



**PENYELESAIAN PENJADWALAN *FLOWSHOP* DENGAN ALGORITMA  
*ARTIFICIAL IMMUNE SYSTEM* DAN ALGORITMA *DIFFERENTIAL  
EVOLUTION PLUS***

**SKRIPSI**

Oleh

**Arista Rosita Dewi  
NIM 101810101028**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2015**



**PENYELESAIAN PENJADWALAN *FLOWSHOP* DENGAN ALGORITMA  
*ARTIFICIAL IMMUNE SYSTEM* 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

**Arista Rosita Dewi  
NIM 101810101028**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2015**

## **PERSEMBAHAN**

Skripsi ini saya persembahkan untuk:

1. Ibu Sulistyowati dan Bapak Sulhan yang senantiasa mendoakan, memberikan bimbingan, semangat serta kasih sayang yang tulus;
2. Mas Dani Pranoto sekeluarga yang kusayangi;
3. Para dosen dan guru penulis sejak taman kanak-kanak sampai perguruan tinggi yang senantiasa membimbing dan membagi ilmu dengan tulus;
4. Almamater Jurusan Matematika FMIPA Universitas Jember, SMAN 1 Rogojampi, SMPN 1 Rogojampi, SDN 2 Bubuk, dan TK Tunas Bangsa yang telah menjadi sarana menuntut ilmu.

**MOTTO**

“Allah akan meninggikan orang-orang yang beriman di antara kamu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.”

(terjemahan Q.S. Al-Mujadalah ayat 11)<sup>\*)</sup>

“Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari suatu urusan), tetaplah bekerja keras untuk (urusan yang lain) dan hanya kepada Tuhanmulah engkau berharap.”

(terjemahan Q.S. Al-Insyirah ayat 6-8)<sup>\*)</sup>

“Anda tidak bisa mengubah orang lain, Anda yang harus menjadi perubahan yang Anda harapkan dari orang lain.”

(Mahatma Gandhi)

---

<sup>\*)</sup> Departemen Agama Republik Indonesia. 2004. *Al-Qur'an dan Terjemahannya*. Bandung: CV. Penerbit Diponegoro.

**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Arista Rosita Dewi

NIM : 101810101028

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penyelesaian Penjadwalan *Flowshop* dengan Algoritma *Artificial Immune System* dan Algoritma *Differential Evolution Plus*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan dalam 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 2015

Yang menyatakan,

Arista Rosita Dewi  
NIM 101810101028

**SKRIPSI**

**PENYELESAIAN PENJADWALAN FLOWSHOP DENGAN  
ALGORITMA *ARTIFICIAL IMMUNE SYSTEM* DAN ALGORITMA  
*DIFFERENTIAL EVOLUTION PLUS***

Oleh  
**Arista RositaDewi**  
**NIM 10181010144**

**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 *Artificial Immune System* dan Algoritma *Differential Evolution Plus*” telah diuji dan disahkan pada:

Hari, tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas  
Jember

Tim Penguji

Ketua,

Sekretaris,

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

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

Anggota Tim Penguji

Anggota I,

Anggota II,

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

Kosala Dwidja Purnomo, S.Si., M.Si.  
NIP.196908281998021001

Mengesahkan  
Dekan,

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



## RINGKASAN

**Penyelesaian Penjadwalan *Flowshop* dengan Algoritma *Artificial Immune System* dan Algoritma *Differential Evolution Plus***; Arista Rosita Dewi, 101810101028; 2015: 40 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penjadwalan produksi dapat diartikan sebagai pengalokasian sumber daya untuk mengerjakan sejumlah pekerjaan dengan tujuan memperoleh jadwal produksi yang optimal. Berdasarkan pola aliran prosesnya, penjadwalan produksi dapat diklasifikasikan dalam dua bentuk, yaitu *flowshop* dan *jobshop*. Salah satu masalah penjadwalan yang sering muncul adalah penjadwalan produksi *flowshop*. Pada permasalahan penjadwalan produksi *flowshop*, setiap *job* dapat diproses dengan urutan yang sama pada paling tidak satu mesin dan satu mesin dapat memproses paling banyak satu *job*.

Algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus* merupakan algoritma yang dapat menyelesaikan permasalahan optimasi. Dalam tugas akhir ini kedua algoritma dibandingkan untuk menyelesaikan penjadwalan produksi *flowshop* dengan tujuan mencari solusi terbaik berdasarkan nilai *makespan* minimum dan tingkat kecepatan kekonvergenan.

Hasil dari penelitian yang diterapkan untuk penjadwalan produksi sepatu menghasilkan urutan jadwal dan nilai *makespan* optimal yang sama yaitu *makespan* sebesar 1484 menit dengan urutan jadwal model sepatu casual – model sepatu ceko – model sepatu fram – model sepatu mogasen – model sepatu pantofel C – model sepatu pantofel B – model sepatu safety – model sepatu pantofel A. Namun pada sepuluh kali pengujian dengan jumlah iterasi yang sama algoritma *Artificial Immune System* memberikan hasil yang berubah-ubah untuk nilai *makespan* pada setiap pengujian, sehingga nilai *makespan* rata-rata yang didapat adalah 1494,6 menit dengan urutan jadwal berbeda, sedangkan pada algoritma



*Differential Evolution Plus* hasil *makespan* selalu menunjukkan nilai yang sama pada setiap pengujian dengan urutan jadwal yang sama pula, sehingga nilai *makespan* rata-rata yang didapat adalah 1484 menit. Hal ini berarti bahwa penggunaan algoritma *Differential Evolution Plus* lebih efektif dibandingkan dengan algoritma *Artificial Immune System* jika diterapkan pada penjadwalan produksi sepatu.

Ditinjau berdasarkan tingkat kecepatan kekonvergenan, algoritma *Artificial Immune System* memiliki tingkat kecepatan kekonvergenan yang lebih baik dibanding dengan algoritma *Differential Evolution Plus* karena lebih cepat konvergen.

## PRAKATA

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan karuniaNya sehingga skripsi yang berjudul “Penyelesaian Penjadwalan *Flowshop* dengan Algoritma *Artificial Immune System* dan Algoritma *Differential Evolution Plus*” dapat terselesaikan. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata 1 (S1) di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember. Sholawat dan salam semoga tercurahkan kepada junjungan Nabi besar Muhammad SAW yang telah menjadi pembawa rahmatan lil’alamin.

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

1. Bapak Ahmad Kamsyakawuni, S.Si., M.Kom. selaku Dosen Pembimbing Utama dan Bapak Kusbudiono, S.Si., M.Si. selaku Dosen Pembimbing Anggota yang telah memberikan bimbingan dan bantuan untuk penyempurnaan skripsi ini;
2. Bapak Drs. Rusli Hidayat, M.Sc. selaku Dosen Penguji I dan Bapak Kosala Dwidja Purnomo, S.Si., M.Si. selaku Dosen Penguji II yang telah memberikan kritik dan saran yang membangun untuk penyempurnaan skripsi ini;
3. Seluruh Dosen dan Karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam yang telah banyak membantu kegiatan belajar dan mengajar selama menjalani perkuliahan;
4. Ibu Sulistyowati dan Bapak Sulhan tercinta yang telah membimbing dengan penuh kasih sayang;
5. Mas Dani Pranoto, S.E. dan Mbak Nurul Khoiriyah, S.E. yang selalu memberi cambukan semangat, nasehat dan motivasi untuk selalu menjadi lebih baik. Serta malaikat kecil Moh Azzam Azmi yang menambah senyum dan semangat;

6. Pak Wek Suroso (Alm) panutan terbaik, Mak Aniah, Mak Yek Surani (Alm) pemberi kasih sayang, dan Anang Bodos (Alm);
7. Pak Tusok dan Mak Tin selaku orangtua kedua yang memberi nasehat serta kasih sayang;
8. Cak Omon sekeluarga, De Suyit sekeluarga, Lek Soad sekeluarga, dan Lek In sekeluarga yang memberi bantuan dan dukungan kekeluargaan yang baik;
9. Orang-orang tersayang Frendi, Syukma, Tanty, Dek Hol, Wanda, Vina yang selalu mendukung dan berbagi suka dan duka;
10. Dek Andri terima kasih untuk printernya, Anggita, Firda yang rela meminjam laptop, penghuni kos Pondok Putri 72 Niken, Tika, Puput, Mbak Opi dan penghuni eksekutif 66 serta Bohemian Club atas kekeluargaannya;
11. Seluruh anggota Unit Kegiatan Mahasiswa Seni TITIK FMIPA Universitas Jember yang menanam rasa kekeluargaan, nilai-nilai positif, dan begitu banyak memberi pengalaman berharga;
12. Ummy, Andina, Laily, Adit, Trian, Diah dan semua teman-teman MATHGIC 2010 tersayang, terima kasih atas kekompakan dan persahabatannya;
13. Semua pihak yang telah banyak membantu dan tidak dapat disebutkan satu per satu.

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

Jember, Juli 2015

Arista Rosita Dewi

DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	ii
HALAMAN PERSEMBAHAN .....	iii
HALAMAN MOTTO .....	iv
HALAMAN PERNYATAAN .....	v
HALAMAN PEMBIMBINGAN .....	vi
HALAMAN PENGESAHAN .....	vii
RINGKASAN .....	viii
PRAKATA .....	x
DAFTAR ISI .....	xii
DAFTAR TABEL .....	xiv
DAFTAR GAMBAR .....	xv
DAFTAR LAMPIRAN .....	xvi
<b>BAB 1. PENDAHULUAN</b> .....	<b>1</b>
<b>1.1 Latar Belakang</b> .....	<b>1</b>
<b>1.2 Rumusan Masalah</b> .....	<b>3</b>
<b>1.3 Batasan Masalah</b> .....	<b>3</b>
<b>1.4 Tujuan</b> .....	<b>3</b>
<b>1.5 Manfaat</b> .....	<b>3</b>
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	<b>4</b>
<b>2.1 Penjadwalan Produksi</b> .....	<b>4</b>
<b>2.2 Penjadwalan <i>Flowshop</i></b> .....	<b>5</b>
<b>2.3 Diagram <i>Gantt</i></b> .....	<b>6</b>
<b>2.4 Algoritma <i>Artificial Immune System</i></b> .....	<b>7</b>
<b>2.5 Algoritma <i>Differential Evolution Plus</i></b> .....	<b>12</b>
<b>2.6 Kriteria Kekonvergenan</b> .....	<b>16</b>
<b>BAB 3. METODE PENELITIAN</b> .....	<b>17</b>

<b>3.1 Data Penelitian</b> .....	17
<b>3.2 Langkah-langkah Penelitian</b> .....	17
<b>BAB 4. HASIL DAN PEMBAHASAN</b> .....	20
<b>4.1 Perhitungan Manual</b> .....	20
4.1.1 Perhitungan Manual dengan Algoritma <i>Artificial Immune System</i> .....	20
4.1.2 Perhitungan Manual dengan Algoritma <i>Differential Evolution Plus</i> .....	24
<b>4.2 Program</b> .....	29
<b>4.3 Penyelesaian Penjadwalan <i>Flowshop</i> Menggunakan Program</b> .....	32
<b>4.4 Tingkat Konvergensi Algoritma</b> .....	36
<b>BAB 5. PENUTUP</b> .....	38
<b>5.1 Kesimpulan</b> .....	38
<b>5.2 Saran</b> .....	38
<b>DAFTAR PUSTAKA</b> .....	40
<b>LAMPIRAN-LAMPIRAN</b>	
<b>A. LAMPIRAN A. PENGUJIAN ALGORITMA DIFFENTIAL EVOLUTION PLUS DENGAN <math>Cr_{min}</math>, <math>Cr_{max}</math>, dan <math>F_{min}</math> BERUBAH</b>	42
<b>B. PENGUJIAN DENGAN ITERASI SAMA DAN JUMLAH POPULASI SAMA</b> .....	45
<b>C. PENGUJIAN DENGAN ITERASI SAMA DAN JUMLAH POPULASI BERBEDA</b> .....	46
<b>D. PENGUJIAN DENGAN ITERASI BERBEDA DAN JUMLAH POPULASI SAMA</b> .....	50

**DAFTAR TABEL**

	Halaman
3.1 Data waktu proses pembuatan sepatu (menit) .....	17
4.1 Data contoh .....	20
4.2 <i>Makespan</i> $J_2 - J_5 - J_3 - J_1 - J_4$ .....	21
4.3 <i>Makespan</i> $J_4 - J_1 - J_5 - J_2 - J_3$ .....	21
4.4 <i>Makespan</i> $J_5 - J_2 - J_1 - J_4 - J_3$ .....	21
4.5 Perhitungan <i>fitness value</i> , nilai probabilitas dan interval .....	22
4.6 Hasil <i>Cloning</i> .....	23
4.7 Urutan <i>job</i> mutasi kebalikan .....	23
4.8 Urutan <i>job</i> mutasi penukaran pasangan .....	24
4.9 <i>Makespan</i> rata-rata untuk $Cr_{min}$ , $Cr_{max}$ , dan $F_{min}$ berbeda .....	33
4.10 <i>Makespan</i> rata-rata untuk jumlah populasi berbeda .....	34
4.11 <i>Makespan</i> rata-rata untuk jumlah generasi (iterasi) berbeda .....	35
4.12 Pengujian Konvergensi .....	36



**DAFTAR GAMBAR**

	Halaman
2.1 Pola aliran <i>Pure Flowshop</i> .....	6
2.2 Pola aliran <i>General Flowshop</i> .....	6
2.3 Peta <i>Gantt (Gantt Chart)</i> .....	7
2.4 Skema langkah-langkah algoritma <i>Artificial Immune System</i> .....	11
2.5 Skema algoritma <i>Differential Evolution Plus</i> .....	15
3.1 Skema langkah-langkah penelitian .....	19
4.1 Tampilan awal program aplikasi .....	29
4.2 Tampilan <i>input</i> data .....	30
4.3 Tampilan <i>input</i> data pembuatan sepatu .....	30
4.4 Tampilan menu proses data .....	31
4.5 Tampilan hasil <i>output</i> urutan <i>job</i> , <i>makespan</i> , dan grafik kekonvergenan	31
4.6 Tampilan <i>Gantt Chat</i> .....	32
4.7 Grafik <i>makespan</i> rata-rata untuk jumlah populasi berbeda .....	35
4.8 Grafik <i>makespan</i> rata-rata untuk jumlah generasi (iterasi) berbeda ..	36

