Bab 2
Landasan Teori

2.1. Pengertian dan Tujuan Penjadwalan

Secara umum masalah penjadwalan menurut **French** (1982) dapat dijelaskan sebagai berikut. Jika ada *n job* $\{J_1, J_2, ..., J_n\}$ harus diproses pada *m mesin* $\{M_1, M_2, ..., M_m\}$. Proses pengerjaan *job* J_1 disebut dengan operasi, O_{ij}. Waktu yang diperlukan untuk memproses operasi O_{ij} pada mesin M_j adalah t_{ij}. Beberapa *job* mungkin memiliki saat pengerjaan paling awal atau saat kedatangan *job* ke *shop* yang disebut *release date*, r_{ij} yang mungkin tidak sama dengan 0, dan juga batas penyelesaian yang disebut *due date*, d_{ij}. Permasalahan penjadwalan adalah menentukan urutan produksi yang memberikan solusi terbaik dengan kriteria sebagai berikut:

- Memenuhi *technological constraint* yang ada, dengan kata lain merupakan jadwal yang *feasibel*.
- Memenuhi beberapa kriteria pengukuran performansi, seperti minimasi *makespan*, minimasi banyaknya *job* yang terlambat dan sebagainya.

Penjadwalan produksi memiliki berbagai definisi antara lain sebagai berikut:

- Menurut **Morton dan Pentico** (1993) penjadwalan yaitu proses pengorganisasian, pemilihan dan pemberian waktu dalam penggunaan sumber daya untuk melaksanakan aktivitas yang diperlukan dalam menghasilkan *output* yang diinginkan, dengan memenuhi waktu yang ditetapkan dan kendala-kendala hubungan antara waktu dan aktivitas.
- Menurut **Conway *et al*** (1967) penjadwalan merupakan proses pengurutan pembuatan produk secara menyeluruh pada sejumlah mesin tertentu dan pengurutan didefinisikan sebagai proses pembuatan produk pada satu mesin tertentu.
- Menurut **Baker** (1974) penjadwalan yaitu proses pengalokasian sumber-sumber untuk memilih sekumpulan tugas dalam jangka waktu tertentu.

Definisi yang diberikan **Baker** (1974) mengandung dua arti, yaitu:

1. Penjadwalan merupakan fungsi pengambilan keputusan yaitu menentukan jadwal (nilai praktis).
2. Penjadwalan merupakan suatu teori, yaitu sekumpulan prinsip-prinsip dasar, model-model, teknik-teknik dan kesimpulan-kesimpulan logis dalam proses pengambilan keputusan yang memberikan pengertian dalam fungsi penjadwalan (nilai konseptual).

Keputusan yang dibuat dalam penjadwalan meliputi:

a. Pengurutan pekerjaan (*sequencing*)
b. Waktu mulai dan selesai pekerjaan (*timing*)
c. Urutan proses suatu pekerjaan (*routing*)

Persoalan penjadwalan timbul apabila beberapa pekerjaan akan dikerjakan secara bersamaan, sedangkan sumber yang dimiliki terbatas. *Input* dari suatu penjadwalan mencakup jenis dan banyaknya *part* yang akan dioperasi, urutan ketergantungan antar operasi, waktu proses untuk masing-masing operasi, serta fasilitas yang dibutuhkan oleh setiap operasi. Sedangkan *output* dari penjadwalan meliputi *dispatch list*, yaitu daftar yang menyatakan urutan pemrosesan *part* serta waktu mulai dan selesai dari pemrosesan *part* (*starting and completion time*).

Menurut **Baker** (1974), tujuan penjadwalan umumnya adalah sebagai berikut:

- Meningkatkan produktivitas mesin, yaitu dengan mengurangi waktu mesin menganggur.
- Mengurangi persediaan barang setengah jadi (*work in proses inventory*) dengan jalan mengurangi jumlah rata-rata pekerjaan yang menunggu dalam antrian suatu mesin karena mesin tersebut sibuk.
- Mengurangi keterlambatan suatu pekerjaan. Setiap pekerjaan mempunyai batas waktu (*due date*) penyelesaian, jika pekerjaan tersebut diselesaikan melewati batas waktu yang ditentukan maka pekerjaan tersebut dinyatakan terlambat. Dengan metoda penjadwalan maka keterlambatan ini dapat dikurangi, baik waktu maupun frekuensi.

Menurut **Narasimhan *et al*** (1985) penjadwalan yang baik seharusnya *simple*, mudah dimengerti dan dapat dilaksanakn oleh pihak manajemen dan oleh siapapun yang menggunakannya. Aturan-aturan penjadwalan seharusnya cukup kuat tetapi mempunyai tujuan yang realistis sehingga cukup *flexible* untuk memecahkan masalah yang tidak terprediksi sebelumnya dan membolehkan suatu perencanaan ulang.

2.2. Elemen Penjadwalan

Penjadwalan mempunyai elemen-elemen penting yang harus diperhatikan seperti *job*, operasi, mesin serta hubungan yang terjadi diantaranya:

1. *Job*

   Job dapat didefinisikan sebagai suatu pekerjaan yang harus diselesaikan untuk mendapatkan suatu produk. *Job* biasanya terdiri dari beberapa operasi yang harus dikerjakan (minimal 1 operasi). Manajemen melalui perencanaan yang telah dibuat/berdasarkan pesanan dari pelanggan memberikan *job* kepada lantai kerja pabrik untuk dikerjakan. Informasi yang dipunyai oleh suatu *job* ketika datang ke lantai kerja pabrik biasanya adalah operasi-operasi yang harus dilakukan di dalamnya (dari bagian *engineering*) saat *job* harus diselesaikan dan pada saat *job* mulai dapat dikerjakan.

2. Operasi

   Operasi adalah himpunan bagian dari *job*. Untuk menyelesaikan suatu *job*, operasi dalam *job* diurutkan dalam suatu urutan pengerjaan tertentu. Urutan tersebut ditentukan pada saat perencanaan proses. Suatu operasi baru dapat dikerjakan apabila operasi/proses yang mendahuluinya sudah dikerjakan terlebih dahulu. Matriks *routing* memberikan informasi mengenai urutan pengerjaan dan jenis mesin yang digunakan dalam setiap operasi. Setiap operasi mempunyai waktu proses.

3. Mesin

   Mesin adalah sumber daya yang diperlukan untuk pengerjaan proses penyelesaian suatu *job*. Setiap mesin hanya dapat memproses satu tugas pada saat tertentu.

