BAB 2

LANDASAN TEORI

2.1. *MAINTENANCE*

2.1.1. Pengertian *Maintenance*

Suatu perawatan mesin dan komponennya sangat diperlukan dalam setiap kegiatan produksi agar mesin dapat digunakan secara optimal sesuai dengan kapasitas produksinya, karena mesin yang bermasalah dapat menggangu jalannya produksi dan dapat berpengaruh langsung kepada hasil produksi. Program perawatan mesin dan komponennya harus benar – benar direncanakan, sehingga waktu terhentinya (*downtime*) aktivitas produksi yang merugikan dapat dikurangi menjadi seminimal mungkin.

Maintenance menurut Sofjan *User* (1999, p124) adalah kegiatan untuk memelihara atau menjaga fasilitas dan peralataan pabrik, dan mengadakan perbaikan, penyesuaian, atau penggantian yang diperlukan untuk mendapatkan suatu kondisi operasi produksi yang memuaskan, sesuai dengan yang direncanakan.

Sedangkan menurut Corder (1985, p1) *maintenance* adalah suatu kombinasi dari berbagai tindakan yang dilakukan untuk menjaga suatu barang, atau memperbaikinya sampai, suatu kondisi yang bisa diterima.

2.1.2. Tujuan *Maintenance*

Tujuan pemeliharaan yang utama menurut Corder (1985, p3) dapat didefinisikan sebagai berikut:

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

Dari keterangan diatas dapatlah dinyatakan bahwa perawatan berkaitan erat dengan proses produksi karena kegagalan perawatan akan sangat menggangu kelancaran proses produksi.

2.1.3. Jenis – Jenis *Maintenance*

Menurut Sofjan *User*, kegiatan pemeliharaan yang dilakukan dalam suatu pabrik dapat dibedakan menjadi dua macam, yaitu *Preventive Maintenance* dan *Corrective Maintenance*.

1. *Preventive Maintenance*

   Preventive Maintenance adalah kegiatan pemeliharaan dan perawatan yang dilakukan untuk mencegah timbulnya kerusakan yang tidak terduga dan menemukan kondisi atau keadaan yang dapat menyebabkan fasilitas produksi mengalami kerusakan pada waktu digunakan dalam proses produksi.

   Menurut *User* (1995, p135), perawatan pencegahan (*Preventive Maintenance),* dibagi menjadi :

a. Perawatan rutin (*routine maintenance*) / *Standing order*

Routine maintenance adalah kegiatan pemeliharaan dan perawatan yang dilakukan secara rutin misalnya setiap hari. Sebagai contoh dari kegiatan *routine maintenance* adalah pembersihan fasilitas/peralatan, pelumasan (*lubrication*) atau pengecekan oli, serta pengecekan isi bahan bakar, dan mungkin termasuk pemanasan/*warming up* dari mesin – mesin selama beberapa menit sebelum dipakai beroperasi sepanjang hari.

b. Perawatan berkala (*periodic maintenance*)

Periodic maintenance adalah kegiatan pemeliharaan yang dilakukan secara berkala atau dalam jangka waktu tertentu, misalnya setiap seminggu sekali. *Periodic maintenance* dapat dilakukan dengan memakai lamanya jam kerja mesin atau fasilitas produksi tersebut sebagai jadwal kegiatan, misalnya setiap seratus jam kerja mesin sekali dan seterusnya. Jadi sifat kegiatan *maintenance* ini tetap secara periodic atau berkala. Kegiatan *periodic maintenance* ini bobotnya lebih berat daripada kegiatan *routine maintenance.*

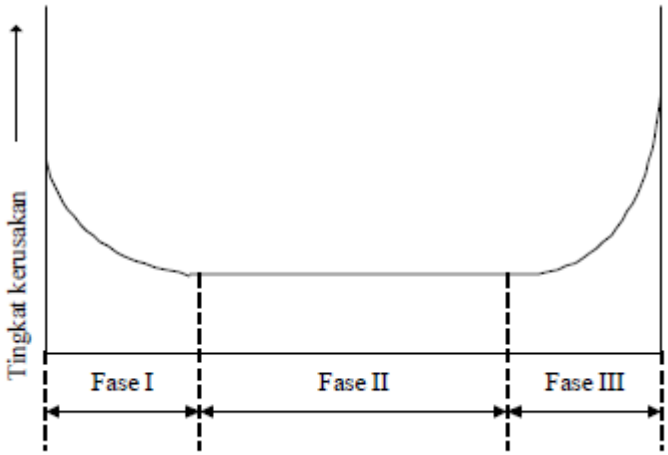
2. *Corrective* atau *Breakdown Maintenance*

Corrective atau *Breakdown Maintenance* adalah kegiatan pemeliharaan dan perawatan yang dilakukan setelah terjadinya suatu kerusakan fasilitas atau peralatan yang mengakibatkan tidak dapat berfungsinya fasilitas atau peralatan tersebut dengan baik. Kegiatan corrective *maintenance* ini sering juga disebut dengan kegiatan perbaikan atau reparasi. *Corrective maintenance* dilakukan karena adanya kerusakan yang terjadi akibat tidak dilakukannya *preventive maintenance* ataupun telah dilakukan

tetapi sampai pada suatu waktu tertentu fasilitas atau peralatan tersebut tetap rusak. Jadi *corrective maintenance* sifatnya menunggu sampai kerusakan terjadi dahulu, baru kemudian dilakukan perbaikan. Maksud tindakan perbaikan ini adalah agar fasilitas atau peralatan tersebut dapat dipergunakan kembali dalam proses produksi, sehingga operasi atau proses produksi dapat berjalan lancar kembali.

2.1.4. Kurva Laju Kerusakan (*Bathub Curve*)

Pola dasar dari fungsi laju kerusakan sesaat yang umum bagi suatu produk dijelaskan melalui kurva yang dikenal denga nama *Bathtub Curve. System* yang laju kerusakannya berbentuk *Bathtub Curve*, mengalami laju kerusakan yang selalu berubah sesuai dengan bertambahnya waktu. Menurut Ebeling (1997, p69-p71), kurva ini memiliki tiga area dengan karakteristik tertentu Karakteristik dari kegagalan atau kerusakan pada produk, mesin ataupun fasilitas sehubungan dengan waktu dapat digambarkan seperti pada gambar di bawah ini:



Gambar 2.1 Bathtub Curve

Dari gambar di atas kita dapat membaginya ke dalam tiga fase yaitu:

1. Fase kerusakan awal (*burn-in/early failures/wear in region*).

   Wilayah Dimana mesin baru mulai digunakan. Pada wilayah ini resiko kerusakan berada berada pada tingkat yang menurun. Terdapat beberap alasan yang menyebabkan terjadinya kerusakan awal ini, diantaranya yaitu pengecekan yang tidak sesuai, pengendalian kualitas yang tidak terpenuhi, material di bawah standar, ketidaksempurnaan perancangan, kesalahan dalam pemasangan dan *set up*, kurangnya kemampuan pekerja dan *Quality Control* yang tidak memenuhi syarat.

2. Fase Kerusakan acak (*Random Failure*)

   Daerah ini ditandai dengan laju kerusakan yang konstan. Fase ini sering juga disebut *Usefull Life Period*. Pada wilayah ini kerusakan sulit diprediksi dan cenderung terjadi secara acak. Contoh penyebab kerusakan pada wilayah ini adalah kesalahan dalam operasional mesin oleh pekerja ataupun perubahan kondisi mesin secara tiba- tiba.

3. Fase Keausan (*wareout*), merupakan wilayah Dimana umur ekonomis dari mesin telah habis dan melewati batas yang diizinkan. Pada fase ini resiko kerusakan akan meningkat (*increasing hazard rate*). Penyebab kerusakan pada wilayah ini umumnya adalah kurangnya perawatan, karena telah dipakai terlalu lama sehingga terjadi karat, keausan, gesekan atau perubahan pada fisik mesin tersebut.

2.2. DISTRIBUSI KERUSAKAN

Fungsi distribusi yang ada pada ilmu statistik sangat berperan didalam teori keandalan. Hal ini dikarenakan penerapan preventive maintenance berhubungan erat dengan permasalahan peluang. Dalam penerapan *preventive maintenance* ini, data waktu kerusakan yang akan dihitung merupakan hasil pengukuran maka data ini termasuk dalam data kontinu. Oleh karena itu, distribusi yang digunakan untuk menghitung waktu kerusakan dan waktu perbaikan adalah dengan distribusi Normal (Gaussian), Lognormal, Exponensial, dan Weibull.

2.2.1. Distribusi Normal (Gaussian)

Distribusi Normal telah berhasil digunakan untuk model kelelahan (*fatigue*) dan keausan (*wear out*) dari mesin. Fungsi kepadatan dari distribusi Normal ini memiliki kurva yang menyerupai lonceng sehingga memiliki nilai simetris terhadap dua parameter yaitu nilai tengah (μ) dan standar deviasi (σ) menurut Ebeling (1997, p69-p71). Fungsi-fungsi dari Distribusi Normal yaitu :

1. Fungsi Kepadatan Probabilitas (*Probability Density Function*)

$$f(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\frac{(t-\mu)^2}{\sigma^2}\right] \text{ untuk } : -\infty < t < \infty$$

2. Fungsi Keandalan (*Realibility Function*)

$$R(t) = 1 - \Phi\left(\frac{t-\mu}{\sigma}\right)$$

3. Fungsi Distribusi Kumulatif (*Cummulative Distribution Function*)

$$F(t) = \Phi\left(\frac{t-\mu}{\sigma}\right)$$

4. Fungsi Laju Kerusakan (*Hazard Rate Function*)

$$\lambda(t) = \frac{f(t)}{1 - \Phi\left(\frac{\ln t - \mu}{\sigma}\right)}$$

2.2.2. Distribusi Lognormal

Distribusi Lognormal dedefinisikan hanya untuk nilai t positif dan lebih sesuai daripada distribusi Normal sebagai distribusi kerusakan. Distribusi ini memiliki dua buah parameter yaitu s, parameter bentuk (*shape parameter*) dan t_{med}, parameter lokasi (*location parameter*).

