

## BAB 2

### LANDASAN TEORI

#### 2.1 Peramalan

Dalam suatu aktivitas produksi ada yang disebut dengan peramalan. Peramalan adalah prediksi mengenai apa yang akan terjadi di masa depan (Russel & Taylor III, 2011, p. 335). Permintaan terhadap suatu produk dapat diestimasi menggunakan teknik peramalan. Tujuan dari peramalan adalah untuk mengurangi ketidakpastian yang terjadi di masa depan dengan memberikan estimasi.

Menurut Render (2012, p.178) ada langkah-langkah untuk melakukan peramalan, diantaranya adalah:

1. Menentukan kegunaan peramalan atau tujuan dari peramalan.
2. Menentukan barang yang akan diramalkan.
3. Menentukan rentang waktu peramalan.
4. Memilih metode peramalan yang akan digunakan.
5. Mengumpulkan data yang dibutuhkan untuk peramalan.
6. Validasi model peramalan.
7. Melakukan peramalan.
8. Implementasi hasil dari peramalan.

Rentang waktu untuk melakukan peramalan dibagi menjadi 3 jenis yaitu jangka waktu pendek, jangka waktu sedang dan jangka waktu panjang. Jangka waktu pendek adalah peramalan untuk 1 sampai 30 hari, jangka waktu sedang adalah dari 1 bulan hingga 1 tahun sedangkan untuk jangka waktu panjang adalah peramalan yang dilakukan untuk lebih dari 1 tahun (Render, 2012, p.178). Jenis peramalan dibagi menjadi tiga kategori utama yaitu *time-series model*, model kausal dan model kualitatif. *Time series model* merupakan model peramalan dengan menggunakan data-data histori, beberapa metode yang termasuk dalam kategori ini adalah *Moving Average*, *Exponential Smoothing*, dan *Weighted Moving Average*. Model Kausal adalah peramalan dengan mempertimbangkan faktor-faktor yang akan mempengaruhi kuantitas dari produk atau barang yang diramalkan contohnya adalah pembelian minuman bersoda tergantung dari cuaca hari itu. Beberapa metode dari kategori ini adalah *Regression Analysis* dan *Multiple Regression*. Model terakhir yaitu model kualitatif adalah peramalan menggunakan keputusan atau pernyataan secara subjektif, yang termasuk dalam kategori ini adalah *Delphi Method*, *Sales Force Composite*, *Jury of Executive Opinion* dan *Consumer Market Survey* (Render, 2012, pp.178-179).

##### 2.1.1 Pola Data Permintaan

Setiap kali menentukan metode peramalan menggunakan peramalan *time series*, pola data perlu diketahui terlebih dahulu. Pola data dilakukan selama interval waktu tertentu. Dalam *time series* terdapat 4 jenis pola data permintaan, yaitu sebagai berikut (Baroto, 2002, pp. 31-35):

###### 1. Pola *Trend*

Pola ini terjadi apabila data permintaan menunjukkan pola kecenderungan gerakan penurunan atau kenaikan dalam jangka panjang. Data yang kelihatannya berfluktuasi apabila dilihat pada rentang waktu yang panjang akan dapat ditarik suatu garis maya. Bila data tersebut memiliki pola data

*trend*, maka metode yang sesuai digunakan adalah regresi linear, *exponential smoothing* atau *double exponential smoothing*.

2. Pola Musiman

Apabila data berfluktuasi, namun fluktuasi terlihat berulang pada suatu interval waktu tertentu, maka data tersebut berpola musiman. Data dipengaruhi oleh musim sehingga biasanya interval perulangan data ini adalah satu tahun. Contohnya adalah penjualan payung dan jas hujan di musim hujan lebih besar dibandingkan penjualannya di musim kemarau. Metode peramalan yang sesuai dengan pola musiman adalah metode *winter*, *moving average* atau *weighted moving average*.

3. Pola Siklikal

Pola ini terjadi apabila fluktuasi permintaan secara jangka panjang membentuk pola sinusoid atau gelombang atau siklus. Untuk menentukan data siklis tidak mudah karena rentang perulangan siklikal tidak menentu. Metode yang sesuai bila data berpola siklikal adalah metode *moving average*, *weighted moving average* dan *exponential smoothing*.

4. Pola Eratik

Pola Eratik atau *Random* terjadi apabila fluktuasi data permintaan dalam jangka panjang tidak dapat digambarkan oleh ketiga pola lainnya. Fluktuasi permintaan bersifat acak atau tidak jelas. Untuk pola ini, tidak ada metode yang direkomendasikan. Keterampilan dan analisis dari peramalan sangat menentukan keakuratan peramalan. Untuk pola ini bisa jadi dibutuhkan metode khusus.

## 2.1.2 Model Peramalan *Time Series*

Model peramalan *time series* adalah teknik statistik yang menggunakan data historikal dalam suatu periode waktu. Asumsi dari model ini adalah bahwa kejadian di masa lampau mempengaruhi kejadian di masa depan. Model ini merupakan model peramalan yang terpopuler yang paling sering digunakan (Russel & Taylor III, 2011, p. 341).

