

BAB III

METODE PENELITIAN

Metode penelitian adalah tahap yang harus ditetapkan terlebih dahulu sebelum melakukan penyelesaian masalah yang sedang dibahas. Bab ini menjelaskan tahapan-tahapan yang dilakukan dalam penelitian mengenai *flexible job shop* dengan algoritma genetika.

3.1 Deskripsi Masalah

Penelitian ini meneliti masalah penjadwalan produksi *flexible job shop*. Misalkan terdapat n *job* yang dinyatakan sebagai I di mana $I = \{I_1, I_2, \dots, I_n\}$ dan m mesin $M = \{M_1, M_2, \dots, M_m\}$ dengan beberapa alternatif mesin untuk setiap operasinya. Untuk menyelesaikan sebuah *job*, diperlukan sejumlah operasi tertentu dengan urutan operasi yang telah ditentukan. Setiap operasi memerlukan satu mesin yang dipilih dari alternatif mesin yang tersedia.

Pada FJS, terdapat dua hal yang perlu dilakukan, yaitu pemilihan mesin (*machine selection*) serta pengurutan operasi (*operation sequence*). Dengan demikian perlu ditentukan penugasan operasi ke mesin serta urutan pengerjaan dari setiap operasi pada setiap mesinnya. Tujuan penyelesaian masalah *flexible job shop* adalah untuk menentukan penugasan yang dapat meminimumkan total waktu yang dibutuhkan untuk menyelesaikan keseluruhan *job* atau biasa disebut sebagai *makespan*. Pada penelitian ini, masalah FJS akan diselesaikan menggunakan algoritma genetika.

3.2 Tahapan Penelitian

Tahapan penelitian yang dilakukan adalah sebagai berikut.

1. Studi Pustaka

Studi pustaka pada penelitian ini dilakukan dengan mempelajari teori dan ilmu yang berhubungan dengan permasalahan yang diteliti. Dengan melakukan studi pustaka, akan diperoleh kumpulan sumber pustaka. Sumber pustaka ini diperoleh dengan mengumpulkan jurnal, skripsi, buku, e-book, dan lain-lain yang berhubungan dengan penjadwalan produksi, *flexible job shop*, dan algoritma genetika.

2. Pengumpulan Data

Data yang digunakan pada penelitian ini terdiri atas data order, data operasi setiap ordernya, data mesin yang terdiri atas dua jenis, yaitu data pekerja dan mesin, dan data waktu proses setiap alternatif mesin.

3. Pemodelan

Pada tahap ini dibangun model optimisasi dari masalah penjadwalan *flexible job shop* dengan terlebih dahulu mendefinisikan himpunan, parameter, dan variabel keputusan dari model optimisasi, serta fungsi tujuan dan fungsi kendala.

4. Penyelesaian Model

Pada tahap ini model matematika akan diselesaikan dengan menerapkan algoritma genetika.

5. Validasi

Validasi dilakukan untuk menguji apakah model optimisasi yang dibangun dan teknik penyelesaiannya memuat kesalahan atau tidak. Validasi dalam penelitian ini dilakukan dengan membandingkan solusi optimal yang dihasilkan dari penerapan model dengan menggunakan bahasa pemrograman Python dengan solusi optimal hasil perhitungan manual dari suatu kasus FJS berukuran kecil. Jika solusinya sama, maka tahapan akan dilanjutkan ke implementasi. Jika Solusi optimal yang dihasilkan tidak sama, maka tahapan akan diulang dari pembangunan model optimisasi.

6. Implementasi

Pada tahapan ini, model dan teknik penyelesaian menggunakan algoritma genetika akan diimplementasikan untuk menyelesaikan masalah penjadwalan produksi batik di salah satu perusahaan di Kabupaten Cirebon.

7. Penarikan Kesimpulan

Dari hasil pengumpulan data, pengolahan data, analisis hasil pengolahan data, serta pencapaian tujuan yang dirumuskan di awal penelitian, selanjutnya akan ditarik kesimpulan.

3.3 Model Optimisasi *Flexible Job Shop*

Masalah FJS akan dimodelkan sebagai model optimisasi merujuk pada Zhang, dkk (2011). Asumsi-asumsi yang diambil pada pemodelan adalah berikut:

1. Semua mesin tersedia pada saat waktu 0 ($t = 0$);
2. Semua *job* datang pada saat waktu 0 ($t = 0$);
3. Setiap operasi hanya dapat diproses pada satu mesin dalam satu waktu tanpa ada interupsi;
4. Setiap mesin hanya dapat melakukan satu operasi dalam satu waktu;
5. Mesin dapat digunakan kembali ketika sebuah *job* dapat dikerjakan di suatu mesin lebih dari satu kali;
6. Untuk setiap *job*, urutan operasi telah ditentukan sebelumnya.

Tahapan pertama dalam pemodelan adalah mendefinisikan parameter dan variabel keputusan. Untuk menyelesaikan masalah FJS, dibentuk model *Mixed Integer Linear Programming* (MILP), Berikut adalah pendefinisian indeks, parameter, dan variabel keputusan yang digunakan oleh model Zhang, dkk (2011).

