

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Aplikasi**

Menurut Pramana (2006) aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, *game*, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia. Menurut Yuhefizar (2012) aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Jadi aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak dengan komputer untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan dan penambahan data yang dibutuhkan. Aplikasi yang dirancang untuk melakukan tugas-tugas tertentu dapat diklasifikasikan menjadi dua, yaitu:

1. Aplikasi spesialis adalah aplikasi dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket adalah aplikasi dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

#### **2.2 Persediaan**

Menurut Ristono (2009) persediaan dapat diartikan sebagai barang-barang yang disimpan untuk digunakan atau dijual pada masa atau periode yang akan datang. Persediaan terdiri dari persediaan bahan baku, persediaan barang setengah jadi, dan persediaan barang jadi. Menurut Alexandri (2009) persediaan adalah suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam suatu periode usaha tertentu atau persediaan barang-barang yang masih

dalam pengerjaan atau proses produksi ataupun persediaan bahan baku yang menunggu penggunaannya dalam suatu proses produksi.

Persediaan barang mempunyai fungsi yang sangat penting bagi perusahaan. Dari berbagai macam persediaan barang yang ada, seperti bahan baku, barang dalam proses, dan barang jadi, perusahaan melakukan penyimpanan atas persediaan barang karena berbagai fungsi, yaitu fungsi yang memungkinkan perusahaan dapat memenuhi permintaan pelanggan, fungsi untuk mempertimbangkan penghematan–penghematan, dan fungsi untuk mengurangi adanya risiko ketidakpastian. Menurut Sumayang (2003) terdapat tiga fungsi lain mengapa persediaan barang diperlukan adalah untuk:

1. Menghilangkan pengaruh ketidakpastian.

Untuk menghadapi ketidakpastian maka pada sistem *inventory* ditetapkan persediaan darurat yang dinamakan *safety stock*. Jika sumber ketidakpastian ini dapat dihilangkan, maka jumlah *inventory* maupun *safety stock* dapat dikurangi.

2. Memberikan waktu luang untuk pengelolaan produksi dan pembelian.

Kadang-kadang lebih ekonomis memproduksi barang dalam proses atau barang jadi dalam jumlah besar atau dalam jumlah paket yang kemudian disimpan sebagai persediaan. Selama persediaan masih ada maka proses produksi dihentikan dan akan mulai kembali bila diketahui persediaan hampir habis. Pertimbangan ini memberikan kemudahan sebagai berikut:

- a. Memberikan keuntungan untuk menyebarkan dan meratakan beban biaya investasi pada sejumlah besar produk.

b. Memungkinkan penggunaan satu peralatan untuk menghasilkan bermacam-macam jenis produk.

3. Mengantisipasi pada *demand* dan *supply*.

*Inventory* disiapkan untuk menghadapi beberapa kondisi yang menunjukkan perubahan *demand* dan *supply*, yaitu:

a. Bila ada perkiraan perubahan harga dan persediaan bahan baku.

b. Sebagai persiapan menghadapi promosi pasar dimana sejumlah besar barang jadi disimpan menunggu penjualan tersebut.

c. Perusahaan yang melakukan produksi dengan jumlah *output* tetap akan mengalami perubahan produk pada kondisi permintaan yang rendah atau pada kondisi musim lesu atau *low season*. Kelebihan produk ini akan disimpan sebagai persediaan yang akan digunakan nanti apabila *output* tidak dapat memenuhi lonjakan permintaan pada musim ramai atau *peak season*.

### 2.3 Pembelian

Menurut Soemarso S.R. (2009) pembelian adalah akun yang digunakan untuk mencatat semua pembelian barang dagangan dalam suatu periode. Proses pembelian merupakan suatu struktur interaksi antara orang-orang, peralatan, metode-metode, dan pengendalian yang dirancang untuk mencapai fungsi-fungsi utama sebagai berikut:

a. Menangani rutinitas pekerjaan yang berulang-ulang dari bagian pembelian dan penerimaan.

b. Mendukung pengambilan keputusan dari orang-orang yang mengatur bagian pembelian dan penerimaan.

c. Membantu dalam penyajian laporan internal dan laporan eksternal.

