

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Peramalan**

Definisi peramalan sendiri sebenarnya beragam; berikut beberapa definisi tentang peramalan (Santoso, 2009):

1. Perkiraan munculnya sebuah kejadian di masa depan berdasarkan data yang ada di masa lampau.
2. Proses menganalisis data historis dan data saat ini untuk menentukan pola di masa mendatang.
3. Proses estimasi dalam situasi yang tidak diketahui.
4. Pernyataan yang dibuat tentang masa depan.
5. Penggunaan ilmu dan teknologi untuk memperkirakan situasi di masa depan.
6. Upaya sistematis untuk mengantisipasi kejadian atau kondisi di masa depan.

Dari beberapa definisi di atas, dapat disimpulkan bahwa peramalan berkaitan dengan upaya memperkirakan apa yang terjadi di masa depan, berbasis pada metode ilmiah (ilmu dan teknologi) serta dilakukan secara sistematis. Walaupun demikian, kegiatan peramalan tidaklah semata-mata berdasarkan prosedur ilmiah atau terorganisir, karena ada kegiatan peramalan yang menggunakan intuisi (perasaan) atau lewat diskusi informal dalam sebuah grup.

Tabel 2.1 Ciri Sebuah Kegiatan Peramalan  
Sumber : Santoso (2009)

Aspek	Peramalan
Fokus	Data di masa lalu
Tujuan	Menguji perkembangan saat ini dan relevansinya di masa mendatang
Metode	Proyeksi berdasar ilmu statistic, diskusi, dan review program
Orang yang terlibat	Pembuat keputusan, petugas administrasi, praktisi, analisis
Frekuensi	Regular (teratur)
Kriteria Keberhasilan	Tidak sekedar akurasi, namun bersifat pembelajaran

Dari kriteria di atas, terlihat bahwa peramalan adalah kegiatan yang bersifat teratur, berupaya memprediksi masa depan dengan menggunakan tidak hanya metode ilmiah, namun juga mempertimbangkan hal-hal yang bersifat kualitatif, seperti perasaan, pengalaman seseorang dan lainnya (Santoso, 2009).

## 2.2 Tahapan Peramalan

Menurut Santoso (2009), untuk mendapatkan hasil peramalan yang baik dan secara efektif dapat menjawab masalah yang ada, kegiatan peramalan sebaiknya mengikuti tahapan baku berikut ini:

### 1. Perumusan masalah dan pengumpulan data

Tahap pertama yang sebenarnya penting dan menentukan keberhasilan peramalan adalah menentukan masalah tentang apa yang akan diprediksi. Formulasi masalah yang jelas akan menuntun pada ketepatan jenis dan banyaknya data yang akan dikumpulkan. Dapat saja masalah telah ditetapkan, namun data yang relevan tidak tersedia, hal ini akan memaksa diadakannya perumusan ulang atau mengubah metode peramalan.

## 2. Persiapan data

Setelah masalah dirumuskan dan data telah terkumpul, tahap selanjutnya adalah menyiapkan data hingga dapat diproses dengan benar. Hal ini diperlukan, karena dalam praktek ada beberapa masalah berkaitan dengan data yang telah terkumpul:

- a. Jumlah data terlalu banyak
- b. Jumlah data justru terlalu sedikit
- c. Data harus diproses terlebih dahulu
- d. Data tersedia namun rentang waktu data tidak sesuai dengan masalah yang ada.
- e. Data tersedia namun cukup banyak data yang hilang (*missing*), yakni data yang tidak lengkap.

## 3. Membangun model

Setelah data dianggap memadai dan siap dilakukan kegiatan prediksi, proses selanjutnya adalah memilih (model) metode yang tepat untuk melakukan peramalan pada data tersebut.

## 4. Implementasi model

Setelah metode peramalan ditetapkan, maka model dapat diterapkan pada data dan dapat dilakukan prediksi pada data untuk beberapa periode ke depan.

## 5. Evaluasi peramalan

Hasil peramalan yang telah ada kemudian dibandingkan dengan data aktual. Metode peramalan tidak dapat memprediksi data di masa depan secara tepat yang ada adalah ketepatan prediksi. Untuk itu, pengukuran kesalahan

peramalan dilakukan untuk melihat apakah metode yang telah digunakan sudah memadai untuk memprediksi sebuah data.