### 2.1.2.1 Metode *Moving Average*

Metode *Moving Average* merupakan metode peramalan yang menggunakan rata-rata permintaan dalam urutan periode waktu yang tetap. Kelebihan dari *Moving Average* adalah sifatnya stabil dan cocok untuk permintaan yang sifatnya stabil. Semakin panjang rentang waktu rata-rata maka hasilnya akan semakin stabil dan halus (Russel & Taylor III, 2011, p. 343). Berikut ini adalah persamaan yang digunakan dalam metode *Moving Average* (Render, 2012, p.185):

$$F_{t+1} = \frac{Y_t + Y_{t-1} + \dots + Y_{t-n+1}}{n}$$

Keterangan:

$F_{t+1}$  = Peramalan untuk bulan berikutnya

$Y_t$  = Permintaan aktual periode t

n = jumlah periode

### 2.1.2.2 Metode *Weighted Moving Average*

Metode *Weighted Moving Average* mirip dengan metode *Moving Average*. Perbedaan kedua metode ini adalah setiap data yang terdekat diberikan bobot yang terbesar dengan asumsi data dari waktu yang terdekat

memberikan kontribusi atau pengaruh yang lebih besar (Russel & Taylor III, 2011, p. 345). Berikut ini adalah persamaan yang digunakan dalam metode *Weighted Moving Average* (Render, 2012, p.185):

$$F_{t+1} = \frac{\sum(\text{weight in period } i)(\text{Actual value period } i)}{\sum(\text{weights})}$$

Keterangan:

$F_{t+1}$  = Peramalan untuk bulan berikutnya

### 2.1.2.3 Metode *Exponential Smoothing*

Metode *Exponential Smoothing* juga merupakan metode peramalan yang menghitung rata-rata dari data, data terbaru akan diberikan bobot lebih besar. Digunakan konstanta yang disebut *smoothing constant* yang merupakan faktor yang memberikan beban pada peramalan metode *Smoothing Constant* (Russel & Taylor III, 2011, p. 347). Berikut ini adalah persamaan yang digunakan dalam metode *Exponential Smoothing* (Render, 2012, p.188):

$$F_{t+1} = F_t + \alpha (Y_t - F_t)$$

Keterangan:

$F_{t+1}$  = Peramalan untuk bulan berikutnya

$Y_t$  = Permintaan aktual periode t

$F_t$  = Peramalan untuk periode t

$\alpha$  = *smoothing constant*

### 2.1.3 Perhitungan Akurasi Peramalan

Tidak ada metode peramalan yang akurasinya tepat dan sempurna, metode yang tepat untuk suatu data belum tentu tepat untuk pola data yang lain. Oleh karena itu perlu dilakukan perhitungan akurasi peramalan. Beberapa kriteria dari ketepatan ramalan yang sering digunakan untuk menghitung akurasi dari metode peramalan model *time series* diantaranya adalah *Mean Absolute Deviation* (MAD), *Mean Square of Error* (MSE), dan *Mean Absolute Percentage of Error* (MAPE) (Baroto, 2002, p.31).

#### 2.1.3.1 *Mean Absolute Deviation* (MAD)

*Mean Absolute Deviation* atau MAD adalah pengukuran untuk ketidaktepatan peramalan yang termudah dan yang paling sering digunakan. MAD adalah rata-rata antara peramalan dengan permintaan aktual. Semakin kecil nilai MAD maka semakin tinggi akurasi dari peramalan (Russel & Taylor III, 2011, p. 357). Persamaan yang digunakan untuk menghitung MAD adalah sebagai berikut (Render, 2012, p.182):

$$MAD = \frac{\sum|\text{forecast error}|}{n}$$

Keterangan:

MAD = *Mean Absolute Deviation*

*forecast error* = selisih antara hasil peramalan dengan permintaan aktual

n = jumlah kesalahan peramalan

#### 2.1.3.2 *Mean Square of Error* (MSE)

*Mean Square of Error* atau MSE adalah pengukuran untuk ketidaktepatan peramalan yang menghitung pangkat dari kesalahan peramalan (Russel &

Taylor III, 2011, p. 361). Persamaan yang digunakan untuk menghitung MSE adalah sebagai berikut (Render, 2012, p.183):

$$MSE = \frac{\sum (forecast\ error)^2}{n}$$

Keterangan:

MSE = *Mean Square of Error*

*forecast error* = selisih antara hasil peramalan dengan permintaan aktual

n = jumlah kesalahan peramalan

### 2.1.3.3 Mean Absolute Percentage of Error (MAPE)

*Mean Absolute Percentage of Error* atau MAPE adalah pengukuran untuk ketidaktepatan peramalan yang menghitung persentase kesalahan dari peramalan, lebih mudah dipahami karena dalam bentuk persentase (Baroto, 2002, p.49). Persamaan yang digunakan untuk menghitung MAPE adalah sebagai berikut (Render, 2012, p.183):

$$MAPE = \frac{\sum \left| \frac{forecast\ error}{actual} \right|}{n} \times 100\%$$

Keterangan:

MAPE = *Mean Absolute Percentage of Error*

*forecast error* = selisih antara hasil peramalan dengan permintaan aktual

*actual* = permintaan aktual

n = jumlah kesalahan peramalan

## 2.2 Persediaan

Pengendalian dan pemeliharaan persediaan merupakan permasalahan umum bagi organisasi dalam berbagai sektor ekonomi dan industri. Persediaan atau inventori adalah bahan mentah, barang dalam proses, barang jadi, bahan pembantu, bahan pelengkap, komponen yang disimpan dalam antisipasinya terhadap pemenuhan permintaan (Baroto, 2002, p.52).

Persediaan adalah komponen, bahan, atau produk jadi yang tersedia di tangan, menunggu untuk digunakan atau dijual. Istilah persediaan sendiri dapat digunakan dalam beberapa perbedaan seperti (Tersine, 1994, pp.2-3):

1. Persediaan bahan baku di tangan (*stock on hand*) pada waktu tertentu (aset yang dapat dilihat, diukur, dan dihitung).
2. Daftar persediaan secara fisik.
3. Jumlah *item* di tangan.
4. Nilai persediaan barang yang dimiliki oleh organisasi pada saat waktu tertentu (untuk perekaman keuangan dan akuntansi).

