

## BAB II TINJAUAN PUSTAKA

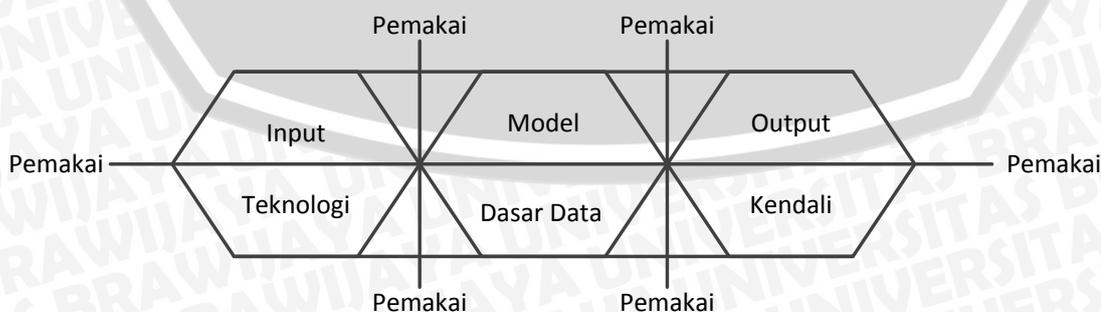
Pada tinjauan pustaka ini akan diuraikan berbagai teori atau referensi yang terkait dan menunjang permasalahan yang akan diteliti. Teori tersebut berkaitan dengan sistem informasi, konsep perancangan dan pengembangan sistem, konsep manajemen pemeliharaan. Bab ini bertujuan untuk mendukung permasalahan yang akan diteliti serta mendukung pembahasan hasil penelitian.

### 2.1. Sistem Informasi

Sistem informasi merupakan kombinasi teratur apa pun dari orang-orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi (O'Brien, 2005). Orang-orang bergantung pada sistem informasi untuk berkomunikasi antara satu sama lain dengan menggunakan berbagai jenis alat fisik (*hardware*), perintah dan prosedur pemrosesan informasi (*software*), saluran komunikasi (jaringan) dan data yang disimpan (sumber daya data) sejak permulaan peradaban.

#### 2.1.1. Komponen Sistem Informasi

Menurut Jogiyanto (2005), sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*) dan blok kendali (*control block*). Seperti yang ditunjukkan pada Gambar 2.1, sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sarannya.



Gambar 2.1 Blok sistem informasi yang berinteraksi.  
Sumber: Jogiyanto (2005).



1. Blok masukan (*input*)

Blok masukan ini mewakili data yang masuk ke dalam sistem informasi. Masukan (*input*) di sini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model

Blok model ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi

Teknologi merupakan *tool box* dalam sistem informasi. Teknologi ini digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirim keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

5. Blok basis data

Basis data (*database*) merupakan data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut dan juga perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas.