DAFTAR LAMPIRAN

	Halaman
<b>A. PENGUJIAN ALGORITMA DIFFENTIAL EVOLUTION PLUS</b>	
<b>DENGAN <math>Cr_{min}</math>, <math>Cr_{max}</math>, dan <math>F_{min}</math> BERUBAH</b> .....	42
<b>A.1 Tabel hasil sepuluh kali pengujian dengan <math>Cr_{min} = 0,2</math>, <math>Cr_{max} = 0,6</math>, dan <math>F_{min} = 0,3</math></b> .....	42
<b>A.2 Tabel hasil sepuluh kali pengujian dengan <math>Cr_{min} = 0,2</math>, <math>Cr_{max} = 0,6</math>, dan <math>F_{min} = 0,6</math></b> .....	42
<b>A.3 Tabel hasil sepuluh kali pengujian dengan <math>Cr_{min} = 0,2</math>, <math>Cr_{max} = 0,6</math>, dan <math>F_{min} = 0,9</math></b> .....	43
<b>A.4 Tabel hasil sepuluh kali pengujian dengan <math>Cr_{min} = 0,5</math>, <math>Cr_{max} = 0,9</math>, dan <math>F_{min} = 0,3</math></b> .....	43
<b>A.5 Tabel hasil sepuluh kali pengujian dengan <math>Cr_{min} = 0,5</math>, <math>Cr_{max} = 0,9</math>, dan <math>F_{min} = 0,6</math></b> .....	44
<b>A.6 Tabel hasil sepuluh kali pengujian dengan <math>Cr_{min} = 0,5</math>, <math>Cr_{max} = 0,9</math>, dan <math>F_{min} = 0,9</math></b> .....	44
<b>B. PENGUJIAN DENGAN ITERASI SAMA DAN JUMLAH POPULASI</b>	
<b>SAMA</b> .....	45
<b>B.1 Tabel hasil sepuluh kali pengujian dengan iterasi sama (200) dan jumlah populasi sama (5) pada algoritma <i>Artificial Immune</i> <i>System</i></b> .....	45
<b>B.2 Tabel hasil sepuluh kali pengujian dengan iterasi sama (200) dan jumlah populasi sama (5) pada algoritma <i>Differential Evolution</i> <i>Plus</i></b> .....	45
<b>C. PENGUJIAN DENGAN ITERASI SAMA DAN JUMLAH</b>	
<b>POPULASI BERBEDA</b> .....	46
<b>C.1a Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 3 pada algoritma <i>Artificial Immune System</i></b>	46

C.1b	Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 3 pada algoritma <i>Differential Evolution Plus</i>	46
C.2a	Tabel hasil sepuluh kali pengujian dengan iterasi =200 dan jumlah populasi = 5 pada algoritma <i>Artificial Immune System</i>	47
C.2b	Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 5 pada algoritma <i>Differential Evolution Plus</i>	47
C.3a	Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 7 pada algoritma <i>Artificial Immune System</i>	48
C.3b	Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 7 pada algoritma <i>Differential Evolution Plus</i>	48
C.4a	Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 10 pada algoritma <i>Artificial Immune System</i>	49
C.4b	Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 10 pada algoritma <i>Differential Evolution Plus</i>	49
<b>D. PENGUJIAN DENGAN ITERASI BERBEDA DAN JUMLAH POPULASI SAMA</b>		
	.....	50
D.1a	Tabel hasil sepuluh kali pengujian dengan iterasi = 10 dan jumlah populasi = 5 pada algoritma <i>Artificial Immune System</i>	50
D.1b	Tabel hasil sepuluh kali pengujian dengan iterasi = 10 dan jumlah populasi = 5 pada algoritma <i>Differential Evolution Plus</i>	50
D.2a	Tabel hasil sepuluh kali pengujian dengan iterasi = 50 dan jumlah populasi = 5 pada algoritma <i>Artificial Immune System</i>	51
D.2b	Tabel hasil sepuluh kali pengujian dengan iterasi = 50 dan jumlah populasi = 5 pada algoritma <i>Differential Evolution Plus</i>	51
D.3a	Tabel hasil sepuluh kali pengujian dengan iterasi = 100 dan jumlah populasi = 5 pada algoritma <i>Artificial Immune System</i>	52
D.3b	Tabel hasil sepuluh kali pengujian dengan iterasi = 100 dan jumlah populasi = 5 pada algoritma <i>Differential Evolution Plus</i>	52
D.4a	Tabel hasil sepuluh kali pengujian dengan iterasi = 150 dan jumlah populasi = 5 pada algoritma <i>Artificial Immune System</i>	53

**D.4b** Tabel hasil sepuluh kali pengujian dengan iterasi = 150 dan jumlah populasi = 5 pada algoritma *Differential Evolution Plus* ..... 53



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Suatu perusahaan perlu melakukan pengaturan dan perencanaan kerja yang baik dalam menjalankan usaha untuk mendapatkan keuntungan yang tinggi, kualitas yang baik, biaya rendah, atau efisiensi sumber daya. Salah satu usaha untuk mencapai perencanaan kerja yang baik dapat dilakukan dengan membuat penjadwalan proses produksi yang terencana.

Penjadwalan secara umum merupakan suatu proses pengaturan sumber daya yang tersedia untuk menyelesaikan suatu pekerjaan dalam periode waktu tertentu agar tujuan yang ingin dicapai terpenuhi. Penjadwalan dapat muncul dalam berbagai bidang, seperti penjadwalan produksi, penjadwalan tenaga kerja, penjadwalan mata kuliah, dan lain-lain. Tujuan penjadwalan adalah untuk mengoptimalkan satu atau beberapa fungsi tujuan, dimana fungsi tujuan bisa bermacam-macam tergantung dari aplikasi masalahnya. Kegagalan membuat jadwal atau menjalankan jadwal yang salah dapat memperlambat selesainya pekerjaan dan menghabiskan biaya yang banyak. Oleh karena itu, masalah penjadwalan menjadi sangat penting agar tercapai efisiensi dalam suatu pekerjaan.

Penjadwalan produksi dapat diartikan sebagai pengalokasian sumber daya untuk mengerjakan operasi-operasi tertentu dengan tujuan memperoleh jadwal produksi yang optimal. Dalam penjadwalan produksi yang dimaksud sebagai operasi adalah *job*, sedangkan yang dimaksud dengan sumber daya adalah mesin. Adapun tujuan dari penjadwalan produksi adalah untuk mencari urutan *job* yang paling optimal. Berdasarkan pola aliran prosesnya, penjadwalan produksi dapat diklasifikasikan dalam dua bentuk, yaitu *flowshop* dan *jobshop*.

Salah satu masalah penjadwalan yang sering muncul adalah penjadwalan produksi *flowshop*. Pada permasalahan penjadwalan produksi *flowshop*, setiap *job* dapat diproses dengan urutan yang sama pada paling tidak satu mesin dan satu mesin dapat memproses paling banyak satu *job*.



Penelitian Ginting (2006) mengenai studi aplikasi metode *Artificial Immune System* dalam penjadwalan *flowshop* menghasilkan bahwa perhitungan *makespan* untuk dua order yang datang dapat memenuhi *due date* tiap order. Dalam penelitian lain, Karolina (2013) melakukan penelitian usulan perbaikan penjadwalan produksi dengan algoritma *Artificial Immune System*. Dalam penelitian tersebut menghasilkan algoritma *Artificial Immune System* lebih baik dibandingkan dengan algoritma Campbell, Dudek, and Smith (CDS) ditinjau dari nilai *makespan*.

Salah satu algoritma yang mempunyai reputasi optimasi global optima yang efektif adalah *Differential Evolution* (DE). Keunggulan dari DE adalah konsep yang sederhana, implementasi yang mudah dan cepat konvergen, namun kinerja DE sangat tergantung dari parameternya (Qian *et al.*, 2008). Wiratno (2012) mengembangkan algoritma *Differential Evolution* pada penjadwalan *flowshop* dengan menambahkan *adaptive parameters*, strategi *local search* dan memodifikasi *crossover* yang menghasilkan algoritma DE yang diusulkan (*Differential Evolution Plus*) memiliki kinerja yang lebih baik dibandingkan dengan algoritma DE murni, algoritma Genetika, dan pada kasus tertentu juga memiliki kinerja yang lebih baik dibandingkan algoritma *Multi-Objective Ant Colony System* (MOACSA). Pada penelitian Sari (2015) yaitu perbandingan algoritma *Simulated Annealing* dan algoritma *Differential Evolution Plus* untuk penjadwalan *flowshop* menunjukkan bahwa pada perbandingan tingkat konvergensi, algoritma *Differential Evolution Plus* memberikan hasil lebih cepat konvergen dibanding dengan algoritma *Simulated Annealing*.

Berdasarkan uraian diatas, menjadi bahan pertimbangan bagi penulis untuk membandingkan hasil dari algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus* pada penjadwalan *flowshop* untuk menentukan algoritma mana yang lebih baik ditinjau dari nilai *makespan* serta tingkat kecepatan kekonvergenan.



## 1.2 Rumusan Masalah

Permasalahan yang akan diselesaikan dalam penulisan tugas akhir ini adalah bagaimana melakukan penjadwalan *flowshop* menggunakan algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus*, kemudian membandingkan hasil kedua algoritma tersebut ditinjau dari nilai *makespan* dan tingkat kecepatan kekonvergenan.

## 1.3 Batasan Masalah

Batasan masalah yang terdapat dalam tugas akhir 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 *transport* (perpindahan material) diabaikan karena jarak antara mesin satu ke mesin lain sangat berdekatan;
- e. Selama proses produksi tidak ada pembatalan *job* maupun adanya sisipan *job*.

## 1.4 Tujuan

Tujuan yang ingin dicapai dalam penulisan tugas akhir ini adalah melakukan penjadwalan *flowshop* dengan algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus*, selanjutnya melakukan perbandingan hasil kedua algoritma tersebut ditinjau dari nilai *makespan* dan tingkat kecepatan kekonvergenan.

## 1.5 Manfaat

Manfaat yang diharapkan dari penulisan tugas akhir ini adalah memberi informasi kepada pembaca tentang algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus* serta perbedaan hasil kedua algoritma tersebut dalam menyelesaikan penjadwalan *flowshop* ditinjau dari nilai *makespan* dan tingkat kecepatan kekonvergenan.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Penjadwalan Produksi

Pengertian penjadwalan secara umum dapat diartikan seperti: “*Scheduling is the allocation of resources overtime to perform collection of risk*”, yang artinya penjadwalan adalah pengalokasian sumber daya yang terbatas untuk mengerjakan sejumlah pekerjaan (Ginting, 2009). Sumber daya ini tidak hanya berupa mesin yang menjalankan pekerjaan tetapi juga mencakup tenaga kerja yang diperlukan untuk mengoperasikan mesin tersebut. Menurut Perdana (2011), penjadwalan juga dapat didefinisikan sebagai pengambilan keputusan tentang penyesuaian aktivitas dan sumber daya dalam rangka menyelesaikan sekumpulan *jobs*/proyek agar tepat pada waktunya dan memiliki kualitas seperti yang diinginkan. Keputusan yang dibuat dalam penjadwalan meliputi pengurutan pekerjaan (*sequencing*); waktu mulai dan selesai pekerjaan (*timing*); dan urutan operasi untuk suatu pekerjaan (*routing*).

Dalam melakukan penjadwalan terdapat berbagai tujuan yang ingin dicapai dengan ketersediaan sumber daya, yaitu (Karolina, 2013):

a. Peningkatan utilisasi penggunaan sumber daya

Peningkatan utilisasi dapat dilakukan dengan menekan waktu menganggur sumber daya tersebut, sehingga total waktu proses dapat berkurang dan produktivitasnya dapat meningkat.

b. Meniminasi *makespan*

Tujuan ini dapat dicapai dengan cara meminimasi jumlah pekerjaan yang menunggu dalam antrian proses ketika sumber daya yang ada masih mengerjakan tugas yang lain. Indikator jumlah antrian pekerjaan ini dinyatakan dengan besaran waktu alir rata-rata.

c. Mengurangi kelambatan

Yang dimaksud mengurangi kelambatan adalah dengan menjamin pemenuhan *due date* setiap order. Sejumlah pekerjaan memiliki batas waktu penyelesaian

(*due date*) dan bila waktu penyelesaian pekerjaan melewati batas waktu yang seharusnya maka akan dikenakan denda atau sanksi.

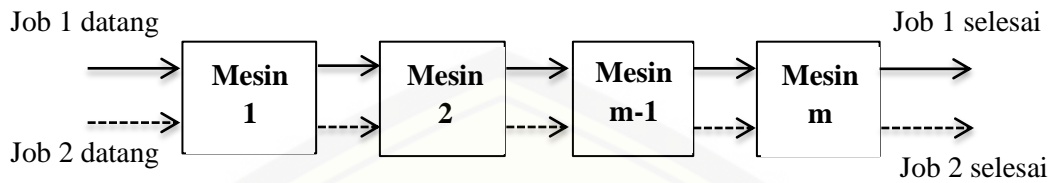
Permasalahan penjadwalan biasanya berkaitan dengan pengurutan pembuatan atau pengerjaan produk secara menyeluruh terhadap sejumlah mesin yang tersedia. Penjadwalan jenis ini biasa disebut dengan penjadwalan produksi. Penjadwalan produksi melibatkan  $n$  pekerjaan (*job*) dan  $m$  mesin dalam proses produksinya, dimana setiap *job* mengandung informasi tentang jenis produk dan jumlah pesanan. Berdasarkan pola aliran prosesnya penjadwalan produksi dibedakan menjadi *flowshop*, dimana *job* akan diproses seluruhnya mengalir pada arah jalur produk yang sama dan *jobshop*, dimana tiap operasi memiliki aliran yang berbeda (Ginting, 2009).

## 2.2 Penjadwalan *Flowshop*

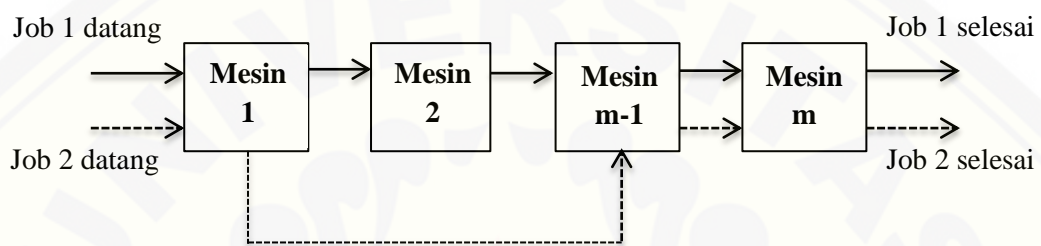
Penjadwalan *flowshop* merupakan model penjadwalan dimana *job-job* yang akan diproses seluruhnya mengalir pada arah atau jalur yang sama. Pada permasalahan penjadwalan *flowshop*, setiap *job* dapat diproses dengan urutan yang sama pada paling tidak satu mesin dan satu mesin dapat memproses paling banyak satu *job*. Mesin-mesin dalam *flowshop* memiliki urutan  $1, 2, \dots, m$  dan operasi dari *job*  $i$  bersamaan dengan urutan mesin yaitu  $(i, 1), (i, 2), \dots, (i, m)$ . 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 meminimumkan *makespan*. *Makespan* adalah total waktu terbesar antara jumlah waktu per *job* atau total waktu proses yang dibutuhkan untuk menyelesaikan sekumpulan *job*.

Penjadwalan *flowshop* dibagi menjadi 2, yaitu *pure flowshop* dan *general flowshop*. Pada aliran pekerjaan *pure flowshop*, semua pekerjaan akan mengalir pada lini produksi yang sama dimana *job* yang datang dikerjakan di semua mesin seperti pada Gambar 2.1. sedangkan pada *general flowshop*, beberapa *job* yang

datang tidak harus dikerjakan pada semua mesin, dapat dilihat pada Gambar 2.2 (Sari, 2015).