1. Himpunan

- I : Himpunan *job*
- J : Himpunan operasi
- M : Himpunan mesin
- M_{ij} : Himpunan mesin yang dapat memproses operasi j pada *job* i
- J_i : Himpunan operasi dari *job* i

2. Parameter

- $O_{i,j}$: Operasi ke- j dari J_i
- J_{i0} : Jumlah total operasi dari J_i
- Ω_{ij} : Mesin yang tersedia dari O_{ij}
- p_{ijk} : Waktu proses dari operasi O_{ij} pada mesin k
- S_{ijk} : Waktu mulai dari operasi O_{ij} pada mesin k
- E_{ijk} : Waktu akhir dari operasi O_{ij} pada mesin k
- c_{ijk} : Waktu penyelesaian operasi O_{ij} pada mesin k

c_i : Waktu penyelesaian *job* i

$L = \sum_{i \in I} J_{i0}$: Jumlah semua operasi dari semua *job*

N : Sebuah bilangan besar

Variabel keputusan dari model optimisasi masalah penjadwalan *flexible job shop* didefinisikan untuk mengidentifikasi apakah operasi dikerjakan pada suatu mesin tertentu. Variabel tersebut didefinisikan sebagai berikut.

$$v_{ijk} = \begin{cases} 1, & \text{jika } O_{i,j} \text{ dikerjakan pada mesin } k \\ 0, & \text{jika lainnya} \end{cases}$$

Kemudian, variabel keputusan juga digunakan untuk mengidentifikasi apakah operasi tertentu dikerjakan mendahului operasi lainnya. Variabel tersebut didefinisikan sebagai berikut.

$$z_{ijhgk} = \begin{cases} 1, & \text{jika } O_{i,j} \text{ mendahului } O_{h,g} \text{ pada mesin } k \\ 0, & \text{jika lainnya} \end{cases}$$

Kendala model optimisasi masalah *flexible job shop* mewakili *hard constraint*, yaitu kendala yang harus dipenuhi atau tidak boleh dilanggar pada penjadwalan. Kendala-kendala tersebut antara lain sebagai berikut.

1. *Makespan* diperoleh setelah diketahui waktu penyelesaian operasi terakhir dari *job-job*.

$$C_{max} \geq c_i, \forall i \in I$$

2. Perbedaan antara waktu mulai dan waktu penyelesaian sekurang-kurangnya waktu proses pada mesin k .

$$c_{ijk} \geq s_{ijk} + p_{kij} - (1 - v_{ijk}) * N, \forall i \in I, \forall j \in J, \forall k \in M_{ij}$$

3. Waktu penyelesaian suatu *job* diperoleh setelah waktu penyelesaian operasi terakhir dari suatu *job*.

$$c_i \geq \sum_{k \in M_{ij}} c_{ijk}, \forall i \in I, \forall j \in J$$

4. Waktu mulai dan waktu selesai dari $O_{i,j}$ sama dengan 0 jika operasi $O_{i,j}$ tidak dikerjakan di mesin k .

$$s_{ijk} + c_{ijk} \leq v_{ijk} * N, \forall i \in I, \forall j \in J, \forall k \in M$$

5. Operasi $O_{i,j}$ dan operasi $O_{h,g}$ tidak dapat dikerjakan pada saat yang sama pada setiap mesin dalam kumpulan $M_{ij} \cap M_{hg}$.

$$s_{ijk} \geq c_{h,gk} - (z_{ijh,gk}) * N, \forall i, h \in I, i \leq h, \forall j, g \in J, \forall k \in M_{ij} \cap M_{hg}$$

$$s_{h,gk} \geq c_{ijk} - (1 - z_{ijh,gk}) * N, \forall i, h \in I, i \leq h, \forall j, g \in J, \forall k \in M_{ij} \cap M_{hg}$$

6. *Precedence relationship* antara operasi-operasi dari sebuah *job* tidak dilanggar (operasi $O_{i,j}$ tidak dimulai sebelum operasi $O_{i,j-1}$ selesai).

$$\sum_{k \in M_{ij}} s_{ijk} \geq \sum_{k \in M_{ij}} c_{ij-1k}, \forall i \in I, \forall j = 2, \dots, |J_i|$$

7. Operasi dikerjakan pada satu dan hanya satu mesin.

$$\sum_{k \in M_{ij}} v_{ijk} = 1, \forall i \in I, \forall j \in J$$

Fungsi tujuan model optimisasi masalah penjadwalan *flexible job shop* adalah meminimumkan waktu penyelesaian (*makespan*) dari setiap *job* yang dikerjakan. Fungsi tujuan tersebut dituliskan sebagai berikut.

Meminimumkan:

$$C_{max}$$

Adapun batasan variabel dari model optimisasi adalah sebagai berikut.

$$s_{ijk} \geq 0, c_{ijk} \geq 0, \forall i \in I, j \in J, k \in M$$

$$C_i \geq 0, \forall i \in I$$

$$v_{ijk} \in \{0,1\} \forall i \in I, j \in J, k \in M$$

$$z_{ijh,gk} \in \{0,1\}, \forall i, h \in I, i \leq h, \forall j, g \in J, \forall k \in M_{ij} \cap M_{hg}$$

Dengan demikian, model optimisasi FJS termasuk dalam kategori *mixed linear integer programming* dan termasuk dalam masalah kombinatorial.