Untuk menjaga keseimbangan permintaan dengan penyediaan bahan baku dan waktu proses diperlukan persediaan, sehingga terdapat empat faktor yang dijadikan sebagai fungsi perlunya persediaan yaitu (Yamit, 2005, pp.5-6):

1. Faktor waktu, menyangkut lamanya proses produksi dan distribusi sebelum barang jadi sampai kepada konsumen. Persediaan dilakukan untuk memenuhi kebutuhan selama waktu tunggu (*lead time*).
2. Faktor ketidakpastian waktu, berasal dari *supplier* menyebabkan perusahaan memerlukan persediaan agar tidak menghambat proses produksi maupun keterlambatan pengiriman kepada konsumen.

3. Faktor ketidakpastian penggunaan, berasal dari dalam perusahaan yang disebabkan oleh kesalahan dalam peramalan permintaan, kerusakan mesin, keterlambatan operasi, bahan cacat, dan berbagai kondisi lainnya.
4. Faktor ekonomis adalah adanya keinginan perusahaan untuk mendapatkan alternatif biaya rendah dalam memproduksi atau membeli item dengan menentukan jumlah yang paling ekonomis.

### 2.3 Sistem Persediaan

Sistem persediaan adalah suatu mekanisme mengenai bagaimana mengelola masukan atau *input* yang sehubungan dengan persediaan menjadi keluaran atau *output* dimana untuk itu diperlukan umpan balik agar *output* memenuhi standar tertentu. Mekanisme sistem ini adalah pembuatan serangkaian kebijakan yang memonitor tingkat persediaan, menentukan persediaan yang harus dijaga, kapan persediaan harus diisi, dan berapa besar pesanan harus dilakukan. Sistem ini bertujuan menetapkan dan menjamin tersedianya produk jadi, barang dalam proses, komponen, dan bahan baku secara optimal, dalam kuantitas serta waktu yang optimal (Baroto, 2002, p. 53). Di dalam masalah persediaan ada dua tinjauan yang dibahas yaitu berapa kuantitas yang dipesan dan kapan dilakukan pemesanan untuk persediaan. Variabel keputusan dalam pengendalian persediaan tradisional dapat diklasifikasikan ke dalam variabel kuantitatif seperti berikut (Baroto, 2002, p. 54):

1. Berapa banyak jumlah barang yang akan dipesan atau dibuat.
2. Kapan pemesanan atau pembuatan harus dilakukan.
3. Berapa jumlah persediaan pengaman.
4. Bagaimana mengendalikan persediaan.

Secara luas, tujuan dari sistem persediaan adalah menemukan solusi optimal terhadap seluruh masalah yang terkait dengan persediaan. Untuk menemukan solusi optimal terdapat batasan yang disebut dengan biaya persediaan. Biaya persediaan adalah semua pengeluaran dan kerugian yang timbul akibat persediaan. Biaya tersebut adalah harga pembelian, biaya pemesanan, biaya penyiapan, biaya penyimpanan, dan biaya kekurangan persediaan (Baroto, 2002, p. 55).

#### 2.3.1 Reorder Point atau Titik Pemesanan

Berkaitan dengan kapan persediaan harus diisi, kita memerlukan pertimbangan keputusan untuk memperhitungkan lamanya waktu tunggu ataupun jangka waktu pengiriman. Waktu kapan memesan persediaan biasanya diekspresikan sebagai istilah ROP atau *Reorder Point* yang menunjukkan pada tingkat persediaan manakah pemesanan harus dilakukan (Render, 2012, p.205).

Persamaan mengasumsikan permintaan selama waktu tunggu ( $d$ ) dan *lead time* ( $L$ ) itu sendiri bersifat konstan. Adapun persamaan untuk menghitung ROP dan mendapatkan nilai dari permintaan selama waktu tunggu adalah:

$$ROP = d \times L$$

$$d = \frac{D}{\text{jumlah hari kerja dalam satu tahun}}$$

Keterangan:

ROP = *Reorder Point*

$d$  = *demand* atau permintaan per hari

- L = *lead time* atau waktu tunggu pemesanan (dalam hari)  
 D = Permintaan dalam satu tahun atau satu periode

Teori persediaan khususnya model *Reorder Point* memberikan kontribusi sebagai dasar untuk studi kasus yang diangkat pada PT. Bangkit Sukses Mandiri. *Reorder Point* membantu memberikan gambaran dalam menentukan kapan pihak perusahaan perlu melakukan pemesanan sebelum barang tersebut habis.

### 2.3.2 Safety Stock atau Persediaan Pengaman

Untuk menghindari kehabisan persediaan yang dapat terjadi akibat permintaan yang tidak konstan, diberikan rekomendasi berupa persediaan dalam jumlah tertentu yang disebut dengan *safety stock* (SS). Titik pemesanan kembali ditentukan menggunakan persamaan dengan *safety stock* sebagai berikut (Baroto, 2002, pp. 77-78):

$$ROP = d \times L + SS$$

Dimana,

$$SS = z \cdot \sigma$$

Keterangan:

SS = jumlah *safety stock*

z = nilai dari tabel kurva normal berdasarkan tingkat pelayanan

$\sigma$  = standar deviasi permintaan

Tabel 2.1 Nilai Z berdasarkan Tingkat Pelayanan

Tingkat Pelayanan (%)	Nilai Z
90	1.28
91	1.34
92	1.41
93	1.48
94	1.55
95	1.65
96	1.75
97	1.88
98	2.05
99	2.33
99,99	3.72

Sumber: (Render, 2012, p.245)