### 2.3 Jenis Data pada Kegiatan Peramalan

Data yang akan diprediksi secara umum dapat dibagi menjadi dua tipe, yakni data kualitatif dan data kuantitatif. Menurut Arsyad (2001), data kualitatif adalah serangkaian observasi di mana tiap observasi yang terdapat dalam sampel (atau populasi) tergolong pada salah satu dari kelas-kelas yang eksklusif secara bersama dan kemungkinannya tidak dapat dinyatakan dalam angka. Sebaliknya data kuantitatif adalah serangkaian observasi yang dapat dinyatakan dengan angka. Data kuantitatif dapat dibagi menjadi dua bagian, yaitu (Santoso, 2009):

1. Data *time series*

Data *time series* adalah data yang ditampilkan berdasarkan waktu, seperti bulanan, data harian, data mingguan atau jenis waktu yang lain. Contoh data *time series* adalah data penjualan bulanan motor di daerah A dari tahun 2000 sampai 2007.

2. Data *cross-sectional*

Data *cross-sectional* adalah data yang tidak berdasar waktu tertentu, namun pada satu (titik) waktu tertentu. Contoh data *cross-sectional* adalah data biaya promosi di sepuluh area pemasaran produk X selama bulan Januari 2008.

### 2.4 Data *Stationer* dan Data Tidak *Stationer*

Data *stationer* adalah data di mana rata-rata nilainya tidak berubah dari waktu ke waktu, atau dapat dikatakan data bersifat stabil. Sebaliknya, data dapat

saja tidak *stationer*, ketika pada uji pola data didapati adanya *trend* atau pola *seasonal* (pengaruh musim) (Santoso, 2009).

## 2.5 Uji Pola Data

Uji pola data pada intinya adalah menguji apakah sebuah data dapat dikatakan *stationer* ataukah tidak. Jika pada data terdapat *trend* atau ada komponen *seasonal* sebaliknya atau siklis, dikatakan bahwa data tidak dapat dikatakan *stationer*. Namun sebaliknya, jika pada data tidak ada *trend*, *seasonal* ataukah siklis, maka data dapat dikatakan *stationer*. Stasioneritas data penting untuk menentukan lebih jauh metode peramalan apa yang tepat dilakukan. Metode untuk data yang *stationer* akan berbeda dengan metode peramalan untuk data yang tidak *stationer* (Santoso, 2009).

Salah satu ciri data *stationer* adalah adanya korelasi antar data penjualan. Otokorelasi dapat membuktikan adanya korelasi antar data penjualan. Pada umumnya, jika sebuah data saling berkorelasi pada jarak waktu yang berdekatan, misalnya antara waktu  $t$  dengan waktu sebelumnya ( $t-1$ ), maka dikatakan data mempunyai kecenderungan berotokorelasi. Besaran korelasi antara data ke  $t$  dan data ke  $t-1$  cukup tinggi, kemudian menurun secara bertahap. Data demikian bisa diduga mempunyai unsur *trend* di dalamnya dan tidak bersifat acak. Sebaliknya, data yang mempunyai korelasi antar waktu yang rendah serta tidak menunjukkan pola penurunan otokorelasi yang bertahap, pada data tersebut dapat dikatakan tidak ada unsur *trend* (Santoso, 2009).

Cara lain yang cukup praktis adalah melihat tampilan grafik data. Jika ada unsur *trend*, maka terlihat data dari waktu ke waktu cenderung semakin naik atau semakin turun. Sebaliknya, jika data *stationer*, grafik akan relatif tidak

menaik atau menurun. Ada pola kenaikan data yang diikuti oleh pola data yang menurun (Santoso, 2009).

Pengujian stasioneritas data dapat dilakukan dengan dua cara yaitu dengan grafik (lebih praktis) atau dengan menghitung autokorelasi. Sebaliknya kedua cara dilakukan secara bersama-sama karena saling melengkapi (Santoso, 2009: 49).

