



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Teori Umum

2.1.1 Pengertian Data

Data adalah sekumpulan fakta yang berupa karakter, angka, simbol, tanda-tanda, isyarat, tulisan, suara, maupun bunyi dimana hal-hal tersebut mempresetasikan keadaan sebenarnya yang selanjutnya digunakan sebagai masukan untuk sistem informasi. Dalam penggunaan sehari-hari, data itu diartikan sebagai pernyataan yang diterima secara apa adanya. Pernyataan ini merupakan hasil pengukuran atau pengamatan pada lokasi kejadian.

Dalam dunia keilmuan, fakta-fakta yang didapat kemudian dikumpulkan menjadi data. Kemudian data tersebut diolah sehingga bisa secara jelas dan tepat diutarakan sehingga orang lain yang mendengarnya itu bisa mengerti walaupun tidak secara langsung orang tersebut mengalaminya sendiri, hal semacam ini disebut deskripsi. Sedangkan pemilihan data sesuai dengan persamaan atau perbedaan yang dikandungnya dinamakan klasifikasi.

2.1.2 Pengertian Informasi

Terdapat banyak sekali pengertian mengenai informasi tetapi pada intinya semua itu sama. Informasi itu sendiri adalah sekumpulan data dimana data

tersebut sudah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan lebih penting juga berguna untuk pengambilan keputusan dimasa mendatang. Sebenarnya informasi ini memiliki banyak arti sesuai dengan konteksnya masing-masing. Dalam bidang ilmu komputer, informasi adalah data yang disimpan, diproses atau ditransmisikan.

2.1.3 Pengertian Sistem

Williams dan Sawyer (2005) berpendapat bahwa sistem itu adalah sekumpulan komponen yang saling berhubungan yang berinteraksi untuk melakukan suatu pekerjaan dalam usaha untuk mencapai suatu tujuan. Sedangkan O'Brien (2005) berpendapat bahwa sebuah sistem itu adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur.

Berdasarkan kedua definisi diatas, dapat disimpulkan bahwa sistem merupakan sekelompok komponen yang berkaitan, berhubungan erat, dan berinteraksi untuk mencapai suatu tujuan tertentu melalui tiga tahapan yaitu input, proses, dan output.

2.1.4 Pengertian Sistem Informasi

Menurut O'Brien (2006), "Sistem Informasi adalah kombinasi teratur dari *brainware, hardware, software*, jaringan komunikasi, dan sumberdaya data

yang dikumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi”.

Menurut Laudon (2007), “Sistem informasi adalah sekumpulan komponen yang saling berhubungan yang bekerja sama mengumpulkan atau mengambil, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan dalam suatu organisasi”.

Menurut Turban (2008), “Sistem informasi adalah pemasok informasi yang berguna untuk memproses data menjadi informasi dan pengetahuan”.

Dari kedua definisi diatas, dapat disimpulkan bahwa sistem informasi adalah komponen-komponen yang saling berhubungan dan bekerjasama untuk menghasilkan informasi yang digunakan untuk mendukung proses pengambilan keputusan.

2.1.5 Konsep Dasar Pemrograman

Menurut Sutabri (2005), “Pemrograman adalah suatu kumpulan instruksi atau kode yang disusun secara logis dan sistematis sehingga merupakan suatu prosedur yang lengkap yang akhirnya dapat digunakan oleh pemakai dalam mengoperasikan aplikasi”.

Bahasa komputer juga merupakan sarana komunikasi yang menjembatani antara manusia (*user*) dan komputer. Bahasa - bahasa yang dipakai oleh *programmer* untuk menuliskan kumpulan instruksi atau program disebut “Bahasa Pemrograman”. Ada dua jenis bahasa pemrograman yaitu :

1. Bahasa Tingkat Tinggi (*High Level Language*)

Bahasa pemrograman ini biasanya bersifat *mechine independent* dalam artian bisa berjalan pada mesin yang berbeda dan juga untuk menjadi *problem oriented language (POL)* yang merefleksikan jenis masalah yang dipecahkan. Untuk penulisannya ditulis dalam *statement* yang sama dengan bahasa inggris dan dalam bahasa *portable*, sehingga pemakai program dapat beralih ke komputer yang lebih baru tanpa menulis ulang program, pemakai komputer yang berbeda dapat membagi program, organisasi yang menghasilkan software bisa menjual software yang sama pada pemakai yang berbeda tanpa harus menulis ulang program untuk tiap jenis komputer. Contoh bahasa tingkat tinggi antara lain adalah *BASIC*, *COBOL*, *ALGOL*, dan sebagainya.

