

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pengertian Sistem**

Secara sederhana, suatu sistem diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variable yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh Kenneth Boulding terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. Kecenderungan manusia yang mendapat tugas memimpin suatu organisasi adalah terlalu memusatkan perhatian pada salah satu komponen saja dari sistem organisasi. Teori sistem mengatakan bahwa setiap unsur pembentuk organisasi adalah penting dan harus mendapatkan perhatian yang utuh supaya manajer dapat bertindak lebih efektif.

Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Unsur atau komponen disini bukan hanya bagian-bagian yang tampak secara fisik, tetapi juga hal-hal yang mungkin bersifat abstrak atau konseptual seperti misi, pekerjaan, kegiatan, kelompok informal, dan lain sebagainya.

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Suatu sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan sangat

menentukan dalam mendefinisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan.

Dapat disimpulkan sistem adalah kegiatan-kegiatan yang saling berhubungan antara satu sama yang lainnya yang terdiri dari objek-objek, unsur-unsur atau komponen-komponen sehingga membentuk suatu kesatuan pemrosesan untuk mencapai tujuan tertentu. (Tata Sutabri; 2005 : 2-8)

## **II.2. Pengertian Informasi**

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Atau Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Menurut Gordon B. Davis menjelaskan kaitannya data dengan informasi adalah data yang telah diproses kedalam suatu bentuk yang mempunyai arti bagi si penerima dan mempunyai nilai nyata dan terasa bagi keputusan yang saat ini atau keputusan mendatang.

“Informasi adalah data yang telah diklarifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan” .

Kualitas dari suatu sistem informasi (*quality of information*) ditentukan oleh tiga faktor yaitu sebagai berikut.

### **1. Akurat**

Akurat berarti bahwa informasi bebas dari kesalahan. Berarti informasi harus bebas dari kesalahan dan tidak menyesatkan harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke

penerima informasi kemungkinan banyak terjadi gangguan yang dapat merubah informasi.

## 2. Tepat waktu

Tepat waktu berarti bahwa informasi datang pada saat dibutuhkan sehingga bermanfaat untuk pengambilan keputusan. Berarti informasi yang datang pada penerima tidak boleh terlambat, informasi yang sudah usang tidak akan mempunyai nilai lagi sebab informasi merupakan landasan dalam mengambil keputusan.

## 3. Relevan

Relevan berarti informasi mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya adalah berbeda. Misalnya informasi mengenai harga produksi untuk ahli teknik perusahaan merupakan informasi yang kurang relevan jika ditujukan kepada akuntan.  
(Tata Sutabri ; 2005 : 23-36)

### **II.3. Pengertian Sistem Informasi**

Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut.

Informasi merupakan hal yang sangat penting dalam pengambilan keputusan, permasalahannya adalah dimana informasi tersebut didapat. Informasi dapat diperoleh dari sistem informasi. Robert A Leitch dan K. Roscoe Davis mendefinisikan sistem informasi sebagai berikut:

Sebuah sistem terdiri atas bagian-bagian atau komponen yang terpadu untuk suatu tujuan. Model dasar dari bentuk sistem ini adalah adanya masukan, pengolahan, dan keluaran. Akan tetapi, sistem ini dapat dikembangkan hingga menyertakan media penyimpanan.

“Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.” (Tata Sutabri ; 2005 : 42)

## **II.4. Sistem Informasi Laporan Keuangan**

### **II.4.1. Pengertian Laporan Keuangan**

Laporan keuangan merupakan hasil dari proses akuntansi yang dapat digunakan sebagai alat untuk berkomunikasi antara data keuangan atau aktivitas suatu perusahaan dengan pihak-pihak yang berkepentingan dengan data atau aktivitas perusahaan tersebut. Analisis keuangan, terutama analisis rasio keuangan adalah alat yang paling bermanfaat untuk menentukan bagaimana aktivitas usaha dijalankan. Dengan pengamatan dan analisis yang memadai atas hasil analisis rasio keuangan dapat membantu manajemen untuk menemukan keunggulan dan kelemahan perusahaan. Untuk analisis rasio keuangan, hanya diperlukan dua jenis laporan keuangan yaitu laporan Neraca dan laporan Laba Rugi. (Varietmi Wira; Jurnal Manajemen dan Kewirausahaan Volume 1 Nomor 1 2013 : 61)

#### II.4.2. Laporan Keuangan untuk Proses Penilaian Kesehatan Perusahaan

Penilaian kesehatan suatu perusahaan dapat dilakukan berdasarkan informasi yang diperoleh dari hasil perbandingan dari satu pos laporan keuangan dengan pos lain yang relevan dan signifikan yang disebut sebagai rasio. Rasio dapat diambil berdasarkan data neraca dan laporan laba rugi pertahun. Berdasarkan SK Menteri BUMN No .:KEP100/MBU/2002 tentang tingkat kesehatan perusahaan analisis- analisis rasio yang digunakan adalah :

