

BAB II

LANDASAN TEORI

Dalam penyelesaian tugas akhir ini digunakan landasan teori yang berkaitan dengan permasalahan yang digunakan untuk menyelesaikan masalah yang ada pada perusahaan.

2.1 Sistem

Menurut J.A.Hall (2001) definisi sistem adalah sekelompok dua atau lebih komponen yang saling berkaitan (*interrelated*) atau subsistem yang bersatu untuk mencapai tujuan yang sama (*common purpose*). Jika komponen tertentu tidak memberikan kontribusi ke tujuan bersama maka bagian itu bukan bagian dari sistem tersebut.

2.2 Informasi

Menurut O'Brien (2003) informasi adalah suatu data yang telah diproses sehingga memiliki arti dan berguna untuk pengguna serta mengandung kepastian yang akurat.

2.3 Analisa dan Perancangan Sistem

Menurut Cahyono (2013) analisis dan perancangan sistem merupakan kegiatan menganalisa *input* data atau aliran data secara sistematis, memproses atau mentransformasikan data, menyimpan data, dan menghasilkan *output* informasi

dalam bisnis khusus. Analisis dan perancangan sistem digunakan untuk menganalisa, merancang, dan mengimplementasikan fungsi bisnis yang bisa dicapai melalui penggunaan sistem informasi terkomputerisasi.

Analisis dan desain sistem ditujukan untuk menyusun analisa dan perancangan sistem informasi, usaha banyak yang memungkinkan hal tidak terpikirkan diatasi dengan berbagai cara. Hal ini dianggap sebagai serangkaian proses yang secara sistematis untuk meningkatkan bisnis melalui penggunaan sistem informasi terkomputerisasi.

2.4 Penjadwalan Produksi

Menurut Gaspersz (2012) penjadwalan adalah pengurutan pembuatan atau pengerjaan produk secara menyeluruh yang dikerjakan pada beberapa buah mesin. Dengan demikian masalah *sequencing* senantiasa melibatkan pengerjaan sejumlah komponen yang sering disebut dengan istilah 'job'. *Job* sendiri masih merupakan komposisi dari sejumlah elemen-elemen dasar yang disebut aktivitas atau operasi. Tiap aktivitas ini membutuhkan alokasi sumber daya tertentu selama periode waktu tertentu yang sering disebut dengan waktu proses.

Penjadwalan merupakan alat ukur bagi perencanaan agregat. Pesanan-pesanan *actual* pada tahap ini ditugaskan pertama kalinya pada sumber daya tertentu (fasilitas, pekerja, dan peralatan), kemudian dilakukan pengurutan kerja pada tiap-tiap pusat pemrosesan sehingga dicapai optimalitas utilisasi kapasitas yang ada. Pada penjadwalan ini, permintaan produk yang tertentu (jenis dan jumlah) dari *Master*

Production Schedule (MPS) ditugaskan pada pusat pemrosesan tertentu untuk periode harian.

Menurut Tanuwijaya dan Setyawan (2012) penjadwalan produksi (*production scheduling*) secara umum didefinisikan sebagai suatu proses dalam perencanaan dan pengendalian produksi yang merencanakan produksi dan pengalokasian sumber daya pada suatu waktu tertentu dengan memperhatikan kapasitas sumber daya yang ada. Sumber daya yang terbatas dengan sejumlah produksi yang harus dikerjakan menjadi persoalan dalam penjadwalan perencanaan dan pengendalian produksi diperlukan suatu proses penjadwalan.

2.5 Tujuan Penjadwalan

Tujuan dari aktifitas penjadwalan tersebut adalah sebagai berikut:

1. Meningkatkan penggunaan sumber daya atau mengurangi waktu tunggu sehingga total waktu proses dapat berkurang dan produktifitas dapat meningkat.
2. Mengurangi persediaan barang setengah jadi atau mengurangi sejumlah pekerjaan menunggu dalam antrian ketika sumber daya yang ada masih mengerjakan tugas yang lain. Teori *Baker* (2012) mengatakan, jika aliran kerja suatu jadwal konstan maka akan mengurangi rata-rata persediaan barang setengah jadi.
3. Mengurangi beberapa keterlambatan pada pekerjaan yang mempunyai batas waktu penyelesaian sehingga akan meminimisasi *penalty cost* (biaya keterlambatan).

