Bab 2
Tinjauan Pustaka

2.1. Definisi Penjadwalan

Penjadwalan adalah kegiatan pengalokasian sumber-sumber atau mesin-mesin yang ada untuk menjalankan sekumpulan tugas dalam jangka waktu tertentu. (Baker,1974). Penjadwalan produksi adalah suatu kegiatan memasukkan sejumlah produk yang telah direncanakan ke dalam proses pengerjaannya (John E Biegel,1992). Penjadwalan adalah proses pengurutan pembuatan produk secara menyeluruh pada beberapa mesin (Conway,et,al,1967). Penjadwalan juga didefinisikan sebagai rencana pengaturan urutan kerja serta pengalokasian sumber, baik waktu maupun fasilitas untuk setiap operasi yang harus diselesaikan (Vollman,1998). Dari beberapa definisi yang telah disebutkan maka dapat ditarik satu definisi "Penjadwalan adalah suatu kegiatan perancangan berupa pengalokasian sumber daya baik mesin maupun tenaga kerja untuk menjalankan sekumpulan tugas sesuai prosesnya dalam jangka waktu tertentu".

2.2. Penjadwalan dan Fungsi Manajemen

Keputusan mengenai penjadwalan merupakan kepentingan sekunder dalam keputusan manajemen secara umum. Secara umum keputusan dasar manajemen diidentifikasikan melalui beberapa pertanyaan berikut;

1. Produk atau jasa apakah yang ingin disajikan?
2. Pada skala berapakah produk atau jasa dimaksud disajikan?
3. Sumber daya apa yang digunakan untuk membuatnya?

Ketiga pertanyaan di atas di tentukan jawabannya oleh fungsi *planning* dari manajemen. Kaitannya dengan penjadwalan bahwa penjadwalan akan menjadi relevan ketika ketersediaan sumber daya telah ditentukan. Jadi penjadwalan dan fungsi manejerial dalam hal ini *planning* tidak dapat dipisahkan. Ada interaksi dua arah antara fungsi *planning* dengan penjadwalan, dimana fungsi *planning* menentukan jumlah sumber daya yang harus digunakan untuk memproduksi

barang/jasa dan penjadwalan akan mengevaluasi kebutuhan sumber daya yang yang digunakan untuk berproduksi. Sehingga fungsi *planning* dapat mengambil keputusan yang berhubungan dengan efisiensi dan efektifitas produksi.

2.3. Masalah Penjadwalan

Masalah penjadwalan muncul karena adanya keterbatasan waktu, tenaga kerja, jumlah mesin, sifat dan syarat pekerjaan yang akan dilaksanakan. Secara umum ada dua permasalahan utama yang akan diselesaikan melalui penjadwalan, yaitu penentuan pengalokasian mesin yang akan digunakan untuk menyelesaikan suatu proses produksi dan pengurutan waktu pemakaian mesin tersebut.

Masalah penjadwalan dapat ditinjau dari berbagai aspek diantaranya;

a. Mesin (terbagi atas penjadwalan mesin tunggal, penjadwalan dua mesin, dan penjadwalan m mesin)
b. Aliran proses (terbagi atas *job shop* dan *flow shop*). Aliran proses *job shop* memungkinkan pekerjaan melalui lintasan yang berbeda antar jenisnya. Sedangakan aliran *flow shop* sebaliknya.
c. Pola kedatangan pekerjaan, secara statis maupun dinamis. Dimana jika semua pekerjaan datang secara bersamaan dan semua fasilitas tersedia pada saat kedatangan pekerjaan disebut pola kedatangan pekerjaan statis. Sedangkan jika pekerjaan datang secara acak selama masa penjadwalan disebut pola kedatangan pekerjaan dinamis.
d. Elemen penjadwalan, mengenai ketidakpastian pada pekerjaan dan mesin. Terdiri dari elemen penjadwalan deterministik dan elemen penjadwalan stokastik. Jika elemen penjadwalannya deterministik, maka terdapat kapasitas tentang pekerjaan dan mesin, misalnya tentang waktu kedatangan, waktu *set up* dan waktu proses. Sebaliknya jika tidak terdapat kepastian mengenai pekerjaan dan mesin, maka disebut elemen penjadwalan stokastik.

