Bab 2
Landasan Teori

2.1. Definisi Penjadwalan.

Penjadwalan didefinisikan sebagai rencana pengaturan kerja serta pengalokasian sumber baik waktu maupun fasilitas untuk setiap operasi yang harus diselesaikan. Penjadwalan dapat diartikan sebagai proses pengalokasian sumber-sumber untuk memilih tugas dalam jangka waktu tertentu. Definisi lain mengatakan bahwa penjadwalan ialah proses pengurutan pembuatan produk secara menyeluruh pada sejumlah mesin dalam jangka waktu tertentu.

Penjadwalan merupakan alat ukur yang baik bagi perencanaan agregat. Pesanan-pesanan aktual pada tahap ini akan ditugaskan pertama kalinya pada sumber daya tertentu (fasilitas, pekerja dan peralatan), kemudian dilakukan pengurutan kerja pada tiap-tiap pusat pemrosesan sehingga dicapai optimalitas, utilitas, kapasitas yang ada. Penjadwalan juga dapat didefinisikan sebagai pengambilan keputusan tentang penyesuaian aktivitas sumber daya dalam rangka menyelesaikan sekumpulan *job* atau suatu proyek tepat pada waktunya dan mempunyai kualitas seperti yang diinginkan (*Morton,* 1993).

Dari sekian banyak penjadwalan yang telah ada pada saat ini, inti dari sumber visi tersebut adalah:

a. Penjadwalan berfungsi sebagai alat pengambialan keputusan.
b. Penjadwalan merupakan teori yang berisi prinsip-prinsip dasar, model, teknik, dan kesimpulan logis dalam pengambilan keputusan.

Untuk menyelesaikan masalah penjadwalan yang dihadapi, dapat digunakan beberapa pendekatan. Pendekatan-pendekata tersebut dikelompokan menjadi 2, yaitu:

a. Pendekatan tradisional yang meliputi metoda-metoda penelitian operasional.

b. Pendekatan yang lebih modern yang mencakup gabungan antara metoda penelitain operasional, intelegensia tiruan, simulasi kejadian, dan ide-ide yang diambil dari teori kontrol.

2.1.1. **Tujuan Penjadwalan**

Beberapa tujuan yang ingin dicapai dengan dilaksanakannya penjadwalan produksi adalah:

1. Meningkatkan utilitas sumber daya atau mengurangi waktu tunggunya, sehingga total waktu proses dapat berkurang dan produktivitas dapat meningkat.
2. Mengurangi *makespan,* yang juga berarti menurunkan *flow time* rata-rata dan *work in process* rata-rata.
3. Mengurangi persediaan barang setengah jadi atau mengurangi sejumlah pekerjaan yang menunggu dalam antrian ketika sumber daya yang ada masih mengerjakan tugas yang lain. Teori *Baker* mengatakan, jika aliran kerja suatu jadwal konstan, maka antrian yang mengurangi rata-rata waktu alir akan mengurangi waaktu persediaan.
4. Meminimasi biaya produksi.
5. Mengurangi persediaan barang setengah jadi dengan jalan mengurangi jumlah rata-rata pekerjaan yang menungggu antrian suatu mesin yang dalam keadaan sibuk. Hal ini bertujuan untuk menghindari biaya *flow time,* yaitu biaya penyimpanan produksi setengah jadi.
6. Memenuhi keinginan konsumen, baik itu dalam hal kualitas produk yang dihasilkan maupun dalam ketepatan waktu.
7. Membantu dalam pengambilan keputusan mengenai perencanaan kapasitas pabrik dan jenis kapasitas yang dibutuhkan sehingga penambahan biaya yang mahal dapat dihindarkan.

Pada saat merencanakan suatu jadwal produksi, yang harus dipertimbangkan adalah ketersediaan sumber daya yang dimiliki, baik berupa tenaga kerja, peralatan (*processor*) ataupun bahan baku. Karena sumber daya yang dimiliki dapat berubah-ubah (terutama operator dan baha baku), maka penjadwalan dapat kita lihat merupakan proses yang dinamis.

Adapun tipe keputusan yang akan diperoleh dari pelaksanaan penjadwalan tersebut berupa:

1. Pengurutan pekerjaan (*sequencing*).
2. Penugasan (*dispatching*).
3. Pengurutan operasi suatu *job* (*routing*).
4. Penentuan waktu mulai dan selesai pekerjaan (*timing*).

Persoalan penjadwalan timbul apabila, beberapa pekerjaan (*job*) akan dikerjakan bersamaan, sedangkan sumber daya seperti mesin (peralatan) jumlahnya terbatas. Untuk mencapai hasil yang optimal dengan keterbatasan sumber daya yang dimiliki, maka diperlukan adanya penjadwalan sumber-sumber tersebut secara efisien.

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

a. *Job*

   Job dapat didefinisikan sebagai suatu pekerjaan yang harus diselesaiakan untuk mendapatkan suatu produk. *Job* biasanya terdiri dari beberapa operasi yang harus dikerjakan (minimal 1 operasi). Manajemen melalui perencanaan yang telah dibuat atau berdasarkan pesanan dari pelanggan memberikan *job* kepada bagian *shop floor* untuk dikerjakan. Informasi yang dipunyai oleh suatu *job* ketika datang kebagian *shop floor* biasanya adalah operasi-operasi yang harus dilakukan didalamnya, saat *job* harus diselesaikan dan saat *job* mulai dikerjakan.

b. Operasi

   Operasi adalah himpunan bagian dari *job*, untuk menyelesaikan suatu *job*, operasi-operasi dalam *job* diurutkan dalam suatu urutan pengerjaan tertentu. Urutan tersebut ditentukan pada saat perencanaan proses. Suatu operasi baru dapat dikerjakan apabila operasi atau proses yang mendahuluinya sudah dikerjakan terlebih dahulu. Setiap operasi mempunyai waktu proses, waktu proses (t_{ij}) adalah waktu pengerjaan yang diperlukan untuk melakukan

operasi tersebut. Waktu proses operasi untuk suatu *job* biasanya telah diketahui sebelumnya dan mempunyai nilai tertentu.

c. Mesin

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

2.1.2. **Klasifikasi Penjadwalan**

Masalah penjadwalan dapat diklasifikasikan berdasarkan faktor-faktor:

1. Waktu
2. Tenaga kerja
3. Jumlah mesin
4. Sifat dan syarat pekerjaan

Dua permasalahan utama yang hendak diselesaikan dengan menggunakan penjadwalan:

a. Penentuan mesin yang akan digunakan (pengalokasian mesin) untuk menyelesaikan suatu proses produksi.
b. Penjadwalan penentuan waktu pemakaian mesin tersebut (pengurutan).

2.1.3. ***Input* Sistem Penjadwalan**

Kita harus menentukan kebutuhan-kebutuhan kapasitas-kapasitas dari *order-order* yang dijadwalkan dalam hal macam dan jumlah sumber daya yang digunakan. Untuk produk-produk tertentu, informasi ini diperoleh dari lembar kerja operasi (berisi keterampilan dan peralatan yang dibutuhkan, waktu standar, dll) dan BOM (berisikan kebutuhan-kebutuhan akan komponen, sub komponen dan bahan pendukung). Kualitas dari keputusan-keputusan penjadwalan sangat dipengaruhi oleh ketepatan estimasi *input-input* diatas.

Oleh karena itu, pemeliharaan catatan terbaru tentang status tenaga kerja dan peralatan yang tersedia dan perubahan kebutuhan kapasitas yang diakibatkan perubahan desain produk.