Seperti distribusi *Weibull*, Lognormal ini dapat memiliki bentuk yang berbeda, sering dijumpai kasus Dimana data yang sesuai dengan distribusi *Weibull* sesuai pula dengan distribusi Lognormal, yaitu :

1. Fungsi kepadatan Probabilitas (*Probability Density Function*)

$$f(t) = \frac{1}{\sqrt{2\pi}st} \exp\left[-\frac{1}{2s^2}\left(\ln\frac{t}{t_{med}}\right)^2\right] \text{ untuk } t \geq 0$$

2. Fungsi Keandalan (*Realibility Function*)

$$R(t) = 1 - F(t)$$

3. Fungsi Distribusi Kumulatif (*Cummulative Distribution Function*)

$$F(t) = \Phi\left(\frac{1}{s}\ln\frac{t_R}{t_{med}}\right)$$

4. Fungsi Laju Kerusakan (*Hazard Rate Function*)

$$\lambda(t) = \frac{f(t)}{1 - \Phi\left(\frac{1}{s}\ln\frac{t}{t_{med}}\right)}$$

2.2.3. Distribusi Exponensial

Distribusi ini ialah salah satu distribusi kerusakan yang biasa terjadi di dalam teknik keandalan. Distribusi Eksponensial memiliki laju kerusakan yang konstan terhadap waktu dan kerusakan yang bersifat acak. Distribusi Eksponensial merupakan salah satu dari distribusi keandalan yang paling mudah dianalisis menurut *Ebeling* (1997,p41). Menurut *Ebeling* (1997,p42), parameter yang digunakan dalam distribusi ini adalah λ. Parameter λ didefinisikan sebagai rata-rata kedatangan kerusakan yang terjadi. Dengan $\lambda(t) = \lambda$, $t \geq 0$, $\lambda > 0$, maka didapatkan fungsi-fungsi dari distribusi Eksponensial yaitu :

1. Fungsi Keandalan (*Realibility Function*)

$$R(t) = e^{-\lambda t}$$

2. Fungsi Kepadatan Probabilitas (*Probability Density Function*)

$$f(t) = \lambda e^{-\lambda t}$$

3. Fungsi Distribusi Kumulatif (*Cummulative Distribution Function*)

$$F(t) = 1 - e^{-\lambda t}$$

4. Fungsi Laju Kerusakan (*Hazard Rate Function*)

$$\lambda(t) = \lambda = \frac{f(t)}{R(t)}$$

2.2.4. Distribusi Weibull

Distribusi *Weibull* merupakan distribusi yang paling banyak digunakan untuk data waktu kerusakan dalam *analysis* keandalan terutama untuk menghitung umur komponen, karena distribusi ini dapat digunakan baik untuk laju kerusakan

meningkat maupun menurun. Menurut *Ebeling* (1997,p58), parameter yang digunakan ada dua, yaitu :

β (Beta) = parameter bentuk (*shape parameter*)

θ (Teta) = parameter skala (*scale parameter*)

Parameter yang digunakan dalam distribusi ini adalah β dan θ dan dengan mengasumsikan $\theta > 0$, $\beta > 0$, $t \geq 0$ maka didapatkan fungsi – fungsi dari distribusi Weibull yaitu :

1. Fungsi Keandalan (*Reliability Function*)

$$R(t) = e^{-\left(\frac{t}{\theta}\right)^{\beta}}$$

2. Fungsi Kepadatan Probabilitas (*Probability Density Function*)

$$f(t) = \frac{\beta}{\theta}\left(\frac{t}{\theta}\right)^{\beta-1} e^{-\left(\frac{t}{\theta}\right)^{\beta}}$$

3. Fungsi Distribusi Kumulatif (*Cummulative Distribution Function*)

$$F(t) = 1 - e^{-\left(\frac{t}{\theta}\right)^{\beta}}$$

4. Fungsi Laju Kerusakan (*Hazard Rate Function*)

$$\lambda(t) = \frac{\beta}{\theta}\left(\frac{t}{\theta}\right)^{\beta-1}$$

Seperti telah dijelaskan melalui fungsi-fungsi di atas, parameter β berpengaruh terhadap Distribusi *Weibull*, hal ini dijelaskan melalui Tabel berikut menurut *Ebeling* (1997,p64).

Tabel 2.1 Pengaruh Nilai β pada Distribusi *Weibull*

Nilai	**Sifat Distribusi**
$0 < \beta < 1$	*Decreasing Failure Rate* (DFR)
$\beta = 1$	*Constant Failure Rate* (CFR)
$1 < \beta < 2$	*Increasing Failure Rate* (IFR), *concave*
$\beta = 2$	*Rayligh Distribution* (LFR)
$\beta > 2$	*Increasing Failure Rate* (IFR), *convex*
$3 \leq \beta \leq 4$	*Increasing Failure Rate* (IFR), *approaches normal distribution*

Efek β terhadap distribusi ini adalah bentuk kurva kerusakan :

- Untuk beberapa nilai yang berbeda $\beta < 1$, berarti Fungsi Kepadatan Probabilitas (PDF) sama dengan eksponensial.
- Untuk nilai β yang besar $\beta \geq 3$, berarti PDF berbetuk simetris seperti distribusi normal.
- Untuk $1 < \beta < 3$, berarti PDF berbentuk miring atau tidak simetris.
- $\beta = 1$, berarti $\lambda(t)$ konstan dan distribusinya identik dengan eksponensial dengan $\lambda = 1/\theta$

Sedangkan nilai θ ialah parameter skala yang memengaruhi nilai rata-rata dan sebaran dari distribusi tersebut.

2.3. IDENTIFIKASI DISTRIBUSI KERUSAKAN DAN PERBAIKAN

Menurut *Ebeling* (1997, p358), maksud dari pengidentifikasian distribusi ini adalah untuk menunujukkan melalui tes statistik dalam hal menerima atau menolak

suatu hipotesis bahwa kerusakan atau perbaikan yang diteliti berasal dari suatu distribusi tertentu.

2.3.1. Index of Fit

Menurut *Walpolle* (1982, p340), persamaan regresi adalah persaamaan matematik yang memungkinkan kita untuk meramalkan nilai-nilai suatu peubah tak bebas dari nilai-nilai satu atau lebih peubah bebas. Hal ini dijelaskan melalui persamaan :

$$\hat{y} = a + bx$$

Dimana : a = Menyatakan intersep atau perpotongan dengan sumbu tegak

b = Kemiringan atau gradiennya

Lambang $\hat{y}$ digunakan untuk membedakan nilai ramalan yang dihasilkan garis regresi dengan nilai pengamatan y yang sesungguhnya untuk nilai x tertentu. Sedangkan nilai gradien dinyatakan dalam :

$$b = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

Untuk Distribusi *Weibull*, Normal dan Lognormal

$$b = \frac{\sum_{i=1}^{n} x_i . v_i}{\sum_{i=1}^{n} x_i^2}$$

Untuk Distribusi Eksponensial

Dimana : n = Jumlah kerusakan yang terjadi

Intersep : $\hat{y} = a + bx$

Menurut *Walpolle* (1982, p370-371), *Analysis* korelasi mencoba mengukur kekuatan hubungan antara dua peubah melalui sebuah bilangan yang disebut *index of fit* atau koefisien korelasi atau koefisien korelasi momen hasil-kali *pearson* yang dilambangkan dengan r. Dengan koefisien korelasi ini, dua peubah dapat diukur hubungannya meskipun memiliki satuan yang berbeda.

$$r = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{\sqrt{\left[n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right]\left[n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2\right]}}$$

Nilai r berada antara -1 sampai dengan1, nilai r yang mendekati -1 atau 1 menunujukkan hubungan yang kuat antara dua peubah acak, sedangkan nilai r yang mendekati nol menunjukkan hubungan yang lemah bahkan mungkin tidak ada hubungan antara kedua peubah acak tersebut.

2.3.2. Identifikasi Awal

Menurut *Ebeling* (1997,p362), identifikasi awal untuk waktu kerusakan dan waktu perbaikan dapat dilakukan dengan dua cara, yaitu dengan *probability plot* dan *least-square curve fitting.*

Probability plot digunakan ketika ukuran sampel terlalu kecil atau bisa juga digunakan untuk data yang tidak lengkap. Metode ini dibuat dengan cara membuat grafik dari data waktu kerusakan atau perbaikan, bila data tersebut menghampiri suatu distribusi maka grafik tersebut akan berbentuk garis lurus.

Cara kedua, yaitu dengan metode *least-square curve fitting*. Metode inilah yang akan dipakai pada pengolahan data. Metode ini dinilai lebih akurat daripada *probability plot* karena subjektivitas untuk menilai kelurusan sebuah garis menjadi berkurang.

Dalam mengidentifikasikan distribusi suatu komponen digunakan *index of fit* (r) yang merupakan ukuran hubungan linear antara peubah x dan y. Pada metode *least-square curve fitting*, distribusi dengan nilai *index of fit* yang terbesarlah yang terpilih. Perhitungan umum pada metode *least-square curve fitting* yaitu :

$$F(t_i) = \frac{i - 0{,}3}{n + 0{,}4}$$

Dimana : i = Data waktu ke-t

n = r = Jumlah kerusakan yang terjadi untuk data lengkap

n = N = Jumlah data yang diamati untuk data sensor

Perhitungan khusus untuk tiap distribusi adalah :

- Distribusi Eksponensial

  $$x_i = t_i$$

  $$y_i = ln\left(\frac{1}{1 - F(t_i)}\right)$$

  Parameter : λ = $b = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$

  Dimana : i = Urutan data kerusakan (1,2,3,...,n)

  t_i = Data kerusakan ke-i

- Distribusi *Weibull*

  $$x_i = \ln t_i$$

  $$y_i = \ln ln\left(\frac{1}{1 - F(t_i)}\right)$$

  Parameter : β = b dan $\theta = e^{\left(\frac{-a}{b}\right)}$

- Distribusi Normal

  $$x_i = t_i$$

$$y_i = z_i = \Phi^{-1}[F(t_i)] = \frac{t_i - \mu}{\sigma}$$

Parameter : $\sigma = \frac{1}{b}$ dan $\mu = -\frac{a}{b}$

- Distribusi Lognormal

$$x_i = \ln t_i$$

$$y_i = z_i = \Phi^{-1}[F(t_i)] = \frac{1}{s}\ln t - -\frac{1}{s}\ln t_{med}$$

Parameter : $s = \frac{1}{b}$ dan e^{-sa}

2.3.3. Uji Kecocokan Distribusi (*Goodness of Fit Test*)

Setelah mendapatkan distribusi terpilih, lantas selanjutnya adalah uji kecocokan distribusi. Uji kecocokan distribusi atau *Goodness of Fit Test* ini adalah membandingkan dua hipotesis yang berlawanan, yaitu :

Ho : Data kerusakan atau perbaikan mendekati suatu distribusi tertentu.

H1 : Data kerusakan atau perbaikan tidak menghampiri suatu distribusi tertentu.

