

## BAB II

### LANDASAN TEORI

#### 2.1 Penjadwalan

Penjadwalan di dalam suatu sistem produksi merupakan suatu hal yang sangat penting untuk diperhatikan. Penjadwalan terhadap mesin produksi diperlukan agar proses produksi yang akan dilakukan dapat berjalan lancar sesuai dengan jadwal produksi yang sudah di tentukan. Penjadwalan didefinisikan sebagai proses pengalokasian sumber daya untuk mengerjakan sekumpulan tugas dalam jangka waktu tertentu (Baker & Trietsch, 2009).

##### 2.1.1 Klasifikasi Penjadwalan

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 *Parallel-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 rutenya pada proses tersebut. Namun ada juga model lainnya di mana setiap *job* diperbolehkan untuk melewati mesin sejenis lebih dari satu kali pada rutenya. 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.

### 2.1.3 Terminologi Penjadwalan

Proses penjadwalan memiliki beberapa terminologi 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.

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

- *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.

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

- *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.

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

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

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

- *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.

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

- *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.

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

#### 2.1.4 Parameter Performansi Penjadwalan

Parameter performansi digunakan untuk menentukan metode mana yang lebih baik untuk diterapkan pada perusahaan. Parameter performansi yang dapat digunakan antara lain (Ginting, 2009).

1. Efisiensi Indeks ( $E_i$ ) adalah perbandingan antara metode usulan dengan metode yang sedang diterapkan.

$$Ei = \frac{\text{Makespan Perusahaan}}{\text{Makespan Usulan}} \quad (7)$$

Apabila  $Ei = 1$  mengindikasikan bahwa kedua metode memiliki performansi yang sama. Jika  $Ei < 1$  maka metode usulan memiliki performansi yang kurang baik dibandingkan dengan metode yang diterapkan perusahaan dan begitu sebaliknya.

2. *Relative Error (RE)* digunakan untuk menentukan seberapa jauh perbedaan *makespan* yang dihasilkan oleh kedua metode.

$$RE = \frac{\text{Makespan usulan} - \text{Makesp perusahaan}}{\text{Makespan perusahaan}} \quad (8)$$

### 2.1.5 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.

### 2.1.6 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:

#### 1. Mesin rusak

Sebuah mesin mengalami kerusakan, maka mesin tersebut tidak dapat digunakan untuk proses produksi. Hal tersebut akan menyebabkan terhentinya 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.

## 2. Penambahan pesanan baru

Proses produksi yang 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.

## 3. 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.

## 4. 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.

#### 5. 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.

### 2.2 Penjadwalan *Job Shop*

Penjadwalan *job shop* adalah pengurutan pekerjaan untuk lintasan produk yang tidak beraturan (tata letak pabrik berdasarkan proses). Penjadwalan pada proses produksi tipe *job shop* lebih sulit dibandingkan dengan penjadwalan *flow shop*. Hal ini disebabkan karena:

1. *Job Shop* menangani variasi produk yang sangat banyak dengan pola aliran yang berbeda-beda melalui *work center*.
2. Peralatan pada *job shop* digunakan secara bersama-sama oleh bermacam-macam *order* dalam prosesnya, sedangkan peralatan *flow shop* digunakan khususnya hanya untuk satu jenis produk.
3. *Job* yang berbeda mungkin ditentukan oleh prioritas yang berbeda pula. Hal ini mengakibatkan *order* tertentu yang dipilih harus diproses seketika pada saat *order* tertentu ditugaskan pada suatu *work center*, sedangkan pada *flow shop* tidak terjadi permasalahan seperti diatas karena keseragaman *output* yang diproduksi untuk persediaan. Prioritas *order* pada *flow shop* dipengaruhi terutama pada pengirimannya dibandingkan tanggal pemrosesan.

