

## BAB 3

### LANDASAN TEORI

#### 3.1. Pengertian Sistem Produksi

Sistem produksi adalah sekumpulan orang, peralatan, dan prosedur yang terorganisir untuk menyelesaikan operasi manufaktur pada sebuah perusahaan.

Menurut Groover (2001), ada beberapa tipe sistem produksi pada operasi manufaktur, yaitu :

##### a. *Low Quantity Production*

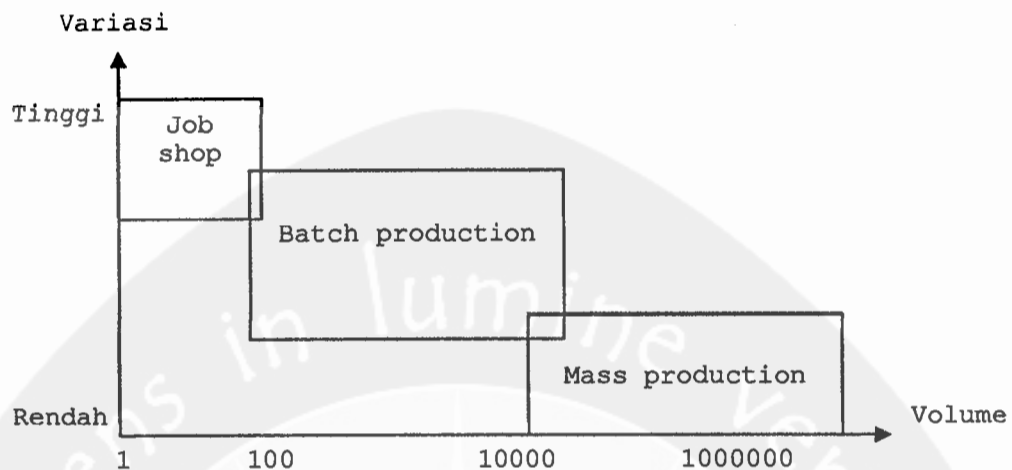
Tipe fasilitas produksi ini biasanya memiliki kuantitas range dari 1 sampai 100 unit/tahun, biasa disebut *job shop*, di mana membuat kuantitas produksi rendah namun variasi produk tinggi. Order konsumen untuk item-item sering bersifat khusus, dan order ulang tidak pernah terjadi. Peralatan pada *job shop* bersifat *general purpose* dan kemampuan pekerjanya *highly skilled*.

##### b. *Medium Quantity Production*

Kuantitas range pada tipe fasilitas produksi ini antara 100 sampai 10000 unit/tahun. Tipe fasilitas produksi ini biasa disebut *batch production*. *Batch production* biasanya digunakan pada situasi *make to stock*. Tipe produksi ini berada di antara *job shop* dan *flow shop*.

##### c. *High Production*

Kuantitas range pada tipe fasilitas produksi ini diatas 10000 unit/tahun. Tipe fasilitas produksi ini disebut *mass production*. Variasi produk pada *mass production* rendah namun kuantitas produksi sangat tinggi sehingga sistem produksi ini bersifat *make to stock*.



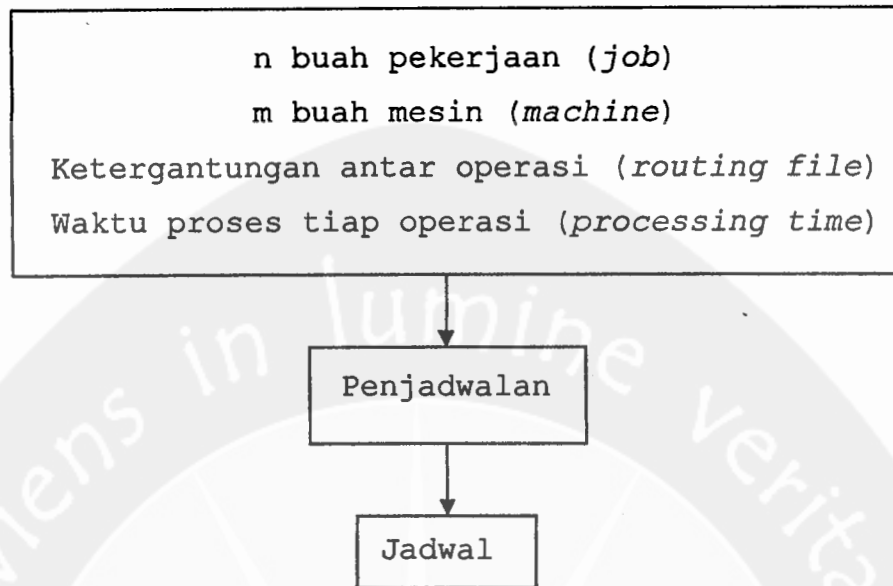
**Gambar 3.1. Sistem produksi menurut Groover (2001)**