2.3. Klasifikasi Masalah Penjadwalan

Masalah penjadwalan dapat diklasifikasikan berdasarkan faktor-faktornya yaitu:

1. Berdasarkan jumlah mesin yang digunakan menjadi dua bagian:
   - Penjadwalan pada mesin tunggal
   - Penjadwalan pada mesin ganda
2. Berdasarkan pola kedatangan *job*:
   - Pola kedatangan statis yaitu sejumlah pekerjaan datang secara bersamaan pada *shop* yang menganggur dan langsung siap untuk dikerjakan. Selanjutnya jika ada pekerjaan baru yang datang, maka penanganan penjadwalan diutamakan dulu pada kumpulan pekerjaan yang pertama.
   - Pola kedatangan dinamis yaitu *shop* bekerja secara terus menerus (kontinu) serta sifat kedatangan *job* tidak menentu dan diramalkan dengan statistik.
3. Berdasarkan sifat informasi yang diterima:

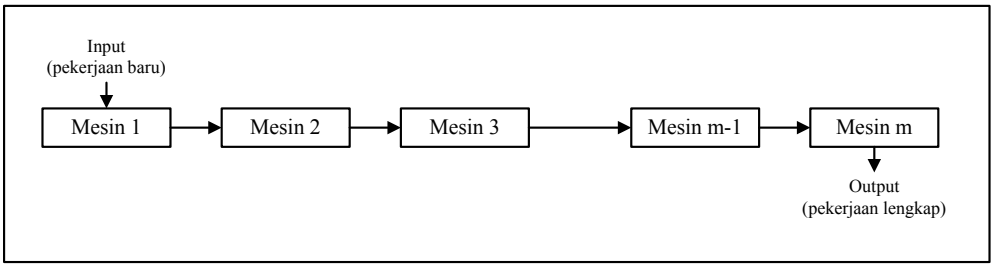
   Perilaku elemen-elemen penjadwalan dibedakan atas deterministik dan stokastik. Model deterministik dikenal dengan adanya kepastian informasi tentang elemen-elemen yang ada. Model stokastik mengandung ketidakpastian, sehingga informasi dapat diramalkan dengan statistik. Elemen-elemen yang dimaksud adalah:
   - Karakteristik pekerjaan dari segi kedatangan, batas waktu penyelesaian dan perbedaan kepentingan pekerjaan-pekerjaan tersebut.
   - Karakteristik pekerjaan dari segi banyaknya operasi. Susunan mesin, waktu operasi dan lain-lain.
   - Karakteristik mesin dari segi jumlah dan kapasitas mesin, kemampuan serta kecocokan tiap msin dengan tugas yang diberikan.
4. Berdasarkan aliran proses:
   - Pola aliran *flow shop*

     Flow shop ditandai dengan semua pekerjaan mengikuti lintas yang sama dari satu mesin ke masin yang lain melalui urutan mesin yang sama. Jika mesin diberi nomor, maka mesin untuk operasi B akan lebih besar daripada nomor mesin untuk operasi A pada pekerjaan yang sama jika A mendahului B. Pola *flow shop* seperti ini dikenal sebagai *pure flow shop*

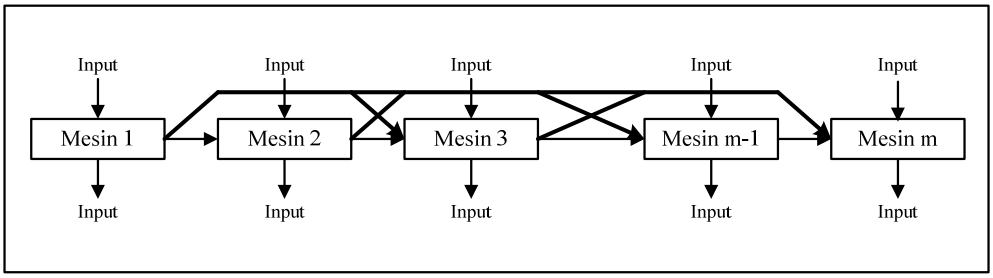
(gambar 2.1), ada pula yang dikenal sebagai *general flow shop* (gambar 2.2), yaitu pekerjaan mengikuti lintasan yang sama tetapi urutan mesin berbeda ini disebabkan oleh suatu *shop* dapat menangani pekerjaan yang bervariasi sedangkan pekerjaan yang datang ke *shop* tidak harus dikerjakan di semua mesin.

- Pola aliran *job shop*

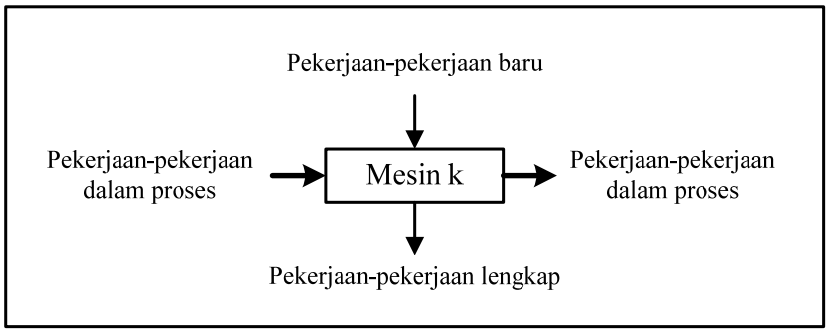
  Pada pola *job shop* (gambar 2.3), aliran proses dalam *shop* tidak mempunyai pola yang umum. Nomor mesin yang sama mungkin akan muncul dua kali atau lebih pada operasi-operasi suatu pekerjaan. Perbedaan pola aliran tersebut dapat dilihat pada gambar aliran pekerjaan dibawah ini:



Gambar 2.1 Pola aliran kerja *pure flow shop* (sumber: **Baker** [1974])



Gambar 2.2 Pola aliran kerja *general flow shop* (sumber: **Baker** [1974])



Gambar 2.3 Pola aliran kerja *job shop* (sumber: **Baker** [1974])

2.4. Beberapa Istilah Dalam Penjadwalan

Dalam pembahasan masalah penjadwalan, sering dijumpai beberapa istilah yang umum digunakan, antara lain:

- Waktu Proses (t_i)

  Waktu yang diperlukan untuk menyelesaikan suatu pekerjan ke-*i*. Waktu proses ini telah mencakup waktu untuk persiapan dan pengaturan proses.

- *Ready Time* (r_i)

  Waktu saat pengerjaan ke-*i* dapat dikerjakan (siap dijadwalkan).

- *Due Date* (d_i)

  Batas waktu akhir suatu pekerjaan ke-*i* boleh diselesaikan. Lewat batas ini suatu pekerjaan dinyatakan terlambat (*tardy*).

- *Makespan* (*MS*)

  Waktu yang dibutuhkan untuk menyelesaikan seluruh pekerjaan, mulai dari operasi pertama *job* urutan pertama sampai operasi terakhir *job* urutan terakhir. Untuk kasus satu mesin *makespan* merupakan penjumlahan waktu proses dari seluruh pekerjaan.

- *Completion Time*/Waktu Penyelesaian (C_i)

  Rentang waktu sejak waktu pertama operasi *job* ke-*i* dimulai hingga *job* ke-*i* diselesaikan.

- *Lateness* (L_i)

  Penyimpangan dari waktu penyelesaian hingga saat *due date*.

  Bila:

  $$L_i = C_i - d_i$$

  $L_i = C_i - d_i < 0$, saat penyelesaian memenuhi batas akhir (*early job*)

  $L_i = C_i - d_i > 0$, saat penyelesaian melampui batas akhir (*tardy job*)

- *Tardiness* (T_i)

  Keterlambatan penyelesaian suatu pekerjaan dari saat *due date*.

  $$F = \max_{1 \le i \le n} \{0, L_i\}$$

- *Slack Time* (S_i)

  Waktu sisa yang tersedia bagi suatu pekerjaan

  $$S_i = d_i - t_i$$

- *Flow Time* (F_i)

  Rentang waktu mulai dari pekerjaan ke-*i* siap dikerjakan hingga pekerjaan diselesaikan.