Gambar 2.1 Pola aliran *Pure Flowshop*

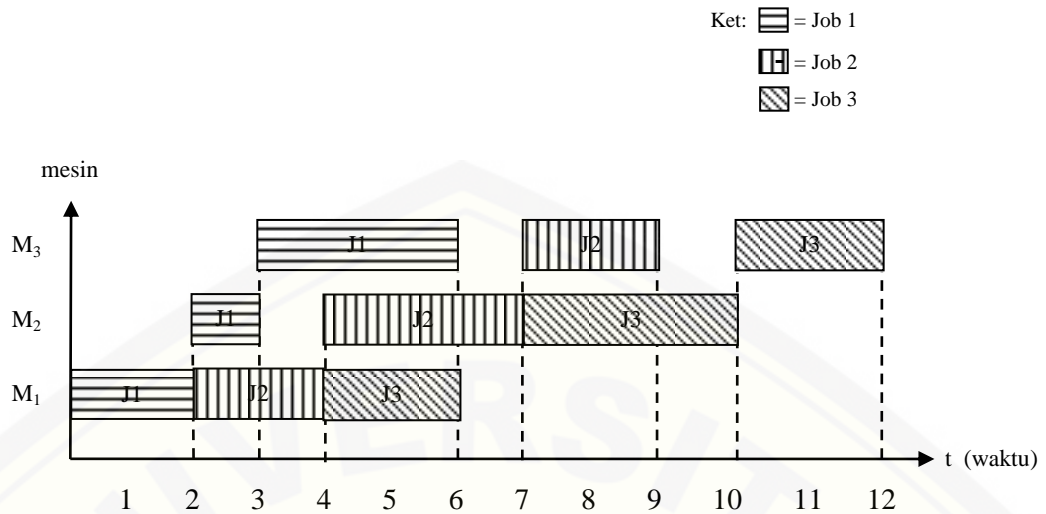


Gambar 2.2 Pola aliran *General flowshop*

### 2.3 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 satu model yang sederhana dan umum digunakan secara luas yakni peta *Gantt* (*Gantt 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. *Gantt Chart* dapat ditunjukkan pada Gambar 2.3.





Gambar 2.3 Peta Gantt (Gantt Chart)

Dari *Gantt Chart* kemudian ditentukan urutan (*sequence*) dari *job* yang memberikan kriteria penjadwalan terbaik, misalnya waktu pemrosesan tersingkat, utilitas mesin, *idletime* minimum, dan lain-lain (Ginting, 2009).

#### 2.4 Algoritma *Artificial Immune System*

*Artificial Immune System* (AIS) merupakan sistem komputasi yang diinspirasi oleh teori imunologi, mengamati fungsi, prinsip dan mekanisme kekebalan yang diaplikasi dalam pemecahan masalah. Aplikasi dari AIS mencakup pengamanan dan jaringan komputerisasi, deteksi kesalahan dan penyimpangan, optimisasi, analisis data dan penjadwalan. Mekanisme penggunaan sistem kekebalan yang efisien membuat AIS dapat digunakan dalam masalah penjadwalan. Masing-masing solusi (jadwal lengkap) adalah antibodi. Ribuan *clone* dari individu terbaik digenerasikan. *Clone* dimutasi dan *clone* terbaik dipilih sebagai solusi. Representasi penjadwalan tidak berbeda dengan *Genetic Algorithm*. Algoritma AIS dan Algoritma Genetik memiliki kesamaan karena merupakan turunan dari proses *evolutionary* (Ginting, 2006).

Parameter-parameter yang terdapat dalam *Artificial Immune System* akan mempengaruhi kinerja dari model penjadwalan yang digunakan. Adapun parameter yang terdapat dalam algoritma *Artificial Immune System* adalah sebagai berikut ini.

- a. Jumlah Generasi ( $G$ ) merupakan parameter yang menentukan jumlah pengulangan perhitungan dari antibodi yang sudah terpilih.
- b. Ukuran populasi antibodi ( $N$ ) merupakan parameter yang menentukan jumlah banyaknya urutan yang dibuat secara acak terhadap *job-job* yang akan dikerjakan.

Jadwal yang mungkin pada algoritma AIS direpresentasikan oleh *string* dari  $n$  *job*. Elemen  $n$  dari *string* adalah *job* yang akan diurutkan. Oleh karena itu *string* disusun dari permutasi elemen  $n$  *job*. *String-string* tersebut diterima sebagai antibodi dari *Artificial Immune System*. Algoritma diteruskan untuk mencari solusi dengan evolusi antibodi. Evolusi didasarkan pada dua prinsip dasar dari sistem kekebalan yaitu pemilihan *clone* dan *affinity maturation*.

- a. Proses pemilihan *clone*.

Nilai *affinity* dari masing-masing jadwal dihitung dengan fungsi *affinity*. Rumus fungsi *affinity* sebagai berikut:

$$Affinity(z) = \frac{1}{makespan(z)} \quad (2.1)$$

dimana  $z$  adalah antibodi yang akan dihitung. Dari rumus diatas dapat diketahui nilai *makespan* yang paling rendah sama dengan nilai *affinitas* yang paling tinggi. Pada proses pemilihan *clone*, *cloning* antibodi dilakukan dengan membangkitkan bilangan random sebanyak parameter ukuran populasi antibodi ( $N$ ). Prosedur ini memberikan kesempatan untuk *makespan* ( $C_{max}$ ) dengan nilai yang lebih rendah untuk dipilih dan di *cloning*. Untuk itu akan ada lebih banyak *clone* antibodi dengan nilai yang lebih rendah dari pada *clone* dengan nilai *makespan* yang lebih tinggi pada populasi *clone* generasi selanjutnya. Pemilihan probabilitas masing-masing antibodi dihitung dengan prosedur sebagai berikut (Ginting, 2006):

- 1) Untuk masing-masing antibodi pada populasi dihitung nilai *makespan*.
- 2) Mencari nilai *makespan* maksimum ( $Max C_{max}$ ).
- 3) Untuk masing-masing antibodi dihitung nilai *fitness* dengan rumus sebagai berikut:

$$fitness\ value = (Max\ C_{max} + 1) - (makespan\ antibodi) \quad (2.2)$$



- 4) Untuk masing-masing antibodi, dicari probabilitas yang terpilih dengan rumus sebagai berikut ini.

$$Probability = \frac{fitness\ value\ i}{total\ fitness\ value} \quad (2.3)$$

Dalam algoritma digunakan ukuran tetap dari populasi antibodi, juga digenerasikan sejumlah *clone* dengan ukuran yang sama dari populasi antibodi. *Clone-clone* yang digenerasikan dari masing-masing antibodi, berubah sesuai dengan pemilihan probabilitas antibodi.

- b. Proses *affinity maturation*.

Ada dua tahap prosedur mutasi yang digunakan. Tahap pertama dilalui dengan prosedur mutasi kebalikan. Untuk pengurutan *string*, *i* dan *j* merupakan dua posisi pada pengurutan *string*. Urutan *string* yang lain ditentukan dengan membalikkan urutan *job* antara posisi *i* dan *j* dimana *i* adalah *job* awal sedangkan *j* adalah *job* akhir. Jika nilai *makespan* dari urutan hasil mutasi lebih kecil dari urutan awal maka hasil mutasi ditempatkan pada urutan awal. Jika tidak, urutan akan dimutasi lagi dengan mutasi penukaran pasangan. Untuk mengurutkan *string*, *i* dan *j* secara acak dipilih dua posisi pada urutan *string*. Urutan *string* yang lain ditentukan dengan menukar *job* pada posisi *i* dan *j*. Jika nilai *makespan* dari urutan mutasi lebih kecil dari pada urutan asli, simpan hasil mutasi dan tempatkan pada urutan asli. Jika pada algoritma tidak ditemukan urutan yang lebih baik dari kedua prosedur mutasi maka gunakan urutan asli.

Berikut ini adalah langkah-langkah penjadwalan Algoritma *Artificial Immune System* pada kasus *flowshop*:

- a. Menginputkan data dan menentukan parameter.

Data-data awal yang dibutuhkan dalam penjadwalan yaitu produk (*job*) yang akan dijadwalkan beserta kuantitasnya, jenis-jenis beserta jumlah mesin yang digunakan untuk penjadwalan, waktu proses untuk setiap operasi dari setiap produk, dan *routing* operasi dari setiap produk. Data waktu proses digunakan untuk membuat matriks *routing*. Data urutan proses untuk membuat matriks *routing* dan jumlah *job* merupakan banyaknya komponen yang akan dijadwalkan, dan data jenis mesin serta jumlahnya digunakan untuk mengetahui jenis-jenis mesin dan jumlah setiap jenis mesin yang digunakan

untuk menyelesaikan pekerjaan tersebut. Parameter yang terdapat dalam *Artificial Immune System* adalah jumlah generasi ( $G$ ) dan ukuran populasi antibodi ( $N$ ).

b. Membangkitkan solusi awal

Dalam langkah ini jadwal awal yang mungkin terjadi ditetapkan secara random sebanyak parameter ukuran populasi antibodi ( $N$ ) yang telah ditentukan sebelumnya. Kemudian menghitung nilai *makespan* seluruh jadwal yang ada dan mengurutkan berdasarkan nilai *makespan* yang terkecil hingga terbesar. Selanjutnya menghitung *fitness value* dan nilai probabilitas dengan persamaan (2.2) dan persamaan (2.3). Hasil *makespan* terkecil yang diperoleh akan disimpan sebagai solusi terbaik untuk tahap awal.

c. *Cloning*

Tahap selanjutnya adalah mencari solusi baru dari solusi terbaik yang dihasilkan dari tahap sebelumnya. Solusi baru dibentuk dengan mengkloning setiap solusi awal sesuai bilangan random yang dibangkitkan sebanyak parameter populasi antibodi ( $N$ ).

d. Mutasi kebalikan

Untuk semua *cloning* dilakukan mutasi sehingga mendapatkan antibodi (jadwal) baru. Apabila nilai *makespan* hasil mutasi kebalikan lebih kecil dari *makespan* solusi awal maka dilakukan mutasi penukaran pasangan. Jika tidak, urutan *job* yang akan dipakai selanjutnya adalah urutan *job* awal.

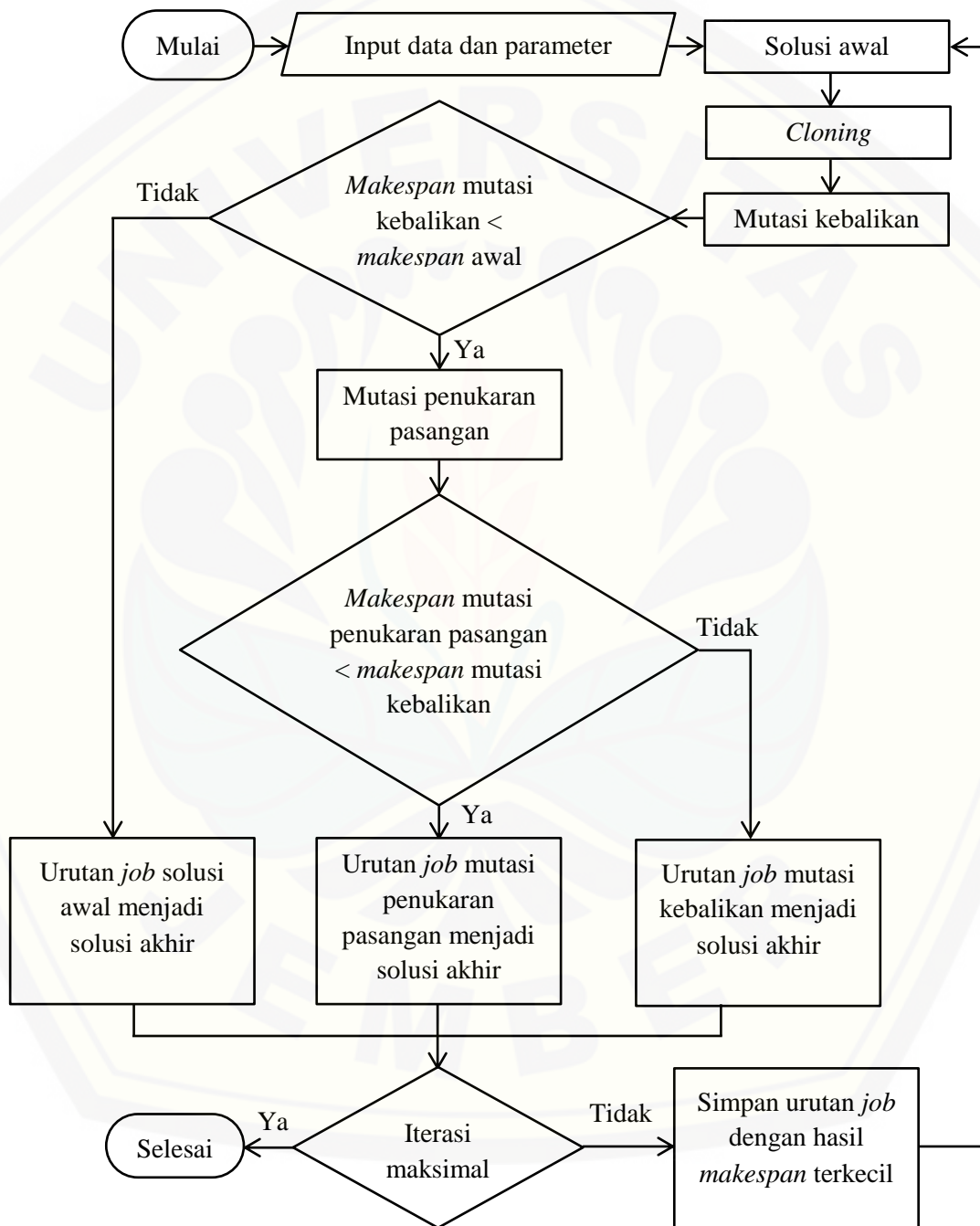
e. Mutasi penukaran pasangan.

Urutan *job* pada mutasi kebalikan mengalami mutasi kembali kemudian ditentukan nilai  $i$  dan  $j$  secara acak dimana urutan  $i$  dan  $j$  akan ditukar secara membalik.  $i$  dan  $j$  merupakan *job-job* yang akan ditukar. Membandingkan *makespan* urutan *job* mutasi kebalikan dengan *makespan* urutan *job* mutasi penukaran pasangan. Jika nilai *makespan* mutasi penukaran pasangan lebih kecil dari nilai *makespan* mutasi kebalikan maka urutan *job* mutasi penukaran pasangan menjadi solusi akhir, jika tidak maka urutan *job* mutasi kebalikan menjadi solusi akhir yang ada.

## f. Mengecek iterasi maksimal

Iterasi maksimal ditentukan untuk memutuskan apakah iterasi selesai atau tidak. Jika iterasi maksimal, maka selesai, solusi terbaik adalah solusi yang optimal. Jika tidak, perhitungan kembali dari langkah b sampai langkah f.

Prosedur algoritma *Artificial Immune System* dapat dilihat sebagai berikut:



Gambar 2.4 Skema langkah-langkah algoritma *Artificial Immune System*

## 2.5 Algoritma *Differential Evolution Plus*

*Differential Evolution* (DE) adalah salah satu metode metaheuristik yang pemakaiannya cukup luas dalam bidang rekayasa. 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 populasi yang sekarang digunakan untuk memandu proses pencarian solusi yang lebih baik.

Tvrdik (2006) menyebutkan bahwa efisiensi pencarian dari DE sangat sensitif terhadap penentuan nilai parameter  $F$  yang berfungsi untuk mengendalikan tingkat pertumbuhan populasi dan nilai parameter  $Cr$  yang berfungsi untuk mengendalikan fraksi nilai variabel yang disalin dari vektor mutan (Nurdiansyah, 2011).