### 3.2. Penjadwalan Produksi Secara Umum

Penjadwalan dapat diartikan sebagai proses pengalokasian sumber daya yang terbatas untuk memilih tugas dalam jangka waktu tertentu (Baker dalam Setiawan, 2003).

Penjadwalan dapat juga merupakan suatu usaha menggunakan waktu agar diperoleh efisiensi terhadap waktu yang sesuai dengan kebutuhan. Permasalahan dalam penjadwalan adalah bagaimana mengalokasikan pekerjaan ke mesin, pada kondisi mesin mempunyai kapasitas dan jumlah yang terbatas.

Secara umum, penjadwalan dapat dinyatakan sebagai persoalan seperti pada gambar 3.2.



**Gambar 3.2. Persoalan Penjadwalan**

Penjadwalan produksi mempunyai beberapa tujuan umum, antara lain :

1. Mengurangi terjadinya penumpukan barang setengah jadi pada lintasan produksi (*waiting line*).
2. Meningkatkan produktivitas mesin dengan mengurangi waktu tunggu mesin (*idle time*).
3. Mengurangi keterlambatan karena waktu proses suatu pekerjaan telah melampaui jatuh temponya.
4. Mengurangi waktu siklus produksi sehingga biaya produksi dapat ditekan.

Ada beberapa istilah dalam penjadwalan, diantaranya adalah :

1. *Start Time*

Start time adalah waktu *job* mulai dikerjakan.

2. *Completion Time*

*Completion time* adalah waktu *job* selesai dikerjakan.

3. *Processing Time*

Waktu proses adalah penilaian peramalan terhadap berapa lama (waktu) untuk menjadikan suatu pekerjaan itu lengkap.

4. *Due date*

Adalah batas waktu penyelesaian pekerjaan yang tidak dapat dilanggar, apabila dilanggar maka akan benar-benar diperhitungkan kelambatannya, dan dianggap sebagai hukuman atas keterlambatan yang terjadi.

5. *Lateness*

Adalah selisih antara waktu penyelesaian dari pekerjaan dengan *due date* nya. Jika bernilai positif maka pekerjaan itu terlambat.

6. *Earliness*

Adalah selisih antara *due date* dengan saat selesai untuk pekerjaan yang selesai lebih awal.

7. *Slack Time*

Ukuran dari perbedaan antar sisa waktu dari *due date* dan *processing time*.

8. *Makespan*

Jika  $C_i$  merupakan *completion time* dari operasi terakhir *job i* dalam *batch n job*. Maka *makespan* adalah  $C_{max} = \max (C_1, C_2, \dots, C_n)$

*Makespan* adalah *completion time job* terakhir.

Permasalahan yang biasa dilakukan adalah meminimasi  $C_{max}$ , atau meminimasi *makespan*. Kriteria ini biasanya digunakan untuk mengukur *level utilitas* dari mesin.

### 9. *Total Completion Time*

*Total completion time* adalah jumlah semua *completion time job-job*. Biasanya disebut sebagai *flow time*.

### 10. *Waiting Time*

*Waiting time* adalah waktu dari kedatangan sampai proses pengerjaannya dimulai.