2.4. *Input* Kegiatan Penjadwalan

Dalam melakukan aktivitas penjadwalan diperlukan *input* berupa kebutuhan kapasitas dari *order-order* yang akan dijadwalkan baik itu jenis serta jumlah sumber daya yang digunakan. Informasi ini dapat diperoleh dari:

- ✓ Lembar kerja operasi yang berisi keterampilan dan peralatan yang dibutuhkan, serta waktu standar pengerjaan.
- ✓ Juga dari *Bill of Material* (BOM) yang berisi kebutuhan-kebutuhan akan komponen, sub komponen dan bahan pendukung.
- ✓ Catatan terbaru mengenai status tenaga kerja, peralatan yang tersedia, yang akan berpengruh pada kualitas keputusan penjadwalan yang diambil.

2.5. *Output* Penjadwalan

Penjadwalan yang dibuat akan menghasilkan jadwal aktivitas-aktivitas sebagai berikut:

1. Pembebanan, meliputi penyesuaian kebutuhan kapasitas untuk *order-order* yang diterima dengan kapasitas yang tersedia melalui penugasan pesanan pada fasilitas, operator dan peralatan tertentu.
2. Pengurutan, yakni berupa penugasan tentang *order-order* mana yang diprioritaskan untuk diproses terlebih dahulu jika suatu fasilitas harus memproses banyak *job*.
3. Prioritas *job*, yakni berupa prioritas kerja tentang *order-order* mana yang diseleksi dan diprioritaskan untuk diproses.
4. Aktivitas pengendalian kinerja penjadwalan yang dilakukan melalui peninjauan kembali status *order-order* pada saat melalui sistem tertentu serta mengatur kembali urut-urutannya.
5. *Up-dating* jadwal, dilakuakan sebagai refleksi kondisi operasi yang terjadi dengan merevisi prioritas-prioritas.

2.6. Manfaat Penjadwalan

Banyak manfaat yang dapat diambil melalui kegiatan penjadwalan. Khususnya untuk di lantai produksi, penjadwalan akan mampu memberikan:

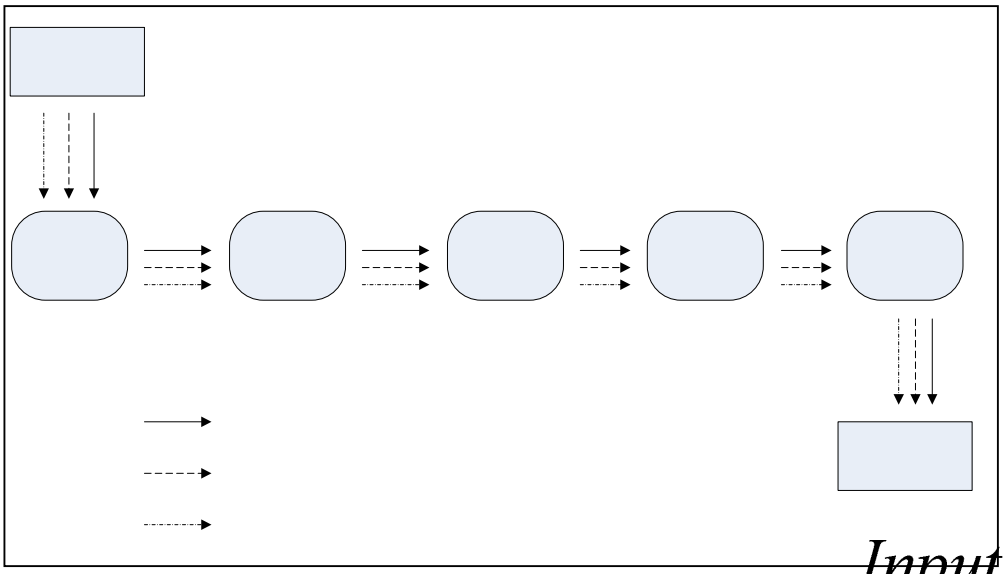
1. Peningkatan produktivitas melalui minimasi waktu menganggur mesin.
2. Penigkatan effisiensi pemakaian fasilitas peralatan, mesin dan sumber daya manusia.
3. Acuan informasi dalam mengestimasi kemampuan perusahaan dalam menyelesaikan *order* konsumen.
4. Kontribusi penting dalam pengendalian produksi guna mencapai pemenuhan target produksi.
5. Minimasi keterlambatan batas waktu penyelesaian pesanan (*due date*) melalui:
    a. Minimasi jumlah pekerjaan yang terlambat
    b. Minimasi maksimum waktu keterlambatan