4. Membantu pengambilan keputusan mengenai perencanaan kapasitas pabrik dan jenis kapasitas yang dibutuhkan sehingga penambahan biaya yang mahal dapat dihindarkan.

2.6 Input dan Output Penjadwalan

Dalam buku Pengendalian dan Perencanaan Produksi menurut Ginting (2007) menyebutkan *input* dan *output* penjadwalan sebagai berikut:

2.6.1 Input Penjadwalan

Pekerjaan yang merupakan alokasi kapasitas untuk *order*, penugasan prioritas pekerjaan dan pengendalian jadwal produksi informasi terperinci, informasi tersebut akan menyatakan *input* dari sistem penjadwalan. Kebutuhan- kapasitas dari *order* yang dijadwalkan dalam jumlah dan macam sumber daya yang digunakan. Untuk produk tertentu, informasi ini bisa diperoleh dari lembar kerja operasi dan BOM. Kualitas dari keputusan penjadwalan sangat dipengaruhi oleh ketetapan estimasi *input* tersebut.

2.6.2 Output Penjadwalan

1. Pembebanan (*Loading*)

Pembebanan melibatkan penyesuaian kebutuhan kapasitas untuk *order* yang diterima atau diperkirakan dengan kapasitas yang tersedia. Pembebanan dilakukan dengan menugaskan *order* pada fasilitas, operator, dan peralatan tertentu.

2. Pengurutan (*Sequencing*)

Pengurutan merupakan penugasan tentang *order* mana yang diprioritaskan untuk diproses dahulu bila suatu fasilitas memproses banyak *job*.

2.7 Jenis-jenis Penjadwalan

Ada beberapa jenis penjadwalan menurut Gaspersz (2014) yaitu:

2.7.1 Penjadwalan *Flow Shop*

Penjadwalan *flow shop* merupakan suatu pergerakan unit yang terus-menerus melalui suatu rangkaian stasiun kerja disusun berdasarkan produk. Susunan suatu proses produksi jenis *flow shop* dapat diterapkan dengan tepat untuk produk dengan desain yang stabil dan diproduksi, secara banyak (*volume* produk) sehingga investasi dengan tujuan khusus (*special purpose*) yang dapat secepatnya kembali.

2.7.2 Penjadwalan *Batch*

Banyak dari pabrik dengan jenis *Make To Stock* (MTS) memproduksi produk yang berbeda pada fasilitas yang umum. Sebagai contoh, pabrik minuman ringan mungkin memproduksi beberapa rasa minuman yang berbeda pada satu fasilitas. Perusahaan sabun mungkin mengemas produknya dalam beberapa ukuran yang berbeda pada lintasan pengepakan. Pada kasus seperti ini, produk tersebut umumnya diproduksi dalam ukuran *batch*. Keputusan yang dihadapi oleh manajer produksi dalam sistem produksi *batch* adalah “berapa” jumlah produksi dalam setiap *batch*-nya

berikut dengan urut-urutannya, atau perintah mengenai produk-produk mana saja yang harus dibuat secara *batch*.

2.7.3 Penjadwalan *Job Shop*

1. *Job Shop Loading*

Ketika suatu pesanan (*job shop*), pada kegiatan pertama dari penjadwalan adalah menugaskan pesanan tersebut kepada bermacam pusat kerja untuk diproses.

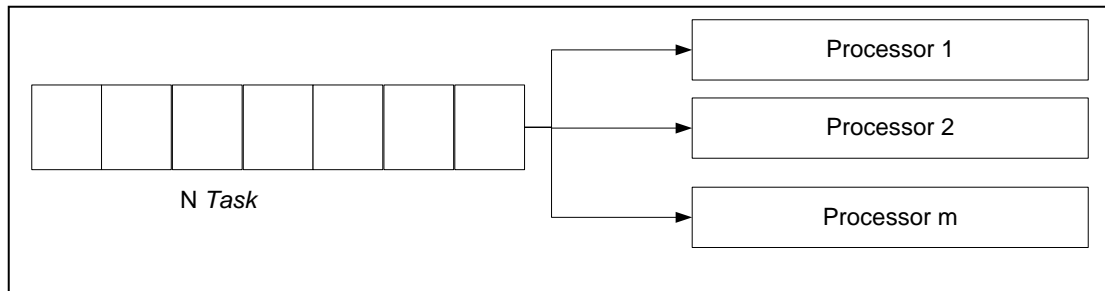
Permasalahan *loading* menjadi lebih sederhana ketika suatu *job* tidak dapat dipisah. Meskipun hal ini sering terjadi, biasanya suatu industri sering dalam prakteknya melakukan pemisahan *job* dan menugaskan bagian-bagian terpisah dari *job* tersebut kepada pusat kerja yang berbeda, untuk tujuan meningkatkan utilisasi sumber daya. Permasalahan ini maka dapat diasumsikan tidak ada pemisah *job* maka *shop loading* dapat dibuat dengan mudah dengan menggunakan *gant chart* dan metode penugasan.