## 2.4 Permodelan Sistem dan Simulasi

Simulasi adalah tiruan atau imitasi operasi dari proses atau sistem yang terjadi di dunia nyata dalam waktu tertentu (Banks, Carson II, Nelson, & Nicol, 2010, p. 21). Simulasi dapat dilakukan baik secara manual maupun menggunakan komputer. Dalam simulasi, komputer digunakan untuk mengevaluasi model secara numerikal dan data dikumpulkan untuk mengestimasi karakteristik sistem yang sesungguhnya. (Law, 2007, p. 2) Adapun kegunaan dari simulasi adalah untuk memodelkan sistem dengan tujuan menjawab pertanyaan bagaimana-jika di dalam dunia nyata yang tidak dapat dieksekusi langsung melainkan harus dianalisis (Banks, Carson II, Nelson, & Nicol, 2010, p. 21).

Jenis model simulasi dibagi menjadi 3 dimensi diantaranya adalah (Law, 2007, pp. 4-6):

1. Statik dan Dinamik  
Simulasi statik adalah representasi sistem dimana waktu tidak memiliki peran, contohnya adalah sistem konveyor pada pabrik. Simulasi dinamik adalah kebalikan dari simulasi statik.
2. Deterministik dan Stokastik  
Simulasi yang tidak memiliki komponen probabilistik atau komponen acak (*random*) disebut dengan simulasi deterministik sehingga hasil dari sistem dapat didefinisikan berdasarkan *input*-nya. Sedangkan simulasi stokastik memiliki komponen probabilistik.
3. Kontinu dan Diskrit  
Simulasi kontinu adalah simulasi yang terus berubah sepanjang periode waktu, contohnya adalah aliran air pada waduk pembangkit listrik. Simulasi diskrit terjadi ketika status variabel berubah secara diskrit pada kurun waktu tertentu (Banks, Carson II, Nelson, & Nicol, 2010, p. 32).

*Discrete-event simulation* adalah permodelan sistem yang berubah dalam rentang waktu dengan representasi dimana status variabel berubah secara instan atau seketika (Law, 2007, p. 6). Konsep ini akan digunakan untuk memodelkan proses bisnis dalam penelitian. Adapun konsep yang digunakan dalam *discrete-event simulation* adalah sebagai berikut (Banks, Carson II, Nelson, & Nicol, 2010, p. 107):

1. *System*, sekumpulan dari *entities* yang berinteraksi bersama dalam suatu waktu untuk mencapai satu tujuan.
2. *Model*, representasi abstrak dari sistem, biasanya mencakup hubungan struktural, logikal dan matematikal yang mendeskripsikan sistem yaitu *state*, *entity*, *attribute*, *process*, *event*, *activity*, dan *delay*.
3. *System state*, sekumpulan variabel yang berisi seluruh informasi yang dibutuhkan untuk mendeskripsikan sistem pada satu waktu.
4. *Entity*, objek atau komponen apapun dalam sistem yang membutuhkan representasi eksplisit, contohnya adalah mesin, *customer*.
5. *Attributes*, sifat yang dimiliki *entity*, contohnya adalah prioritas dari *customer* yang menunggu.
6. *List*, sekumpulan *entity*, contohnya adalah semua *customer* yang menunggu dalam antrian.
7. *Event*, kejadian yang tidak lama yang mengubah status dari sistem, contohnya adalah kedatangan *customer*.
8. *Activity*, durasi lamanya waktu yang diketahui kapan waktu mulainya.
9. *Delay*, durasi lamanya waktu yang tidak terdefinisi berapa lama, tidak akan diketahui sampai satu proses selesai.
10. *Clock*, variabel yang merepresentasikan waktu simulasi.

Simulasi dapat dijalankan dengan menggunakan perangkat lunak atau *software*. Beberapa *software* yang dapat membantu untuk melakukan simulasi diantaranya adalah AnyLogic, Arena, AutoMod, Enterprise Dynamics, ExtendSim, Flexsim, ProModel, SIMUL8 dan lain sebagainya (Banks, Carson II, Nelson, & Nicol, 2010, pp. 164-170). Masing-masing *software* memiliki tingkat dukungan untuk analisis statistik yang berbeda. Hasil dari simulasi juga bukan sebuah jawaban atau solusi, hasil dari simulasi perlu dievaluasi dan diamati lebih lanjut untuk diambil keputusan terkait dengan simulasi tersebut.

### 2.4.1 Distribusi Data

Sebelum memodelkan sistem dalam *software*, harus diamati terlebih dahulu distribusi dari data waktu sistem. Distribusi data sendiri terbagi menjadi *continuous* dan *discrete*. *Continuous Random Variates* diantaranya adalah *Uniform*, *Exponential*, *Gamma*, *m-Erlang*, *Weibull*, *Normal*, *Lognormal*, *Beta*, *Triangular* dan lain sebagainya (Law, 2007, pp. 447-459). *Discrete Random Variates* diantaranya adalah *Bernoulli*, *Discrete Uniform*, *Binomial*, *Geometric*, *Poisson* dan lain sebagainya (Law, 2007, pp. 459-466).

Beberapa penjelasan untuk distribusi dan karakteristik dari datanya adalah sebagai berikut (Kelton, Sadowski, & Sturrock, 2007, pp. 589-602):

1. *Exponential*

Parameter dari distribusi ini adalah rata-rata atau *mean* ( $\beta$ ). Distribusi ini biasanya digunakan untuk memodelkan waktu kedatangan, namun kurang cocok digunakan untuk memodelkan waktu *delay*.