3.4 Teknik Penyelesaian

Pada penelitian ini, masalah penjadwalan *flexible job shop* akan diselesaikan dengan Algoritma Genetika. Algoritma genetika adalah algoritma dengan ruang lingkup yang luas dan cepat untuk menemukan solusi, sehingga cocok diimplementasikan pada masalah kombinatorial yang kompleks. Terdapat enam hal yang perlu diperhatikan dalam penggunaan algoritma genetika, yaitu pendefinisian individu di mana individu menyatakan salah satu solusi yang mungkin,

pendefinisian nilai *fitness* yang merupakan ukuran baik tidaknya sebuah individu, pembangkitan populasi awal, proses reproduksi, evaluasi, dan proses seleksi.

Langkah penyelesaian *flexible job shop* dengan algoritma genetika adalah sebagai berikut menurut Zhang, dkk (2011).

1. Inisiasi Parameter Awal

Algoritma genetika bekerja dengan dipengaruhi oleh beberapa parameter yang harus ditetapkan sebelum algoritma dijalankan. Parameter yang digunakan adalah sebagai berikut.

a. Ukuran populasi (*pop size*)

Ukuran populasi merupakan jumlah kromosom yang akan diproses bersama dalam satu siklus generasi. Ukuran populasi menunjukkan banyak atau sedikitnya variasi kemungkinan untuk melakukan reproduksi antar induk.

b. Maksimum generasi (*max gen*)

Maksimum generasi merupakan jumlah generasi yang akan dihasilkan. Maksimum generasi dapat dijadikan sebagai kriteria pemberhentian algoritma genetika. Semakin besar nilai maksimum generasi, maka akan menentukan besarnya ruang eksplorasi sehingga memungkinkan nilai yang dihasilkan bersifat global optimum.

c. *Crossover rate* (C_r)

Crossover rate mengacu pada kemungkinan atau probabilitas dengan mana dua individu dalam populasi akan dipilih untuk melakukan *crossover* sehingga menghasilkan keturunan baru. Nilai *crossover rate* (C_r) ditetapkan antara 0 sampai 1.

d. *Mutation rate* (M_r)

Mutation rate mengacu pada seberapa sering terjadi mutasi pada gen-gen penyusun kromosom. Nilai *mutation rate* (M_r) ditetapkan antara 0 sampai 1.

e. *Num crossover*

Num crossover menunjukkan berapa kali suatu individu mengalami *crossover*. *Num crossover* dapat dihitung dengan rumus berikut.

$$\text{Num crossover} = C_r \times \text{pop size}$$

f. *Num* mutasi

Num mutasi menunjukkan berapa kali suatu individu mengalami mutasi. *Num* mutasi dapat dihitung dengan rumus berikut.

$$\text{Num mutasi} = M_r \times \text{pop size}$$

2. Representasi Kromosom

Dalam populasi, kromosom direpresentasikan dengan tabel berisi kombinasi dari *job*, operasi, mesin, dan waktu mulai operasinya. Setiap individu terdiri atas dua kromosom, yaitu kromosom A dan kromosom B. Kromosom A disebut dengan *machine selection*, mendefinisikan pemilihan mesin dari setiap operasinya dan waktu mulai operasinya dan kromosom B disebut dengan *operation sequence*, mendefinisikan urutan operasi dari setiap mesin berdasarkan pemilihan mesin pada kromosom A (Lehmann, 2002).

Pada *machine selection*, inisialisasi populasi dilakukan dengan setiap gen ganjil dibangkitkan bilangan *random*. Bilangan *random* yang dibangkitkan berada dalam rentang $(1, \Omega_{ij})$, di mana Ω_{ij} merupakan jumlah mesin yang tersedia pada operasi *j* pada *job i*. Nilai bilangan *random* mesin tersebut memiliki panjang *array* sebanyak jumlah operasi dari seluruh *job* (L). Setiap nilai bilangan bulat merupakan index *array* alternatif mesin dari setiap operasi. Hal ini dapat dilihat pada Tabel 3.1 panjang kromosom A ialah 5 sesuai dengan jumlah operasi dari seluruh *job*. Dapat dilihat bilangan random untuk operasi $O_{(1,1)}$ ialah 1, maka mesin ke-1 dari alternatif mesin $O_{(1,1)}$ dipilih, yaitu M_1 .

Kemudian pada gen genap dilakukan perhitungan waktu mulai untuk setiap operasi pada mesin yang terpilih. Waktu mulai suatu *job* dimulai dari 0 dengan ketentuan waktu mulai adalah waktu akhir dari sub-pekerjaan terakhir yang dijalankan pada mesin tersebut. *Machine selection* dibaca dari kiri ke kanan.