2. *Job Shop Sequencing*

Penjadwalan *job shop* meliputi aturan prioritas *sequencing*. Aturan prioritas *sequencing* diaplikasikan untuk seluruh *job* yang sedang menunggu dalam antrian. Bila pusat kerja untuk satu *job* baru maka *job* dengan prioritas terdahulu diproses. Pemilihan prioritas *sequencing* tersebut mempertimbangkan efisiensi penggunaan fasilitas dengan kriteria antara lain: biaya setup, biaya persediaan, waktu menganggur stasiun kerja, persentase waktu menganggur dan rata-rata jumlah *job* yang menunggu.

3. Penjadwalan *Parallel*

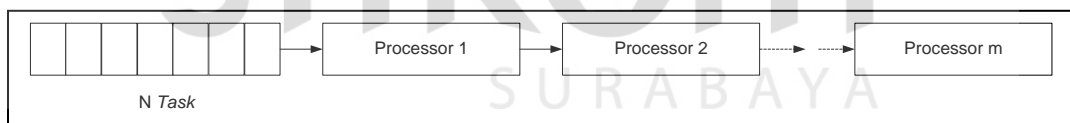
Jika “N” buah pekerjaan dapat dioperasikan bersamaan pada “m” buah. Pada kondisi seperti ini, masing-masing pekerjaan hanya dikerjakan oleh satu mesin.



Gambar 2.1 Penjadwalan *Parallel*

4. Penjadwalan Seri

Jika “N” buah pekerjaan harus melalui “m” buah “p” secara berurutan. Disini, setiap tugas atau pekerjaan harus melewati masing-masing mesin. Dalam hal ini usahakan untuk mendapatkan penjadwalan, serta memilih penjadwalan yang terbaik untuk melaksanakan sesuai dengan kriteria yang digunakan.



Gambar 2.2 Penjadwalan Seri

2.8 Aturan-aturan Prioritas

Menurut Tanuwijaya dan Setyawan (2012) aturan prioritas memberikan urutan pekerjaan yang harus dilaksanakan dalam proses produksi dengan satu mesin. Aturan prioritas digunakan untuk mengurangi waktu penyelesaian, jumlah pekerjaan

dalam sistem, utilisasi, dan keterlambatan kerja melalui penggunaan mesin yang optimal.

1. *First Come First Serve* (FCFS)
2. *Earliest Due Date* (EDD)
3. *Shortest Processing Time* (SPT)
4. *Longest Processing Time* (LPT)
5. *Critical Ratio* (CR)

2.8.1 *Earliest Due Date* (EDD)

Menurut Gaspersz (2012) metode EDD ini merupakan pengurutan pekerjaan berdasarkan batas waktu (*due date*) tercepat. Pekerjaan dengan jatuh tempo paling awal harus dijadwalkan terlebih dahulu daripada pekerjaan jatuh tempo belakangan. Aturan ini bertujuan untuk meminimasi keterlambatan maksimum (*maximum latenes*) atau meminimasi ukuran keterlambatan maksimum (*maximum tardiness*) suatu pekerjaan. Buruknya aturan ini menyebabkan jumlah pekerjaan yang terlambat menjadi banyak, serta akan menambah keterlambatan rata-rata (*mean tardiness*).