2. Bahasa Tingkat Rendah (*Low Level Language*)

Bahasa yang dimengerti oleh mesin komputer adalah instruksi dalam bahasa mesin (*mechine language*) yang merupakan bahasa tingkat rendah. Salah satu jenis bahasa komputer yang termasuk bahasa tingkat rendah ini disebut dengan *Assembler*. Instruksi-instruksi yang dibuat dalam bahasa pemrograman disebut program sumber (*source program*), sedangkan program yang sudah diterjemahkan kedalam bahasa mesin disebut program object (*object program*). Ciri

bahasa tingkat rendah bisa dikenal dari penulisan instruksinya dimana penulisannya mendekati instruksi-instruksi dalam bahasa mesin.

2.2 Teori Khusus

2.2.1 Normalisasi

Menurut Tata Sutabri (2005) “Untuk dapat melakukan langkah-langkah sesuai yang diberikan oleh metodologi pengembangan sistem yang terstruktur, maka dibutuhkan suatu alat dan teknik untuk dapat melaksanakannya”. Alat – alat atau teknik yang dimaksud itu berupa gambar, tabel, kode, atau diagram.

Linda Marinda (2004) berpendapat bahwa “Normalisasi adalah proses pengelompokan elemen dan menjadi table-tabel yang menunjukkan *entity* dan relasinya”. Dimana kondisi yang diuji untuk membantu mengurangi atau mencegah timbulnya masalah yang berhubungan dengan pengolahan data di dalam database.

Terdapat beberapa macam kunci (*key function*) yang biasa digunakan dalam pengolahan database yaitu sebagai berikut :

a. *primary key*

Kunci utama adalah atribut yang berfungsi untuk mengidentifikasi setiap *record* secara unik. *Primary key* harus *field* yang benar-benar unik dan tidak boleh bernilai *Null*.

b. *candidate key*

Kunci calon adalah salah satu rangkaian yang mempunyai nilai unik untuk membedakan atau mengidentifikasi nilai-nilai kombinasi yang unik diantara semua kejadian yang spesifik dari entitas.

c. *alternate key*

Kunci kandidat yang tidak dipakai sebagai *primary key*.

d. *foreign key*

satu atribut yang melengkapi satu hubungan (*relationship*)

Dalam proses normalisasi ini terdapat beberapa tahapan yang harus dilakukan yaitu:

a) Bentuk tidak normal (*unnormalized form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada aturan untuk mengikuti suatu format tertentu, dalam bentuk ini data itu belum lengkap dan masih ada terjadi duplikasi. Hal ini terjadi karena data yang dikumpulkan itu apa adanya sesuai kedatangannya.

b) Bentuk normal kesatu (*1NF/first normal form*)

Bentuk normal kesatu memiliki ciri yaitu data di bentuk dalam *flat file* (file data rata-rata), data dibentuk dalam suatu *record* demi *record* dan nilai dari *field* berupa *atomic value*. Tidak ada atribut yang bernilai ganda.

c) Bentuk normal kedua (*2NF/second normal form*)

Bentuk normal kedua memiliki syarat yaitu bentuk data telah memenuhi kriteria dari bentuk normal pertama. Hubungan setiap atribut antara *non-primary-key* dan bentuk normal pertama semua bergantung pada *primary key*.

d) Bentuk normal ketiga (*3NF/third normal form*)

Bentuk normal ketiga bentuk yang relasinya bentuk normal kedua dan semua atribut bukan primer dan keduanya tidak memiliki hubungan transitif. Dengan kata lain, atribut-atribut yang ada itu bukanlah kunci, tetapi atribut tersebut bergantung pada *primary key* secara menyeluruh.

e) *Boyce-Codd normal form (BCNF)*

Untuk menjadi *bcnf*, relasi harus dalam bentuk normal pertama dan setiap atribut bergantung pada atribut *super key*. (Begg 2010)

2.2.2 Konsep DFD

Data flow diagram (DFD) adalah representasi grafik dari sebuah sistem.