2. *Normal*

Parameter untuk distribusi ini adalah rata-rata atau *mean* ( $\mu$ ) dan standar deviasi ( $\sigma$ ). Distribusi digunakan untuk situasi dimana teorema batas sentral berlaku.

3. *Uniform*

Parameter untuk distribusi ini adalah nilai minimum (a) dan nilai maksimum (b). Distribusi ini digunakan apabila nilai berada dalam rentang yang hampir sama.

### 2.4.2 Arena

Adapun *software* yang akan digunakan dalam penelitian adalah Arena. Arena adalah produk dari Rockwell Automation yang bertujuan untuk mensimulasikan sistem baik yang kontinu maupun diskrit (Banks, Carson II, Nelson, & Nicol, 2010, p. 165). Arena memiliki arsitektur yang terbuka dan berhubungan langsung dengan Visual Basic Application atau VBA. Beberapa keandalan Arena diantaranya adalah:

1. Dapat bekerja dalam platform apapun.
2. Hasil simulasi dapat dilihat langsung.
3. Tidak memerlukan *interpreter* khusus.
4. Memberikan *framework* model secara terintegrasi yang meliputi animasi, analisis data *input*, verifikasi model dan analisis data *output*.

Modul-modul akan digunakan dalam penelitian ketika menggunakan *software* Arena untuk mensimulasikan sistem informasi yang telah diterapkan. Adapun beberapa modul *flowchart* umum yang perlu diketahui adalah sebagai berikut:

1. Modul *Create* : Modul ini digunakan pada awal mula berangkatnya suatu *entity* pada modul simulasi.
2. Modul *Dispose* : Modul ini merupakan titik akhir dari *entities* pada model simulasi.
3. Modul *Process* : Modul ini merupakan metode proses utama dalam simulasi. Pilihan menggunakan *resource* tersedia dalam modul ini. Didalamnya terdapat pilihan untuk menggunakan *submodel* dan menspesifikasikan logika hierarki.



4. Modul *Decide* : Modul ini menunjukkan proses pengambilan keputusan dalam sebuah sistem jika terdapat kondisi tertentu.
5. Modul *Batch*, Modul *Separate*, Modul *Assign* dan Modul *Record*.

## 2.5 Sistem Informasi

### 2.5.1 Pengertian Sistem

Menurut O'Brien (2008, p.24), sistem didefinisikan sebagai kumpulan komponen-komponen yang saling berhubungan, dengan definisi batasan (*boundary*) yang jelas dan saling bekerja sama untuk mencapai kesamaan tujuan dengan menerima *input* dan menghasilkan *output* dalam proses transformasi yang terorganisir. Sistem memiliki tiga komponen atau fungsi dasar yang berinteraksi, yaitu:

1. *Input*, melibatkan penangkapan dan perakitan elemen yang masuk ke dalam sistem untuk diproses.
2. *Process*, melibatkan proses perubahan yang mengubah *input* menjadi *output*.
3. *Output*, melibatkan perpindahan elemen yang telah dihasilkan oleh proses ke tujuan akhir.

### 2.5.2 Pengertian Informasi

Menurut Romney dan Steinbart (2006, p.5), informasi didefinisikan sebagai data yang telah diorganisir dan diproses untuk memberikan pengertian kepada penggunanya. Informasi digunakan untuk membuat keputusan atau untuk meningkatkan proses pembuatan keputusan. Dalam fungsi secara umum, pengguna akan membuat keputusan yang lebih baik dengan bertambahnya kuantitas dan kualitas informasi. Sedangkan menurut McLeod (2007, p.8), informasi dibedakan dari data. Informasi didefinisikan sebagai data yang telah diproses dan telah memiliki arti. Informasi akan memberikan apa yang belum diketahui terhadap penggunanya.

### 2.5.3 Pengertian Sistem Informasi

Menurut Satzinger (2005, pp.6-7), sistem informasi adalah kumpulan komponen-komponen yang saling berkaitan yang berfungsi untuk satu tujuan yaitu mengumpulkan, memproses, menyimpan, dan menyediakan informasi yang dibutuhkan untuk menyelesaikan tugas-tugas bisnis. Sedangkan menurut Whitten (2004, p.12), sistem informasi merupakan pengaturan susunan dari orang, data, proses-proses, dan teknologi informasi yang saling berinteraksi untuk memproses, mengoleksi, menyimpan, dan menyediakan sebagai *output* informasi yang dibutuhkan untuk mendukung kegiatan suatu organisasi. Sistem informasi digunakan oleh organisasi untuk menangkap dan mengatur data untuk menghasilkan informasi yang mendukung pekerja, pelanggan, *supplier*, dan rekan kerja. Sistem informasi dianggap penting sebagai kemampuan untuk memperoleh keunggulan kompetitif dan disadari bagi organisasi bahwa karyawan perlu berpartisipasi dalam proses pengembangan sistem informasi.

## 2.6 Perancangan Sistem Informasi

Untuk melakukan perancangan sistem informasi diperlukan perencanaan yang matang. Perencanaan tersebut ditinjau dari dua sudut, diantaranya adalah (Satzinger, 2005, pp.16 – 17):

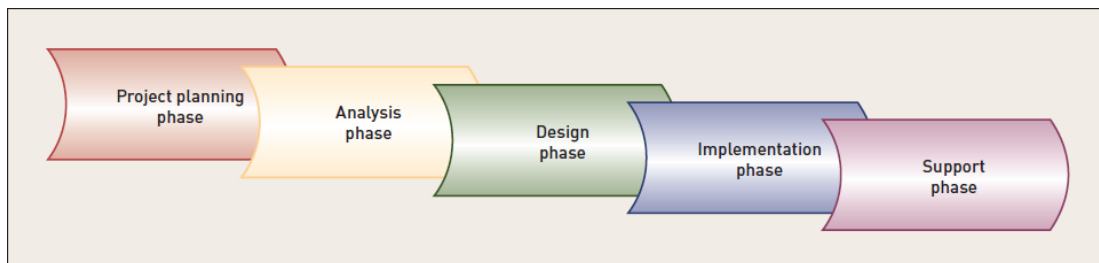
1. Perencanaan Arsitektur Aplikasi, yaitu deskripsi dari sistem informasi yang menjalankan fungsi bisnis sesuai dengan kebutuhan perusahaan.