Beberapa permasalahan yang sering diambil adalah :

1. Minimasi *mean waiting time*
2. Minimasi *maximum waiting time* :  $\text{Min}(\text{max}(W_i))$

### 3.3. Model-model dalam Penjadwalan

Salah satu karakteristik utama dalam model penjadwalan adalah konfigurasi mesin-mesinnya. Ada beberapa konfigurasi mesin yang penting (Pinedo, 2005).

#### 1. *Single Machine Model*

Banyak sistem produksi mengikuti model mesin tunggal. Sebagai contoh, jika ada satu *bottleneck* di dalam sebuah lingkungan dengan banyak mesin, maka urutan *job* pada *bottleneck* secara khas menentukan performansi dari seluruh sistem. Pada kasus seperti ini, wajar untuk menjadwalkan *bottleneck* terlebih dahulu dan seluruh operasi lain (*upstream* dan *downstream*) dijadwalkan setelah itu.

#### 2. *Parallel Machine Model*

Banyak lingkungan produksi terdiri dari beberapa stage atau *work center*, setiap *work center* dengan sejumlah mesin paralel. Mesin pada *work center* bisa identik, sehingga *job* dapat diproses pada salah satu mesin yang tersedia. Jika salah satu *work center* merupakan

*bottleneck*, maka *bottleneck* dapat dimodelkan ke dalam mesin paralel dan dianalisis secara terpisah.

### 3. *Flow Shop Model*

Banyak industri dan perakitan yg menentukan pekerjaan harus mengerjakan beberapa operasi pada sejumlah mesin yang berbeda. Jika rute semua *job* bersifat serupa contoh semua *job* melewati mesin-mesin yang sama dalam urutan yang sama, lingkungan ini disebut *flow shop*. Mesin disetup secara series.

### 4. *Job Shop Model*

Dalam multi operasi, *job* sering kali memiliki rute yang berbeda. Lingkungan semacam ini disebut *Job shop*. *Job shop* yang paling sederhana mengasumsikan bahwa *job* boleh diproses pada mesin tertentu paling banyak pada rutanya melalui sistem. Atau sebuah *job* boleh diproses pada sebuah mesin beberapa kali pada rutanya melalui sistem tersebut.

### 5. *Suppy Chain Model*

Beberapa model biasa mengasumsikan lingkungan produksi terdiri dari jaringan fasilitas yang terhubung dengan setiap fasilitas sebagai (*flexible*) *flow shop* atau sebagai (*flexible*) *job shop*.

## 3.4. *Forward Scheduling dan Backward Scheduling*

Menurut Pinedo (2005), beberapa sistem umum bergantung pada prosedur yang lebih rinci, seperti *forward loading* dan *backward loading*.

### 1. *Forward Loading*

Biasa juga disebut *forward scheduling*, pekerjaan disisipi dimulai dari awal penjadwalan (Pinedo, 2005).

*Forward scheduling* adalah pengurutan pekerjaan dengan bertolak dari arah waktu nol (*zero time*) dan bergerak menuju masa yang akan datang sesuai dengan arah pergerakan waktu. Pada penjadwalan maju ini biasanya akan dihasilkan jadwal yang dapat dijalankan (*feasible*) tetapi jadwal tersebut mungkin tidak memenuhi *due date* yang wajar (Halim dalam Lesmana, 2003).

## 2. *Backward Loading*

Biasa juga disebut *backward scheduling*, Penjadwalan dimulai dari akhir penjadwalan, dari *due date* (Pinedo, 2005).

*Backward scheduling* adalah penjadwalan yang dimulai dari *due date*, mundur ke waktu nol atau *zero time*. Pada penjadwalan mundur, pekerjaan terakhir dalam urutan pekerjaan dilaksanakan pertama kali yang kemudian dilanjutkan pada pekerjaan yang sebelumnya sampai pekerjaan pertama (Halim dalam Lesmana, 2003).

Penyisipan ini, baik *forward* dan *backward*, dilaksanakan dengan beberapa aturan utama. Beberapa prosedur yang lebih canggih awalnya mengidentifikasi *work center* atau mesin *bottleneck*; Prosedur menghitung dan menentukan *job* mana yang harus diproses pada mesin tersebut dan menjadwalkan *job-job* pada mesin tersebut dengan beberapa prosedur algoritma. Setelah *bottleneck* dijadwalkan, menjadwalkan *job-job* pada mesin lain melalui baik *forward loading* maupun *backward loading*.