Ukuran keberhasilan dari suatu pelaksanaan aktivitas penjadwalan khususnya penjadwalan *job shop* adalah meminimasi kriteria-kriteria keberhasilan sebagai berikut:

1. Rata-rata waktu alir (*mean flow time*) akan mengurangi persediaan barang setengah jadi.

2. Rata-rata keterlambatan (*mean tardiness*).
3. Jumlah *job* yang terlambat akan meminimasi dari maksimum ukuran keterlambatan.
4. Jumlah mesin yang menganggur.
5. Jumlah persediaan.

Masalah yang terjadi pada penjadwalan dapat diselesaikan dengan cara:

1. *Sequencing*, dapat diselesaikan dengan metode *priority rule* dan *queue*.
2. *Timing*, awal dan akhir tiap *job* dihitung berdasarkan pada urutan, *routing* dan waktu proses.

### 2.2.1 Ruang Jawab Persoalan Penjadwalan *Job Shop*

Persoalan *job shop* mempunyai ciri khas yaitu aliran penjadwalan pekerjaan dalam *shop* tidak searah. Persoalan ini biasanya membutuhkan matriks waktu proses yang menyatakan waktu pemrosesan tiap operasi dari suatu *job* dan matriks *routing* yang menunjukkan urutan mesin untuk mengerjakan beberapa operasi dari suatu *job*. Penyelesaian persoalan  $n$  *job*  $m$  mesin pada lintasan produksi *job shop*  $(n!)^m$  jadwal, suatu jadwal dikatakan layak jika memenuhi kriteria:

1. Tidak ada *overlap* diantara waktu serta proses operasi.
2. Hubungan ketergantungan antar operasi setiap *job* tidak dilanggar.

Jadwal yang layak tersebut dapat diklasifikasikan sebagai berikut (Backer, 1974):

#### 1. Jadwal Semi Aktif

Jadwal semi aktif adalah *set* jadwal yang tidak satupun operasi yang dapat dikerjakan lebih awal tanpa mengubah susunan operasi pada mesin.

#### 2. Jadwal Aktif

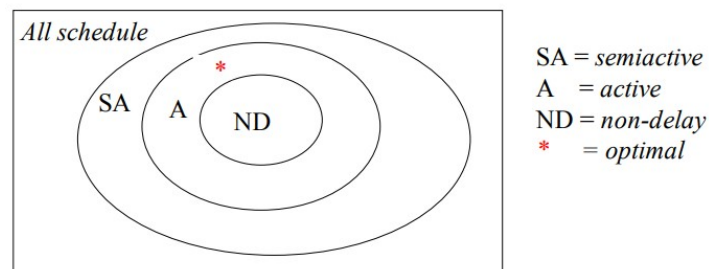
Jadwal aktif pada pengerjaannya didasari pada waktu penyelesaian ( $r_j$ ), jika penjadwalan aktif menggunakan aturan *dispatching rules SPT* (*Shortest Processing Time*) maka proses operasi penjadwalan dimulai berdasarkan waktu penyelesaian paling minimum.

#### 3. Jadwal *Non Delay*

Jadwal *non delay* adalah *set* jadwal yang tidak satupun mesin dibiarkan menganggur jika pada saat yang sama terdapat operasi yang memerlukan

mesin tersebut. Jadwal *non delay* pada pengerjaannya didasari pada waktu awal penjadwalan ( $c_j$ ), jika penjadwalan *non delay* menggunakan aturan *dispatching rules SPT (Shortest Processing Time)* maka proses operasi penjadwalan dimulai berdasarkan waktu awal paling minimum.

Hubungan antara ketiga jenis jadwal diatas dapat diilustrasikan pada diagram Venn dibawah ini:



**Gambar 2.1** Jenis Jadwal pada Penjadwalan *Job Shop*  
(Sumber: Backer, *Principles of Sequencing and Scheduling*, 1974)

### 2.2.2 Teknik Penyelesaian Masalah *Job Shop*

Penyelesaian masalah *job shop* dapat menggunakan beberapa teknik pendekatan, diantaranya sebagai berikut:

#### 1. Teknik Pendekatan Optimasi

Teknik pendekatan optimal merupakan pendekatan yang memberikan solusi terbaik terhadap suatu permasalahan ditinjau dari kriteria tertentu. Pendekatan optimal memiliki dua metode yaitu metode program linier dan metode *branch and bound*. Kedua metode ini memiliki tingkat kesulitan yang tinggi dan belum tentu menghasilkan jadwal yang benar-benar optimal.