Algoritma DE plus pada penelitian ini merupakan pengembangan dari algoritma DE murni dengan melakukan tiga modifikasi. Pertama, memodifikasi pengendalian nilai parameter  $F$  dan  $Cr$  (disebut *adaptive parameters*) dengan cara menghitung parameter  $F$  dan  $Cr$  pada tiap generasi dengan formula tertentu sehingga nilai parameter  $F$  dan  $Cr$  tiap generasi berubah-ubah. Kedua, menambahkan prosedur *local search* pada algoritma DE untuk meningkatkan kualitas solusi yang dihasilkan. Ketiga, *crossover* yang pada mulanya merupakan langkah untuk menyilangkan populasi target dengan populasi mutan, dimodifikasi menjadi langkah untuk memilih populasi target yang akan dimutasi. Langkah ini bertujuan untuk mengurangi waktu komputasi (Wiratno *et al.*, 2012).

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

a. Inisialisasi populasi

Sebelum melakukan inisialisasi terhadap titik populasi maka perlu dilakukan penentuan batas atas ( $ub$ ) dan batas bawah ( $lb$ ). Untuk membangkitkan nilai awal generasi  $g = 0$ , variabel ke- $j$  dan vektor ke- $i$  bisa diwakili dengan persamaan berikut:

$$x_{j,i,0} = lb_j + rand_j(0,1)(ub_j - lb_j) \quad (2.4)$$

dimana,



$x$  = individu target

$j$  = variabel job  $j$

$i$  = populasi ke- $i$

$lb$  = batas bawah

$ub$  = batas atas

Bilangan random dibangkitkan dengan fungsi  $rand()$ , 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  $x_i$  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  $u_{i,g}$ .

$$u_{i,g} = u_{j,i,g} = \begin{cases} u_{j,i,g}, & \text{jika } j (rand_j(0,1) \leq Cr, \text{ atau } j = j_{rand} \\ u_{j,i,g}, & \text{jika yang lain} \end{cases} \quad (2.5)$$

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

$$Cr = Cr_{min} + G \cdot \frac{Cr_{max} - Cr_{min}}{MAXGEN} \quad (2.6)$$

$Cr_{min}$  dan  $Cr_{max}$  adalah nilai terkecil dan terbesar dari probabilitas *crossover*,  $G$  adalah iterasi pada saat waktu *trunning time*, sedangkan  $MAXGEN$  adalah jumlah maksimum iterasi yang diujicobakan. Tujuan dari penentuan nilai  $Cr$  adalah meningkatkan keragaman vektor yang akan mengalami *crossover* dan menghindari dari *local optima*.



## c. Mutasi

Setelah tahap *crossover*, 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 sebagai berikut:

$$u_{i,g} = x_{r0,g} + F(x_{r1,g} - x_{r2,g}) \quad (2.7)$$

Dimana  $F$  adalah parameter mutasi dan  $x_{r0,g}$ ,  $x_{r1,g}$ ,  $x_{r2,g}$  adalah nilai yang dipilih secara acak dari nilai  $x_i$  pada tahap inisialisasi populasi. Faktor skala  $F \in (0,1)$  adalah bilangan real positif yang mengendalikan tingkat pertumbuhan populasi. Pada langkah ini nilai parameter  $F$  tiap generasi akan berubah-ubah dengan menghitung nilai parameter pada tiap generasi dengan formula:

$$F = \begin{cases} \max\left(F_{min}, 1 - \left|\frac{f_{max}}{f_{min}}\right|\right) & \text{jika } \left|\frac{f_{max}}{f_{min}}\right| < 1 \\ \max\left(F_{min}, 1 - \left|\frac{f_{min}}{f_{max}}\right|\right) & \text{jika yang lain} \end{cases} \quad (2.8)$$

Dimana  $f_{min}$  adalah nilai fungsi minimum dari populasi dan  $f_{max}$  adalah nilai fungsi maksimum dari populasi.  $F_{min}$  merupakan input parameter yang memastikan  $F \in [F_{min}, 1]$ .

## d. Seleksi

Jika *trial vector*  $u_{i,g}$ , mempunyai fungsi tujuan lebih kecil dari fungsi tujuan vektor target yaitu  $x_{i,g}$ , maka  $u_{i,g}$  akan menggantikan posisi  $x_{i,g}$  dalam populasi pada generasi berikutnya. Jika sebaliknya, target akan tetap pada posisinya dalam populasi.

$$x_{i,g+1} = \begin{cases} u_{i,g}, & \text{jika } f(v_{i,g}) \leq f(x_{i,g}) \\ x_{i,g}, & \text{jika sebaliknya} \end{cases} \quad (2.9)$$

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* adalah sebagai berikut:

- 1) Ubah nilai individu  $x_i(t)$  menjadi permutasi job  $\pi_{i,0}$ ;

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 < (nx(n-1)).

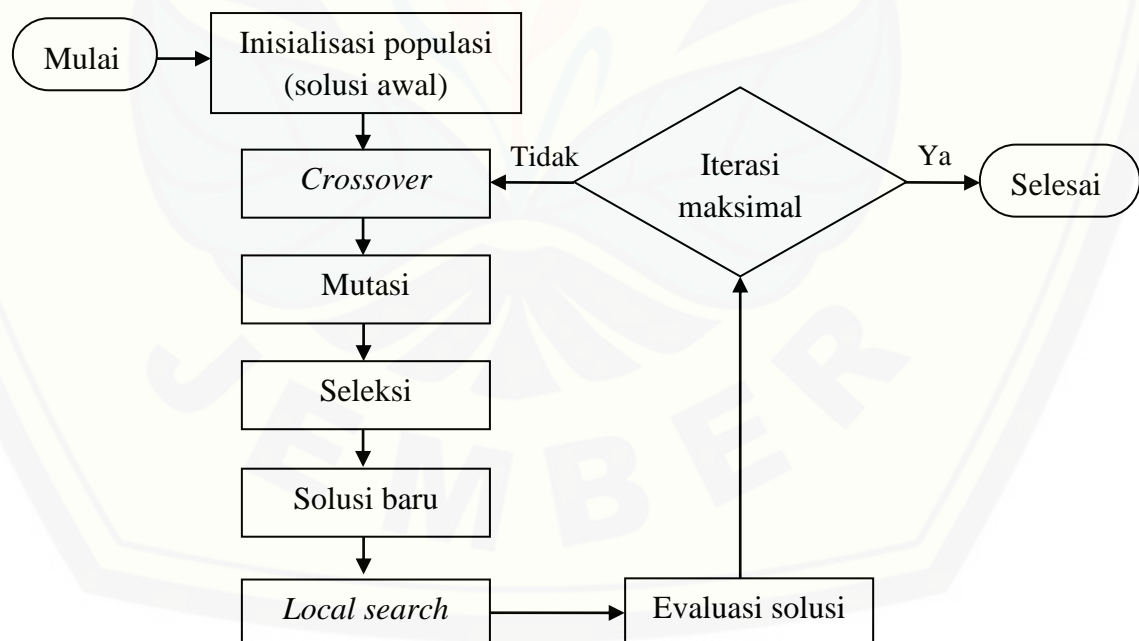
4) jika  $f(\pi_i) < f(\pi_{i_0})$ , maka  $\pi_{i_0} = \pi_i$

dimana  $\pi_{i_0}, \pi_i, \pi_{i_{-1}}$  adalah permutasi *job*,  $u$  adalah *job* yang dipilih secara acak, dan  $v$  adalah 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 2.5.



Gambar 2.5 Skema algoritma *Differential Evolution Plus*

## 2.6 Kriteria Kekonvergenan

Menurut Sivanandam dan 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* atau 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 tugas akhir ini merupakan data sekunder dari waktu pemrosesan tiap pekerjaan pada tiap mesin untuk produksi sepatu pada UD. Duta Kulit yang terletak di Singosari Kabupaten Malang. Data ini diambil dari tugas akhir Laila (2014) yang berupa data waktu proses pembuatan delapan macam model sepatu pada tiap mesin dalam satuan waktu yaitu menit. Proses produksi dilakukan dengan menggunakan delapan buah mesin. Data tersebut disajikan dalam Tabel 3.1.

Tabel 3.1 Data waktu proses pembuatan sepatu (menit)

Mesin	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$
<i>Job</i>								
$J_1$	72	36	48	100	48	120	24	54
$J_2$	108	72	120	108	48	120	36	54
$J_3$	84	48	72	102	48	120	30	54
$J_4$	120	60	168	156	66	120	48	54
$J_5$	138	74	180	162	66	120	48	54
$J_6$	74	42	78	124	42	120	24	54
$J_7$	84	44	96	150	48	120	42	54
$J_8$	90	54	120	116	54	120	30	54

$J_1$ = model sepatu casual;  $J_2$ = model sepatu pantofel A;  $J_3$ = model sepatu pantofel B;  $J_4$ = model sepatu pantofel C;  $J_5$ = model sepatu safety;  $J_6$ = model sepatu ceko;  $J_7$ = model sepatu fram;  $J_8$ = model sepatu mogasen;  $M_1$ = mesin pola;  $M_2$ = mesin seset;  $M_3$ = mesin jahit;  $M_4$ = mesin cetak;  $M_5$ = mesin gerinda;  $M_6$ = mesin perekatan;  $M_7$ = mesin press;  $M_8$ = mesin finishing.

Sumber: Laila (2014).

#### 3.2 Langkah-langkah Penelitian

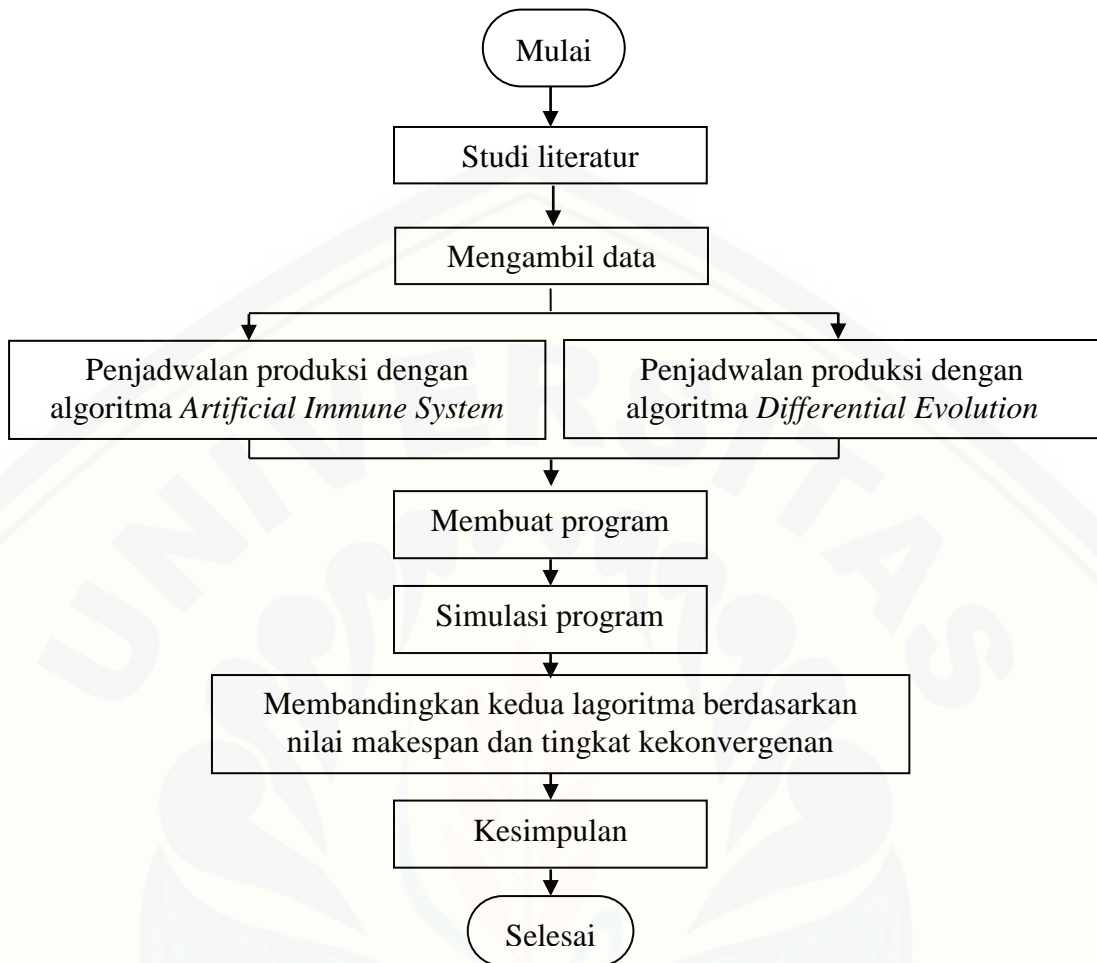
Dalam melakukan penelitian, maka terlebih dahulu dibuat langkah-langkah sistematis yang menjadi proses pengerjaan sehingga penelitian dapat dijalankan secara lebih terencana dan terstruktur. Langkah-langkah penelitian

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 *Artificial Immune System* dan algoritma *Differential Evolution* yang didapat dari buku-buku referensi, artikel, blog, 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 produksi sepatu yang terdapat pada tugas akhir Laila (2014);
- c. Menjadwalkan produksi *flowshop* menggunakan algoritma *Artificial Immune System* dan algoritma *Differential Evolution* 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* dan tingkat kecepatan kekonvergenan. Nilai *makespan* 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 diberikan 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 4. HASIL DAN PEMBAHASAN

### 4.1 Perhitungan Manual

Untuk mengetahui penerapan algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus* dalam menyelesaikan masalah penjadwalan *flowshop*, berikut akan diberikan contoh perhitungan manual menggunakan data contoh berukuran kecil. Data tersebut berupa data waktu proses produksi lima *job* pada tiga mesin yang dapat dilihat pada Tabel 4.1. Perhitungan secara manual menggunakan data kecil ini bertujuan untuk mempermudah pembaca dalam memahami langkah kerja kedua algoritma.

Tabel 4.1 Data contoh

Mesin <i>Job</i>	Mesin 1	Mesin 2	Mesin 3
<i>Job 1</i>	80	150	100
<i>Job 2</i>	50	50	70
<i>Job 3</i>	60	75	80
<i>Job 4</i>	70	100	50
<i>Job 5</i>	75	60	90

Data waktu proses produksi lima *job* dan tiga mesin dalam satuan menit.

#### 4.1.1 Perhitungan Manual dengan Algoritma *Artificial Immune System*

Pada subbab 2.4 telah dijelaskan langkah-langkah penjadwalan dengan algoritma *Artificial Immune System*, berikut ini dilakukan perhitungan manual dalam satu iterasi.

##### a. Menentukan parameter

Langkah awal yang dilakukan adalah menentukan parameter awal terlebih dahulu yaitu jumlah generasi ( $G$ ) = 1, dan ukuran populasi antibodi ( $N$ ) = 3.

##### b. Membangkitkan solusi awal

Pada langkah ini, algoritma *Artificial Immune System* membutuhkan solusi awal dalam menyelesaikan penjadwalan *flowshop* yang berupa urutan jadwal

*job*. Urutan jadwal *job* diperoleh dengan membangkitkan secara random sebanyak parameter ukuran populasi antibodi ( $N$ ) yang telah ditentukan sebelumnya. Misalkan urutan jadwal yang dibangkitkan secara random yaitu

Antibodi 1  $J_2 - J_5 - J_3 - J_1 - J_4$

Antibodi 2  $J_4 - J_1 - J_5 - J_2 - J_3$

Antibodi 3  $J_5 - J_2 - J_1 - J_4 - J_3$

Urutan jadwal tersebut selanjutnya dihitung nilai *makespan* seperti ditunjukkan pada Tabel 4.2, Tabel 4.3, dan Tabel 4.4.

