8505	565	10.6401	8505	622	10.5255	8505	28	11.054	8505	1230	10.6971
5	8505	382	10.6061	8505	184	10.5785	8505	1151	10.4977	8505	869	10.6999
6	8505	676	10.7131	8505	277	10.9395	8505	60	10.8117	8505	745	10.7194
7	8505	203	10.636	8505	325	10.6223	8505	725	10.5572	8505	24	10.6538
8	8505	343	10.6023	8505	611	10.596	8505	228	10.5937	8505	13	10.4982
9	8505	1308	10.6306	8505	312	10.5514	8505	884	10.7439	8505	21	10.7
10	8505	64	10.6038	8505	53	10.6308	8505	128	10.6915	8505	575	10.735
Rata-rata	8505	382.1	10.63024	8505	346.3	10.62706	8505	435.1	10.69228	8505	437.3	10.71852

B.2 Algoritma DE Plus

No	Populasi=30, B. Bawah=0, B. Atas=1, CR min=0.3, CR max=0.9, F min=0.5			Populasi=30, B. Bawah=0, B. Atas=1, CR min=0.4, CR max=0.8, F min=0.5			Populasi=30, B. Bawah=0, B. Atas=1, CR min=0.5, CR max=0.7, F min=0.5		
	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time
1	8505	30	103.8702	8505	91	103.1773	8505	93	102.8344
2	8505	8	102.4659	8505	50	99.9266	8505	33	102.781
3	8505	54	100.5119	8505	80	99.2762	8505	47	101.5103
4	8505	43	100.7535	8505	25	99.644	8505	30	100.762
5	8505	7	100.5156	8505	5	99.8738	8505	67	100.1587
6	8505	59	101.9667	8505	47	99.7616	8505	64	101.6809
7	8505	57	100.9355	8505	53	101.4807	8505	46	102.5584
8	8505	33	101.212	8505	50	100.574	8505	90	99.8023

9	8505	129	100.7957	8505	43	99.8956	8505	87	102.3399
10	8505	32	100.3195	8505	95	100.8674	8505	96	101.019
Rata-rata	8505	45.2	101.33465	8505	53.9	100.44772	8505	65.3	101.54469



LAMPIRAN C. Hasil Percobaan untuk Data *car7* (2000 Iterasi)