2. Perencanaan Arsitektur Teknologi. yaitu deskripsi dari perangkat keras, perangkat lunak dan perangkat komunikasi yang dibutuhkan dalam perancangan sistem informasi.

Untuk merancang sistem informasi digunakan pendekatan yang disebut *System Development Life Cycle* (Satzinger, 2005, p.38).

## 2.7 *System Development Life Cycle* (SDLC)

*System Development Life Cycle* adalah seluruh proses membangun, merancang, menggunakan dan mengembangkan sistem informasi. Selain SDLC ada banyak pendekatan lainnya yang sering digunakan untuk membangun sistem informasi. Pemilihan pendekatan ini tergantung dari kebutuhan dan kesesuaian dari kebutuhan perusahaan (Satzinger, 2005, p.39). SDLC sendiri terbagi menjadi beberapa fase utama. Fase tersebut yaitu fase perencanaan proyek, fase analisis, fase perancangan, fase implementasi dan yang terakhir adalah fase perawatan (Satzinger, 2005, p. 40).



Sumber: (Satzinger, 2005, p.40)

Gambar 2.1 Fase dalam *System Development Life Cycle*

Fase pertama yaitu *project planning activities* atau fase perencanaan proyek, kegiatan yang dilakukan dalam fase ini adalah merencanakan, mengatur, dan menjadwalkan proyek. Kemudian fase berikutnya yaitu *analysis activities* atau fase analisis. Pada fase ini kegiatan yang dilakukan berfokus kepada pengertian secara menyeluruh mengenai permasalahan bisnis yang harus diselesaikan dan menentukan kebutuhan bisnis. Pada fase berikutnya yaitu *design activities* atau fase perancangan, dilakuan pembangunan sistem yang baru dengan menggunakan alat baik secara grafis maupun tulisan mengenai kebutuhan kemudian diterjemahkan ke dalam struktur program dan algoritma. Tahapan keempat yaitu *implementation activities* atau fase implementasi merupakan tahapan penting dalam membangun sebuah sistem yaitu menerapkan sistem ke dalam bentuk secara nyata dan digunakan. Setelah keempat fase awal dijalankan secara benar, maka dapat dilakukan fase terakhir yaitu *support phase* atau fase perawatan. *Support phase* merupakan fase dimana kegiatan berfokus pada melakukan pembaruan dan memelihara sistem setelah dijalankan (Satzinger, 2005, p. 40).

## 2.8 Diagram-Diagram UML

Salah satu metode pengembangan sistem yang menyediakan panduan untuk menyelesaikan setiap aktivitas secara spesifik yang paling awal muncul adalah *Unified Process*. Fase di dalam *Unified Process* terdiri dari *inception*, *elaboration*, *construction*, serta *transition* (Satzinger, Jackson, & Burd, 2005, pp. 45-47). Salah satu konsep dalam melakukan analisis dan perancangan untuk sistem informasi adalah dengan membuat diagram-diagram UML. Diagram

UML merupakan penerapan studi dari OOA (*Object Oriented Analysis*). Menurut Whitten (2004, p.430) diagram UML (*Unified Modeling Language*) merupakan kumpulan konvensi permodelan yang digunakan untuk menjabarkan sistem perangkat lunak dalam konteks objek.

Beberapa diagram-diagram yang akan digunakan untuk menganalisis kebutuhan sistem dengan menggunakan UML adalah sebagai berikut:

#### 1. *Activity Diagram*

*Activity diagram* merupakan jenis *workflow diagram* yang menggambarkan aktivitas pengguna di dalam sistem secara berurutan (Satzinger, 2005, p.142). Sedangkan menurut Whitten (2004, p.450) *activity diagram* merupakan diagram yang digunakan untuk menggambarkan secara grafis aliran proses bisnis, langkah dari *use case*, atau logika dari *method* sebuah objek.

Dalam *activity diagram*, ada beberapa simbol dasar yang digunakan dalam penggambarannya. Tanda panah menunjukkan urutan antara aktivitas. Bulatan hitam digunakan untuk mendenotasikan permulaan dan akhir dari aliran aktivitas. Simbol *diamond* merupakan tanda pemilihan aliran proses yang akan dilalui dari satu aliran atau aliran yang lain. Tanda garis hitam tebal merupakan pemisah antara kegiatan *concurrent* (kegiatan yang terjadi bersamaan) atau menggabungkan kembali kegiatan-kegiatan tersebut.

#### 2. *Event Table*

*Event table* merupakan katalog dari *use case* yang menjelaskan kejadian-kejadian dalam bentuk tabel informasi. *Event table* mencakupi kolom dan baris yang menjabarkan perincian setiap kejadian. Setiap baris pada *event table* mencatat informasi mengenai suatu kejadian dan *use case* nya, sedangkan setiap kolom merepresentasikan bagian kunci dari informasi mengenai kejadian dan *use case* tersebut. (Satzinger, 2005, p.174-175). Parameter kunci yang digunakan dalam *event table* adalah:

1. *Event*, mencatat kejadian yang menyebabkan sistem melakukan aktivitas.
2. *Trigger*, menjelaskan bagaimana sistem mengetahui *event* terjadi.
3. *Source*, menjelaskan agen eksternal yang menyebabkan data masuk ke sistem.
4. *Use case*, menjelaskan apa yang sistem lakukan ketika *event* terjadi.
5. *Response*, menjelaskan hasil keluaran yang diberikan sistem terhadap *event*.
6. *Destination*, menjelaskan agen eksternal manakah yang memperoleh hasil keluaran yang diberikan oleh sistem.