Tabel 4.2 *Makespan*  $J_2 - J_5 - J_3 - J_1 - J_4$

Mesin	M <sub>1</sub>		M <sub>2</sub>		M <sub>3</sub>	
	S	E	S	E	S	E
<i>J</i> <sub>2</sub>	0	50	50	100	100	170
<i>J</i> <sub>5</sub>	50	125	125	185	185	275
<i>J</i> <sub>3</sub>	125	185	185	260	275	355
<i>J</i> <sub>1</sub>	185	265	265	415	415	515
<i>J</i> <sub>4</sub>	265	335	415	515	515	565

S = *Start* atau mulai; E = *End* atau berhenti

Tabel 4.3 *Makespan*  $J_4 - J_1 - J_5 - J_2 - J_3$

Mesin	M <sub>1</sub>		M <sub>2</sub>		M <sub>3</sub>	
	S	E	S	E	S	E
<i>J</i> <sub>4</sub>	0	70	70	170	170	220
<i>J</i> <sub>1</sub>	70	150	170	320	320	420
<i>J</i> <sub>5</sub>	150	225	320	380	420	510
<i>J</i> <sub>2</sub>	225	275	380	430	510	580
<i>J</i> <sub>3</sub>	275	335	430	505	580	660

S = *Start* atau mulai; E = *End* atau berhenti

Tabel 4.4 *Makespan*  $J_5 - J_2 - J_1 - J_4 - J_3$

Mesin	M <sub>1</sub>		M <sub>2</sub>		M <sub>3</sub>	
	S	E	S	E	S	E
<i>J</i> <sub>5</sub>	0	75	75	135	135	225
<i>J</i> <sub>2</sub>	75	125	135	185	225	295
<i>J</i> <sub>1</sub>	125	205	205	355	355	455
<i>J</i> <sub>4</sub>	205	275	355	455	455	505
<i>J</i> <sub>3</sub>	275	335	455	530	530	610

S = *Start* atau mulai; E = *End* atau berhenti

Berdasarkan ketiga tabel tersebut, dapat diketahui bahwa nilai *makespan* solusi awal dari yang terkecil hingga terbesar adalah 565, 610, dan 660 menit. Selanjutnya menghitung *Fitness Value*, Nilai Probabilitas dan Interval Antibodi. Perhitungan *fitness value* menggunakan persamaan 2.2, untuk perhitungan nilai probabilitas menggunakan persamaan 2.3, sedangkan penentuan interval antibodi dengan menjadikan nilai probabilitas kumulatif menjadi patokan akhir nilai interval yang ada. Seluruh perhitungan dapat dilihat pada Tabel 4.5.

Tabel 4.5 Perhitungan *fitness value*, nilai probabilitas dan interval

Antibodi	Urutan Job	Makespan	Fitness Value	Probabilitas	Probabilitas Kumulatif	Interval
Antibodi 1	$J_2 - J_5 - J_3 - J_1 - J_4$	565	96	0,64864	0,64864	0 – 0,64864
Antibodi 3	$J_5 - J_2 - J_1 - J_4 - J_3$	610	51	0,34459	0,99323	0,64865 – 0,99323
Antibodi 2	$J_4 - J_1 - J_5 - J_2 - J_3$	660	1	0,00676	1	0,99324 – 1

Berdasarkan Tabel 4.5 tersebut, dapat diketahui bahwa antibodi 1 dengan urutan job  $J_2 - J_5 - J_3 - J_1 - J_4$  mempunyai nilai *makespan* terkecil yaitu 565 menit. Solusi ini yang akan disimpan sebagai solusi terbaik untuk tahap awal.

c. *Cloning*

Pada tahap *cloning* terlebih dahulu dibangkitkan bilangan acak antara 0 sampai 1 sebanyak parameter populasi antibodi ( $N$ ) yang telah ditentukan sebelumnya. Misalkan bilangan acak yang dibangkitkan adalah sebagai berikut ini.

0,996

0,587

0,881

Selanjutnya *cloning* setiap antibodi sesuai bilangan acak yang dibangkitkan seperti pada tabel 4.6.

Tabel 4.6 Hasil *cloning*

Bilangan Random	Clone	Urutan Job	Makespan
0,996	Antibodi 2	$J_4 - J_1 - J_5 - J_2 - J_3$	660
0,587	Antibodi 1	$J_2 - J_5 - J_3 - J_1 - J_4$	565
0,881	Antibodi 3	$J_5 - J_2 - J_1 - J_4 - J_3$	610

## d. Mutasi Kebalikan

Mutasi kebalikan merupakan penukaran dua *job* yang dibangkitkan secara random. Misalkan dua *job* tersebut  $i$  dan  $j$ , dengan  $i = 2$  (urutan *job* ke-2) dan  $j = 4$  (urutan *job* ke-4). Kemudian membalikkan antara posisi  $i$  dan  $j$ . Semua *cloning* dilakukan mutasi kebalikan untuk mendapatkan antibodi baru seperti pada Tabel 4.7. Selanjutnya bandingkan nilai *makespan* antara solusi awal dan solusi baru hasil mutasi kebalikan. Nilai *makespan* yang dihasilkan dari urutan hasil mutasi kurang (lebih kecil) dari solusi awal, maka perlu dilakukan mutasi kembali yaitu mutasi penukaran pasangan.

Tabel 4.7 Urutan *job* mutasi kebalikan

Bilangan Random	Clone	Urutan Job	Makespan	New String mutasi kebalikan	Makespan	Antibodi Terpilih
0,996	Antibodi 2	$J_4 - J_1 - J_5 - J_2 - J_3$	660	$J_4 - J_2 - J_5 - J_1 - J_3$	610	$J_4 - J_2 - J_5 - J_1 - J_3$
0,587	Antibodi 1	$J_2 - J_5 - J_3 - J_1 - J_4$	565	$J_2 - J_1 - J_3 - J_5 - J_4$	600	$J_2 - J_5 - J_3 - J_1 - J_4$
0,881	Antibodi 3	$J_5 - J_2 - J_1 - J_4 - J_3$	610	$J_5 - J_4 - J_1 - J_2 - J_3$	645	$J_5 - J_2 - J_1 - J_4 - J_3$

## e. Mutasi Penukaran Pasangan

Berdasarkan Tabel 4.7 urutan *job* antibodi 2 hasil mutasi kebalikan akan dimutasi kembali. Pada mutasi penukaran pasangan dipilih dua posisi *job* secara random yaitu  $i$  dan  $j$  dari antibodi 2 hasil mutasi kebalikan kemudian menukar kedua posisi tersebut.



Tabel 4.8 Urutan *job* mutasi penukaran pasangan

Antibodi Terpilih	Makespan	<i>i</i>	<i>j</i>	New String mutasi penukaran pasangan	Makespan
$J_4 - J_2 - J_5 - J_1 - J_3$	610	2	5	$J_4 - J_3 - J_5 - J_1 - J_2$	625
$J_2 - J_5 - J_3 - J_1 - J_4$	565	-	-	-	-
$J_5 - J_2 - J_1 - J_4 - J_3$	610	-	-	-	-

Selanjutnya bandingkan nilai *makespan* antara solusi hasil mutasi kebalikan dan solusi baru hasil mutasi penukaran pasangan. Nilai *makespan* yang dihasilkan dari urutan hasil mutasi penukaran pasangan kurang (lebih kecil) dari solusi mutasi kebalikan, maka solusi hasil mutasi penukaran pasangan menjadi antibodi akhir sedangkan jika tidak, maka solusi mutasi kebalikan menjadi antibodi akhir. Urutan *job*  $J_2 - J_5 - J_3 - J_1 - J_4$  mempunyai hasil *makespan* terkecil yaitu 565 menit.

Antibodi baru yang terpilih untuk generasi 1 adalah

$$J_4 - J_2 - J_5 - J_1 - J_3$$

$$J_2 - J_5 - J_3 - J_1 - J_4$$

$$J_5 - J_2 - J_1 - J_4 - J_3$$

f. Mengecek iterasi maksimal

Perhitungan untuk generasi 1 selesai. Urutan *job*  $J_2 - J_5 - J_3 - J_1 - J_4$  mempunyai hasil *makespan* terkecil yaitu 565 menit. Untuk generasi 2 perhitungan kembali dari langkah b dengan antibodi awal merupakan antibodi baru yang terpilih pada generasi 1. Perhitungan dilakukan hingga iterasi maksimal terpenuhi dan didapatkan urutan jadwal optimal dengan nilai *makespan* minimum.

#### 4.1.2 Penyelesaian Manual dengan Algoritma *Differential Evolution Plus*

Pengolahan data contoh berikut ini akan dilakukan secara manual dalam satu iterasi menggunakan algoritma *Differential Evolution Plus*. Pada perhitungan sebelum iterasi dimulai, dilakukan input parameter-parameter terlebih dahulu. Misalnya populasi yang dibangkitkan = 3, batas atas ( $Ub$ ) = 1, batas bawah ( $Lb$ ) = 0, nilai terkecil probabilitas *crossover* ( $Cr_{min}$ ) = 0,3, nilai terbesar probabilitas *crossover* ( $Cr_{max}$ ) = 0,9, nilai minimum parameter mutasi ( $F_{min}$ ) = 0,5, dan

jumlah iterasi maksimum ( $MAXGEN$ ) = 100. Langkah penyelesaiannya adalah sebagai berikut ini.

a. Inisialisasi populasi

Misal dibangkitkan 3 populasi dengan tiap populasi merepresentasikan sejumlah *job*. Untuk generasi awal nilai individu pada tiap populasi ( $x_i$ ) dibangkitkan menggunakan persamaan (2.4).

$$x_i = \begin{bmatrix} 0,571 & 0,911 & 0,321 & 0,283 & 0,657 \\ 0,925 & 0,258 & 0,478 & 0,718 & 0,412 \\ 0,525 & 0,012 & 0,811 & 0,137 & 0,021 \end{bmatrix}$$

Masing-masing populasi diberi indeks sesuai dengan jumlah *job*, sehingga Nilai  $x_i$  memiliki indeks sebagai berikut:

$$x_i = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

Nilai individu pada tiap populasi diurutkan menggunakan prosedur SPV (*Smallest Position Value*) untuk mendapatkan solusi awal. Sehingga solusi awal pemutasi *job* sebagai berikut:

$$x_i = \begin{bmatrix} 4 & 3 & 1 & 2 & 5 \\ 2 & 5 & 3 & 4 & 1 \\ 2 & 5 & 4 & 1 & 3 \end{bmatrix}$$

b. Menghitung fungsi objektif

Nilai fungsi objektif disini yaitu *makespan*, dimana pada tiap populasi dihitung untuk menentukan  $f_{min}$  dan  $f_{max}$  yang merupakan nilai minimum dan maksimum fungsi objektif.

Untuk urutan *job* : [4 3 1 2 5], *makespan* = 655 menit  $\rightarrow f_{max}$

Untuk urutan *job* : [2 5 3 4 1], *makespan* = 610 menit  $\rightarrow f_{min}$

Untuk urutan *job* : [2 5 4 1 3], *makespan* = 625 menit

Untuk iterasi,  $G = 1$

c. *Crossover*

Tahap *crossover* dilakukan untuk menentukan populasi akan dimutasi atau tidak. Sebelum tahap *crossover*, dihitung nilai parameter *crossover* ( $Cr$ ) menggunakan persamaan (2.6) sehingga diperoleh hasil:

$$\begin{aligned}
 Cr &= Cr_{min} + G \cdot \frac{Cr_{max} - Cr_{min}}{MAXGEN} \\
 &= 0,5 + 1 \cdot \left( \frac{0,9 - 0,5}{100} \right) \\
 &= 0,5 + (1 \cdot 0,004) \\
 &= 0,504
 \end{aligned}$$

Selanjutnya dibangkitkan bilangan random pada tiap populasi. Apabila bilangan random  $\leq Cr$  maka individu pada populasi ke  $- i$  akan dimutasi, apabila bilangan random  $> Cr$  maka individu pada populasi ke  $- i$  tidak dimutasi. Misal bilangan random yang dibangkitkan adalah:

$$x_1 = 0,175 < Cr$$

$$x_2 = 0,258 < Cr$$

$$x_3 = 0,716 > Cr$$

maka yang akan dimutasi adalah individu pada populasi 1 dan 2 ( $x_1$  dan  $x_2$ ).

#### d. Mutasi

Sebelum tahap mutasi, terlebih dahulu dihitung nilai parameter mutasi ( $F$ ) menggunakan persamaan (2.8).

Karena  $\left| \frac{f_{max}}{f_{min}} \right| = \left| \frac{616}{597} \right| = 1,032 > 1$ , maka nilai  $F$  yang didapat = 0,5.

Selanjutnya dilakukan mutasi pada individu  $x_1$  dan  $x_2$  dengan persamaan (2.7). Sehingga diperoleh hasil mutasi pada populasi ke- $i$  ( $v_i$ ) berikut ini.

$$v_i = \begin{bmatrix} 0,031 & 0,344 & 1,109 & -0,676 & 0,355 \\ 0,965 & 1,134 & 0,156 & 0,393 & -0,263 \\ 0,525 & 0,012 & 0,811 & 0,137 & 0,021 \end{bmatrix}$$

Hasil mutasi dicek apakah diluar interval batas atas dan batas bawah. Jika hasil mutasi  $>$  batas atas maka diganti dengan 1, jika hasil mutasi  $<$  batas bawah maka diganti dengan 0, sedangkan jika hasil mutasi bernilai antara 0-1 maka nilainya tetap. Sehingga didapatkan hasil:

$$\begin{bmatrix} 0,031 & 0,344 & 1 & 0 & 0,355 \\ 0,965 & 1 & 0,156 & 0,393 & 0 \\ 0,525 & 0,012 & 0,811 & 0,137 & 0,021 \end{bmatrix}$$

Hasil perhitungan diatas diurutkan menggunakan prosedur SPV, sehingga

$$\text{didapat permutasi } job: \begin{bmatrix} 4 & 1 & 2 & 5 & 3 \\ 5 & 3 & 4 & 1 & 2 \\ 2 & 5 & 4 & 1 & 3 \end{bmatrix}$$

Untuk urutan  $job : [4 \ 1 \ 2 \ 5 \ 3]$ ,  $makespan = 660$  menit  $\rightarrow f(v_i) = 660$

Untuk urutan  $job : [5 \ 3 \ 4 \ 1 \ 2]$ ,  $makespan = 630$  menit 630

Untuk urutan  $job : [2 \ 5 \ 4 \ 1 \ 3]$ ,  $makespan = 625$  menit 625

e. Seleksi

Pada tahap seleksi akan dibandingkan nilai  $makespan$  hasil mutasi  $f(v_i)$  dan nilai  $makespan$  awal  $f(x_i)$ . Jika  $f(v_i) < f(x_i)$  maka  $x_i = v_i$ , jika tidak maka  $x_i = x_i$ . Sehingga didapatkan hasil seleksi sebagai berikut ini.