##### 1. *Return On Equity* (ROE)

$$\text{Return on Equity} = \frac{\text{Laba setelah pajak}}{\text{modal sendiri}} \times 100\%$$

Definisi :

Laba setelah Pajak adalah Laba setelah Pajak dikurangi dengan laba hasil penjualan dari :

- Aktiva tetap
- Aktiva Non Produktif
- Aktiva Lain-lain
- Saham Penyertaan Langsung
  - a. Modal Sendiri adalah seluruh komponen Modal Sendiri dalam neraca perusahaan pada posisi akhir tahun buku dikurangi dengan komponen Modal sendiri yang digunakan untuk membiayai Aktiva Tetap dalam Pelaksanaan dan laba tahun berjalan. Dalam Modal sendiri tersebut di atas termasuk komponen kewajiban yang belum ditetapkan statusnya.

- b. Aktiva Tetap dalam pelaksanaan adalah posisi pada akhir tahun buku Aktiva Tetap yang sedang dalam tahap pembangunan.

**Tabel II.1. Daftar Skor Penilaian ROE**

ROE (%)	SKOR	
	Infra	Non Infra
15 < ROE	15	20
13 < ROE ≤ 15	13,5	18
11 < ROE ≤ 13	12	16
9 < ROE ≤ 11	10,5	14
7,9 < ROE ≤ 9	9	12
6,6 < ROE ≤ 7,9	7,5	10
5,3 < ROE ≤ 6,6	6	8,5
4 < ROE ≤ 5,3	5	7
2,5 < ROE ≤ 4	4	5,5
1 < ROE ≤ 2,5	3	4
0 < ROE ≤ 1	1,5	2
ROE = 0	1	0

( Sumber : SK Menteri BUMN No .:KEP100/MBU/2002)

## 2. Return On Investment (ROI)

$$\text{Return on invesment} = \frac{\text{EBIT} + \text{Penyusutan}}{\text{capital employed}} \times 100\%$$

Definisi :

- a. EBIT adalah laba sebelum bunga dan pajak
- b. Penyusutan depresiasi, amortisasi dan depleksi

- c. Capital employed adalah posisi pada akhir tahun buku Total Aktiva dikurangi aktiva tetap dalam pelaksanaan

**Tabel II.2. Daftar Skor Penilaian ROI**

ROI (%)	SKOR	
	Infra	Non Infra
18 < ROI	10	15
15 < ROI ≤ 18	9	13,5
13 < ROI ≤ 15	8	12
12 < ROI ≤ 13	7	10,5
10,5 < ROI ≤ 12	6	9
9 < ROI ≤ 10,5	5	7,5
7 < ROI ≤ 9	4	6
5 < ROI ≤ 7	3,5	5
3 < ROI ≤ 5	3	4
1 < ROI ≤ 3	2,5	3
0 < ROI ≤ 1	2	2
ROI < 0	0	1

( Sumber : SK Menteri BUMN No .:KEP100/MBU/2002)

### 3. Rasio kas/*Cash Rasio* (CAR)

Cash Ratio =

$$\frac{\text{Kas + bank + surat berharga jangka panjang}}{\text{hutang lancar}} \times 100\%$$

Definisi :

- a. Kas, bank dan surat berharga jangka pendek dilihat pada posisi masing-masing di akhir tahun
- b. Hutang lancar adalah posisi seluruh kewajiban lancar pada akhir tahun

**Tabel II.3. Daftar Skor Penilaian Rasio Kas**

Rasio Kas = x (%)	SKOR	
	Infra	Non Infra
$x > 35$	3	5
$25 \leq x < 35$	2,5	4
$15 \leq x < 25$	2	3
$10 \leq x < 15$	1,5	2
$5 \leq x < 10$	1	1
$0 \leq x < 5$	0	0

( Sumber : SK Menteri BUMN No .:KEP100/MBU/2002)

#### 4. *Current Rasio* ( Rasio Lancar )

$$\text{Current Ratio} = \frac{\text{Aktiva Lancar}}{\text{hutang lancar}} \times 100\%$$

Definisi :

- a. Aktiva Lancar adalah posisi Total Aktiva Lancar pada akhir tahun buku
- b. Hutang Lancar adalah posisi seluruh kewajiban Lancar pada akhir tahun buku

