

BAB II

LANDASAN TEORI

2.1 Sistem Aplikasi

Menurut Jogianto (2004), sistem berasal dari bahasa latin “Systema“ dan bahasa Yunani “Sustema“ yang berarti “satu kesatuan yang atas komponen atau elemen-elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi”. Sistem adalah beberapa komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan dengan menerima *input* dan menghasilkan output. Dari definisi sistem diatas, dapat disimpulkan bahwa sistem adalah suatu jaringan yang saling berhubungan dan saling memiliki keterkaitan antara bagian dan prosedur-prosedur yang ada terkumpul dalam satu organisasi untuk melakukan kegiatan untuk mencapai suatu tujuan tertentu.

2.2 Perencanaan

Menurut Usman (2011), perencanaan adalah kegiatan yang akan dilaksanakan di masa yang akan datang untuk mencapai tujuan dan dalam perencanaan itu mengandung beberapa unsur, diantaranya sejumlah kegiatan yang ditetapkan sebelumnya, adanya proses, hasil yang ingin dicapai, dan menyangkut masa depan dalam waktu tertentu. Adapun tujuan dari perencanaan sebagai berikut:

1. Untuk memberikan pengarahan baik untuk manajer maupun karyawan non manajerial.
2. Untuk mengurangi ketidakpastian. Ketika seorang manajer membuat rencana, manajer dipaksa untuk melihat jauh ke depan, meramalkan perubahan,

memperkirakan efek dari perubahan tersebut, dan menyusun rencana untuk menghadapinya.

3. Untuk meminimalisir pemborosan. Dengan kerja yang terarah dan terencana, karyawan dapat bekerja lebih efisien dan mengurangi pemborosan. Selain itu, dengan rencana, seorang manajer juga dapat mengidentifikasi dan menghapus hal-hal yang dapat menimbulkan inefisiensi dalam perusahaan.
4. Untuk menetapkan tujuan dan standar yang digunakan dalam fungsi selanjutnya, yaitu proses pengontrolan dan evaluasi. Proses pengevaluasian atau evaluating adalah proses membandingkan rencana dengan kenyataan yang ada. Tanpa adanya rencana, manajer tidak akan dapat menilai kinerja perusahaan.

1.2.1 Persediaan

Menurut Herjanto (2008), persediaan adalah bahan atau barang yang disimpan yang akan digunakan untuk memenuhi tujuan tertentu. Persediaan dapat berupa bahan mentah, bahan pembantu, barang dalam proses, barang jadi, ataupun suku cadang. Bisa dikatakan tidak ada perusahaan yang beroperasi tanpa persediaan, meskipun sebenarnya persediaan hanyalah suatu sumber dana menganggur, karena sebelum persediaan digunakan berarti dana yang terkait di dalamnya tidak dapat digunakan untuk keperluan lain. Beberapa fungsi penting persediaan bagi perusahaan, sebagai berikut:

1. Menghilangkan risiko keterlambatan pengiriman bahan baku atau barang yang dibutuhkan perusahaan.
2. Menghilangkan risiko jika material yang dipesan tidak baik sehingga harus dikembalikan.

3. Menghilangkan risiko terhadap kenaikan harga barang atau inflasi.
4. Untuk menyimpan bahan baku yang dihasilkan secara musiman sehingga perusahaan tidak akan kesulitan jika bahan tersebut tidak terdapat dipasaran.
5. Mendapatkan keuntungan dari pembelian berdasarkan diskon dipasaran.
6. Mendapatkna keuntungan dari pembelian berdasarkan diskon kuantitas.
7. Memberikan pelayanan kepada pelanggan dengan tersedianya barang yang diperlukan.

1.2.2 Perencanaan Persediaan

Menurut Prawisentono (2001), perencanaan persediaan adalah kegiatan memperkirakan kebutuhan persediaan baik secara kualitatif maupun kuantitatif. Adapun tujuan perencanaan sebagai berikut:

1. Agar jumlah persediaan kebutuhan yang disediakan tidak terlalu sedikit dan tidak terlalu banyak, artinya dalam jumlah yang cukup efektif.
2. Operasional perusahaan dapat berjalan secara efektif dan efisien.
3. Implementasi penyediaan kebutuhan demi untuk kelancaran proses produksi dapat disediakan dengan investasi modal dalam jumlah yang memadai.