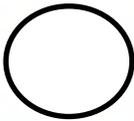
DFD ini menggambarkan komponen-komponen yang ada dalam sebuah sistem beserta aliran-aliran data. DFD biasanya digunakan untuk dokumentasi sistem informasi yang ada atau sistem yang sedang berjalan. (Kendall 2003)

Kelebihan penggunaan DFD menurut Kendall (*Analisis dan Perancangan Sistem*, p263 2006) adalah sebagai berikut :

- 1) Kebebasan dari menjalankan implementasi teknis dari suatu sistem terlalu dini.

- 2) Pemahaman yang lebih mengenai interaksi atau hubungan antara sistem dan subsistem yang saling terkait.
- 3) Mengkomunikasikan pengetahuan tentang sistem dengan pengguna yang lain melalui aliran data.
- 4) Melakukan analisa pada sistem mengenai data yang diolah. maksudnya adalah data yang diproses itu apakah sudah ditetapkan atau belum.

Notasi Dasar DFD

NOTASI	KETERANGAN
	Himpunan Entitas Luar Himpunan entitas luar yaitu sebuah elemen sistem (misalnya perangkat keras, seseorang atau program lainnya) yang menghasilkan informasi bagi transformasi oleh perangkat lunak atau menerima informasi yang dihasilkan oleh perangkat lunak.
	Aliran Data Menggambarkan aliran data dari satu proses ke proses lainnya. Anak panah menunjukan aliran data.
	Proses Proses atau fungsi yang mentransformasikan data secara umum.
	Penyimpanan Data (<i>data source</i>) Penyimpanan data berfungsi untuk menyimpan informasi yang digunakan oleh perangkat lunak.
	<i>OFF-PAGE CONECTOR</i> <i>Off-page connector</i> digunakan untuk menghubungkan proses dengan proses lain satu tingkat di atasnya.

Tabel 2.1 Notasi DFD

a. Persyaratan DFD

Ada persyaratan yang harus dipenuhi ketika membuat DFD untuk model dari sebuah sistem yaitu:

1. Di dalam DFD, *external entity* tidak dapat bertemu dengan data store.
2. Data store tidak dapat bertemu dengan *external entity*.
3. Setiap proses harus ada *data flow* yang masuk dan *data flow* yang keluar.
4. Di dalam DFD, *external entity* dan *external entity* tidak boleh bertemu.

b. Simbol Yang Digunakan

Ada beberapa symbol yang digunakan dalam DFD yaitu :

1. *External entity*

External entity ini dapat berupa orang, organisasi, atau sistem lain yang berada dilingkungan luarnya yang akan memberikan *input* dan menerima *output*.

2. *Data flow*

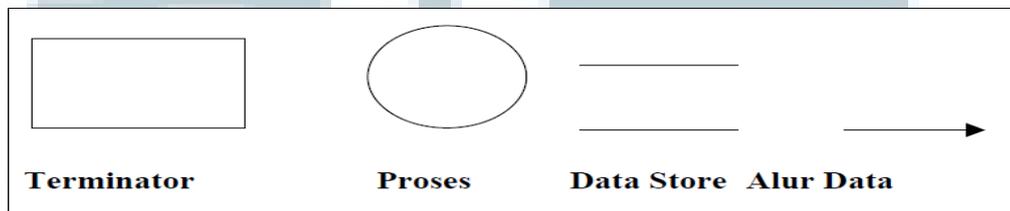
Arus data (*data flow*) diberi simbol anak panah. Arus data ini mengalir di antara *process*, *data store* dan *external entity*. Arus ini merupakan arus dari data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.