Dapat disimpulkan bahwa pembelian adalah suatu kegiatan yang dilakukan oleh industri terutama produksi untuk memperoleh bahan baku, perlengkapan atau peralatan. Tujuan utamanya adalah memperoleh bahan dengan biaya serendah mungkin yang konsisten dengan kualitas yang sesuai standar yang ditentukan. Fungsi dari pembelian untuk memastikan bahwa ada keseimbangan antara persediaan bahan dengan tingkat inventaris sehingga perusahaan dapat mempertahankan posisi labanya sepanjang menyangkut biaya bahan baku dan agar dapat terus beroperasi.

#### **2.4 Bahan Baku**

Menurut Mulyadi (2005) bahan baku adalah bahan yang membentuk bagian menyeluruh. Menurut Kholmi (2003) bahan baku merupakan bahan yang membentuk bagian besar produk jadi, bahan baku yang diolah dalam perusahaan manufaktur dapat diperoleh dari pembelian lokal, impor atau hasil pengolahan sendiri. Bahan baku meliputi semua barang dan bahan yang dimiliki perusahaan dan digunakan untuk proses produksi. Menurut Kholmi (2003) bahan baku memiliki beberapa faktor yang perlu diperhatikan, yaitu:

1. **Perkiraan pemakaian**

Merupakan perkiraan tentang jumlah bahan baku yang akan digunakan oleh perusahaan untuk proses produksi pada periode yang akan datang.

2. **Harga bahan baku**

Merupakan dasar penyusunan perhitungan dari perusahaan yang harus disediakan untuk investasi dalam bahan baku tersebut.

3. **Biaya-biaya persediaan**

Merupakan biaya-biaya yang dibutuhkan oleh perusahaan untuk pengadaan bahan baku.

4. Kebijakan pembelanjaan

Merupakan faktor penentu dalam menentukan berapa besar persediaan bahan baku yang akan mendapatkan dana dari perusahaan.

5. Pemakaian sesungguhnya

Merupakan pemakaian bahan baku yang sesungguhnya dari periode lalu dan merupakan salah satu faktor yang perlu diperhatikan.

6. Waktu tunggu

Merupakan tenggang waktu yang tepat maka perusahaan dapat membeli bahan baku pada saat yang tepat pula, sehingga resiko penumpukan ataupun kekurangan persediaan dapat ditekan seminimal mungkin.

## 2.5 Pengendalian Persediaan Bahan Baku

Suatu pengendalian persediaan yang dijalankan oleh suatu perusahaan sudah tentu memiliki tujuan-tujuan tertentu. Pengendalian persediaan yang dijalankan adalah untuk menjaga tingkat persediaan pada tingkat yang optimal sehingga diperoleh penghematan-penghematan untuk persediaan tersebut. Menurut Handoko (2000) pengendalian adalah fungsi manajerial yang sangat penting karena persediaan fisik banyak perusahaan melibatkan investasi terbesar dalam persediaan aktiva lancar. Menurut Assauri (2004) pengendalian persediaan adalah salah satu kegiatan dari urutan kegiatan-kegiatan yang bertautan erat satu sama lain dalam seluruh operasi produksi perusahaan tersebut sesuai dengan apa yang telah direncanakan lebih dahulu baik waktu, jumlah kualitas maupun biayanya. Untuk

mengendalikan persediaan maka harus memenuhi persyaratan-persyaratan sebagai berikut:

- a. Terdapat gudang yang cukup luas dan teratur dengan pengaturan tempat bahan atau barang yang tetap dan identifikasi bahan atau barang tertentu.
- b. Sentralisasi kekuasaan dan tanggung jawab pada satu orang dapat dipercaya terutama penjaga gudang.
- c. Suatu sistem pencatatan dan pemeriksaan atas penerimaan bahan baku atau barang.
- d. Pengawasan mutlak atas pengeluaran bahan baku atau barang.
- e. Pencatatan yang cukup teliti yang menunjukkan jumlah yang dipesan yang dibagikan atau dikeluarkan dan yang tersedia dalam gudang.
- f. Pemeriksaan fisik bahan atau barang yang ada dalam persediaan secara langsung.
- g. Perencanaan untuk menggantikan barang-barang yang telah dikeluarkan. Barang-barang yang telah lama dalam gudang dan barang-barang yang sudah usang dan ketinggalan zaman.
- h. Pengecekan untuk menjamin dapat efektifnya kegiatan rutin.