Uji ini terdiri dari perhitungan statistik berdasarkan data yang diamati kemudian dibandingkan dengan nilai kritik pada tabel. Pada umumnya jika tes statistik lebih kecil daripada nilai kritik, maka terima Ho dan bila sebaliknya maka terima H1.

Pada dasarnya ada dua tipe uji kecocokan distribusi yaitu uji secara umum (*General Tests*) dan uji spesifik (*Specific Tests*). Uji secara spesifik lebih akurat dibandingkan dengan uji secara umum karena lebih dikhususkan untuk satu jenis distribusi, sedangkan uji secara umum digunakan untuk lebih dari satu jenis distribusi.

Menurut Ebeling (1997, p393), pengujian yang akan dilakukan adalah Uji *Bartllet* untuk distribusi Eksponensial, Uji *Kolmogorov-Smirnov* untuk distribusi Normal dan Lognormal serta Uji *Mann* untuk distribusi Wiebull.

- ✓ **Uji *Bartllet* untuk distribusi Eksponensial**

Hipotesis yang digunakan untuk uji ini adalah :

H_0 : Data berdistribusi Eksponensial

H_1 : Data tidak berdistribusi Eksponensial

Uji statistiknya :

$$B = \frac{2r\left[ln\left(\frac{1}{r}\right)\sum_{i=1}^{r} t_i - \left(\frac{1}{r}\right)\sum_{i=1}^{r} \ln t_i\right]}{1 + \frac{(r+1)}{6r}}$$

Dimana : r = jumlah kerusakan

t_i = data waktu kerusakan ke-i

B = nilai uji statistik untuk *Bartllet's Test*

H_0 diterima apabila nilai B jatuh dalam wilayah kritis

$$\chi^2_{1-\frac{\alpha}{2},r-1} < B < \chi^2_{\frac{\alpha}{2},r-1}$$

- ✓ **Uji *Mann* untuk Distribusi Weibull**

Hipotesis yang digunakan untuk uji ini adalah :

H_0 : Data berdistribusi Weibull

H_1 : Data tidak berdistribusi Weibull

Uji statistiknya :

$$M = \frac{k_1 \sum_{i=k_1+1}^{r-1}\left[\frac{(\ln t_{i+1} - \ln t_i)}{M_i}\right]}{k_2 \sum_{i=1}^{k_2}\left[\frac{(\ln t_{i+1} - \ln t_i)}{M_i}\right]}$$

$$k_1 = \left[\frac{r}{2}\right], k_2 = \left[\frac{r-1}{2}\right]$$

Dimana : $M_i = Z_{i+1} - Z_i$

$$Z_i = ln\left[-\ln\left(1 - \frac{i-0,5}{n+0,25}\right)\right]$$

Keterangan : M = nilai uji statistik untuk *Mann's Test*

t_i = data wakttu kerusakan ke-i

ti_{+1} = data waktu kerusakan ke-(i+1)

r = n : jumlah unit yang diamati

Bila $M > F_{crit}$ maka H_1 diterima. Nilai F_{crit} diperoleh dari tabel distribusi F dengan $v_1 = 2k_1$ dan $v_2 = 2k_2$.

✓ **Uji *Kolmogorov-Smirnov* untuk Distribusi Normal dan Lognormal**

Hipotesis yang digunakan untuk uji ini adalah :

H_0 : Data berdistribusi Normal (Lognormal)

H_1 : Data tidak berdistribusi Normal (Lognormal)

Uji statistiknya : $D_n = \max \{D_1, D_2\}$

Dimana : $D_1 = max_{1 \le i \le n}\left\{\phi\left(\frac{t_i - \bar{t}}{s}\right) - \frac{i-1}{n}\right\}$

$$D_2 = max_{1 \le i \le n}\left\{\frac{i}{n} - \phi\left(\frac{t_i - \bar{t}}{s}\right)\right\}$$

$$\bar{t} = \sum_{i=1}^{n} \frac{t_i}{n} \; dan \; s^2 = \frac{\sum_{i=1}^{n}(t_i - \bar{t})^2}{n-1}$$

Keterangan : ti = data waktu antar kerusakan ke-i

t = data waktu antar kerusakan

s = standar deviasi

n = banyaknya data kerusakan

Bila $D_n < D_{crit}$ maka terima H_0, dan bila sebaliknya maka terima H_1. Nilai D_{crit} diperoleh dari tabel *critical value for the Kolmogorov-Smirnov test for normality*.

2.3.4 Penentuan Estimasi Parameter (*Maximum Likelihood Estimator*)

Setelah pengujian kecocokan distribusi data telah dilakukan, maka selanjutnya menetukan parameter-parameter. Walaupun sebelumnya pada *least-square curve fitting* telah dihitung parameter-parameter dari distribusi, namun parameter-parameter tersebut bukan merupakan parameter terbaik. Estimasi parameter dengan *Maximum Likelihood Estimator* (MLE) memberikan hasil estimasi yang lebih akurat.

2.3.4.1 MLE untuk Distribusi Eksponensial

Baik untuk data lengkap maupun data sensor, parameter λ diperoleh dari :

$$\lambda = \frac{r}{T}$$

Dimana : r = jumlah kerusakan

T = total waktu pengujian

2.3.4.2 MLE untuk Distribusi *Weibull*

Menurut Ebeling (1997,p377) untuk data lengkap dan sensor tunggal, parameter β diperoleh dengan menyelesaikan persamaan berikut :

$$g(\beta) = \frac{\left(\sum_{i=1}^{r} t_i^{\beta} \ln t_i\right) + (n-r) t_s^{\beta} \ln t_s}{\left(\sum_{i=1}^{r} t_i^{\beta}\right) + (n-r) t_s^{\beta}} - \frac{1}{\beta} - \frac{1}{r} \sum_{i=1}^{r} \ln t_i = 0$$

Sedangkan parameter θ diperoleh dari :

$$\theta = \left\{\frac{1}{r}\left[\sum_{i=1}^{r} t_i^{\beta} + (n-r)t_s^{\beta}\right]\right\}^{\frac{1}{\beta}}$$

Dimana : n = jumlah unit yang diamati

r = jumlah kerusakan yang terjadi

r = n untuk data lengkap

t_i = data waktu kerusakan ke-i

t_s = 1 untuk data lengkap

Persamaan di atas hanya dapat dipecahkan secara numerik. Oleh karena itu, digunakan metode *Newton-Raphson* untuk memecahkan persamaan non linier tersebut yaitu dengan menggunakan persamaan :

$$\beta_{j+1} = \beta_j - \frac{g(\beta_j)}{g'(\beta_j)} \quad domain\ g'(x) = \frac{dg(x)}{dx}$$

yang harus dipecahkan secara iterasi sampai mencapai nilai β_j yang maksimum atau nilai g(β) yang mendekati nol. Maka terlebih dahulu adalah mencari turunan pertama dari g (β) yaitu :

$$g'(\beta) = \frac{\left(\sum_{i=1}^{r} t_i^{\beta} \ln^2 t_i\right)\left(\sum_{i=1}^{r} t_i^{\beta}\right)\left(\sum_{i=1}^{r} t_i^{\beta} \ln t_i\right)}{\left(\sum_{i=1}^{r} t_i^{\beta}\right)} + \frac{1}{\beta^2}$$

Agar penyelesaian iterasi metode *Newton-Raphson* lebih mudah maka nilai β_j awal yang digunakan adalah nilai β yang didapat melalui metode *least square* agar menjadi awal yang baik.

2.3.4.3 MLE untuk Distribusi Normal

Menurut Ebeling (1997,p378) parameter μ dan σ yang digunakan adalah :

$$\mu = \bar{x} = \bar{t}_i = \frac{\sum_{i=1}^{n} t_i}{n}$$

$$\sigma = \sqrt{\frac{(n-1)s^2}{n}} \text{ ; dengan } s = \sqrt{\sum_{i=1}^{n} \frac{(t_i - \bar{t}_i)^2}{n-1}}$$

Dimana : t_i = data waktu kerusakan ke-i

n = jumlah unit yang diamati

2.3.4.4 MLE untuk Distribusi Lognormal

Menurut Ebeling (1997,p378) parameter μ, t_{med} dan s yang digunakan adalah :

$$\mu = \sum_{i=1}^{n} \frac{\ln t_i}{n}$$

$$t_{med} = e^{\mu}$$

$$s = \sqrt{\frac{\sum_{i=1}^{n} (\ln t_i - \mu)^2}{n}}$$

Dimana : t_i = data waktu kerusakan ke-i

n = jumlah unit yang diamati

t_{med} = waktu kerusakan median

2.3.4. MTTF (*Mean Time To Failure*)

Menurut Ebeling (1997,p26), MTTF atau *Mean Time To Failure* adalah nilai rata-rata interval atau selang waktu kerusakan dari suatu distribusi kerusakan yang didefinisikan oleh *probability density function* f(t) sebagai berikut :

$$MTTF = E(t) = \int_0^{\infty} t\, f(t)\, dt$$

atau dapat juga dinyatakan sebagai :

$$MTTF = \int_0^\infty R(t)dt$$

Perhitungan MTTF untuk keempat distribusi adalah sebagai berikut :

1. Distribusi Eksponensial : $MTTF = \frac{1}{\lambda}$
2. Distribusi *Weibull* : $MTTF = \theta\Gamma\left(1+\frac{1}{\beta}\right)$ $domain\ \Gamma(x) =$ fungsi gamma
3. Distribusi Normal : MTTF = μ
4. Distribusi Lognormal : $MTTF = t_{med}e^{\frac{\sigma^2}{2}}$

2.3.5. MTTR (*Mean Time To Repair*)

Menurut Ebeling (1997, p192-193), MTTR atau *Mean Time To Repair* adalah nilai rata-rata atau nilai yang diharapkan dari waktu perbaikan. MTTR dinyatakan sebagai :

$$MTTR = \int_0^\infty t\,h(t)\,dt = \int_0^\infty \left(1 - H(t)\right)dt$$

Dimana : h(t) = fungsi kepadatan probabilitas untuk data waktu perbaikan

H(t) = fungsi distribusi kumulatif untuk data waktu perbaikan

Perhitungan MTTR untuk keempat distribusi adalah sebagai berikut :

1. Distribusi Eksponensial : $MTTR = \int_0^\infty \frac{e^{-t/MTTR}}{MTTR} = 1 - e^{-t/MTTR}$ dan $r = \frac{1}{MTTR}$
2. Distribusi *Weibull* : $MTTF = \theta\Gamma\left(1+\frac{1}{\beta}\right)$ $domain\ \Gamma(x) =$ fungsi gamma
3. Distribusi Normal : MTTF = μ
4. Distribusi Lognormal : $MTTF = t_{med}e^{\frac{\sigma^2}{2}}$

2.4. METODE PERAWATAN MESIN

Menurut Jardine (1973,p94), dalam melakukan tindakan perawatan pencegahan mesin atau *preventive maintenance* ada berbagai metode yang dapat digunakan. Salah satu tujuan dari penelitian ini adalah meminimasi *downtime* yang terjadi akibat jerusakan pada mesin, oleh karena itu pendekatan yang dipakai dalam perhitungan interval penggantian pencegahan dan interval waktu pemeriksaahn adlah dengan menggunakan kriteria minimasi *downtime*.