2.5. Kriteria Evaluasi Jadwal

Keberhasilan suatu penjadwalan dapat diukur dengan besaran-besaran yang melibatkan informasi dari suatu pekerjaan yang dijadwalkan. Hal ini dapat tercapai dengan menentukan kriteria jadwal yang diinginkan. Penentuan kriteria jadwal dapat didasarkan atas karakteristik pekerjaan dan peralatannya, seperti yang dikemukakan oleh **Conway *et al*** (1967), sebagai berikut:

- Kriteria berdasarkan atribut tugas

  Dilihat dalam hubungan dengan tugas yang ada, misalnya: minimasi *flow time*, minimasi *lateness*, minimasi jumlah *job* yang terlambat (*tardy*)

- Kriteria berdasarkan atribut *shop*

  Dilihat dalam hubungannya dengan *shop* (alat/mesin), seperti: maksimasi utilitas *shop*, minimasi waktu *set up* mesin, penyeimbangan waktu beban mesin

Jika terdapat *n* pekerjaan yang akan dijadwalkan, maka tingkat keberhasilan dapat dinilai dari besaran-besaran berikut:

- *Makespan* (maksimum *completion time*)

$$MS = max\ (C_i)$$

- *Mean flow time (*waktu penyelesaian rata-rata)

$$\overline{F} = \frac{1}{n}\sum_{i=1}^{n} F_i$$

- *Mean lateness* (rata-rata keterlambatan)

$$\overline{L} = \frac{1}{n}\sum_{i=1}^{n} L_i$$

- *Mean tardiness* (rata-rata keterlambatan non-negatif)

$$\overline{T} = \frac{1}{n}\sum_{i=1}^{n} T_i$$

- Maksimum *flow time* (maksimum waktu penyelesaian seluruh pekerjaan)

$$F_{max} = \max_{1 \le i \le n} \{F_i\}$$

- Maksimum *lateness* (maksimum keterlambatan)

$$L_{max} = \max_{1 \le i \le n} \{L_i\}$$

- Maksimum *tardiness* (maksimum keterlambatan non-negatif

$$T_{max} = \max_{1 \le i \le n} \{T_i\}$$

- *Mean shop utility* (rata-rata utilitas *shop*)

$$\overline{U} = \frac{\sum_{i=1}^{n} P_i}{m.F_{\max}}$$

- *Number of tardy job* (jumlah pekerjaan yang terlambat)

$$N = \sum_{i=1}^{n} \delta_i$$

Dimana:

$\delta_i = 1$, bila $T_i > 0$

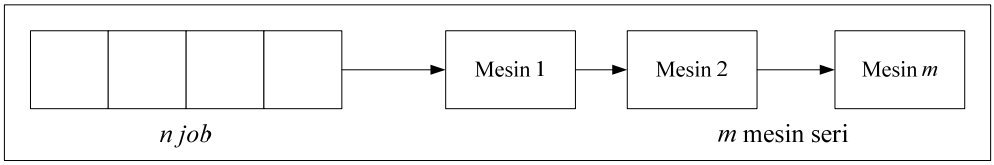
$\delta_i = 0$, bila $T_i = 0$

- *Total weigthed tardines* (*tardiness* tertimbang total)

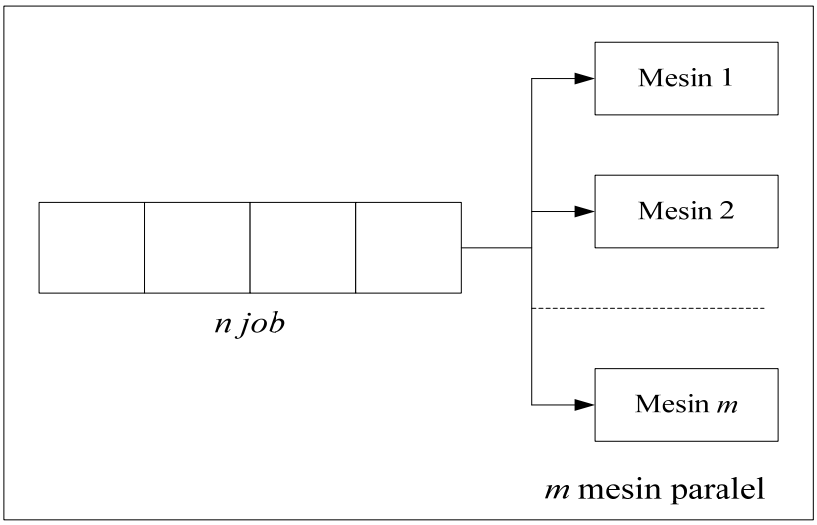
$$T_w = \sum_{i=1}^{n} W_i.T_i$$

2.6. Penjadwalan *n Job* Pada *m* Mesin

Pada model pertama klasifikasi masalah penjadwalan yang dibahas pada sub-bab sebelumnya, terdapat penjadwalan untuk mesin ganda atau penjadwalan *n job* pada *m* mesin. Model ini terbagi lagi untuk mesin seri dan mesin paralel. Untuk mesin seri, tiap pekerjaan harus dikerjakan pada beberapa mesin secara berurutan (gambar 2.4), sedangkan untuk mesin paralel, tiap pekerjaan hanya dikerjakan pada satu mesin (gambar 2.5), dengan pekerjaan yang terlibat bersifat *independent*.



Gambar 2.4 *n job* pada *m* mesin seri



Gambar 2.5 *n job* pada *m* mesin paralel

2.7. Penjadwalan Dalam Sistem Produksi *Job Shop*

Penjadwalan mempunyai metode yang berbeda-beda untuk setiap tipe sistem produksi karena setiap sistem mempunyai karakteristik yang berbeda satu dengan yang lainnya. Demikian pula dengan sistem produksi *job shop*. Ciri khas persoalan *job shop* adalah aliran pekerjaan dalam *shop* tidak searah (*non unidirectional*). Waktu proses dan *routing* dari sejumlah *job* yang akan dijadwalkan ke dalam suatu tabel matriks yang disebut matriks waktu proses dan matriks *routing*, kemudian hasil penjadwalan digambarkan dalam *Gantt Chart* (Peta Gantt).

Dalam permasalahan *job shop*, *n job* harus diproses pada *m* mesin dengan asumsi-asumsi yang ditentukan. Setiap *job* mempunyai beberapa operasi yang harus diproses di mesin yang berbeda. Oleh karena itu diperlukan pengurutan (*sequencing*) dari tiap operasi pada masing-masing *job*, agar diperoleh performansi yang baik. Kriteria performansi yang baik sangat bergantung pada tujuan dan kebijakan manajemen. Sehingga dalam penjadwalan *job shop*

diperlukan *input* berupa jumlah *job*, jumlah operasi dalam tiap *job* dan urutan proses beserta mesin yang memprosesnya (*routing*). Waktu proses adalah waktu yang diperlukan dalam mengerjakan suatu *job* sedangkan *routing* adalah urutan tipe mesin yang diperlukan untuk mengerjakan suatu *job*, kemudian *job* yang akan dijadwal ditabulasikan ke dalam tabel matriks waktu proses dan waktu *routing*. Pada matriks waktu proses, operasi 1 dari *job* 1 dikerjakan di mesin 1. Untuk menulis kondisi tersebut sering disebut notasi O_{ij} untuk merepresentasikan proses operasi dari *job i* pada operasi *j* dan mesin *k* dan t_{ijk} untuk merepresentasikan waktu proses dari *job i* pada operasi *j* dalam mesin *k*.