6. Blok kendali (*control*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

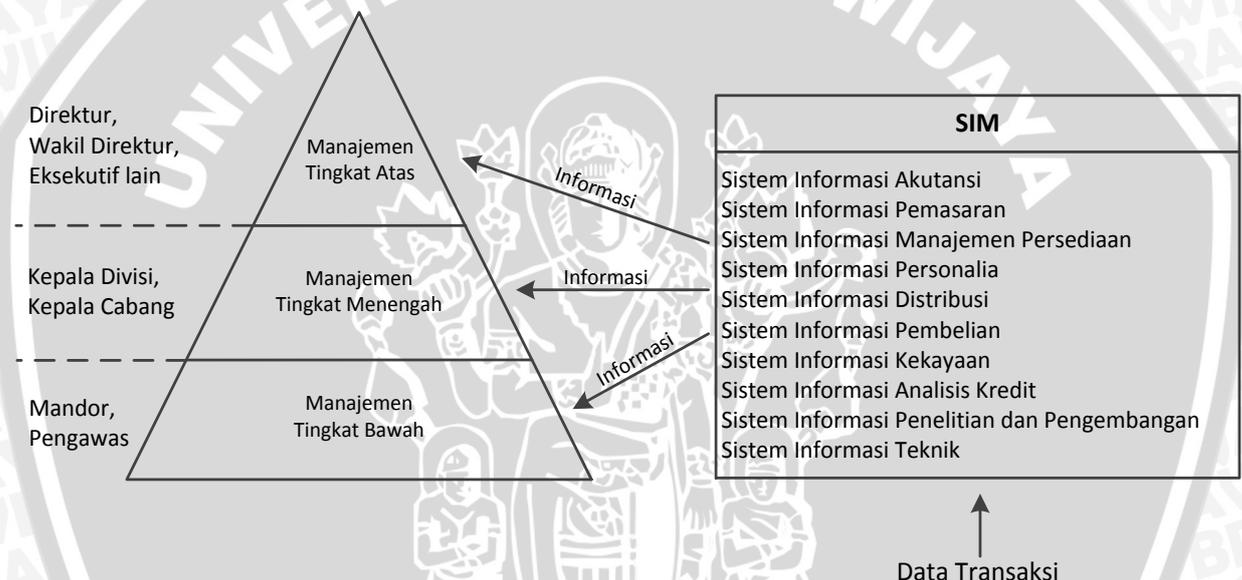
### 2.1.2. Sistem Informasi Manajemen

Menurut Laudon (2004), sistem informasi manajemen (SIM) suatu studi mengenai sistem informasi yang fokus pada penggunaan sistem informasi dalam bisnis dan manajemen. SIM mengkombinasi teori-teori pengetahuan komputer, pengetahuan manajemen dan operasi riset dengan suatu orientasi praktis ke arah pengembangan

solusi sistem atas permasalahan nyata dan mengelola sumber-sumber teknologi informasi.

SIM juga disebut kumpulan dari manusia dan sumber-sumber daya modal di dalam suatu organisasi yang bertanggung jawab mengumpulkan dan mengolah data untuk menghasilkan informasi yang berguna untuk semua tingkatan manajemen di dalam kegiatan perencanaan dan pengendalian (Jogiyanto, 2005).

SIM melakukan fungsi-fungsi untuk dapat menyediakan semua informasi yang mempengaruhi semua operasi dalam suatu organisasi, sehingga SIM terdiri dari berbagai sistem informasi di dalamnya. Sistem-sistem informasi tersebut dimaksudkan untuk memberikan informasi kepada semua tingkatan manajemen, yaitu manajemen tingkat bawah, manajemen tingkat menengah dan manajemen tingkat atas.



Gambar 2.2 Informasi dari SIM untuk semua tingkatan manajemen.  
Sumber: Jogiyanto (2005).

## 2.2. Sistem Basis Data

Basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil *query* basis data disebut sistem manajemen basis data (*database management system*, DBMS).

Menurut Indrajadi (2011), basis data merupakan kumpulan terpadu dari elemen data logis yang saling berhubungan yang mengonsolidasi banyak catatan yang

sebelumnya disimpan dalam *file* terpisah dan dirancang untuk memenuhi kebutuhan informasi yang dibutuhkan oleh suatu organisasi.

Tujuan utama DBMS adalah untuk menyediakan tinjauan abstrak dari data bagi *user*. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil dengan efisien. Pertimbangan efisien yang digunakan adalah bagaimana merancang struktur data yang kompleks, tetapi tetap dapat digunakan oleh pengguna yang masih awam, tanpa mengetahui kompleksitas struktur data. Basis data juga menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data yang terpusat, seperti adanya duplikasi data, hubungan antar data tidak jelas, organisasi data dan *update* data menjadi rumit. Jadi tujuan dari pengaturan data dengan menggunakan basis data adalah:

1. Menyediakan penyimpanan data untuk dapat digunakan oleh organisasi saat sekarang dan masa yang akan datang.
2. Kemudahan pemasukan data, sehingga meringankan tugas operator dan menyangkut pula waktu yang diperlukan oleh pemakai untuk mendapatkan data serta hak-hak yang dimiliki terhadap data yang ditangani.
3. Pengendalian data untuk setiap siklus agar data selalu *up-to-date* dan dapat mencerminkan perubahan spesifik yang terjadi di setiap sistem.
4. Pengamanan data terhadap kemungkinan penambahan, pengubahan, pengrusakan dan gangguan-gangguan lain.