#### 2. Teknik Pendekatan Heuristik

Teknik pendekatan heuristik digunakan dalam masalah penjadwalan untuk jumlah operasi lebih kompleks dengan waktu penyelesaian yang relatif lebih cepat. Pendekatan heuristik tidak menghasilkan jadwal yang optimal, namun pendekatan heuristik dapat menghasilkan jadwal yang lebih baik yang mendekati optimal. Teknik pendekatan heuristik antara lain:

- *Priority Dispatching Rules*
- *Sampling Procedures*

- *Probabilistic Dispatching Procedures*

### 2.3 Model Penjadwalan Mesin Heuristik

Salah satu metode penyelesaian permasalahan penjadwalan yang cukup efektif adalah metode algoritma heuristik, yaitu jenis algoritma yang termasuk ke dalam jenis algoritma sub-optimal. Algoritma sub-optimal tidak menjamin tercapainya suatu solusi yang optimal (*best solution*), namun algoritma heuristik secara efektif dapat mengatasi permasalahan yang cukup sulit yang berskala besar dengan cara mencari *good solution* yang dapat memuaskan semua kriteria dengan waktu komputasi yang relatif kecil, selain itu algoritma heuristik juga mudah diimplementasikan dan bersifat fleksibel (Pinedo, 2008).

#### 2.3.1 Dispatching Priority Rules

*Heuristic Dispatching Rules* merupakan metode penjadwalan yang menggunakan aturan prioritas sebagai salah satu kriteria dalam menyusun urutan pekerjaan pada setiap mesin yang ada. Aturan prioritas merupakan aturan yang sering digunakan dalam menyelesaikan masalah-masalah penjadwalan karena cukup mudah dalam implementasinya dan rata-rata waktu penyelesaian pekerjaan lebih cepat. Penelitian ini difokuskan kepada tiga aturan *Heuristic Dispatching Rules* yaitu *LPT*, *SPT* dan *EDD*.

##### 1. *Longest Processing Time (LPT)*

Merupakan metode penjadwalan yang memberikan prioritas tertinggi pada waktu penyelesaian *job* paling lama diselesaikan.

##### 2. *Sortest Processing Time (SPT)*

Prioritas tertinggi diberikan pada *job* yang memiliki waktu proses terpendek. Aturan ini merupakan teknik terbaik untuk meminimalkan aliran pekerjaan dan meminimalkan jumlah pekerjaan rata-rata dalam sistem. Kelemahan utamanya adalah pekerjaan yang memiliki waktu pemrosesan panjang dapat tidak dikerjakan secara terus menerus, karena pekerjaan yang memiliki waktu pemrosesan pendek selalu didahulukan. Pelanggan dapat melihat hal ini secara samar dan penyesuaian berkala untuk pekerjaan yang panjang harus dilakukan.

### 3. Earliest Due Date (EDD)

Prioritas yang tertinggi diberikan pada *job* yang memiliki *due date* paling cepat. Aturan ini bekerja baik ketika keterlambatan menjadi sebuah persoalan utama. Meminimalkan keterlambatan yang mungkin perlu untuk pekerjaan yang memiliki penalti setelah tanggal tertentu.

