

BAB II

LANDASAN TEORI

2.1 Penjadwalan

2.1.1 Pengertian Penjadwalan

Penjadwalan dalam proses produksi merupakan sesuatu yang cukup penting, dalam proses penjadwalan dapat menentukan waktu yang dibutuhkan untuk proses produksi serta memprediksi jumlah produksi yang akan dihasilkan perusahaan dalam periode tertentu. Tujuan dari penjadwalan adalah untuk mengoptimalkan penggunaan sumber daya yang ada sehingga tujuan produksi secara keseluruhan dapat terpenuhi (Narasimhan, 1995).

Penjadwalan produksi dapat diklasifikasi berdasarkan beberapa kriteria sebagai berikut:

1. Penjadwalan produksi secara umum, yaitu:
 - a. Penjadwalan maju (*forward scheduling*)
Operasi penjadwalan dimulai dari tanggal penerimaan order secara maju. Konsekuensinya terjadi persediaan bahan baku sampai pada gilirannya diproses.
 - b. Penjadwalan mundur (*backward scheduling*)
Teknik penjadwalan dimulai dari waktu penyelesaian operasi terakhir. Keuntungannya adalah mengurangi persediaan barang setengah jadi (*work in process*).
2. Penjadwalan produksi berdasarkan mesin yang digunakan, yaitu:
 - a. Model *Single-Machine*
Model ini hanya berisi sejumlah m mesin yang sejenis, yang harus memproses sejumlah *job* yang terdiri dari satu operasi. Setiap *job* dapat dikerjakan pada salah satu mesin yang ada tersebut. Model ini banyak digunakan pada penjadwalan yang menggunakan metode dekomposisi, yaitu suatu model penjadwalan yang memecahkan permasalahan penjadwalan yang kompleks ke dalam sejumlah permasalahan yang lebih sederhana dalam bentuk *single-machine*.

b. Model *Paralle-Machine*

Merupakan suatu bentuk umum dari model *single-machine*. Model ini terdapat pada industri yang proses produksinya terdiri dari berbagai tahapan, di mana tiap tahapan tersebut terdiri dari sejumlah mesin sejenis yang tersusun secara paralel. Setiap *job* dapat dikerjakan pada salah satu mesin yang ada. Ada suatu kondisi di mana mesin-mesin yang tersusun secara paralel tersebut tidak identik satu sama lain dalam hal kemampuan berprosesnya.

c. Model *Flow Shop*

Suatu proses manufaktur seringkali harus melewati banyak operasi yang membutuhkan jenis mesin yang berbeda pada tiap operasinya. Jika rute yang harus dilewati untuk setiap *job* adalah sama, maka bentuk konfigurasi ini disebut juga model *flow shop*. Mesin-mesin pada model ini disusun secara seri dan pada saat sebuah *job* selesai diproses pada satu mesin, maka *job* tersebut akan meninggalkan mesin tersebut untuk kemudian mengisi antrian pada mesin berikutnya untuk diproses.

d. Model *Job Shop*

Dalam suatu proses manufaktur yang memerlukan banyak operasi, seringkali rute yang harus dilalui setiap *job* adalah tidak sama. Model seperti ini disebut juga model *job shop*. Bentuk sederhana dari model ini mengasumsikan bahwa setiap *job* hanya melewati satu jenis mesin sebanyak satu kali dalam rutanya pada proses tersebut. Namun ada juga model lainnya di mana setiap *job* diperbolehkan untuk melewati mesin sejenis lebih dari satu kali pada rutanya. Model ini disebut juga *job shop* dengan *recirculation* (pengulangan).

3. Penjadwalan produksi berdasarkan pola kedatangan pekerjaan, yaitu:

a. Penjadwalan statis

Pekerjaan datang bersamaan dan siap dikerjakan pada mesin yang tidak bekerja. Kondisi semua stasiun kerja dan perlengkapannya selalu tersedia pada saat itu.

b. Penjadwalan dinamis

Pekerjaan datang terus-menerus pada waktu yang berbeda-beda. Pendekatan yang sering digunakan pada penjadwalan ini adalah penggunaan aturan *dispatching* yang berbeda untuk setiap stasiun kerja.

4. Penjadwalan produksi berdasarkan sifat informasi yang diterima, yaitu:
 - a. Penjadwalan deterministik
Informasi yang diperoleh bersifat pasti, seperti waktu kedatangan *job*, waktu *setup*, dan waktu proses.
 - b. Penjadwalan stokastik
Informasi yang diperoleh tidak pasti, tetapi memiliki kecenderungan yang jelas atau menyangkut adanya distribusi probabilitas tertentu, misalnya kedatangan pekerjaan bersifat acak (Satrio, 2007).