### 2.3. Pengembangan Sistem

Pengembangan sistem (*system development*) dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada (Jogiyanto, 2005). Sistem yang lama perlu diperbaiki atau perlu diganti disebabkan oleh beberapa hal, yaitu:

1. Adanya permasalahan-permasalahan yang timbul di sistem lama yang berupa:
  - a. Ketidakterbacaan dalam sistem lama yang menyebabkan sistem lama tidak beroperasi sesuai dengan yang diharapkan.
  - b. Pertumbuhan organisasi yang menyebabkan harus disusunnya sistem yang baru. Pertumbuhan organisasi ini dapat berupa kebutuhan informasi yang semakin luas, *volume* pengolahan data semakin meningkat dan lain-lain.

2. Untuk meraih kesempatan-kesempatan (*opportunities*).

Dalam keadaan pasar yang bersaing, kecepatan informasi atau efisiensi waktu sangat menentukan berhasil atau tidaknya strategi dan rencana-rencana yang telah disusun untuk meraih kesempatan-kesempatan yang ada.

3. Adanya instruksi-instruksi (*directives*).

Penyusunan sistem baru dapat juga karena adanya instruksi dari atas pimpinan ataupun dari luar organisasi, seperti misalnya peraturan pemerintah.

Jogiyanto (2005) juga mengemukakan bahwa dengan telah dikembangkannya sistem yang baru, maka diharapkan akan terjadi peningkatan-peningkatan di sistem yang baru. Peningkatan-peningkatan ini berhubungan dengan PIECES, yaitu sebagai berikut:

1. *Performance* (kinerja), peningkatan terhadap kinerja sistem yang baru sehingga menjadi lebih efektif. Kinerja dapat diukur dari *throughput* dan *response time*. *Throughput* adalah jumlah pekerjaan yang dapat dilakukan suatu saat tertentu. Sedangkan *response time* adalah rata-rata waktu yang tertunda diantara dua transaksi.
2. *Information* (informasi), peningkatan terhadap kualitas informasi yang disajikan.
3. *Economy* (ekonomis), peningkatan terhadap manfaat-manfaat atau keuntungan-keuntungan atau penurunan-penurunan biaya yang terjadi.
4. *Control* (pengendalian), peningkatan terhadap pengendalian untuk mendeteksi dan memperbaiki kesalahan-kesalahan serta kecurangan-kecurangan yang ada dan akan terjadi.
5. *Efficiency* (efisiensi), peningkatan terhadap efisiensi operasi. Efisiensi ini berhubungan dengan bagaimana sumber daya tersebut digunakan dengan pemborosan yang paling minimum.
6. *Services* (pelayanan), peningkatan terhadap pelayanan yang diberikan oleh sistem.

### 2.3.1. Prinsip Pengembangan Sistem

Dalam melakukan proses pengembangan sistem, beberapa prinsip harus tidak boleh dilupakan. Prinsip-prinsip ini adalah sebagai berikut:

1. Sistem yang dikembangkan adalah untuk manajemen.
2. Sistem yang dikembangkan adalah investasi modal yang besar yang juga mempertimbangkan 2 hal berikut ini:
  - a. Semua alternatif yang ada harus diinvestigasi.
  - b. Investasi yang terbaik harus bernilai.