2.4.1. Interval Waktu Penggantian Pencegahan dengan Kriteria Minimasi *Downtime*

Tujuan dilakukan penggantian pencegahan adalah untuk menentukan waktu terbaik untuk melakukan penggantian pencegahan sehingga dapat meminimasi total *downtime* per unit waktu. Konstruksi model penggantian pencagahan adalah sebagai berikut :

1. T_f = *downtime* yang dibutuhkan untuk melakukan penggantian kerusakan
2. T_p = *downtime* yang dibutuhkan untuk melakukan penggantian pencegahan
3. f(t) = fungsi kepadatan probabilitas waktu kerusakan

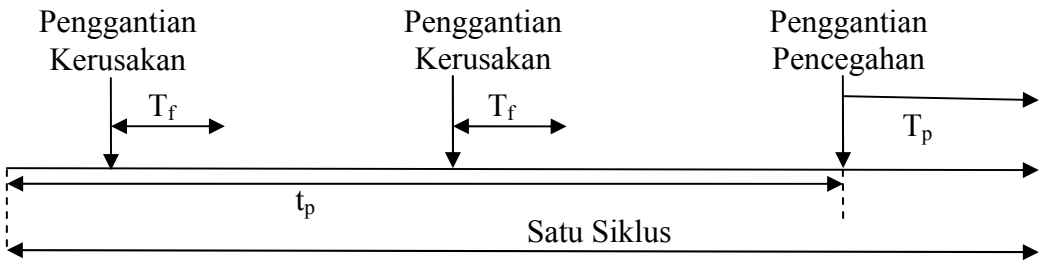
Pada metode ini ada dua model standar bagi permasalahan penggantian, yaitu model *Block Replacement* dan model *Age Replacement*.

1. *Block Replacement*

   Menurut Jardine (1973,p95), Model Penggantian Pencegahan ini dilakukan pada suatu interval yang tetap, serta digunakan jika diinginkan adanya suatu konsistensi

terhadap interval penggantian pencegahan yang telah ditentukan walaupun sebelumnya telah terjadi penggantian yang disebabkan oleh kerusakan.

Dalam model ini akan terdapat kemungkinan Dimana komponen yang baru dipasang setelah penggantian kerusakan harus mengalami penggantian lagi pada saat tiba waktu penggantian pencegahan dalam kurun waktu yang sangat berdekatan. Model penggantian ini diilustrasikan pada gambar berikut :



Gambar 2.2 Model *Block Replacement*

Total *downtime* per unit waktu untuk penggantian pencegahan pada saat t_p dinotasikan dengan $D(t_p)$, yaitu :

$$D(t_p) = \frac{\text{Ekspetasi } downtime \text{ karena kerusakan} + Downtime \text{ karena penggantian kerusakan}}{\text{Panjang siklus}}$$

Downtime karena kerusakan = Jumlah kerusakan pada interval $(0,t_p)$ x Waktu yang dibutuhkan untuk penggantian kerusakan

$= H(t_p) \times T_f$

$$\text{Maka, } D(t_p) = \frac{H(t_p) \times T_f + T_p}{t_p + t_p}$$

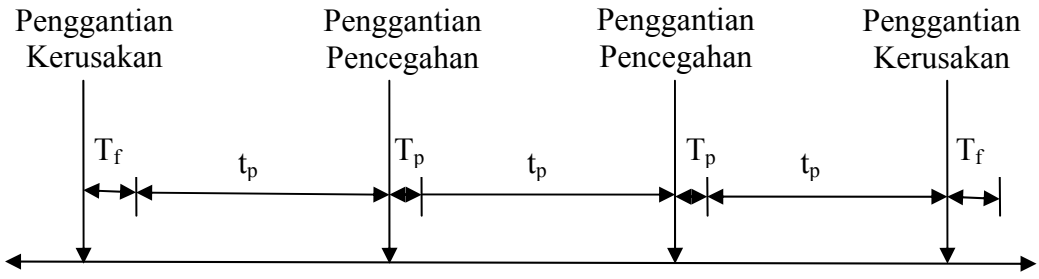
Dimana : t_p = interval waktu penggantian pencegahan

2. *Age Replacement*

Menurut Jardine (1973,p96), Model Penggantian Pencegahan ini dilakukan tergantung pada umur pakai dari komponen. Penggantian pencegahan dilakukan dengan menetapkan kembali interval waktu penggantian pencegahan berikutnya sesuai dengan interval yang telah ditentukan jika terjadi kerusakan yang menuntut dilakukan tindakan penggantian. Terdapat dua macam siklus penggantian pada model ini, yaitu :

- ✓ Siklus pertama ditentukan oleh komponen yang telah mencapai umur penggantian (t_p) sesuai dengan apa yang telah direncanakan atau siklus pencegahan yang diakhiri dengan kegiatan penggantian pencegahan (*Preventive Replacement*).
- ✓ Siklus kedua ditentukan oleh komponen yang telah mengalami kerusakan sebelum mencapai waktu penggantian yang telah ditetapkan sebelumnya atau siklus kerusakan yang diakhiri dengan kegiatan penggantian kerusakan (*Failure Replacement*).

Model penggantian pencegahan ini diilustrasikan pada gambar berikut :



Gambar 2.3 Model *Age Replacement*

Total *downtime* per unit waktu untuk penggantian pencegahan pada saat t_p dinotasikan dengan $D(t_p)$, yaitu :

$$D(t_p) = \frac{\text{Total ekspetasi } \textit{downtime} \text{ per siklus}}{\text{Ekspetasi panjang siklus}}$$

✓ Total ekspektasi *downtime* per siklus (EDS) adalah :

Downtime yang terjadi pada siklus pencegahan (*Preventive Cycle*) x Probabilitas terjadinya siklus pencegahan + ekspektasi *downtime* yang terjadi pada siklus kerusakan (*Failure Cycle*) x Probabilitas terjadinya siklus kerusakan.

$$EDS = T_p.R(t_p) + T_f.[1 - R(T_p)]$$

✓ Ekspektasi panjang siklus kerusakan (EPS) adalah :

Panjang siklus pencegahan x probabilitas terjadinya siklus pencegahan + ekspektasi panjang siklus kerusakan x probabilitas terjadinya siklus kerusakan.

$$EPS = (t_p + T_p).R(t_p) + [M(t_p) + T_f].[1 - R(t_p)]$$

Menurut Jardine (1973,p18), jika f(t) merupakan fungsi kepadatan probabilitas maka probabilitas terjadinya siklus pencegahan [$R(t_p)$] adalah sama dengan probabilitas munculnya kerusakan setelah waktu t_p yang ditunjukkan oleh daerah yang diarsir. Maka :

$$R(t_p) = \int_{t_p}^{\infty} f(t_p)dt_p$$

Menurut Jardine (1973,p89), untuk menentukan ekpektasi panjang siklus kegagalan perlu diperhatikan nilai tengah dari distribusi waktu kerusakan (*Mean Time To Failure* = MTTF) dari suatu distribusi adalah sebagai berikut :

$$MTTF = \int_{-\infty}^{\infty} tf(t)\,dt$$

Menurut Jardine (1973,p96), Dimana pada distribusi normal selang waktu kerusakan ini merupakan rata – rata dari distribusi tersebut. Jika penggantian pencegahan dilakukan pada waktu t_p maka nilai tengah dari distribusi kerusakannya [$M(t_p)$] adalah sebagai berikut :

$$M(t_p) = \frac{\int_{-\infty}^{t_p} t f(t) dt}{1 - R(t_p)}$$

Untuk menyederhanakan perhitungan $M(t_p)$ maka digunakan aturan 1/3 *Simpson*, yaitu ada sebuah titik ekstra ditengah antara f(a) dan f(b), titik ketiga akan dapat dihubungkan dengan sebuah *Problem*.

Menurut Evans (1997, p546-548), Aturan 1/3 Simpson menggunakan penghampiran polinomial interpolasi kuadratik. Jika suatu fungsi :

$$I = \int_a^b f(x) dx, \text{maka}$$

$$I = \int_a^b f(x) dx \cong \frac{h}{3}[(f_0 + f_n) + 4(f_1 + f_3 + .. + f_{n-1}) + 2(f_2 + f_4 + .. + f_{n-2})]$$

Dimana : a,b = selang pembatas

$$h = \frac{b - a}{n}$$

n = sub bagian selang (nilai n harus genap)

Jadi menurut Jardine (1973, p96), total *downtime* per unit waktu adalah :

$$D(t_p) = \frac{T_p.R(t_p) + T_f[1 - R(t_p)]}{(t_p + T_p).R(t_p) + [M(t_p) + T_f][1 - Rt_p]}$$