2.1.2 Tujuan Penjadwalan

Penjadwalan memiliki beberapa tujuan yang diharapkan dicapai setelah adanya penjadwalan. Adapun tujuan penjadwalan produksi antara lain:

- Memenuhi waktu pesanan
- Meminimumkan total waktu penyelesaian (*makespan*).
- Menghasilkan tingkat kegunaan mesin.
- Menetapkan urutan pekerjaan yang tepat.
- Meminimumkan biaya produksi dan tenaga kerja (Nahmias, 1997).

2.2 Asumsi Dasar Penjadwalan

Penyelesaian suatu masalah penjadwalan biasanya memberlakukan beberapa asumsi yang menyangkut karakteristik tugas, operasi mesin yang digunakan dan waktu proses. Hal ini dimaksudkan untuk menyederhanakan masalah penjadwalan tersebut. Asumsi-asumsi dasar tersebut (Baker, 2009), antara lain:

1. Karakteristik *job*

- *Job* terdiri dari urutan operasi yang telah ditentukan.
- Suatu operasi hanya bisa dikerjakan pada satu tipe mesin dari setiap tipe mesin dalam *shop*.

- Waktu proses diketahui dengan pasti seperti halnya *due date*.
- Urutan waktu *set-up* bersifat independen dan waktu transportasi antar mesin dapat diabaikan.
- Operasi yang sedang dikerjakan pada mesin tidak dapat di interupsi.
- Operasi tidak dapat dimulai sampai operasi pendahulunya diselesaikan.
- Setiap mesin hanya dapat memproses satu operasi pada satu waktu.
- Setiap *part* (bagian) hanya dapat diproses di suatu mesin pada satu mesin.

2. Karakteristik operasi

- Setiap operasi merupakan satu kesatuan, walaupun mungkin terdiri dari beberapa unit.
- Setiap operasi yang telah dimulai pengerjaannya pada suatu mesin harus diselesaikan.
- Setiap operasi tidak boleh diproses lebih dari satu mesin pada waktu yang sama.
- Setiap operasi dikerjakan menurut yang telah disusun dan tidak didasarkan pada urutan lainnya.
- Setiap operasi boleh diproses satu kali pada mesin yang sama.
- Setiap operasi dapat diproses pada beberapa jenis mesin yang mampu melaksanakan operasi tersebut.
- Setiap *job* hanya mempunyai satu *routing* dalam memproses operasional.

3. Karakteristik mesin

- Setiap mesin hanya memproses satu tugas pada suatu saat tertentu.
- Setiap mesin secara kontinyu siap untuk dibebani tugas selama proses.
- Penjadwalan apabila tidak mengalami interupsi akibat kerusakan atau perawatan.
- Setiap mesin operasi sesuai dengan informasi waktu dan distribusi yang diketahui secara tepat.

4. Karakteristik Waktu Proses

- Waktu proses telah dikerahui baik rata-rata maupun distribusinya.
- Waktu proses independen terhadap jadwal, artinya urutan *set-up time*.

- Bersifat independen dan *move time* antara mesin dapat diabaikan.
- Setiap waktu proses secara implisit sudah mencakup waktu pemindahan benda kerja, *set-up* dan penghentian mesin.

2.3 Teknik Penjadwalan *Job Shop*

Teknik-teknik penjadwalan untuk jenis penjadwalan *Job Shop* dapat dibedakan dalam dua bentuk yaitu:

1. Teknik Penjadwalan Optimal

Teknik dapat dilakukan dengan berbagai cara, diantaranya:

- Teknik *Integer Programming*
- Teknik *Branch And Bound*

2. Pendekatan *Heuristik*

Teknik ini dapat dilakukan dengan berbagai cara, diantaranya:

- *Priority Dispatching*
- *Sampling*
- *Probabilistik Dispatching*

2.4 *Priority Dispatching Rules*

Aturan prioritas digunakan untuk memenuhi *job* mana yang akan dikerjakan terlebih dahulu pada metode *Priority Dispatching*. Baker (2009) mengklasifikasikan aturan-aturan prioritas ke dalam 2 tipe, yaitu:

1. Aturan Prioritas Lokal

Pada aturan prioritas ini penugasan didasarkan pada informasi yang berkaitan dengan *job* yang berada pada antrian suatu mesin secara individual. Aturan yang termasuk pada tipe ini adalah:

- *Shortest processing Time (SPT)*

Prioritas tertinggi diberikan pada *job* yang memiliki waktu proses terpendek. Aturan ini cenderung mengurangi *work in process*, *mean flow* serta *mean lateness*.

