

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Istilah sistem bukanlah hal yang asing bagi kebanyakan orang. Dari segi Etimologi kata sistem sebenarnya berasal dari bahasa Yunani yaitu "*Systema*" didalam bahasa Inggris juga dikenal dengan nama "System" yang dapat diartikan yaitu sehimpunan bagian atau komponen lain yang saling berhubungan secara teratur dan merupakan satu keseluruhan yang tidak terpisahkan. Menurut filsuf Stoa, bahwa sistem adalah gabungan dari keseluruhan langit dan bumi yang bekerja bersama-sama, sehingga dapat kita lihat bahwa sistem terdiri dari unsur-unsur yang bekerja sama membentuk satu keseluruhan dan apabila salah satu dari keseluruhan tersebut hilang atau tidak berfungsi, maka gabungan keseluruhan tidak lagi dapat disebut suatu sistem. Berikut ini adalah definisi kata sistem menurut beberapa ahli.

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Jogiyanto HM; 2005:1).

Pendekatan sistem merupakan jaringan kerja dari prosedur lebih menekankan urutan operasi di dalam sistem. Prosedur adalah urutan tulis menulis, biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi (Jogiyanto HM; 2005:1).

Sistem mempunyai maksud tertentu. Ada yang menyebutkan maksud dari sistem adalah untuk mencapai suatu tujuan (*goal*) dan ada yang menyebutkan untuk mencapai suatu sasaran (*objectives*). *Goal* biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit. Bila merupakan suatu sistem utama, seperti misalnya sistem bisnis, maka istilah *goal* lebih tepat di terapkan. Untuk sistem akuntansi atau sistem-sistem yang lainnya yang merupakan bagian atau subsistem dari sistem bisnis, maka istilah *objectives* yang lebih tepat. Jadi tergantung dari ruang lingkup dari mana memandang sistem. Seringkali tujuan dan sasaran digunakan bergantian dan tidak dibedakan (Jogiyanto HM; 2005:3).

### **II.1.1. Karakteristik Sistem**

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya (Hanif Al Fatta;2007:5). Berikut ini adalah karakteristik sistem :

1. Batasan (*boundary*)

Penggambaran dari suatu elemen atau unsure mana yang termasuk di dalam sistem dan mana yang di luar sistem.

2. Lingkungan (*environment*)

Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.

3. Masukan (*input*)

Sumber daya (data, bahan baku, peralatan, energi)

#### 4. Keluaran (*output*)

Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.

#### 5. Komponen (*component*)

Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.

#### 6. Penghubung (*interface*)

Tempat di mana komponen atau sistem dan lingkungan bertemu atau berinteraksi.

#### 7. Penyimpanan (*storage*)

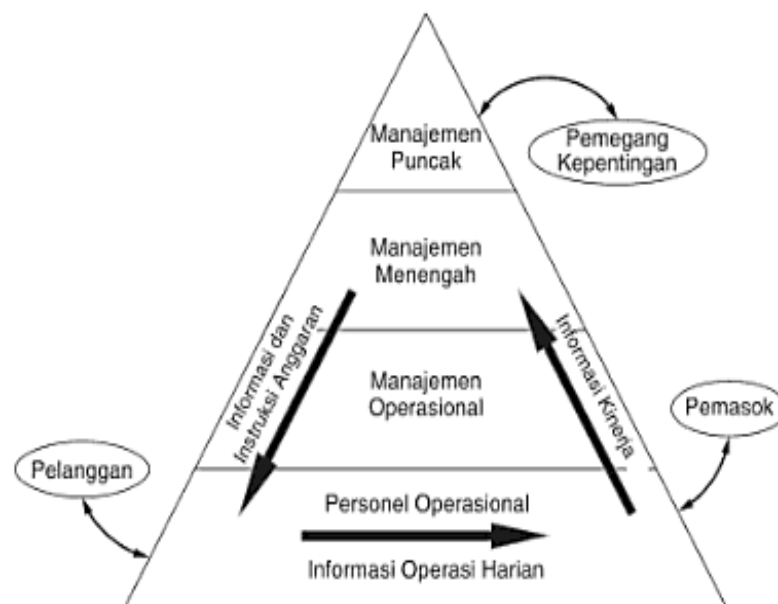
Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama (Hanif Al Fatta; 2007:6).

## **II.2. Informasi**

Untuk memahami pengertian sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.

### II.2.1. Sistem Informasi Akuntansi

Studi mengenai sistem informasi akuntansi dimulai dengan mengenali informasi sebagai sumber daya bisnis. Sama dengan sumber daya bisnis lainnya bahan baku, modal, dan tenaga kerja informasi sangat penting bagi perusahaan modern untuk bertahan hidup. Setiap hari banyak sekali arus informasi ke para pengambil keputusan serta berbagai pengguna lainnya untuk memenuhi berbagai kebutuhan internal. Selain itu, arus informasi juga keluar dari perusahaan ke pengguna luar, seperti pelanggan, pemasok, dan pemegang kepentingan (*stakeholder*). Gambar II.1. menyajikan gambaran umum dari berbagai arus informasi (*information flow*) internal dan eksternal piramida tersebut (James A.Hall;2007:4).