#### 3. *Use Case Diagram*

*Use case diagram* digunakan untuk menunjukkan interaksi pengguna (*actors*) dengan suatu sistem. *Actor* merupakan pengguna dari suatu sistem yang secara langsung berinteraksi dengan sistem itu sendiri (Satzinger, 2005, p.244). Dalam *use case diagram* digambarkan *actor* berupa figur sederhana menyerupai orang, lalu garis yang menghubungkan pesan dan *actor*. Digambarkan pula bulatan yang disebut *automation boundary*. *Automation boundary* mendenotasikan batasan antara lingkungan, dimana aktor berada, serta komponen internal dari sistem komputer.

#### 4. *Use Case Description*

Menggunakan *use case diagram* membantu identifikasi berbagai proses yang dilakukan oleh pengguna dan sistem baru perlu dukung. Dalam proses perancangan perlu dilakukan tingkat deskripsi yang lebih rinci dan lebih banyak mengenai proses. Untuk membuat perincian yang komprehensif, kita harus memahami secara menyeluruh tahapan untuk menyelesaikan proses bisnis. Dengan bantuan *use case description*, dapat dirincikan urutan langkah dalam setiap *use case* (Satzinger, 2005, p.220). *Use case description* menuliskan *main flow* dan *exception condition*. *Main flow* merupakan langkah yang dilakukan dalam pengerjaan suatu *use case* dan *exception condition* menerangkan kondisi-kondisi dimana *use case* tidak berjalan sesuai langkah yang seharusnya.

#### 5. *Class Diagram*

*Class diagram* merupakan deskripsi grafis dari struktur objek sistem statis, yang menunjukkan objek kelas yang membangun sebuah sistem beserta hubungan satu sama lain (Whitten, 2004, p.455). *Class diagram* digunakan untuk menunjukkan kelas dari objek tertentu di dalam suatu sistem. Menurut pendapat Satzinger (2005, p.185), *class diagram* memiliki tiga bagian penting, yaitu sebagai berikut:

1. *Class Name*, yang merupakan nama dari suatu *class*.
2. *Attribute*, yang merupakan atribut-atribut yang dimiliki oleh suatu *class*.
3. *Method*, yang menjelaskan apa saja yang bisa dilakukan oleh objek-objek di dalam suatu *class*.

#### 6. *System Sequence Diagram*

*System Sequence Diagram* (SSD) merupakan diagram yang menunjukkan urutan pesan antara aktor eksternal dan sistem pada saat skenario atau *use case* tertentu (Satzinger, 2005, p.213). SSD digunakan untuk mendefinisikan hasil masukan dan hasil keluaran serta urutan yang sistematis dari hasil masukan dan keluaran tersebut. SSD digunakan dalam konjungsi dengan deskripsi rinci atau dengan *activity diagram* untuk menunjukkan langkah pemrosesan dan interaksi antara sistem dan aktor.

#### 7. *Statechart Diagram*

Menurut Whitten (2004, pp.700-701) *statechart diagram* merupakan diagram yang menggambarkan kombinasi status asumsi objek pada masa waktu aktifnya, kejadian yang memicu transisi antar status, dan aturan yang memperbolehkan transisi antar objek tersebut.

Dalam sebuah *statechart diagram*, ada istilah-istilah yang digunakan yaitu:

1. *Pseudocode*, merupakan titik awal dari *statechart* dan diindikasikan dengan bulatan hitam.
2. *State*, merupakan kondisi pada masa objek ketika melakukan kegiatan tertentu.
3. *Transition*, merupakan pergerakan status objek dari satu ke yang lain.
4. *Guard*, merupakan kondisi yang memungkinkan transisi terjadi, berupa kondisi benar atau salah.
5. *Message event*, merupakan pemicu transisi yang menyebabkan objek meninggalkan status yang lama.

Beberapa diagram-diagram yang akan digunakan untuk rancangan sistem dengan menggunakan UML adalah sebagai berikut:

#### 1. *First-Cut Design Class Diagram*

*First-cut design class diagram* digunakan untuk membuat proses desain. Diagram dibuat dengan memperluas *class diagram* yang telah dibuat sebelumnya. Langkah awal adalah dengan melakukan elaborasi atribut dengan tipe dan nilai dari informasi. Lalu terakhir, ditambahkan navigasi panah navigasi (Satzinger, 2005, p.309).

## 2. *Sequence Diagram*

*Sequence diagram* digunakan untuk menjelaskan interaksi beberapa objek pada waktu tertentu secara berurutan (Satzinger, 2005, p.252). Sedangkan menurut Whitten (2004, p.702) *sequence diagram* merupakan diagram UML yang memodelkan logika *use case* dengan menggambarkan kembali interaksi pesan antara objek-objek dalam urutan rangkaian waktu.

Dalam pengembangan lebih lanjut pada proses desain, dibuat pengembangan *sequence diagram* menjadi tiga macam, antara lain adalah *first-cut sequence diagram*, *view layer sequence diagram*, dan *data access sequence diagram* (Satzinger, 2005, pp.316-324).