- *Least Work Remaining (LWKR)*

Prioritas tertinggi diberikan pada *job* yang memiliki sisa waktu yang terpendek.

- *First Come First Served (FCFS)*

Prioritas diberikan kepada *job* yang datang terlebih dahulu.

- *Most Work Remaining (MWKR)*

Prioritas tertinggi diberikan pada *job* yang memiliki waktu proses terbanyak.

- *Most Operation Remaining (MOPNR)*

Prioritas tertinggi diberikan pada *job* yang memiliki waktu proses terbanyak dan terpanjang.

2. Aturan Prioritas Global

Aturan prioritas global memanfaatkan informasi atau status dari mesin-mesin yang lainnya. Aturan yang tergolong tipe ini adalah:

- *Anticipates Work In Next Queue (AWINQ)*

Prioritas tertinggi diberikan kepada operasi yang berbeda pada stasiun dengan antrian terpendek.

- *First Of First On (FOFO)*

Prioritas tertinggi diberikan kepada operasi yang selesai paling awal.

Selain itu juga pengklasifikasian aturan ini berdasarkan informasi-informasi yang bersifat dinamis (Baker, 2009), diantaranya:

1. Aturan statis

Tipe ini memandang setiap *job* mempunyai prioritas yang sama. Prioritas yang termasuk ke dalam tipe ini adalah:

- *First Arrival at The Shop Served (FASFS)*

Prioritas tertinggi diberikan pada *job* yang tiba paling awal di shop.

- *Total Work (TWORK)*

Prioritas yang tertinggi diberikan pada *job* yang memiliki total proses untuk seluruh operasi yang dilaksanakan paling sedikit.

- *Earliest Due Date (EDD)*

Prioritas yang tertinggi diberikan pada *job* yang memiliki *due date* paling cepat.

2. Aturan dinamis

Tipe ini memberikan prioritas yang berbeda untuk operasi-operasi yang berbeda pada suatu *job*. Aturan yang termasuk dalam tipe ini adalah:

- *Operation Due Date (OPNDD)*

Due date operasi suatu *job* dapat ditentukan dari perbandingan interval antara *due date job* dengan waktu kedatangan operasi. Disini prioritas tertinggi diberikan pada operasi *due date* tercepat.

- *First Come First Server (FCFS)*

Prioritas yang tertinggi diberikan pada operasi yang terlebih dahulu masuk ke dalam antrian suatu mesin.

- *Slact Time per Operation (ST/O)*

Prioritas yang tertinggi diberikan pada *job* yang memiliki harga *slack time per operation* paling kecil.

- *Slack Time (ST)*

Slack time diperoleh dengan cara mengurangi waktu proses dari *due date*. Prioritas tertinggi diberikan pada *job* yang memiliki *slack time* paling kecil.

2.5 Kriteria Kinerja Job Scheduling

Proses penjadwalan memiliki beberapa kriteria untuk menentukan baik buruknya hasil penjadwalan tersebut (Baker, 2009) diantaranya adalah:

- *Processing Time* (t_i), merupakan waktu yang diperlukan untuk menyelesaikan suatu operasi atau proses termasuk waktu *setup*.
- *Completion Time* (C_i), merupakan rentang waktu mulai dari pekerjaan pertama dimulai sampai dengan pekerjaan selesai dikerjakan.

$$C_i = F_i + r_i \dots \dots \dots (1)$$

- *Flow Time* (F_i), merupakan rentang waktu antara saat pekerjaan siap diproses hingga saat pekerjaan tersebut selesai dikerjakan. Dengan begitu maka *flow*

time adalah *processing time* ditambah dengan waktu tunggu sebelum pekerjaan diproses.

$$F_i = C_i - r_i \dots \dots \dots (2)$$

- *Due Date* (d_i), merupakan batas waktu penyelesaian dari suatu pekerjaan.
- *Waiting Time* (W_i), merupakan waktu tunggu dari suatu pekerjaan saat selesai diproses hingga saat mulai operasi berikutnya.

$$W_i = C_i - r_i - \sum_{i=1}^m t_{ij} \dots \dots \dots (3)$$

- *Slack* (SL_i), merupakan ukuran perbedaan antara waktu sisa dari batas waktu tugas dengan waktu prosesnya.