**Tabel II.4. Daftar Skor Penilaian Rasio Lancar**

Current Ratio = x (%)	SKOR	
	Infra	Non Infra
125 <= x	3	5
110 <= x < 125	2,5	4
100 <= x < 110	2	3
95 <= x < 100	1,5	2
90 <= x < 95	1	1
x < 90	0	0

( Sumber : SK Menteri BUMN No .:KEP100/MBU/2002)

#### 5. Collection Periods (COP)

$$CP = \frac{\text{Total Piutang Usaha}}{\text{Total Pendapatan Usaha}} \times 365 \text{ hari}$$

Definisi:

- a. Total Piutang usaha adalah posisi piutang setelah dikurangi cadangan penyisihan pada akhir tahun buku
- b. Total pendapatan usaha adalah jumlah pendapatan usaha selama tahun buku

**Tabel II.5. Daftar Skor Penilaian CP**

CP = x (hari)	SKOR	
	Infra	Non Infra
x <= 60	4	5

$60 < x \leq 90$	3,5	4,5
$90 < x \leq 120$	3	4
$120 < x \leq 150$	2,5	3,5
$150 < x \leq 180$	2	3
$180 < x \leq 210$	1,6	2,4
$210 < x \leq 240$	1,2	1,8
$240 < x \leq 270$	0,8	1,2
$270 < x \leq 300$	0,4	0,6
$300 < x$	0	0

( Sumber : SK Menteri BUMN No .:KEP100/MBU/2002)

6. *Inventory Turn Over*/Perputaran persediaan (PP)

$$\text{Perputaran Persediaan} = \frac{\text{Total Persediaan}}{\text{Total pendapatan usaha}} \times 365 \text{ hari}$$

Definisi :

- a. Total persediaan adalah seluruh persediaan yang ada, yang digunakan untuk proses produksi pada akhir tahun buku
- b. Total pendapatan usaha adalah jumlah pendapatan usaha selama tahun buku

**Tabel II.6. Daftar Skor Penilaian PP**

PP = x (hari)	SKOR	
	Infra	Non Infra
$x \leq 60$	4	5

$60 < x \leq 90$	3,5	4,5
$90 < x \leq 120$	3	4
$120 < x \leq 150$	2,5	3,5
$150 < x \leq 180$	2	3
$180 < x \leq 210$	1,6	2,4
$210 < x \leq 240$	1,2	1,8
$240 < x \leq 270$	0,8	1,2
$270 < x \leq 300$	0,4	0,6
$300 < x$	0	0

( Sumber : SK Menteri BUMN No .:KEP100/MBU/2002)

#### 7. Total Asset Turn Over (TATO)

$$\text{Total Asset Turn Over} = \frac{\text{Total Pendapatan}}{\text{Capital employed}} = x 100\%$$

Definisi :

- a. Total pendapatan adalah total pendapatan usaha dan non usaha tidak termasuk hasil penjualan aktiva tetap
- b. *Capital employed* adalah total aktiva dikurangi aktiva tetap dalam pelaksanaan pada akhir tahun buku

**Tabel II.7. Daftar Skor Penilaian TATO**

TATO = x (%)	SKOR	
	Infra	Non Infra
$120 < x$	4	5

105 < x <= 120	3,5	4,5
90 < x <= 120	3	4
75 < x <= 90	2,5	3,5
60 < x <= 75	2	3
40 < x <= 60	1,5	2,5
20 < x <= 40	1	2
x <= 20	0,5	1,5

( Sumber : SK Menteri BUMN No .:KEP100/MBU/2002)

#### 8. Total Modal Sendiri Terhadap Total Asset (TMS thd TA)

$$\text{Total modal Sendiri thd Total Asset} = \frac{\text{Total modal sendiri}}{\text{Total Aktiva}} \times 100\%$$

Definisi :

- a. Total modal sendiri adalah seluruh komponen modal pada akhir tahun buku diluar dana-dana yang belum ditentukan statusnya
- b. Total aktiva adalah *total asset*/aktiva dikurangi dana-dana yang belum ditentukan statusnya, pada akhir tahun buku. (SK Menteri BUMN No .:KEP100/MBU/2002 )

**Tabel II.8. Daftar Skor Penilaian TMSTTA**

TMS thd TA (%) = x	SKOR	
	Infra	Non Infra
x < 0	0	50
0 <= x < 10	2	4

$10 \leq x < 20$	3	6
$20 \leq x < 30$	4	7,25
$30 \leq x < 40$	6	10
$40 \leq x < 50$	5,5	9
$50 \leq x < 60$	5	8,5
$60 \leq x < 70$	4,5	8
$70 \leq x < 80$	4,25	7,5
$80 \leq x < 90$	4	7
$90 \leq x < 100$	3,5	6,5

( Sumber : SK Menteri BUMN No .:KEP100/MBU/2002)

## II.5. Database

Menurut James Martin dalam bukunya *Database Organization* “*Database* adalah suatu kumpulan data terhubung (*Interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangka data (*controlled redundancy*) dengan cara tertentu sehingga mudah digunakan atau ditampilkan kembali.”