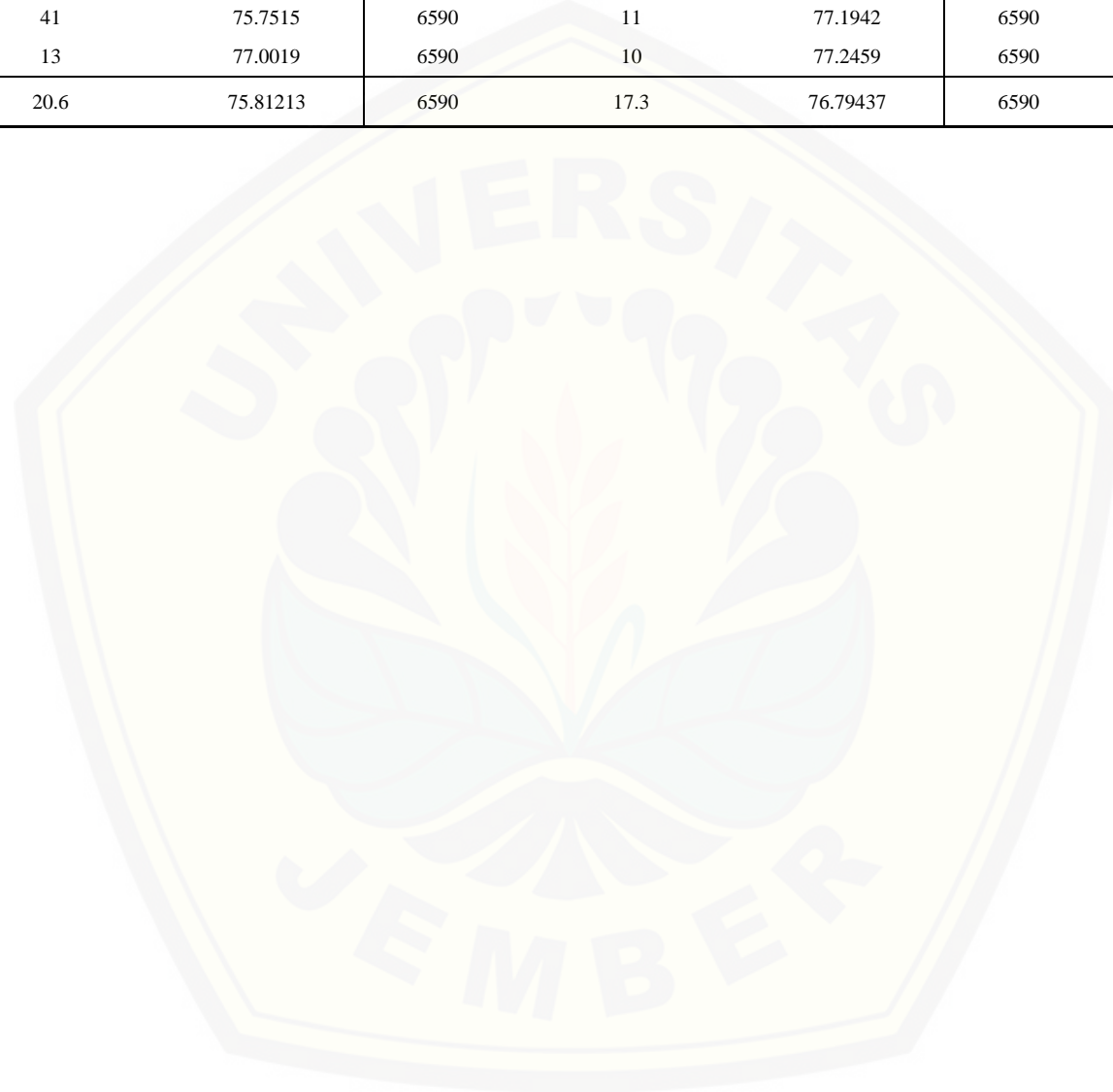
C.1 Algoritma CEGA

No	N=30, rho=0.1, alfa=0.5, Pc=1			N=30, rho=0.1, alfa=0.6, Pc=1			N=30, rho=0.1, alfa=0.7, Pc=1			N=30, rho=0.1, alfa=0.8, Pc=1		
	<i>Makespan</i>	Iterasi Konvergen	<i>Running Time</i>	<i>Makespan</i>	Iterasi Konvergen	<i>Running Time</i>	<i>Makespan</i>	Iterasi Konvergen	<i>Running Time</i>	<i>Makespan</i>	Iterasi Konvergen	<i>Running Time</i>
1	6590	100	8.9838	6590	93	8.8501	6590	1	8.8012	6590	17	9.2066
2	6590	132	8.8633	6590	9	8.9374	6590	26	8.874	6590	10	9.0635
3	6590	14	8.8324	6590	118	9.1018	6590	21	9.0528	6590	696	9.1958
4	6590	56	8.9152	6590	1	8.9002	6590	51	8.8856	6590	31	9.0798
5	6590	7	8.956	6590	41	8.9184	6590	59	8.8899	6590	12	9.124
6	6590	171	8.9857	6590	12	9.0636	6590	26	9.0282	6590	745	9.1107
7	6590	23	8.9031	6590	8	9.1346	6590	12	9.1182	6590	2	9.0602
8	6590	366	8.9417	6590	11	9.1058	6590	988	9.1282	6590	42	9.0589
9	6590	242	8.8916	6590	119	9.1318	6590	290	9.0821	6590	40	9.1386
10	6590	72	8.9867	6590	36	9.0626	6590	37	9.0779	6590	5	9.0867
Rata-rata	6590	118.3	8.92595	6590	44.8	9.02063	6590	151.1	8.99381	6590	160	9.11248

C.2 Algoritma DE Plus

No	Populasi=30, B. Bawah=0, B. Atas=1, CR min=0.3, CR max=0.9, F min=0.5			Populasi=30, B. Bawah=0, B. Atas=1, CR min=0.4, CR max=0.8, F min=0.5			Populasi=30, B. Bawah=0, B. Atas=1, CR min=0.5, CR max=0.7, F min=0.5		
	<i>Makespan</i>	Iterasi Konvergen	<i>Running Time</i>	<i>Makespan</i>	Iterasi Konvergen	<i>Running Time</i>	<i>Makespan</i>	Iterasi Konvergen	<i>Running Time</i>
1	6590	47	75.8023	6590	10	75.9796	6590	40	76.2426
2	6590	19	75.0659	6590	24	74.8734	6590	10	74.8915
3	6590	4	75.0904	6590	19	77.4097	6590	20	77.8078
4	6590	30	75.4372	6590	9	77.7021	6590	10	77.7101
5	6590	13	77.2976	6590	8	75.509	6590	26	76.5026
6	6590	5	75.5257	6590	26	77.026	6590	26	77.872
7	6590	2	75.3925	6590	22	77.437	6590	15	78.2228
8	6590	32	75.7563	6590	34	77.5668	6590	7	78.2865

9	6590	41	75.7515	6590	11	77.1942	6590	30	77.0709
10	6590	13	77.0019	6590	10	77.2459	6590	3	77.8718
Rata-rata	6590	20.6	75.81213	6590	17.3	76.79437	6590	18.7	77.24786



LAMPIRAN D. Hasil Percobaan untuk Data *car8* (2000 Iterasi)