$$SL_i = d_i - t_i \dots \dots \dots (4)$$

- *Lateness* (L_i), merupakan waktu antara saat selesai dengan batas waktu penyelesaiannya (*due dates*). *Lateness* dapat bernilai negatif (*Earliness*) maupun positif (*Tardiness*). Apabila *lateness* bernilai negatif, maka terjadi *earliness* (E_i) yang berarti bahwa pekerjaan selesai sebelum waktunya. Apabila *lateness* bernilai positif, maka terjadi *tardiness* (T_i) yang berarti bahwa pekerjaan selesai melebihi waktu yang ditentukan.

$$L_i = C_i - d_i \dots \dots \dots (5)$$

- *Makespan* (M), merupakan keseluruhan waktu yang diperlukan untuk menyelesaikan semua pekerjaan. Dengan meminimalisasi nilai *makespan* maka akan dihasilkan *lead time* yang lebih singkat, efisiensi mesin yang lebih tinggi, serta persediaan barang setengah jadi yang lebih kecil.

$$M = \sum_{i=1}^{n+1} S_{[i-1],[i]} + \sum_{i=1}^n t_i \dots \dots \dots (6)$$

- *Manufacturing lead time*, merupakan waktu suatu job berada pada *shop floor* yang terdiri dari waktu setup, waktu operasi, dan waktu non operasi.

Beberapa kriteria performansi dalam program *job shop* diantaranya:

- i : subscript for jobs, $i = 1, \dots, n$
 j : subscript for machines, $j = 1, \dots, m$
 P_{ij} : processing time of job i on machine j
 d_i : due date of job i

r_i	: ready time of job i
C_i	: completion time of job i
F_i	: flow time of job i
W_i	: total waiting time of job i
L_i	: lateness of job i
E_i	: earliness of job i
T_i	: tardiness of job i
N_t	: number of unfinished jobs in the shop at time t
w_i	: assigned weight of job i
MC	: weighted mean completion time (bobot rata-rata waktu penyelesaian)
W_{max}	: maximum waiting time (waktu menunggu maksimum)
MW	: weighted mean waiting time (bobot rata-rata waktu menunggu)
F_{max}	: maximum flow time (waktu alir maksimum)
MF	: weighted mean flow time (bobot rata-rata waktu alir)
L_{max}	: maximum lateness (kelambatan maksimum)
ML	: weighted mean lateness (bobot rata-rata kelambatan)
E_{max}	: maximum earliness
ME	: weighted mean earliness
T_{max}	: maximum tardiness
MT	: weighted mean tardiness
NT	: number of tardy jobs
WIP	: mean work in process (rata-rata kerja dalam proses)
MU	: mean machine utilization (rata-rata utilitas mesin)
TJC	: total job costs, including idle, busy, early, and late costs
TMC	: total machine costs (biaya total mesin), including idle, and busy costs
TC	: total costs (biaya total)

2.6 Kendala-kendala Dalam Proses Penjadwalan

Proses penjadwalan produksi seringkali memiliki beberapa hambatan yang dapat mempengaruhi terhadap proses produksi. Beberapa hambatan yang sering terjadi pada proses penjadwalan produksi antara lain:

- **Mesin rusak**

Pada saat sebuah mesin mengalami kerusakan, maka mesin tersebut tidak dapat digunakan untuk proses produksi. Hal tersebut akan menyebabkan terhentanya proses produksi sebuah perusahaan. Kejadian tersebut akan membuat permintaan konsumen tidak dapat dipenuhi dengan baik. Untuk menyelesaikan permasalahan ini dapat dilakukan dengan cara melakukan penjadwalan ulang (*rescheduling*). Proses penjadwalan ulang dilakukan dimulai pada saat mesin mengalami kerusakan, proses produksi pada saat mesin berjalan dengan baik tidak perlu dilakukan penjadwalan ulang. Untuk memudahkan dalam proses penjadwalan ulang maka perlu dicari beberapa informasi seperti nomor mesin yang rusak, waktu pada saat mesin rusak, serta waktu yang dibutuhkan untuk menyelesaikan perbaikan mesin.

- **Penambahan pesanan baru**

Pada saat proses produksi sedang berjalan ada kemungkinan bahwa akan ada pesanan baru dari konsumen, masuknya pesanan baru dari pihak konsumen tersebut dapat menyebabkan kacanya penjadwalan yang telah dilakukan sebelumnya. Untuk mengatasi permasalahan ini maka dapat dilakukan juga dengan cara penjadwalan ulang. Sama seperti pada saat terjadinya kerusakan mesin, penjadwalan ulang dimulai pada saat pesanan baru datang, proses produksi pada saat pesanan belum datang tidak diperhitungkan. Beberapa informasi yang perlu diperhatikan dalam proses penjadwalan ulang ini adalah produk yang dipesan, *routing* pesanan, jumlah pesanan, serta *due date* yang baru.