Dalam persoalan pola aliran proses, waktu proses *job shop* sama dengan waktu proses *job* lain (*flow shop, general flow shop*) dari sejumlah *n job*, tetapi *routing job shop* tidak sama dengan *routing job* lain dari sejumlah *m* mesin yang dijadwalkan. Hal inilah yang membedakan persoalan *job shop* dengan *flow shop*, dimana *flow shop* mempunyai *routing* yang sama untuk sejumlah *m* mesin, seperti contoh tabel di bawah ini:

Tabel 2.1 Matriks waktu proses *job shop*

	Operasi		
	1	2	3
Job 1	4	3	2
Job 2	1	4	4
Job 3	3	2	3
Job 4	3	3	1

Tabel 2.2 Matriks waktu proses *flow shop*

	Operasi		
	1	2	3
Job 1	4	3	2
Job 2	1	4	4
Job 3	3	2	3
Job 4	3	3	1

Tabel 2.3 Matriks *routing job shop*

	Operasi		
	1	2	3
Job 1	1	2	3
Job 2	2	1	3
Job 3	3	2	1
Job 4	2	3	1

Tabel 2.4 Matriks *routing flow shop*

	Operasi		
	1	2	3
Job 1	1	2	3
Job 2	1	2	3
Job 3	1	2	3
Job 4	1	2	3

Setelah input dari masing-masing *job* telah didefinisikan, proses selanjutnya dalam penugasan operasi dari *job* dalam setiap mesin. Penugasan ini mempunyai aturan yang bermacam-macam dan penggunaannya biasanya ditentukan oleh kebijaksanaan manajemen dan berdasarkan produksinya serta kriteria yang diinginkan. Sebuah jadwal yang layak adalah kumpulan dari hubungan keterkaitan (*procedence relation*), yang memberikan urutan proses yang lengkap pada setiap mesin. Hal tersebut harus memenuhi tiga kondisi berikut:

a) *Routing* tiap *job* di ikuti
b) Setiap mesin hanya memproduksi satu *job* pada suatu waktu, dan pemrosesan suatu *job* dapat diinterupsi
c) Waktu proses dari tiap operasi telah ditentukan dan pemrosesan suatu operasi tidak dapat diinterupsi.

Hasil penjadwalan yang layak (*feasible*) dapat dibedakan menjadi 4 jenis, yaitu:

1. Jadwal Semi Aktif

   Merupakan set jadwal yang tidak ada kemungkinan untuk memindahkan suatu operasi ke waktu yang lebih kecil tanpa mengubah urutan operasi secara keseluruhan.
2. Jadwal Aktif

   Merupakan set jadwal yang tidak satu operasi pun dapat dipindahkan lebih awal tanpa menunda operasi lain.
3. Jadwal *Non Delay*

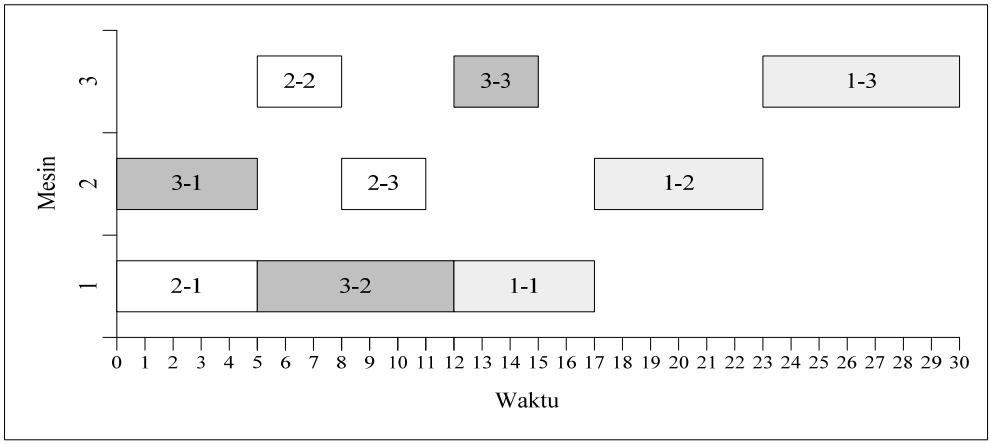
   Merupakan set jadwal yang tidak membiarkan mesin menganggur jika pada saat yang sama terdapat operasi yang membutuhkan mesin tersebut.
4. Set Jadwal Optimal

   Kumpulan jadwal yang tidak terdapat jadwal lain yang memiliki tingkat preferensi lebih tinggi dari set jadwal optimal.

Peta Gantt (*Gantt Chart*) dikembangkan oleh Henri L. Gantt semasa perang dunia I. Peta Gantt merupakan representasi grafis dari pekerjaan-pekerjaan yang harus diselesaikan, digambarkan dalam bentuk susunan blok-blok batang yang analog dengan waktu penyelesaian pekerjaan-pekerjaan tersebut.

Keuntungan dari peta Gantt adalah:

- Semua pekerjaan diperlihatkan secara grafis dalam suatu peta yang mudah dipahami.
- Kemajuan pekerjaan mudah diamati/diperiksa pada setiap waktu karena sudah tergambar dengan jelas.
- Dalam situasi keterbatasan sumber, penggunaan peta Gantt memungkinkan evaluasi lebih awal mengenai penggunaan sumber seperti yang telah direncanakan.



Gambar 2.6 Contoh peta Gantt 3 *job* dengan 3 mesin

2.7.1. Karakteristik dan Asumsi dalam Penjadwalan *Job Shop*

Baker (1974), mengemukakan beberapa karakteristik dan asumsi penjadwalan dari sistem produksi *job shop*, karakteristik dan asumsi tipe ini akan digunakan untuk keperluan simulasi dari *job shop* yang dinamis, sebagai berikut:

1. *Job* terdiri dari urutan operasi yang telah ditentukan.
2. Suatu operasi hanya bisa dikerjakan pada satu tipe mesin.
3. Hanya ada satu mesin dari tiap tipe mesin dalam *shop*.
4. Waktu proses diketahui dengan pasti seperti halnya *due date*.
5. Urutan waktu *set up* bersifat *independent* dan waktu transportasi antar mesin dapat diabaikan.
6. Operasi yang sedang dikerjakan pada suatu mesin tidak dapat diinterupsi.
7. Suatu operasi tidak dapat dimulai sampai operasi pendahulunya diselesaikan.
8. Setiap mesin hanya dapat memproses satu operasi pada suatu waktu.
9. Setiap *part* hanya dapat diproses disuatu mesin pada suatu waktu.

- Asumsi mengenai pekerjaan:
  a. Setiap pekerjaan diselesaikan menurut jadwal yang telah disusun.
  b. Setiap pekerjaan merupakan satu kesatuan.
  c. Setiap pekerjaan yang telah dimulai harus diselesaikan tanpa menyisipkan pekerjaan lain diantaranya.
  d. Setiap pekerjaan tidak boleh diproses pada lebih dari satu mesin pada saat yang bersamaan.
  e. Setiap pekerjaan hanya mempunyai satu operasi dan dapat dikerjakan pada mesin yang mana saja.
- Karakteristik Mesin
  a. Setiap mesin dapat mengerjakan pekerjaan manapun yang dibebankan kepadanya.
  b. Setiap mesin, secara kontinu, siap untuk dibebankan pekerjaan selama perioda penjadwalan, tanpa mengalami interupsi oleh kerusakan/pemeliharaan mesin.
  c. Setiap mesin hanya dapat melakukan satu operasi dari suatu pekerjaan (*job*) dalam satu waktu tertentu.
  d. Operator yang menangani mesin selalu tersedia.