Operation sequence diperoleh berdasarkan inisialisasi *machine selection*. *Operation sequence* menunjukkan urutan operasi yang harus dikerjakan pada setiap mesinnya. *Operation sequence* dibaca dari kiri ke kanan. Contoh representasi dari kromosom A dapat dilihat pada Tabel 3.1, sedangkan representasi dari kromosom B dapat dilihat pada Tabel 3.2.

Tabel 3.1 Contoh Representasi Kromosom A

Simbol Operasi	$O_{(1,1)}$		$O_{(1,2)}$		$O_{(2,1)}$		$O_{(2,2)}$		$O_{(2,3)}$	
Waktu Mulai & Mesin	0	1	0	3	1	2	5	1	7	2
Simbol Mesin	M_1		M_3		M_2		M_1		M_2	

Sumber: (Lehmann, 2002)

Tabel 3.2 Contoh Representasi Kromosom B

M_1		M_2		M_3
$O_{(1,1)}$	$O_{(2,2)}$	$O_{(2,1)}$	$O_{(2,3)}$	$O_{(1,2)}$

Sumber: (Lehmann, 2002)

3. Inisialisasi Populasi

Proses pembangkitan populasi awal dilakukan secara acak. Populasi tersebut berisi beberapa kromosom yang telah didefinisikan sebelumnya. Banyaknya populasi awal yang dibangkitkan ditentukan oleh parameter ukuran populasi (*pop size*).

4. Hitung Nilai *Fitness*

Nilai *fitness* dari setiap kromosom dihitung dengan menjumlahkan semua waktu penyelesaian operasi dari suatu *job* serta pelanggaran setiap operasinya. Waktu penyelesaian operasi dari suatu *job* dinyatakan dengan c_{ijk} , sedangkan waktu proses suatu operasi pada suatu mesin dinyatakan dengan p_{ijk} . Untuk setiap operasi, waktu penyelesaian dihitung sebagai waktu mulai operasi ditambah dengan waktu proses operasi di mesin yang sesuai. Kemudian, diterapkan beberapa batasan (*restrictions*) yang memengaruhi nilai *fitness*, yaitu sebagai berikut:

1. Untuk setiap *job*, waktu penyelesaian operasi saat ini harus kurang dari waktu mulai operasi berikutnya dan waktu mulai operasi saat ini harus kurang dari waktu penyelesaian operasi berikutnya. Jika ada pelanggaran terhadap kriteria ini, maka jumlah pelanggaran (*fouls*) = 4 ditambahkan.
2. Operasi harus dilakukan pada mesin yang memungkinkan. Jika operasi dilakukan pada mesin yang tidak memungkinkan, maka jumlah pelanggaran (*fouls*) = 2 ditambahkan.

3. Untuk setiap mesin, operasi harus dimulai pada waktu 0 dan satu mesin harus memproses satu operasi pada satu waktu. Jika ada pelanggaran terhadap kriteria ini, maka jumlah pelanggaran (*fouls*) = 2 ditambahkan.
4. Jika tidak ada operasi yang dimulai pada waktu 0 di setiap mesin, maka jumlah pelanggaran (*fouls*) = 2 ditambahkan.
5. Jumlah pelanggaran ditambahkan ke nilai *fitness*.

Maka, dari kromosom pada langkah inialisasi populasi dihitung sebagai berikut.

$$\sum_{i=1}^{J_i} p_{ijk} + fouls$$

5. *Crossover*

Untuk mendapatkan individu baru, salah satu cara yang dilakukan adalah melakukan persilangan (*crossover*). *Crossover* dilakukan dengan melibatkan dua induk *parent* dengan tujuan akan menghasilkan kromosom baru atau *offspring*. Pada penelitian ini, *crossover* dilakukan dengan metode *partially mapped crossover* (PMX). Diambil *crossover rate* (C_r) sebesar 0,1. Langkah-langkah *partially mapped crossover* (PMX) adalah sebagai berikut :

- 1) Ambil 2 induk dengan nilai *fitness* terbesar kemudian tentukan dua titik secara *random* pada kromosom induknya. Contoh induk 1 dapat dilihat pada Tabel 3.3 dan Tabel 3,4, sedangkan induk 2 pada Tabel 3.5 dan Tabel 3.6. Titik *random* yang diambil pada Induk 1 diberi blok warna kuning, sedangkan titik *random* yang diambil pada Induk 2 diberi blok warna hijau.

Tabel 3.3 Kromosom A Induk 1 pada Proses *Crossover*

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_1	M_3	M_2	M_1	M_2

Tabel 3.4 Kromosom B Induk 1 pada Proses *Crossover*

M_1		M_2		M_3
$O_{(1,1)}$	$O_{(2,2)}$	$O_{(2,1)}$	$O_{(2,3)}$	$O_{(1,2)}$

Tabel 3.5 Kromosom A Induk 2 pada Proses *Crossover*

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_2	M_3	M_3	M_3	M_2

Tabel 3.6 Kromosom B Induk 2 pada Proses *Crossover*

M_2		M_3		
$O_{(1,1)}$	$O_{(2,3)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$

- 2) Lakukan penukaran titik *random* antara induk 1 dan induk 2. Kromosom hasil penukaran pada Induk 1 dapat dilihat pada Tabel 3.7 dan Tabel 3.8, sedangkan kromosom hasil penukaran pada Induk 2 dapat dilihat pada Tabel 3.9 dan Tabel 3.10.