Dalam *database* dapat digunakan oleh satu atau lebih program aplikasi secara optimal., data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya; data disimpan sedemikian rupa sehingga penambahan, pengambilan, dan modifikasi dapat dilakukan dengan mudah dan terkontrol.

Perangkat lunak yang mengelola *database* disebut sistem manajemen *database (database management system)*-DBMS. Semua DBMS memiliki suatu pengolah bahasa deskripsi data yang digunakan untuk menciptakan *database* serta pengelola suatu pengelola *database* yang menyediakan isi *database* bagi pemakai.

Sistem *database* memiliki beberapa kriteria, yaitu:

1. Bersifat *data oriented* dan bukan *program oriented*.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah *database*-nya.
3. Dapat berkembang dengan mudah baik *volume* maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah.
5. Dapat digunakan dengan cara-cara yang berbeda.
6. Kerangkapan data (*data redundancy*) minimal.

James F. Courtney Jr. Dan David B. Paradise menjelaskan dalam buku *Database System for Management*, sistem *database* adalah sekumpulan *database* yang dapat dipakai secara bersama-sama, personal-personal yang merancang dan mengelola *database*, teknik-teknik untuk merancang dan mengelola *database*, serta komputer untuk mendukungnya.

Sistem *database* mempunyai beberapa elemen penyusun sistem. Elemen-elemen pokok penyusun sistem *database* yaitu:

1. *Data base*
2. *Software* (perangkat lunak)

Perangkat lunak yang digunakan dalam suatu sistem *database* terdiri dari duamacam, yaitu *Data Base Management System (DBMS)* dan *Data Base Application Software (DBAS)*.

### 3. *Hardware* (Perangkat Keras)

Perangkat keras dalam suatu sistem *database* mempunyai komponen utama yang berupa unit pusat pengolah (*Central Processing Unit* atau CPU) dan unit penyimpanan (*Storage Unit*)

### 4. Manusia (*brainware*)



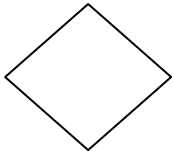
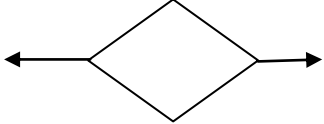
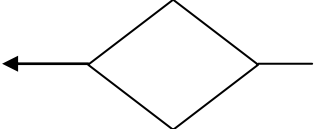
Manusia merupakan elemen penting pada sistem *database* karena manusia merupakan penggerak sistem. ." (Tata Sutabri; 2005:160-171)

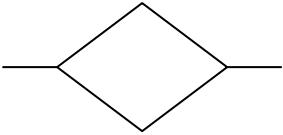
## II.5.1. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram (ERD)* adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan. ERD digunakan untuk melakukan pemodelan terhadap struktur data dan hubungannya. Penggunaan ERD dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah *database* yang baik.

*Entity* dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen-komponen data (atribut atau *field*) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia *database* *entity* memiliki atribut yang berperan sebagai kunci primer atribut deskriptif. Hal ini berarti setiap *entity* memiliki himpunan yang diperlukan sebuah *Primary Key* untuk membedakan anggota-anggota dalam himpunan tersebut.

Tabel II.9. Notasi *Entity Relationship Diagram*

Notasi	Komponen	Keterangan
	Entitas	Individu yang mewakili suatu objek dan dapat dibedakan dengan objek yang lain.
	Atribut	Properti yang dimiliki oleh suatu entitas, dimana dapat mendeskripsikan karakteristik dari entitas tersebut.
	Relasi	Menunjukkan hubungan diantara sejumlah entitas yang berbeda.
	Relasi 1:1	Relasi yang menunjukkan bahwa setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak satu entitas pada himpunan entitas kedua
	Relasi 1:N	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak atau sebaliknya. Setiap entitas dapat berelasi dengan banyak entitas pada himpunan entitas yang lain

	Relasi N: N	Hubungan ini menunjukkan bahwa setiap entitas pada himpunan entitas yang pertama dapat berhubungan dengan banyak entitas pada himpunan entitas yang kedua, demikian juga sebaliknya
---	-------------	---

( Sumber : Sri Mulanto ; 2010 : 30 )

### II.5.2. Normalisasi

Normalisasi data adalah proses dimana tabel-tabel pada *database* dites dalam hal saling ketergantungan diantara *field-field* pada sebuah tabel.