Secara garis besar algoritma dalam *Dispatching Priority Rules* dirumuskan sebagai berikut:

	RULE	DATA	ENVIRONMENT
1	SIRO	-	-
2	ERD	$r_j$	$1 \mid r_j \mid \text{Var}(\sum(C_j - r_j)/n)$
3	EDD	$d_j$	$1 \mid \mid L_{\max}$
4	MS	$d_j$	$1 \mid \mid L_{\max}$
5	SPT	$p_j$	$Pm \mid \mid \sum C_j; Fm \mid p_{ij} = p_j \mid \sum C_j$
6	WSPT	$w_j, p_j$	$Pm \mid \mid \sum w_j C_j$
7	LPT	$p_j$	$Pm \mid \mid C_{\max}$
8	SPT-LPT	$p_j$	$Fm \mid \text{block}, p_{ij} = p_j \mid C_{\max}$
9	CP	$p_j, prec$	$Pm \mid \text{prec} \mid C_{\max}$
10	LNS	$p_j, prec$	$Pm \mid \text{prec} \mid C_{\max}$
11	SST	$s_{jk}$	$1 \mid s_{jk} \mid C_{\max}$
12	LFJ	$M_j$	$Pm \mid M_j \mid C_{\max}$
13	LAPT	$p_{ij}$	$O2 \mid \mid C_{\max}$
14	SQ	-	$Pm \mid \mid \sum C_j$
15	SQNO	-	$Jm \mid \mid \gamma$

**Gambar 2.2** Algoritma *Dispatching Priority Rules*

(Sumber: Pinedo, *Scheduling: Theory, Algorithms and Systems*, 2007)

### 2.3.2 Algoritma Local Search

*Local Search* merupakan bagian dari algoritma *meta-heuristic* yang secara garis besar merupakan jenis algoritma *improvement*. Proses kerja pada algoritma *Local Search* yaitu diawali berdasarkan suatu jadwal *existing* yang kemudian didapatkan jadwal yang lebih baik dengan manipulasi jadwal saat ini. Prosedur dalam algoritma *Local Search* tidak menjamin solusi yang optimal, tetapi bisa untuk menentukan solusi yang lebih baik dari kondisi saat ini. Iterasi dalam proses perbaikan dari algoritma ini melakukan pencarian terhadap kondisi saat ini dan mengevaluasi berbagai solusi yang tersedia. Terdapat langkah pemilihan dan penolakan dari solusi yang tersedia untuk mencapai solusi yang lebih baik (Pinedo, 2008).

Secara garis besar terdapat empat proses dalam algoritma *Local Search* antara lain yaitu:

1. Proses merepresentasikan kondisi saat ini
2. Proses *neighbourhood design*
3. Proses pencarian semua solusi yang ditemukan
4. Proses kriteria penerimaan dan penolakan

Merepresentasikan suatu jadwal bukan suatu hal yang mudah, berbeda jika jika permasalahannya terkait dengan penjadwalan mesin tunggal yang bisa dilakukan dengan melakukan permutasi dari sejumlah  $n$  operasi. Terkait penjadwalan *Job Shop*  $m$  mesin dapat dirangkai berturut-turut yang dimana masing-masing mewakili permutasi dari  $n$  operasi pada mesin tersebut. Berdasarkan informasi tersebut waktu mulai dan waktu semua operasi (*makespan*) dapat dihitung.

*Neighbourhood design* merupakan aspek penting dalam algoritma *Local Search*. Contoh pada permasalahan penjadwalan *Job Shop* adalah penggunaan konsep jalur kritis. Jalur kritis dalam penjadwalan *Job Shop* terdiri dari serangkaian operasi yang mana pertama dimulai pada waktu  $t = 0$  dan selesai pada waktu  $t = C maks$ . Waktu penyelesaian setiap operasi pada jalur kritis sama dengan waktu mulai operasi berikutnya pada jalur ini. Proses pengoptimalan *makespan* dalam kasus ini dalam pengurutan operasi pada jalur kritis. Proses sederhana dari proses ini dimulai dari mengatur jadwal urutan operasi yang sesuai pada mesin yang dapat diperoleh dengan menukar pasangan operasi yang berdekatan pada jalur kritis.

Kriteria penerimaan-penolakan dalam algoritma *Local Search* merupakan prosedur yang membedakan dengan algoritma lainnya. Perbedaan ini dijelaskan pada penggunaan algoritma *Simulated annealing* dan algoritma *Tabu search*. Penggunaan kriteria penerimaan-penolakan dalam *Simulated annealing* didasarkan pada proses probabilistik sedangkan dalam *Tabu search* didasarkan pada proses deterministik.