2.7.2. Teknik-teknik Penjadwalan *Job Shop*

Dalam memecahkan masalah penjadwalan dalam sistem produksi yang bersifat *job shop* dapat digunakan beberapa teknik. Teknik-teknik dalam melakukan penjadwalan dapat dibedakan dalam dua bentuk, yaitu:

- Teknik Pendekatan Optimal
  1. Teknik *integer programing*.
  2. Teknik *branch and bound*.
- Teknik Pendekatan *Heuristik*
  1. *Priority dispatching*
  2. *Sampling*
  3. *Probabilistik dispatching*

2.8. Aturan Prioritas Dalam Penjadwalan *Job Shop*

Penjadwalan dengan pendekatan *heuristik* menggunakan aturan pengurutan atau *priority dispatching*, pada *priority dispatching* ditentukan aturan prioritasnya untuk memilih satu operasi diantara operasi-operasi yang mengalami gangguan. Penetuan prioritas bergantung pada tujuan yang ingin dicapai. Beberapa aturan prioritas diberikan di bawah ini menurut **Fogarty** (1991), adalah

- *Fewest Operation* (FO) :

  Prioritas teringgi diberikan pada produk dengan sisa operasi paling sedikit. Aturan ini biasanya mengurangi WIP, waktu penyelesaian produk, dan rata-rata waktu keterlambatan.

- *Shortest Processing Time* (SPT) :

  Prioritas tertinggi pada operasi dengan waktu proses terpendek, aturan ini biasanya meminimasi WIP, rata-rata keterlambatan (*mean lateness*) dan waktu penyelesaian rata-rata (*mean flow time*) produk.

- *Shortest Total Processing Time* (STPT)

  Prioritas tertinggi pada operasi dengan total waktu proses terpendek.

- *Fisrt Come Fisrt Serve* (FCFS)

  Prioritas tertinggi diberikan pada operasi yang masuk St (stasiun ke-t) lebih dulu. Aturan ini cocok untuk tipe organisasi dimana konsumen mementingkan waktu pelayanan.

- *Earliest Due Date* (EDD)

  Prioritas tertinggi diberikan pada produk dengan *due date* (batas waktu penyelesaian) terpendek. Aturan ini berjalan dengan baik bila waktu proses *job-job* relatif sama.

- *Slack time* (ST)

  Prioritas tertinggi diberikan pada produk dengan *slack time* lebih kecil.

  ST (*slack time*) = DD (*due date*)- Pd (*present date*) – TORT (*total operation time remaining*).

- *Critical Ratio* (CR)

  Prioritas tertinggi diberikan pada produk dengan CR paling kecil.

  CR = {d_j (tanggal penyerahan) – t (tanggal saat ini)}/{(t_j+t)(waktu yang diperlukan untuk penyelesaian operasi)}.

2.9. Klasifikasi Sistem Manufaktur

Secara umum pembagian klasifikasi sistem manufaktur menurut **Fogarty *et al*** (1991) adalah sebagai berikut:

a. *Make To Stock* : mengubah komponen tingkat rendah dan bahan mentah keseluruhan menjadi produk akhir untuk mengantisipasi *order* pelanggan.
b. *Assemble to order* : mengubah komponen tingkat rendah dan bahan mentah menjadi level manufaktur tertentu dan membentuk *order* pelanggan bila menerima pesanan.
c. *Make to order* : sangat sedikit atau sama sekali tidak memiliki material tingkat rendah hingga pesanan pelanggan diterima.
d. *Engineer to order* : sangat sedikit mengetahui tentang apa yang akan diproduksi hingga pesanan diterima dan membuat spesifikasi *engineering*nya.

Dalam dua situasi produksi terakhir, pesanan pelanggan memainkan peranan terpenting dalam sistem produksi dan sistem kontrol produksi seluruh aktivitas dikendalikan oleh pesanan pelanggan.

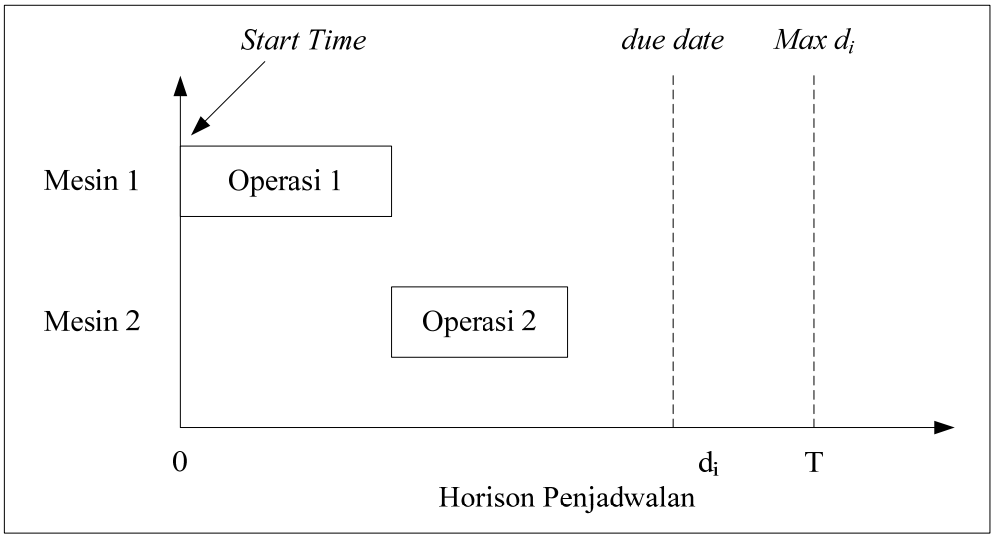
2.10. Metode-metode Penjadwalan

Proses perancangan algoritma penjadwalan dengan memperhatikan aktivitas perawatan pencegah merupakan pengembangan dari rancangan algoritma penjadwalan yang sudah ada. Beberapa metode penjadwalan *heuristik* yang telah dikenal antara lain:

1. Metode *Forward*

   Metode ini menjadwalkan proses kerja dalam setiap sumber daya mulai sejak awal produksi (saat $t = 0$) sampai dapat diselesaikannya keseluruhan produk yang direncanakan. Tujuan dari metode ini adalah menjadwalkan produksi apabila titik waktu mulainya telah ditentukan menjadwalkan sebelumnya dan tidak diberikan batas waktu harus diselesaikannya keseluruhan produk tersebut.

Kelebihan metode ini adalah bahwa dalam penjadwalan dapat disusun secara *short processing time* (SPT) sehingga didapat suatu penjadwalan produksi dengan *flow time* yang minimum. Sedangkan kelemahannya adalah adanya kemungkinan untuk melewati batas waktu (*due date*) yang ditetapkan oleh konsumen. Metode ini menggunakan algoritma dimana pada saat *job* datang maka langsung dijadwalkan pada mesin yang tersedia, sedangkan untuk *job* selanjutnya apabila menggunakan mesin yang sama maka dilakukan penjadwalan dengan memperhitungkan waktu selesai dari *job* sebelumnya.