3. *Process*

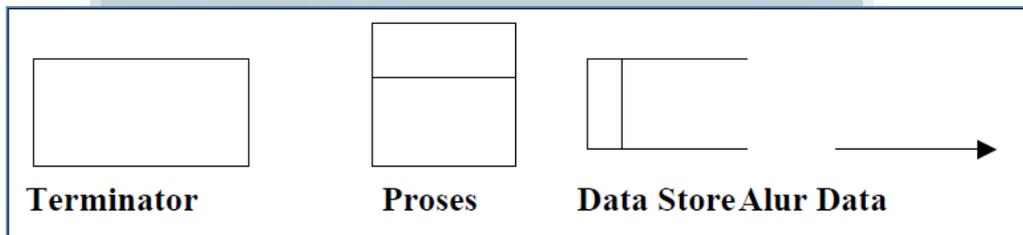
Proses merupakan kegiatan yang dilakukan oleh orang atau mesin komputer. Proses dapat ditunjukkan dengan dengan lingkaran atau juga dengan sigi empat dengan sudut tumpul.

4. *Data store*

Symbol yang digunakan berupa sepasang garis *horizontal parallel*, dimana salah satu ujungnya tertutup.



Gambar 2.1 Komponen DFD (Yourdan dan DeMarco)

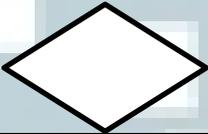


Gambar 2.2 Komponen DFD (Gene dan Serson)

2.2.3 Konsep ERD

Entity-Relationship Diagram (ERD) merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu himpunan entitas yang memiliki atribut dengan himpunan entitas yang lain dalam satu sistem yang terintegrasi.

Berikut adalah Symbol-sombol yang terdapat pada *Entity-Relationship Diagram (ERD)*.

NOTASI	KETERANGAN
	Himpunan Entitas,yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu ke satu, satu ke banyak dan banyak ke banyak.
	Atribut, yaitu karakteristik dari <i>entity</i> atau relasi yang merupakan penjelasan detail tentang entitas.
	Garis, hubungan antara <i>entity</i> dengan atributnya dan himpunan entitas dengan himpunan relasinya.

Tabel 2.2 Simbol ERD

Atribut menentukan property suatu objek data. Atribut dapat digunakan untuk menamai sebuah contoh dari entitas, menggambarkan contoh, dan membuat referensi ke kecontoh yang lain pada table yang lain. Satu atribut atau lebih harus didefinisikan sebagai sebuah pengidentifikasi, yang mana atribut pengidentifikasi ini akan menjadi sebuah kunci untuk menemukan sebuah contoh dari entitas (Pressman 2002). Relasi menunjukan adanya hubungan diantara sejumlah entitas yang semuanya itu berasal dari banyak entitas yang berbeda. Kardinalitas merupakan spesifikasi dari sejumlah entitas dari satu entitas yang bisa dihubungkan ke entitas lainnya. Kardinalitas biasanya

diekspresikan secara sederhana dengan 'satu' atau 'banyak'. Dengan pertimbangan kombinasi 'satu' dan 'banyak', dua himpunan entitas dapat dihubungkan sebagai satu-ke-satu (1:1), satu-ke-banyak (1:N), dan banyak-ke-banyak(M:N).

a. Satu-Ke-Satu (1:1)

Suatu entitas dari himpunan entitas A dapat berhubungan dengan satu dan hanya satu dari himpunan entitas B, dan sebuah entitas dari himpunan entitas B hanya dapat berhubungan dengan satu entitas dari himpunan entitas A.

Misalnya, seorang suami hanya memiliki satu orang istri, dan seorang istri hanya mempunyai seorang suami (di New Jersey)

b. Satu-Ke-Banyak

Suatu entitas dari himpunan entitas A dapat berhubungan dengan satu atau lebih entitas dari himpunan entitas B, tetapi sebuah entitas dari himpunan entitas B hanya dapat berhubungan dengan satu entitas dari himpunan entitas A.

Misalnya, seorang ibu kandung bisa memiliki banyak anak, tetapi seorang anak hanya memiliki satu orang ibu kandung.

c. Banyak-Ke-Banyak

Suatu entitas dari himpunan entitas A dapat berhubungan dengan satu atau lebih entitas dari himpunan entitas B, sementara sebuah entitas dari himpunan entitas B dapat berhubungan dengan satu atau lebih entitas dari himpunan

entitas A. misalnya, seorang paman memiliki banyak keponakan, dan seorang keponakan dapat memiliki banyak paman.