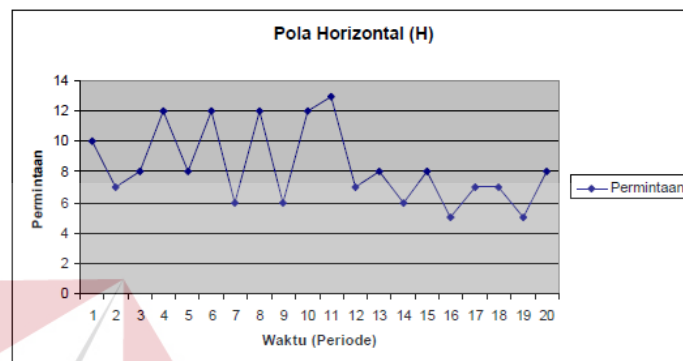
1.3 Peramalan

Menurut Gasperz (2002), Peramalan adalah metode untuk memperkirakan suatu nilai di masa depan dengan menggunakan data masa lalu. Peramalan juga dapat diartikan sebagai seni atau ilmu untuk memperkirakan kejadian pada masa yang akan datang, sedangkan aktivitas peramalan merupakan suatu fungsi bisnis yang berusaha memperkirakan penjualan dan penggunaan produk sehingga produk-produk itu dapat dibuat dalam kuantitas yang tepat.

1.3.1 Pola Data

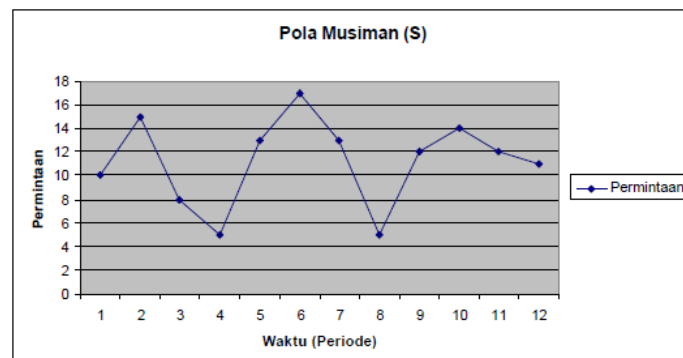
Pola data dapat dibedakan menjadi empat jenis, yaitu:

1. Pola *Horizontal* (H) terjadi apabila nilai data berfluktuasi di sekitar rata-rata yang konstan. Hal ini terjadi pada suatu produk yang penjualannya tidak meningkat atau menurun selama waktu tertentu. Grafik pola *horizontal* disajikan pada gambar 2.1.



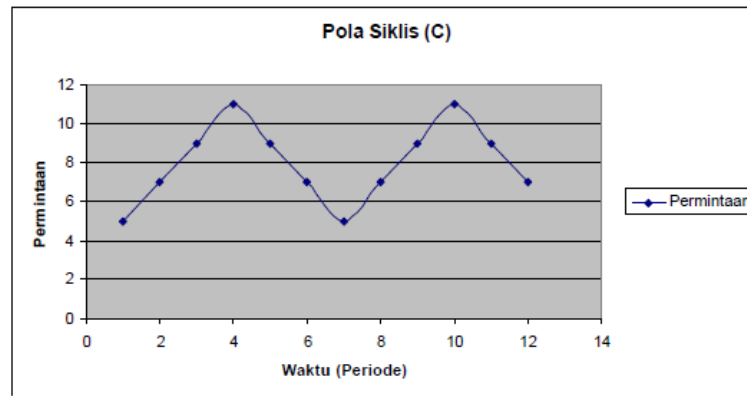
Gambar 2.1 Pola Data Horizontal (H)

2. Pola Musiman (S) terjadi apabila data terlihat berfluktuasi, namun fluktuasi tersebut terlihat berulang dalam suatu *interval* tertentu. Hal ini terjadi karena dipengaruhi oleh faktor musiman seperti faktor cuaca, musim libur panjang yang akan berulang secara periodik setiap tahunnya. Grafik pola musiman disajikan pada gambar 2.2.