2.7. Jenis-jenis Aliran Proses Produksi

Adapun jenis-jenis aliran proses menurut Baker, yang secara umum dimiliki banyak perusahaan yaitu:

a. Aliran proses *flow shop*

   Yaitu lantai produksi yang memroses produknya dengan urutan proses yang sama terhadap semua komponen produk yang bersangkutan dari mulai bahan awal sampai produk selesai. Jadi setiap pekerjaan yang telah diproses pada suatu mesin dan kemudian sedang diproses pada mesin yang lain tidak dapat diproses kembali pada mesin yang telah dilalui sebelumnya. Lebih jelas aliran proses *flow shop* dapat dilihat pada gambar 2.1.



Gambar 2.1. Gambar aliran proses *flow shop*
(sumber : Kenneth R. Baker, *Introduction to sequencing and Scheduling,1974, hal 136*)

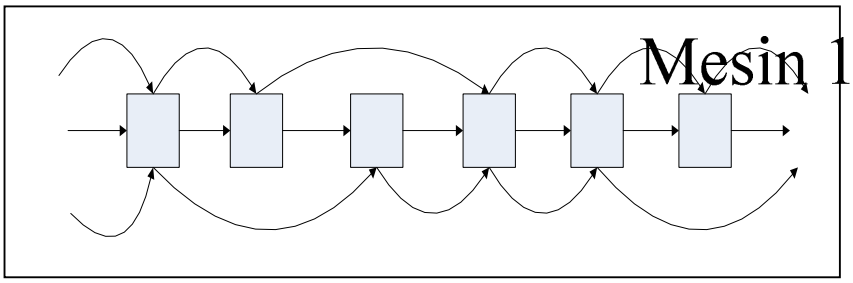
Adapun variasi dari aliran proses *flow shop* yaitu:

1. *Simple Flow Shop*

   Semua jenis pekerjaan melalui urutan proses yang sama, seperti terlihat pada gambar 2.1.

2. *Skip Flow Shop*

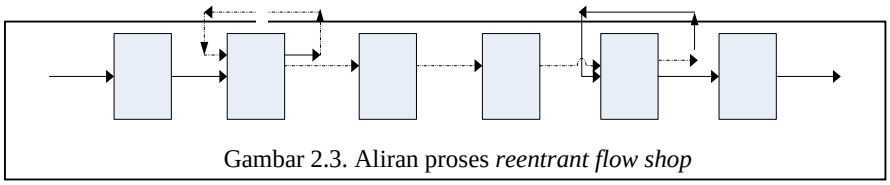
   Aliran pekerjaan pada jenis aliran proses ini cenderung melalui urutan proses yang sama, tetapi ada beberapa pekerjaan yang tidak diproses pada mesin-mesin tertentu. Lebih jelasnya dapat di lihat pada gambar 2.2.