2.2.4 Konsep Sistem

Sistem informasi manajemen atau dikenal dengan SIM terpadu adalah penerapan sistem informasi dalam organisasi guna mendukung informasi di setiap tingkatan manajemen.

a. Karakteristik Sistem

Sebuah sistem memiliki karakteristik tertentu, antara lain yaitu :

1. Komponen sistem (*component*)

Suatu sistem biasanya terdiri dari komponen-komponen yang saling bekerjasama membentuk satu kesatuan dan saling berinteraksi.

2. Batasan sistem (*boundary*)

Daerah yang membatasi satu sistem dengan sistem lain atau lingkungan luarnya. Batasan sistem ini menunjukkan ruang lingkup (*scope*) dari sistem itu sendiri.

3. Lingkungan luar sistem (*environment*)

Segala sesuatu yang berada diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan sistem ini bisa merugikan dan bisa juga menguntungkan. Jika sifatnya menguntungkan, maka

harus dijaga dan dipelihara. Jika sifatnya merugikan, maka harus dikendalikan atau ditahan agar tidak mengganggu sistem.

4. Penghubung sistem (*interface*)

Penghubung sistem adalah media yang menghubungkan sistem satu dengan subsistem yang lainnya. Penghubung ini yang memungkinkan sumber daya mengalir dari satu subsistem ke subsistem yang lain. Contoh, keluaran (*output*) dari satu sistem akan menjadi masukan (*input*) dari subsistem lainnya.

5. Masukkan sistem (*input*)

Input merupakan data yang dimasukkan ke dalam sistem yang dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah data yang dimasukkan ke dalam sistem supaya sistem tersebut bisa beroperasi sedangkan *signal input* adalah data yang diproses untuk mendapatkan *output*.

6. Keluaran sistem (*output*)

Output hasil dari data yang telah di proses atau diolah dan menjadi berguna untuk organisasi dalam pengambilan keputusan di masa yang akan datang.

7. Pengolahan (*process*)

Bagian yang akan mengolah masukan (*input*) menjadi keluaran (*output*).

8. Tujuan sistem (*objective goal*)

Setiap sistem yang dibuat pasti memiliki tujuan (*goal*) atau sasaran (*objectives*). Sasaran dari sebuah sistem sangat penting, karena menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan. Sebuah sistem dikatakan berhasil apabila mengenai sasaran atau tujuan yang diinginkan.

b. Klasifikasi Sistem

1. Sistem abstrak dan Sistem fisik

Sistem abstrak adalah sistem yang bentuknya tidak tampak secara fisik, dalam artian sistem ini berupa ide-ide atau suatu pemikiran, sedangkan sistem fisik adalah sistem yang ada atau tampak secara fisik.

2. Sistem alamiah dan Sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, sedangkan sistem buatan manusia adalah sistem yang melibatkan interaksi manusia dengan mesin.

3. Sistem terbuka dan Sistem tertutup

Sistem terbuka merupakan sistem yang berhubungan dan mempengaruhi lingkungan luarnya, sedangkan sistem tertutup adalah sistem yang tidak terpengaruh oleh lingkungan luarnya.

4. Sistem *deterministic* dan Sistem *probabilistic*

Sistem *deterministic* adalah sistem yang beroperasi dengan tingkahlaku yang dapat diprediksi, misalnya sistem pada komputer, sedangkan sistem *probabilistic* merupakan sistem yang kondisi masa depannya tidak dapat diprediksi. (Sutarbi 2005)

2.2.5 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang membentuk blok bangunan (*building blok*), yang terdiri dari komponen masukan (*input*), komponen keluaran (*output*), komponen orang (*people*), komponen jaringan (*network*), komponen perangkat keras (*hardware*), komponen perangkat lunak (*software*), komponen basis data (*database*), dan komponen kontrol. Komponen-komponen tersebut saling berinteraksi membentuk satu kesatuan untuk mencapai sasaran.

1. Masukan (*input*)

Input yang dimaksud adalah semua data yang masuk ke sistem informasi.