3. Sistem yang dikembangkan memerlukan orang yang terdidik.
4. Tahapan kerja dan tugas-tugas yang harus dilakukan dalam proses pengembangan sistem.
5. Proses pengembangan sistem tidak harus urut.
6. Jangan takut membatalkan proyek.
7. Dokumentasi harus ada untuk pedoman dalam pengembangan sistem.

### 2.3.2. *System Development Life Cycle (SDLC)*

*Systems Development Life Cycle* atau siklus hidup pengembangan sistem adalah proses yang digunakan oleh analis sistem untuk mengembangkan sistem informasi, mulai dari perencanaan, penentuan kebutuhan, perancangan, validasi, sampai pelatihan dan penyerahan kepada konsumen.

Para siklus hidup pengembangan sistem (SDLC) adalah model konseptual yang digunakan dalam manajemen proyek yang menggambarkan tahap-tahap yang terlibat dalam suatu proyek pengembangan sistem informasi, dari studi kelayakan awal melalui pemeliharaan aplikasi selesai (Furqon, 2010).

Sedangkan Kahfie (2012) mengemukakan bahwa SDLC merupakan sebuah siklus hidup pengembangan perangkat lunak yang terdiri dari beberapa tahapan-tahapan yang sangat penting dalam keberadaan perangkat lunak yang dilihat dari segi pengembangannya yang dilakukan oleh analis sistem dan *programmer*. SDLC dalam rekayasa sistem dan rekayasa perangkat lunak juga adalah suatu proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut.

Secara umum, Khafie (2012) menyatakan bahwa SDLC ini memiliki beberapa tahapan yang biasa digunakan dalam pengembangan sistem. Tahapan-tahapan tersebut adalah sebagai berikut:

#### 1. Perencanaan sistem

Tahap perencanaan sistem ini dimulai dengan mempelajari konsep sistem dan permasalahan yang hendak diselesaikan, apakah sistem baru tersebut realistis dalam masalah pembiayaan, waktu, serta perbedaan dengan sistem yang ada sekarang.

#### 2. Analisis sistem

Tahap analisis sistem ini dengan menganalisis konsep sistem, permasalahan dan keperluan yang hendak dibuat.

### 3. Desain sistem

Tahap desain sistem ini dengan mendesain sistem teknologi baru untuk permasalahan yang sama.

### 4. Implementasi sistem

Dalam tahap implementasi sistem ini *software* telah diuji dan siap diimplementasikan kedalam sistem pengguna atau sudah siap diterapkan.

### 5. Pemeliharaan sistem

Dalam tahap pemeliharaan sistem ini terdapat beberapa kegiatan pemeliharaan yaitu:

- a. *Corrective*, memperbaiki desain dan *error* pada program.
- b. *Adaptive*, memodifikasi sistem untuk beradaptasi dengan perubahan lingkungan.
- c. *Perfective*, melibatkan sistem untuk menyelesaikan masalah baru atau mengambil kesempatan (penambahan fitur).
- d. *Preventive*, menjaga sistem dari kemungkinan masalah di masa yang akan datang.

## 2.4. Data Flow Diagram (DFD)

Menurut Jogiyanto (2005), *Data Flow Diagram* biasa dikenal juga dengan sebutan diagram arus data. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured analysis and design*). DFD juga merupakan alat yang cukup populer karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas sertam merupakan dokumentasi dari sistem yang baik.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya telepon, surat dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya *file* kartu, *microfiche*, *harddisc*, *tape* dan sebagainya).

DFD digambarkan dengan menggunakan empat simbol dasar yang merepresentasikan proses (*process*), arus data (*data flow*), penyimpanan data (*data store*), dan entitas eksternal (*external entity*) atau batas sistem (*boundary*).

Tabel 2.1 Simbol-simbol DFD.