## 3.5. Metode-metode dalam Penjadwalan

Dalam melakukan penjadwalan, banyak sekali metode yang dapat digunakan. Beberapa metode yang digunakan

dalam penelitian ini adalah metode SPT (*Shortest Processing Time*), LPT (*Longest Processing Time*), LWKR (*Least Work Remaining*) dan MWKR (*Most Work Remaining*) serta *Earliest Due Date* (EDD), *First Come First Serve* (FCFS), *Most Operation Remaining* (MNOPR), *Random*, *Total Processing Time* (TPT).

- a. SPT (*Shortest Processing Time*) adalah metode yang digunakan untuk meminimasi rata-rata jumlah job yang menunggu untuk diproses. Metode ini mengurutkan dari job dengan waktu proses terendah.
- b. LPT (*longest Processing Time*)  
Job dengan waktu proses yang tinggi akan dikerjakan lebih dahulu. Metode ini bermaksud untuk mengurangi order waktu proses. Ketika ada mesin-mesin paralel, metode ini cenderung untuk menyeimbangkan beban kerja seluruh mesin. Ini menguntungkan dengan menjaga job dengan waktu proses yang pendek untuk dijadwalkan kemudian, karena job-job ini berguna di akhir untuk menyeimbangkan beban kerja (Kemppainen, 2005).
- c. MWKR (*Most Work Remaining*)  
Memilih operasi job dengan suksesor terbanyak.
- d. LWKR (*Least Work Remaining*)  
Memilih operasi dari job dengan suksesor paling sedikit.
- e. FCFS (*First Come First Serve*)  
Memilih operasi yang siap lebih awal.
- f. EDD (*Earliest Due Date*)  
Memilih operasi dengan due date terawal.
- g. MNOPR (*Most Operation Remaining*)  
Memilih operasi dari job dengan suksesor paling sedikit.



*h. Random*

Memilih operasi secara acak (Baker dalam Cendrawati, 2007).

*i. TPT (Total Processing Time)*

*Job* diurutkan dari total proses masing-masing part (pembuatan masing-masing part hingga produk jadi) yang terendah.

### **3.6. Pendekatan *Theory of Constraint***

*Theory of Constraint* adalah sebuah filosofi perbaikan berkelanjutan dimana fokus pada identifikasi dan manajemen kendala untuk mencapai tujuan organisasi.

*Theory of Constraint* dikembangkan oleh Eliyahu Goldratt dan juga disebut *Optimized Production Technology (OPT)*, *synchronous manufacturing*, dan *drum-buffer-rope* oleh beberapa pengarang. *Theory of Constraint* memiliki dua tujuan untuk kebanyakan organisasi bisnis yaitu :

1. Menghasilkan lebih banyak uang sekarang dan masa depan.
2. Membentuk proses *ongoing* atau *continuous improvement* (Tersine, 1994).

Beberapa hal menyatakan *Theory of Constraint* merupakan jenis metode penjadwalan, karena karakter utamanya adalah memperbaiki performansi produksi dengan mengatur beberapa mesin yang penting produksi yang disebut *bottleneck*. *Bottleneck* adalah kunci penting dalam *Theory of Constraint*. Tujuan penjadwalan pada umumnya adalah untuk membuat seluruh resource menjadi tepat guna dan mengurangi biaya produksi. Beberapa mesin penting menentukan efisiensi seluruh lini produksi, membuat

penggunaan yang lebih baik pada *bottleneck* dapat memperbaiki efisiensi seluruh lini produksi menjadi kunci penting dalam *Theory of Constraint* (Hong-yuan, dkk, 2005).

*Theory of Constraint* menggunakan tiga ukuran operasional untuk mengarahkan perusahaan ke tujuan yang hendak dicapainya, yaitu :

1. *Throughput*

Yaitu jumlah uang yang dihasilkan sistem melalui penjualan selama periode tertentu. Jika sistem memproduksi produk yang tidak terjual, itu tidak menghasilkan *throughput*.