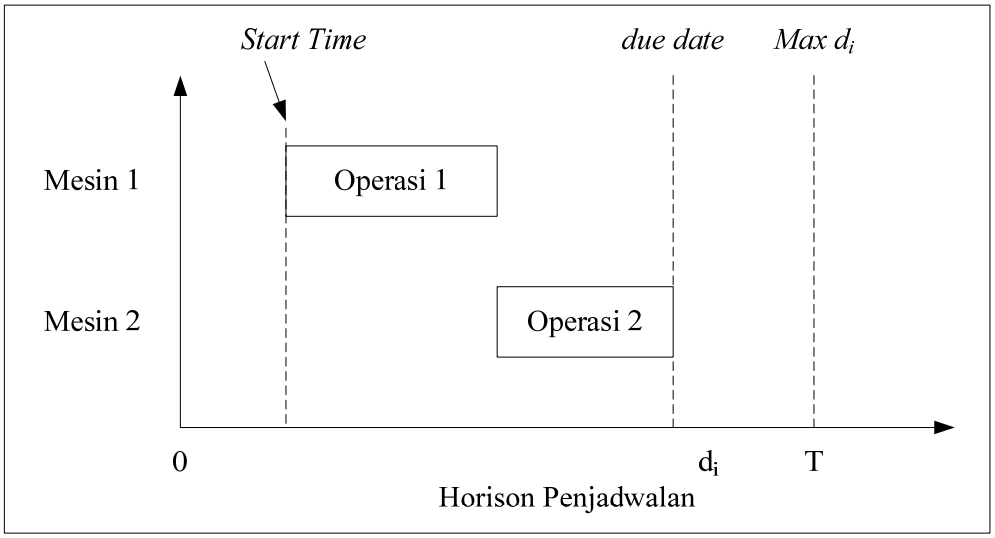


Gambar 2.7 Peta Gantt untuk metode *forward*

2. Metode *Backward*

Pada penjadwalan dengan metode *backward*, operasi yang paling akhir pada *routing* dijadwalkan terlebih dahulu. Kemudian sisa operasi diseimbangkan menurut *order* (pesanan) dengan pendekatan mundur dalam perhitungan satuan waktu tersedia, saat (tanggal) dimulainya penjadwalan dapat ditentukan. Pada metode *backward* menggunakan pendekatan mundur dan waktu mulai tidak harus berada di titik nol (*start time*) dan dalam penyusunan jadwal bergerak mundur dari *due date*. Metode ini menjadwalkan produksi mulai dari batas akhir diselesaikannya keseluruhan produk (*due date*) kemudian mundur terus ke belakang sampai didapatkan waktu mulainya produksi (didapatkan $t = 0$).

Kelebihan dari metode ini adalah bahwa hasil penjadwalan dengan metode ini tidak akan terlambat, karena dijadwalkan mundur atau dengan kata lain bahwa *due date* selalu terpenuhi. Sedangkan kelemahan dari metode ini tidak bisa diterapkan pada penjadwalan *n job m* mesin jika ada beberapa *job* yang mempunyai *due date* yang sama. Kelemahan lain dari penerapan metode ini bahwa penjadwalan tidak dapat mendeteksi adanya sumber daya yang menganggur sehingga utilitas sumber daya yang ada tidak dapat maksimum. Selain itu kemungkinan menghasilkan penjadwalan yang tidak *fleksibel* yaitu waktu mulai proses pertama kali sudah terlewat dari waktu sekarang (waktu saat menjadwalkan).



Gambar 2.8. Peta Gantt untuk metode *backward*

2.11. *Rescheduling* dan Faktor-faktornya

Berikut ini akan dijelaskan mengenai teori yang berkaitan dengan penjadwalan ulang (*rescheduling*) yang terdiri dari konsep *rescheduling* dan faktor-faktornya.

2.11.1. Konsep *Rescheduling*

Dalam pelaksanaan jadwal produksi di tingkat *shop floor* seringkali jadwal yang telah disusun menjadi tidak layak. Karena karakteristik sistem *job shop* yang dinamis dan pola kedatangan *job* yang stokastik serta perubahan-perubahan kondisi dalam *shop floor* yang tidak bisa diperkirakan, seringkali menyebabkan jadwal tidak mampu mengakomodasi perubahan-perubahan kondisi tersebut.

Dua hal yang berhubungan erat dengan perubahan kondisi produksi adalah *event* dan aktivitas. Dengan kata lain *event* adalah perubahan kondisi dalam produksi. Sedangkan aktivitas akan selalu terjadi dua *event*, masing-masing adalah *event* saat dimulainya aktivitas dan *event* saat aktivitas itu berakhir. Seringkali *event* saat berakhirnya aktivitas merupakan *event* saat dimulai aktivitas berikutnya.

Pada sistem manufaktur *event* juga dapat dibedakan menjadi dua, yaitu:

1. *Event* yang dapat diperkirakan (teramal)

   Event yang dapat diperkirakan kejadiannya yang berhubungan dengan produk/benda kerja meliputi, akhir operasi pengerjaan, kedatangan *order* normal yang teramal dan sebagainya. Sedangkan yang berhubungan dengan peralatan produksi antara lain, akhir operasi (pemesinan, transportasi), instalasi peralatan baru, akhir perbaikan, perawatan dan sebagainya.
2. *Event* yang tidak bisa diramalkan (tak teramal)

   Event yang tidak teramal yang berkaitan dengan produk/benda kerja meliputi kedatangan *order* dengan prioritas tinggi atau *order* normal yang tidak teramal, perubahan *due date* dan sebagainya. Sedangkan yang berkaitan dengan peralatan produksi antara lain meliputi: kerusakan peralatan, *power failures*, keterlambatan operasi dan sebagainya.

Dalam bentuk tabel jenis-jenis *event* tersebut dapat diperlihatkan sebagai berikut:

Tabel 2.5 Jenis *event* pada sistem manufaktur

Komponen produksi	*Event-event*	
	Dapat diprediksi	Tidak dapat diprediksi
Produk/benda kerja	• Akhir operasi • Kedatangan *order* normal yang teramal	• Kedatangan *order* dengan prioritas tinggi • Kedatangan *order* normal yang tak teramal • Perubahan *due date* • Perubahan ukuran *lot* • *Delay* mesin
Peralatan produksi	• Akhir operasi (mesin, transportasi) • Instalasi peralatan baru • Perbaikan kerusakan • Perawatan	• Kerusakan • *Power failures* • *Delay* pada saat selesainya perawatan • Keterlambatan operasi • Perawatan kritis yang tak teramal

Jenis *event* yang perlu diperhatikan adalah *event* yang tidak teramal saat terjadinya. *Event-event* ini memerlukan penanganan khusus yang berbeda dengan kondisi normal. Suatu hadwal dalam sistem produksi yang *real time* harus mampu mengatasi dengan cepat perubahan kondisi yang menyebabkan jadwal awal (*initial schedule*) menjadi tidak layak. Oleh karena itu proses penjadwalan ulang atau *rescheduling* sangat diperlukan untuk mengatasi hal tersebut.