## **2.6 *Reorder Point (ROP)***

Menurut Heizer, dan Render (2010) tingkat pemesanan kembali (*Reorder Point / ROP*) adalah suatu titik atau batas dari jumlah persediaan yang ada pada suatu saat dimana pemesanan harus diadakan kembali. Ada beberapa faktor yang mempengaruhi ROP antara lain:

1. *Lead Time.*

*Lead time* adalah waktu yang dibutuhkan untuk menunggu pesanan bahan baku datang atau tenggang waktu antara waktu pesan dengan kedatangan bahan baku

2. Tingkat pemakaian bahan baku rata-rata per satuan waktu tertentu.

3. *Safety Stock*

*Safety stok* adalah jumlah persediaan yang disediakan atau disimpan untuk mengantisipasi unsur ketidakpastian permintaan dan penyediaan.

Dari ketiga faktor di atas, maka reorder point dapat dicari dengan rumus

berikut ini:

$$\text{Reorder point} = (\mathbf{d} \times \mathbf{L}) + \mathbf{ss}$$

Keterangan:

$d$  = Permintaan harian

$L$  = Waktu tunggu pesanan

$ss$  = *Safety Stock*

### 2.7 *Economic Order Quantity (EOQ)*

Menurut Carter (2009) kuantitas pemesanan ekonomis (*Economic Order Quantity* - EOQ) adalah jumlah persediaan yang di pesan pada suatu waktu yang meminimalkan biaya persediaan tahunan. Sedangkan menurut Heizer dan Render (2009) EOQ adalah sebuah teknik kontrol persediaan yang meminimalkan biaya total dari pemesanan dan penyimpanan serta berdasar pada beberapa asumsi:

1. Jumlah permintaan diketahui, konstan, dan independen.
2. Waktu tunggu yakni waktu antara pemesanan dan penerimaan pesanan- telah diketahui dan bersifat konstan.

3. Penerimaan persediaan bersifat instan dan selesai seluruhnya.
4. Tidak tersedia diskon kuantitas.
5. Biaya variabel hanya biaya untuk menyiapkan atau melakukan pemesanan (biaya penyetelan) dan biaya menyimpan persediaan dalam waktu tertentu (biaya penyimpanan dan untuk membawa persediaan).
6. Kehabisan atau kekurangan persediaan dapat sepenuhnya dihindari jika pemesanan dilakukan pada waktu yang tepat.

Model *Economic Order Quantity* (EOQ) merupakan model matematik yang menentukan jumlah barang yang harus dipesan untuk memenuhi permintaan yang diproyeksikan dengan biaya persediaan yang diminimalkan. *Economic Order Quantity* (EOQ) dinyatakan dengan rumus sebagai berikut:

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

Keterangan:

D = Kebutuhan barang per periode

S = biaya *setup*/biaya order per pesanan

H = biaya penyimpanan per periode per unit

## 2.8 Desktop

Menurut Konixbam (2009) *desktop based application* merupakan suatu aplikasi yang mampu berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi internet di suatu komputer otonom dengan *operating system* atau *platform* tertentu. Aplikasi *desktop* difokuskan kepada aplikasi yang lebih independen. Hal tersebut bertujuan untuk mempermudah para pengguna aplikasi *desktop* dalam hal memodifikasi pengaturan aplikasi sehingga efektifitas, efesinsi waktu, dana, dan tenaga dapat lebih ditekankan semaksimal mungkin.

Secara garis besar pada pemrograman terutama pada aplikasi yang berbasis *desktop* dapat dibedakan menjadi dua macam yaitu:

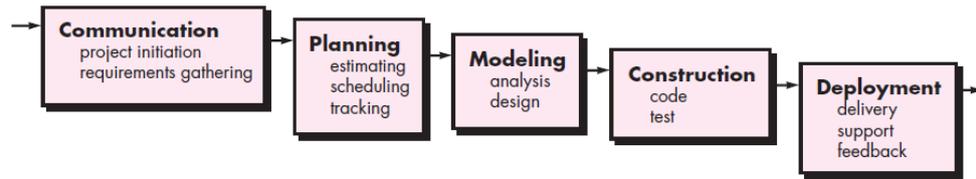
1. Pemrograman konvensional adalah metode yang digunakan untuk mendesain suatu aplikasi, pemrograman dituntut untuk bisa menerapkan baris demi baris kode program agar bisa menghasilkan sebuah bentuk tampilan aplikasi yang dibuat dan memakan waktu lama.
2. Pemrograman visual merupakan metode pembuatan program dimana seorang programmer membuat koneksi antar objek-objek dengan cara menggambar, menunjuk, dan mengklik pada diagram dan *icon* dengan interaksi diagram jalur.