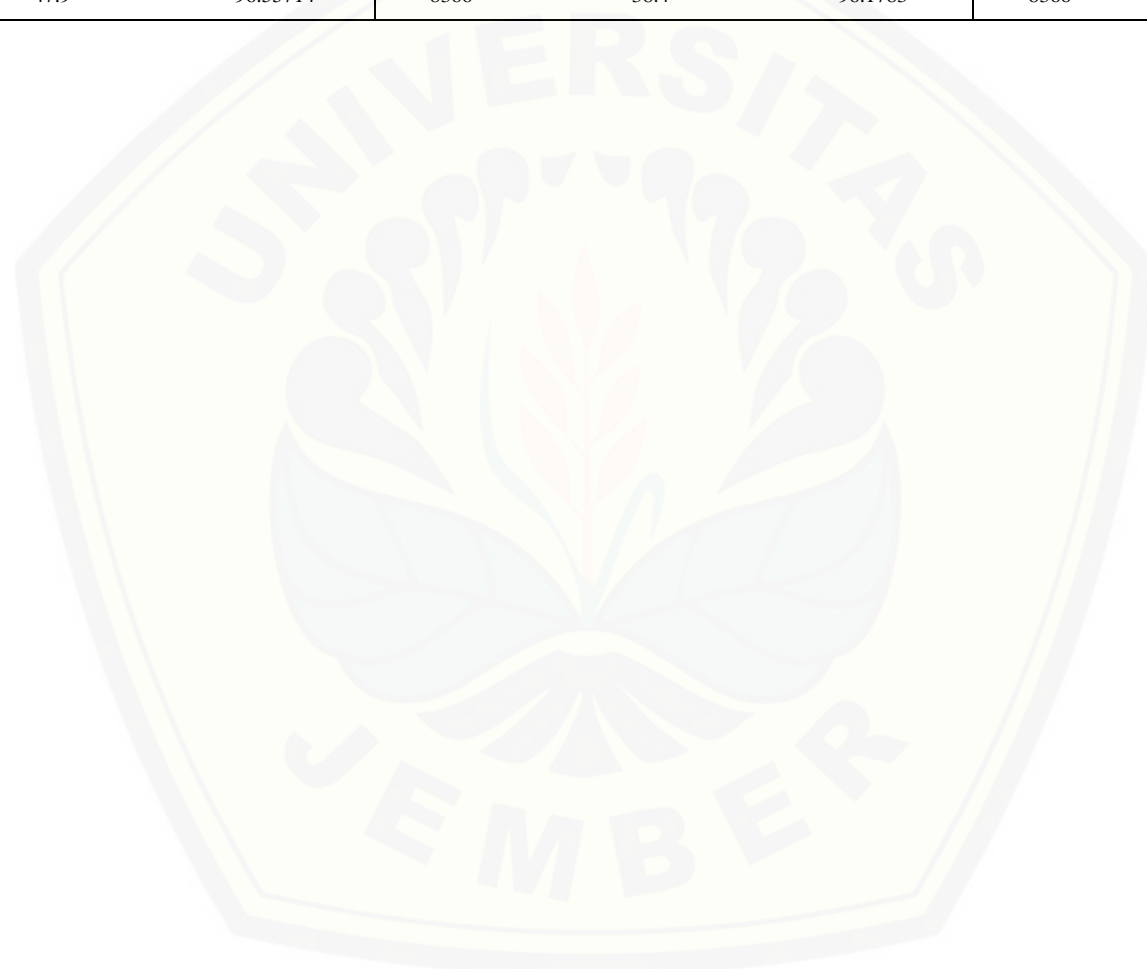
D.1 Algoritma CEGA

No	N=30, rho=0.1, alfa=0.5, Pc=1			N=30, rho=0.1, alfa=0.6, Pc=1			N=30, rho=0.1, alfa=0.7, Pc=1			N=30, rho=0.1, alfa=0.8, Pc=1		
	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time
1	8366	67	10.0644	8366	35	10.1	8366	16	10.0565	8366	34	9.9919
2	8366	28	10.0631	8366	55	10.1915	8366	52	10.1843	8366	70	9.961
3	8366	110	10.0106	8366	79	10.2126	8366	8	10.0478	8366	63	9.972
4	8366	10	10.0413	8366	68	9.9069	8366	95	10.0594	8366	71	10.0155
5	8366	260	10.1546	8366	4	10.0073	8366	170	10.217	8366	79	10.086
6	8366	21	10.1984	8366	59	10.2476	8366	162	10.1847	8366	14	10.0175
7	8366	82	10.2056	8366	23	10.118	8366	53	10.0917	8366	21	9.9904
8	8366	172	10.2333	8366	143	10.1335	8366	12	10.1464	8366	25	10.0188
9	8366	62	10.1664	8366	158	10.1686	8366	117	10.0644	8366	37	10.0145
10	8366	27	10.22	8366	74	10.1286	8366	87	10.0443	8366	141	9.8829
Rata-rata	8366	83.9	10.13577	8366	69.8	10.12146	8366	77.2	10.10965	8366	55.5	9.99505