- **Perubahan prioritas**

Perubahan prioritas dapat terjadi pada saat terjadinya penjadwalan ulang. Dengan adanya penjadwalan ulang berdasarkan pada keadaan yang ada pada saat itu, menyebabkan kemungkinan untuk terjadinya perubahan prioritas pada saat proses produksi.

- **Perubahan *due date***

Perubahan *due date* dapat menyebabkan terganggunya jadwal proses produksi suatu perusahaan. Perubahan *due date* tersebut memiliki 2 buah macam, yaitu *due date* mundur dan *due date* maju. Untuk perubahan *due date* mundur

tidak akan berpengaruh secara signifikan terhadap penjadwalan produksi karena perusahaan akan memiliki waktu tambahan untuk mengerjakan proses produksinya. Sementara untuk perubahan due date maju akan berpengaruh terhadap proses penjadwalan karena perusahaan harus mengubah jadwal agar produk dapat diselesaikan tepat pada waktu yang diinginkan oleh konsumen.

- **Adanya produk yang memerlukan pengulangan operasi**

Apabila ada produk yang dinyatakan cacat, maka proses pengulangan operasi harus dilakukan oleh perusahaan. Hal tersebut dapat mengakibatkan mundurnya jadwal produksi karena mesin harus mengulang proses produksi bagi produk yang mengalami kegagalan tersebut. Sementara produk yang lain akan terbengkalai karena prioritas produksi dipindahkan terhadap proses pengulangan produksi produk yang cacat.

2.7 Genetic Algorithm

Genetic Algorithm (GA) atau Algoritma Genetika merupakan cabang dari algoritma evolusi yang digunakan untuk memecahkan masalah optimasi. Algoritma ini didasarkan pada proses genetik yang terjadi pada makhluk hidup, dimana perkembangan generasi pada suatu populasi yang alami secara lama kelamaan akan mengikuti seleksi alam yaitu dimana yang kuat yang akan bertahan. Dengan mengikuti teori evolusi tersebut maka algoritma genetik ini dapat digunakan untuk memecahkan masalah yang terjadi pada sehari-hari.

Teori ini pertama kali ditemukan oleh John Holland, dalam algoritma ini bekerja dalam sebuah populasi yang terdiri dari individu-individu yang masing-masing individu merepresentasikan solusi yang ada. Dalam kaitan ini individu dilambangkan sebagai nilai *fitness* yang akan digunakan untuk menentukan solusi terbaik yang ada. Dalam prosesnya masing-masing individu tersebut akan melakukan reproduksi dengan cara perkawinan silang dengan individu lainnya untuk menghasilkan keturunan baru yang lebih baik. Dengan cara tersebut maka individu baru akan terus bermunculan, sedangkan untuk individu yang lemah akan mati sendiri. Semakin banyak proses perkawinan silang dilakukan, maka akan semakin banyak kemungkinan solusi yang akan diperoleh (Entin, 2010).

2.7.1 Langkah-langkah dalam *Genetic Algorithm*

Langkah-langkah yang perlu dilakukan dalam pengolahan data dengan menggunakan metode *Genetic Algorithm* adalah sebagai berikut:

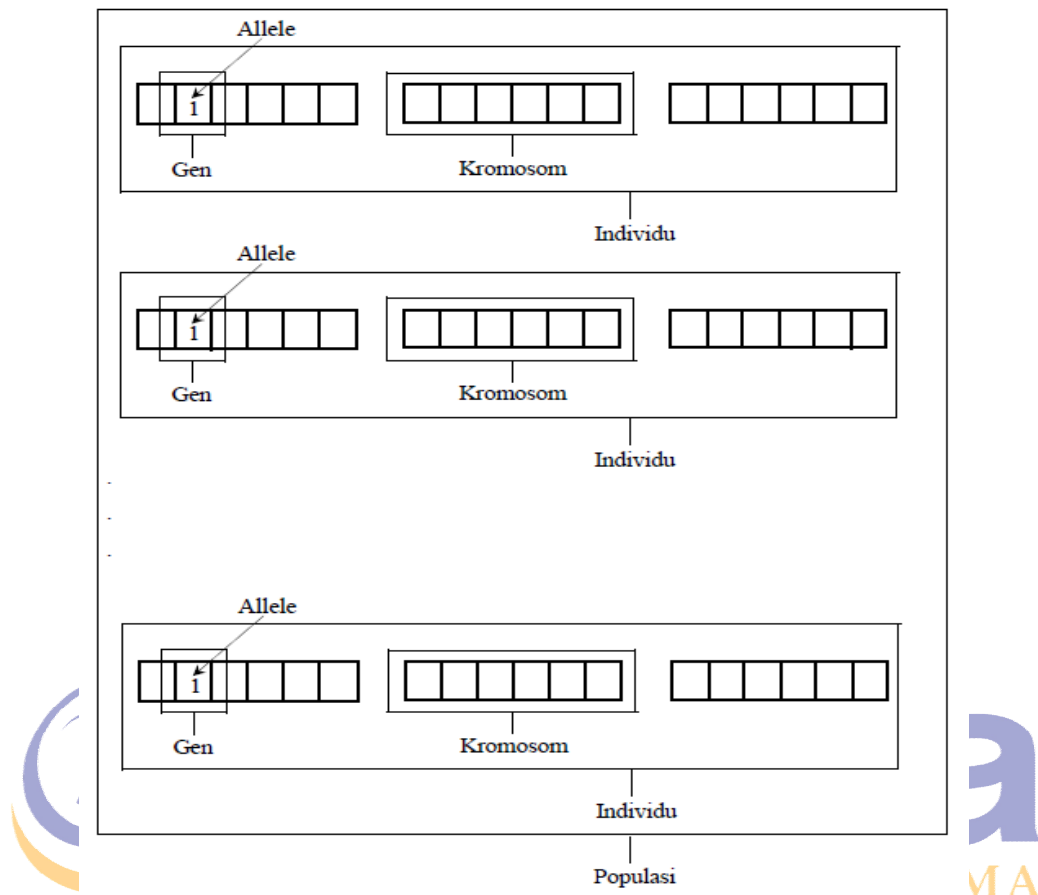
- Mendefinisikan individu, dimana individu merupakan solusi dari permasalahan yang akan diselesaikan.
- Mendefinisikan nilai *fitness*, nilai *fitness* merupakan nilai baik atau tidaknya suatu solusi.
- Pembangkitan populasi awal, proses pembangkitan ini dilakukan dengan menggunakan pembangkitan acak.
- Menentukan proses seleksi yang akan digunakan.
- Menentukan proses perkawinan silang (*crossover*) dan mutasi gen yang akan digunakan.

2.7.2 Istilah dalam *Genetic Algorithm*

Beberapa istilah yang sering digunakan dalam *Genetic Algorithm* adalah sebagai berikut:

- **Individu**, salah satu solusi yang mungkin dilakukan pada metode GA. Sama seperti dalam kehidupan sehari-hari, individu terdiri dari sekumpulan gen.
- **Genotype (gen)**, sebuah nilai yang menyatakan satuan dasar yang membentuk satu kesatuan yang disebut kromosom. Dalam GA gen ini dapat berbentuk nilai biner, float, integer maupun karakter, atau kombinatorial.
- **Allele**, nilai dari gen.
- **Kromosom**, gabungan dari beberapa gen yang membentuk nilai-nilai tertentu.
- **Populasi**, sekumpulan individu yang akan diproses dalam satu siklus evolusi.
- **Generasi**, satu siklus proses evolusi, atau dalam GA disebut satu proses iterasi.
- **Fitness**, suatu nilai yang menyatakan keadaan dari suatu solusi permasalahan yang dicari. Nilai *fitness* menjadi acuan terhadap baik buruknya sebuah solusi

untuk diterapkan pada permasalahan. Dalam *Genetic Algorithm* akan dicari solusi dengan nilai *fitness* tertinggi (Entin, 2010).



Gambar 2.1 Ilustrasi Representasi Permasalahan dalam Algoritma Genetika (sumber : Entin, 2010)

2.7.3 Komponen Utama Dalam *Genetic Algorithm*

Genetic Algorithm memiliki 5 buah komponen utama dalam proses penyelesaiannya, yaitu:

- Teknik pengkodean
- Membangkitkan populasi awal
- Seleksi
- Kawin Silang (*Crossover*)
- Mutasi

A. Teknik Pengkodean

Teknik pengkodean adalah bagaimana proses mengkodekan gen dari kromosom, dimana gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk bit, bilangan *real*, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

Dengan demikian kromosom dapat direpresentasikan dengan menggunakan:

- *String* bit : 10011 dst.
- *Array* bilangan *real* : 65.65, -67.98, 77.34 dst.
- Elemen permutasi : E2, E10, E5 dst.
- Daftar aturan : R1, R2, R3 dst.
- Elemen program : pemrograman genetika
- Struktur lainnya

B. Membangkitkan Populasi Awal

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal. Syarat-syarat yang harus dipenuhi untuk menunjukkan suatu solusi harus benar-benar diperhatikan dalam setiap pembangkitan individunya (Entin, 2010).

