

## BAB 2

### LANDASAN TEORI

#### 2.1 *Maintenance*

##### 2.1.1 *Arti dan Peranan Maintenance*

*Maintenance* adalah adalah “semua tindakan yang diperlukan untuk mempertahankan barang atau peralatan untuk kembali pada kondisi tertentu” (Balbir S, 2006: 2).

Karena *maintenance* diperlukan maka *maintenance* penting bagi sebuah perusahaan untuk melakukan *monitoring* dan menjaga perlengkapan fasilitas kerja. Hal ini dilakukan dengan mendesain, mengatur dan mengecek fungsi pada sebuah objek saat waktu kerja (*uptime*) dan meminimalisasi pemberhentian kerja (*downtime*) yang disebabkan oleh kerusakan dengan itu peran *maintenance* dapat mengontrol kondisi mesin agar saat operasi selalu bekerja dengan baik, meningkatkan kapasitas dan meningkatkan kualitas produksi.

##### 2.1.2 *Corrective Maintenance*

Tindakan perawatan untuk mengembalikan fungsi dari sebuah peralatan produksi yang mengalami kerusakan baik ringan, sedang maupun parah, agar bisa melakukan fungsinya kembali untuk mendukung proses produksi dalam sebuah pabrik. *Corrective maintenance* biasanya memperbaiki masalah yang ada sehingga tindakan dilakukan saat suatu peralatan rusak (Keith Mobley, Higgins, & Wikoff, 2008: 2.4 ).

##### 2.1.3 *Preventive Maintenance*

*Preventive maintenance* adalah rencana pemeliharaan pada pabrik dan perlengkapan pabrik. Hal ini merupakan rancangan yang didesain untuk meningkatkan penggunaan pada mesin (Efendy, 2008: 572).

Untuk mencapai keberhasilan dengan *preventive maintenance* yang telah dikembangkan, harus diuji, dikoreksi dan disesuaikan dengan frekuensi sistem penjadwalan kegiatan. Penjadwalan harus dapat memaksimalkan kondisi objek yang dijadwalkan.

Efendy ( 2008: 572) *preventive maintenance* memiliki dua jenis kegiatan, yaitu:

- ***Routine Maintenance.*** Kegiatan *maintenance* secara rutin dilakukan setiap hari. Kegiatan yang dilakukan berupa pembersihan mesin dan

peralatan, pemberian minyak pelumas, pengecekan bahan bakar dan sebagainya.

- **Periodic Maintenance.** Kegiatan *maintenance* yang dilakukan secara berkala dalam jangka waktu tertentu, misalnya pengecekan setiap satu minggu sekali, satu bulan atau satu tahun sekali.

## 2.2 Parameter dan Fungsi Keandalan

### 2.2.1. Mean Time to Failure (MTTF) dan Mean Time To Repair (MTTR)

*Mean Time To Failure* (MTTF) merupakan nilai rata-rata waktu kegagalan yang akan datang dari sebuah sistem (komponen). Untuk sistem yang dapat direparasi, maka MTTF adalah masa kerja suatu komponen saat pertama kali digunakan atau dihidupkan sampai unit tersebut akan rusak kembali atau perlu di periksa kembali.

*Mean Time To Repair* (MTTR) adalah waktu rata-rata untuk waktu pengecekan atau perbaikan saat komponen atau unit tersebut diperiksa sampai komponen atau unit tersebut digunakan atau dihidupkan kembali. Karena perhitungan MTTF dan MTTR sama maka didefnisikan dengan:

- Distribusi Normal

$$\text{MTTF} / \text{MTTR} = \mu$$

- Distribusi Lognormal

$$\text{MTTF} / \text{MTTR} = \exp \mu$$

- Distribusi Weibull

$$\text{MTTF} / \text{MTTR} = \beta T \left[ \frac{1}{x} + 1 \right]$$

- Distribusi Exponensial

$$\text{MTTF} / \text{MTTR} = \frac{1}{\lambda}$$

Sumber: (Anggono, Julianingsih, & Linawati, 2005)

### 2.2.2. Fungsi Keandalan masing-masing Distribusi

Fungsi keandalan didefinisikan sebagai probabilitas suatu alat akan beroperasi dengan baik tanpa mengalami kerusakan pada suatu periode waktu  $t$  dalam kondisi operasi standar. keandalan untuk masing-masing distribusi berbeda antara satu dengan yang lainnya. Di bawah ini akan diberikan rumus-rumus fungsi keandalan.