Gambar 2.2. Gambar aliran proses *skip flow shop*

3. *Reentrant flow shop*

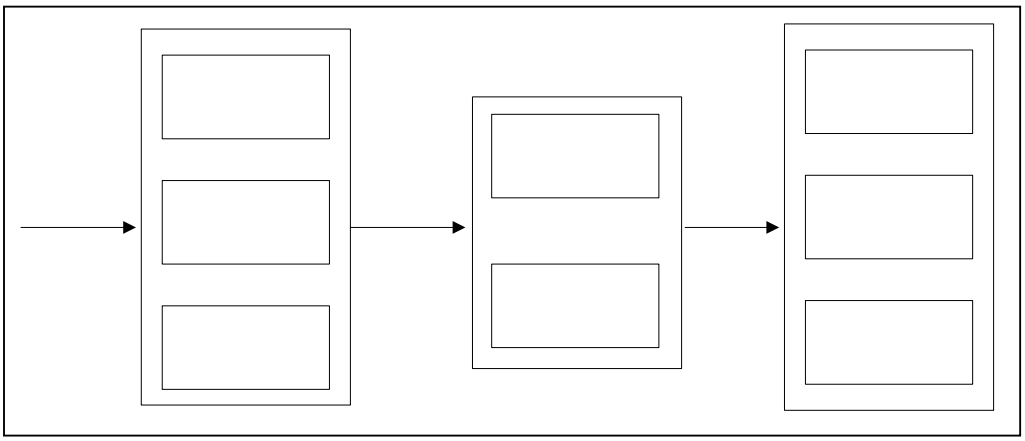
   Yakni aliran proses dimana terdapat penggunaan satu atau beberapa mesin lebih dari sekali dalam membuat produk dimaksud. Lebih jelasnya dapat dilihat pada gambar 2.3.



Gambar 2.3. Aliran proses *reentrant flow shop*

4. *Compound Flow Shop*

   Yakni aliran proses yang memuat kelompok jenis mesin pada setiap tahap prosesnya. Kelompok mesin biasanya berupa mesin mesin paralel seperti terli pada gambar 2.4.



Gambar 2.4. Aliram proses *compound flow shop*

b. Aliran proses *Job Shop*

   Merupakan proses transformasi dimana produk dibuat atas dasar pesanan dalam jumlah tertentu. Setiap *order* dapat mempunyai urutan dari jumlah *lot* atau *batch* atau mesin yang berbeda. Dalam lantai produksi stasiun kerja dan departemen dikelompokkan berdasarkan fungsinya. Kerena setiap *order* dapat mempunyai urutan dari jumlah *lot* atau *batch* atau mesin yang berbeda, maka memungkinkan setiap stasiun kerja memproses beberapa *item* yang berbeda. Artinya bahwa pekerjaan diperbolehkan untuk diproses lebih dari satu kali pada mesin yang sama. Jika setiap mesin hanya dilalui satu kali, disebut aliran proses *job shop* murni dan sebaliknya disebut aliran proses *job shop* umum.

2.8. Definisi Istilah Dalam Penjadwalan

Dalam penjadwalan juga dikenal beberapa istilah diantaranya:

1. *Processing Time* adalah peramalan perkiraan lamanya waktu yang diperlukan untuk menyelesaikan sebuah tugas. *Processing Time* untuk tugas i dinotasikan dengan t_i. $_i$ menyatakan tugas ke $_i$
2. *Due Date* adalah batas waktu penyerahan produk yang dijanjikan kepada pelanggan, dinotasikan dengan d_i.
3. *Lateness* adalah penyimpangan *completion time* dan *due date* sebuah tugas, dinotasikan dengan L_i.
4. *Completion time* adalah rentang waktu antara awal pekerjaan pada tugas pertama, disaat t(waktu) = 0, dan waktu ketika sebuah tugas $_i$ diselesaikan, dinotasikan dengan C_i.
5. *Tardiness* adalah nilai keterlabatan sebuah tugas. Akan bernilai positif jika tugas terlambat dan jika bernilai negatif tugas dinyatakan *early*, dinotasikan dengan T_i.
6. *Early* adalah suatu nilai keterlambatan yang menyatakan bahwa tugas diselesaikan sebelum *due date*-nya.
7. *Slack* adalah sisa waktu antara *due date* dan *Processing Time* sebuah tugas.
8. *Flow time* adalah rentang waktu antara titik dimana sebuah tugas siap dikerjakan dan titik saat selesainya. Merupakan hasil penjumlahan *processing time* dan waktu tunggu tugas sebelum dikerjakan, dinotasikan dengan F_i.
9. *Makespan* waktu penyelesaiaan semua tugas.
10. *Heuristic* adalah suatu prosedur pemecahan masalah untuk menghasilakan hasil yang baik tetapi tidak menjamin hasil yang optimal.
11. *Ready Time* menunjukkan saat suatu pekerjaan (*job*) dapat dikerjakan atau siap dijadwalkan.