2. Keluaran (*output*)

Informasi yang berkualitas dan berguna bagi semua orang yang menggunakan sistem.

3. Data

Secara konseptual, data adalah deskripsi tentang benda, kejadian dan aktivitas yang tidak mempunyai makna atau tidak berpengaruh

secara langsung kepada pemakainya atau disebut sebagai sekumpulan fakta mentah.

4. Orang (*people*)

Semua pihak yang bertanggung jawab sebagai sponsor sistem informasi (*sistem owner*), pengguna sistem (*sistem user*), perancang sistem (*sistem designer*) dan pengembang sistem (*sistem development*)

5. Jaringan (*network*)

Sistem yang menghubungkan semua perangkat dan memungkinkan suatu sumber dipakai secara bersamaan, baik pada waktu dan tempat bersamaan ataupun berbeda.

6. Perangkat keras (*hardware*)

Mencakup piranti-piranti fisik yang berfungsi sebagai media penyimpanan dan pendukung sistem informasi.

7. Perangkat lunak (*software*)

Sekumpulan instruksi atau perintah-perintah yang memungkinkan perangkat keras bisa digunakan untuk memproses data.

8. Basis data (*database*)

Sekumpulan informasi yang saling berkaitan dan berhubungan satu dengan yang lainnya dan tersimpan secara sistematis di dalam perangkat keras sehingga dapat diperiksa menggunakan *software* untuk memperoleh informasi dari basis data tersebut. *Software* yang digunakan untuk memanggil dan mengelola *query* basis data disebut sistem manajemen basis data (*database management sistem, DBMS*).

9. Kontrol

Pengendalian yang dirancang untuk meyakinkan bahwa tidak akan terjadi hal-hal yang dapat merusak sistem informasi ataupun tindakan perbaikan pada sistem yang sudah terlanjur rusak.

2.2.5 Sistem Informasi Penjualan, Pembelian, dan Inventori

a. Penjualan

Penjualan (*selling*) menurut Simamora (2000) adalah “pendapatan lazim dalam perusahaan dan merupakan jumlah kotor yang dibebankan kepada pelanggan atas barang dan jasa”. Sedangkan menurut Marom (2002) “Penjualan artinya menjual barang dagangan sebagai usaha pokok perusahaan yang biasanya dilakukan secara teratur”.

Dari definisi di atas bisa disimpulkan bahwa penjualan merupakan persetujuan dari dua belah pihak yakni penjual dan pembeli, dimana penjual menawarkan suatu produk dengan harapan pembeli dapat menyerahkan sejumlah uang sebagai alat ukur produk tersebut sebesar harga yang disepakati.

b. Pembelian

Pembelian adalah proses penemuan sumber dan pemesanan bahan, jasa, dan perlengkapan. Tujuan utama pembelian adalah memperoleh bahan dengan biaya serendah mungkin yang konsisten dengan kualitas dan jasa yang dipersyaratkan. Menurut Assauri (2008) pembelian

merupakan salah satu fungsi yang penting dalam berhasilnya operasi suatu perusahaan.

c. Inventori

Persediaan (*inventory*) adalah setiap sumber daya yang disimpan yang digunakan untuk memuaskan kebutuhan pelanggan pada saat sekarang maupun masa yang akan datang. Persediaan ini merupakan aspek penting bagi perusahaan yang menjual barang dagang. Menurut Maarif (2003) “Persediaan adalah suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam proses produksi ataupun persediaan barang baku yang masih menunggu untuk digunakan dalam suatu proses produksi”. Menurut Pardede (2005) “Persediaan atau *inventory* adalah sejumlah bahan baku atau barang yang tersedia untuk digunakan sewaktu-waktu dimasa yang akan datang.

Dari beberapa definisi diatas dapat disimpulkan bahwa persediaan itu merupakan aktiva dari suatu perusahaan, apakah dalam bentuk mentah (bahan baku), ataupun barang yang sedang diproses, dan atau barang jadi.