Langkah-langkah iterasi dan formulasi dasar yang digunakan dalam proses *Local Search* ini yaitu:

```

 $x \leftarrow x_0$ 
while stop criteria not met do
  1. Find neighborhood  $\mathcal{N}_x$ 
  2. Find "best" solution in  $\mathcal{N}_x$ :  $x_{\text{best}}$ 
  3.  $x \leftarrow x_{\text{best}}$ 
end while

```

Terdapat beberapa algoritma penjadwalan yang didasari pada *Local Search*, algoritma tersebut antara lain (Pinedo, 2008):

1. *Tabu Search*
2. *Simulated Annealing*
3. Algoritma Genetik

### 2.3.3 Kelebihan dan Kekurangan Algoritma *Local Search*

Penggunaan Algoritma *Local Search* memberikan kelebihan serta kekurangan dalam penerapannya, diantaranya sebagai berikut: (Pinedo, 2008)

#### A. Kelebihan

1. Memberikan *improvement* dari penjadwalan saat ini
2. Proses perancangan relatif cepat dan lebih sederhana jika dibandingkan dengan metode analitik atau optimal
3. Memiliki fleksibilitas relatif tinggi ketika terjadi perubahan proses atau mesin dalam penjadwalan

#### B. Kekurangan

1. Tidak memberikan hasil yang optimal
2. Penggunaan iterasi atau *neighbourhood design* dalam perolehan hasil akhir tidak dijelaskan secara rinci

### 2.4 Peta Gantt (*Gantt Chart*)

Peta *gantt* merupakan *representasi* grafis dari pekerjaan-pekerjaan yang harus diselesaikan dan digambarkan dalam bentuk batang dan analog dengan waktu penyelesaian pekerjaan tersebut. Peta ini dirancang pada saat perang dunia ke-1 oleh Heary L. Gantt. Secara garis besar keuntungan dari peta *Gantt* adalah:

1. Semua pekerjaan diperlihatkan secara grafis dalam satu peta yang mudah dipahami.
2. Kemajuan pekerjaan mudah diamati dan diperiksa setiap waktu karena sudah tergambar dengan jelas.
3. Dalam situasi keterbatasan sumber penggunaan peta *gantt* memungkinkan evaluasi yang lebih awal mengenai penggunaan sumber seperti yang telah direncanakan.

### **2.5 Software Lakin**

Lakin adalah sistem penjadwalan yang dikembangkan di Stern *School of Business*, NYU. Lakin diprogram oleh mahasiswa Universitas Columbia. Lakin dibuat dengan tujuan untuk memperkenalkan siswa dalam memahami teori-teori penjadwalan dan pengaplikasiannya serta mendorong dalam penggunaan algoritma dalam penjadwalan. Aplikasi ini ini dirancang oleh Profesor Michael L. Pinedo dan Profesor Xiuli Chao. Aplikasi ini dapat menyelesaikan beberapa permasalahan penjadwalan diantaranya, yaitu:

1. *Single Machine*
2. *Parallel Machines*
3. *Flow Shop*
4. *Job Shop*
5. *Flexible Flow Shop*
6. *Flexible Job Shop*

Pengoptimalan dalam aplikasi ini dilengkapi dengan beberapa aturan prioritas dan algoritma yaitu diantaranya:

1. *ATCS (Apparent Tardiness Cost with Setups)*
2. *EDD (Earliest Due Date)*
3. *MS (Minimum Slack)*
4. *FCFS (First Come First Served)*
5. *SPT (Shortest Processing Time)*
6. *LPT (Longest Processing Time)*
7. *WSPT (Weighted Shortest Processing Time)*
8. *CR (Critical Ratio)*

9. *General SB Routine*
10. *Shifting Bottleneck / Sum(wT)*
11. *Shifting Bottleneck / Tmax*
12. *Local Search*