2.9. Pembatasan Penjadwalan

Pada umumnya ada dua jenis pembatas penjadwalan yaitu:

1. Pembatas teknologi

   Pembatas ini membentuk urutan proses pada setiap tugas atau dengan kata lain memberikan *routhing* untuk setiap tugas.

2. *Precedence Constrain* (pembatas prioritas)

   Pembatas ini membatasi urutan pemrosesan operasi-operasi dalan suatu tugas.

2.10. Pengukuran *makespan* Sebagai Kriteria Kinerja Penjadwalan

Beberapa cara yang digunakan untuk meminimasi *makespan* adalah:

a. Menggunakan algoritma *McNaughton's* untuk mesin paralel dengan tugas yang berbeda-beda
b. Menggunakan algoritma *HU'S* untuk mesin paralel dengan tugas yang sama
c. Menggunakan algoritma *Muntz-Coffman* untuk dua mesin paralel dengan pengurutan bebas untuk semua tugas yang ada.
d. Menggunakan algoritma *Johnson's* untuk persoalan *flow shop* dengan dua mesin seri.
e. Menggunakan algoritma branch and bound
f. Menggunakan algoritma *Cambel,Dudeck&Smith* untuk persoalan *flow shop* dengan m mesin seri.

2.11. Beberapa Masalah Penjadwalan *Flow Shop* Berdasarkan Jumlah Mesin yang Digunakan dan Cara Penyelesaiaannya.

1. Penjadwalan N *job* pada 2 mesin

   Dalam menyelesaiakan permasalahan ini algoritma yang sering digunakan adalah algoritma johnson yang dikembangkan oleh Selma Johnson dengan kriteria minimasi *makespan*. Adapun prosedur pengurutan pekerjaan adalah sebagai berikut:

   Langkah 1 : Pilih waktu proses terpendek dari semua *job* yang diporoses pada kedua mesin.

Langkah 2 : Jika waktu proses terpendek berada pada mesin I, maka tugaskan *job* tersebut pada urutan pertama. Jika waktu proses terpendek berada pada mesin II, maka tugaskan *job* tersebut pada urutan terakhir.

Langkah 3 : *Job* yang sudah terpilih dikeluarkan dari proses pemilihan dan ulangi langkah I dan II sampai semua *job* mendapat urutan.

2. Penjadwalan N *job* pada 3 mesin

   Penjadwalan ini biasanya ditemui pada lantai produksi dengan aliran proses *flow shop*. Untuk menyelesaikannya digunakan pengembangan algoritma Johnson yang digunakan pada persoalan N *job* pada 2 mesin. Dengan asumsi yang harus dipenuhi bahwa waktu proses minimum dari seluruh pekerjaan (*job*) yang dibebankan pada mesin I harus lebih besar atau sama dengan waktu proses maksimum dari seluruh pekerjaan (*job*) yang dibebankan pada mesin II.