- Distribusi Normal

$$\text{Fungsi Keandalan } R(t) = 1 - \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^t \exp \left[ \frac{-(t-\mu)^2}{2\sigma^2} \right]$$

- Distribusi Lognormal

$$\text{Fungsi Keandalan } R(t) = 1 - \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^t \exp\left[-\frac{(ln t - \mu)^2}{2\sigma^2}\right]$$

- Distribusi Weibull

$$\text{Fungsi Keandalan } R(t) = e^{-\left(\frac{t}{\beta}\right)^\alpha}$$

- Distribusi Exponensial

$$\text{Fungsi Keandalan } R(t) = \exp(-\lambda.t)$$

Sumber: (Anggono, Julianingsih, & Linawati, 2005)

### 2.2.3. Homogenitas Data

Homogenitas dimaksudkan untuk memperlihatkan bahwa dua atau lebih kelompok data sampel berasal dari populasi yang memiliki variansi yang sama. Homogenitas dapat diuji dengan uji levene (Gastwirth, Gel, & Miao, 2009).

Uji levene merupakan tes yang paling umum digunakan, tes yang dilakukan adalah untuk melihat satu atau lebih variasi memiliki varian yang sama pada interval tertentu. Tes ini biasa muncul dalam menggunakan *software* SPSS.

Cara menafsirkan uji levene ini adalah, jika Signifikan > 0,05 maka dapat dikatakan bahwa data adalah homogen. Signifikan adalah nilai kebenaran dari hipotesis yang diterima atau ditolak. secara umum kita menggunakan nilai signifikansi tersebut didasarkan pada tingkat kepercayaan yang diinginkan oleh peneliti.

### 2.3 Modularity

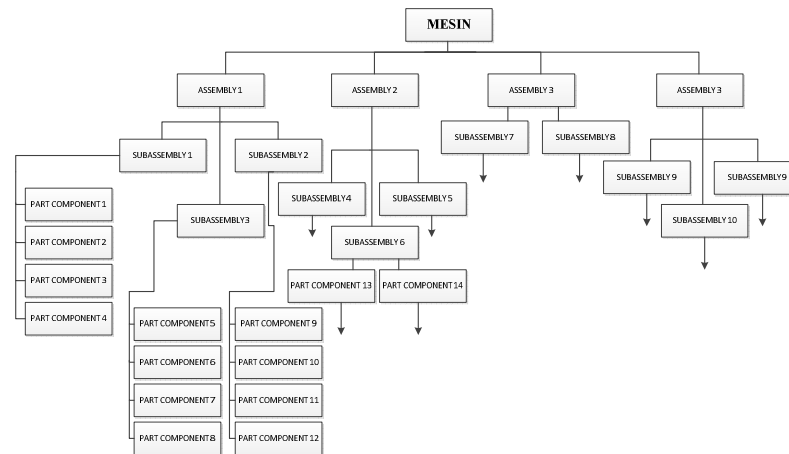
Balbir S (2006: 107) *Modularity* adalah sebuah pengelompokan komponen-komponen yang berbeda, tetapi memiliki kesamaan berdasarkan struktur fungsinya sehingga dapat memudahkan proses perbaikan dan penggantian komponen-komponen tersebut.

Beberapa keuntungan modularisasi adalah membagi tanggung jawab *maintenance*, mendesain waktu *downtime* serta pengerjaan yang lebih singkat dan menurunkan biaya perbaikan dalam sistem *maintenance*.

Prinsip dasar *modularity* adalah pemisahan terhadap komponen yang tidak berhubungan tetapi apabila terdapat hubungan keterkaitan antara dua atau banyak komponen maka akan lebih baik apabila komponen-komponen tersebut dapat digabungkan menjadi sebuah modul.

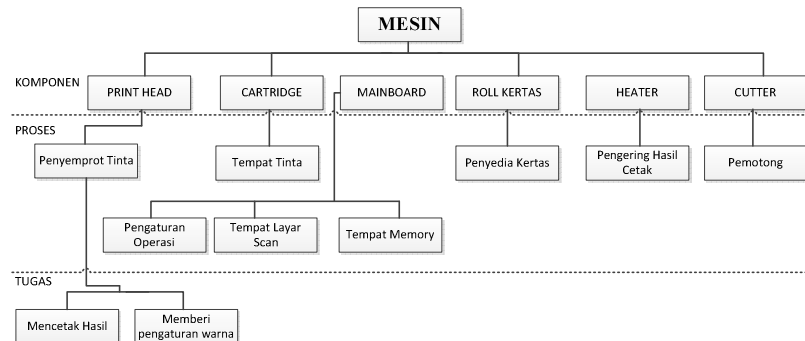
Modul adalah pembagian komponen yang dapat dideskripsikan berdasarkan fungsional. Untuk mengukur analisa dalam membuat modul dapat menggunakan *component tree*, *process graph* dan *matrik* (Gershenson, Prasad, & Allamneni, 1999).

*Component tree* adalah rincian hubungan fisik antar komponen-komponen di tingkat abstraksi. Untuk membangun *component tree*, produk dibagi dalam modul-modul dan komponen-komponen. Lebih lanjut modul-modul tersebut akan diklasifikasikan ke dalam subassemblies, kemudian menjadi individual komponen, dan terakhir atribut produk yang menjelaskan komponen tersebut (Anggono, Julianingsih, & Linawati, 2005).