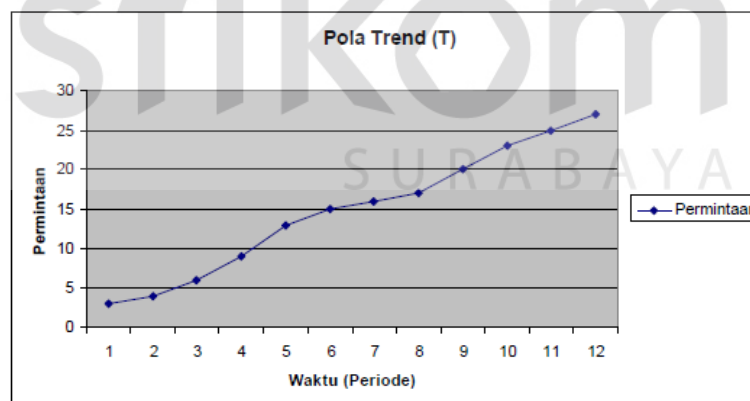
Gambar 2.2 Pola Data Musiman (S)

3. Pola Siklis terjadi apabila fluktuasi permintaan jangka panjang membentuk *sinusoid* atau gelombang / siklus. Biasanya pola ini dipengaruhi oleh siklus bisnis. Grafik pola data siklis disajikan pada gambar 2.3.



Gambar 2.3 Pola Data Siklis (C)

4. Pola *Trend* terjadi apabila data permintaan menunjukkan pola kecenderungan naik atau turun atau bahkan konstan untuk jangka waktu yang panjang. Grafik pola *trend* disajikan pada gambar 2.4.



Gambar 2.4 Pola Data *Trend* (T)

1.3.2 Ukuran Hasil Peramalan

Menurut Nasution & Prasetyawan (2008), ukuran hasil peramalan yang merupakan ukuran kesalahan peramalan merupakan ukuran tentang tingkat

perbedaan antara hasil peramalan dengan permintaan yang sebenarnya terjadi. Ada 3 ukuran yang biasa digunakan, yaitu:

a. Rata-rata Deviasi Mutlak (*Mean Absolute Deviation* = MAD)

MAD merupakan rata-rata kesalahan mutlak selama periode tertentu tanpa memperhatikan apakah hasil peramalan lebih besar atau lebih kecil dibandingkan kenyataannya. Secara sistematis MAS dirumuskan sebagai berikut:

$$\text{MAD} = \frac{\sum_{t=1}^n (Y_t - \hat{Y}_t)}{n} \dots\dots\dots(1)$$

Keterangan:

Y_t = Permintaan aktual pada periode $-t$

\hat{Y}_t = Peramalan permintaan (*forecast*) pada periode $-t$

n = Jumlah periode peramalan yang terlibat

b. Rata-rata Kuadrat Kesalahan (*Mean Square Error* = MSE)

MSE dihitung dengan menjumlahkan kuadrat semua kesalahan peramalan pada setiap periode dan membaginya dengan jumlah periode peramalan. Secara matematis, MSE dirumuskan sebagai berikut:

$$\text{MSE} = \frac{\sum_{t=1}^n (Y_t - \hat{Y}_t)^2}{n} \dots\dots\dots(2)$$

c. Rata-rata Presentase Kesalahan Absolut (*Mean Absolute Percentage Error* = MAPE)

MAPE merupakan ukuran kesalahan relatif. MAPE biasanya lebih berarti dibandingkan MAD karena MAPE menyatakan presentase kesalahan hasil peramalan terhadap permintaan aktual selama periode tertentu yang akan

memberikan informasi presentase kesalahan terlalu tinggi atau terlalu rendah, Secara sistematis, MAPE dinyatakan sebagai berikut:

$$\text{MAPE} = \frac{\sum_{t=1}^n \frac{|Y_t - \hat{Y}_t|}{Y_t}}{n} \times 100\% \dots\dots\dots(3)$$

1.4 Metode Perhitungan Peramalan

1.4.1 Metode Eksponensial *Smoothing Winters*

Metode eksponensial *smoothing* dari winter merupakan salah satu metode dari berbagai macam metode eksponensial *smoothing* untuk jenis data kuantitatif dan runtut waktu. Menurut Arsyad (2001), pengertian dari data runtut waktu adalah data yang dikumpulkan, dicatat dan diobservasi sepanjang waktu secara berurutan. Metode eksponensial *smoothing winters* menggunakan persamaan tambahan yang digunakan untuk mengestimasi adanya pengaruh faktor musim. Estimasi tersebut dinyatakan dalam suatu indeks musiman dan dihitung dengan persamaan pemulusan eksponensial.