D.2 Algoritma DE Plus

No	Populasi=30, B. Bawah=0, B. Atas=1, CR min=0.3, CR max=0.9, F min=0.5			Populasi=30, B. Bawah=0, B. Atas=1, CR min=0.4, CR max=0.8, F min=0.5			Populasi=30, B. Bawah=0, B. Atas=1, CR min=0.5, CR max=0.7, F min=0.5		
	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time	Makespan	Iterasi Konvergen	Running Time
1	8366	52	97.4849	8366	20	95.224	8366	43	93.9319
2	8366	43	97.4678	8366	61	97.2983	8366	42	95.265
3	8366	66	95.2261	8366	27	95.8346	8366	48	95.3264
4	8366	49	95.1862	8366	64	95.704	8366	58	95.5443
5	8366	64	95.327	8366	22	94.2365	8366	30	95.4869
6	8366	35	96.9823	8366	27	97.6776	8366	28	96.8112
7	8366	55	96.8397	8366	28	96.4652	8366	13	95.4341

8	8366	19	97.6883	8366	67	96.1245	8366	62	95.1373
9	8366	42	96.1467	8366	45	96.9962	8366	27	95.8355
10	8366	54	97.2224	8366	23	96.2221	8366	36	94.2892
Rata-rata	8366	47.9	96.55714	8366	38.4	96.1783	8366	38.7	95.30618



LAMPIRAN E. Hasil Percobaan untuk Data *reC07* (2000 Iterasi)