Dimana : T_f = waktu untuk melakukan perbaikan kerusakan komponen

T_p = waktu untuk melakukan penggantian pencegahan

t_p = panjang interval waktu antara tindakan perawatan pencegahan

F(t) = fungsi kepadatan peluang dari waktu kegagalan komponen

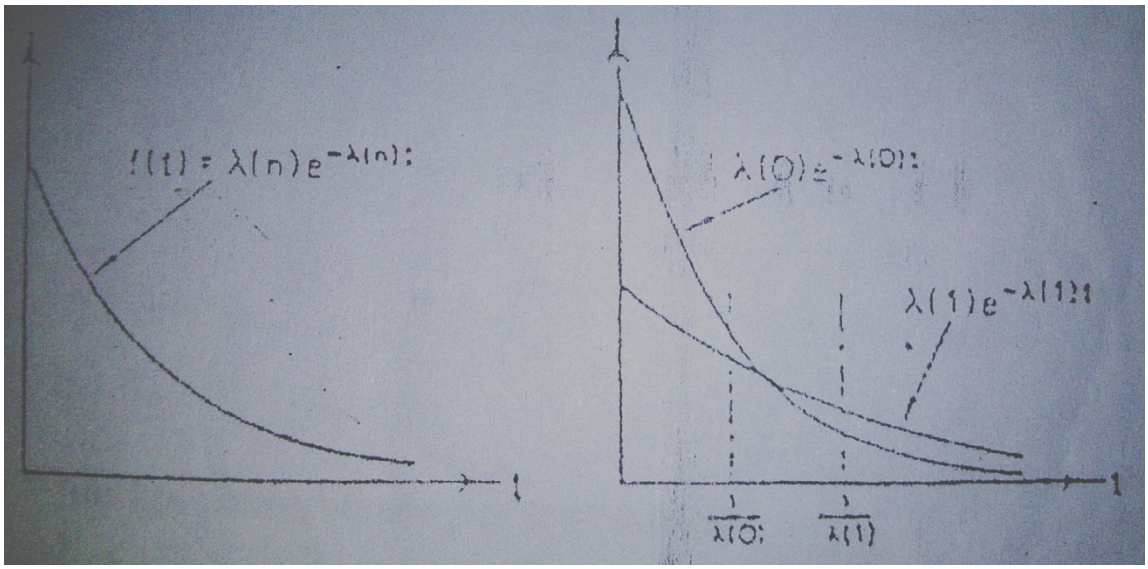
$R(t_p)$ = nilai fungsi keandalan

$M(t_p)$ = nilai harapan panjang siklus kerusakan

2.4.1. Interval Waktu Pemeriksaan Optimal Dengan Kriteria Minimasi *Downtime*

Dalam melaksanakan tindakan perawatan, selain melakukan penggantian pencegahan juga diperlukan pemeriksaan. Model pemeriksaan ini juga brdasarkan kriteria minimasi *downtime*. Melalui model pe\ndekatan ini diharapkan laju kerusakan mesin dapat berkurang sehingga *downtime* dapat diminimasi serta dapat meningkatkan tingkat availibility mesin. Konstruksi model pemeriksaan optimal adalah sebaga berikut:

1. Kerusakan mesin atau komponen terjadi mengikuti distribusi eksponensial dengan MTTF = $1/\lambda$ Dimana λ adalah rata – rata terjadinya kerusakan. Contohnya jika MTTF sebuah komponen = 0.25 tahun, maka rata – rata kerusakan dalam setahun $\lambda = 1/0.25 = 4$.
2. Banyaknya perbaikan kerusakan adalah berdistribusi eksponensial dengan waktu rata – rata = $1/\mu$.
3. Kebijakan pemeriksaan adalah n pemeriksaan per unit waktu. Banyaknya pemeriksaan adalah berdistribusi eksponensial dengan waktu rata – rata = 1/i.
4. Laju kerusakan mesin atau komponen merupakan fungsi dari frekuensi pemeriksaan (n). Hal ini berarti bahwa breakdown dipengaruhi oleh jumlah pemeriksaan yang dilakukan. Oleh karena itu $\lambda = \lambda(n)$.



Gambar 2.4 Pengaruh Jumlah Pemeriksaan Terhadap *Breakdown*

Pada gambar terlihat bahwa :

λ(0) = nilai *breakdown* jika tidak dilakukan sama sekali tindakan pemeriksaan.

λ(1) = nilai *breakdown* jika dilakukan satu kali tindakan pemeriksaan.

Dari gambar tersebut dapat dilihat bahwa efek dilakukannya pemeriksaan adalah meningkatnya nilai *Mean Time To Failure*. Total *downtime* per satuan waktu dapat dijabarkan dalam bentuk suatu fungsi dari frekuensi pemeriksaan (n) dan dinotasikan sebagai D(n).

D(n) = *Downtime* yang terjadi karena perbaikan per unit waktu + *downtime* yang terjadi karena pemeriksaan per unit waktu.

$$D(n) = \frac{\lambda(n)}{\mu} + \frac{n}{i}$$

Dimana : λ(n) = laju kerusakan yang terjadi

1/μ = waktu yang diperlukan untuk melakukan perbaikan

1/i = waktu yang diperlukan untuk melakukan pemeriksaan

n = jumlah pemeriksaan (frekuensi) yang dilakukan per unit waktu

Bila diasumsikan laju kerusakan mesin atau komponen berbanding terbalik dengan jumlah kerusakan, yaitu $\lambda(n) = k/n$ sehingga $\lambda(n) = -(k/n^2)$ dan D(n) merupakan fungsi kontinu terhadap n, maka agar diperoleh n dengan memberikan total *downtime* D(n) minimum adalah dengan melakukan penurunan rumus D(n).

$$D'(n) = \frac{\lambda'(n)}{\mu} + \frac{n}{t}$$

$$= -\frac{k}{n^2 . \mu} + \frac{1}{t} = 0$$

Sehingga,

$$n = \sqrt{\frac{k.t}{n}}$$

Dimana : n = jumlah pemeriksaan per unit waktu i

i = waktu rata-rata pemeriksaan per unit waktu

k = rata-rata kerusakan per unit waktu

μ = waktu rata-rata perbaikan

2.5. Perhitungan Keandalan Tanpa Dan Dengan Perawatan Pencegahan

Menurut Ebeling (1997,p204) Peningkatan keandalan atau *realibility* dapat ditempuh dengan cara melakukan perawatan pencegahan. Model keandalan berikut ini dapat mengurangi efek dari *wearout* dan mempunyai pengaruh yang signifikan terhadap umur hidup *system* dengan mengasumsikan *system* kembali ke kondisi awal setelah dilakukan tindakan perawatan pencegahan.

Bila R(t) adalah keandalan *system* tanpa tindakan perawatan pencegahan, sedangkan T adalah interval waktu antara tindakan perawatan pencegahan, dan $R_m(T)$ adalah keandalan *system* kumulatif setelah tindakan perawatan pencegahan, maka :

$R_m(t) = R(t)$ untuk $0 \leq t < T$

$R_m(t) = R(T)\ R(t-T)$ untuk $T \leq t < 2T$

Sehingga didapatkan persamaan secara umum sebagai berikut :

$R_m(T) = R(T)^n \times R(t-nT)$ untuk $nT \leq t < (n+1)T$

$n = 0, 1, 2, \ldots$

Dimana :

R(t-T) = probabilitas nilai keandalan pada waktu lebih t – T dan *system* akan kembali ke kondisi awal saat T.

$R(T)^n$ = probabilitas nilai keandalan pada n interval tindakan perawatan (T)

R(t-nT) = probabilitas nilai keandalan selama t – nt unit waktu setelah tindakan perawayan pencegahan yang terakhir.

Nilai realibilitas berbeda-beda tergantung dengan distribusi kerusakannya, rumus yang digunakan untuk keempat distribusi adalah :

- Distribusi Eksponensial

  Nilai realibilitas tanpa perawatan pencegahan : $R(t) = e^{-\lambda t}$

  Nilai reliabilitas dengan perawatan pencegahan :

  $R(t - nt) = R(t) = e^{-\lambda t}$ $nT \leq t < (n+1)T$

  Untuk Distribusi Eksponensial, *preventive maintenance* tidak mempunyai efek apapun.

- Distribusi *Weibull*

  Nilai realibilitas tanpa perawatan pencegahan :

$$R(t) = exp\left[-\left(\frac{t}{\theta}\right)^{\beta}\right]$$

  Nilai reliabilitas dengan perawatan pencegahan :

$$R(t-nT) = exp\left[-\left(\frac{t-nT}{\theta}\right)^{\beta}\right] \qquad nT \leq t < (n+1)T$$

- Distribusi Normal

  Nilai realibilitas tanpa perawatan pencegahan :

$$R(t) = 1 - \Phi\left(\frac{t-\mu}{\sigma}\right)$$

  Nilai reliabilitas dengan perawatan pencegahan :

$$R(t-nT) = 1 - \Phi\left(\frac{t-\mu-nT}{\sigma}\right) \qquad nT \leq t < (n+1)T$$

- Distribusi Lognormal

  Nilai realibilitas tanpa perawatan pencegahan :

$$R(t) = 1 - \Phi\left(\frac{1}{s}\ln\frac{t}{t_{med}}\right)$$

  Nilai reliabilitas dengan perawatan pencegahan :

$$R(t-nT) = 1 - \Phi\left(\frac{1}{s}\ln\frac{t-nT}{t_{med}}\right) \qquad nT \leq t < (n+1)T$$

2.6. Single *Minute Exchange of Dies*

Menurut Nicholas (1998,p182) beberapa jenis aktivitas-aktivitas *setup* yang pada umumnya dilakukan di industri adalah sebagai berikut :

Jenis 1 : Persiapan, pengecekkan material, pengcekkan peralatan untuk sebelum proses *setup* berlangsung dan membersihkan mesin, membersihkan tempat kerja, mengecek dan mengmbalikan peralatan, material, dan lain-lain untuk setelah proses *setup* selesai.

Jenis 2 : Memindahkan peralatan, *parts*, dan lain-lain untuk setelah penyelesaian *lot* terakhir lalu menata *parts*, pealatan dan lain-lain untuk sebelum *lot* selanjutnya.

Jenis 3 : Mengukur, men-*setting*, dan mengkalibrasi mesin, peralatan, *fixture* dan *part* pada saat proses berlangsung.

Jenis 4: Memproduksi suatu produk contoh setelah *setting* awal selesai dan mengecek produk contoh tersebut apakah sesuai standar produk. Kemudian menyetel mesin dan memproduksi produk kembali dan seterusnya sampai menghasilkan produk yang sesuai standar.

Menurut Nicholas (1998, p182) Tahapan dalam metode SMED antara lain :

a. Tahap klasifikasi elemen internal dan elemen eksternal

Aktivitas internal adalah aktivitas-aktivitas yang harus dilakukan pada saat mesin mati, waktu internal *setup* sama halnya dengan *downtime*. Aktivitas eksternal adalah aktivitas-aktivitas yang dapat dilakukan pada saat proses produksi berlangsung. Berdasarkan jenis aktivitas, kebanyakan jenis aktivitas 1 adalah aktivitas eksternal, dan kebanyakan jenis aktivitas 2, 3, dan 4 adalah aktivitas internal. Fokus utama pada pengurangan waktu *setup* bukan pada waktu total *setup* (internal+eksternal) tidak juga pada jam kerja adalah sebuah kepentingan sekunder.

b. Tahap konversi elemen internal menjadi elemen eksternal

Tahap ini melibatkan dua pengertian pneting, yaitu :

- Pengujian kembali operas-operasi apakah ada langkah yang salah diasumsikan menjadi internal.

- Mencari jalan untuk mengganti dua langkah ini menjadi aktivitas eksternal.

c. Tahap perbaikan kegiatan internal dan eksternal.

Perbaikan ini dilakukan untuk memberikan usulan ide pengembangan sehingga dapat mengurangi waktu kegiatan.