Tabel 3.7 Kromosom A Induk 1 Penukaran Titik pada Proses *Crossover*

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_1	M_3	M_3	M_3	M_2

Tabel 3.8 Kromosom B Induk 1 Penukaran Titik pada Proses *Crossover*

M_1	M_2	M_3		
$O_{(1,1)}$	$O_{(2,3)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$

Tabel 3.9 Kromosom A Induk 2 Penukaran Titik pada Proses *Crossover*

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_2	M_3	M_2	M_1	M_2

Tabel 3.10 Kromosom B Induk 2 Penukaran Titik pada Proses *Crossover*

M_1	M_2		M_3	
$O_{(2,2)}$	$O_{(1,1)}$	$O_{(2,1)}$	$O_{(1,2)}$	$O_{(2,3)}$

Diantara dua titik tersebut akan terdapat hubungan pemetaan yang disebut dengan *mapping*. Area pemetaan tersebut dapat didefinisikan seperti pada Tabel 3.11 sebagai berikut.

Tabel 3.11 Hubungan Pemetaan *Crossover*

$O_{(2,1)}$	$O_{(2,2)}$
M_2	M_1
↕	↕
$O_{(2,1)}$	$O_{(2,2)}$
M_3	M_3

- 3) Lakukan pemetaan kromosom hasil pertukaran titik pada langkah ke 2, pertahankan titik yang berada dalam hubungan pemetaan. Jika terdapat angka yang sama pada area dalam dan luar pemetaan, maka angka yang harus diganti adalah angka yang berada di luar area pemetaan. Pergantian angka sesuai dengan hasil definisi *mapping* pada Langkah 3, sehingga akan menghasilkan 2 kromosom baru atau *offspring*. *Offspring* 1 dapat dilihat pada Tabel 3.12 dan Tabel 3.13, sedangkan *Offspring* 2 dapat dilihat pada Tabel 3.13 dan Tabel 3.14.

Tabel 3.12 *Offspring* Kromosom A Induk 1 pada Proses *Crossover*

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_1	M_3	M_3	M_3	M_2

Tabel 3.13 *Offspring* Kromosom B Induk 1 pada Proses *Crossover*

M_1	M_2	M_3		
$O_{(1,1)}$	$O_{(2,3)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$

Tabel 3.14 *Offspring* Kromosom A Induk 2 pada Proses *Crossover*

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_2	M_3	M_2	M_1	M_2

Tabel 3.15 *Offspring* Kromosom B Induk 2 pada Proses *Crossover*

M_1	M_2		M_3	
$O_{(2,2)}$	$O_{(1,1)}$	$O_{(2,1)}$	$O_{(1,2)}$	$O_{(2,3)}$

6. Mutasi

Mutasi merupakan proses untuk mendapatkan individu baru atau reproduksi dalam algoritma genetika, selain *crossover*. Proses mutasi dilakukan pada *offspring* hasil *crossover*. Pada penelitian ini, mutasi dilakukan dengan metode *reciprocal exchange mutation*. Diambil *mutation rate* (M_r) sebesar 0,1. Langkah-langkah *reciprocal exchange mutation* adalah sebagai berikut:

- 1) Ambil *offspring* hasil *crossover* yang akan dilakukan mutasi. Kromosom induk 1 yang akan dilakukan mutasi dapat dilihat pada Tabel 3.16 dan Tabel 3.17, sedangkan induk 2 dapat dilihat pada Tabel 3.18 dan Tabel 3.19.

Tabel 3.16 Kromosom A Induk 1 pada Proses Mutasi

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_1	M_3	M_3	M_3	M_2

Tabel 3.17 Kromosom B Induk 1 pada Proses Mutasi

M_1	M_2	M_3		
$O_{(1,1)}$	$O_{(2,3)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$

Tabel 3.18 Kromosom A Induk 2 pada Proses Mutasi

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_2	M_3	M_2	M_1	M_2

Tabel 3.19 Kromosom B Induk 2 pada Proses Mutasi

M_1	M_2		M_3	
$O_{(2,2)}$	$O_{(1,1)}$	$O_{(2,1)}$	$O_{(1,2)}$	$O_{(2,3)}$

- 2) Pilih nilai penukaran gen pada Induk 1 dan 2 secara acak. Gen terpilih pada induk 1 diberi blok kuning, sedangkan pada induk 2 gen yang terpilih diberi warna hijau. Kromosom Induk 1 dapat dilihat pada Tabel 3.20 dan 3.21, sedangkan Induk 2 dapat dilihat pada Tabel 3.22 dan 3.23.