Teknik dalam pembangkitan awal ini ada beberapa cara, diantaranya adalah sebagai berikut:

- ***Random generator***

Inti dari cara ini adalah melibatkan pembangkitan bilangan *random* untuk setiap gen sesuai dengan representasi kromosom yang digunakan. Jika menggunakan representasi biner, salah satu penggunaan *random generator* adalah penggunaan rumus berikut untuk pembangkitan populasi awal:

$$IPOP = \text{round}\{\text{random}(N_{ipop}, N_{bits})\} \dots \dots \dots (7)$$

Dimana i_{pop} adalah gen yang nantinya berisi pembulatan dari bilangan *random* yang dibangkitkan sebanyak N_{ipop} (jumlah populasi) \times N_{bits} (jumlah gen tiap kromosom).

- **Pendekatan Tertentu (Memasukan Nilai Tertentu kedalam Gen)**

Cara ini adalah dengan memasukan nilai tertentu kedalam gen populasi awal yang dibentuk.

- **Permutasi Gen**

Salah satu cara permutasi gen dalam pembangkitan awal adalah penggunaan permutasi *Josephus* dalam permasalahan kombinatorial seperti TSP.

C. Seleksi

Seleksi digunakan untuk memilih individu-individu mana saja yang akan dipilih untuk proses kawin silang dan mutasi. Seleksi digunakan untuk mendapatkan calon yang baik. “Induk yang baik akan menghasilkan keturunan yang baik”. Semakin tinggi nilai *fitness* suatu individu semakin besar kemungkinannya untuk terpilih.

Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Nilai *fitness* ini yang nantinya akan digunakan pada tahap-tahap seleksi berikutnya. Masing-masing individu dalam wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai obyektif dirinya sendiri terhadap nilai obyektif dari semua individu dalam wadah seleksi tersebut.

1) Seleksi Dengan Mesin *Roulette*

Metode seleksi dengan mesin *roulette* ini merupakan metode yang paling sederhana dan sering dikenal dengan nama *stochastic sampling with replacement*. Cara kerja metode ini adalah sebagai berikut:

- Dihitung nilai *fitness* dari masing-masing individu (f_i dimana i adalah individu ke-1 sampai ke- n)
- Dihitung total *fitness* semua individu.
- Dihitung probabilitas masing-masing individu.
- Dari probabilitas tersebut, dihitung jatah masing-masing individu pada angka 1 sampai 100.

- Dibangkitkan bilangan random antara 1 sampai 100.
- Dari bilangan random yang dihasilkan, ditentukan individu mana yang terpilih dalam proses seleksi.

2) Seleksi Dengan Turnamen

Pada seleksi dengan turnamen, ditetapkan nilai suatu *tour* untuk individu-individu yang dipilih secara *random* dari suatu populasi. Individu-individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk. Parameter yang digunakan pada metode ini adalah ukuran *tour* yang bernilai antara 2 sampai N (jumlah individu dalam suatu populasi) (Entin, 2010).