Untuk menghitung autokorelasi digunakan rumus sebagai berikut (Santoso, 2009):

$$r_k = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2} \dots\dots\dots(1)$$

dengan:

$Y_t$  = data saat ini (yang ke t)

$\bar{Y}$  = rata-rata data

$Y_{t-k}$  = data pada periode k sebelum data saat ini. Jika k=1, maka menjadi  $Y_{t-1}$  atau data satu lag dari periode sebelumnya. Jika k=2, maka menjadi  $Y_{t-2}$  atau data

dua lag dari periode awal.

$n$  = jumlah data

Setelah nilai autokorelasi didapat, maka langkah selanjutnya menguji nilai autokorelasi dengan menggunakan uji t, proses pengujian dengan menggunakan t, Santoso (2009):

1. Merumuskan hipotesis, yang secara standar dapat dinyatakan sebagai berikut:

$H_0: \rho = 0$ , atau koefisien korelasi yang didapat tidak signifikan

H1:  $1 \bullet 0$ , atau koefisien korelasi yang didapat memang nyata.

2. Menghitung t hitung dan t tabel.

t hitung didapat dengan rumus:

$$t = \frac{r_1}{SE(r_1)} \dots\dots\dots(2)$$

dengan

$$SE = \frac{\sqrt{1+2 \sum_{i=1}^{k-1} r_i^2}}{n} \dots\dots\dots(3)$$

t tabel dihitung dari tabel t pada tingkat kepercayaan tertentu (biasanya 95%) dan dengan  $df = n - 1$ .

3. Pengambilan keputusan:

Jika t hitung < t tabel, H0 diterima

Jika t hitung > t tabel, H0 ditolak

## 2.6 Pemulusan Eksponensial *Winter's*

Salah satu bentuk pemulusan lain yang berguna dikembangkan oleh Winter's pada awal dekade 1960-an. Metode ini memberikan hasil yang serupa dengan pemulusan eksponensial linier, tetapi memiliki manfaat tambahan dalam kemampuannya untuk menangani data musiman di samping data yang memiliki *trend* (Makridakis dan Wheelwright, 1992).

Pemulusan eksponensial linier dan musiman *Winter's* didasari oleh tiga persamaan, yang masing-masing memuluskan satu faktor yang berkaitan dengan satu diantara tiga komponen pola - faktor *random*, *trend*, dan musiman. Dalam hal ini, metode ini serupa dengan pemulusan eksponensial linier, yang memuluskan

faktor *random* dan menyesuaikan dengan *trend*. Tetapi, metode *Winter's* mencakup parameter tambahan untuk menangani faktor musiman.

Keempat persamaan yang digunakan dalam model Winter adalah sebagai berikut (Arsyad, 1994):

1. Pemulusan Eksponensial:

$$A_t = \alpha \frac{Y_t}{S_{t-L}} + (1 - \alpha)(A_{t-1} + T_{t-1}) \dots \dots \dots (4)$$

2. Estimasi *Trend*:

$$T_t = \beta(A_t - A_{t-1}) + (1 - \beta)T_{t-1} \dots \dots \dots (5)$$

3. Estimasi Musiman:

$$S_t = \gamma \frac{Y_t}{A_t} + (1 - \gamma)S_{t-L} \dots \dots \dots (6)$$

4. Ramalan pada periode  $p$  di masa datang

$$\hat{Y}_{t+p} = (A_t + pT_t)S_{t-L+p} \dots \dots \dots (7)$$

dengan:

$A_t$  = nilai pemulusan yang baru

$\alpha$  = konstanta pemulusan untuk data ( $0 \leq \alpha \leq 1$ )

$Y_t$  = data yang baru atau yang sebenarnya pada periode  $t$

$\beta$  = konstanta pemulusan untuk estimasi trend ( $0 \leq \beta \leq 1$ )

$T_t$  = estimasi *trend*

$\mu$  = konstanta pemulusan untuk estimasi musiman ( $0 \leq \mu \leq 1$ )

$S_t$  = estimasi musiman

$p$  = periode yang diramalkan



$L$  = panjangnya musim

$\hat{Y}_{t+p}$  = ramalan pada periode  $p$

$t$  = waktu

### 2.7 Mean Absolute Deviation (Rata-rata Penyimpangan Absolut)

Rata-rata penyimpangan absolut (MAD, *Mean Absolute Deviation*) merupakan penjumlahan kesalahan prakiraan tanpa menghiraukan tanda aljabarnya dibagi dengan banyaknya data yang diamati, yang dirumuskan sebagai berikut (Herjanto, 2008).

$$\text{MAD} = \frac{\sum |e_i|}{n}$$

Dalam MAD, kesalahan dengan arah positif atau negatif akan diberlakukan sama, yang diukur hanya besar kesalahan secara absolut.