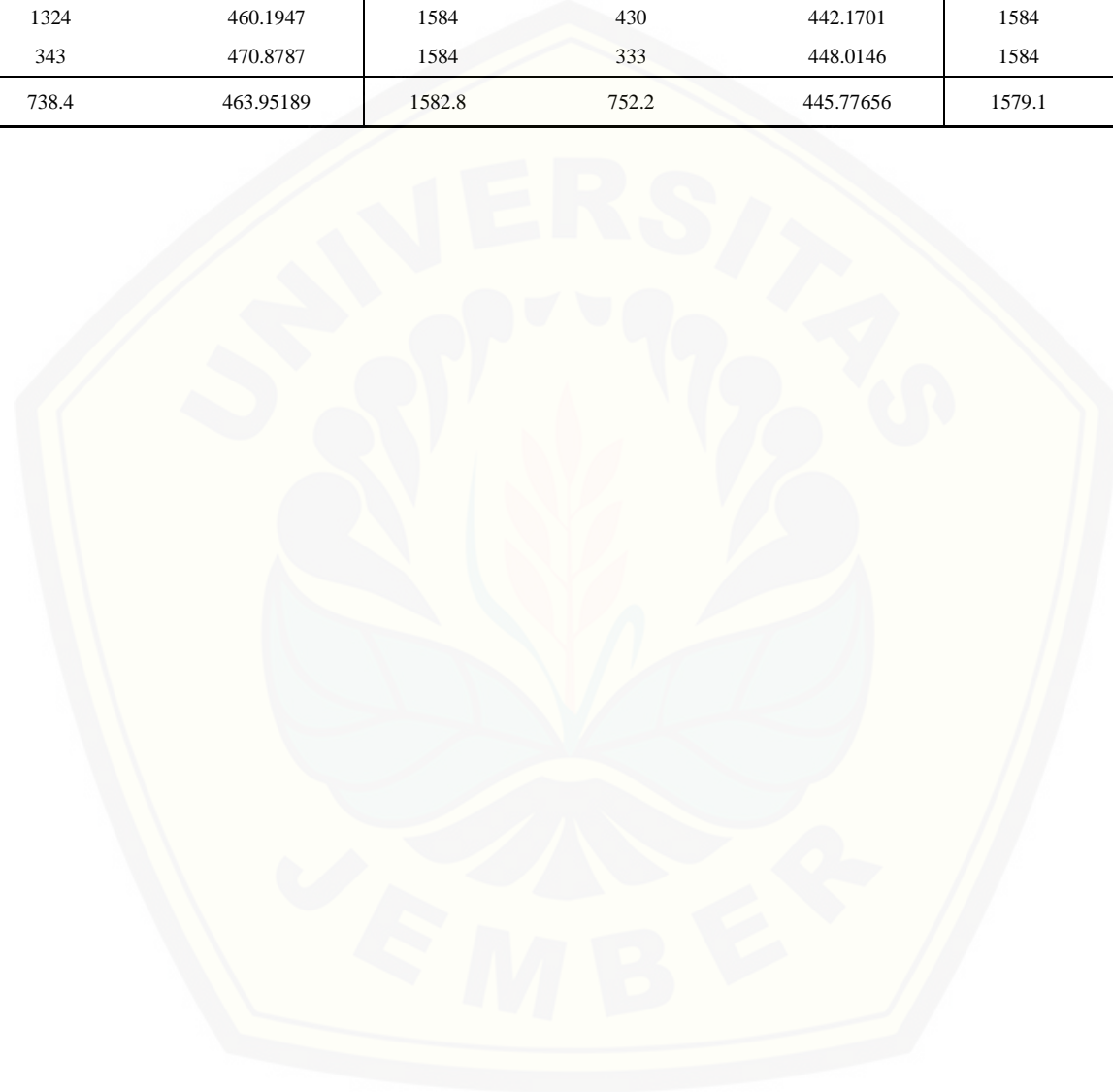
E.1 Algoritma CEGA

No	N=40, rho=0.1, alfa=0.5, Pc=1			N=40, rho=0.1, alfa=0.6, Pc=1			N=40, rho=0.1, alfa=0.7, Pc=1			N=40, rho=0.1, alfa=0.8, Pc=1		
	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>
1	1584	1371	22.7842	1584	735	22.1725	1622	128	22.9486	1626	528	23.1799
2	1609	258	23.314	1595	766	22.4247	1584	319	22.832	1598	82	23.3014
3	1584	136	22.8164	1619	126	22.8994	1572	1499	22.9684	1584	617	22.9406
4	1599	588	22.9017	1584	802	22.5392	1584	1066	22.7756	1584	302	23.1985
5	1584	1970	22.838	1595	136	22.5551	1584	605	22.7274	1599	902	23.1285
6	1599	460	22.9372	1602	1093	22.3953	1584	1983	22.9715	1566	1347	23.0421
7	1584	508	23.025	1619	29	22.2085	1584	701	22.9599	1584	794	22.9781
8	1584	1065	23.081	1584	391	22.2543	1595	113	22.7765	1584	55	23.1768
9	1584	99	22.9525	1584	376	22.2921	1584	177	22.4443	1584	521	22.5861
10	1584	157	22.9254	1584	302	22.5357	1584	532	22.5092	1566	874	22.8922
Rata-rata	1589.5	661.2	22.95754	1595	475.6	22.42768	1587.7	712.3	22.79134	1587.5	602.2	23.04242

E.2 Algoritma DE Plus

No	Populasi=40, B. Bawah=0, B. Atas=1, CR min=0.3, CR max=0.9, F min=0.5			Populasi=40, B. Bawah=0, B. Atas=1, CR min=0.4, CR max=0.8, F min=0.5			Populasi=40, B. Bawah=0, B. Atas=1, CR min=0.5, CR max=0.7, F min=0.5		
	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>
1	1584	124	461.5693	1584	469	449.4149	1584	645	448.7726
2	1577	1252	465.543	1584	364	448.6777	1568	480	451.8975
3	1584	509	463.9303	1584	714	444.0872	1566	1558	458.6968
4	1584	647	458.8682	1577	1780	448.2373	1584	442	454.4813
5	1584	489	468.2926	1579	1940	445.3148	1584	691	458.9809
6	1568	1870	460.9544	1584	662	448.2094	1572	1470	456.5601
7	1584	592	460.044	1584	469	441.7263	1584	400	462.9632
8	1584	234	469.2437	1584	361	441.9133	1581	647	458.053

9	1577	1324	460.1947	1584	430	442.1701	1584	282	457.6856
10	1584	343	470.8787	1584	333	448.0146	1584	668	450.286
Rata-rata	1581	738.4	463.95189	1582.8	752.2	445.77656	1579.1	728.3	455.8377



LAMPIRAN F. Hasil Percobaan untuk Data *reC13* (2000 Iterasi)