NO.	Nama Simbol	Simbol
1.	Proses ( <i>process</i> )	 Atau 
2.	Arus data ( <i>data flow</i> )	
3.	Penyimpanan data ( <i>data store</i> )	 Atau 
4.	Entitas eksternal ( <i>external entity</i> ) atau batas sistem ( <i>boundary</i> )	

Sumber: Jogiyanto (2005)

1. Entitas eksternal (*external entity*) atau batas sistem (*boundary*)

Entitas eksternal (*external entity*) atau batas sistem (*boundary*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. Penyimpanan data (*data store*)

Penyimpanan data (*data store*) digunakan dalam DFD untuk merepresentasikan situasi kapan dan dimana sistem mengambil data, karena satu atau lebih proses akan menggunakan penyimpanan data ini di lain waktu.

3. Arus data (*data flow*)

Arus data (*data flow*) ini mengalir di antara proses, penyimpanan data, dan entitas eksternal. Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

#### 4. Proses (*process*)

Proses (*process*) adalah kegiatan atau kerja yang dilakukan orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.

### 2.5. Entity Relationship Diagram (ERD)

Dalam mengembangkan sebuah sistem, sistem analis harus mampu mengidentifikasi data dan proses serta menguraikannya dengan model. Suatu *Entity Relationship Diagram* (ERD) digunakan untuk mengidentifikasi tipe entitas di dalam suatu sistem yang diuraikan dengan data dan relasi di antara entitas.

Berbeda dengan DFD ataupun *flowchart* sistem, ERD tidak menggambarkan aliran informasi atau data dalam sistem. Satu dari entitas-entitas yang ada harus diposisikan di atas, di kiri atau di kanan maupun di bawah bagian dari entitas lain, tetapi posisi ini tidak menggambarkan hubungan berupa aliran informasi dari entitas yang satu ke entitas yang lain.

Relasi sendiri adalah suatu asosiasi yang ada antara dua atau lebih entitas dan digambarkan dengan suatu garis yang diberi label (berupa kata kerja). Sementara itu, dalam ERD juga dikenal dengan istilah kardinalitas yaitu batasan (*constraint*) untuk relasi yang menyatakan berapa banyak entitas mempunyai relasi satu dengan yang lainnya.

Tabel 2.2 Kardinalitas Minimum dan Maksimum pada Relasi Dua Entitas dalam ERD

No.	Notasi	Derajat Relasi Maks-Min	Nama Relasi	Keterangan
1.	—○+	( 0 , 1 )	<i>Zero to One</i>	Relasi satu dengan salah satu atribut boleh tidak berelasi (relasi kosong)
2.	—○∞	( 0 , N )	<i>Zero to Many</i>	Relasi banyak dengan salah satu atribut dapat tidak berelasi
3.	—#	( 1 , 1 )	<i>One to One</i>	Relasi satu dengan salah satu atribut harus berelasi
4.	—+∞	( 1 , N )	<i>One to Many</i>	Relasi banyak dengan salah satu atribut harus mempunyai relasi

Sumber: Lio (2010).

## 2.6. Visual Basic .NET

Visual Studio .NET merupakan suatu perangkat lunak yang dapat memudahkan *programmer* dalam membuat program-program yang berteknologi .NET. Melalui Visual Studio .NET, *programmer* dapat mengetikkan program pada editor yang tersedia, menggunakan fasilitas untuk mencari kesalahan program, menciptakan *user interface* secara interaktif, membentuk program menjadi file .exe hingga menjalankan program yang telah dibuat. Semua fasilitas yang disediakan Visual Studio tersebut terangkum menjadi satu kesatuan aplikasi yang disebut dengan istilah *Visual Studio Integrated Development Environment (VS-IDE)* yang secara sederhana dapat diartikan sebagai lingkungan atau tempat pembuatan program yang terintegrasi pada Visual Studio.