### 2.8 Sistem Persediaan

Sistem persediaan adalah serangkaian kebijaksanaan dan pengendalian yang memonitor tingkat persediaan dan menentukan tingkat yang harus dijaga, kapan persediaan harus diisi, dan berapa besar pesanan yang harus dilakukan, menurut T. Hani Handoko (1991).

### **Jenis-jenis persediaan fisik**

Menurut T. Hani Handoko (1991), ada beberapa jenis persediaan. Setiap jenis mempunyai karakteristik khusus tersendiri dan cara pengelolaannya yang berbeda. Menurutnya jenisnya, persediaan dapat dibedakan atas :

1. Persediaan bahan mentah atau bahan baku (*raw materials*), yaitu barang-barang berwujud seperti baja, kayu, dan komponen-komponen lainnya yang digunakan dalam proses produksi. Bahan mentah dapat diperoleh dari sumber-sumber alam atau dibeli dari supplier dan atau dibuat sendiri oleh perusahaan untuk digunakan pada proses produksi selanjutnya.
2. Persediaan komponen-komponen rakitan (*purchased parts/components*), yaitu persediaan barang-barang yang terdiri dari komponen-komponen yang diperoleh dari perusahaan lain, dimana secara langsung dapat dirakit menjadi suatu produk.
3. Persediaan bahan pembantu atau bahan penolong (*supplies*), yaitu persediaan barang-barang yang diperlukan dalam proses produksi, tetapi tidak merupakan bagian atau komponen barang jadi.
4. Persediaan barang dalam proses (*work in process*), yaitu persediaan barang-barang yang merupakan keluaran dari tiap-tiap bagian dalam proses produksi atau yang telah diolah menjadi suatu bentuk, tetapi masih perlu diproses lebih lanjut menjadi barang jadi.
5. Persediaan barang jadi (*finished goods*), yaitu persediaan barang-barang yang telah selesai diproses atau diolah dalam pabrik dan siap untuk dijual atau dikirim kepada pelanggan.

Ada 3 (tiga) jenis biaya yang dikeluarkan oleh perusahaan sehubungan dengan pelaksanaan persediaan barang dalam suatu perusahaan yaitu biaya

pemesanan (*ordering cost*), biaya penyimpanan (*holding cost*) dan biaya persediaan (*inventory cost*).

### **2.8.1 Biaya Pemesanan (*Ordering Cost*)**

Merupakan biaya yang terkait secara langsung dengan kegiatan pemesanan yang dilakukan oleh perusahaan. Dengan demikian maka semakin sering perusahaan mengadakan pemesanan, maka biaya pemesanan akan semakin tinggi.

Biaya pemesanan didapatkan dari hasil perhitungan jumlah siklus pemesanan yang dilakukan dikalikan dengan biaya setiap kali melakukan pemesanan.

Contoh dari biaya pemesanan yaitu :

- a. Biaya administrasi
- b. Biaya pengiriman
- c. Biaya bongkar muat barang

### **2.8.2 Biaya Penyimpanan (*Carrying Cost*)**

Biaya penyimpanan merupakan biaya yang harus ditanggung oleh perusahaan sehubungan dengan adanya barang yang harus disimpan di dalam gudang perusahaan. Jumlah dari biaya penyimpanan untuk perhitungan kuantitas pembelian optimal akan terkait langsung dengan jumlah unit barang yang disimpan dalam gudang perusahaan.