Perumusan tersebut memperlihatkan bahwa estimasi indeks musiman (Y_t/A_t) dikalikan dengan σ . alasan mengapa Y_t dibagikan A_t adalah menyatakan nilainya sebagai suatu indeks, agar dapat dihitung rata-ratanya dengan indeks musiman yang dihaluskan sampai periode $t-1$. Keempat persamaan yang digunakan dalam model winter adalah sebagai berikut:

Eksponensial *Smoothing*

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

Estimasi *Trend*

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

Estimasi Musiman

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

Ramalan pada periode p di masa datang

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

Keterangan:

A_t = nilai *smoothing* yang baru

α = konstanta *smoothing* untuk data ($0 \leq \alpha \leq 1$)

Y_t = data yang baru atau yang sebenarnya pada periode t

β = konstanta *smoothing* untuk estimasi trend ($0 \leq \beta \leq 1$)

T_t = estimasi trend

μ = konstanta *smoothing* untuk estimasi musiman ($0 \leq \mu \leq 1$)

S_t = estimasi musiman

p = periode yang diramalkan

L = panjangnya musim

Y_{t-p} = ramalan pada periode p

Persamaan 4 memperbarui nilai-nilai pemulusan. Dalam persamaan tersebut Y_t dibagi dengan S_{t-L} , dan hal ini akan menghilangkan pengaruh musiman dalam data asli Y_t setelah estimasi musiman dan estimasi trend dimuluskan dalam persamaan 5 dan 6, peramalan dilakukan dengan persamaan 7. Untuk

meminimalkan MSE (*Mean Squared Error*), teknik winter lebih baik dari Brown dan Hold, sehingga teknik ini dapat dikatakan lebih baik dari kedua model tersebut.

Ekspensial *smoothing* adalah teknik yang sudah umum dipakai untuk peramalan jangka pendek, keuntungan utama penggunaan teknik ini adalah biaya yang rendah dan kemudahan pemakainya. Dasar dalam ekspensial *smoothing* adalah rata-rata tertimbang pengukuran-pengukuran pada masa lalu. Dasar pertimbangannya adalah bahwa rata-rata masa lalu mengandung informasi mengenai apa yang akan terjadi dimasa yang akan datang. Oleh karena data masa lalu mengandung fluktuasi *random* dan informasi mengenai pola variable, maka diperlukan usaha untuk memuluskan data-data ini. Pendekatan ini mengasumsikan bahwa fluktuasi-fluktuasi ekstrem menyatakan tingkat pengaruh *random* dalam rangkaian data.

1.4.2 Metode ARIMA (Autoregressive Integrated Moving Average)

Menurut Whitten (2007), ARIMA pertama kali dikembangkan oleh George Box dan Gwilym Jenkins untuk pemodelan analisis deret waktu. ARIMA sering juga dipanggil Box-Jenkins models. ARIMA mewakili tiga pemodelan yaitu dari *autoregressive* model (AR), *moving average*(MA), dan *autoregressive* dan *moving average* model (ARMA). Tahapan pelaksanaan dalam pencarian model yaitu:

1. Identifikasi model sementara dengan menggunakan data masa lalu untuk mendapatkan model dari ARIMA. Tahap identifikasi dilakukan dengan mengamati pola estimasi ACF (*Autocorellation Function*) dan PACF (*Partial*

Autocorellation Function) yang diperoleh dari data yang selanjutnya digunakan untuk mendapatkan dugaan model yang sesuai dengan pola data.

2. Penafsiran atau estimasi parameter dari model ARIMA dengan menggunakan data masa lalu.
3. Pengujian diagnostik untuk menguji kelayakan model. Bila model tidak layak maka lakukan langkah identifikasi, estimasi, pengujian diagnostik hingga mendapat model yang layak.
4. Penerapan, yaitu peramalan nilai data deret berkala yang akan datang menggunakan metode yang telah diuji.