2.1.4. *Output* Sistem Penjadwalan

Untuk memastikan bahwa suatu aliran kerja yang lancar melalui tahapan produksi, maka sistem penjadwalan harus membentuk aktivitas-aktivitas *output* sebagai berikut:

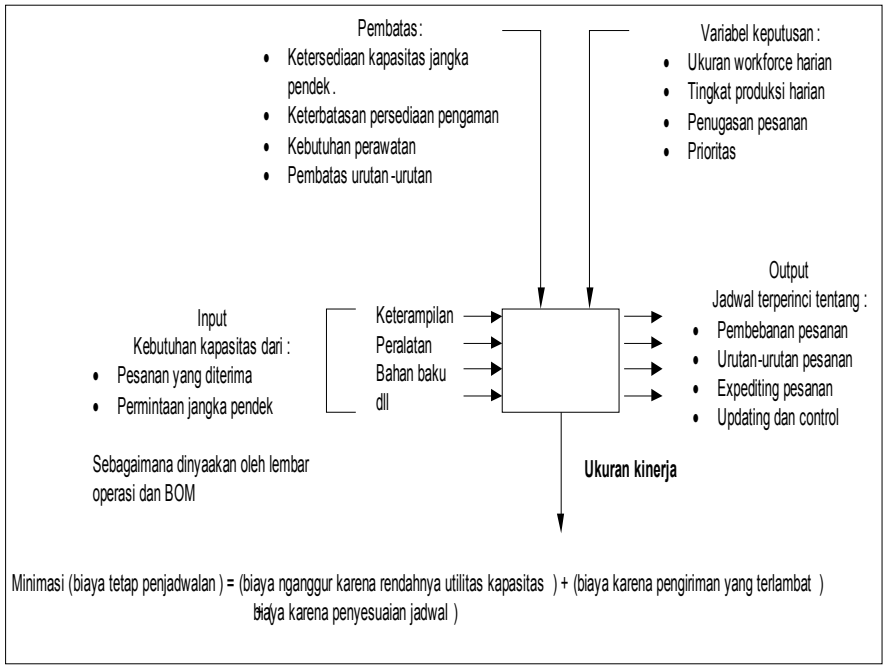
1. Pembebaan (*loading*)

   Pembebanan melibatkan penyesuaian kebutuhan kapasitas untuk *order-order* yang diterima/diperkirakan dengan kapasitas yang tersedia. Pembebanan dilakukan dengan menugaskan *order-order* pada fasilitas-fasilitas, operator-operator dan peralatan tertentu.
2. Pengurutan (*sequencing*)

   Pengurutan ini merupakan penugasan tentang *order-order* mana yang diprioritaskan untuk diproses dahulu bila suatu fasilitas harus memproses banyak *job*.
3. Prioritas *job* (*dispatching*)

   Prioritas *job* merupakan prioritas kerja tentang *job-job* mana yang diseleksi dan diprioritaskan untuk diproses.
4. Pengendalian kinerja penjadwalan, dilakukan dengan:
   a. Meninjau kembali status *order-order* pada saat melalui sistem tertentu.
   b. Mengatur kembali urutan-urutan, misalnya: *expediting, order-order* yang jauh dibelakang atau mempunyai prioritas utama.
5. *Up-dating* jadwal, dilakukan sebagai refleksi kondisi operasi yang terjadi dengan merefisi prioritas-prioritas.

Bila digambarkan elemen-elemen *output-input*, prioritas-prioritas dan ukuran kinerja dari sistem penjadwalan akan tampak seperti pada gambar 2.1. dibawah ini:



Gambar 2.1. Elemen-Elemen Sistem Penjadwalan.

2.2. Permasalahan Penjadwalan

Permasalahan penjadwalan dapat dilihat dari:

1. Mesin:
   a. Mesin tunggal
   b. 2 mesin
   c. m mesin
2. Aliran proses:
   a. *Job shop*
   b. *Flow shop*
3. Pola kedatangan:
   a. Statis
   b. Dinamis
4. Elemen penjadwalan:
   a. Deterministik
   b. Stokastik

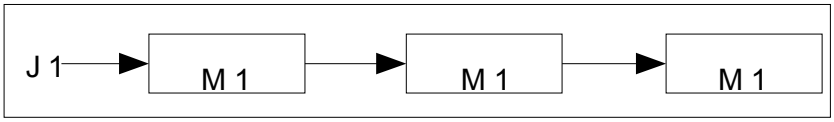
2.2.1. Pola aliran proses.

Terdapat 2 aliran produksi :

1. *Flow shop,* yang cenderung memiliki keamanan urutan operasi (*routing*) untuk semua *job.*

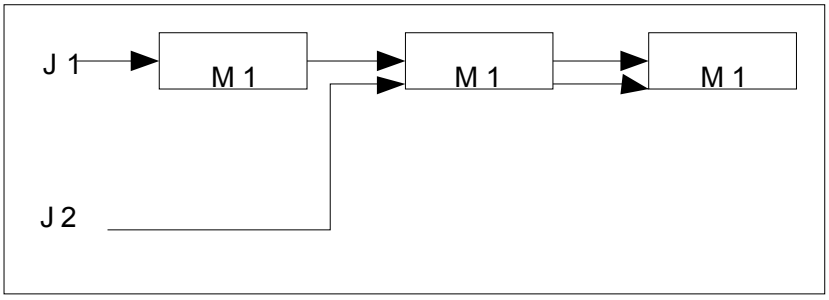
Flow shop dibedakan atas:

a. *Pure flow shop*, yaitu *flow shop* yang memiliki jalur yang sama untuk semua tugas. Lintasan prosesnya dapat dilihat lebih jelas lagi pada gambar 2.2.



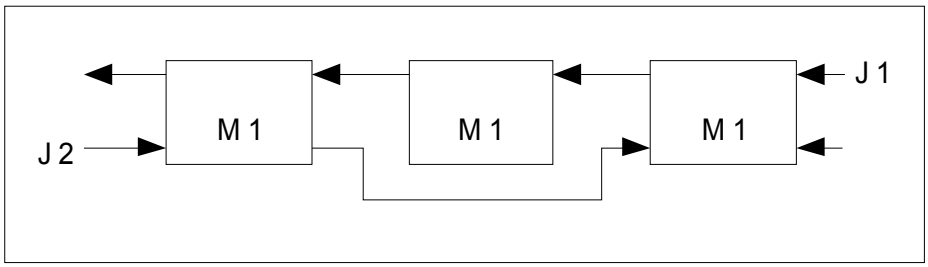
Gambar 2.2. Lintasan Proses *Pure Flow shop.*

b. *General flow shop*, yaitu *flow shop* yang memiliki pola aliran yang berbeda. Ini disebabkan adanya variasi dalam pekerjaan tugas, sehingga tugas yang datang tidak harus dikerjakan pada semua mesin. Jadi mungkin suatu proses dilewati. Lintasan prosesnya dapat dilihat pada gambar 2.3.



Gambar 2.3. Lintasan Proses *General Flow shop*

2. *Job shop* pada aliran *job shop*, masing-masing *job* memiliki urutan operasi yang unik. Setiap *job* bergerak dari satu mesin/stasiun kerja menuju mesin/stasiun kerja lainnya dengan pola yang random. Lintasan prosesnya dapat dilihat pada gambar 2.4.



Gambar 2.4. Lintasa Proses *Job shop*.

Proses *job shop* mempunyai karakteristik dari pengaturan peralatan yang sama berdasarkan fungsi. Sebagaimana aliran *job* dari stasiun kerja ke stasiun kerja lainnya, atau dari suatu departemen ke departemen lainnya, maka karakteristik prosses *job shop* adalah sebagai berikut:

1. Peralatan penanganan material dan peralatan produksi multi-guna dapat diatur dan dimodifikasi untuk menangani produk yang berbeda.
2. Produk-produk yang berbeda diproses dalam *lot-lot* atau *batch.*
3. Pemrosesan *order-order* membutuhkan pengendalian dan perencanaan yang terperinci sehubungan dengan variasi pola-pola aliran dan pemisahan stasiun-stasiun kerja.
4. Pengendalian membutuhkan informasi tentang *job* dan *shop* yang terperici, meliputi urutan proses, prioritas *order*, waktu yang dibutuhkan oleh setiap *job* stasiun dari setiap *job in process,* kapasitas dari stasiun kerja, dan kapasitas yang dibutuhkan dari stasiun kerja kritis pada suatu periode.
5. Beban-beban stasiun kerja yang berbeda secara mencolok, masing-masing memiliki persentase utilitas kapasitas yang berbeda.
6. Ketersediaan sumber-sumber meliputi: material, personal, dan peralatan harus dikoordinasi dengan perencanaan *order*.
7. Sejumlah material *work in process* cenderung meningkat. Hal ini dalam aliran proses menyebabkan antrian-antrian dan *work in process* yang panjang.
8. Menggunakan teknik-teknik penjadwalan tradisional,total waktu dari awal operasi pertama sampai operasi terakhir, relatif panjang dibandingkan dengan total waktu operasi.