$$f(x_i) = 655$$

$$610$$

$$625$$

$$x_i = \begin{bmatrix} 0,571 & 0,911 & 0,321 & 0,283 & 0,657 \\ 0,925 & 0,258 & 0,478 & 0,718 & 0,412 \\ 0,525 & 0,012 & 0,811 & 0,137 & 0,021 \end{bmatrix}$$

$$\text{Dengan permutasi } job: \begin{bmatrix} 4 & 3 & 1 & 2 & 5 \\ 2 & 5 & 3 & 4 & 1 \\ 2 & 5 & 4 & 1 & 3 \end{bmatrix}$$

f. *Local Search*

Hasil dari seleksi, pada langkah *local search* ini akan dikenai prosedur *insert-based local search* yaitu dengan menyisipkan  $job$  ke- $u$  pada posisi  $v$  dimana  $u \neq v$ . Pada tahap seleksi, telah diperoleh urutan permutasi  $job$  beserta nilai objektifnya, yang selanjutnya disebut  $f(\pi_{i_0})$  yang merupakan nilai  $makespan$  permutasi  $job$  untuk populasi ke -  $i$  tahap *local search* ke -  $0$ .

$$\pi_{i_0} = \begin{bmatrix} 4 & 3 & 1 & 2 & 5 \\ 2 & 5 & 3 & 4 & 1 \\ 2 & 5 & 4 & 1 & 3 \end{bmatrix} \rightarrow \begin{array}{l} f(\pi_{i_0}) = 655 \\ 610 \\ 625 \end{array}$$

Secara acak dipilih  $job$  ke- $u$  dan posisi  $v$  dengan  $u \neq v$ , kemudian sisipkan  $job$   $u$  ke posisi  $v$  pada setiap populasi serta hitung nilai  $makespan$  yang selanjutnya disebut dengan  $f(\pi_i)$  yaitu nilai  $makespan$  permutasi  $job$  untuk populasi ke -  $i$ .

$$\text{Urutan } job : [3 \ 4 \ 1 \ 2 \ 5] \rightarrow f(\pi_i) = 645$$

$$\text{Urutan } job : [4 \ 1 \ 2 \ 5 \ 3] \quad 610$$

$$\text{Urutan } job : [4 \ 1 \ 2 \ 5 \ 3] \quad 640$$

Prosedur *insert-based local search* dengan proses penyisipan *job u* ke posisi *v* dilakukan *looping* beberapa kali dimana kriteria pemberhentian *looping*  $< n(n - 1)$  dengan *n* adalah jumlah *job*. Untuk *looping* pertama didapatkan:

$$\text{Urutan } job : [3 \ 1 \ 2 \ 5 \ 4] \longrightarrow f(\pi_{i-1}) = 600$$

$$\text{Urutan } job : [4 \ 5 \ 3 \ 2 \ 1] \quad 605$$

$$\text{Urutan } job : [3 \ 2 \ 4 \ 1 \ 5] \quad 625$$

Setelah diperoleh  $f(\pi_i)$  dan  $f(\pi_{i-1})$ , kemudian bandingkan nilai keduanya.

Urutan *job* dengan nilai *makespan* lebih kecil akan terpilih sehingga hasil perbandingan diperoleh sebagai berikut:

$$\text{Urutan } job : [3 \ 1 \ 2 \ 5 \ 4] \longrightarrow f(\text{hasil } 1) = 600$$

$$\text{Urutan } job : [4 \ 5 \ 3 \ 2 \ 1] \quad 605$$

$$\text{Urutan } job : [3 \ 2 \ 4 \ 1 \ 5] \quad 625$$

Setelah *looping* dilakukan hingga kriteria pemberhentian, maka diperoleh urutan *job* beserta nilai *makespan* yang disebut  $f(\pi_i)$  dari hasil *looping*.

$$\text{Urutan } job : [3 \ 5 \ 2 \ 1 \ 4] \longrightarrow f(\pi_i) = 565$$

$$\text{Urutan } job : [3 \ 4 \ 5 \ 2 \ 1] \quad 595$$

$$\text{Urutan } job : [5 \ 3 \ 2 \ 1 \ 4] \quad 565$$

Selanjutnya bandingkan  $f(\pi_i)$  dari hasil *looping* dengan  $f(\pi_{i-1})$  pada tahap seleksi sehingga akan diperoleh urutan *job*, nilai *makespan* dan nilai  $x_i$  seperti berikut ini:

$$\text{Urutan } job : [3 \ 5 \ 2 \ 1 \ 4] \longrightarrow f(x_i) = 565$$

$$\text{Urutan } job : [3 \ 4 \ 5 \ 2 \ 1] \quad 595$$

$$\text{Urutan } job : [5 \ 3 \ 2 \ 1 \ 4] \quad 565$$

$$x_i = \begin{bmatrix} 0,571 & 0,911 & 0,321 & 0,283 & 0,657 \\ 0,925 & 0,258 & 0,478 & 0,718 & 0,412 \\ 0,525 & 0,012 & 0,811 & 0,137 & 0,021 \end{bmatrix}$$

g. Mengecek iterasi maksimal

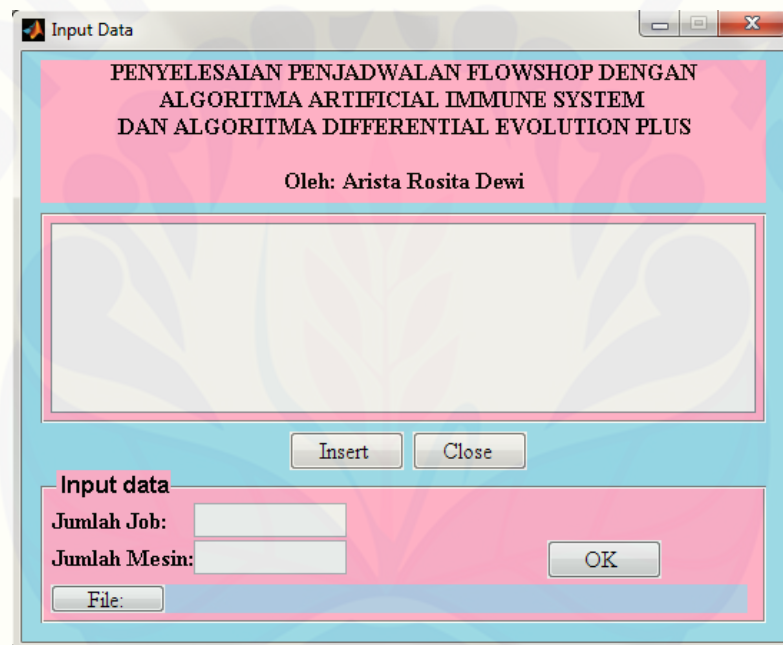
Untuk iterasi selanjutnya, kembali pada langkah *crossover*. Proses dalam algoritma *Differential Evolution Plus* ini dilakukan hingga iterasi maksimal



terpenuhi sehingga akan diperoleh urutan *job* yang optimal dengan nilai *makespan* yang minimum.

#### 4.2 Program

Pada tugas akhir ini, untuk memberikan kemudahan dalam perhitungan maka dibuat suatu program yang cara kerjanya berdasarkan perhitungan penjadwalan *flowshop* dengan algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus*. Gambar 4.1 merupakan tampilan awal dari program yang telah dibuat.



Gambar 4.1 Tampilan awal program aplikasi

Berdasarkan Gambar 4.1 terdapat beberapa langkah menjalankan program. Berikut penjelasan dari tiap langkah dalam menjalankan program aplikasi penjadwalan *flowshop* ini.

a. *Input* Ukuran Masalah

Hal yang pertama kali dilakukan yaitu meng-*input* terlebih dahulu jumlah *job* dan jumlah mesin yang akan diproses pada kolom “Input data”. Dalam tugas akhir ini, ukuran masalah penjadwalan *flowshop* yang akan diselesaikan adalah  $8 \times 8$  yang artinya terdiri dari delapan *job* yang akan diproses pada delapan mesin,

sehingga jadwal yang terbentuk sebanyak 64 digit waktu dalam satuan menit. Setelah meng-*input* jumlah *job* dan jumlah mesin, selanjutnya klik tombol “OK” sehingga akan muncul tabel input data seperti pada Gambar 4.2.

Gambar 4.2 Tampilan *input* data

Pengguna bisa secara manual meng-*input* data waktu proses setiap *job* pada setiap mesin sesuai dengan data yang ada (Tabel 3.1). Pengguna juga bisa menggunakan tombol “File” untuk membuka *file* “Data waktu pembuatan sepatu.txt” yang sebelumnya sudah disimpan dalam satu *folder* beserta dengan program, sehingga akan muncul tampilan seperti Gambar 4.3. Selanjutnya klik tombol “Insert” untuk mulai memproses data.

	Mesin 1	Mesin 2	Mesin 3	Mesin 4	Mesin 5	Mesin 6	Mesin 7	Mesin 8
Job 1	72	36	48	100	48	120	24	54
Job 2	108	72	120	108	48	120	36	54
Job 3	84	48	72	102	48	120	30	54
Job 4	120	60	168	156	66	120	48	54
Job 5	138	74	180	162	66	120	48	54
Job 6	74	42	78	124	42	120	24	54
Job 7	84	44	96	150	48	120	42	54
Job 8	90	54	120	116	54	120	30	54

Gambar 4.3 Tampilan *input* data pembuatan sepatu

### b. Proses Data

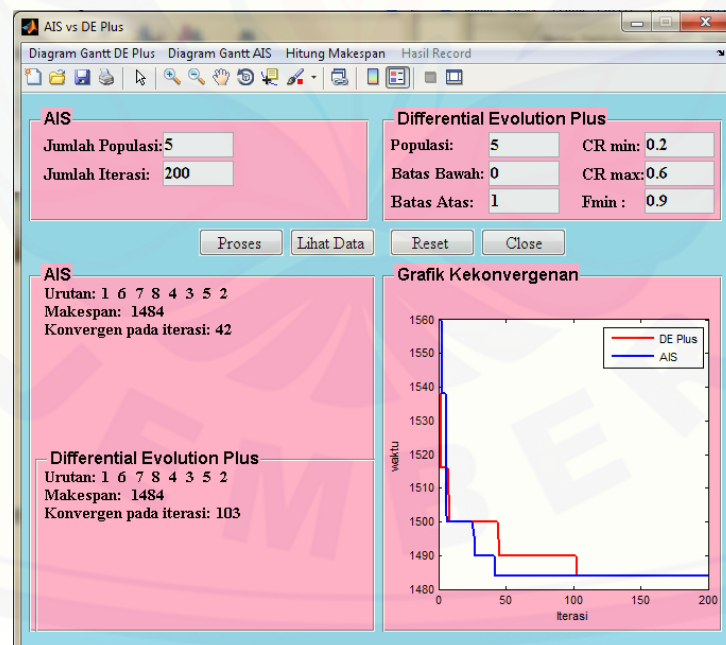
Data yang sebelumnya telah di *input*, selanjutnya akan diproses. Untuk memproses data dengan kedua algoritma, pada Gambar 4.4 dapat dilihat beberapa menu untuk memproses data. Menu “Proses” digunakan untuk mengetahui hasil dari kedua algoritma. Menu “Lihat Data” digunakan untuk menampilkan data waktu proses setiap *job* pada setiap mesin yang telah di *inputkan* di awal.



Gambar 4.4 Tampilan menu proses data

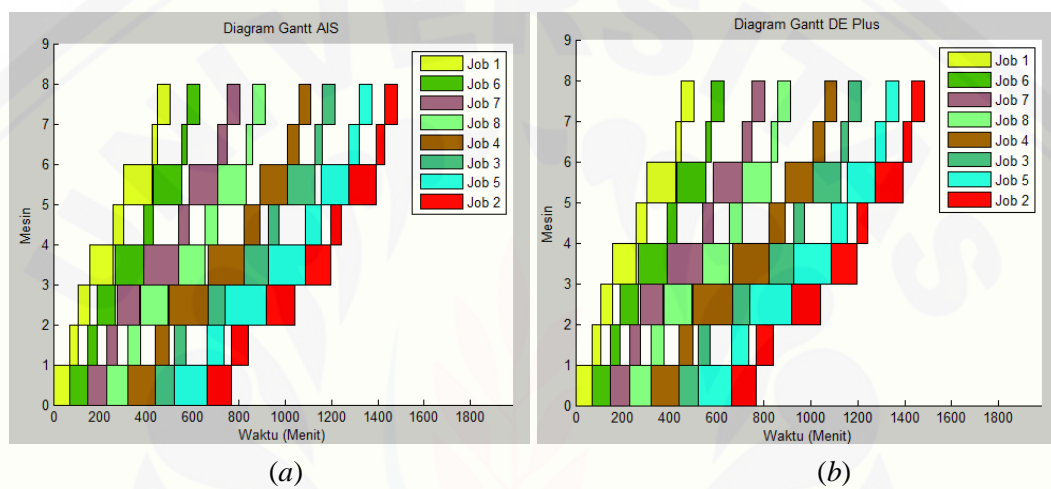
### c. Output Program

Hasil akhir yang ditampilkan dalam perhitungan menggunakan program yang telah dibuat adalah jadwal urutan *job* dengan nilai *makespan*, grafik kekonvergenan, dan diagram *gantt chart* yang dihasilkan dari masing-masing algoritma. Gambar 4.5 menunjukkan hasil dari algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus*.



Gambar 4.5 Tampilan hasil *output* urutan *job*, *makespan*, dan grafik kekonvergenan

Untuk melihat *gantt chart* dengan cara klik menu “Diagram Gantt DE Plus” untuk algoritma *Differential Evolution Plus*, dan “Diagram Gantt AIS” untuk algoritma *Artificial Immune System*. Gambar 4.6 (a) dan Gambar 4.6 (b) merupakan *gantt chart* untuk algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus*. Sumbu horisontal pada *gantt chart* menunjukkan urutan mesin dari masing-masing algoritma, sedangkan sumbu vertikal menunjukkan pergerakan waktu.



Gambar 4.6 Tampilan *Gantt Chart*

### 4.3 Penyelesaian Penjadwalan *Flowshop* Menggunakan Program

Simulasi program dilakukan beberapa kali pengujian dengan mengubah nilai parameter kedua algoritma.

#### a. Pengujian nilai $Cr_{min}$ , $Cr_{max}$ , dan $F_{min}$

Pada algoritma *Differential Evolution Plus* parameter yang digunakan yaitu populasi = 5, batas bawah = 0, batas atas = 1, untuk nilai  $Cr_{min}$ ,  $Cr_{max}$ , dan  $F_{min}$  diubah-ubah. Nilai  $Cr_{min}$  yang digunakan adalah 0,2 dan 0,5. Nilai  $Cr_{max}$  yang digunakan 0,6 dan 0,9. Dan nilai  $F_{min}$  yang digunakan adalah 0,3, 0,6, dan 0,9. Pengujian perubahan parameter tersebut semuanya dilakukan sebanyak sepuluh kali *running*. Hasil dari pengujian perubahan parameter tersebut dapat dilihat pada lampiran A. Tabel 4.9 menunjukkan rata-rata nilai makespan untuk  $Cr_{min}$ ,  $Cr_{max}$ , dan  $F_{min}$  berbeda.