Sedangkan untuk Visual Basic .NET sendiri, menurut Sibero (2010), adalah bahasa pemrograman yang dikembangkan oleh perusahaan Microsoft. Visual Basic .NET merupakan pengembangan dari versi sebelumnya, yaitu Visual Basic 6.0, yang memiliki karakteristik mudah untuk dipahami namun andal dalam mengikuti tren teknologi perangkat lunak. Perbedaan mendasar antara Visual Basic .NET dengan versi-versi sebelumnya adalah kemampuan OOP (*Object Oriented Programming*) yang telah ditanamkan pada Visual Basic .NET. Saat ini Visual Basic .NET telah dikolaborasikan dengan beberapa jenis aplikasi, seperti aplikasi *desktop* dan aplikasi berbasis *web*.

## 2.7. Pemeliharaan (*Maintenance*)

Pemeliharaan merupakan fungsi yang penting dalam suatu pabrik. Sebagai suatu usaha menggunakan fasilitas/peralatan produksi agar kontinuitas produksi dapat terjamin dan menciptakan suatu keadaan operasi yang memuaskan sesuai dengan rencana. Selain itu, fasilitas/peralatan produksi tersebut tidak mengalami kerusakan selama dipergunakan sebelum jangka waktu tertentu yang direncanakan tercapai.

Pemeliharaan (*maintenance*) juga didefinisikan sebagai suatu kombinasi dari berbagai tindakan yang dilakukan untuk menjaga suatu barang dalam, atau memperbaikinya sampai suatu kondisi yang bisa diterima (Corder, 1992).

### 2.7.1. Tujuan Pemeliharaan

Menurut Corder (1992), tujuan pemeliharaan yang utama dapat didefinisikan dengan jelas sebagai berikut:

1. Memperpanjang usia kegunaan aset yaitu setiap bagian dari suatu tempat kerja, bangunan dan isinya.
2. Menjamin ketersediaan optimum peralatan yang dipasang untuk produksi atau jasa dan mendapatkan laba investasi (*return of investment*) maksimum yang mungkin.
3. Menjamin kesiapan operasional dari seluruh peralatan yang diperlukan dalam kegiatan darurat setiap waktu, misal unit cadangan, unit pemadam kebakaran dan penyelamat, dan sebagainya.
4. Menjamin keselamatan orang yang menggunakan sarana tersebut.

### 2.7.2. Jenis-jenis Pemeliharaan

Menurut Corder (1992) secara umum, ditinjau dari saat pelaksanaan pekerjaan pemeliharaan dikategorikan dalam dua cara, yaitu:

#### 1. Pemeliharaan terencana (*planned maintenance*)

Pemeliharaan terencana adalah pemeliharaan yang dilakukan secara terorganisir untuk mengantisipasi kerusakan peralatan di waktu yang akan datang, pengendalian dan pencatatan sesuai dengan rencana yang telah ditentukan sebelumnya. Pemeliharaan terencana dibagi menjadi dua aktivitas utama yaitu:

##### a. Pemeliharaan pencegahan (*Preventive Maintenance*)

Pemeliharaan pencegahan adalah kegiatan pemeliharaan dan perawatan yang dilakukan untuk mencegah timbulnya kerusakan-kerusakan yang tidak terduga dan menentukan kondisi atau keadaan yang menyebabkan fasilitas produksi mengalami kerusakan pada waktu digunakan dalam proses produksi (Assauri, 2004).

##### b. Pemeliharaan korektif (*Corrective Maintenance*)

Menurut Prawirosentono (2000), pemeliharaan korektif adalah pemeliharaan yang dilakukan karena adanya hasil produk yang tidak sesuai dengan rencana. Kegiatan ini dimaksudkan agar fasilitas/peralatan tersebut dapat digunakan kembali dalam operasi sehingga proses produksi dapat berjalan lancar kembali.

#### 2. Pemeliharaan tak terencana (*unplanned maintenance*)

Pemeliharaan tak terencana yaitu pemeliharaan darurat, yang didefinisikan sebagai pemeliharaan dimana perlu segera dilaksanakan tindakan untuk mencegah

akibat yang serius, misalnya hilangnya produksi, kerusakan besar pada peralatan, atau untuk keselamatan kerja.