9. Para pekerja langsung biasanya memiliki *skill* (keahlian) yang lebih tinggi dan lebih terlatih dari pada pekerja untuk operasi *flow process.*

2.2.2. **Pola kedatangan pekerjaan (*job*).**

Pola kedatangan pekerjaan dapat dibedakan atas :

1. Pola kedatangan statis, yaitu pola dimana pekerjaan datang secara bersamaan dan semua fasilitas tersedia pada saat kedatangan *job.*
2. Pola kedatangan dinamis, yaitu pola dimana pekerjaan datang secara acak/kedatangan pekerjaan tidak menentu.

2.2.3. Sifat informasi.

Dibagi menjadi 2 bagian yaitu:

1. Informasi bersifat deterministik, yaitu suatu informasi yang didalamnya terdapat kepastian tentang pekerjaan dan mesin, misalnya mengenai waktu kedatangan pekerjaan dan waktu proses.
2. Informasi bersifat stokastik, yaitu model didalamnya terdapat ketidakpastian mengenai pekerjaan dan mesin.

Informasi-informasi ini bersifat meliputi informasi yang berhubungan dengan karakteristik *job*, yaitu saat kedatangan, batas waktu penyelesaian, perbedaan kepentingan diantara *job* yang dijadwalkan, banyaknya operasi, waktu proses tiap operasi. Di samping itu terdapat pula infomasi yang menyangkut karakteristik mesin seperti jumlah mesin, kapasitas, fleksibilitas serta efisiensi penggunaan yang berbeda untuk *job* yang berbeda.

2.3. Penjadwalan terbagi menjadi mejadi 3 kategori:

1. *Single Machine Schedulling:*
    a. *Shortest processing time* (SPT)
    b. *Earliest due date* (EDD)
    c. *Minimazing number of job tardy*
2. *Flow shop schedulling*:
    a. Johnson’s algorithm
3. *Job shop schedulling*:

a. *Priority dispatching rules*

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 penjadwalan dapat diselesaikan dengan cara:
   1. *Squenching*, bisa diselesaikan dengan metode:
      a. *Priority rule*
      b. *Queue*
   2. *Timming*, awal dan akhir tiap *job* dihitung berdasarkan pada urutan, *routing* dan waktu proses.

❖ Metode-metode masalah penjadwalan:
   1. Heuristic
   2. Matematis
   3. simulasi

❖ Beberapa definisi dan notasi:
   a. *Processing time* (t_i): waktu yang diperlukan untuk menyelesaikan satu operasi termasuk persiapan dan pengaturan proses.
   b. *Due date* (d_i): batas waktu yang diperbolehkan untuk menyelesaikan suatu pekerjaan.
   c. *Completion time* (c_i): rentang waktu mulai dari awal (t = 0) sampai pekerjaan selesai dikerjakan.
   d. *Lateness* (L_i): perbedaan antara *kompletion time* dengan *due date*, sehingga bisa (+) positif atau (-) negatif.

$$Li = c_i - d_i < 0 \text{ (negatif)}$$: saat penyelesaia memenuhi batas.

$L_i = c_i - d_i > 0$ (positif): saat penyelesaia memenuhi batas = *tardy job*.

e. *Tardiness* (Ti): keterlambatan penyelesaian suatu pekerjaan dari saat *due date*.

f. *Slack time* (Si): waktu sisa yang tersedia bagi suatu pekerjaan (waktu proses-*due date*). $S_i = d_i - t_i$

g. *Flow time* (Fi): waktu antara saat dimana pekerjaan 1 telah siap untuk dikerjakan sampai pekerjaan selesai.

h. *Waiting time* (Wi): waktu tunggu pekerjaan I dari saat pekerjaan siap dikerjakan sampai sampai saat operasi pendahuluan selesai.

i. *Makespan* (Ms): jangka penyelesaian suatu penjadwalan (penjumlahan seluruh waktu proses). $Ms = C_{max}$

j. *Ready time* (Ri): menunjukansaat pekerjaan ke-i dapat dikerjakan (siap dijadwdalkan).

2.3.1. *Single Machine Schedulling.*

Penjadwalan ini menyelesaikan masalah penentuan urutan *job* yang akan diselesaikan pada suatu mesin.

- *Input* data :

  r_i = *ready time*

  t_i = *processing time for job i*

  d_i = *due time*

- *Output* data :

  c_i = *completion time of job i.*

  F_i = *flow time of job i*

  T_i = *tardiness of job i*

  Q_i = *queue time of job i*

  J_r = *number of job in process at time T*

1. *Shortest processing time* (SPT)

Job dengan waktu yang paling singkat dijadwalkan pada urutan yang pertama, kemudian diikuti dengan *job* yang lebih besar.

Dengan pejadwalan SPT dapat diminimasi:

Mean completion time ..... $\bar{c} = \frac{1}{n}\sum_{i=1}^{n} c_i$

Mean flow time ..... $\bar{F} = \frac{1}{n}\sum_{i=1}^{n} F_i$

Mean lateness ..... $\bar{L} = \frac{1}{n}\sum_{i=1}^{n} C_i - d_i$

Mean queue time ..... $\bar{Q} = \frac{1}{n}\sum_{i=1}^{n} Q_i$

Maksimum queue time..... $Q_{\max}$

Mean number of jobs in process..... $\bar{j} = \frac{1}{n}\sum_{i=1}^{n} j_i$

Ada beberapa aturan dalam perhitungan SPT yaitu diantaranya:

a. Aturan SPT meminimumkan *flow time* rata-rata.

   Flow time rata-rata akan minimum jika n *tasks* yang akan diproses disebuah mesin di urut menurut waktu pemrosesan terpendek (*short processing time*).

b. Aturan SPT meminimumkan *mean lateness*.

   Bila n *job* yang akan diproses di sebuah mesin, maka *mean lateness* akan minimum dengan mengurutkan *tasks* tersebut menurut aturan SPT.

2. *Earliest due date* (EDD).

Job dengan *due date* yang paling awal dijadwakan pada urutan yang pertama. Dengan penjadwalan EDD dapat meminimasi:

a. Maximum *lateness* (Lmax)

b. Maximum *tardiness* (Tmax)

Aturan EDD dengan meminimumkan *maximum lateness* pada satu mesin terjadi apabila terdapat n *tasks* yang akan diproses pada sebuah mesin maka *maximum task lateness* dan *maximum task tardiness* akan minimum dengan menurut *tasks* sesuai dengan urutan EDD.

3. *Minimazing Number Job Of Tardy.*

Bertujuan untuk mengurangi jumlah *job* yang terlambat.

Prosedur:

Tahap I : urutkan *job* berdasarkan EDD, nyatakan sebagai set *job* E.

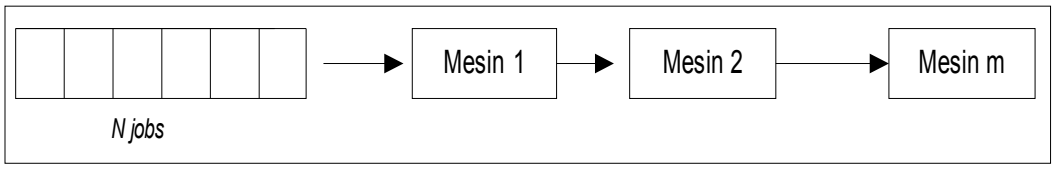
Tahap II : jika tidak ada *job* yang terlambat pada E, penjadwalan tersebut sudah optimal, jika tidak *job* yang pertama terlambat, nyatakan sebagai *job* ke k.