Tabel 4.9 *Makespan* rata-rata untuk  $Cr_{min}$ ,  $Cr_{max}$ , dan  $F_{min}$  berbeda

$Cr_{min}$	$Cr_{max}$	$F_{min}$	<i>Makespan</i> rata-rata DE Plus
0,2	0,6	0,3	1485,2
0,2	0,6	0,6	1484,6
0,2	0,6	0,9	1484
0,5	0,9	0,3	1484,6
0,5	0,9	0,6	1485,2
0,5	0,9	0,9	1485,2

*Makespan* dalam satuan menit.

Berdasarkan Tabel 4.9 dan Lampiran A dapat diketahui bahwa nilai  $Cr_{min}=0,2$ ,  $Cr_{max}=0,6$ , dan  $F_{min}=0,9$  menunjukkan nilai *makespan* yang lebih kecil dan stabil daripada yang lain. Berdasarkan hal ini, untuk pengujian selanjutnya akan digunakan nilai  $Cr_{min}=0,2$ ,  $Cr_{max}=0,6$ , dan  $F_{min}=0,9$ .

b. Pengujian dengan iterasi sama dan jumlah populasi sama

Pada algoritma *Artificial Immune System* parameter yang digunakan terdiri dari jumlah populasi = 5, dan jumlah generasi = 200. Sedangkan pada algoritma *Differential Evolution Plus* parameter yang digunakan yaitu populasi = 5, batas bawah = 0, batas atas = 1, untuk nilai  $Cr_{min}=0,2$ ,  $Cr_{max}=0,6$ , dan  $F_{min}=0,9$  dan iterasi = 200.

Untuk mengetahui apakah solusi yang dihasilkan akan selalu sama pada setiap kali pengujian, maka dilakukan pengujian sebanyak sepuluh kali dengan jumlah iterasi yang sama yaitu 200 iterasi. Hal ini bertujuan untuk menguji seberapa baik solusi yang diperoleh masing-masing algoritma jika diproses dalam jumlah iterasi yang sama. Hasil *output* sepuluh kali pengujian ini ditunjukkan dalam Lampiran B. Terlihat bahwa urutan jadwal yang dihasilkan algoritma *Artificial Immune System* berbeda dalam setiap pengujian, namun nilai *makespan* terlihat sama pada beberapa pengujian. Urutan jadwal dengan *makespan* terbaik diperoleh saat pengujian ke-1, 4, dan ke-5 yaitu sebesar 1484 menit dengan urutan jadwal  $J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$ . Sedangkan untuk algoritma *Differential Evolution Plus* urutan jadwal dan nilai *makespan* dalam setiap pengujian adalah sama dengan urutan jadwal  $J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 -$



$J_2$  dan nilai makespan 1484 menit. Keadaan tersebut menunjukkan bahwa algoritma *Artificial Immune System* memberikan hasil yang berubah-ubah untuk nilai *makespan* pada setiap pengujian, sehingga nilai *makespan* rata-rata yang didapat adalah 1494,6 menit sedangkan pada algoritma *Differential Evolution Plus* hasil *makespan* selalu menunjukkan nilai yang sama pada setiap pengujian, sehingga nilai *makespan* rata-rata yang didapat adalah 1484 menit. Dapat diketahui bahwa hasil optimal urutan jadwal dan nilai *makespan* yang dihasilkan kedua algoritma adalah sama. Hal ini berarti bahwa penggunaan algoritma *Differential Evolution Plus* lebih efektif dibandingkan dengan algoritma *Artificial Immune System* jika diterapkan pada penjadwalan produksi sepatu.

Adanya perbedaan solusi pada setiap *output* yang dihasilkan oleh kedua algoritma dikarenakan kedua algoritma merupakan algoritma dimana proses dimulai dengan membangkitkan solusi awal kemudian proses pencarian solusi baru juga dibangkitkan secara acak sehingga tidak menjamin solusi yang dihasilkan selalu sama dengan optimum global. Selain itu permasalahan optimasi penjadwalan *flowshop* merupakan masalah dengan ruang solusi yang amat luas, maka solusi yang dihasilkan tidak harus paling optimum, tetapi kualitasnya mendekati optimum global dan bisa diterima.

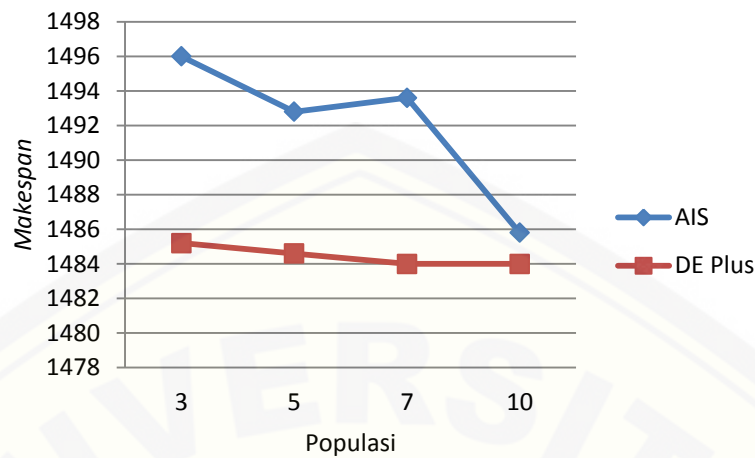
c. Pengujian dengan iterasi sama dan jumlah populasi berbeda

Pengujian dilakukan dengan merubah jumlah populasi yaitu 3, 5, 7 dan 10 dengan iterasi tetap yaitu 200. Masing-masing pengujian perubahan parameter dilakukan sebanyak sepuluh kali. Hasil dari pengujian perubahan parameter dapat dilihat pada Lampiran C. Berdasarkan lampiran tersebut dapat dibuat rata-rata nilai *makespan* dengan jumlah populasi berubah dan jumlah iterasi tetap seperti pada Tabel 4.10.

Tabel 4.10 *Makespan* rata-rata untuk jumlah populasi berbeda

Populasi	Generasi (iterasi)	<i>Makespan</i> rata-rata	
		AIS	DE Plus
3	200	1496	1485,2
5	200	1492,8	1484,6
7	200	1493,6	1484
10	200	1485,8	1484

*Makespan* dalam satuan menit.



Gambar 4.7 Grafik *makespan* rata-rata untuk jumlah populasi berbeda

Gambar 4.7 diatas menunjukkan bahwa dengan jumlah generasi (iterasi) tetap yaitu 200 tetapi ukuran populasi yang berbeda maka semakin tinggi ukuran populasi maka nilai *makespan* yang dihasilkan akan semakin kecil dan stabil pada kedua algoritma.

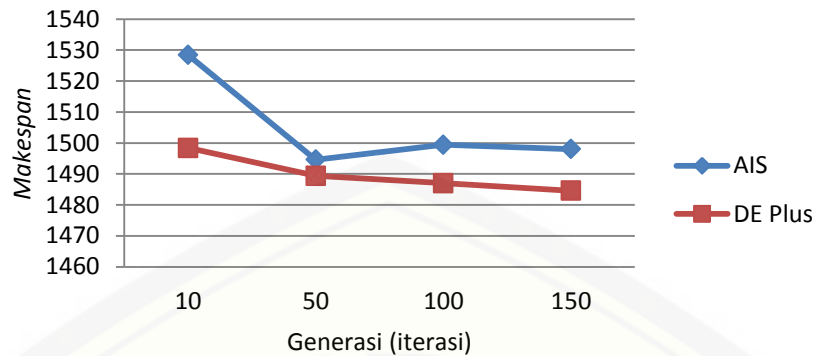
d. Pengujian dengan iterasi berbeda dan jumlah populasi sama

Pengujian dilakukan dengan merubah jumlah iterasi yaitu 10, 50, 100, dan 150 dengan jumlah populasi tetap yaitu 5. Masing-masing pengujian perubahan parameter dilakukan sebanyak sepuluh kali. Hasil dari pengujian perubahan parameter dapat dilihat pada Lampiran D. Berdasarkan lampiran tersebut dapat dibuat rata-rata nilai *makespan* dengan jumlah populasi tetap dan jumlah iterasi berubah seperti pada Tabel 4.11.

Tabel 4.11 *Makespan* rata-rata untuk jumlah generasi (iterasi) berbeda

Populasi	Generasi (iterasi)	<i>Makespan</i> rata-rata	
		AIS	DE Plus
5	10	1528,4	1498,4
5	50	1494,6	1489,4
5	100	1499,4	1487
5	150	1498	1484,6

*Makespan* dalam satuan menit.



Gambar 4.8 Grafik *makespan* rata-rata untuk jumlah generasi (iterasi) berbeda

Berdasarkan Gambar 4.8 semakin tinggi jumlah generasi (iterasi) dengan nilai ukuran populasi tetap maka nilai *makespan* rata-rata yang dihasilkan dari beberapa pengujian semakin kecil dan stabil pada kedua algoritma. Hal itu dikarenakan dengan semakin banyaknya generasi maka kemungkinan solusi jadwal yang dihasilkan pada proses kloning dan mutasi pada algoritma AIS akan semakin banyak serta pada proses *crossover*, mutasi dan *local search* pada algoritma DE Plus akan semakin banyak pula. Dengan semakin banyak solusi jadwal yang mungkin maka akan dihasilkan solusi jadwal yang optimal.

#### 4.4 Tingkat Konvergensi Algoritma

Untuk mengetahui tingkat kecepatan kekonvergenan dalam setiap iterasi pada masing-masing algoritma, dapat dilihat pada Tabel 4.12. pengujian dilakukan sebanyak sepuluh kali.

Tabel 4.12 Pengujian Konvergensi

Pengujian ke-	Iterasi Konvergen	
	Algoritma <i>Artificial Immune System</i>	Algoritma <i>Differential Evolution Plus</i>
1	42	103
2	19	51
3	4	59
4	128	52
5	167	21
6	4	134
7	7	171
8	17	168
9	7	196
10	24	45

Berdasarkan Tabel 4.12, dihitung rata-rata dari kedua algoritma untuk mengetahui pada rata-rata iterasi ke berapa kedua algoritma mulai konvergen. Dapat diketahui bahwa dalam sepuluh kali pengujian, algoritma *Artificial Immune System* memiliki rata-rata 41,9 sehingga dikatakan konvergen mulai iterasi ke-42 dan algoritma *Differential Evolution Plus* memiliki rata-rata 100 sehingga dikatakan konvergen mulai iterasi ke-100. Hal ini menunjukkan algoritma *Artificial Immune System* lebih cepat konvergen dari pada algoritma *Differential Evolution Plus*. Berdasarkan semua pengujian yang telah dilakukan seperti yang dapat dilihat pada semua lampiran juga menunjukkan bahwa algoritma *Artificial Immune System* lebih cepat konvergen dari pada algoritma *Differential Evolution Plus*. Sehingga berdasarkan tingkat kecepatan kekonvergenan yang dihasilkan pada semua pengujian yang telah dilakukan, menunjukkan bahwa algoritma *Artificial Immune System* lebih cepat konvergen dibanding algoritma *Differential Evolution Plus*. Namun kecepatan kekonvergenan tidak menjamin algoritma yang cepat konvergen adalah algoritma yang terbaik.

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang telah diuraikan pada bab 4, dapat diambil kesimpulan sebagai berikut ini:

- a. Penjadwalan produksi *flowshop* yang diterapkan untuk produksi sepatu menggunakan algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus* menghasilkan urutan jadwal dan nilai *makespan* optimal yang sama yaitu *makespan* sebesar 1484 menit dengan urutan jadwal model sepatu casual – model sepatu ceko – model sepatu fram – model sepatu mogasen – model sepatu pantofel C – model sepatu pantofel B – model sepatu safety – model sepatu pantofel A. Namun pada sepuluh kali pengujian dengan jumlah iterasi yang sama algoritma *Artificial Immune System* memberikan hasil yang berubah-ubah untuk nilai *makespan* pada setiap pengujian, sehingga nilai *makespan* rata-rata yang didapat adalah 1494,6 menit dengan urutan jadwal berbeda, sedangkan pada algoritma *Differential Evolution Plus* hasil *makespan* selalu menunjukkan nilai yang sama pada setiap pengujian dengan urutan jadwal yang sama pula, sehingga nilai *makespan* rata-rata yang didapat adalah 1484 menit. Hal ini berarti bahwa penggunaan algoritma *Differential Evolution Plus* lebih efektif dibandingkan dengan algoritma *Artificial Immune System* jika diterapkan pada penjadwalan produksi sepatu.
- b. Algoritma *Artificial Immune System* memiliki tingkat kecepatan kekonvergenan yang lebih baik dibanding dengan algoritma *Differential Evolution Plus* karena lebih cepat konvergen.

### 5.2 Saran

Penjadwalan *flowshop* menggunakan algoritma *Artificial Immune System* dan algoritma *Differential Evolution Plus* ini masih bisa dikembangkan, misalnya



membandingkan ataupun memodifikasi dengan algoritma lainnya. Selain itu, juga dapat diterapkan pada permasalahan lain selain penjadwalan *flowshop*.



**DAFTAR PUSTAKA**

- Ginting, R. 2009. *Penjadwalan Mesin*. Yogyakarta: Graha Ilmu.
- Ginting, R., Ginting, T.U. Hidayat S. 2006. Studi Aplikasi Metode *Artificial Immune System* Dalam Penjadwalan Flowshop. *Jurnal sistem teknik industry*. **7(2)**: 15-21.
- Karolina, V.E. 2013. “Usulan Perbaikan Penjadwalan Produksi Dengan Menggunakan Metode Artificial Immune System”. Tidak dipublikasikan. Skripsi. Bandung: Jurusan Teknik Industri Fakultas Teknik Universitas Kristen Maranatha.
- Laila, I. 2014. “Penerapan Algoritma Simulated Annealing dan Algoritma Tabu Search Pada Produksi Sepatu Dengan Penjadwalan Flowshop”. Tidak diterbitkan. Skripsi. Jember: FMIPA Universitas Jember.
- Nurdiansyah, R. 2011. *Pengembangan Algoritma Differential Evolution Untuk Penjadwalan Flowshop Multi Objektif Dengan Banyak Mesin*. Surabaya : Jurusan Teknik Industri Fakultas Teknik Industri Institut Teknologi Sepuluh Nopember.
- Perdana, S. B. 2011. “Aplikasi Algoritma Artificial Immune System (AIS) Pada Penjadwalan Jobshop Dalam Pembuatan Spring Bed”. Tidak dipublikasikan. Skripsi. Jember: Jurusan Matematika Fakultas MIPA Universitas Jember.
- Qian, B., Wang, L., Huang D., Wang, W.L., dan Wang, X. 2008. A Hybrid Differential Evolution Method for Permutation Flow-Shop Scheduling. *The International Journal of Advanced Manufacturing Technology*. **38**: 757-777.
- Sari, T.K. 2015. “Perbandingan Algoritma *Simulated Annealing* Dan Algoritma *Differential Evolution Plus* untuk Penjadwalan *Flowshop*”. Tidak dipublikasikan. Skripsi. Jember: Jurusan Matematika Fakultas MIPA Universitas Jember.
- Sivanandam, S.N. & Deepa, S.N. 2008. *Introduction to Genetic Algorithms*. India: PSG College of Technology Coimbatore.
- Wiratno, S.E., Nurdiansyah, R., Santosa, B. 2012. Algoritma Differential Evolution Untuk Penjadwalan Flow Shop Banyak Mesin Dengan Multi

Obyektif. *Jurnal Teknik Industri Institut Teknologi Sepuluh Nopember*.  
**13(1):** 1-6.