Sedangkan menurut Asyari (2007), membagi pemeliharaan menjadi:

1. Pemeliharaan pencegahan (*Preventive Maintenance*) : Pemeliharaan pencegahan adalah pemeliharaan yang bertujuan untuk mencegah terjadinya kerusakan, atau cara pemeliharaan yang direncanakan untuk pencegahan. Ruang lingkup pekerjaan preventif termasuk inspeksi, perbaikan kecil, pelumasan dan penyetelan, sehingga peralatan atau mesin-mesin selama beroperasi terhindar dari kerusakan.
2. Pemeliharaan korektif (*Corrective Maintenance*): Pemeliharaan korektif adalah pekerjaan pemeliharaan yang dilakukan untuk memperbaiki dan meningkatkan kondisi fasilitas atau peralatan sehingga mencapai standar yang dapat di terima. Dalam perbaikan dapat dilakukan peningkatan- peningkatan sedemikian rupa, seperti melakukan perubahan atau modifikasi rancangan agar peralatan menjadi lebih baik.
3. Pemeliharaan berjalan (*Running Maintenance*): Pemeliharaan ini dilakukan ketika fasilitas atau peralatan dalam keadaan bekerja. Pemeliharaan berjalan diterapkan pada peralatan-peralatan yang harus beroperasi terus dalam melayani proses produksi,
4. Pemeliharaan prediktif (*Predictive Maintenance*): Pemeliharaan prediktif ini dilakukan untuk mengetahui terjadinya perubahan atau kelainan dalam kondisi fisik maupun fungsi dari sistem peralatan. Biasanya pemeliharaan prediktif dilakukan dengan bantuan panca indra atau alat-alat monitor yang canggih,
5. Pemeliharaan setelah terjadi kerusakan (*Breakdown Maintenance*): Pekerjaan pemeliharaan ini dilakukan ketika terjadinya kerusakan pada peralatan, dan untuk memperbaikinya harus disiapkan suku cadang, alat-alat dan tenaga kerjanya,
6. Pemeliharaan Darurat (*Emergency Maintenance*): Pemeliharaan ini adalah pekerjaan pemeliharaan yang harus segera dilakukan karena terjadi kemacetan atau kerusakan yang tidak terduga.
7. Pemeliharaan berhenti (*Shutdown Maintenance*): Pemeliharaan berhenti adalah pemeliharaan yang hanya dilakukan selama mesin tersebut berhenti beroperasi.
8. Pemeliharaan rutin (*Routine Maintenance*): Pemeliharaan rutin adalah pemeliharaan yang dilaksanakan secara rutin atau terus-menerus.
9. *Design out maintenance* adalah merancang ulang peralatan untuk menghilangkan sumber penyebab kegagalan dan menghasilkan model kegagalan yang tidak lagi atau lebih sedikit membutuhkan *maintenance*.

### 2.7.3. Keandalan

Menurut Laksono (2007), keandalan merupakan kemampuan atau tingkat berfungsinya suatu peralatan. Suatu peralatan dikatakan dapat berfungsi jika keandalannya baik dan dikatakan tidak berfungsi jika keandalannya buruk.

Keandalan suatu mesin dapat diartikan sebagai tingkat kemampuan suatu alat untuk dapat berfungsi dengan baik. Kondisi keandalan dari suatu mesin ada dua, yaitu keandalan mesin bernilai 1 bila mesin dapat berfungsi dengan baik, sebaliknya keandalan mesin bernilai 0 bila mesin dalam kondisi rusak atau tidak dapat dijalankan. Dapat dikatakan bahwa nilai keandalan suatu mesin bersifat probabilitas ( $0 \leq \text{nilai keandalan} \leq 1$ ).