Contoh kasus penjadwalan produksi menurut Tanuwijaya dan Setyawan (2012):

Tabel 2.1 Contoh Kasus Penjadwalan

Pekerjaan	Waktu Pemrosesan (hari)	Batas Waktu Pekerjaan (Hari)
A	6	8
B	2	6

Pekerjaan	Waktu Pemrosesan (hari)	Batas Waktu Pekerjaan (Hari)
C	8	18
D	3	15
E	9	23

Contoh kasus penjadwalan produksi (Tanuwijaya dan Setyawan (2012)). Dari contoh kasus yang ada pada penyelesaian dengan metode EDD menghasilkan urutan B-A-D-C-E sehingga bisa diketahui sebagai berikut:

Tabel 2.2 Penyelesaian Kasus Dengan Metode EDD pada Penjadwalan

Pekerjaan	Waktu Pemrosesan (Hari)	Aliran Waktu	Batas Waktu Pekerjaan (Hari)	Keterlambatan
B	2		6	0
A	6	8	8	0
D	3	11	15	0
C	8	19	18	1
E	9	28	23	5
Jumlah	28	68		6

Dengan menggunakan aturan EDD, menghasilkan ukuran efektifitas sebagai berikut:

1. Waktu penyelesaian rata-rata = Jumlah aliran waktu total/Jumlah pekerjaan.

$$\text{Waktu penyelesaian rata-rata} = 68 \text{ hari}/5.$$

$$\text{Jadi, waktu penyelesaian rata-rata} = 13,6 \text{ hari.}$$

2. Utilisasi = Jumlah waktu proses total/Jumlah aliran waktu total.

Utilisasi = $28/68$.

Jadi, utilisasi = 41,20 %.

3. Jumlah pekerjaan rata-rata dalam sistem = Jumlah aliran waktu total/Waktu proses pekerjaan total.

Jumlah pekerjaan rata-rata dalam sistem = $68 \text{ hari}/28 \text{ hari}$.

Jadi, jumlah pekerjaan rata-rata dalam sistem = 2,43 pekerjaan.

4. Keterlambatan pekerjaan rata-rata = Jumlah hari terlambat/Jumlah pekerjaan

Keterlambatan pekerjaan rata-rata = $6/5$

Jadi, keterlambatan pekerjaan rata-rata = 1,2 hari.

2.8.2 *Shortest Processing Time (SPT)*

Menurut Tanuwijaya dan Setyawan (2012) *Shortest Processing Time (SPT)* merupakan metode yang memprioritaskan penyelesaian proses produksi berdasarkan waktu proses terpendek. Aturan ini didasarkan atas pemikiran bahwa apabila suatu pekerjaan dapat diselesaikan dengan cepat, maka mesin lain di bagian berikut akan menerima pekerjaan lebih cepat sehingga pekerjaan mengalir dengan cepat dan pemanfaatan yang tinggi. Tujuan metode ini adalah mencapai utilisasi yang maksimum dari mesin tersebut. Tetapi kelemahan metode ini adalah menunda-nunda suatu pekerjaan yang mempunyai waktu panjang sehingga jika tanggal jatuh tempo pekerjaan tersebut sangat dekat maka pekerjaan tersebut akan selesai jauh pada tanggal jatuh tempo yang diinginkan.

Contoh kasus penjadwalan produksi (Tanuwijaya dan Setyawan (2012)). Dari contoh kasus penyelesaian dengan metode SPT menghasilkan urutan B-D-A-C-E sehingga bisa diketahui sebagai berikut:

Tabel 2.3 Penyelesaian Kasus Dengan Metode SPT Pada Penjadwalan

Pekerjaan	Waktu Pemrosesan (Hari)	Aliran Waktu	Batas Waktu Pekerjaan (Hari)	Keterlambatan
B	2	2	6	0
D	3	5	15	0
A	6	11	8	3
C	8	19	18	1
E	9	28	23	5
Jumlah	28	65		9

- Waktu penyelesaian rata-rata = Jumlah aliran waktu total/Jumlah pekerjaan.
Waktu penyelesaian rata-rata = $65/5$.
Jadi, waktu penyelesaian rata-rata = 13 hari.
- Utilisasi = Jumlah waktu proses total/Jumlah aliran waktu total.
Utilisasi = $28/65$.
Jadi, utilisasi = 43,10%.
- Jumlah pekerjaan rata-rata dalam sistem = Jumlah aliran waktu total/waktu proses pekerjaan total.
Jumlah pekerjaan rata-rata dalam sistem = $65 \text{ hari} / 28 \text{ hari}$.

Jadi, jumlah pekerjaan rata-rata dalam sistem = 2,32 pekerjaan.

4. Ketelambatan pekerjaan rata-rata = Jumlah hari terlambat /Jumlah pekerjaan.

Keterlambatan pekerjaan rata-rata = $9/5$.

Jadi, keterlambatan pekerjaan rata-rata = 1,8 hari.