**LAMPIRAN A. PENGUJIAN ALGORITMA DIFFENTIAL EVOLUTION PLUS DENGAN  $Cr_{min}$ ,  $Cr_{max}$ , dan  $F_{min}$  BERUBAH**

**A.1 Tabel hasil sepuluh kali pengujian dengan  $Cr_{min} = 0,2$ ,  $Cr_{max} = 0,6$ , dan  $F_{min} = 0,3$**

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	93
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_2 - J_3 - J_5$	1490	107
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	65
4	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_5$	1490	10
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	100
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	110
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	185
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	45
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	184
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	52

**A.2 Tabel hasil sepuluh kali pengujian dengan  $Cr_{min} = 0,2$ ,  $Cr_{max} = 0,6$ , dan  $F_{min} = 0,6$**

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	41
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	49
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	92
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	9
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	136
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	58
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	76
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	117
9	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_5$	1490	20
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	145

**A.3 Tabel hasil sepuluh kali pengujian dengan  $Cr_{min} = 0,2$ ,  $Cr_{max} = 0,6$ , dan  $F_{min} = 0,9$**

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	2
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	74
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	31
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	80
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	132
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	60
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	32
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	79
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	143
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	91

**A.4 Tabel hasil sepuluh kali pengujian dengan  $Cr_{min} = 0,5$ ,  $Cr_{max} = 0,9$ , dan  $F_{min} = 0,3$**

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	77
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	89
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	67
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	35
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	186
6	$J_1 - J_3 - J_7 - J_8 - J_6 - J_4 - J_5 - J_2$	1490	63
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	101
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	102
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	68
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	20



**A.5 Tabel hasil sepuluh kali pengujian dengan  $Cr_{min} = 0,5$ ,  $Cr_{max} = 0,9$ , dan  $F_{min} = 0,6$**

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	30
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	178
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	4
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_2 - J_3 - J_5$	1490	30
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	75
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	31
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	65
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	87
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_2 - J_5$	1490	29
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	37

**A.6 Tabel hasil sepuluh kali pengujian dengan  $Cr_{min} = 0,5$ ,  $Cr_{max} = 0,9$ , dan  $F_{min} = 0,9$**

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	122
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	27
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	69
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	34
5	$J_1 - J_6 - J_3 - J_7 - J_8 - J_4 - J_5 - J_2$	1490	48
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	83
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	158
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	168
9	$J_1 - J_6 - J_3 - J_7 - J_8 - J_4 - J_5 - J_2$	1490	10
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	72

## LAMPIRAN B. PENGUJIAN DENGAN ITERASI SAMA DAN JUMLAH POPULASI SAMA

### B.1 Tabel hasil sepuluh kali pengujian dengan iterasi sama (200) dan jumlah populasi sama (5) pada algoritma *Artificial Immune System*

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	42
2	$J_1 - J_7 - J_6 - J_4 - J_2 - J_3 - J_5 - J_8$	1502	19
3	$J_6 - J_7 - J_8 - J_1 - J_2 - J_3 - J_4 - J_5$	1500	4
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	128
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	167
6	$J_6 - J_7 - J_3 - J_8 - J_1 - J_4 - J_5 - J_2$	1500	4
7	$J_3 - J_6 - J_7 - J_2 - J_1 - J_4 - J_5 - J_8$	1502	7
8	$J_6 - J_7 - J_8 - J_2 - J_3 - J_1 - J_4 - J_5$	1500	17
9	$J_1 - J_6 - J_7 - J_3 - J_8 - J_4 - J_5 - J_2$	1490	7
10	$J_6 - J_7 - J_8 - J_3 - J_4 - J_1 - J_5 - J_2$	1500	24

### B.2 Tabel hasil sepuluh kali pengujian dengan iterasi sama (200) dan jumlah populasi sama (5) pada algoritma *Differential Evolution Plus*

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	103
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	51
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	59
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	52
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	21
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	134
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	171
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	168
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	196
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	45

**LAMPIRAN C. PENGUJIAN DENGAN ITERASI SAMA DAN JUMLAH POPULASI BERBEDA**

**C.1a Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 3 pada algoritma *Artificial Immune System***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	49
2	$J_6 - J_7 - J_1 - J_3 - J_8 - J_4 - J_5 - J_2$	1500	42
3	$J_6 - J_7 - J_1 - J_8 - J_3 - J_4 - J_5 - J_2$	1500	8
4	$J_6 - J_7 - J_8 - J_2 - J_3 - J_1 - J_4 - J_5$	1500	69
5	$J_6 - J_7 - J_2 - J_1 - J_4 - J_3 - J_5 - J_8$	1496	130
6	$J_1 - J_6 - J_7 - J_3 - J_8 - J_4 - J_5 - J_2$	1490	92
7	$J_1 - J_7 - J_6 - J_4 - J_3 - J_2 - J_5 - J_8$	1502	56
8	$J_1 - J_6 - J_7 - J_3 - J_8 - J_4 - J_5 - J_2$	1490	51
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	161
10	$J_6 - J_1 - J_3 - J_7 - J_4 - J_2 - J_5 - J_8$	1514	54

**C.1b Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 3 pada algoritma *Differential Evolution Plus***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	110
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	159
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	100
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	150
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_2 - J_3 - J_5$	1490	111
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	78
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	57
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	68
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	164
10	$J_1 - J_3 - J_7 - J_6 - J_8 - J_4 - J_5 - J_2$	1490	38

**C.2a Tabel hasil sepuluh kali pengujian dengan iterasi =200 dan jumlah populasi = 5 pada algoritma *Artificial Immune System***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_6 - J_7 - J_8 - J_3 - J_4 - J_1 - J_2 - J_5$	1500	6
2	$J_6 - J_7 - J_2 - J_1 - J_8 - J_3 - J_4 - J_5$	1500	34
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	4
4	$J_1 - J_7 - J_6 - J_4 - J_2 - J_3 - J_5 - J_8$	1502	24
5	$J_6 - J_7 - J_8 - J_3 - J_4 - J_2 - J_1 - J_5$	1500	10
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	37
7	$J_6 - J_7 - J_3 - J_8 - J_1 - J_4 - J_5 - J_2$	1500	54
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	34
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	126
10	$J_1 - J_6 - J_7 - J_3 - J_8 - J_4 - J_5 - J_2$	1490	58

**C.2b Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 5 pada algoritma *Differential Evolution Plus***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	111
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	116
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	188
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	67
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	63
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	30
7	$J_1 - J_6 - J_7 - J_3 - J_8 - J_4 - J_5 - J_2$	1490	63
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	52
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	25
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	53

**C.3a Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 7 pada algoritma *Artificial Immune System***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_6 - J_7 - J_1 - J_8 - J_2 - J_3 - J_4 - J_5$	1500	14
2	$J_6 - J_1 - J_3 - J_7 - J_4 - J_5 - J_2 - J_8$	1514	13
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	74
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	46
5	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_2$	1490	34
6	$J_1 - J_3 - J_6 - J_7 - J_8 - J_4 - J_5 - J_2$	1490	7
7	$J_1 - J_6 - J_7 - J_3 - J_8 - J_4 - J_5 - J_2$	1490	26
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	49
9	$J_6 - J_7 - J_8 - J_1 - J_2 - J_3 - J_4 - J_5$	1500	16
10	$J_6 - J_7 - J_2 - J_3 - J_8 - J_1 - J_4 - J_5$	1500	7

**C.3b Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 7 pada algoritma *Differential Evolution Plus***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	160
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	82
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	52
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	105
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	107
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	113
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	142
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	49
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	87
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	149



**C.4a** Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 10 pada algoritma *Artificial Immune System*

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_5$	1490	32
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	19
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	52
4	$J_6 - J_7 - J_2 - J_1 - J_4 - J_3 - J_5 - J_8$	1496	29
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	28
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	148
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	79
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	83
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	101
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	16

**C.4b** Tabel hasil sepuluh kali pengujian dengan iterasi = 200 dan jumlah populasi = 10 pada algoritma *Differential Evolution Plus*

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	62
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	124
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	8
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	97
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	52
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	80
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	84
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	28
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	68
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	28

**LAMPIRAN D. PENGUJIAN DENGAN ITERASI BERBEDA DAN JUMLAH POPULASI SAMA**

**D.1a Tabel hasil sepuluh kali pengujian dengan iterasi = 10 dan jumlah populasi = 5 pada algoritma *Artificial Immune System***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_6 - J_8 - J_2 - J_1 - J_4 - J_3 - J_5 - J_7$	1526	10
2	$J_7 - J_6 - J_2 - J_1 - J_8 - J_4 - J_3 - J_5$	1530	5
3	$J_7 - J_1 - J_6 - J_8 - J_3 - J_2 - J_5 - J_4$	1530	1
4	$J_7 - J_3 - J_1 - J_8 - J_6 - J_4 - J_2 - J_5$	1530	8
5	$J_1 - J_7 - J_6 - J_4 - J_5 - J_8 - J_2 - J_3$	1550	1
6	$J_7 - J_8 - J_1 - J_4 - J_3 - J_5 - J_6 - J_2$	1530	7
7	$J_1 - J_7 - J_8 - J_2 - J_6 - J_5 - J_3 - J_4$	1528	4
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_5 - J_2 - J_3$	1520	7
9	$J_6 - J_7 - J_1 - J_2 - J_3 - J_5 - J_4 - J_8$	1528	3
10	$J_7 - J_8 - J_1 - J_2 - J_3 - J_4 - J_5 - J_6$	1512	7

**D.1b Tabel hasil sepuluh kali pengujian dengan iterasi = 10 dan jumlah populasi = 5 pada algoritma *Differential Evolution Plus***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_8 - J_3 - J_7 - J_4 - J_2 - J_5$	1502	7
2	$J_6 - J_7 - J_3 - J_8 - J_4 - J_2 - J_1 - J_5$	1500	7
3	$J_6 - J_7 - J_8 - J_3 - J_4 - J_2 - J_1 - J_5$	1500	9
4	$J_6 - J_7 - J_1 - J_8 - J_2 - J_3 - J_4 - J_5$	1500	5
5	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_5$	1490	9
6	$J_3 - J_7 - J_6 - J_4 - J_1 - J_2 - J_5 - J_8$	1506	6
7	$J_6 - J_3 - J_7 - J_8 - J_1 - J_4 - J_5 - J_2$	1500	1
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	8
9	$J_3 - J_7 - J_8 - J_2 - J_6 - J_4 - J_1 - J_5$	1512	7
10	$J_1 - J_6 - J_7 - J_8 - J_8 - J_3 - J_2 - J_5$	1490	5

**D.2a Tabel hasil sepuluh kali pengujian dengan iterasi = 50 dan jumlah populasi = 5 pada algoritma *Artificial Immune System***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_6 - J_7 - J_1 - J_4 - J_3 - J_2 - J_5 - J_8$	1502	33
2	$J_3 - J_1 - J_6 - J_7 - J_4 - J_2 - J_5 - J_8$	1514	2
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	31
4	$J_6 - J_1 - J_7 - J_8 - J_2 - J_4 - J_3 - J_5$	1500	18
5	$J_6 - J_7 - J_2 - J_1 - J_4 - J_3 - J_5 - J_8$	1496	40
6	$J_6 - J_7 - J_1 - J_4 - J_3 - J_2 - J_5 - J_8$	1502	9
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	29
8	$J_1 - J_6 - J_3 - J_7 - J_8 - J_4 - J_5 - J_2$	1490	30
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	33
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_2 - J_3 - J_5$	1490	50

**D.2b Tabel hasil sepuluh kali pengujian dengan iterasi = 50 dan jumlah populasi = 5 pada algoritma *Differential Evolution Plus***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	13
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_2 - J_3 - J_5$	1490	26
3	$J_1 - J_6 - J_8 - J_7 - J_2 - J_3 - J_4 - J_5$	1496	43
4	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_5$	1490	31
5	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_5$	1490	26
6	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_5$	1490	22
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	8
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_2 - J_3 - J_5$	1490	47
9	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_5$	1490	41
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_2 - J_5$	1490	17

**D.3a Tabel hasil sepuluh kali pengujian dengan iterasi = 100 dan jumlah populasi = 5 pada algoritma *Artificial Immune System***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_6 - J_3 - J_7 - J_8 - J_2 - J_1 - J_4 - J_5$	1500	81
2	$J_6 - J_7 - J_2 - J_1 - J_4 - J_3 - J_5 - J_8$	1496	5
3	$J_7 - J_8 - J_1 - J_4 - J_2 - J_3 - J_5 - J_6$	1512	11
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	81
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	92
6	$J_7 - J_8 - J_1 - J_4 - J_3 - J_2 - J_5 - J_6$	1512	10
7	$J_1 - J_7 - J_6 - J_2 - J_3 - J_4 - J_5 - J_8$	1502	24
8	$J_1 - J_7 - J_6 - J_4 - J_2 - J_3 - J_5 - J_8$	1502	28
9	$J_6 - J_3 - J_7 - J_8 - J_4 - J_1 - J_5 - J_2$	1500	20
10	$J_1 - J_7 - J_6 - J_4 - J_3 - J_2 - J_5 - J_8$	1502	38

**D.3b Tabel hasil sepuluh kali pengujian dengan iterasi = 100 dan jumlah populasi = 5 pada algoritma *Differential Evolution Plus***

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_2 - J_5$	1490	19
2	$J_1 - J_3 - J_6 - J_7 - J_8 - J_4 - J_5 - J_2$	1490	15
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_2 - J_3 - J_5$	1490	38
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	61
5	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	71
6	$J_1 - J_3 - J_6 - J_7 - J_8 - J_4 - J_5 - J_2$	1490	45
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_2 - J_3 - J_5$	1490	29
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	57
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	43
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	55

**D.4a** Tabel hasil sepuluh kali pengujian dengan iterasi = 150 dan jumlah populasi = 5 pada algoritma *Artificial Immune System*

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_3 - J_7 - J_8 - J_4 - J_5 - J_2$	1490	18
2	$J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_1 - J_5$	1500	14
3	$J_6 - J_7 - J_8 - J_3 - J_1 - J_4 - J_5 - J_2$	1500	10
4	$J_6 - J_7 - J_8 - J_3 - J_4 - J_2 - J_1 - J_5$	1500	11
5	$J_7 - J_8 - J_2 - J_1 - J_4 - J_3 - J_5 - J_6$	1506	34
6	$J_6 - J_7 - J_2 - J_1 - J_8 - J_3 - J_4 - J_5$	1500	30
7	$J_6 - J_7 - J_8 - J_3 - J_1 - J_4 - J_5 - J_2$	1500	13
8	$J_6 - J_7 - J_8 - J_1 - J_4 - J_3 - J_2 - J_5$	1500	31
9	$J_6 - J_7 - J_1 - J_3 - J_8 - J_4 - J_5 - J_2$	1500	55
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	134

**D.4b** Tabel hasil sepuluh kali pengujian dengan iterasi = 150 dan jumlah populasi = 5 pada algoritma *Differential Evolution Plus*

Pengujian ke-	Urutan Jadwal	Makespan (menit)	Konvergen
1	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	116
2	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	20
3	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	55
4	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	121
5	$J_1 - J_6 - J_7 - J_8 - J_2 - J_3 - J_4 - J_5$	1490	27
6	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	120
7	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	105
8	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	24
9	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	139
10	$J_1 - J_6 - J_7 - J_8 - J_4 - J_3 - J_5 - J_2$	1484	45