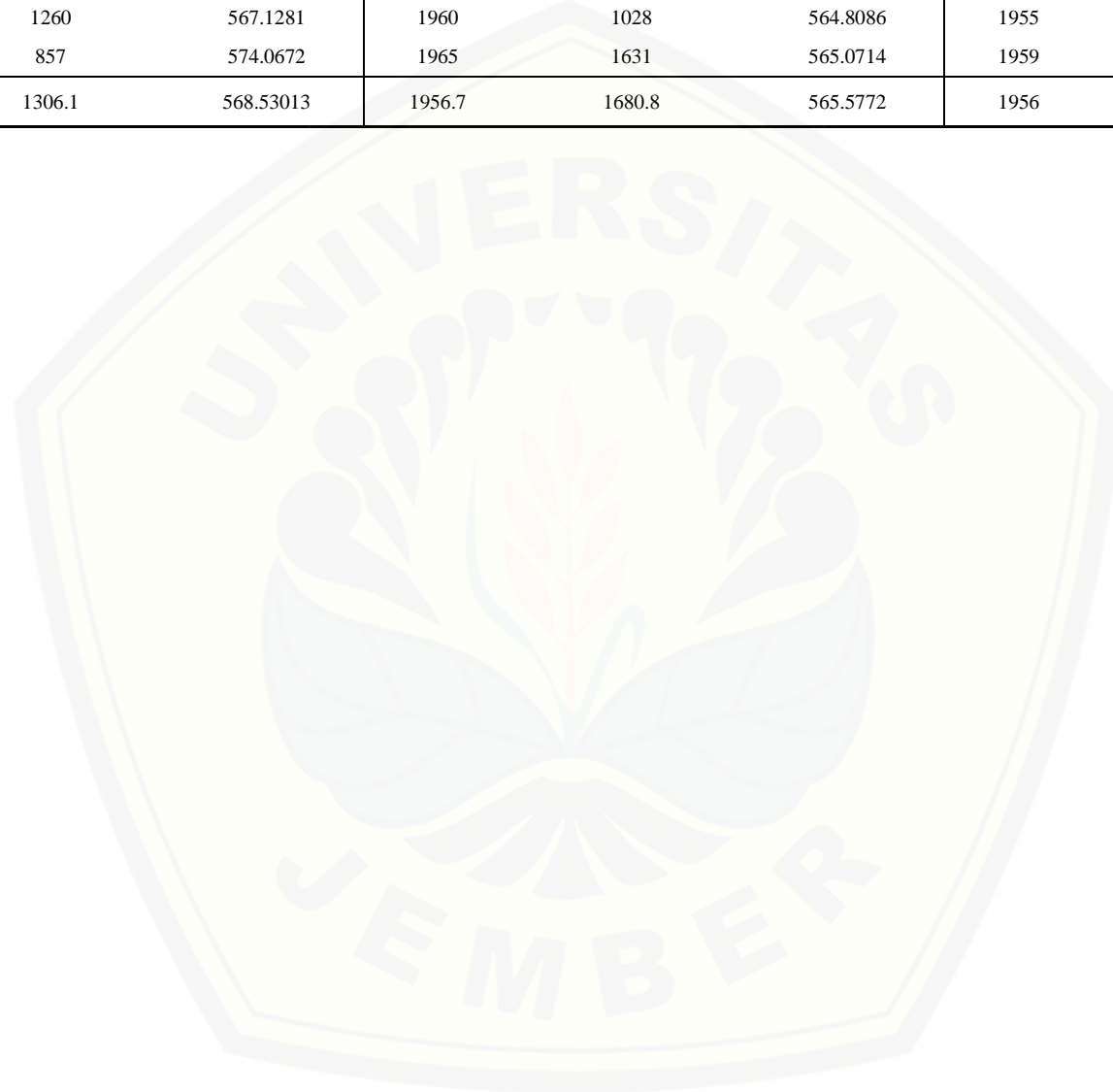
F.1 Algoritma CEGA

No	N=40, rho=0.1, alfa=0.5, Pc=1			N=40, rho=0.1, alfa=0.6, Pc=1			N=40, rho=0.1, alfa=0.7, Pc=1			N=40, rho=0.1, alfa=0.8, Pc=1		
	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>
1	1986	1624	27.6906	1956	1991	27.9477	1981	1510	27.6811	1957	1842	28.4829
2	1995	341	28.0087	1968	422	27.7649	1955	485	29.6232	2011	534	28.5989
3	1954	1433	28.1109	1963	1175	27.9902	2004	1742	28.846	1972	1985	28.5975
4	1969	1498	27.8102	1986	703	27.6119	1945	629	30.7805	1977	182	28.7205
5	1959	1620	27.8799	1944	560	27.8226	1996	1835	28.6915	1977	1756	28.5604
6	1994	242	27.8458	1978	1800	27.9409	1985	1814	28.4859	1938	826	28.6369
7	1998	194	27.9309	1954	1601	27.8373	1976	1933	28.495	1987	1070	28.9852
8	1969	1550	27.8487	1968	1653	27.8946	1995	840	28.4485	1989	1328	28.7108
9	1972	780	28.1655	1966	1746	27.8472	1957	1641	28.8324	1966	835	28.6825
10	1956	1008	27.9024	1998	641	27.9664	1952	281	28.1857	1964	178	28.7068
Rata-rata	1975.2	1029	27.91936	1968.1	1229.2	27.86237	1974.6	1271	28.80698	1973.8	1053.6	28.66824

F.2 Algoritma DE Plus

No	Populasi=40, B. Bawah=0, B. Atas=1, CR min=0.3, CR max=0.9, F min=0.5			Populasi=40, B. Bawah=0, B. Atas=1, CR min=0.4, CR max=0.8, F min=0.5			Populasi=40, B. Bawah=0, B. Atas=1, CR min=0.5, CR max=0.7, F min=0.5		
	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>
1	1956	1040	570.2123	1970	1801	568.1002	1967	1743	573.8406
2	1970	1776	565.0819	1954	1412	569.4038	1955	1933	572.1385
3	1964	713	567.7473	1951	1972	566.2643	1949	1840	579.1362
4	1970	1544	567.314	1954	1887	560.0302	1950	1947	580.2199
5	1965	1848	574.4601	1947	1390	564.6476	1958	1965	587.5674
6	1963	1539	565.7042	1943	1979	566.2526	1949	1855	580.2118
7	1953	1640	566.7869	1955	1725	565.9612	1956	935	582.0332
8	1970	844	566.7993	1968	1983	565.2321	1962	1152	579.4132

9	1954	1260	567.1281	1960	1028	564.8086	1955	1987	578.8216
10	1964	857	574.0672	1965	1631	565.0714	1959	1643	576.8214
Rata-rata	1962.9	1306.1	568.53013	1956.7	1680.8	565.5772	1956	1700	579.02038



LAMPIRAN G. Hasil Percobaan untuk Data *reC19* (2000 Iterasi)