Tabel 3.20 *Exchange Point* Kromosom A Induk 1 pada Proses Mutasi

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_1	M_3	M_3	M_3	M_2

Tabel 3.21 *Exchange Point* Kromosom B Induk 1 pada Proses Mutasi

M_1	M_2	M_3		
$O_{(1,1)}$	$O_{(2,3)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$

Tabel 3.22 *Exchange Point* Kromosom A Induk 2 pada Proses Mutasi

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_2	M_3	M_2	M_1	M_2

Tabel 3.23 *Exchange Point* Kromosom B Induk 2 pada Proses Mutasi

M_1	M_2		M_3	
$O_{(2,2)}$	$O_{(1,1)}$	$O_{(2,1)}$	$O_{(1,2)}$	$O_{(2,3)}$

- 3) Tukarkan nilai gen pada induk 1 dan induk 2 sehingga terbentuk kromosom baru atau *offspring*. *Offspring* 1 hasil mutasi dapat dilihat

pada Tabel 3.24 dan 3.25, sedangkan *offspring* 2 hasil mutasi dapat dilihat pada Tabel 3.26 dan 3.27.

Tabel 3.24 Hasil Mutasi Kromosom A Induk 1

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_1	M_3	M_3	M_3	M_3

Tabel 3.25 Hasil Mutasi Kromosom B Induk 1

M_1	M_3			
$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$

Tabel 3.26 Hasil Mutasi Kromosom A Induk 2

$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,2)}$	$O_{(2,3)}$
M_2	M_2	M_2	M_1	M_2

Tabel 3.27 Hasil Mutasi Kromosom B Induk 2

M_1	M_2			
$O_{(2,2)}$	$O_{(1,1)}$	$O_{(1,2)}$	$O_{(2,1)}$	$O_{(2,3)}$

7. Evaluasi

Pada tahap ini, dihitung nilai *fitness* dari *offspring* yang telah diperoleh dari proses *crossover* dan mutasi, sehingga diperoleh jumlah populasi baru sebanyak populasi awal ditambah dengan individu hasil *crossover* dan mutasi. Jumlah individu baru adalah sebagai berikut.

$$\text{jumlah populasi baru} = \text{pop size} + \text{jumlah child crossover} + \text{jumlah child mutasi}$$

8. Seleksi

Seleksi dilakukan dengan metode *binary tournament selection* untuk menghasilkan populasi baru untuk generasi selanjutnya. Langkah-langkah *binary tournament selection* adalah sebagai berikut:

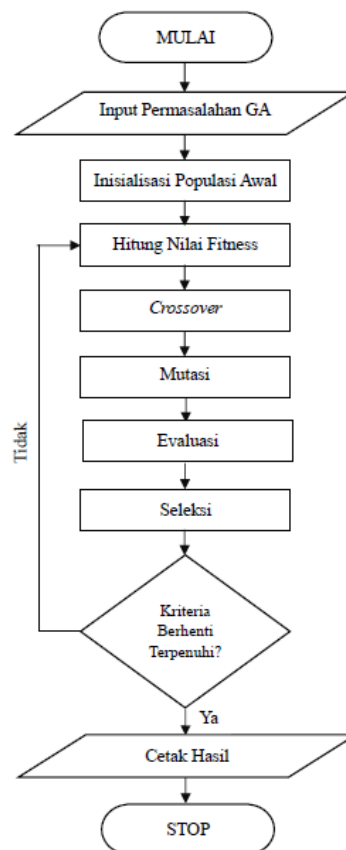
1. Pilih 2 individu yang akan diseleksi secara acak. Sebut individu yang terpilih sebagai Individu 1 dan Individu 2. Ulangi pengacakan tersebut sampai sebanyak jumlah populasi awal dan Individu 1 \neq Individu 2.
2. Bandingkan nilai *fitness* antara Individu 1 dan Individu 2. Pilih individu dengan nilai *fitness* terbesar.

Tahapan seleksi di atas merupakan akhir dari sebuah generasi. Selanjutnya, individu terpilih pada generasi tersebut akan digunakan sebagai populasi awal bagi generasi selanjutnya.

9. Kriteria Penghentian Algoritma

Lakukan langkah 1 sampai langkah 8 sebanyak parameter jumlah generasi yang ditentukan. Pencarian dengan algoritma genetika akan berhenti saat iterasinya telah mencapai jumlah maksimum generasi yang ditentukan. Kemudian, analisis hasil dari iterasi per generasinya dan ambil nilai *fitness* terkecil. Langkah algoritma genetika dapat dilihat pada diagram alur Gambar 3.1 dibawah.

Untuk memperjelas langkah kerja algoritma genetika pada masalah penjadwalan *flexible job shop*, maka diberikan contoh masalah penjadwalan *flexible job shop* berikut. Pada contoh ini, terdapat 4 *job* dengan terdapat 3 operasi pada masing-masing *job*-nya. Adapun alternatif mesin yang digunakan adalah 5 mesin. Tabel 3.28 menunjukkan data *job*, operasi setiap *job*-nya, serta alternatif mesin yang dapat digunakan untuk mengerjakan *job* tersebut.