1.4.3 Pemilihan Metode Peramalan Terbaik

Menurut Hanke dan Wichern (2005), metode eksponensial *smoothing winters* adalah metode yang populer digunakan karena tingkat keakuratan yang baik dan murah dalam pengaplikasiannya. Metode ini juga lebih mudah untuk dimengerti dan diterapkan ada suatu kasus jika dibandingkan dengan metode lain seperti metode ARIMA dari Box Jenkins. Selain itu, penggunaan metode ini juga dibutuhkan dalam bisnis dengan skala kecil hingga menengah atau dalam bisnis yang tidak mempunyai staf khusus di bidang statistika. Berikut perbandingan metode antara metode eksponensial *smoothing winters* dengan metode ARIMA dari Box Jenkins pada tabel 3.

Tabel 2.1 Perbandingan metode peramalan

Metode ARIMA	Metode Eksponensial <i>Smoothing Winters</i>
1. Data yang digunakan adalah data stasioner	1. Data yang digunakan dapat bersifat stasioner maupun tidak stasioner
2. Perlu uji keacakan data dengan melihat koefisien autokorelasi	2. Tidak perlu

3. Berdasar analisis pemilihan model <i>trend</i> musiman ARIMA	3. Berdasarkan analisis model regresi deret waktu sederhana
---	---

1.5 *Re-order Point* (Titik Pemesanan Kembali)

Menurut Sofyan (2004), titik atau tingkat pemesanan kembali atau *re-order point* adalah suatu titik atau batas dimana persediaan yang ada pada suatu saat dimana pemesanan harus diadakan kembali.

Persamaan matematis untuk menghitung ROP mengasumsikan permintaan selama waktu tunggu dan waktu tunggu itu sendiri adalah konstan. Besarnya penggunaan bahan selama bahan-bahan yang dipesan belum diterima, ditentukan oleh 3 faktor, yaitu:

1. *Lead time*
2. Tingkat penggunaan rata-rata
3. *Safety Stock*

Dalam menghitung *re-order point* menggunakan rumus sebagai berikut:

$$ROP = (D \times T) + SS \dots\dots\dots(8)$$

Keterangan:

ROP = Pemesanan kembali (*re-order point*)

SS = *Safety Stock*

D = Hasil perhitungan peramalan permintaan barang

T = *Lead time*

Dalam menentukan pemesanan kembali tersebut, ada empat sistem yang umumnya digunakan dengan beberapa variasi, yaitu sistem tinjauan terus-menerus, sistem tinjauan periodik, sistem jumlah tetap dan siste tepat waktu, secara singkat dapat dijelaskan sebagai berikut:

a. Sistem tinjauan terus menerus (*perpetual review system*)

Dalam sistem ini penunjaian dilakukan terus-menerus, yang berarti setiap kali perlu dipesan, amak harus dipesan. Perhitungan kapan perlu dipesan adalah apabila jumlah persediaan sudah mencapai jumlah atau tingkat tertentu. Jumlah tertentu ini disebut titik pemesanan kembali. Namun, pendekatan dengan menggunakan titik pemesanan kembali ini tidak hanya digunakan dalam sistem ini, tetapi juga digunakan dalam sistem dalam jumlah tetap.

b. Sistem tinjauan periodik (*periodic review system*)

Dalam sistem ini tinjauan atau perhitungan pemesanan kembali dilakukan setiap waktu tertentu, misalnya setiap 1 bulan, 3 bulan, 6 bulan atau setiap periode waktu tertentu yang ditetapkan. Penentuan ini didasarkan atas beberapa pertimbangan seperti jenis barang, kepentingan barang tersebut dalam perusahaan, dan sebagainya. Tidak peduli persediaan masih banyak atau tidak, setiap waktu tertentu harus dihitung kembali. Proses perhitungan pemesanan kembali ini tidak berarti berakibat harus memesan kembali, jadi ada tiga kemungkinan yaitu memesan kembali, tidak memesan lagi karena persediaan masih banyak atau membatalkan persediaan yang sedang berjalan karena persediaan kebanyakan.

c. Sistem jumlah tetap (*fixed quantity system*)

Dalam sistem ini yang menonjol adalah setiap kali memesan, jumlah yang dipesan selalu sama dan apabila harga satuannya sama, amak harga yang

dipesan juga sama. Mengenai kapan dipesan, tergantung frekuensi yang paling ekonomis.

d. Sistem tepat waktu (*just in time*)

Dalam sistem ini andalan diletakkan pada konsep tepat waktu, yang diberlakukan pada semua kegiatan yang berhubungan dengan produksi yaitu tepat waktu pemesanan, tepat waktu pembelian, tepat waktu kedatangan barang, tepat waktu produksi dan sebagainya.