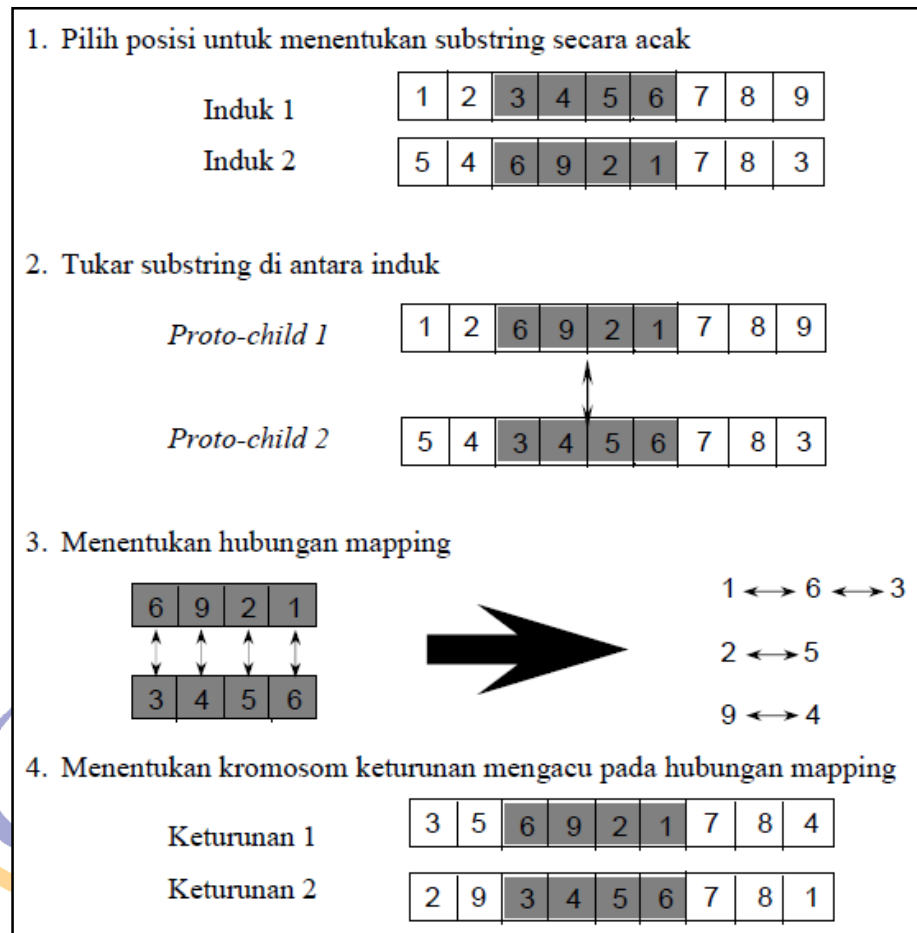
D. Crossover

Kawin silang (*crossover*) adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. *Crossover* menghasilkan titik baru dalam ruang pencarian yang siap diuji. Operasi ini tidak selalu dilakukan pada semua individu yang ada. Individu dipilih secara acak untuk dilakukan *crossing* dengan P_c antara 0,6 sampai dengan 0,95. Jika *crossover* tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunan. Prinsip dari *crossover* ini adalah melakukan operasi (pertukaran, aritmatika) pada gen-gen yang bersesuaian dari dua induk untuk menghasilkan individu baru. Proses *crossover* dilakukan pada setiap individu dengan probabilitas *crossover* yang ditentukan (Entin, 2010).

Untuk proses *crossover* dengan gen berbentuk permutasi metode yang digunakan adalah *Partial-Mapped Crossover (PMX)*. PMX diciptakan oleh Goldberg dan Lingle. PMX merupakan rumusan modifikasi dari kawin silang 2 point. Hal yang penting dalam PMX adalah kawin silang 2 point ditambah dengan beberapa prosedur tambahan. PMX mempunyai langkah kerja sebagai berikut:

- **Langkah 1:** tentukan 2 posisi pada kromosom dengan aturan acak. Substring yang berada dalam dua posisi ini dinamakan daerah pemetaan.
- **Langkah 2:** tukar kedua substring antar induk untuk menghasilkan *proto-child*.
- **Langkah 3:** tentukan hubungan pemetaan diantara 2 daerah pemetaan.

- **Langkah 4:** tentukan kromosom keturunan dengan mengacu pada hubungan pemetaan.



Gambar 2.2 Ilustrasi Prosedur PMX
(Sumber: Entin, 2010)

E. Mutasi

Operator berikutnya pada algoritma genetika adalah mutasi gen. operator ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Kromosom anak dimutasi dengan menambahkan nilai random yang sangat kecil (ukuran langkah mutasi), dengan probabilitas yang rendah. Peluang mutasi P_m didefinisikan sebagai persentase dari total gen pada populasi yang mengalami mutasi. Peluang mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil,

banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila peluang mutasi ini terlalu besar, maka akan terlalu banyak gangguan acak sehingga anak akan kehilangan kemiripan dari induknya dan juga algoritma kehilangan kemampuan untuk belajar dari histori pencarian.

Ada beberapa pendapat mengenai laju mutasi ini. Ada yang berpendapat bahwa laju mutasi sebesar $1/n$ akan menghasilkan hasil yang cukup baik. Ada juga yang beranggapan bahwa laju mutasi tidak tergantung pada ukuran populasinya. Kromosom hasil mutasi harus diperiksa, apakah masih berada pada domain solusi, dan bila perlu dilakukan perbaikan (Entin, 2010).