Untuk mendapatkan biaya penyimpanan, perlu diketahui bahwa separuh dari persediaan yang ada di perusahaan adalah setengah dari jumlah bahan baku yang dibeli oleh perusahaan. Maka biaya penyimpanan didapatkan dari perhitungan setengah kali jumlah unit yang yang dibeli dikalikan dengan biaya penyimpanan per unit per tahun.

Contoh dari biaya penyimpanan yaitu :

- a. Biaya kerusakan barang
- b. Biaya sewa gudang per unit
- c. Resiko tidak terpakainya barang karena usang

### **2.8.3 Biaya persediaan (*Inventory Cost*)**

Istilah persediaan adalah suatu istilah umum yang menunjukkan segala sesuatu atau sumber daya-sumber daya organisasi yang disimpan dalam antisipasinya terhadap pemenuhan permintaan, menurut T. Hani Handoko (1991).

Yang dimaksud dengan biaya persediaan adalah seluruh total biaya yang dikeluarkan oleh perusahaan berkaitan dengan pengadaan barang untuk memenuhi stock barang di gudang.

Untuk mendapatkan biaya persediaan ini, dapat menjumlahkan antara biaya pemesanan dengan biaya penyimpanan. Semakin sering pesanan dilakukan dan semakin lama barang disimpan di gudang, maka akan semakin tinggi pula biaya persediaan yang dikeluarkan oleh perusahaan.

### **2.9 Metode *Economic Order Quantity* (EOQ)**

Metode manajemen persediaan yang paling terkenal adalah *model-model* EOQ atau *Economic Lot Size* (ELS). Metode-metode ini dapat digunakan baik untuk barang-barang yang dibeli maupun yang diproduksi sendiri. Model EOQ adalah nama yang biasa digunakan untuk barang-barang yang dibeli, sedangkan ELS digunakan untuk barang-barang yang diproduksi secara internal.

Menurut T. Hani Handoko (1991), perbedaan pokoknya adalah untuk ELS, biaya pemesanan meliputi biaya penyiapan pesanan untuk dikirimkan ke perusahaan dan biaya penyiapan mesin-mesin yang diperlukan untuk mengerjakan

pesanan. Model EOQ digunakan untuk menentukan kuantitas pesanan persediaan yang meminimalkan biaya langsung penyimpanan persediaan dan biaya kebalikannya (*inverte cost*) pemesanan persediaan.

Rumusan EOQ yang biasa digunakan adalah :

$$EOQ = \sqrt{\frac{2SD}{H}}$$

dimana :

D = Permintaan per periode waktu.

S = Biaya pemesanan (persiapan pesanan dan penyiapan mesin) per pesanan.

H = Biaya penyimpanan per unit per tahun.

## 2.10 Metode *Re-Order Point* (ROP)

*Reorder point* (ROP) atau titik pemesanan kembali adalah suatu titik minimum atau batas dari jumlah persediaan yang ada pada suatu saat dimana pemesanan harus kembali dilakukan. Menurut Herjanto (2008), *Reorder Point* merupakan jumlah persediaan yang menandai saat harus dilakukan pemesanan ulang sedemikian rupa sehingga kedatangan atau penerimaan barang yang dipesan adalah tepat waktu (dimana persediaan di atas persediaan pengaman sama dengan nol).

*Reorder Point* terjadi apabila jumlah persediaan yang dimiliki berkurang mendekati nol, dengan demikian perusahaan harus menentukan berapa banyaknya minimal tingkat persediaan yang harus dipertimbangkan agar tidak terjadi kekurangan atau kehabisan persediaan.

Rumus ROP yang biasa digunakan adalah :

$$\text{Reorder point (ROP)} = (\text{LT} \times \text{AU}) + \text{SS}$$

Dimana:

LT = *Leadtime* (jangka waktu antara pesanan pelanggan dan pengiriman produk akhir).

AU = *Average Usage* (rata-rata permintaan).

SS = *Safety Stock* (stok pengaman).

### 2.10.1 *Leadtime* atau Waktu Tunggu

Untuk memesan suatu barang sampai barang itu datang diperlukan jangka waktu yang bisa bervariasi dari beberapa jam sampai beberapa bulan. Perbedaan waktu antara saat memesan sampai saat barang datang dikenal dengan istilah waktu tenggang atau *lead time* (Herjanto, 2008).