Tahap III : pilih *job* yang memiliki *processing time* yang terpanjang diantara *job* sebelum *job* k dan pindahkan *job* tersebut ke *set*, sesuaikanlah waktu penyelesaian untuk semua *job.*

2.3.2. *Squenching* n *job* pada pada m mesin

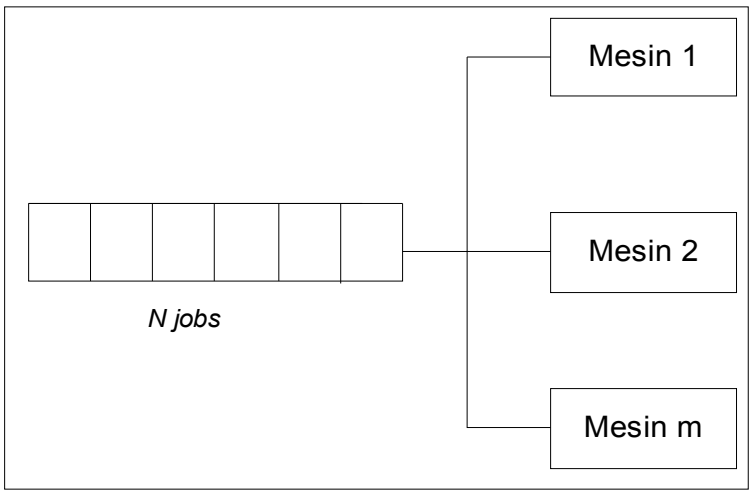
Terdapat 2 kemungkinan dalam pengurutan *job*, yaitu:

1. Menggunakan mesin seri.



Gambar 2.5. Rangkaian Mesin Seri.

2. Menggunakan mesin pararel.



Gambar 2.6. Rangkaian Mesin Pararel.

Untuk kasus pengurutan pada mesin seri ada 3 algoritma, yaitu:

1. Algoritma *Johnson* (minimasi *makespan* pada 2 mesin).
2. Algoritma *Campble* (minimasi *makespan* pada m mesin).
3. Algoritma *Campble, Dudek dan Smith* (minimasi *makespan* pada m mesin).

Untuk kasus pengurutan pada mesin pararel ada 6 algoritma, yaitu:

1. Meminimumkan *mean flow time.*
2. Mengurangi *makespan* dan *mean flow time.*
3. Megurangi maksimum *tardiness.*
4. Mengurangi *tardiness.*
5. Mengurangi *mean tardiness.*
6. Mengurangi jumlah *job of tardy.*

2.3.2.1. Pengurutan Pada Mesin Seri.

1. Algorithma Johnson

Prosedur Johnson digunakan terhadap pengalokasian 2 mesin untuk suatu rangkaian produksi.

Prosedurnya:

Tahap I : tentukan min (t_i1, t_i2) cari nilai proses terkecil dari 2 mesin tersebut.

Tahap II :

a. Jika proses *time* yang minimum memerlukan mesin 1, *job* tersebut mendapat urutan terakhir dari urutan-urutan yang tersedia.
b. Jika proses *time* yang minimum memerlukan mesin, *job* tersebut mendapat urutan terakhir dari urutan-urutan yang tersedia.

Tahap III : pindahkan *job* yang telah diberi urutan, kembali ketahap I, sampai semua urutan terisi.

2. Algoritma Camble (perluasan Johnson)

Digunakan untuk persoalan n *job* di 3 mesin. Adapun langkah-langkahnya, yaitu sebagai berikut:

a. Periksa apakah syarat terpenuhi jika ada n *job* di mesin M1, M2, dan M3 dengan waktu proses M1, M2, M3.

M1, minimum $\geq$ M2 maksimum

M3, minimum $\geq$ M2 maksimum

b. Jika syarat terpenuhi, jadikan 2 kelompok mesin K dan N

Ki = gabungan M1, dan M2

Ni = gabungan M2, dan M3

c. Selesaikan dengan algorithma *Johnson*
d. Buat *schedulle* dengan *gant chart*

3. Algoritma *Campble, Smith, Dudek* (CDS)

Digunakan untuk menyelesaikan n *job* dimesin m mesin. Adapun langkah-langkah penyelesaiannya adalah sebagai berikut:

Jika ada M mesin:

1. Ambil mesin 1 dan mesin M (mesin yang lain dianggap tidak ada) lakukan algorithma *Johnson*. (diperoleh *squence* 1, hitung *makespan*)
2. Ambil mesin 1, mesin 2 dan mesin M_1:
   a. Gabungkan waktu proses mesin 1 dan 2 (K_i = M1 + M2)
   b. Gabungkan waktu proses mesin M dengan M_1 (L_i = MM1 + MM_1)
3. Lakukan algorithma Johnson.
   a. Ambil mesin 1, 2, 3 dan mesin M, M-1, M-2
   b. Gabungkan waktu proses mesin 1, 2, dan 3 (P_i = $M1_i$ + $M2_i$ + $M3_i$)
   c. Gabungkan waktu proses mesin M, M-1, M-2 (Q_i = MM_i + $MM\text{-}1_i$ + $MM\text{-}2_i$)
4. Lakukan terus sampai semua mesin teranalisa (diperoleh M-1 *squence* dengan M-1 *makespan*).
5. Pilih *makespan* terkecil. Urutkan pengerjaan yang menghasilkan *makespan* terkecil tersebut yang terpilih.

2.3.2.2. Pengurutan Mesin Pararel .

Ada beberapa cara untuk menyelesaikan permasalahan pada pengurutan mesin pararel , yaitu:

1. Minimasi mean flow time

Prosedur:

a. Urutkan seluruh *job* menurut aturan SPT.
b. Ambil tugas satu persatu dari aturan SPT dan jadwalkan pada mesin dengan beban kerja tersingkat. Bila terdapat beban yang sama, pilih salah satu secara sembarang.

2. Mengurangi *makespan* dan *mean flow time*.

Prosedurnya:

a. Urutkan seluruh *job* menurut aturan LPT (*long process time*).
b. Jadwalkan urutan LPT pada mesin yang bebannya tersingkat. Bila terdapat mesin dengan beban yang sama, pilih salah satu secara sembarang.
c. Bila seluruh *job* telah dijadwalkan, ubah urutannya disetiap mesin menurut aturan SPT.

3. Mengurangi maksimum *tardiness*.

Prosesurnya :

a. Urutkan seluruh *job* dengan arturan EDD (*earliest due date*)
b. Jadwalkan pada mesin dengan beban kerja tersingkat. Bila terdapat mesin dengan beban yang sama, pilih salah satu secara sembarang.

4. Mengurangi *tardiness*

Prosedurnya:

a. Urutkan seluruh *job* dengan aturan *slack*
b. Jadwalkan mesin dengan beban kerja tersingkat. Bila terdapat mesin dengan beban yang sama, pilih salah satu secara sembarang.

5. Mengurangi mean *tardiness*

Prosedurnya:

a. Urutkan seluruh *job* berdasarkan *priority rule* yang sudah dihitung (SPT, EDD, *Slack*).
b. Dari ketiga hasil *priority rule* tersebut minimasi mean *tardiness* dengan mengurutkannya berdasarkan algoritma *Wilkenson-Irwin*.
c. Pilihlah dari ketiga alternatif yang memiliki mean *tardiness* terkecil.

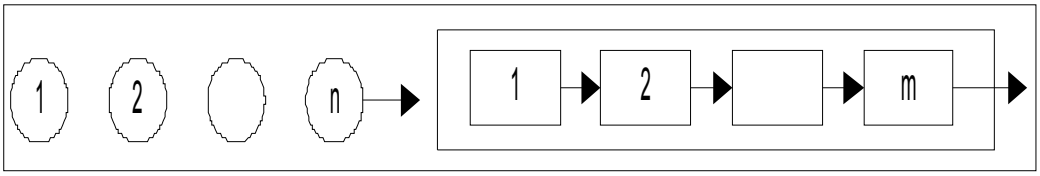
6. Menguragi jumlah *job tardy*

Prosedurnya:

a. Urutkan seluruh *job* menurut aturan EDD
b. Ambil *job* yang *tardy* di masing-masing mesin, apabila mempunyai t_i terpanjang letakan pada urutan terakhir pada mesin yang bersangkutan. Perbaharui *completion time* masing-masing *job* dan kembali ke prosedur 2.