2.9 Evaluasi Hasil Aturan Penjadwalan

Dari kedua metode yang digunakan pada aturan prioritas di atas, dapat diringkas sebagai berikut:

Tabel 2.4 Hasil Perhitungan Kriteria Setiap Metode

Aturan	Waktu Penyelesaian Rata-rata (Hari)	Utilisasi (%)	Jumlah Pekerjaan Rata-rata Sistem	Keterlambatan Rata-rata (Hari)
SPT	13,00	43,10	2,23	1,80
EDD	13,60	41,20	2,43	1,20

2.10 Sistem Antrian

Menurut Herjanto (2007) sistem antrian sebagai suatu keadaan dimana terdapat *input* (dapat berupa manusia, benda, dan pekerjaan) yang akan dilayani/diproses, masuk ke dalam daerah tunggu dan mengantri untuk selanjutnya mendapatkan pelayanan, dan akhirnya keluar dari sistem.

2.11 System Development Life Cycle (SDLC)

Menurut Cahyono (2013) *System Development Life Cycle* (SDLC) ini didapat dari rekayasa lain yang saling terkait, model ini menawarkan pembuatan perangkat

lunak secara lebih nyata sesuai dengan tahapan, yaitu: *Requirements* (analisis sistem), *Analysis* (analisis kebutuhan sistem), *Design* (perancangan), *Coding* (implementasi), *Testing* (uji coba sistem), dan *Maintenance* (pemeliharaan).

1. *Requirements* (Analisis Kebutuhan)

Requirement dihasilkan dari konsultasi yang dilakukan terlebih dahulu dengan pengguna sistem. Kemudian keseluruhan data yang ada akan dirangkum sehingga dapat dimengerti oleh seluruh pihak terkait.

2. *Design* (Perancangan)

Setelah proses analisis selesai maka akan dibuat sebuah *design* sistem yang membagi kebutuhan menjadi sistem perangkat keras dan perangkat lunak serta arsitektur dalam bentuk *design database*, DFD, ERD, antarmuka pengguna atau *Graphical User Interface* (GUI) dibutuhkan untuk sistem.

3. *Coding* (Implementasi)

Rancangan yang telah dibuat dalam tahap sebelumnya akan diterjemahkan ke dalam suatu bentuk atau bahasa yang dapat dibaca dan diterjemahkan oleh komputer untuk diolah. Tahap ini juga dapat disebut dengan tahap implementasi, yaitu tahap yang mengkonversi hasil perancangan sebelumnya ke dalam sebuah bahasa pemrograman yang dimengerti oleh komputer. Kemudian komputer akan menjalankan fungsi-fungsi yang telah didefinisikan sehingga mampu memberikan layanan-layanan kepada penggunanya.

4. *Testing* (Uji coba sistem)

Rancangan aplikasi yang sudah lengkap selanjutnya dilakukan pengujian untuk meyakinkan bahwa persyaratan perangkat lunak sudah dipenuhi. Setelah uji coba

baru kemudian sistem disampaikan *user*. *Testing software* dilakukan untuk mendapatkan informasi *reliable* terhadap *software* dengan cara termudah dan paling efektif, antara lain:

- a. Apakah *software* telah siap digunakan?
- b. Apa saja resikonya?
- c. Apa saja kemampuannya?
- d. Apa saja keterbatasannya?
- e. Apa saja masalahnya?
- f. Apakah telah berlaku seperti yang diharapkan?

5. ***Black box Testing***

Black box testing dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*. Kategori *error* yang akan diketahui melalui *black box testing* adalah sebagai berikut:

- a. Fungsi yang hilang atau tidak benar.
- b. *Error* dari antarmuka.
- c. *Error* dari struktur data atau akses *eksternal database*.
- d. *Error* dari kinerja atau tingkah laku.
- e. *Error* dari inisialisasi dan terminasi.

Test didesain untuk menjawab pertanyaan sebagai berikut:

- a. Bagaimana validasi fungsi yang digunakan untuk *test case*?

- b. Bagaimana tingkah laku kinerja dari sistem yang digunakan untuk *test case*?
- c. Kategori masukan apa saja yang bagus digunakan untuk *test case*?
- d. Apakah sebagian sistem sensitif terhadap suatu nilai masukan tertentu?
- e. Bagaimana batasan suatu kategori masukan ditetapkan?
- f. Sistem mempunyai toleransi jenjang dan volume data apa saja?
- g. Apa saja akibat dari kombinasi data tertentu yang akan terjadi pada operasi dari sistem?