Adapun tahap dari normalisasi yaitu :

a. *Decomposition*

*Decomposition* adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenai dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di dekomposisi.

b. Bentuk tidak normal (*Unnormalized Form*)

Bentuk ini semua data yang ada pada *entity* (diambil atributnya) masih ditampung dalam tabel besar. Data yang ada pada tabel ini masih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak tertata rapi.

Adapun contohnya adalah sebagai berikut:

**Tabel II.10. Bentuk Tidak Normal**

Kd_agt	Nm_Agt	Almt	Kota	Telp	Tp_Lhr	Tg_Lhr	Jk	Tg_Msk	No_Trans
--------	--------	------	------	------	--------	--------	----	--------	----------

--	--	--	--	--	--	--	--	--	--

Kd_agt	Kd_Buku	Tg_Pnjm	Tg_KmbI	Tg_Tempo	Status	Denda	Jml_Pjm

Kd_Bk	Tg_Input	Jenis	Pengarahg	Judul	Penerbit	Thn_Trbt	Harga	Tmp_Bk	Ket	jumlah

(Sumber : Sri Mulanto ; 2010 : 33)

c. Bentuk normal kesatu ( 1NF/ *First Normal Form*)

Pada tahapan ini tabel di dekomposisikan dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki atribut ganda dan komposit. Contohnya adalah tabel pada bentuk tidak normal di pisahkan menjadi 3 tabel, yaitu tabel anggota, transaksi dan tabel buku.

**Tabel II.11. Tabel Anggota**

Kd_Agt	Nm_Agt	Almt	Kota	Telp	Tpt_Lhr	Tg_Lhr	Jk	Tg_Msk
A001	Herry	Krakatau	Smg	-	Smg	10/10/10	L	10/10/10

(Sumber : Sri Mulanto ; 2010 : 34)

**Tabel II.12. Tabel Buku**

Kd_Bk	Tg_Input	Jns_Bk	Pengarang	Judul	Penerbit

1.1.1	01/10/09	Fiksi	Acong	Akar Pahit	Bunga

Thn_Trbt	Harga	Tmp_Bk	Ket_Bk	Jumlah
2009	10000	01	Baru	5

(Sumber : Sri Mulanto ; 2010 : 34)

**Tabel II.13. Transaksi Peminjaman Buku**

No_Trans	Kd_Agt	Kd_Bk	Tg_Pnjm	Tgl_Kmbl	Tgl_Tmp	Status	Denda	Jml_Pjm

(Sumber : Sri Mulanto ; 2010 : 34)

d. Bentuk normal kedua (2NF/ *Second Normal Form*)

Pada tahapan ini dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer (*Primary Key*) tabel tersebut. Dalam hal ini semua tabel telah memenuhi bentuk normal form yang kedua, adapun kunci primer masing-masing tabel adalah sebagai berikut :

1. Untuk tabel anggota kunci primernya Kd\_Agt
2. Untuk tabel buku kunci primernya Kd\_Bk
3. Untuk tabel transaksi kunci primernya No\_Bk

e. Bentuk normak ketiga (3NF/ *Third Normal Form*)

Selain atribut pada tabel selain kunci primer atau kunci utama harus bergantung pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien. Biasanya penggunaan bentuk normal (normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang dihasilkan telah memiliki kualitas untuk membentuk sebuah database yang dapat diandalkan. Dan semua tabel pada normalisasi kedua telah memenuhi bentuk normal tahap ketiga. (Sri Mulanto ; 2010 : 33-35 )

### **II.5.3. Kamus Data**

Kamus data atau yang disebut juga dengan istilah *System Data Dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data harus mencerminkan keterangan yang jelas tentang data yang dicatatnya, untuk itu kamus data harus memuat hal-hal sebagai berikut :

1. Nama arus data, kamus data dibuat berdasarkan diagram arus data, maka nama dari arus data juga harus dicatat di kamus data, sehingga orang yang membaca diagram arus data dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di diagram arus data dapat langsung mencarinya dengan mudah.
2. Alias, dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya.