2.3.3. *Flow shop Schedulling.*

Penjadwalan *flow shop* adalah proses penentuan urutan pengerjaan untuk suatu lintas produksi yang dapat digunakan beberapa jenis produk.



Gambar 2.7. sketsa *flow shop*.

Persoalan penjadwalan pada *flow shop* sangat banyak, diantaranya adalah:

1. *Simple flow shop:*
   - *Continous*

     1 *job* di 1 set mesin kontinyu → tidak ada masalah *schedulling.*
   - *Repetitive*

     N *job* di 1 set mesin (*intermitten*) tanpa waktu *set up* → perlu *squenching.*
   - *Batch flow*

     N *job* di 1 *set* mesin dengan waktu set up tertentu → perlu *squenching*
2. Variasi *flow shop:*
   - *Skip shop*

     N *job* di 1*set* mesin dimana proses pengerjaannya melalui *flow* yang sama tetapi beberapa mesin di lewat.
   - *Re entrant flow shop*

     Job di 1 *set* mesin dimana urutan pengerjaan ke-n *job* melalui urutan yang sama tetapi ada mesin yang memproses 2 kali.
   - *Compound flow shop*

     N *job* di 1 *set*, tetapi di salah satu bagian mesinnya bisa diganti dengan satu *group* mesin (pararel).
   - *Finite queue flow shop*

     N *job* di 1 *set* mesin, tetapi kapasitas *storage* disetiap mesin berbeda.

Pada persoalan *simple flow shop*:

1. Persoalan sederhana.
2. Bisa diselesaikan dengan *proirity rule.*

Pada variasi *flow shop:*

1. Persoalan lebih rumit.
2. Biasanya diselesaikan dengan heuristic (Palmer, Gupta, Branch and Bound).

2.3.4. *Job Shop Schedulling*

Penjadwalan *job shop* adalah pengurutan pekerjaan untuk litasan produk yang tidak beraturan (tata letak pabrik berdasarka proses).

Penjadwalan pada proses produksi tipe *job shop* lebih sulit dibandingkan dengan penjadawalan *flow shop*. Hal ini disebabkan oleh 3 alasan:

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 yang dipilih harus diproses seketika pada saat *order* tersebut 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.

2.3.4.1. 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 pemprosesan tiap operasi dari suatu *job* dan matriks *routing* yang menunjukan urutan mesin untuk mengerjakan beberapa operasi dari suatu *job*.

Penyelesaian persoalan penjadwalan n *job* m mesin pada lintasan produksi *job shop* $(n!)^m$jadwal. Suatu jadwal dikatakan layak jika memenuhi kriteria (Forgaty; 1991);

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 (SA)

Jadwal semi aktif diperoleh jika tidak ada satu operasi pun yang dapat dikerjakan lebih awal tanpa mengubah susunan operasi lainnya. Penjadwalan ini diusulkan oleh *Giffler* dan *Thompson* diformulasikan oleh (Backer; 1974):

Notasi yang digunakan adalah sebagai berikut:

P_{St} = Suatu jadwal parsial yang mengandung sejumlah t operasi yang telah dijadwalkan

S_t = Kumpulan operasi yang siap dijalankan pada saat t.

t = *Stage* (tahap)

Rj = Saat paling awal operasi j siap dijadwalkan/dikerjakan.

t_j = Waktu operasi ke j

- Langkah 1, t = 0, $P_{st} = 0$

  Tentukan operasi yang siap akan dijadwalkan.

  $i, j, k\,(t_j)$: i = *job*; j = operasi; k = mesin.

  (*job*, operasi, mesin)

- Langkah 2, cari C*j minimum $\left(c_j \ni S_t\right)$

  Untuk k (mesin tertentu) identifikasi mesin yang digunakan untuk $c_j \rightarrow n$

  Operasi yang digunakan pada stage t:

  a. Operasi C*j
  b. Operasi dengan mesin n dengan syarat $r_j \geq c^*{}_j$
  c. Jika operasi dikerjakan pada mesin yang sama maka akan dilakukan dengan aturan prioritas yang digunakan.

- Langkah 3, gantikan operasi yang dijadwalakan dengan operasi berikutnya.
- Langkah 4, periksa apakah semua operasi selesai dijadwalkan, apabila belum kembali ke langkah 2.

2. Jadwal aktif (A)

Pada jadwal aktif tidak ada operasi yang dapat dilakuakan lebih awal tanpa menunda operasi lainnya atau melanggar *technological constrain* (memberikan urutan proses pada setiap *job*, atau dengan kata lain memberikan *routing* untuk setiap *job*).

Adapun langkah-langkahnya yaitu sebagai berikut :

P_{St} = Suatu jadwal parsial yang mengandung sejumlah t operasi yang telah dijadwalkan

S_t = *Set* operasi-operasi yang siap dijadwalkan pada saat stage ke-t.

t = *Stage* (tahap)

Rj = Saat paling awal dimana operasi $j \in s_t$ siap dijadwalkan/dikerjakan.

C_j = Saat paling awal operasi $j \in s_t$ dapat diselesaikan dimana $j = R_j + t_{ij}$

T_{ij} = Waktu pemrosesan operasi dari *job* ke i pada operasi ke j

Saat paling awal, yaitu suatu operasi dapat dimulai (R_j*) pada mesin m ditentukan oleh penyelesaian dari operasi pendahulunya (j-1) dan penyelesaian operasi terakhir pada mesin m, sehingga berlaku R_j* = maks (R_{j-1}, t_m)

Algoritma dari penjadwalan ini adalah sebagai berikut:

- Langkah 1, t = 0, P_{st} = 0

  Tentukan operasi yang siap akan dijadwalkan.

  $i, j, k\,(t_j)$: i = *job*; j = operasi; k = mesin.

  (*job*, operasi, mesin)

- Langkah 2, cari R_j minimum $\left(R_j \ni S_t\right)$

  Untuk k (mesin tertentu) identifikasi mesin yang digunakan untuk $R_j \rightarrow n$

  Untuk proses yang digunakan dimesin yang sama, digunakan aturan prioritas untuk menentukan aturan operasi yang masuk jadwal.

- Langkah 3, gantikan operasi yang dijadwalakan dengan operasi berikutnya.
- Langkah 4, periksa apakah semua operasi selesai dijadwalkan, apabila belum kembali ke langkah 2.

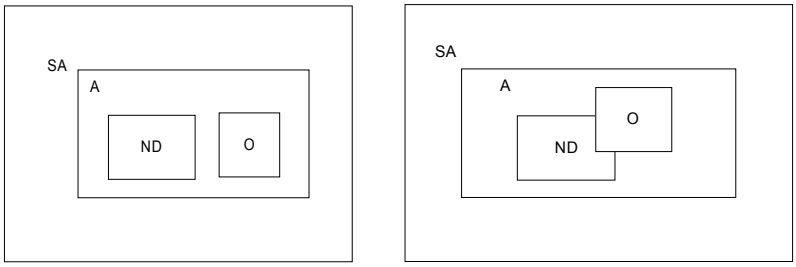
3. Jadwal optimal (O)

   Jadwal yang memberikan ukuran performansi yang optimal.

4. Jadwal *non delay* (ND)

Pada jadwal *non delay* ini tidak ada yang biarkan menganggur sementara pada saat ini terdapat operasi yang dapat dikerjakan pada mesin tersebut. *Baker* (1974) menyatakan bahwa jadwal *non delay* merupakan subset dari jadwal aktif, tetapi jadwal optimal belum tentu berada dalam jadwal *non delay.*

Ruang jawab persoalan penjadwalan *job shop* secara lebih jelas dapat dilihat pada gambar 2.8.



$ND \cap O = 0$ $ND \cap O \neq 0$

Gambar 2.8. Ruang Jawab Persoalan Penjadwalan *Job Shop*.