2.12. Algoritma Hybrid untuk Penjadwalan *Flow Shop*

Algoritma Hybrid merupakan hasil pengembangan dari Srinivasan (12) sebagai perbaikan pada programa dinamis yang dikembangkan untuk pengembangan masalah $\overline{T}$ (*mean tardy*) pada penjadwalan mesin tunggal. Secara khusus Algoroitma hybrid sebenarya dikembangkan dengan dua kehandalan yang ditemukan oleh Emmons (4). yakni:

A_i : Susunan *job-job* yang telah ditunjukkan untuk mengikuti *job* i dalam pengurutan optimal.

A'_i : komplemen dari susunan A_i

B_j : Susunan *job-job* yang ditunjukkan untuk membatasi *job* i dalam pengurutan optimal.

Dengan menggunakan notasi *ipj* jika : $t_i < t_j$ atau $t_i = t_j$ dan $d_i = d_j$.

σ melambangakan suatu permutasi parsial (urutan parsial dari tugas-tugas) dan σ_i melambangakan permutasi parsial setiap i dengan catatan sampai akhir σ. Juga $q(\sigma,k)$ melabangkan maksimum *completion time* pada mesin k untuk pengurutan parsial σ. Nilai ini biasanya diperoleh melalui hubungan berulang dari $q(\sigma_i,k)$ =

max{ q(σ,k-1), q(σ,k)}+t_{ik} untuk k =1,2,,,m dimana untuk melengkapinya kita memilih q(σ_i,0)= q(ϕ,k)=0.

Suatu sifat untuk masalah *flow shop* yang telah diperkenalkan adalah sifat 6.3. yang berisi perkiraan pengurutan parsial $\sigma^{(1)}$ dan $\sigma^{(2)}$ yang berisi *job* yang sama (dalam order yang berbeda). "Jika $q_2^{(1)} \le q_2^{(2)} dan\_ q_3^{1} \le q_3^{2}$ kemudian $\sigma^{(1)}$ mendominasi $\sigma^{(2)}$ maka $\sigma^{(2)}$ tidak perlu dipertimbangkan dalam pencarian solusi optimum". Hal ini secara langsung menjadikan sifat ini sebagai alat pengeliminasi dalam pencarian jadwal yang optimal. Pada *stage* k, dihasilkan seluruh urutan parsial yang tidak mendominasi ukuran k, dan sifat 6.3. telah diaplikasikan untuk menetukan pasangan dari urutan parsial yang berisi *job* yang sama. Kemudian sisa urutan parsial yang tidak mendominasi digunakan untuk format pengurutan parsial pada stage K+1. Catatan, tidak ada pengecekan dominasi pada keseluruhan, suatu metode enumerasi akan membentuk 64 urutan parsial dalam permasalahan 4 *job*. Dengan menggunakan sifat 6.3. dalam masalah ini hanya memerlukan 32 urutan parsial untuk membentuknya.

Perkiraan sekarang adalah π dan π' menggambarkan dua pengurutan parsial didalam *job* yang berbeda dan tidak termasuk *job* dalam urutan parsial yang diberikan . Kemudian format lain dari sebuah sifat dominan adalah

if [relation] then $q(\sigma ij\pi\pi', m) \le q(\sigma ij\pi i\pi', m)$

sebuah hasil seperti ini diindikasikan bahwa *job* i mendominasi *job* j dengan mengacu kepada σ. Dengan kata lain semua uruan yang dimulai dengan pengurutan parsial σj dapat tereliminasi dari pertimbangan dalam pencarian optimasi penjadwalan. Berbagai riset eliminasi kondisi ini diteliti oleh McMahon(12) dan oleh Szware(16) orang yang menemukan sifat di bawah ini.

$\Delta k = q(\sigma ij, k) - q(\sigma j, k)$

sifat 6.4. if $\Delta_{k-1} \le \Delta k \le t_{ik}$ untuk semua k, $2 \le k \le m$ then $q(\sigma ij\pi\pi', m) \le q(\sigma j\pi i\pi', m)$.

Jadi *job* i mendominasi *job* j dengan memperhatikan σ.