2. *Inventory*

Yaitu jumlah uang yang telah ditanamkan sistem terhadap barang-barang yang hendak dijual, termasuk pabrik, properti dan perlengkapannya. Definisi *inventory* ini berbeda dengan definisi yang lama karena tidak memasukkan penambahan nilai tenaga kerja dan *overhead*. Ini meliputi semua *resources* yang digunakan untuk produksi, seperti *plant* dan *equipment* bersama material yang dibeli.

3. *Operating expense*

Yaitu jumlah uang yang dikeluarkan sistem untuk mengubah *inventory* menjadi *throughput* untuk jangka waktu tertentu. *Operating expense* termasuk seluruh biaya selain biaya variabel yang timbul untuk memperoleh *throughput*. *Operating expense* berfokus pada biaya konversi dan tidak termasuk uang yang dikeluarkan untuk pembelian *inventory* (Tersine, 1994).

### 3.7. Masalah Penjadwalan Constraint

*Constraint* dalam sistem produksi mungkin muncul dari kondisi di mana permintaan yang ada melebihi kapasitas tersedia. Dalam literatur *Theory of Constraint*, *Capacity Constraint Resources* (CCR) didefinisikan sebagai setiap *resource* yang membatasi *throughput*.

Ide-ide dalam sistem produksi *Theory of Constraint* memusatkan perhatian pada dua masalah utama, yaitu *product mix* dan *product flow*. Dalam masalah *product mix*, dikatakanlah bahwa dengan adanya *internal constraint* pada sistem, tidak seluruh permintaan dapat diproduksi saat dibutuhkan. Oleh karena itu dengan menggunakan pendekatan *Theory of Constraint*, manajer mencari dan memilih *product mix* untuk memaksimalkan nilai sistem secara keseluruhan yang mampu diproduksi departemen atau bagian yang *constraint* (Simon, dkk dalam Lesmana, 2003).

Penekanan prinsip-prinsip *Theory of Constraint* adalah pada performansi setiap organisasi yang dibatasi oleh *constraint* dan memaksimalkan performansi keseluruhan organisasi membutuhkan maksimasi performansi dari setiap *constraint* pada sistem.

Permasalahan seputar *Theory of Constraint* juga dapat dianalogikan dengan dua orang yang sedang berjalan bersama dengan saling mengikatkan dirinya dengan menggunakan tali, dimana kecepatan orang yang pertama dua kali lebih cepat dari kecepatan orang yang kedua. Jika keadaan ini dibiarkan, maka akan ada dua kemungkinan. Kemungkinan pertama adalah orang pertama akan jatuh karena tertahan berat badan orang yang kedua. Dan kemungkinan kedua adalah orang kedua akan terseret oleh orang pertama. Untuk mengantisipasi keadaan ini, maka

orang pertama harus menyesuaikan kecepatan orang kedua yang dapat dianalogikan sebagai mesin *constraint*. Hal ini bisa diantisipasi juga dengan menambah kapasitas mesin *constraint* apabila belum maksimal, tetapi hal ini berdampak akan menambah *operating expense* atau biaya proses. Usaha untuk menyesuaikan kecepatan mesin-mesin lain dengan mesin *constraint* dilakukan dengan menggunakan *Theory of Constraint* (Russel dan Fry dalam Lesmana, 2003).

### **3.8. Langkah-langkah *Theory of Constraint***

Goldratt memperkenalkan lima langkah dalam *Theory of Constraint* yang berfokus pada proses, yaitu :

1. Identifikasi *constraint*.
2. Mengambil keputusan bagaimana mengeksploitasi *constraint* tersebut.
3. Mensubordinasikan atau menyesuaikan operasi-operasi lainnya terhadap *constraint*.
4. Mengantisipasi *constraint* demi meningkatkan *throughput* atau kapasitas produksi.
5. Secara terus menerus mengembangkan sistem tersebut (Lesmana, 2003).