Resecheduling sendiri sebenarnya merupakan sebuah proses iteratif yang terdiri dari dua langkah:

1. Evaluasi jadwal yang sudah ada tergantung pada perubahan kondisi, permintaan dan kendala-kendala (*constrain*). Jika hasil dari revisi jadwal diterima maka proses berhenti.
2. Jika jadwal belum diterima, tentukan solusi yang lebih baik. Jadwal revisi atau jadwal yang sudah ada direvisi lagi berdasarkan solusi yang lebih baik (*improved solution*). Jika hasil ini diterima, berhenti. Jika tidak maka ulang langkah kedua sampai jadwal bisa diterima. Suatu jadwal yang bisa diterima adalah jadwal yang mampu mengatasi perubahan kondisi. Pada kenyataannya definisi sebuah jadwal yang layak adalah tergantung kebutuhan pemakai.

Selain itu *rescheduling* dapat dipandang sebagai koreksi atas jadwal yang ada. Koreksi yang diperlukan bagi suatu perubahan yang menyebabkan suatu jadwal menjadi layak dapat dibagi menajadi tiga, yaitu:

1. *Dispatch Corection* (durasi; dalam jam)

   Adapun tipe perubahan yang dapat diatasi antara lain:

   - Perubahan waktu proses aktivitas
   - Transport *delay*
   - Mesin minor lambat
   - Mesin minor macet/*startup*
   - Mesin kritis rusak/*startup*
2. *Mid Reactive Corection* (durasi; 1-2 hari)

   Faktor *rescheduling* adalah:

   - *Job rework*

- *Job scrap*
- *Job* menunggu karena bahan baku tidak ada
- Transport rusak
- Mesin rusak/*startup*
- Stasiun kerja berhenti/*startup*

3. *Major Reactive Corection*

   Faktor *rescheduling* adalah:

   - *Order* utama dihilangkan
   - Transportasi utama rusak
   - Mesin *bottleneck* rusak

2.11.2. Faktor-faktor *Rescheduling*

Faktor-faktor *rescheduling* adalah faktor-faktor yang menyebabkan suatu jadwal harus diperbaharui sesuai dengan faktor-faktor yang mempengaruhi perubahan kondisi *shop floor*. Suatu *rescheduling* yang dilakukan karena ada faktor-faktor perubahan kondisi yang tidak teramalkan dalam *shop floor* disebut "*Event Driven Rescheduling*". Selain itu *rescheduling* ada yang dilakukan secara periodik dan disebut dengan *periodic rescheduling*. Adapun faktor-faktor *rescheduling* dapat dikelompokkan menjadi dua yaitu yang berasal dari lingkungan internal *shop floor* dan yang berasal dari lingkungan luar *shop floor*.

Faktor-faktor *rescheduling* yang berasal dari lingkungan internal *shop floor* antara lain:

1. Mesin rusak
2. Bahan baku tidak ada
3. Aktivitas perawatan
4. Adanya produk yang memerlukan pengulangan operasi
5. Kasus adanya produk yang cacat

Faktor-faktor *rescheduling* eksternal adalah perubahan kondisi yang menyebabkan suatu jadwal awal menjadi tidak layak, yang berasal dari luar *shop floor*. Faktor-faktor tersebut antara lain:

1. Penambahan *order* baru
2. Adanya perubahan prioritas
3. Perubahan *due date*
4. Adanya *job* yang dibatalkan

2.12. Model Penjadwalan Dinamis Pada Sistem Produksi *Job Shop* Dengan Kriteria Minimasi Total Ongkos *Earliness* dan *Tardiness*

Model penjadwalan yang dibuat terdiri dari dua tahap, yaitu:

1. Algoritma Penjadwalan *Backward Non Delay* (Zaini *et al* [2001])

   Algoritma penjadwalan *backward non delay* digunakan untuk membuat jadwal awal yang dimulai dari: membagi horison perencanaan (interval waktu dari saat nol sampai *due date*) menjadi beberapa perioda yang ditentukan sebagai interval antara dua *due date* yang berurutan, penjadwalan dimulai dari interval terakhir.
2. Algoritma Penjadwalan *Backward Non Delay* pada Kondisi Dinamis (Zaini *et al* [2001])

   Algoritma penjadwalan *backward non delay* pada kondisi dinamis digunakan sebagai penjadwalan ulang untuk memberikan jawaban terhadap kedatangan *job* baru bila waktu penyelesaiannya sesuai dengan *due date* dan mengkonfirmasi ulang bila penyelesaian *job* baru tidak sesuai dengan *due date*.

2.12.1. Algoritma Penjadwalan *Backward Non Delay*

Untuk algoritma penjadwalan ini, notasi-notasi yang akan digunakan adalah:

P_{st} = jadwal parsial yang berisi operasi dari suatu *job* yang telah terjadwal

S_t = kumpulan *task* yang siap dijadwalkan pada tahap *t*, yaitu *task* yang tidak memiliki *task* pendahulu (*predecessors*) atau *task* yang memiliki *predecessors* yang telah terjadwalkan

$Q_{ij}(x_{ij})$ = total ongkos *earliness* dan *tardiness*

x_{ij} = saat selesai operasi ke *j job* ke *i*

α_{ij} = biaya penyimpanan *job* ke *i* operasi ke *j* per satuan waktu

β_{ij} = biaya keterlambatan (*tardiness*) *job* ke *i* per satuan waktu

σ_j = saat akhir dimana operasi $j \in S_t$ selesai dikerjakan

d = *due date*

d_h = *due date* ke h

i = indeks *job*, i=1, 2, ..., n

T_i = himpunan operasi dari *job* ke i

T_{ij} = operasi j *job* ke i

F_i = himpunan operasi terakhir dari *job* ke i

j = indeks operasi, j=1, 2, ...,F_i

H_h = interval antar dua *due date*

m = indeks mesin, m=1, 2, ...,b

Algoritma ini berisi langkah-langkah untuk menghasilkan jadwal awal.

Langkah 0 : Tahap inisiasi

- Identifikasi jumlah dan *routing job*

Langkah 1 : Urutkan *due date* dan beri indeks h, h = 1, 2, ..., v dengan h = 1 sebagai *due date* terbesar

Langkah 2 : Hitung interval antar *due date* dan nyatakan sebagai H_h, dimana $H_h = d_h - d_{h+1}$ dengan h = 1, 2, ..., v dimana d_h = *due date* ke-h

Langkah 3 : Hitung *due date* masing-masing operasi dari suatu *job* dan nyatakan sebagai d_{ij} dimana;

$d_{ij} = d_i \qquad j = F_i$

$d_{ij} = d_{ij+1} - t_{ij} \quad j = 1, 2, ..., F_i\text{-}1$

Langkah 4 : Set h = 1, t = 0 dan P_{st} = 0 dimana *st* adalah kumpulan operasi yang berada dalam H_h. (penjadwalan dilakukan secara mundur)

Langkah 5 : Tentukan σ^* dimana $\sigma^* = \max_{j \in St} \{\sigma_j\}$ dan m^* dimana σ^* dilakukan

Langkah 6 : Pilih dan tentukan operasi $j \in S_t$ dimana operasi j dilakukan di m^*.

Apakah operasi $j > 1$?

Pada saat pemilihan apakah operasi j yang berada dalam S_t dan dilakukan di m^* dimana σ^* dilakukan lebih dari satu? Jika tidak, lanjutkan ke langkah 7, jika ya, maka lakukan proses pemilihan operasi j dengan menggunakan aturan prioritas *last due date* (LDD), lanjutkan ke langkah 7.

Aturan LDD merupakan kebalikan dari EDD dimana pada LDD yang diprioritaskan adalah operasi dari suatu *job* yang memiliki *due date* terbesar.