2.7. Sistem Informasi

2.7.1 Pengertian Sistem

Sistem menurut Jogiyanto (2003, p34) adalah ”Kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu”. Menurut O’Brien (2005, p29) sistem merupakan sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur.

Sistem semacam ini memiliki tiga komponen atau fungsi yang berinteraksi:

- *Input* melibatkan penangkapan dan perakitan berbagai elemen yang memasuki sistem untuk diproses.
- Pemrosesan melibatkan proses transformasi yang mengubah *input* menjadi *output*.
- *Output* melibatkan pemindahan elemen yang telah diproduksi oleh proses transformasi ketujuan akhir.

Berdasarkan definisi di atas, maka dapat disimpulkan bahwa yang dimaksud dengan sistem adalah jaringan prosedur pengelompokan data yang dilakukan oleh manusia mulai dari pengumpulan data (*input*) kemudian *analysis* data (proses) yang terdiri dari pengolahan, penyimpanan dan penghapusan data ataupun informasi sebagai hasil olahan, sampai akhirnya pengambilan data untuk penyebaran informasi (*output*).

2.7.2 Pengertian Informasi

O'brien (2005, p38) mendefinisikan informasi merupakan data yang telah diubah menjadi konteks yang berarti dan berguna bagi pemakai akhir tertentu. Dengan kata lain bahwa informasi merupakan data yang dapat dimengerti oleh pengguna dan memiliki arti. Mc Leod (2001, p145) berpendapat bahwa informasi dikatakan berkualitas jika data tersebut bersifat relevan, akurat, tepat pada waktunya dan lengkap, yaitu :

- Relevan artinya informasi yang diberikan harus sesuai dengan yang dibutuhkan. Apabila kebutuhan informasi ini untuk suatu organisasi, maka informasi tersebut harus sesuai dengan kebutuhan informasi di berbagai tingkatan dan bagian yang ada dalam organisasi tersebut.
- Akurat artinya informasi harus mencerminkan keadaan yang sebenarnya. Pengujian terhadap hal ini biasanya dilakukan melalui pengujian yang dilakukan oleh dua orang atau lebih yang berbeda dan apabila hasil pengujian tersebut menghasilkan hasil yang sama data tersebut dianggap akurat.
- Tepat waktu artinya informasi harus tersedia pada saat yang dibutuhkan untuk memecahkan masalah sebelum situasi krisis menjadi tidak

terkendali atau kesempatan menghilang. Informasi yang datang pada penerima tidak boleh terlambat karena informasi yang sudah usang tidak mempunyai nilai lagi.

- Lengkap artinya bahwa informasi yang diperoleh menyajikan gambaran lengkap dari suatu permasalahan atau penyelesaian.

2.7.3 Pengertian Sistem Informasi

Menurut O'brien(2005, p5) sistem informasi merupakan kombinasi teratur apapun dari orang-orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan sistem informasi dalam sebuah organisasi.

Dari pengertian diatas sistem informasi merupakan gabungan dari beberapa elemen-elemen yang digunakan untuk memberikan informasi yang berarti. Menurut O'brien (2005, p10) terdapat tiga alasan mendasar untuk semua aplikasi bisnis dalam teknologi informasi, yaitu :

1. Mendukung proses dan operasi bisnis
2. Mendukung pengambilan keputusan para pegawai dan managernya
3. Mendukung berbagai strategi unuk keunggulan kompetitif

Alasan diperlukannya sistem informasi dalam suatu organisasi adalah sebagai berikut:

- Perkembangan teknologi yang semakin kompleks.
- Semakin pendeknya waktu untuk pengambilan keputusan.
- Lingkungan bisnis yang semakin kompetitif.

- Untuk sinkronisasi aktivitas – aktivitas dalam organisasi sehingga semua sumber daya dapat dimanfaatkan seefektif mungkin.
- Pengaruh kondisi ekonomi internasional.
- Meningkatnya kompleksitas dari aktifitas bisnis / organisasi.

Dalam suatu organisasi, sistem informasi memiliki beberapa peran dasar yaitu sistem informasi berusaha memberikan informasi aktual tentang lingkungan dari organisasi tersebut sehingga organisasi mendapat gambaran yang akurat tentang lingkungannya. Dalam permasalah aliran informasi, sistem informasi selalu berusaha agar elemen-elemen di dalam organisasi selalu kompak dan harmonis dimana tidak terjadi duplikasi kerja dan lepas satu sama lain. Melalui hal tersebut dapat dilihat manfaat dari sistem informasi adalah:

- Memprediksi masa depan
- Melancarkan operasi organisasi
- Menstabilkan beroperasinya organisasi
- Membantu pengambilan keputusan.
- Menjadikan organisasi lebih efisien dan lebih efektif
- Lebih cepat tanggap dalam merespon perubahan
- Mengelola kualitas *output*
- Memudahkan melakukan fungsi control

2.8. *Analysis* dan Perancangan Berorientasi Objek *(Object-Oriented Analysis and Design*)

Object-Oriented Analysis and Design adalah suatu metode yang digunakan untuk menganalisis dan merancang suatu sistem dengan menggunakan pendekatan berorientasi pada objek (Mathiassen, 2000, p35). *Object* memiliki arti suatu entitas yang memiliki identitas (identity), *state* dan *behavior* (Mathiassen, 2000, p4). Pada *analysis*, identitas objek menggambarkan bagaimana seorang *user* membedakannya dari objek yang lain, dan *behavior* objek digambarkan melalui *event* yang dilakukannya. Pada perancangan, identitas sebuah objek digambarkan dengan bagaimana cara objek lain mengenalinya sehingga dapat diakses dan *behavior* objek digambarkan dengan *operation* yang dapat dilakukan objek tersebut dapat mempengaruhi objek lain di dalam sistem. Pendekatan ini menggunakan objek dan *Class* sebagai konsep utamanya.

- Objek

  Objek merupakan suatu entitas yang memiliki identitas, *state* dan *behavior*. (Mathiassen, 2000, p4). Sebagai contoh misalnya kita memilih barang sebagai suatu objek, maka barang tersebut harus memiliki identitas, status dan perilaku yang berbeda dengan objek lain, demikian juga dengan cara pengaksesannya.

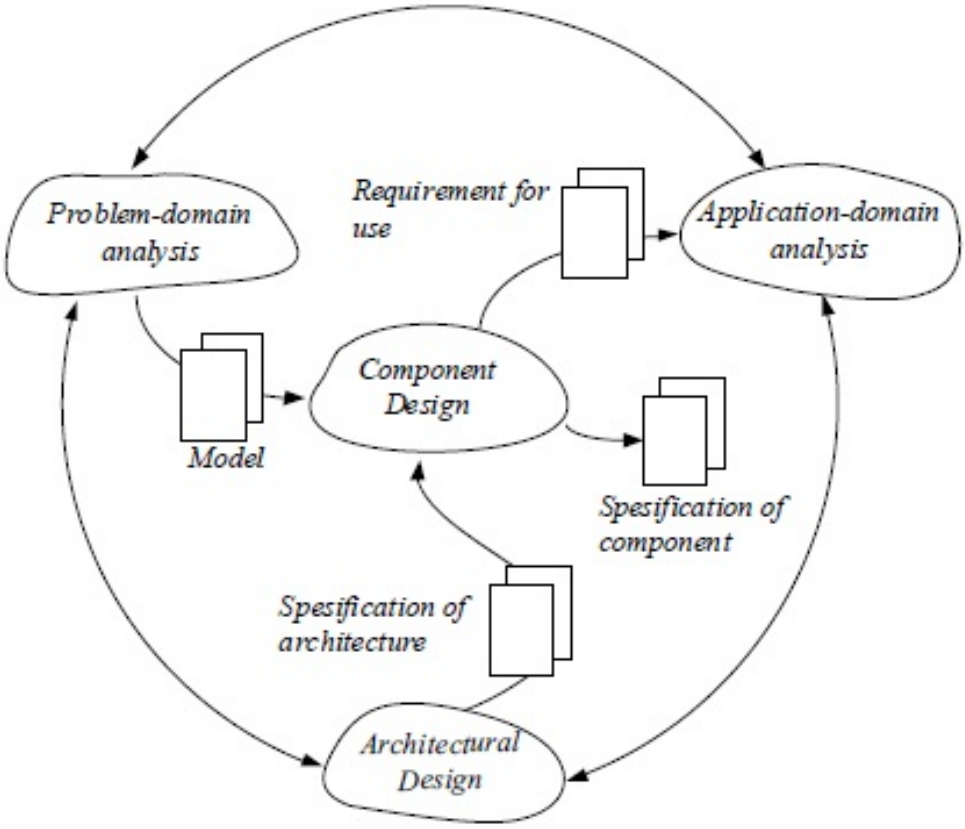
- *Class*

  Class merupakan sebuah deskripsi dari sekumpulan objek yang memiliki struktur, pola *behavior* dan atribut yang sama (Mathiassen, 2000, p4). Sebagai contoh misalnya sekumpulan *Class* barang mungkin mengandung objek barang yang spesifik, tetapi *Class* yang sama juga mengandung banyak barang lainnya, Dimana masing-masing objek yang ada di dalamnya memiliki identitas, status dan perilakunya masing-masing yang unik.

2.9. Kegiatan Utama dalam *Object-Oriented Analysis and Design* (OOAD)

Dengan pendekatan *Object-Oriented Analysis and Design* (OOAD), baik eksekusi yang sudah ada maupun pengaturan kerja baru dideskripsikan. Pada OOA ditekankan adalah *state* pandang dari *user*. Pada OOAD setidaknya ada dua hal penting di dalamnya yaitu (Mathiassen, 2000, p135-136):

- OOAD adalah metode untuk menganalisis dan merancang sistem. Metode harus dilengkapi dengan teori dan metode yang berkaitan dengan perancangan dari pengaturan proses kerja.
- OOAD adalah metode *Object* oriented.
- Jika penting, metode harus dilengkapi dengan metode pengembangan sistem lainnya yang akan mendukung fokus yang lebih kuat pada penggunaan *analysis* dan perancangan.. Dalam penggunaan OOAD menurut Mathiassen (2000, p14-44) terdapat empat aktivitas utama dan digambarkan seperti berikut ini:



Sumber : Mathiassen (2000, p15).