Suatu catatan hubungan dari seluruh persamaan diberikan di bawah ini:

1. $\Delta_{k-1} \le \Delta_k \le t_{ik}$ $\quad 1 \le k \le m$
2. $\Delta_{k-1} \le t_{ik} \_ dan \_ \Delta_k \le t_{ik}$ $\quad 1 \le k \le m$
3. $\max \{\Delta_1, \Delta_2, .., \Delta_k\} \le t_{ik}$ $\quad 1 \le k \le m$
4. $\Delta_k \le \min\{t_{ik,.....} t_{im}\}$ $\quad 1 \le k \le m$
5. $\Delta_{k-1} \le \Delta_k \le \min\{t_{ik}, ...t_{im}\}$ $\quad 1 \le k \le m$

Hanya salah satu dari lima format yang digunakan dalam solusi algoritma, dan pilihan harus berdasarkan pada keamanan (untuk suatu versi perhitungan dari metode di maksud).

Prosedur dasar dengan menggunakan sifat 6.4. dipengaruhi dengan algoritma berikut:

1. Untuk menghasilkan urutan parsial ambil *job-job* yang tidak dimasukkan dalam urutan menjadi inisial kadidat untuk urutan berikutnya.
2. Pilih i yang tidak sama dengan j menjadi calon *job* yang akan dijadwalkan
3. Bandingkan urutan σij dan urutan σj menggunakan sifat 6.4 untuk menentukan apakah *job* i dominan *job* j dengan memperhatikan σ. Jika *job* i mendominasi *job* j, pindahkan *job* j dari kandidat penyusunan dan ke langkah 4
4. Ulangi langkah 2 untuk semua *job* j lainnya yang menjadi kandidat penyusunan
5. Ulangi langkah 1 untuk semua *job* i yang pada awalanya termasuk kandidat penyusunan
6. Setiap sisa *job* i yang menjadi kandidat disusun setelah melakukan *loop* dalam 5 langkah digunakan untuk penambahan format pengurutan parsial, σi. Untuk setiap penambahan urutan parsial ulangi dari langkah 1 lagi.

Jika sifat 6.3. dimasukkan dalam pengecekkan dominasi pada setiap stage akan menurunkan jumlah pengurutan parsial.

if [relation] then $q(\sigma ij\pi\pi', m) \le q(\sigma ij\pi i\pi', m)$

setiap bagian yang terpengaruh dalam inisial urutan parsial σi, dan setiap urutan yang telah ditugaskan pada *job* j pada posisis terakhir tidak perlu dimasukkan dalam pencariuan optimum. Menggunakan dasar pemikiran Szware (17) yang diperoleh dari tambahan sifat dominan sifat 6.5. Jika $\max\{\Delta_1, \Delta_2, ..., \Delta_m\} \le t_{im}$ then $q(\sigma ij\pi\pi', m) \le q(\sigma ij\pi i\pi', m)$.

Jadi *job* j didominasi pada posisi terakhir dengan mengacu pada σi. Karena hasil ini menyiapkan perbedaan kriteria eliminasi dibandingkan sifat 6.4. Szware merekomendasikan dua cara yang dilakukan sebagai lawan pemgimplementasian suatu algoritma eliminasi. Menandakan perbedaan antara jenis *state* dominan diatas dan jenis dominan yang digunakan dalam algoritma hybrid dalam bagian 4 bab3. Tujuan yang akan dicapai adalah untuk meminimasi *makespan*.