G.1 Algoritma CEGA

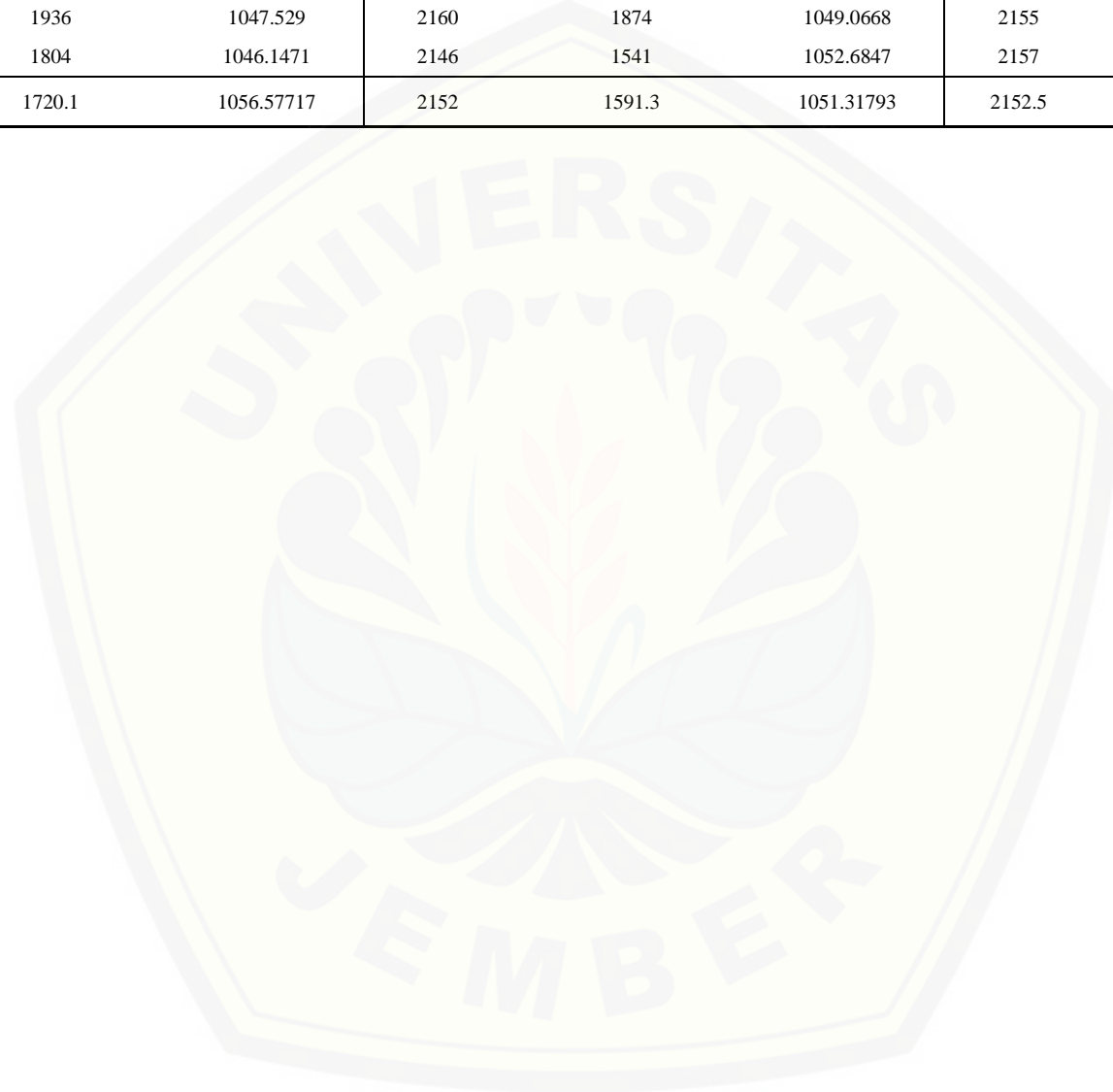
No	N=50, rho=0.1, alfa=0.5, Pc=1			N=50, rho=0.1, alfa=0.6, Pc=1			N=50, rho=0.1, alfa=0.7, Pc=1			N=50, rho=0.1, alfa=0.8, Pc=1		
	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>
1	2163	547	36.4184	2168	838	36.6846	2141	1533	36.3977	2165	1158	38.0828
2	2197	726	37.5576	2182	726	37.4978	2222	944	37.1578	2151	1284	37.6615
3	2162	1918	37.0281	2159	757	37.9114	2199	203	37.1138	2149	1307	37.3372
4	2186	436	36.7163	2149	1802	37.4511	2184	1377	37.308	2172	420	37.2774
5	2155	620	36.6968	2168	838	36.7876	2154	1181	37.2176	2150	859	37.4123
6	2181	786	37.0036	2182	726	37.0795	2196	231	37.261	2159	1228	37.3979
7	2184	1928	39.2761	2159	757	37.1347	2152	1911	36.7952	2195	1463	37.3109
8	2147	1556	36.6962	2149	1802	37.4363	2162	679	37.205	2154	1607	37.5841
9	2172	648	36.7547	2175	881	37.3304	2192	914	37.0492	2155	520	37.1553
10	2212	1989	36.8435	2151	1382	37.4023	2203	153	37.8132	2177	483	37.2163
Rata-rata	2175.9	1115.4	37.09913	2164.2	1050.9	37.27157	2180.5	912.6	37.13185	2162.7	1032.9	37.44357