2.3.4.2. Teknik-Teknik Penyeleaian Masalah *Job Shop*.

1. Teknik pendekatan optimal

Teknik pendekatan optimal merupakan pendekatan yang memberikan solusi terbaik terhadap suatu permasalahan ditinjau dari kriteria tertentu. Pendekatan optimal akan menghasilkan jadwal yang optimal, namun pendekatan optimal untuk permasalahan penjadwalan dengan jumlah operasi dan mesin yang relatif besar akan menyebabkan tingkat kesulitan penyelesaian masalah menjadi tinggi dan membutuhkan waktu yang lebih banyak.

Pendekatan optimal memiliki 2 metode, yaitu:

a. Metode *Integer Programing*
b. Metode *Branch and Bound*

2. Teknik pendekatan heuristik.

Teknik ini digunakan dalam masalah penjadwalan untuk jumlah operasi yang lebih kompleks dengan waktu penyelesaian yang relatif lebih cepat. Walaupun

pendekatan heuristik tidak menghasilkan jadwal yang optimal, namun pendekatan heuristik dapat menghasilkan jadwal yang baik dan mendekati optimal.

Teknik pendekatan heuristik terbagi atas:

a. Priority dispatching rules.

Adalah salah satu jenis metode peramalan dimana waktu siap (*ready time*) dari setiap mesin ditentukan sedemikian rupa sehingga berurutan naik. Keputusan pemilihan produk yang akan diproses dapat dilakukan pada saat mesin siap menerima produk (mesin menganggur). Pada teknik *dispatching* digunakan aturan prioritas untuk memilih salah satu operasi diantara operasi-operasi yang mengalami konflik pada mesin m* pada setiap tahap.

❖ **Aturan Prioritas.**

Aturan prioritas ini digunakan untuk memenuhi *job* mana yang akan dikerjakan terlebih dahulu (Baker; 1974) mengklasifikasikan aturan-aturan prioritas kedalam 2 tipe, yaitu:

1. Aturan Prioritas Lokal

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

a. *Shortest Processing Time* (SPT)
   Prioritas tertinggi diberikan pada *job* yang memiliki waktu proses terpendek.

b. *Least Work Remaining* (LWKR).
   Prioritas tertinggi diberikan pada *job* yang memiliki sisa waktu yang terpendek.

c. *First come first served* (FCFS)
   Job yang pertama datang itulah *job* yang dikerjakan terlebih dahulu.

d. *Most work remaining* (MWKR)
   Prioritas tertinggi diberikan pada *job* yang memiliki waktu proses terbanyak.

e. *Most operation remaining* (MOPNR)
   Prioritas tertinggi diberikan pada *job* yang memiliki waktu proses terbanyak dan terpanjang.

2. Aturan Prioritas Global.

Tipe ini memanfaatkan informasi/status dari mesin-mesin yang lainnya. Aturan yang tergolong kedalam tipe ini adalah:

a. *Anticipates Work In Next Queue* (AWINQ)

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

b. *First Of First On* (FOFO)

   Prioritas tertinggi diberikan kepada operasi yang selesai paling awal.

Selain itu (Backer; 1974) juga mengklasifikasikan aturan ini berdasarkan informasi-informasi yang bersifat dinamis. Pembagian tersebut sebagai berikut:

1. Aturan statis

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

a. *First Arrival at the Shop Served* (FASFS).

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

b. *Total Work* (TWORK).

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

c. *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:

a. *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.

b. *First Come First Serve* (FCFS)

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

c. *Slack Time per Operation* (ST/O)

Prioritas yang tertinggi diberikan pada *job* yang memiliki harga ST/O paling kecil.

d. *Salck Time* (ST)

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

Untuk menyelesaikan permasalahan *job shop* banyak cara yang dapat di tempuh diantaranya dengan metoda matematis, heuristik, dan simulasi. Kebanyakan untuk menyelesaikan permasalahan ini digunakan metode *heuristic*, salah satunya adalah adanya prioritas rules.

Biasanya *priority rules* ini dipakai baik untuk operations *schedulling* maupun *dispatching*. Ada beberapa aturan yang bisa digunakan dalam *priority rules* ini, yaitu:

1. Random

   Memilih *job* dalam antrian dengan kemungkinan yang sama bagi setiap *job*.
2. *Critical Ratio* (CR)

   Prioritas yang tertinggi diberikan pada *job* CR terkecil.
3. *Most Work Remaining* (MWKR)

   Prioritas yang tertinggi diberikan pada *job* yang memiliki sisa waktu proses terlama.
4. *Shorthest Procesing Time* (SPT)

   Urutan *job* berdasarkan waktu yang terpendek. Aturan ini cenderung mengurangi *work in process, mean flow time* serta *mean lateness.*
5. *Least Work Remaining* (LWR)

   Aturan ini mempertimbangkan *succeccive operation* yaitu semua operasi yang tergantung dari operasi yang bersangkutan.
6. *Least Set-Up* (LSU).

   Memilih *job* yang memiliki waktu set-up yang terkecil, dengan demikian akan meminimasi *change over time* (perubahan waktu yang berlebihan).

b. ***Sampling Procedurs.***

c. Probabilistic Dispatching Procedurs.

2.3.4.3. Matriks Routing Dan Matrik Waktu Untuk Persoalan *Job shop*

Dalam penjadwalan *job shop* diperlukan *input* berupa jumlah *job*, jumlah operasi dalam tiap *job* dan urutan operasi beserta mesin yang memprosesnya (*routing*). Hal ini ditampilkan dalam bentuk matriks waktu proses, yang menyatakan urutan mesin yang memproses tiap-tiap urutan operasi.

Suatu penjadwalan digambarkan dengan susunan balok-balok, dimana dari tiap balok merupakan *triplet* dari *job*-operasi-mesin. Panjang balok menyatakan waktu proses pekerjaan yang bersangkutan. Notasi *triplet* yang digunakan pada tiap balok (i,j,k) dimana i menunjukan nomor pekerjaan, j menyatakan urutan pekerjaan, dan k mewakili mesin yang diperlukan. Masalah penjadwalan ini dideskripsikan secara grafis dengan menggunakan skala *gantt chart*. Contoh matriks waktu proses dan matriks *routing* mesin (*Baker*; 1974) dapat dilihat pada table 2.1 dan 2.2

Tabel 2.1. *Matriks* Waktu Proses

job	Operasi		
	1	2	3
1	3	2	4
2	4	4	1
3	2	2	3
4	3	3	3

Tabel 2.2. *Matriks Routing*

job	Operasi		
	1	2	3
1	2	3	1
2	1	3	2
3	2	1	3
4	3	1	2

Pada matriks waktu proses, operasi 1 dari *job* 1 mewakili waktu proses 3 satuan waktu dan pada *matriks routing*, operasi 1 dan *job* 1 dikerjakan dimesin 1. untuk menulis kondisi tersebut sering kali digunakan nitasi O_{ijk} untuk mempersentasikan suatu operasi j dari *job* i diproses dimesin k dan t_{ijk} untuk mempersentasikan

waktu proses oprasi j dari *job* i diproses dimesin k. setelah *input* dari masing-masing telah didefinisikan, proses selanjutnya adalah penugasan operasi *job* dari tiap mesin. Penugasan ini mempunyai aturan yang bermacam-macam dan penggunaannya biasanya ditentukan oleh kebijaksanaan manajemen berdasarkan sistem produksinya serta kriteria optimalitas yang diinginkan.

2.4. Tipe Lingkungan Penjadwalan.

Lingkungan penjadwalan dalam suatu sistem produksi dapat dibedakan beberapa macam yang masing-masing mempunyai karakteristik yang berbeda. Tipe-tipe lingkungan penjadwalan dalam sistem produksi, antara lain:

1. *Classic Job shop*

   Karakteristik sistem produksi ini adalah produknya diskrit, alirannya kompleks, *job*nya unik dan *part-part*nya tidak multi *purpose* (kegunaan).

2. *Open Job shop*

   Sistem produksi ini hampir sama dengan *job shop,* perbedaannya terletak pada *job* yang berulang dan *part* yang *multi purpose.* Selain pada sistem produksi ini *job-job* yang dikerjakan seringkali mempunyai alternatif *routing.*

3. *Batch Shop*

   Proses produksinya bisa diskrit atau kontinyu, alirannya kurang kompleks, banyak *job* berulang, *part multi purpose,* pengelompokan dan penentuan ukuran *lot* menjadi suatu yang penting.