Langkah 7 : Masukkan operasi *j* yang sudah terpilih pada P_{st+1}

Langkah 8 : Hilangkan operasi *j* yang telah dijadwalkan pada S_t

Langkah 9 : Perbaharui *available time* untuk setiap mesin

Langkah 10 : Apakah operasi untuk setiap *job* telah dijadwalkan

- Jika ya, lanjutkan ke langkah 13.
- Jika tidak, lanjutkan ke langkah 11.

Langkah 11 : Apakah $d_i - \sigma_j + t_i < H_h$

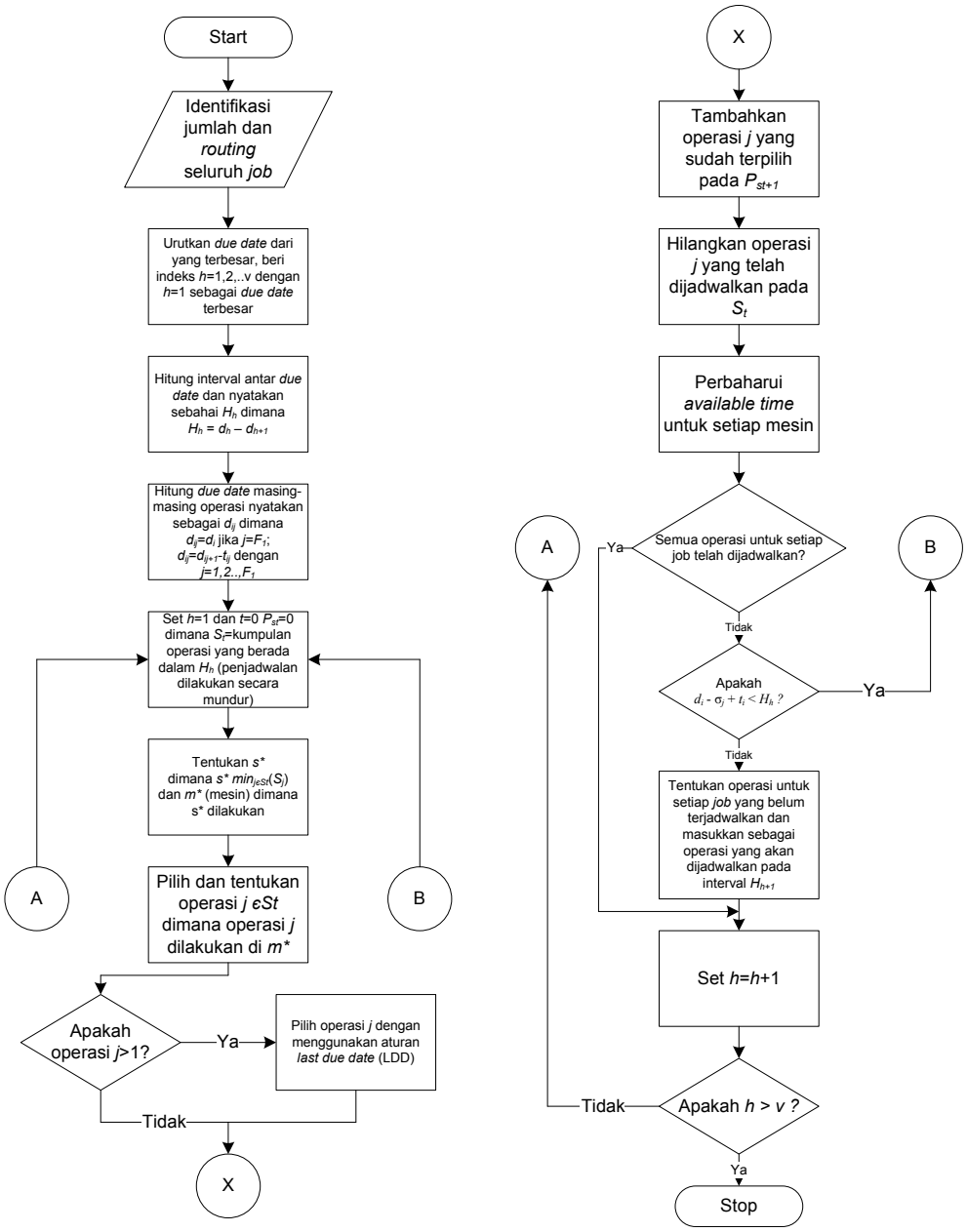
- Kurangkan *due date* dengan jumlah dari σ_j (saat paling awal dimana operasi $j \in S_t$ mulai dikerjakan) dengan t_i (waktu operasi) dan bandingkan dengan H_h (waktu interval antar *due date*)
- Jika ya ($d_i - \sigma_j + t_i < H_h$), kembali ke langkah 4
- Jika tidak ($d_i - \sigma_j + t_i \geq H_h$), lanjutkan ke langkah 12

Langkah 12 : Tentukan operasi untuk setiap *job* yang belum terjadwalkan dan masukan sebagai operasi yang akan dijadwalkan pada interval H_{h+1}

Langkah 13 : Set $h = h + 1$

Langkah 14 : Apakah $h > v$

- Jika ya, maka proses penjadwalan telah selesai
- Jika tidak, lanjutkan dan kembali ke langkah 4



Gambar 2.9 Diagram alir algoritma penjadwalan *backward non-delay* (Zaini et al[2001])

Langkah perhitungan ongkos :

Hitung besarnya $Q_{ij}(x_{ij})$ dimana

a. Untuk $\forall_i, j \in F_i$

Untuk $x_{ij} < d_i$:

$$\underset{j \in F_i}{Q_{ij}(x_{ij})} = \left(\sum_{i,(j \epsilon T_i - F_i)} \alpha_{ij} \right) d_i + \alpha_{ij} (d_i - x_{ij})$$

untuk $x_{ij} \geq d_i$:

$$\underset{j \in F_i}{Q_{ij}(x_{ij})} = - \beta_i d_i + (\beta_i + \sum_{i,(j \in T_i - F_i)} \alpha_{ij}) x_{ij}$$

b. Untuk $\forall_i$, $(j \in T_i - F_i)$

$$\underset{j \in (T_i - F_i)}{Q_{ij}(x_{ij})} = - \sum_{i,(j \in T_i - F_i)} \alpha_{ij} . x_{ij}$$

2.12.2. Algoritma Penjadwalan *Backward Non Delay* pada Kondisi Dinamis

Algoritma ini berisi langkah-langkah untuk mengantisipasi kedatangan *order* baru saat $T = T_m$

Langkah 15 : Periksa apakah ada operasi *job* yang belum selesai diproses

- Jika ya, periksa operasi yang belum selesai tersebut di setiap stasiun kerja dan saat selesai *job* tersebut
- Jika tidak, lanjutkan ke langkah 18

Langkah 16 : Perbaharui nilai T_m menjadi saat penyelesaian operasi dari *job* yang belum selesai

Langkah 17 : Tentukan operasi yang sudah dan belum selesai hingga T_m

Langkah 18 : Jadwalkan kembali operasi yang belum selesai dan operasi dari *job* yang baru dengan algoritma *backward non delay* (langkah 0 – 14)

Langkah 19 : Jika *job* untuk semua *due date* dapat terpenuhi maka *job* baru dapat diterima. Jika tidak semua *due date* dapat terpenuhi, lakukan konfirmasi ulang untuk *job* baru dengan *due date* hasil langkah 18.