3. Bentuk data, yang dapat mengalir yaitu :
  - a. Dokumen dasar atau formulir
  - b. Dokumen hasil cetakan komputer
  - c. Laporan tercetak
  - d. Tampilan dilayar monitor
  - e. Variabel
  - f. Parameter
  - g. *Field*
4. Arus data, menunjukkan darimana data mengalir dan kemana data akan menuju. Keterangan arus data ini perlu dicatat di kamus data supaya memudahkan mencari arus data ini di diagram arus data.
5. Penjelasan, untuk lebih memperjelas lagi makna dari arus data di kamus data, maka bagian penjelasan dapat diisi dengan keterangan- keterangan tentang arus data tersebut.
6. Periode, ini menunjukkan kapan terjadinya arus data. Periode perlu dicatat di kamus data karena dapat digunakan untuk mengidentifikasi kapan input data harus dimasukkan ke sistem, kapan proses dari program harus dilakukan dan kapan laporan harus dihasilkan.
7. Volume, dalam hal ini yang perlu dicatat di kamus data adalah tentang volume rata-rata dan volume puncak dari arus data.  
  
Kamus data ini sangat membantu analisis sistem dalam mendefinisikan data.
8. Struktur data, menunjukkan arus data yang dicatat di kamus data terdiri dari item-item data apa saja. (Jogiyanto ; 2005: 725-728)

#### **II.5.4. SQL Server**

SQL adalah singkatan dari *Structured Query Language* dan merupakan bahasa yang dirancang khusus untuk berkomunikasi dengan *database*. SQL Server merupakan sebuah *Relational Database Management System* ( RDBMS) buatan Microsoft, yang dirancang untuk mendukung program dengan arsitektur *client/server*, dimana database diletakkan pada komputer pusat yang disebut *Server*, dan informasi digunakan bersama-sama oleh pengguna yang menjalankan program di dalam komputer yang disebut *client*. (Eko Hari Atmoko ;2013:2)

Tujuan dari dari SQL adalah untuk memungkinkan user melakukan aktivitas sebagai berikut :

- a. Membuat struktur relasi dan *basisdata*
- b. Melakukan operasi penyisipan, perubahan, dan penghapusan data dari tabel
- c. Melakukan *query* sederhana dan kompleks. (Indrajani ; 2008: 177)

#### **II.6. Unified Modelling Language (UML)**

##### **II.6.1. Pengertian UML**

Pemodelan (*modeling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model piranti lunak dapat dianalogikan seperti pembuatan blueprint pada pembangunan gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin kompleks sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik.

*Unified Modelling Language* (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, dan sebagainya. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek.

Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group. Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek. (Sri Dharwiyanti; 2006 : 2)

## **II.6.2. Pengertian Diagram UML**

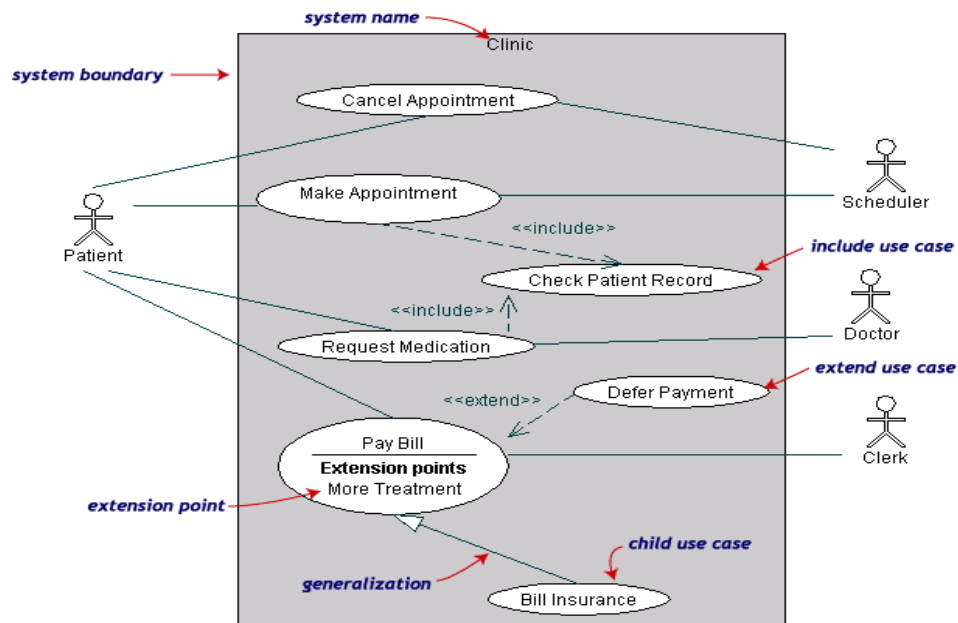
### **1. Use Case Diagram**

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan

*behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Contoh *use case diagram* :



**Gambar II.1. Contoh Use Case Diagram**

(Sumber : Sri Dharwiyanti ; 2006 : 5)