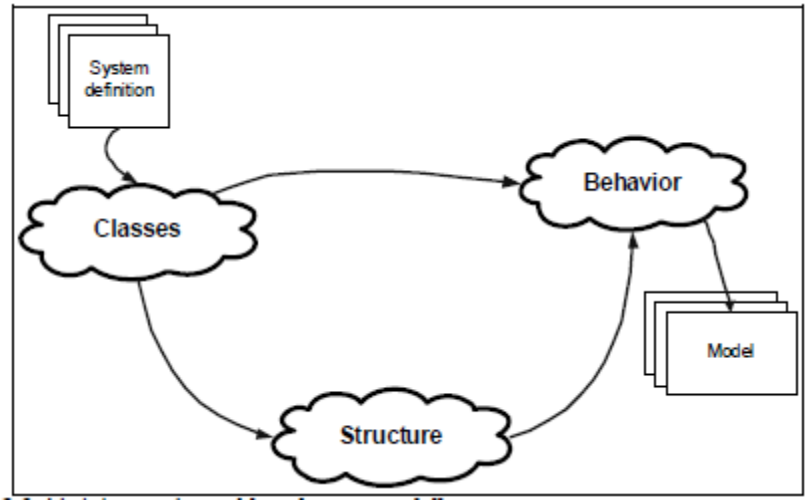
Gambar 2.5 Aktivitas Utama OOAD

Menurut Mathiassen et al. (2000,p.24), *sistem definition* adalah sebuah deskripsi singkat dari *system* yang terkomputerisasi yang dijelaskan dalam bahasa natural. Tujuan dari *system definition* adalah untuk memilih *system* aktual yang akan dikembangkan. *System definition* di sini dapat berupa narasi singkat mengenai *system* yang akan dikembangkan mencakup kegunaan dan kebutuhan dari *system* yang akan dikembangkan agar dapat memenuhi kebutuhan informasi dalam perusahaan.

Tabel 2.2 Tabel FACTOR *Criteria*

Functionality	Fungsi *system* yang mendukung *application* Dimana
Application Dimana	Bagian dari organisasi, administrasi, monitor, atau kontrol Problem Domain
Conditions	setelah *system* akan dikembangkan dan digunakan
Technology	Teknologi yang digunakan dalam pengembangan *system* dan teknologi yang akan menjalankan *system*
Object	*Object* utama dalam Problem Domain
Responsibility	Tanggung jawab keseluruhan *system* dalam hubungannya dengan *context*

2.9.1 *Analysis Problem* Domain.



Gambar 2.6 Aktivitas *Problem Domain*

Sumber *:* Mathiassen et al. (2000, p.46)

Problem Domain analysis dibagi menjadi tiga aktivitas seperti yang diperlihatkan pada gambar 2.6 di atas. Pada *Problem Domain analysis* terdapat tiga aktivitas utama yaitu:

2.9.1.1 Classes

Aktivitas ini meliputi pendefinisian dan pembuatan karakteristik Problem Domain dengan memilih *Class* dan *event* yang menghasilkan *event table*. Menurut Mathiassen et al. (2000, p.53), *Class* merupakan deskripsi dari kumpulan objek yang memiliki struktur, pola *behavior* dan atribut yang sama. Kegiatan *Class* akan menghasilkan *event table*. Dimensi horizontal dari *event table* berisi kelaskelas yang terpilih, sementara dimensi vertikal berisi *event-event* terpilih dan tanda cek digunakan untuk mengindikasikan objek-objek dari *Class* yang berhubungan dalam *event* tertentu.

2.9.1.2 Structure

aktivitas ini menekankan pada penggambaran hubungan antara *class* dan *object* yang ada pada *problem domain* sehingga menghasilkan *class* diagram.

Menurut Mathiassen et al. (2000, p.69) kegiatan ini bertujuan untuk menjelaskan hubungan struktural antara *Class-Class* dan objek-objek pada Problem Domain. Ada empat tipe hubungan struktural Dimana keempatnya dibagi ke dalam dua bagian yaitu:

1. *Class* structure
   - Generalization adalah suatu *Class* yang umum (*Class* super) yang menggambarkan properti umum untuk suatu grup yang memiliki *Class* khusus (sub *Class*).
   - Cluster adalah suatu koleksi dari *Class* yang berhubungan.
2. *Object* structure
   - Aggregation : adalah suatu objek superior (keseluruhan) yang berisi jumlah dari objek atau bagiannya.

– Association : adalah hubungan yang berarti antar sejumlah objek. Hasil dari kegiatan stuktur ini adalah *Class* diagram. *Class* Diagram menghasilkan ringkasan model *Problem*-Dimana yang jelas dengan menggambarkan semua struktur hubungan statik antar *Class* dan objek yang ada dalam model dari *system* yang berubah-ubah.

2.9.1.3 *Behavior*

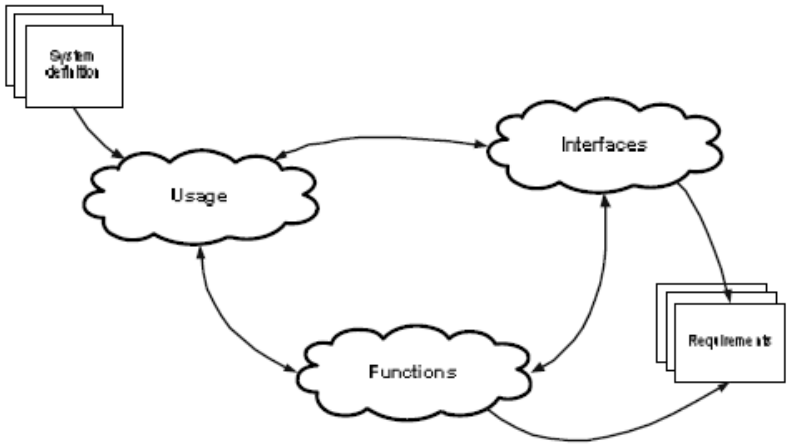
Aktivitas ini menggambarkan properti yang dinamis dan atributatribut dari setiap *Class* yang dipilih. Menurut Mathiassen et al. (2000, p.89), kegiatan ini bertujuan untuk memberi model dinamis pada Problem Domain. Tugas utama dalam kegiatan ini adalah menggambarkan pola perilaku (*behavior pattern*) dan atribut dari setiap *Class*. Hasil dari kegiatan ini adalah statechart diagram.

Menurut Mathiassen (2000, p93) ada 3 notasi untuk *behavior* pattern yaitu:

- *Sequence*, Dimana *event* muncul satu per satu secara berurutan.
- Selection, Dimana terjadi pemilihan satu *event* dari sekumpulan *event* yang muncul.
- Iteration, Dimana sebuah *event* muncul sebanyak nol atau beberapa kali.

2.9.2 *Analysis Application domain*

Menurut Mathiassen et al. (2000,p.115), *application domain* adalah organisasi yang mengatur, memonitor, atau mengendalikan Problem Domain. Hasil dari *application domain* adalah list lengkap dari kebutuhan pengguna *system* secara keseluruhan. Gambar di bawah ini menunjukkan aktivitas dalam *application domain analysis.*



Gambar 2.7 Aktivitas *Application domain*

Sumber : Mathiassen et. al. (2000, p.115)

2.9.2.1 Usage

Menurut Mathiassen et al. (2000,p.119), kegiatan usage merupakan kegiatan pertama dalam *analysis application domain* yang bertujuan untuk menentukan bagaimana aktor-aktor berinteraksi dengan *system* yang dituju. Definisi actor itu sendiri adalah suatu abstraksi dari pengguna atau *system* lain yang berhubungan dengan sasaran dari *system*, sedangkan pengertian use case adalah suatu pola dari interaksi antara *system* dan aktor dari *application domain*. Hasil dari *analysis* kegiatan usage ini adalah deskripsi lengkap dari semua use case dan aktor yang ada digambarkan dalam *table* aktor dan use case diagram. Dengan demikian dapat dikatakan bahwa usecase diagram adalah sebuah diagram yang menggambarkan pola hubungan interaksi antara actor dan *system*, serta menjelaskan apa saja yang actor lakukan dengan menggunakan *system*.

2.9.2.2 Function

Menurut Mathiassen et al. (2000, p.138), function adalah suatu fasilitas untuk membuat suatu model yang berguna untuk actors. Function memfokuskan pad

bagaimana cara sebuah *system* dapat membantu aktor dalam melaksanakan pekerjaan mereka. Function memiliki empat tipe berbeda yaitu:

a. *Update*, fungsi ini disebabkan oleh *event* Problem Domain dan menghasilkan perubahan dalam *state* atau keadaan dari model tersebut.
b. *Signal*, fungsi ini disebabkan oleh perubahan keadaan atau *state* dari model yang dapat menghasilkan reaksi pada konteks.
c. *Read*, fungsi ini disebabkan oleh kebutuhan informasi dalam pekerjaan aktor dan mengakibatkan *system* menampilkan bagian yang berhubungan dengan informasi dalam model.
d. *Compute,* fungsi ini disebabkan oleh kebutuhan informasi dalam pekerjaan aktor dan berisi perhitungan yang melibatkan informasi yang disebabkan oleh aktor atau model, hasil dari fungsi ini adalah tampilan dari hasil komputasi.

Tujuan dari kegiatan function adalah untuk menentukan kemampuan *system* memproses informasi. Hasil dari kegiatan ini adalah sebuah daftar lengkap dari fungsifungsi dengan spesifikasi dari fungsi yang kompleks.

2.9.2.3 Interfaces

Menurut Mathiassen et al. (2000, p.151), *Interface*s adalah fasilitas yang membuat suatu model dan fungsi dapat dipakai oleh aktor. Ada dua jenis *interface* atau antar muka yaitu : antar muka pengguna yang menghubungkan pengguna dengan *system* (*user interface*) dan antar muka *system* yang menghubungkan *system* dengan *system* yang lainnya (*system interface*). Hasil dari kegiatan ini adalah sebuah deskripsi elemen-elemen *user interface* dan elemen-elemen *system interface* yang lengkap, Dimana kelengkapan menunjukkan pemenuhan kebutuhan *user*. Hasil ini dilengkapi dengan

sebuah diagram navigasi yang menyediakan sebuah ringkasan dari elemen-elemen *user interface* dan perubahan antara elemen-elemen tersebut.