Gambar 2.1 Contoh *Component Tree*

*Process graph* adalah gambaran setiap kegiatan dalam setiap komponen. Komponen-komponen dikelompokkan bersama berdasarkan kesamaan kegiatan yang dimiliki.



Gambar 2.2 Contoh *Process Graph*

*Matriks* untuk sebuah produk adalah penggabungan *component tree* dan *process graph*, modularity matriks tersebut dibangun dalam satu evaluasi untuk menyimpan data *similarity* dan satu untuk menyimpan *dependencies*.

Perhitungan modularity untuk *maintenance* menggunakan *modularity evaluation matriks* karena dengan *matriks* perhitungan nilai homogenitas sudah dapat dibandingkan dan dikelompokkan dalam satu modul. Perhitungan ini dilakukan dengan cara mengolah data kerusakan komponen untuk melihat nilai homogenitas antar komponen dengan *software* SPSS, kemudian nilai homogenitas tersebut dimasukkan dalam *matriks*.

Untuk nilai homogenitas antar komponen yang tinggi akan dimasukkan dalam satu modul dibandingkan nilai homogenitas antar komponen yang rendah. Untuk nilai homogenitas antar komponen ada yang tidak homogen maka tidak dikelompokkan dalam satu modul.

Modularity Evaluation Matrik				Component						Process							
				SubAssembly1				Assy2		Process1			Pro2				
				Component1		Component2		Component3		Task1		Task2	Task3				
Attribut1		Attribut2		Attribut3		Attribut4		Attribut5		Attribut6		Subtask1	Subtask2	Subtask3	Subtask4	Subtask5	
Component	SubAssembly 1	Component1	Attribut														
			Attribut2														
		Component2	Attribut3														
	Attribut4																
	Assy2	Component3	Attribut5														
			Attribut6														
Process	Process1	Task1	Subtask1														
			Subtask2														
		Task2	Subtask3														
	Subtask4																
	Pro2	Task3	Subtask5														
			Subtask6														

Tabel 2.3 Contoh *Matriks*

## 2.4 Age Replacement

Menurut Nakagawa dan Mizutani (2009) dalam Dania, Purwaningsih, & Aristiono (2011: 50), terdapat dua model penentuan jadwal optimal penggantian berdasarkan minimasi *downtime* yaitu *optimal preventive replacement interval (Block Replacement)* dan *optimal preventive replacement age (Age Replacement)*.

Menurut T. Wang (2002) dalam Dania, Purwaningsih, & Aristiono (2011: 50) menyatakan bahwa *age replacement policy* lebih efisien dibandingkan dengan *block replacement* karena dalam *block replacement* ada kemungkinan komponen diganti sebelum mencapai umur maksimal. Maka jadwal penggantian selanjutnya tetap mengikuti jadwal penggantian yang telah ditetapkan, sehingga pemborosan mungkin terjadi.

Sementara dengan model *age replacement* terjadi apabila adanya kerusakan komponen sebelum jadwal penggantian yang telah ditetapkan

maka jadwal penggantian selanjutnya berdasarkan interval waktu yang telah ditetapkan dari perbaikan kerusakan terakhir dilakukan.

## 2.5 Sistem

Menurut Parno (2012) sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Christianti & Bobby, 2011).

### 2.5.1. Sistem Informasi

Sistem informasi menurut Robert A. Leitch merupakan suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan (Christianti & Bobby, 2011).

### 2.5.2. Guna Informasi

Kegunaan sistem informasi menurut (O'Brien & Marakas, 2010) adalah:

1. Kontributor penting untuk efisiensi operasional dan produktif karyawan.
2. Sumber utama informasi dan dukungan yang diperlukan untuk mempromosikan keputusan yang efektif.
3. Sebuah sumber yang diakui nilai perusahaan.
4. Komponen kunci dari sumber daya infrastruktur dan kemampuan jaringan saat ini
5. Salah satu objek yang dapat diolah sebagai *input* untuk sistem informasi
6. Data yang berasal dari informasi dapat disimpan agar mempermudah dalam mengakses informasi.
7. Informasi dapat diolah dan menghasilkan *output* ditunjukkan dengan beberapa tampilan.

### 2.5.3. Objek Oriented

Pendekatan *object oriented* dalam penulisan berdasarkan dari model *unified modeling language* (UML). Model ini digunakan untuk menganalisa kasus dan mendapatkan *domain class* dari pengguna sistem yang terlibat. permodelan yang dilakukan dalam UML adalah membuat *use case diagram*, *use case descriptions*, *activity diagram*, *system sequence diagram* dll (Satzinger, Jackson, & Burd, 2010).