1.6 *Economic Order Quantity* (EOQ)

Menurut Heizer & Render (2015), EOQ adalah sebuah teknik kontrol persediaan yang meminimalkan biaya total dari pemesanan dan penyimpanan serta berdasar pada beberapa asumsi:

- a. Jumlah permintaan diketahui, konstan, dan independen.
- b. Waktu tunggu yakni waktu antara pemesanan dan penerimaan pesanan diketahui dan konstan.
- c. Penerimaan persediaan bersifat instan dan selesai seluruhnya.
- d. Biaya variabel hanya biaya untuk menyiapkan atau melakukan pemesanan (biaya penyetelan) dan biaya menyimpan persediaan dalam waktu tertentu (biaya penyimpanan dan membawa).
- e. Kehabisan atau kekurangan persediaan dapat sepenuhnya dihindari jika pemesanan dilakukan pada waktu yang tepat.

Berikut rumus yang digunakan dalam perhitungan persediaan:

$$EOQ = \sqrt{\frac{2CR}{H}} \dots\dots\dots(9)$$

Keterangan:

EOQ = jumlah/nilai EOQ (unit).

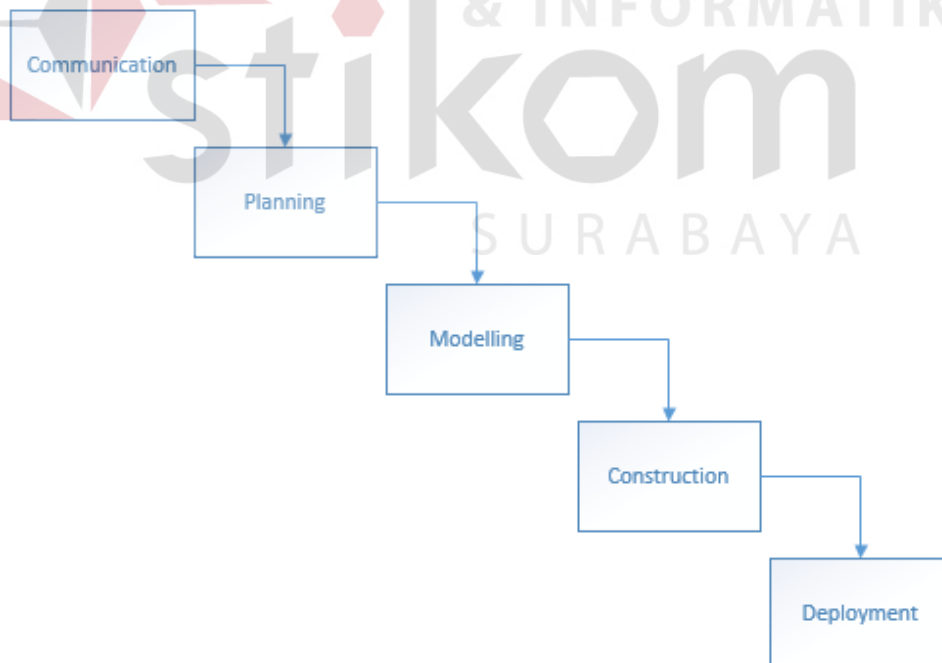
C = biaya pemesanan per pesanan.

R = permintaan per periode (unit).

H = biaya penyimpanan.

1.7 System Development Life Cycle (SDLC)

Menurut Pressman (2015), di dalam *software development life cycle* terdapat beberapa model diantaranya adalah model *waterfall*, terkadang disebut sebagai siklus hidup klasik, menunjukkan sistematis, pendekatan sekuensial untuk penyebaran perangkat lunak yang dimulai dengan proses *communication* kemudian berlangsung melalui *planning*, *modelling*, *construction*, dan *deployment* yang berakhir pada dukungan yang berkelanjutan dari terselesainya *software*.