Gambar 3.1 *Flow Chart* Algoritma Genetika Pada Studi Kasus

Tabel 3.28 Contoh Data *Job*, Operasi, dan Mesin

Job	Simbol Operasi	M1	M2	M3	M4	M5
J_1	$O_{1,1}$	9	6		12	
	$O_{1,2}$			6		7
	$O_{1,3}$		2	4		
J_2	$O_{2,1}$	5			7	9
	$O_{2,2}$		4	6	8	
	$O_{2,3}$			4		8
J_3	$O_{3,1}$	3				
	$O_{3,2}$		3		5	
	$O_{3,3}$	8		10		13
J_4	$O_{4,1}$	6		9		10
	$O_{4,2}$		3			
	$O_{4,3}$	2			4	

Sumber: Zhang (2011).

Penjelasan contoh data *job*, operasi, dan mesin adalah sebagai berikut. Misalnya ditetapkan *pop size* sebanyak 10, *crossover rate* (C_r) sebesar 0,1, dan *mutation rate* (M_r) sebesar 0,1. Langkah pertama adalah inisialisasi populasi. Inisialisasi populasi dilakukan dengan membangkitkan bilangan *random* pada mesin yang akan digunakan. Inisialisasi populasi dapat dilihat pada Tabel 3.29 sampai Tabel 3.48 sebagai berikut.

Tabel 3.29 Kromosom A Individu 1

2	2	1	1	3	2	3	2	1	3	3	2
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M2	M2	M1	M1	M3	M2	M3	M2	M1	M3	M3	M2

Tabel 3.30 Kromosom B Individu 1

M1		M2		M4			M5		
O21	O33	O11	O13	O22	O32	O43	O12	O23	O41

Tabel 3.31 Kromosom A Individu 2

2	1	3	1	3	3	1	2	2	2	2	2
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M2	M1	M3	M1	M3	M3	M1	M2	M2	M2	M2	M2

Tabel 3.32 Kromosom B Individu 2

M1		M2	M3			M4		
O21	O31	O11	O12	O33	O41	O22	O32	O43

Tabel 3.33 Kromosom A Individu 3

3	1	1	1	2	3	2	1	3	1	3	1
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M1	M1	M1	M2	M3	M2	M1	M3	M1	M3	M1

Tabel 3.34 Kromosom B Individu 3

M1			M2		M3		M4	M5
O21	O41	O43	O13	O32	O12	O22	O11	O33

Tabel 3.35 Kromosom A Individu 4

3	1	2	3	1	3	3	1	2	1	3	1
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M1	M2	M3	M1	M3	M3	M1	M2	M1	M3	M1

Tabel 3.36 Kromosom B Individu 4

M1		M2		M3			M4	M5
O41	O43	O22	O32	O12	O13	O33	O11	O21

Tabel 3.37 Kromosom A Individu 5

3	3	2	1	2	1	2	2	1	1	3	2
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M3	M2	M1	M2	M1	M2	M2	M1	M1	M3	M2

Tabel 3.38 Kromosom B Individu 5

M1			M3			M4		
O21	O33	O41	O13	O22	O23	O11	O32	O43

Tabel 3.39 Kromosom A Individu 6

2	2	3	2	3	3	1	3	2	3	2	3
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M2	M2	M3	M2	M3	M3	M1	M3	M2	M3	M2	M3

Tabel 3.40 Kromosom B Individu 6

M1	M2	M3	M4		M5	
O31	O11	O33	O21	O22	O12	O42

Tabel 3.41 Kromosom A Individu 7

3	1	1	1	3	2	3	2	1	2	3	3
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M1	M1	M1	M3	M2	M3	M2	M1	M2	M3	M3

Tabel 3.42 Kromosom B Individu 7

M1		M2	M3		M4			M5
O12	O33	O13	O12	O41	O11	O22	O32	O23

Tabel 3.43 Kromosom A Individu 8

2	1	2	3	1	2	2	3	1	2	3	1
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M2	M1	M2	M3	M1	M2	M2	M3	M1	M2	M3	M1

Tabel 3.44 Kromosom B Individu 8

M1		M2		M3			M5	
O33	O43	O22	O11	O12	O13	O41	O23	O21

Tabel 3.45 Kromosom A Individu 9

3	1	1	2	3	3	1	3	2	3	2	3
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M1	M1	M2	M3	M3	M1	M3	M2	M3	M2	M3

Tabel 3.46 Kromosom B Individu 9

M1	M2	M3		M4			
O31	O13	O12	O33	O11	O21	O22	O41

Tabel 3.47 Kromosom A Individu 10

1	1	2	2	1	3	1	2	2	2	3	1
O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M1	M1	M2	M2	M1	M3	M1	M2	M1	M2	M3	M1

Tabel 3.48 Kromosom B Individu 10

M1			M2	M3				M4	
O11	O31	O43	O22	O12	O13	O33	O41	O21	O32

Setelah dilakukan inisialisasi populasi, selanjutnya adalah menentukan nilai *fitness* dari setiap kromosomnya. Perhitungan nilai *fitness* dapat dilihat pada tabel 3.49 sebagai berikut.