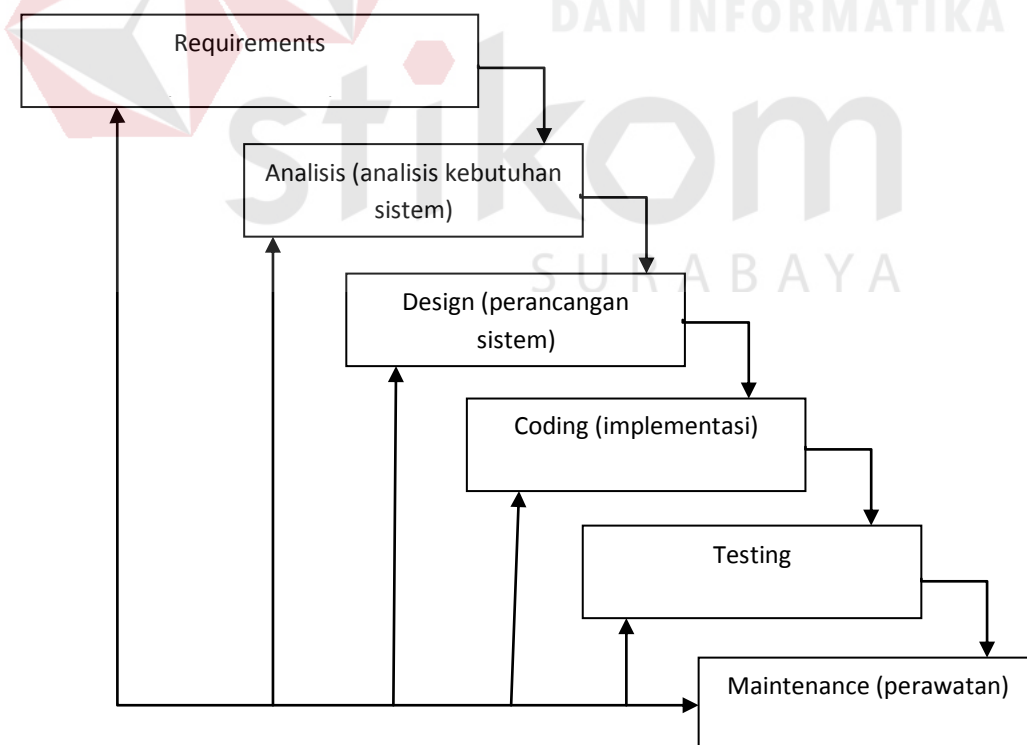
### 2.10.2 *Safety Stock* atau Persediaan Pengaman

Karena adanya waktu tenggang, perlu adanya persediaan yang dicadangkan untuk kebutuhan selama menunggu barang datang, yang disebut sebagai persediaan pengaman (*safety stock*). Persediaan pengaman berfungsi untuk melindungi atau menjaga kemungkinan terjadinya kekurangan barang, misalnya karena penggunaan yang lebih besar dari perkiraan semula atau keterlambatan dalam penerimaan barang yang dipesan. Persediaan pengaman disebut juga dengan istilah persediaan penyangga (*buffer stock*) atau persediaan besi (*iron stock*). Bagi perusahaan dagang, persediaan pengaman juga dimaksudkan untuk menjamin pelayanan kepada pelanggan terhadap ketidakpastian dalam pengadaan barang (Herjanto, 2008).

## 2.11 System Development Life Cycle (SDLC)

Menurut Pressman (1997), Model ini biasa disebut juga dengan model *waterfall* atau disebut juga *classic life cycle*. Adapun pengertian dari SDLC ini adalah suatu pendekatan yang sistematis dan berurutan. Tahapan-tahapannya adalah *Requirements* (analisis sistem), *Analysis* (analisis kebutuhan sistem), *Design* (perancangan), *Coding* (implementasi), *Testing* (pengujian) dan *Maintenance* (perawatan).

Model eksplisit pertama dari proses pengembangan perangkat lunak, berasal dari proses-proses rekayasa yang lain. Model ini memungkinkan proses pengembangan lebih terlihat. Hal ini dikarenakan bentuknya yang bertingkat ke bawah dari satu fase ke fase lainnya, model ini dikenal dengan model *waterfall*, seperti terlihat pada Gambar 2.1.