Aplikasi *desktop* memiliki beberapa keunggulan yaitu:

1. Dapat berjalan dengan independen, tidak perlu menggunakan sebuah *web browser*.
2. Tidak memerlukan koneksi internet
3. Prosesnya lebih cepat dibanding aplikasi *web*
4. Lebih aman dari gangguan pencurian data dan virus

## **2.9 System Development Life Cycle**

Menurut Dennis, Wixom & Roth (2012) *system development life cycle* adalah proses menentukan bagaimana suatu sistem dapat mendukung kebutuhan bisnis, merancang sistem, membangun sistem tersebut, dan memberikannya kepada pengguna. Sedangkan Menurut Pressman (2010), salah satu model pengembangan sistem adalah dengan model waterfall. Waterfall model adalah model yang paling populer dan sering dianggap sebagai pendekatan klasik dalam daur hidup pengembangan sistem. Adapun tahapan-tahapannya dapat dilihat pada Gambar 2.1.



Gambar 2.1 *System Development Life Cycle*  
(Sumber: Pressman, 2010)

### 1. *Communication*

Tahap *communication* merupakan tahapan yang digunakan untuk pengumpulan data. Data yang dikumpulkan ini untuk merancang dan membangun aplikasi. Tahapan dalam pengumpulan data ini meliputi proses wawancara, dokumentasi, observasi.

### 2. *Planning*

Tahap ini merupakan tahap dimana akan dilakukan estimasi mengenai kebutuhan-kebutuhan yang diperlukan untuk membuat sebuah sistem. Selain itu proses analisa kebutuhan pengguna dan fungsional serta penjadwalan dalam proses pengerjaan juga ditentukan pada tahap ini.

### 3. *Modeling*

Kemudian mulai masuk pada tahap perancangan dimana perancang menerjemahkan kebutuhan sistem kedalam representasi untuk menilai kualitas sebelum tahap selanjutnya dikerjakan. Tahap ini lebih difokuskan pada atribut program, seperti struktur data, arsitektur perangkat lunak, dan detail prosedur.

### 4. *Construction*

Tahap ini merupakan tahap dimana perancangan diterjemahkan ke dalam bahasa yang dimengerti oleh mesin. Setelah itu dilakukan pengetesan/ pengujian terhadap sistem yang telah dibuat.

## 5. *Deployment*

Setelah proses pengkodean dan pengujian selesai, dilakukan pengiriman yang artinya implementasi kepada masyarakat luas. Pada tahap ini juga dilakukan pemeliharaan, perbaikan, dan pengembangan agar sistem tersebut tetap dapat berjalan sebagaimana fungsinya.

### 2.10 *Database*

Menurut Kustiyaningsih (2011) *database* adalah struktur penyimpanan data. Untuk menambah, mengakses dan memproses data yang disimpan dalam sebuah *database* komputer, diperlukan sistem manajemen *database* seperti MySQL Server.

Menurut Anhar (2010) *database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur file yang menyusun sebuah *database* adalah *Data record* dan *field*. *Database* memiliki beberapa tipe relasi yang digunakan untuk menghubungkan tabel-tabel yang terdapat didalamnya. Tipe relasi tersebut adalah *one to one*, *one to many* dan *many to many*. Untuk melakukan interaksi antara program dengan *database* dibutuhkan sebuah DBMS.

DBMS (*Database Management System*) adalah sebuah sistem *software* yang memungkinkan pengguna untuk mendefinisikan, memuat, memelihara, dan mengatur akses ke dalam *database* (Connolly dan Begg, 2010). Sebuah DBMS menyediakan beberapa fasilitas sebagai berikut:

#### 1. *Data Definition Language* (DDL)

DDL adalah sebuah bahasa yang mengijinkan *database administrator* atau pengguna untuk menggambarkan dan memberi nama dari entitas, *attribute*, dan *relationship* yang dibutuhkan untuk aplikasi bersama dengan semua kepercayaan

yang berhubungan dengan batasan keamanan. Contoh dari operasi DDL adalah *create*, *drop*, dan *alter* tabel.