Gambar 2.5 Tahapan *System Development Life Cycle* model *waterfall*

a. *Communication*

Langkah ini merupakan analisis terhadap kebutuhan *software* dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan *customer*, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel maupun dari internet.

b. *Planning*

Proses ini merupakan lanjutan dari proses *communication (analysis requirement)*. Tahapan ini menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software*, termasuk rencana yang akan dilakukan.

c. *Modelling*

Proses ini menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosuderal. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

d. *Construction*

Proses pembuatan kode. *Coding* atau pengkodean merupakan penerjemah desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi.

e. *Deployment*

Tahapan ini bisa dikatakan akhir dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

1.8 Black Box

Menurut Romeo (2003), *black box* testing merupakan *testing* yang dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black box testing* juga disebut sebagai *behavioral testing*, *spesification-based testing*, *input/output testing* atau *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*. Kategori *error* yang akan diketahui dengan menggunakan *black box testing* ini adalah sebagai berikut:

1. Fungsi yang hilang atau tidak benar.
2. *Error* dari *interface*.
3. *Error* dari struktur data atau akses *external database*.
4. *Error* dari kinerja atau tingkah laku sistem.
5. *Error* dari inisialisasi dan terminasi.

Tak seperti *white box testing*, yang dipakai pada awal proses *testing*. *Black box testing* digunakan pada tahap akhir dan berfokus pada domain informasi. Tes didisain untuk menjawab pertanyaan sebagai berikut:

1. Bagaimana validasi fungsi yang akan dites?
2. Bagaimana tingkah laku dan kinerja sistem dites?
3. Kategori masukan apa saja yang bagus digunakan untuk test cases?
4. Apakah sebagian sistem sensitif terhadap suatu nilai masukan tertentu?
5. Bagaimana batasan suatu kategori masukan ditetapkan?
6. Sistem mempunyai toleransi jenjang dan volume data apa saja?
7. Apa saja akibat dari kombinasi data tertentu yang akan terjadi pada operasi sistem?

Dengan menerapkan teknik *black box*, dapat dibuat sekumpulan *test cases* yang memuaskan kriteria-kriteria sebagai berikut [MYE79]:

- a. *Test case* yang mengurangi jumlah *test cases* (lebih dari satu) yang didisain untuk mencapai *testing* yang masuk akal.
- b. *Test cases* yang dapat memberikan informasi tentang kehadiran kelas-kelas dari *error*.

1.9 Skala Likert

Menurut Riduwan (2005), Angket atau kuisioner adalah daftar pertanyaan yang diberikan kepada orang lain yang bersedia memberikan respon sesuai dengan permintaan pengguna. Tujuan dari angket adalah mencari informasi dari responden tanpa khawatir bial responden memberikan jawaban yang tidak sesuai dengan kenyataan.

Perhitungan skor penilaian untuk setiap pertanyaan (QS) didapatkan dari jumlah pengguna (PM) dikalikan dengan skala nilai (N). Jumlah skor tertinggi (STot) didapatkan dari skala tertinggi (NT) dikalikan dengan jumlah pertanyaan (QTot) dikalikan total pengguna (Ptot). Sedangkan nilai presentase akhir (Pre) diperoleh dari jumlah skor hasil pengumpulan data (JSA) dibagi jumlah skor tertinggi (Stot) dikalikan 100%. Berikut rumus dari skala likert sebagai berikut:

$$QS(n) = PM \times N \dots\dots\dots(10)$$

$$STot = NT \times Qtot \times Ptot \dots\dots\dots(11)$$

$$Pre = JSA / STot \times 100\% \dots\dots\dots(12)$$

Keterangan:

QS(n) = Skor Pertanyaan ke-n

PM = Jumlah Pengguna

N = Skala Nilai

STot = Total Skor Tertinggi

NT = Skala Nilai Tertinggi

Qtot = Total Pertayaan

Ptot = Total Pengguna

Pre = Persentase Akhir (%)

JSA = Jumlah Skor Akhir

Menurut Husain (2008), ada dua bentuk pernyataan yang menggunakan skala likert yaitu bentuk pernyataan positif dan bentuk pernyataan negatif. Bentuk jawaban skala likert adalah sangat layak, layak, kurang layak, tidak layak dan sangat tidak layak. Skala pengukuran nilai persentase dapat dilihat pada tabel 2.1.

Tabel 2.2 Keterangan Persentase Nilai

Nilai	Keterangan
0%-20%	Sangat Layak
21%-40%	Tidak Layak
41%-60%	Kurang Layak
61%-80%	Layak
81%-100%	Sangat Layak