Gambar 2.1 *System Development Life Cycle* (SDLC) Model Waterfall  
Sumber : Pressman (1997)

Penjelasan-penjelasan SDLC Model *Waterfall*, adalah sebagai berikut:

a. *Requirement* (Analisis Sistem)

Dalam merancang sebuah perangkat lunak, yang pertama harus dilakukan adalah membangun semua elemen sistem yang diperlukan. Sistem merupakan hal yang penting dalam membuat sebuah perangkat lunak, karena perangkat lunak harus berhubungan langsung dengan elemen lainnya seperti perangkat keras, basis data, dan manusia. Tahap ini didefinisikan sebagai sebuah tahap yang menghasilkan sebuah kondisi yang diperlukan oleh pengguna untuk menyelesaikan permasalahan ataupun mencapai sebuah tujuan. Tahap ini bertujuan untuk mengumpulkan kebutuhan-kebutuhan pengguna dan kemudian mentransformasikan ke dalam sebuah deskripsi yang jelas dan lengkap.

b. *Analysis* (Analisis Kebutuhan Sistem)

Pada tahap ini dalam perancangan perangkat lunak, perlu mengetahui karakteristik dasar dari perangkat lunak yang akan dirancang, seperti fungsi, bentuk, dan tampilan dari perangkat lunak tersebut. Tahap analisis sistem ini bertujuan untuk menjabarkan segala sesuatu yang nantinya akan ditangani oleh perangkat lunak. Tahapan ini adalah tahapan pemodelan yang merupakan sebuah representasi *object* di dunia nyata.

c. *Design* (Perancangan Sistem)

Untuk membuat suatu perangkat lunak perlu dirancang struktur datanya, arsitektur perangkat lunak, detail prosedur dan karakteristik tampilan yang akan disajikan. Tahap perancangan perangkat lunak yang merupakan proses multi langkah dan berfokus pada beberapa atribut perangkat lunak yang berbeda, yaitu: struktur data, arsitektur perangkat lunak dan detail algoritma. Proses ini



menterjemahkan kebutuhan ke dalam sebuah model perangkat lunak yang dapat diperkirakan kualitasnya sebelum memulai tahap implementasi.

d. *Coding* (Implementasi)

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

a. *Testing* (Pengujian)

Pengujian program dilakukan untuk mengetahui apabila terjadi kesalahan pada program yang telah dibuat. Dapat juga digunakan untuk memastikan apakah *input* proses dengan benar, sehingga dapat menghasilkan *output* yang sesuai. Tahap ini terdapat 2 metode pengujian perangkat yang dapat digunakan, yaitu: metode *black-box* dan *white-box*. Pengujian dengan metode *black-box* merupakan pengujian yang menekankan pada fungsionalitas dari sebuah perangkat lunak tanpa harus mengetahui bagaimana struktur di dalam perangkat lunak tersebut. Sebuah perangkat lunak yang diuji menggunakan metode *black-box* dikatakan berhasil jika fungsi-fungsi yang ada telah memenuhi spesifikasi kebutuhan yang telah dibuat sebelumnya. Pengujian dengan menggunakan metode *white-box* yaitu menguji struktur internal perangkat lunak dengan melakukan pengujian pada algoritma yang digunakan oleh perangkat lunak.

b. *Maintenance* (Perawatan)

Jika aplikasi tersebut telah sesuai, akan diberikan kepada pengguna dan terdapat penyesuaian atau perubahan sesuai dengan keadaan yang diinginkan, sehingga membutuhkan perubahan terhadap aplikasi tersebut. Tahap ini dapat pula diartikan sebagai tahap penggunaan perangkat lunak yang disertai dengan perawatan dan perbaikan. Perawatan dan perbaikan suatu perangkat lunak diperlukan, termasuk di dalamnya adalah pengembangan, karena dalam prakteknya ketika perangkat lunak digunakan terkadang masih terdapat kekurangan ataupun penambahan fitur-fitur baru yang dirasa perlu.