4. *Flow shop*

   Proses produksinya bisa diskrit atau kontinyu, aliran linear, *job* mempunyai kemiripan yang tinggi, pengelompokan dan penentuan ukuran *lot* menjadi suatu yang penting.

5. *Batch/Flow shop*

   Mirip dengan *flow shop,* dengan perbedaan mempunyai proses *batch* yang kontinyu.

6. *Manufakturing Cell*

   Proses produksinya diskrit, mempunyai tipe open *job shop* atau *batch shop* yang terotomatis.

7. *Assembly Shop*

   Versi perakitan (*Assembly Version*) dari *open job shop* atau *batch shop.*

8. *Assembly Line*

   Volume produksinya tinngi dan variasinya rendah.

9. *Transfer Line*

   Sistem ini bercirikan volume produk sangat tinggi dan bervariasi rendah, fasilitas produksi yang linear dengan operasi yang terotomatis.

10. *Flexsible Transfer Line*

    Versi yang lebih modern dari sel dan lini transfer dimaksudkan untuk memperoleh keuntungan dari tingginya volume produksi ke *item job shop.*

2.5. Kriteria Optimalitas Dalam Penjadwalan.

Pemilihan suatu sistem penjadwalan, pendekatan atau teknik yang digunakan tergantung pada tujuan jadwal dan kriteria optimalitas yang menjadi titik berat dan menjadi perhatian manajemen. Sehingga tujuan-tujuan dan kebijakan manajemen adalah dasar dari suatu penjadwalan.

Berikut ini adalah beberapa perhitungan atau pengukuran dan istilah yang sering digunakan dalam menentukan kriteria optimalitas/performansi dalam suatu sistem produksi *job shop* (*Backer*; 1974):

1. *Process Time* (t_j)

   Waktu yang diperlukan untuk menyelesaikan suatu operasi j, termasuk didalamnya waktu untuk persiapan (*set-up*).

2. *Makespan* (M_s)

   Bisa disebut juga *flow time* maksimum, yaitu jangka waktu penyelesaian suatu penjadwalan yang merupakan jumlah seluruh waktu proses.

$$M_s = \sum_{j=1}^{n} t_j$$

3. *Reday Time* (r_j)

   Waktu yang dibutuhkan suatu *job* pada saat siap untuk dijadwalkan.

4. *Due date* (d_l)

   Batas waktu penyelesaian yang ditentukan untuk *job* i atau batas waktu akhir suatu *job* boleh diselesaikan.

$$F_i = t_I + W_i$$

5. *Waiting Time* (W_1)

   Waktu tunggu seluruh operasi dari suatu *job*.

$$W_1 = \sum W_j$$

6. *Flow Time* (F_1)

   Lamanya *job* i berada dilantai pabrik. *Flow time* dihitung sejak *job* dijadwalkan sampai *job* selesai dikerjakan.

$$F_1 = t_1 + W_1$$
$$F_1 = C_1 - r_j$$

7. *Completion Time* (C_1)

   Waktu yang dibutuhkan sejak satu *job* mulai di *set-up* sampai selesai diproses.

$$C_1 = F_1 + r_1$$

8. *Mean Flow Time*

   Rata-rata waktu yang dihabiskan oleh setiap *job* dilantai produksi.

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

9. *Lateness* (L_l)

   Besarnya simpangan waktu penyelesaian *job* i terhadap *due date* yang telah ditentukan untuk *job* tersebut.

$$L_i = C_1 - d_1$$

   $L_i < 0$, saat penyelesaian memenuhi batas akhir.

   $L_i > 0$, saat penyelesaian melewati batas akhir.

10. *Mean Lateness*

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

11. *Tardiness* (Ti)

    Keterlambatan penyelesaian suatu pekerjaan hingga saat *due date*nya. Ditunjukan oleh *lateness* yang berharga positif. Jika *lateness* berharga negatif maka besarnya *tardiness* adalah nol.

$$Ti = \max (Li,0)$$

12. *Mean Tardiness*

    Rata-rata keterlambatan seluruh *job* yang dijadwalkan.

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

13. *Earliness* (Ei)

Penyelesaian suatu *job* lebih awal dari *due date*nya atau *lateness* yang bernialai negatif.

$$Ei = \min (Li,0)$$

14. *Slack time* (Si)

Waktu sisa yang tersedia bagi suatu *job*.

$$Si = d_i - t_i$$

15. *Utilitas mesin* (U)

Merupakan bagian dari kapasitas mesin yang dibebani untuk menjalankan proses-proses yang dibutuhkan terhadap waktu yang tersedia.

$$U = \frac{\sum_{i=1}^{n} t_i}{C_{\max}}$$

16. *Number of Tardy Job*

Menunjukan kuantitas *job* yang terlambat.

$$N_i = \sum_{i=1}^{n} d_i$$

Dimana :

$d_i = 1$ jika $Ti \geq 0$

$di = 0$ jika $Ti \leq 0$

17. *Critical Ratio* (CR)

Prioritas tertinggi diberikan kepada *job* yang memiliki CR terkecil.

$$C = \frac{a_{j(t)}}{P_j}$$

Dimana :

$$a_{j(t)} = d_j - t$$

$a_{j(t)}$ = *allowance*

d_j = *due date*

Pj = waktu yang dibutuhkan untuk menyelesaikan operasi j, sehingga ;

$$Pj = a_{j(t)} - Sj$$

Sj = slack time

Sedangkan kriteria optimalisasi penjadwalan yang sering digunakan pada saat ini adalah :

2.5.1. Berkaitan dengan waktu

a. Minimasi *mean flow time*
b. Minimasi *max flow time*
c. Minimasi *mean terdiness*
d. Minimasi *mean lateness*
e. Minimasi *max terdiness*

2.5.2. Berkaitan dengan ongkos

kriteria ini lebih menekankan pada ongkos yang timbul dari penjadwalan yang dilakukan seperti ongkos penalti akibat keterlambatan, ongkos *flow time*, dan ongkos *inventory*. Adapun tujuan akhir yang ingin dicapai adalah mendapatkan ongkos yang minimal.

Beberapa kriteria yang berkaitan secara langsung terhadap ongkos ini adalah :

- ❖ Minimasi jumlah produk yang mengalami keterlambatan
- ❖ Maksimasi utilitas mesin

2.5.3. Kriteria gabungan

Merupakan gabungan atau kombinasi dari beberapa optimalitas.

Sedangkan *Simon French* (1982) mengelompokan beberapa besaran yang digunakan untuk mengukur keberhasilan suatu penjadwalan adalah sebagai berikut:

1. Berdasarkan waktu penyelesaian:
   - ❖ Minimasi maksimal *makespan*
   - ❖ Minimasi rata-rata *makespan*
   - ❖ Minimasi waktu penyelesaian maksimal
2. Berdasarkan batas penyelesaian:
   - ❖ Minimasi maksimal keterlambatan
   - ❖ Minimasi rata-rata waktu keterlambatan
   - ❖ Minimasi jumlah *job* yang terlambat

- ❖ Minimasi rata-rata jumlah *job* yang terlamabat

3. Berdasarkan penggunaan sumber:
   - ❖ Rata-rata jumlah *job* yang menunggu untuk diproses
   - ❖ Rata-rata waktu mesin menunggu

2.6. Asumsi Dasar Penjadwalan.

Dalam menyelesaikan suatu masalah penjadwalan biasanya diberlakukannya beberapa asumsi yang menyangkut karakteristik tugas, operasi mesin yang digunakan, dan waktu proses. Hal ini dimaksudkan untuk menyederhanakan masalah penjadwalan itu sendiri. Asumsi-asumsi dasar tersebut (*Baker*; 1974), antara lain;