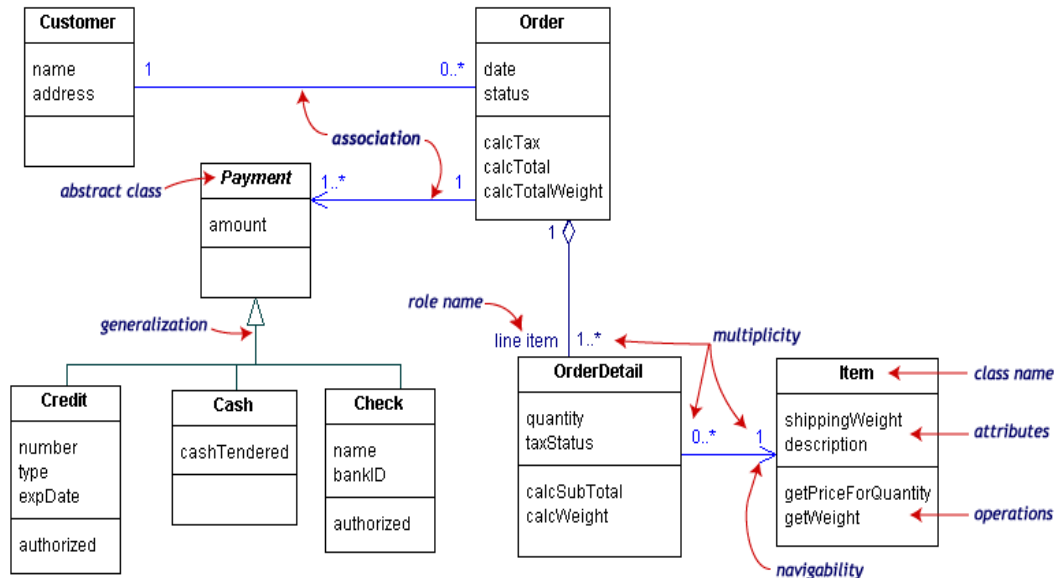
## 2. Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda



**Gambar II.2. Contoh Class Diagram**

(Sumber : Sri Dharwiyanti 2006 : 6)

### 3. Statechart Diagram

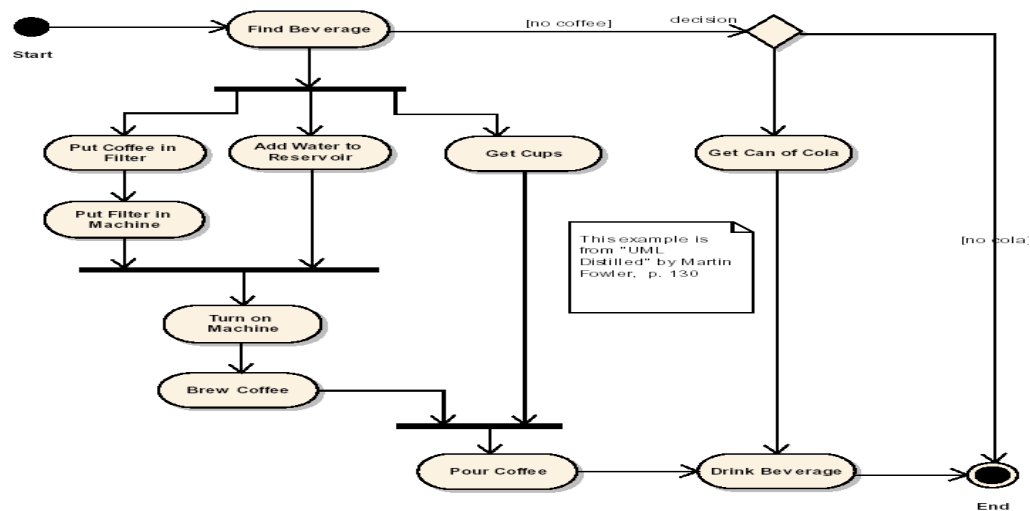
*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

### 4. Activity Diagram

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state*

sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Contoh *Activity Diagram* tanpa *Swimlane*:



**Gambar II.3. Contoh *Activity Diagram* tanpa *Swimlane***

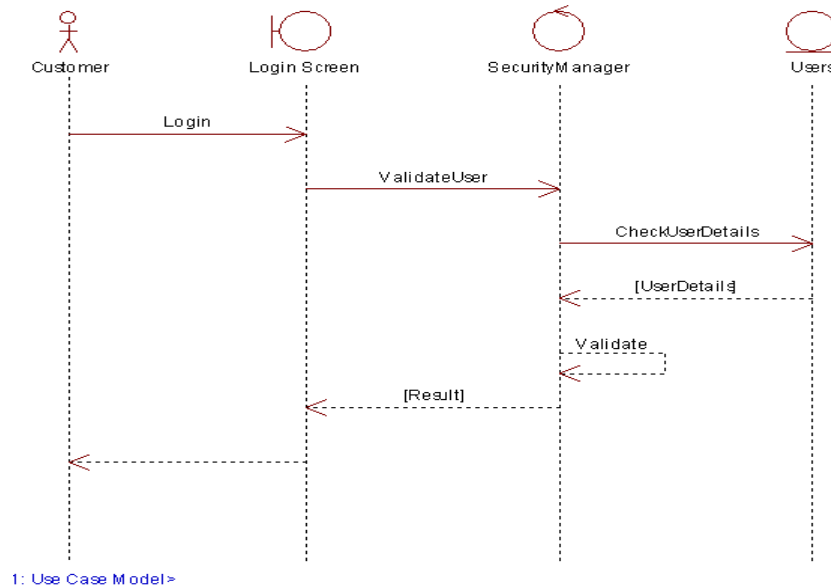
(Sumber : Sri Dharwiyanti 2006 : 8)

## 5. *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

Contoh *sequence diagram* :



**Gambar II.4. Contoh *Sequence Diagram***

**( Sumber : Sri Dharwiyanti ; 2006 : 9 )**

## 6. *Collaboration Diagram*

*Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.

## 7. *Component Diagram*

*Component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. *Component diagram* fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

## 8. *Deployment Diagram*

*Deployment diagram* menggambarkan *detail* bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisik. ( Sri Dharwiyanti ; 2006 : 9 )

## II.7. Visual Basic

*Visual Basic* adalah bahasa pemrograman generasi ke tiga dari *Microsoft* dengan IDE (*Integrated Develovement Environment*) atau pemrograman pengembangan terpadu. *Visual Basic* di rancang untuk mudah digunakan baik oleh *programer* ataupun pemula.

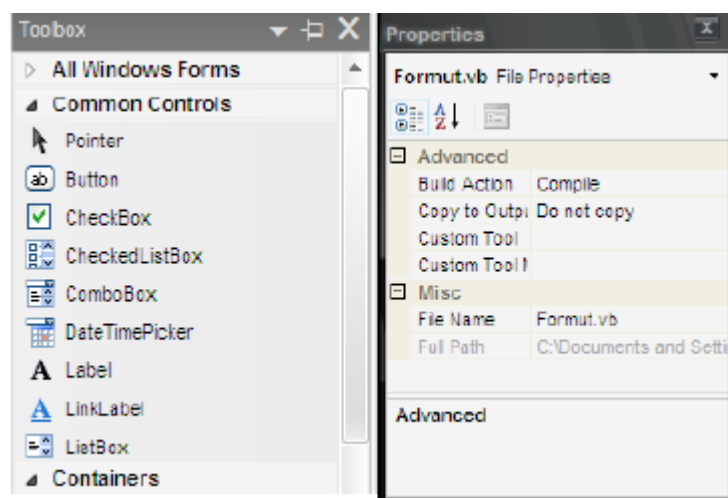
Selain digunakan untuk membuat program sederhana *Visual Basic* juga digunakan untuk membangun program kompleks, dengan menggunakan bahasa *basic* serta memungkinkan pengembangan program dengan cepat dengan antarmuka grafis (GUI). (Eko Hari Atmoko ; 2013 : 1)



**Gambar II.5. Start Page Microsoft Visual Studio.Net**

( Sumber: Fajar Rahadian; Jurnal Pendidikan Dompot Dhuafa Edisi 1  
2011: 3)

*Toolbox* digunakan untuk pemilihan kontrol–kontrol yang akan digunakan pada program yang akan dirancang. Kontrol ini merupakan kontrol standar yang digunakan oleh aplikasi *Windows*, dan kontrol–kontrol tambahan yang disebut *ActiveX*. Kontrol yang ada pada jendela ini dapat ditambah dan dikurangi sesuai kebutuhan. Jendela *Properties* merupakan jendela yang digunakan untuk mengatur properti sebuah objek. Jendela *Properties* ini terbagi dalam dua bagian yaitu *Alphabetic* dan *Catagirozed*. Perbedaan dari keduanya hanyalah cara menampilkan *properties* dalam sebuah objek. Pada bagian *Alphabetic*, properti diatur berdasarkan urutan abjad, sedangkan di bagian *Catagorized*, properti diatur dalam kelompok-kelompok kategori. (Fajar Rahadian; Jurnal Pendidikan Dompot Dhuafa Edisi 1 2011: 3)



**Gambar II.6. Tampilan *Toolbox* dan Jendela *Properties* Microsoft Visual Studio .net**

**( Sumber: Fajar Rahadian; Jurnal Pendidikan Dompot Dhuafa Edisi 1 2011: 3)**