Langkah selanjutnya adalah melakukan reproduksi algoritma genetika, yaitu dengan *crossover* dan mutasi. Misalkan ditetapkan *crossover rate* (C_r) sebesar 0,1 hasilnya diperoleh individu melakukan *crossover* sebanyak 1 kali dan *mutation rate* (M_r) sebesar 0,1 sehingga individu melakukan mutasi sebanyak 1 kali. Individu yang akan direproduksi adalah 2 individu terbaik dari nilai *fitness* yang telah dihitung. Jadi, individu yang akan dilakukan *crossover* adalah individu 5 dan individu 3. Tabel 3.50 dan Tabel 3.51 adalah kromosom A dari individu yang akan dilakukan *crossover*.

Tabel 3.49 Nilai *Fitness* Setiap Kromosom

Individu	Nilai <i>Fitness</i>
1	65
2	59
3	58
4	59
5	57
6	61
7	66
8	59
9	62
10	61

Tabel 3.50 *Crossover* Kromosom A Induk 1

O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M3	M2	M1	M2	M1	M2	M2	M1	M1	M3	M2

Tabel 3.51 *Crossover* Kromosom A Induk 2

O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M1	M1	M1	M2	M3	M2	M1	M3	M1	M3	M1

Blok warna kuning dan hijau merupakan titik *crossover* yang dipilih secara acak. Kemudian pada Tabel 3.52 didefinisikan pemetaan hubungan titik tersebut.

Tabel 3.52 Hubungan Pemetaan *Crossover*

O32	O33
M2	M1
↕	↕
O32	O33
M1	M3

Langkah selanjutnya adalah melakukan *crossover* dengan metode *Partially Mapped Crossover*, sehingga diperoleh *offspring* yang ditunjukkan pada Tabel 3.53 dan Tabel 3.54 sebagai berikut.

Tabel 3.53 *Offspring 1 Hasil Crossover*

O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M3	M2	M1	M2	M1	M2	M1	M3	M1	M3	M2

Tabel 3.54 *Offspring 2 Hasil Crossover*

O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M1	M1	M1	M2	M3	M2	M2	M1	M1	M3	M1

Langkah selanjutnya adalah melakukan mutasi dengan metode *reciprocal exchange mutation* pada *offspring* hasil *crossover*. Tabel 3.55 dan Tabel 3.56 menunjukkan *offspring* hasil *crossover* yang akan dilakukan mutasi. Blok yang diberi warna kuning dan hijau adalah titik yang dipilih secara acak untuk dilakukan mutasi.

Tabel 3.55 Mutasi Kromosom A Induk 1

O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M3	M2	M1	M2	M1	M2	M1	M3	M1	M3	M2

Tabel 3.56 Mutasi Kromosom A Induk 2

O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M1	M1	M1	M2	M3	M2	M2	M1	M1	M3	M1

Setelah dilakukan penukaran titik pada induk mutasi, maka akan diperoleh *offspring* hasil mutasi yang ditunjukkan pada Tabel 3.57 dan Tabel 3.58 sebagai berikut.

Tabel 3.57 Kromosom A Induk 1 Hasil Mutasi

O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M3	M2	M1	M2	M1	M2	M1	M3	M1	M1	M2

Tabel 3.58 Kromosom A Induk 2 Hasil Mutasi

O11	O12	O13	O21	O22	O23	O31	O32	O33	O41	O42	O43
M3	M1	M1	M3	M2	M3	M2	M2	M1	M1	M3	M1

Langkah selanjutnya adalah hitung nilai *fitness* dari *offspring* yang telah diperoleh dari proses *crossover* dan mutasi. Hasilnya adalah diperoleh jumlah individu berjumlah populasi awal ditambah dengan individu hasil *crossover* dan mutasi. Hasil evaluasi dapat dilihat pada Tabel 3.59.

Tabel 3.59 Hasil Evaluasi

Individu	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Nilai Fitness	65	59	58	59	57	61	66	59	62	61	60	55	64	51

Langkah selanjutnya adalah melakukan seleksi. Seleksi dilakukan dengan metode *binary tournament selection*. Tabel 3.60 menunjukkan pemilihan individu sebanyak populasi awal, sedangkan Tabel 3.61 menunjukkan perbandingan nilai *fitness* dari pemilihan individu tersebut.

Tabel 3.60 Pemilihan Individu pada Proses Seleksi

Individu 1	Individu 2	Kondisi	Individu 2
3	8	tidak sama	
2	6	tidak sama	
8	14	tidak sama	
4	4	sama	5
1	4	tidak sama	
12	11	tidak sama	
4	12	tidak sama	
6	8	tidak sama	
4	1	tidak sama	
9	3	tidak sama	

Tabel 3.61 Perbandingan Nilai *Fitness* pada Proses Seleksi

<i>Fitness</i> Individu 1	<i>Fitness</i> Individu 2	Individu Terpilih
58	59	58
59	61	59
59	51	51
59	57	57
65	59	59
55	60	55
59	55	55
61	59	59
59	65	59
62	58	58
Nilai <i>fitness</i> terbaik		59

Tahapan seleksi di atas merupakan akhir dari sebuah generasi. Hasilnya adalah nilai *fitness* terbesar yaitu kromosom 12, ditandai dengan blok berwarna biru.