#### 2.7.3.1. Fungsi Keandalan

Menurut Laksono (2007), keandalan suatu peralatan dapat berkurang sejalan dengan bertambahnya waktu, sehingga dapat dikatakan bahwa keandalan merupakan fungsi dari waktu. Untuk mengukur keandalan diperlukan fungsi keandalan yang disebut juga *reliability* ( $R(t)$ ).

$$\text{Fungsi keandalan : } R(t) = 1 - \int_0^t f(t) dt \quad (2-1)$$

Keterangan :  $R(t) = \text{reliability}$

$f(t)$  = fungsi kepadatan probabilitas, kemungkinan gagal untuk periode tertentu.

#### 2.7.3.2. Mean Time To Failure (MTTF)

Menurut Laksono (2007), *mean life* adalah rata-rata waktu ekspektasi terjadinya kegagalan dari unit-unit identik yang beroperasi pada kondisi normal. *Mean life* juga disebut MTTF. MTTF sendiri merupakan waktu rata-rata kegagalan dari suatu komponen. MTTF hanya berlaku pada peralatan yang jika mengalami kerusakan maka komponennya harus diganti. MTTF diperlukan untuk mengetahui kualitas dan kemampuan dari komponen yang digunakan.

$$\text{Fungsi MTTF : } MTTF = E(t) = \int_0^{\infty} R(t) dt \quad (2-2)$$

Keterangan : MTTF = *mean time to failure* (rata-rata waktu kerusakan)

$R(t)$  = *reliability*

$t$  = waktu

### 2.7.3.3. Mean Time Between Failure (MTBF)

*Mean Time Between Failure* (MTBF) atau rata-rata waktu kerusakan adalah ekspektasi bisa pakai dari suatu sistem atau peralatan, seperti yang dinyatakan oleh Laksono (2007). MTBF bermanfaat untuk mengetahui kinerja dan kemampuan dari peralatan yang digunakan dan dapat didefinisikan sebagai berikut:

$$\text{Fungsi MTBF : MTBF} = \int_0^{\infty} R(t) dt = \int_0^{\infty} e^{-(t/\eta)^\beta} dt \quad (2-3)$$

$$= -\frac{\eta^\beta}{\beta \cdot t^{\beta-1}} e^{-(t/\eta)^\beta} \quad (2-4)$$

Keterangan : MTBF = *mean time between failure* (rata-rata waktu kerusakan)

R(t) = *reliability*

t = waktu

### 2.7.4. Maintainability

Menurut Laksono (2007), *maintainability* adalah probabilitas mesin yang mengalami kerusakan dapat dioperasikan kembali dalam suatu selang *down time* tertentu. Untuk mengoptimalkan *maintainability* sistem ada dua faktor yang perlu diperhatikan, yaitu model perawatan (*maintenance model*) dan perancangan untuk mendapatkan tingkat *maintainability* tertentu.

Jika  $f(t)$  adalah fungsi *density* probabilitas terhadap waktu yang dibutuhkan untuk mempengaruhi tindakan (*repair*, *overhaul*, atau *replacement*), maka *maintainability* dari suatu peralatan dapat didefinisikan sebagai berikut:

$$\text{Maintainability} = \int_0^t f(t) dt \quad (2-5)$$

Keterangan :  $f(t)$  = *density* probabilitas terhadap waktu yang dibutuhkan untuk penanganan.

Terdapat beberapa perhitungan-perhitungan dalam *maintainability* yang salah satunya menggunakan *Mean Time Between Maintenance* (MTBM) atau waktu rata-rata di antara perawatan. MTBM sendiri meliputi kebutuhan perawatan preventif (terjadwal) dan perawatan korektif (tidak terjadwal).

$$\text{MTBM} = \frac{\text{Total Waktu Operasi}}{\text{Frekuensi Pemeliharaan}} \quad (2-6)$$

$$\text{Fpt} = \frac{1 - (\lambda \times \text{MTBM})}{\text{MTBM}} \quad (2-7)$$

Keterangan :  $\lambda$  = laju kerusakan

Fpt = laju perawatan preventif