Adapun arti dari setiap notasi yang digunakan sebagai berikut:

σ : urutan parsial dari setiap job yang akan dijadwalkan.

q(σ,k) : maksimum *completion time* pada mesin k untuk pengurutan parsial σ

k : urutan proses mesin

t_{ik} : waktu proses *job* i di mesin k

$q_2^{(1)}$: *ready time job* 2 di mesin 1

$q_2^{(2)}$: *ready time job* 2 di mesin 2

π : *Job-job* yang belum dijadwalkan

π' : *Job-job* yang telah dijadwalkan

m : jumlah jenis mesin / jumlah tahapan proses

2.13. Algoritma *Ant-Colony Optimization*

Dalam upaya mencapai minimasi *makespan* tidak terlepas dari masalah penjadwalan dalam meminimasi total *flow time* dan *comletion time*. Dalam algoritma ini dikenal terminologi bahwa dengan mengabaikan kondisi pada mesin, proses yang tidak simultan dari *job* dan tidak ada *job* terlewat, *completion time* dari jadwal parsial σa pada tahap *j* dapat diperoleh dari persamaan berikut:

$$q(\sigma a, j) = \max[q(\sigma a, j), q, (\sigma a, j-1)] + t_{aj} \quad \text{.....(2.4)}$$

Dimana $q(\sigma a,0) = 0$ dan $q(\phi, j) = 0$, j = 1,2,...,m; ϕ adalah inisial jadwal yang tidak berlaku. Total *flow time job* dalam σa didapatkan melalui persamaan

$$F\sigma a = F\sigma + q(\sigma a, m) \quad \text{.....(2.5)}$$

Adapun prinsip kerja dari *algoritma ant-colony optimization* sebagai berikut:
Jika *job* a tidak harus menunggu untuk diproses pada suatu tahap, *completion time* dari *job* a pada tahap m, saat *job* a mengikuti *job* b (*job* terakhir dalam jadwal parsial σ) diperoleh melalui persamaan:

$$q(\sigma a, m) = q(\sigma,1) + \sum_{j=1}^{m} t_{aj} \quad \text{.....(2.6)}$$

Namun jika *job* a harus menuggu sebelum diproses pada satu tahap, maka *completion time* dari *job* a diperoleh melalui persamaan:

$$q(\sigma a, m) = q(\sigma,1) + W_{ba} + \sum_{j=1}^{m} t_{aj} \quad \text{.....(2.7)}$$

Dimana W_{ba} diperoleh dari:

$$\sum_{j=2}^{m} \max[q(\sigma, j) - q(\sigma a, j-1), 0] \quad \text{.....(2.8)}$$

Dengan demikian didapatkan persamaan

$$q(\sigma a, m) = \sum_{i=1}^{n'} t_{[i]1} + W_{ba} + \sum_{j=1}^{m} t_{aj} \quad \text{.....(2.9)}$$

Jika semua *job* telah dijadwalkan, total *flow time* dari semua *job* dalam jadwal σ dihitung dengan persamaan:

$$F\sigma = \sum_{i-1}^{n-1} (n-i) t_{[i]1} + \sum_{i=2}^{n} W_{[i-1][i]} + \sum_{i=1}^{n} \sum_{j=1}^{m} t_{ij} \quad \text{.....(2.10)}$$

Dimana $W_{[i-1][i]}$ menunjukkan jumlah waktu mengunggu *job* (i) dalam berbagai tahapan saat mengikuti *job* (i-1) dalam jadwal σ. Dengan menggunakan aturan (*sort processing time)* SPT maka akan meminimasi total *flow time* dan akan meminimasi *makespan*.

Adapun arti dari setiap notasi yang digunakan sebagai berikut:

n	: jumlah *job* yang dijadwalkan
m	: jumlah tahapan proses / jumlah jenis mesin
σ	: urutan parsial dari setiap job yang akan dijadwalkan
n'	: jumlah job dalam urutan parsial (σ)
t_{ij}	: waktu proses job i pada tahap j
π	: *job-job* yang belum dijadwalkan
a,c	: *job-job* dalam susunan π
$q(\sigma, j)$	: *completion time* jadwal parsial σ pada jenis mesin j
$q(\sigma a, j)$	: *completion time* jadwal parsial σa pada jenis mesin j saat *job* a dimasukkan pada jadwal parsial σ
W_{ba}	: jumlah waktu menunggu *job a* sebelum berbagai tahap saat mengikuti *job b* dalam σ