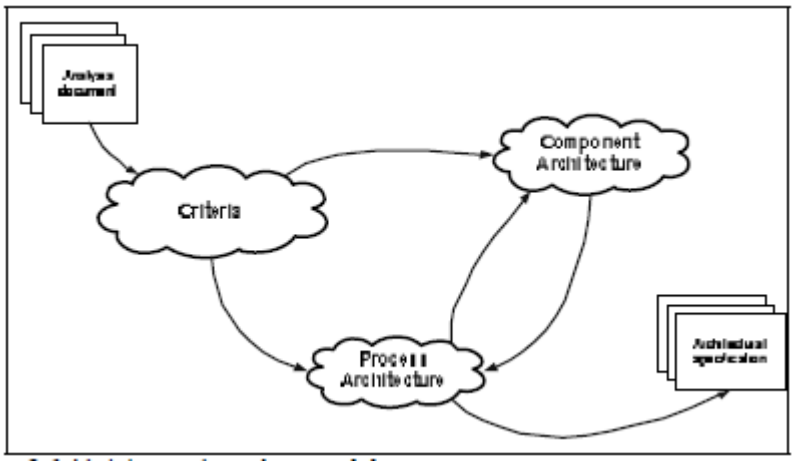
2.9.2.4 *Sequence* Diagram

Menurut Mathiassen et al. (2000, p340), *sequence* diagram menjelaskan tentang interaksi diantara beberapa objek dalam jangka waktu tertentu. *Sequence* diagra melengkapi *Class* diagram, yang menjelaskan situasi yang umum dan statis. Sebuah *sequence* diagram dapat mengumpulkan rincian situasi yang kompleks dan dinamis melibatkan beberapa dari kebanyakan *Object* yang digeneralisasikan dari *Class* pada *Class* diagram.

Menurut Bennet et al. (2006, p252-253), *sequence* diagram secara semantic ekuivalen dengan diagram komunikasi untuk interaksi yang sederhana. Sebuah *sequence* diagram menunjukkan interaksi antara objek yang disusun dalam satu *sequence*. Dalam *sequence* diagram yang diadaptasi dari Bennet, et al.(2006, p.252), terdapat satu buah notasi yang disebut fragment. Fragment ini biasa digunakan dalam setiap tipe UML diagram. Fragment yang digunakan pada *sequence* diagram dimaksudkan untuk memperjelas bagaimana *sequence* ini saling dikombinasikan. Fragment terdiri dari beberapa jenis interaction operator yang menspesifikasikan tipe dari kombinasi fragment.

2.9.3 Architecture Design

Menurut Mathiassen et al. (2000, p.173), tujuan dari *architecture design* adalah untuk menstrukturkan sebuah *system* yang terkomputerisasi. Aktivitas-aktivitas yang dilakukan dalam architecture design dapat dilihat pada Gambar 2.7 dibawah ini



Gambar 2.8 *Aktivitas Architecture Design*

Sumber : Mathiassen, et.al. (2000, p.176)

2.9.3.1 Criteria

Pada kegiatan ini, kita mendefinisikan kondisi dan rancangan seperti apakah yang digunakan pada perancangan. Kondisi adalah peluang dan batasan teknikal, organisasional dan manusia yang terlibat dalam pelaksanaan tugas. Criteria yang digunakan dalam menentukan kualitas dari *software* yang akan dibuat ditunjukkan melalui tabel di bawah ini:

Tabel 2.3 Criteria

Criterion	**Ukuran**
Usable	Kemampuan *system* beradaptasi dengan konteks teknikal Dan organisasional.
Secure	Pencegahan akses ilegal terhadap data dan fasilitasperusahaan
Efficient	Eksploitasi ekonomis dari fasilitas technical platform
Correct	Kesesuaian dengan kebutuhan
Reliable	Fungsi dapat dijalankan secara tepat

Maintainable	Biaya untuk mencari dan memperbaiki kerusakan *system*
Testable	Biaya untuk menjamin bahwa *system* melakukan fungsinya
Flexible	Biaya memodifikasi *system*
Comprehensible	Usaha yang diperlukan untuk memahami *system*
Reusable	Penggunaan bagian dari *system* ke dalam *system* lain yangberkaitan
Portable	Biaya memindahkan *system* ke technical platform lainInteroperable Biaya pemasangan *system* dengan *system* lain

Sumber : Mathiassen et al, 2000, p.178

Tidak ada ukuran dan cara-cara yang pasti untuk menghasilkan suatu desain yang baik. Menurut Mathiassen et al. (2000, p.186), sebuah desain yang baik memiliki tiga ciri-ciri yaitu :

1. Tidak memiliki kelemahan yang bersifat major Syarat ini menyebabkan adanya pendekatan pada evaluasi dari kualitas berdasarkan review atau eksperimen dan membantu dalam menentukan prioritas dari kriteria yang akan mengatur dalam kegiatan desain.
2. Menyeimbangkan beberapa kriteria Konflik sering terjadi antar kriteria, oleh sebab itu untuk menentukan kriteria mana yang akan diutamakan dan bagaimana cara untuk menyeimbangkannya dengan kriteria-kriteria yang lain bergantung pada situasi *system* tertentu.
3. Usable, flexible, dan comprehensible Kriteria-kriteria ini bersifat universal dan digunakan pada hampir setiap proyek pengembangan *system*.

2.9.3.2 Component Architecture

Menurut Mathiassen et al. (2000, p.189-200), arsitektur komponen adalah sebuah struktur *system* yang terdiri dari komponen-komponen yang saling berhubungan. Komponen sendiri merupakan kumpulan dari bagian-bagian program yang membentuk suatu kesatuan dan memiliki fungsi yang jelas. Beberapa pola umum dalam desain komponen arsitektur :

2.9.3.3 Process Architecture

Menurut Mathiassen et al. (2000, p.209-219), arsitektur proses adalah struktur dari eksekusi *system* yang terdiri dari proses-proses yang saling tergantung. *System* berorientasi objek yang berjalan terdiri dari banyak sekali objek, diantaranya Active *Object* merupakan objek yang telah diberikan sebuah proses dan komponen program, sebuah modul fisik dari kode program.

Beberapa pola distribusi dalam kegiatan desain process architecture :

1 *Centralized pattern*

Pada pola ini semua data ditempatkan pada server dan *client* hanya menghandle *user interface* saja. Keseluruhan model dan semua fungsi bergantung pada server, dan *client* hanya berperan seperti terminal.

2 Distributed pattern

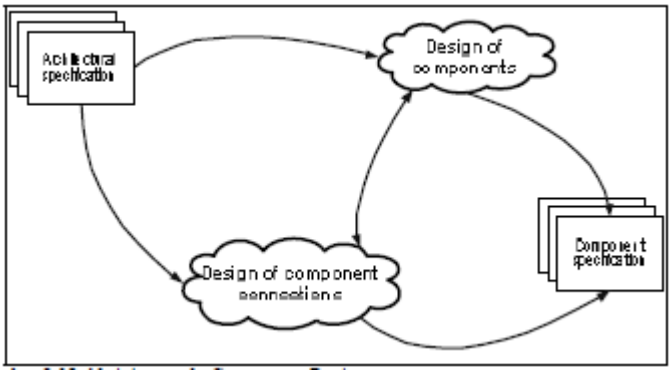
Pola ini merupakan kebalikan dari *centralized pattern*. Semua didistribusikan kepada *client* dan server hanya diperlukan untuk melakukan update model diantara *client*s.

3 Decentralized pattern

Pola ini dapat dikatakan merupakan gabungan dari kedua pola sebelumnya. Pada pola ini, *client* mengimplementasikan model yang local, sedangkan server-nya memakai model common (umum)."

2.9.4 Component Design

Menurut Mathiassen et al. (2000, p.231), desain komponen bertujuan untuk menentukan implementasi kebutuhan dalam kerangka arsitektural. Hasil dari kegiatan ini adalah spesifikasi dari komponen yang saling berhubungan. *Component design* diilustrasikan pada gambar 2.9 dibawah ini.



Gambar 2.9 Aktivitas *Component Design*
Sumber : Mathiassen, et. al. (2000, p232)

2.9.4.1 Model Component

Menurut Mathiassen et al. (2000, p.235), Model component adalah suatu bagian dari *system* yang mengimplementasikan *Problem Domain*. Tujuan dari komponen model adalah untuk mengirimkan data sekarang dan *historic* ke *function*, *interface* dan pengguna dan *system* yang lain.

Langkah – langkah yang harus dilakukan adalah mempresentasikan private *event*, mempresentasikan common *event* dan restrukturisasi *Class*. Hasil yang didapat dalam model component adalah *Class* diagram dari model component yang sudah direvisi

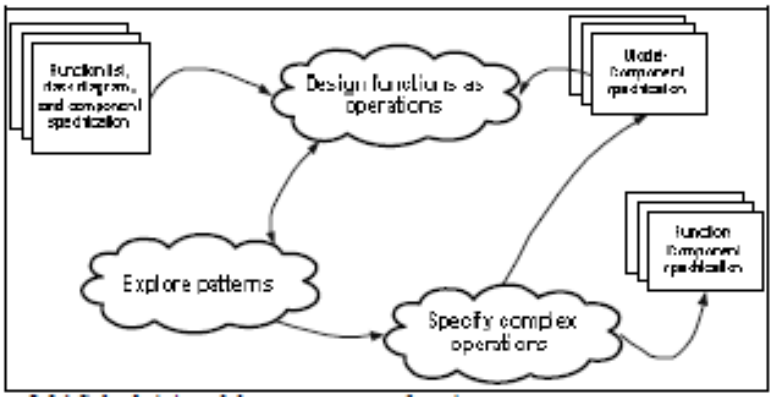
Revisi dari *Class* diagram dapat dilakukan dengan memperhatikan *private event* dan c*ommon event*. *Private event* adalah *event* yang hanya melibatkan hanya satu *Object* Dimana.

2.9.4.2 *Function Architecture*

Menurut Mathiassen et al. (2000, p.251), function component adalah bagian dari *system* yang mengimplementasikan kebutuhan fungsional. Tujuan dari komponen

function adalah untuk memberikan akses bagi *user interface* dan komponen *system* lainnya ke model, oleh karena itu function component adalah penghubung antara model dan usage.

Hasil dari kegiatan ini adalah *Class* diagram untuk komponen function dan perpanjangan dari *Class* diagram komponen model. Berikut adalah sub kegiatan dalam perancangan komponen function dapat dilihat pada Gambar 2.10 dibawah ini:



Gambar 2.10 Aktivitas Function Architecture

Sumber : Mathiassen, et. al. (2000, p252)

Sub aktivitas ini menghasilkan kumpulan operasi yang dapat mengimplementasikan fungsi *system* seperti yang ditentukan dalam Problem Domain *analysis* dan function list. Berikut adalah sub kegiatan dalam component function :

1 Merancang function sebagai *operation*, yaitu mengidentifikasi tipe utama dari function tersebut. Ada empat tipe function yaitu update, read, compute dan signal

2 Menelusuri pola yang dapat membantu dalam implementasi function sebagai *operation*.

3 Spesifikasikan operasi yang kompleks. Ada tiga cara untuk melakukannya yaitu *operation specification*, *sequence* diagram dan statechart diagram.