## 3. *Updated Design Class Diagram*

Dalam membuat *updated design class diagram*, *class diagram* yang sebelumnya telah dibuat ditambahkan format untuk menampilkan tipe data bagi masing-masing atribut serta *method* yang terjadi dalam setiap *class*.

## 4. *Package Diagram*

*Package diagram* merupakan diagram tingkat tinggi yang memungkinkan perancang mendesain asosiasi kelas dari kelompok yang berkaitan. Dalam *package diagram*, dibagi kelompok menjadi tiga bagian, yaitu *view layer*, *domain layer*, dan *data access layer*. Setiap kelas akan diletakkan kepada kelompok yang berkaitan sesuai yang telah dirancang sebelumnya. Dan kemudian ditunjukkan pula *dependency relationship* yang merupakan hubungan antar elemen yang mempengaruhi elemen lain, sehingga perancang dapat lebih mudah menganalisis perubahan (Satzinger, 2005, pp.341-342).

## 5. *Communication Diagram*

*Communication diagram* berguna untuk menampilkan pandangan yang berbeda dari *use-case*. Untuk *actor*, objek, dan pesan, *communication diagram* menggunakan simbol yang sama seperti yang digunakan pada *sequence diagram*, namun simbol untuk *lifeline* dan aktivasi tidak digunakan. Simbol yang digunakan adalah berbeda, disebut sebagai *link* (Satzinger, 2005, pp.334-335).

## 2.9 Database

Perancangan sistem informasi juga mencakup tentang *database* yang berfungsi untuk menyimpan data. *Database* ini dapat dimodifikasi sesuai dengan kebutuhan pengguna. Menurut Satzinger (2005, p.492), *Database Management System* adalah perangkat lunak sistem yang mengelola dan mengontrol akses ke *database* sedangkan, *Relational Database Management System* adalah sebuah sistem manajemen *database* yang menyimpan data ke dalam tabel.

Langkah-langkah yang dilakukan dalam membuat *Relational Database*:

1. Membuat sebuah tabel untuk setiap jenis entitas.
2. Memilih sebuah *primary key* untuk setiap tabel.
3. Menambah *foreign key* untuk setiap tabel untuk merepresentasikan relasi *one to many*.

4. Membuat tabel baru yang merepresentasikan relasi *many to many*.
5. Menentukan batasan integritas referensial.
6. Mengevaluasi kualitas skema dan melakukan perubahan yang diperlukan.
7. Memilih tipe data yang sesuai dan pembatasan nilai (jika perlu) untuk setiap bidang.

## 2.10 Keamanan Informasi

Dalam perancangan sistem informasi, ada suatu isu yang harus diperhatikan yaitu mengenai keamanan dari sistem informasi. Keamanan didefinisikan sebagai kualitas atau keadaan aman—bebas dari bahaya (Whitman & Mattord, 2010, p. 3). Keamanan informasi adalah proteksi terhadap informasi dan karakteristik yang kritical termasuk sistem dan perangkat lunak yang digunakan dan disimpan (Whitman & Mattord, 2010, p. 4).

Beberapa area dari keamanan untuk sistem informasi adalah (Whitman & Mattord, 2010):

1. *Physical Security*, yaitu strategi keamanan untuk menjaga aset fisik dan tempat kerja dari berbagai ancaman seperti api, akses yang tidak sah dan bencana alam.
2. *Operations Security*, berfokus untuk menjaga kemampuan organisasi untuk melakukan aktivitas operasional tanpa adanya interupsi atau bahaya.
3. *Communication Security*, memberikan proteksi terhadap media komunikasi organisasi, teknologi, dan isi, serta kemampuan untuk menggunakan peralatan agar mencapai tujuan organisasi.
4. *Network Security*, memberikan proteksi terhadap data organisasi dari jaringan. Pada alat, koneksi dan konten dari jaringan.

Karakteristik utama dari keamanan informasi yang paling diinginkan harus dimengerti agar karakteristik tersebut dapat dicapai. Karakteristik tersebut diantara lain adalah (Whitman & Mattord, 2010, pp. 6-8):

1. *Confidentiality*, adalah karakteristik informasi dimana hanya pihak yang memiliki hak special yang mampu mengakses informasi tertentu. Untuk melindungi sifat kerahasiaan ada beberapa mekanisme yang bisa dilakukan seperti klasifikasi informasi, kriptografi, penyimpanan dokumen yang aman dan lain sebagainya.
2. *Integrity*, adalah kualitas data secara keseluruhan utuh, lengkap dan tidak rusak. Integritas dari informasi dapat terancam pada saat dimasukkan, disimpan atau dipindahkan. Virus dan *Worms* juga dapat membahayakan integritas informasi.
3. *Availability*, adalah karakteristik dari informasi yang memungkinkan akses pengguna pada informasi tanpa gangguan.
4. *Privacy*, informasi yang disimpan dalam sistem terkadang bersifat pribadi, artinya informasi tersebut diberikan untuk digunakan sesuai dengan tujuan dari pemilik informasi memberikannya.
5. *Identification*, karakteristik dimana sistem dapat mengenali pengguna individual. Identifikasi adalah tahap awal mendapatkan akses pada material yang diamankan. Contohnya adalah penggunaan *user login*.
6. *Authentication*, adalah mekanisme control yang membuktikan bahwa pengguna tersebut memiliki identitas yang dia nyatakan. Contohnya adalah penggunaan *password* ketika pengguna melakukan *user login*.

7. *Authorization*, pemberian hak akses, artinya setiap pengguna belum tentu memiliki hak akses yang sama.
8. *Accountability*, adalah sifat informasi yaitu ketika ada kontrol mengenai siapa yang menggunakan sistem pada saat tersebut.