2.2.6 Kekuatan Kata Sandi

Password atau kata sandi adalah sekumpulan karakter atau juga berupa string yang digunakan oleh *user* atau pengguna jaringan, sistem operasi atau aplikasi dimana sistem operasi atau aplikasi tersebut mendukung banyak pengguna (*multi user*) untuk memverifikasi identitas pengguna itu sendiri

kepada system keamanan yang dimiliki oleh jaringan, sistem operasi ataupun aplikasi tersebut. *Password* juga bias diartikan sebagai kata rahasia yang berguna sebagai pengenalan.

Sistem keamanan akan membandingkan kode-kode yang dimasukkan oleh pengguna (yang terdiri atas nama pengguna/*user name* dan *password*) dengan daftar atau basis data yang disimpan oleh sistem keamanan sistem atau jaringan tersebut (dengan menggunakan metode autentikasi tertentu, seperti halnya kriptografi, *hash* atau lainnya). Jika kode yang dibandingkan cocok, maka sistem keamanan akan mengizinkan akses kepada pengguna tersebut terhadap layanan dan sumber daya yang terdapat di dalam jaringan atau sistem tersebut, sesuai dengan level keamanan yang dimiliki oleh pengguna tersebut. Idealnya, kata kunci merupakan gabungan dari karakter teks alfabet (*A-Z, a-z*), angka (*0-9*), tanda baca (*!?,.-*) atau karakter lainnya yang tidak dapat (atau susah) ditebak oleh para *intruder* sistem atau jaringan. Meskipun begitu, banyak pengguna yang menggunakan kata sandi yang berupa kata-kata yang mudah diingat, seperti halnya yang terdapat dalam kamus, ensiklopedia (seperti nama tokoh, dan lainnya), atau yang mudah ditebak oleh *intruder* sistem.

Kekuatan sebuah *password* itu sangatlah penting dalam dunia perbisnisan saat ini. Hal ini dilakukan untuk melindungi akun atau informasi penting dalam perusahaan. Kekuatan *password* tergantung pada berbagai jenis karakter yang digunakan oleh pengguna, panjang keseluruhan dari *password*,

dan apakah *password* dapat ditemukan di kamus atau sering dijumpai. Selain itu *password* harus setidaknya 14 karakter atau lebih. *Password* sebaiknya sesuatu yang mudah bagi user untuk diingat, namun sulit bagi orang lain untuk menebak.

2.2.7 Platform .Net

Microsoft dengan visinya mengembangkan sistem aplikasi terdistribusi menembus era internet kembali membuat strategi baru. Microsoft berambisi menghubungkan informasi dan pelayanan software yang terintegrasi. *Tools* yang dikembangkan Microsoft semuanya dimulai dengan platform pengembangan berbasis .NET.

.NET *framework* adalah implementasi aktual dari teknologi berbasis .NET. Dalam framework .NET ini terdiri dari beberapa kumpulan pelayanan dan *class* pembangunan yang dapat digunakan untuk mengakses fungsi sistem operasi.

Visual Studio .NET adalah *IDE* yang digunakan untuk membangun aplikasi berbasis .NET. (Hartono n.d.)

Framework .NET merupakan komponen windows terpisah yang mendukung pengembangan dan eksekusi generasi berikutnya dari aplikasi dan layan web XML. Framework .NET diciptakan untuk tujuan:

- a) Menyediakan lingkungan pemrograman yang berorientasi object dan konsisten, dimana kode bisa disimpan dan dijalankan secara local, dieksekusi secara local tapi didistribusikan internet.
- b) Menyediakan lingkungan eksekusi yang bebas dari konflik perangkat lunak.
- c) Menyediakan lingkungan eksekusi yang aman, termasuk kode yang dibuat oleh orang ketiga yang tidak diketahui atau semi-dipercaya.
- d) Menyediakan lingkungan eksekusi kode yang menghilangkan masalah kinerja dari lingkungan *script*.
- e) Membuat pengembang berpengalaman dan konsisten pada beragam aplikasi, seperti aplikasi berbasis windows maupun aplikasi yang berbasis web.
- f) Untuk membangun komunikasi pada semua standar industri dan memastikan kode yang digunakan berdasarkan kerangka .NET dan bisa terintegrasi dengan kode lainnya. (What It's .NET n.d.)

U
M
N