G.2 Algoritma DE Plus

No	Populasi=50, B. Bawah=0, B. Atas=1, CR min=0.3, CR max=0.9, F min=0.5			Populasi=50, B. Bawah=0, B. Atas=1, CR min=0.4, CR max=0.8, F min=0.5			Populasi=50, B. Bawah=0, B. Atas=1, CR min=0.5, CR max=0.7, F min=0.5		
	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>
1	2155	1611	1047.0977	2153	1305	1034.9387	2143	1689	1051.1678
2	2142	1733	1047.5487	2141	1378	1064.1601	2155	1688	1049.5768
3	2152	1346	1055.8361	2168	1690	1057.0943	2154	1956	1054.0055
4	2152	1980	1047.2573	2145	1876	1068.3525	2141	1744	1054.6938
5	2152	1281	1043.6777	2153	1305	1040.0329	2163	1397	1056.8472
6	2142	1740	1049.0622	2141	1378	1041.2476	2152	1996	1054.8789
7	2150	1927	1139.6622	2168	1690	1045.735	2151	1831	1047.8252
8	2146	1843	1041.9537	2145	1876	1059.8667	2154	1940	1086.9602

Digital Repository Universitas Jember

9	2136	1936	1047.529	2160	1874	1049.0668	2155	1873	1053.7589
10	2146	1804	1046.1471	2146	1541	1052.6847	2157	1969	1055.338
Rata-rata	2147.3	1720.1	1056.57717	2152	1591.3	1051.31793	2152.5	1808.3	1056.50523



LAMPIRAN H. Hasil Percobaan untuk Data *reC27* (2000 Iterasi)

H.1 Algoritma CEGA

No	N=50, rho=0.1, alfa=0.5, Pc=1			N=50, rho=0.1, alfa=0.6, Pc=1			N=50, rho=0.1, alfa=0.7, Pc=1			N=50, rho=0.1, alfa=0.8, Pc=1		
	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>
1	2446	1279	45.3614	2484	1007	45.1191	2451	1794	46.0522	2451	1794	46.0984
2	2441	1324	45.0159	2434	1969	46.1188	2419	1393	45.0409	2419	1393	45.6964
3	2461	1912	45.6393	2484	1007	44.9467	2464	1250	45.8212	2464	1250	45.6721
4	2446	1279	44.9287	2434	1969	45.2257	2487	573	45.2343	2487	573	45.7198
5	2441	1324	45.5371	2481	798	45.4574	2512	323	45.399	2512	323	45.7742
6	2461	1912	45.2447	2435	755	45.108	2459	1044	45.513	2459	1044	45.5244
7	2396	873	45.6119	2432	826	44.819	2448	1248	45.1357	2448	1248	45.6454
8	2493	1178	46.0195	2423	1692	45.2994	2463	535	45.2911	2463	535	46.2496
9	2479	1921	45.4461	2433	1566	45.1407	2467	523	46.9118	2467	523	45.3264
10	2469	1710	45.317	2460	1281	44.9876	2455	1841	45.8301	2455	1841	46.0106
Rata-rata	2453.3	1471.2	45.41216	2450	1287	45.22224	2462.5	1052.4	45.62293	2462.5	1052.4	45.77173

H.2 Algoritma DE Plus

No	Populasi=50, B. Bawah=0, B. Atas=1, CR min=0.3, CR max=0.9, F min=0.5			Populasi=50, B. Bawah=0, B. Atas=1, CR min=0.4, CR max=0.8, F min=0.5			Populasi=50, B. Bawah=0, B. Atas=1, CR min=0.5, CR max=0.7, F min=0.5		
	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>	<i>Makespan</i>	<i>Iterasi Konvergen</i>	<i>Running Time</i>
1	2445	1618	1314.968	2444	1024	1349.1003	2446	1616	1315.1444
2	2457	1526	1295.0663	2447	1930	1310.8431	2430	1952	1312.3019
3	2434	1431	1308.0885	2444	1024	1312.3584	2450	1301	1318.0048
4	2445	1618	1319.0702	2447	1930	1310.161	2452	1718	1323.6507
5	2457	1526	1296.4208	2443	1615	1314.234	2448	1634	1322.4409
6	2434	1431	1290.6969	2447	1845	1314.819	2447	1301	1332.9404
7	2446	1783	1311.6178	2426	1938	1291.7263	2457	1733	1321.1233
8	2451	1993	1314.4501	2446	1843	1295.419	2453	1829	1331.0223

9	2445	1334	1291.1857	2433	1866	1315.7214	2446	1616	1298.8587
10	2458	1621	1307.683	2444	1426	1332.9751	2430	1952	1310.7423
Rata-rata	2447.2	1588.1	1304.92473	2442.1	1644.1	1314.73576	2445.9	1665.2	1318.62297