## 2. *Data Manipulation Language* (DML)

DML adalah sebuah bahasa yang menyediakan sekumpulan operasi yang mendukung operasi manipulasi data di dalam *database*. Contoh dari operasi DML adalah *insert* dan *update*.

### 3. Menyediakan kontrol akses ke *database*, sebagai contoh:

- a. *Security system*, dimana mencegah pengguna yang tidak mempunyai hak untuk mengakses *database*.
- b. *Integrity system*, dimana menjaga konsistensi dari data.
- c. *Concurrency control system*, dimana mengizinkan akses yang terbagi dalam *database*.
- d. *Recovery control system*, dimana mengembalikan kondisi *database* sebelum kegagalan *hardware* atau *software*.
- e. *User-accessible catalog*, dimana berisi deskripsi dari data dalam *database*.

## 2.11 SQL Server

Menurut Djuandi (2002) SQL Server merupakan sebuah sistem arsitektur terbuka yang dapat memungkinkan para pengembang program memperluas dan menambahkan fungsi-fungsi ke dalam *database* tersebut. SQL Server merupakan produk *Relational Database Management System* (RDBMS) yang dibuat oleh Microsoft. SQL Server banyak digunakan pada dunia bisnis, pendidikan serta pemerintahan sebagai solusi *database* atau penyimpanan data. Pada tahun 2008 Microsoft mengeluarkan SQL Server 2008. SQL Server 2008 ini memiliki beberapa fitur yang dapat digunakan antara lain:

1. *XML Support*. Dengan fitur ini, pengguna dapat menyimpan dokumen XML dalam suatu tabel, meng-*query* data ke dalam *format* XML melalui Transact-SQL dan lain sebagainya.
2. *Multi-Instance Support*. Fitur ini memungkinkan pengguna untuk menjalankan beberapa *database engine* SQL Server pada mesin yang sama.
3. *Data Warehousing and Business Intelligence (BI) Improvements*. SQL Server dilengkapi dengan beberapa fungsi yang dapat digunakan untuk keperluan *Business Intelligence* melalui *Analysis Services*. SQL Server 2008 juga ditambahi dengan *tools* yang digunakan untuk keperluan *data mining*.
4. *Performance and Scalability Improvements*. SQL Server menerapkan *distributed partitioned views* yang memungkinkan pengguna untuk membagi *workload* ke beberapa server sekaligus.
5. *Query Analyzer Improvements*. Fitur ini memiliki beberapa fungsi yang dapat digunakan antara lain: *integrated debugger*, *object browser*, dan *fasilitas object search*.
6. *DTS Enhancement*. Fitur ini mampu digunakan untuk memperhatikan *primary key* dan *foreign key constraints*. Ini berguna pada saat migrasi tabel dari RDBMS lain.
7. *Transact-SQL Enhancements*. T-SQL dapat mendukung UDF (*User-Definable Function*). Ini memungkinkan pengguna untuk menyimpan rutin-rutin ke dalam *database engine*.

### **2.12 Black Box Testing**

Menurut Nidhra dan Dondeti (2012) *black box testing* disebut juga sebagai *functional testing*, sebuah teknik pengujian fungsional yang merancang *test case*

berdasarkan informasi dari spesifikasi. Menurut Pressman (2010) *black-box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan pengguna untuk memperoleh set kondisi masukan yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. Pengujian melibatkan pemahaman rinci mengenai domain aplikasi, masalah bisnis yang dipecahkan oleh aplikasi dan misi yang dilakukan sistem.

*Black box testing* digunakan untuk menguji fungsi-fungsi yang tersedia dari aplikasi apakah telah berjalan sebagaimana mestinya. Pengujian *black box testing* berusaha untuk menemukan fungsi yang berjalan tidak sesuai atau hilang, kesalahan *interface*, kesalahan dalam struktur data, serta kesalahan inisialisasi. Pengujian *black box* paling baik dilakukan oleh penguji yang memiliki kemampuan untuk memahami desain sistem, setidaknya pada tingkatan yang cukup tinggi sehingga mereka dapat secara efektif menemukan *bug* atau *error* umum untuk jenis desain.