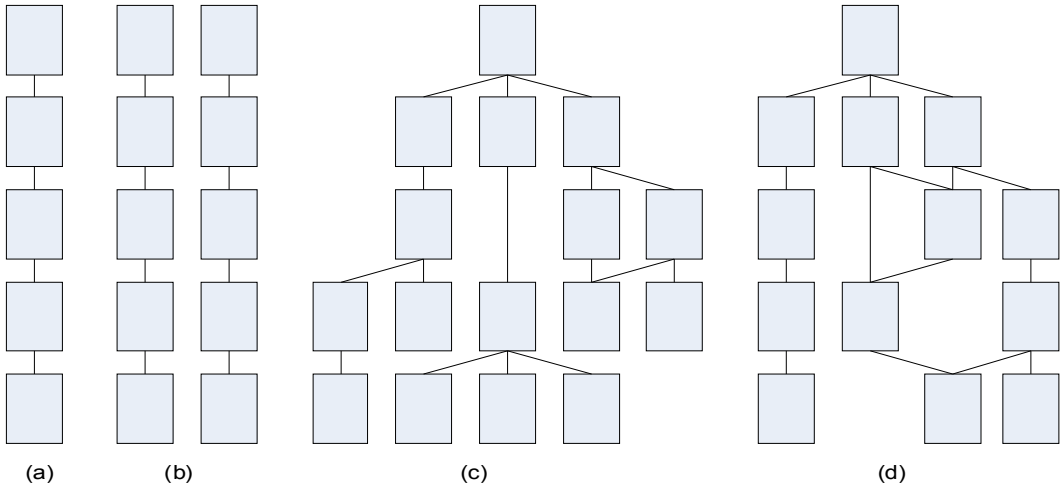
1. Karakteristik *job*
   - ❖ *Job* terdiri dari urutan operasi yang telah ditentukan
   - ❖ Suatu operasi hanya bisa dikerjakan pada satu tipe mesin dari setiap tipe mesin dalam *shop*.
   - ❖ Waktu proses diketahui dengan pasti seperti halnya *due date*.
   - ❖ Urutan waktu *set-up* bersifat independen dan waktu transportasi antar mesin dapat diabaikan.
   - ❖ Operasi yang sedang dikerjakan pada mesin tidak dapat diinterupsi.
   - ❖ 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 proses 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 berdasarkan pada urutan lainnya.
   - ❖ Setiap operasi boleh diproses lebih dari satu kali dimesin yang sama.

- ❖ Setiap operasi dapat diproses pada beberapa jenis mesin yang mampu melaksanakan operasi tersebut.
- ❖ Setiap *job* hanya mempunyai satu *routing* dalam memproses operasi-operasinya.

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 keruksakan atau perawatan.
   - ❖ Setiap mesin beroprasi sesuai dengan informasi waktu dan distribusi yang diketahui secara tepat.
4. Karakteristik waktu proses
   - ❖ Waktu proses telah diketahui 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

2.6.1. Penjadwalan Produk Bersetruktur Multi Level.

Berdasarkan strukturnya, setiap produk dapat dibedakan atas produk yang berstruktur satu level dan produk yang berstruktur multi level. Produk-produk yang berstruktur satu level biasanya disebut juga sebagai komponen tunggal yaitu produk-produk yang hanya membubuhkan serangkaian operasi pemesinan dengan urutan proses (*routing*) tertentu. Produk-produk yang mempunyai struktur multi level terdiri dari sejumlah *part* (komponen) yang harus diproses melalui serangkaian operasi pemesinan dan peralatan hingga menjadi produk akhir. *Billington* et.al. (1983) menggambarkan struktur produk ini seperti pada gambar 2.9.



Gambar 2.9. Empat Macam Struktur Produk

(a) seri: satu produk satu level; (b); pararel: multi produk satu level dengan proses seri
(c); *assembly*: satu produk multi level, tidak terjadi pengalaman komponen
(d); general: satu produk multi level, terjadi kesamaan komponen

Meskipun tidak dinyatakan secara eksplinsit, persoalan penjadwalan (Dimyati dkk; 1999) pada umumnya berkaitan dengan penjadwalan produk-produk yang berstruktur satu level, pada hal pada system manufaktur yang sebenarnya produk-produk yang berstruktur multi level lebih banyak dijumpai. Pada struktur multi level, hubungan ketergantungan (*precedence relationship*) diantara proses pemesinan dan proses perakitan dapat dinyatakan sebagai suatu diagraph G.

Setiap tahapan proses dinyatakan dengan proses, dimana penomoran level produk dengan level 0 untuk produk akhir, kemudian nomor level bertambah satu pada setiap langkah mundur hinnga mencapai komponen dasar.

2.7. Metode Penjadwalan

Proses perancangan algoritma penjadwalan dengan memperhatikan aktivitas perawatan pencegah merupakan pengembangan dari rancangan algoritma penjadwalan heruristik yang telah dikenal antara lain (Santoso; 1994);

1. Metode Forward

Metode ini menjadwalkan proses kerja dalam setiap sumber daya mulai sejak awal produkasi (saat t = 0) sampai dapat diselesaikannya suatu produk yang direncanakan. Tujuan dari metode ini adalah menjadwalkan produksi apabila titik waktu mulainya telah ditentukan sebelumnya dan tidak diberikan batas waktu harus diselesaikannya keseluruhan suatu produk. Kelebihan metode ini adalah bahwa penjadwalan dapat disusun secara *short processing time* (SPT) sehingga didapat suatu penjadwalan produksi dengan *flow time* yang minimum. Kelemahannya adalah karena titik awal ditentukan lebih dahulu, maka ada resiko keterlambatan sehingga dapat menimbulkan kerugian berupa *penalty cost* dan kepercayaan konsumen, selain itu jika selesai sebelum *due date* maka ada ongkos simpan.

2. Metode Backward

Metode ini menjadwalkan produksi mulai dari batas akhir diselesaikannya keseluruhan produk (*due date*) kemudian terus mundur kebelakang sampai didapatkan waktu mulainya produksi. Kelebihan metode ini adalah hasil penjadwalan dengan metode *backward* tidak akan terlambat, karena dijadwalkan mundur atau dengan kata lain *due date* selalu dapat dipenuhi. Kelemahan metode ini tidak bisa diterapkan pada penjadwalan n *job* m mesin jika ada beberapa *job* yang mempunyai *due date* yang sama dan operasi terakhir pada mesin 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 ada kemungkinan menghasilkan penjadwalan yang tidak fleksibel yaitu waktu mulai proses pertama kali sudah terlewat dari waktu sekarang (waktu saat menjadwalkan).

3. Metoda Kompromi

Metoda ini merupakan gabungan antara *forward* dan metoda *backward*, sehingga dari penggabungan metoda tersebut maka dapat mereduksi kelemahan metoda masing-masing. Tujuan dari metode ini adalah membentuk suatu penjadwalan dengan *flow time* yang minimum, dapat memenuhi *due date* nya serta memiliki tingkat utilitas sumber daya yang maksimum. Kelebihan dari metode ini adalah adanya beberapa kriteria optimalisasi yang dapat tercakup langsung dengan metode tersebut, yaitu minimasi *flow time*, pemenuhan *due date*, maksimasi

utilitas sumber daya. Kelemahannya adalah bahwa metode ini lebih rumit perhitungan serta masih belum mampu mendeteksi kriteria optimalisasi minimasi ongkos produksi.

4. Metode Forced

Metode ini merupakan metode penjadwalan dimana ada satu atau beberapa kegiatan yang dipaksakan untuk dilaksanakan pada suatu waktu tertentu sehingga tidak dapat lagi ditukar atau diganti dengan operasi yang lainnya. Kegiatan yang dipaksakan tersebut, misalnya memerlukan mesin produksi khusus yang harus sewa dengan harga yang sangat mahal tiap sama waktunya, sehingga semua operasi yang dibutuhkan mesin tersebut dipaksakan untuk dikerjakan pada waktu mesin tersebut disewa. Kelebihan metode ini adalah kemampuannya untuk mengatasi beberapa permasalahan yang sering dihadapi oleh *shop floor* terutama adanya jadwal khusus yang harus dikerjakan pada satuan waktu tertentu. Kelemahannya adalah bahwa dengan memaksakan beberapa bagian didalam penjadwalan maka diperoleh suatu penjadwalan produksi yang optimal.