**Gambar II.1. Sistem Informasi Akuntansi**  
(Sumber : James A. Hall; 2007:5)

Berbagai aktivitas ini terdiri atas kegiatan perusahaan yang berorientasi pada produk, seperti produksi, penjualan, dan distribusi. Di atas tingkat dasarnya,

perusahaan dibagi menjadi tiga lapis manajemen yaitu manajemen operasional, manajemen menengah, dan manajemen puncak. Manajemen operasional secara langsung bertanggung jawab atas aktivitas perencanaan dan koordinasi jangka pendek yang dibutuhkan untuk mencapai tujuan perusahaan. Manajemen puncak bertanggung jawab atas perencanaan jangka waktu yang lebih panjang dan menetapkan berbagai tujuan perusahaan. Tiap orang dalam perusahaan, mulai dari operasional perusahaan hingga manajemen puncak, membutuhkan informasi untuk menyelesaikan pekerjaan.

### **II.3. Jurnal (*Journal*)**

Jurnal adalah buku harian tempat mencatat semua transaksi secara sistematis dan kronologis berdasarkan bukti-bukti transaksi. Setiap transaksi yang terjadi dalam suatu perusahaan harus disertai dengan bukti transaksi. Bukti transaksi ini berguna sebagai sumber pencatatan ke dalam jurnal dan menjadi alat pertanggungjawaban kepada atasan bahwa transaksi sudah dilakukan. Setiap transaksi dapat dinyatakan dalam debit dan kredit terhadap akun-akun yang ada di buku besar. Nama akun dan jumlah yang di debit, dicantumkan pada sisi sebelah kiri, kemudian di bawah agak ke kanan di tulis nama akun dan jumlah yang di kredit. Setiap ayat jurnal terdiri paling tidak satu akun yang di debit dan satu akun yang di kredit. Jumlah debit harus sama dengan jumlah kredit. Pada umumnya, terdapat formulir khusus guna pencatatan. Formulir ini di sebut buku harian atau jurnal. Dalam jurnal setiap bukti transaksi di catat secara kronologis (Soemarso S.R; 2009:94).

#### **II.4. Buku Besar**

Transaksi-transaksi yang terjadi selama satu periode berpengaruh terhadap penambahan atau pengurangan berbagai jenis aktiva, kewajiban, modal, pendapatan dan beban. Untuk memperoleh informasi pada saat di perlukan maka perlu ada catatan tersendiri. Pencatatan khusus yang di gunakan untuk mencatat dan menggolongkan transaksi sejenis di sebut akun. Kumpulan akun yang saling berhubungan dan merupakan satu kesatuan , disebut buku besar(*ledger*). Formulir-formulir ini dapat di buat dalam bentuk kartu dan lembaran kertas (Soemarso S.R; 2009:64).

#### **II.5. Pengertian Akuntansi**

*American Accounting Association* mendefinisikan akuntansi sebagai merupakan proses mengidentifikasi, mengukur, dan melaporkan informasi ekonomi, untuk memungkinkan adanya penilaian dan keputusan yang jelas dan tegas bagi mereka yang menggunakan informasi tersebut.

Defenisi ini mengandung 2 pengertian, yakni:

##### **1. Kegiatan Akuntansi**

Bahwa akuntansi merupakan proses yang terdiri dari identifikasi, pengukuran dan pelaporan informasi ekonomi.

##### **2. Kegunaan Akuntansi**

Bahwa informasi ekonomi yang dihasilkan oleh akuntansi diharapkan berguna dalam penilaian dan pengambilan keputusan mengenai kesatuan usaha yang bersangkutan.

Akuntansi (*accounting*) adalah merupakan proses identifikasi, pencatatan dan komunikasi terhadap transaksi ekonomi dari suatu entitas (perusahaan). Jadi secara umum terdapat tiga aktivitas dalam akuntansi yaitu:

1. Aktivitas identifikasi (*identifying*), dalam aktivitas ini akan dilakukan identifikasi terhadap transaksi yang terjadi dalam suatu entitas (perusahaan).
2. Aktivitas pencatatan (*recording*), dalam aktivitas ini semua transaksi ekonomi atau transaksi keuangan yang telah diidentifikasi pada tahap pertama akan dicatat secara kronologis dan sistematis dengan ukuran nilai moneter tertentu, dan
3. Aktivitas komunikasi (*communication*), dalam aktivitas ini akan dilakukan pelaporan dan distribusi terhadap informasi akuntansi yang berupa laporan keuangan kepada para pemakai laporan keuangan atau pihak yang berkepentingan. Pemakai laporan keuangan terdiri atas pemakai di dalam perusahaan (*internal user*) misalnya manajemen dan karyawan, serta pemakai di luar perusahaan (*external user*), misalnya kreditur, investor dan fiskus.

Jadi secara teknis, akuntansi merupakan kumpulan prosedur-prosedur untuk mencatat, mengklasifikasikan, mengikhtisarkan dan melaporkan dalam bentuk laporan keuangan. Semua transaksi yang telah terjadi selama periode akuntansi akan diikhtisarkan dalam buku besar dan laporan keuangan.

Tujuan utama akuntansi adalah menyajikan informasi ekonomi (*economic information*) dari suatu kesatuan ekonomi (*economic entity*) kepada pihak-pihak yang berkepentingan. Dalam buku ini yang di maksud dengan kesatuan ekonomi adalah badan usaha (*business enterprice*). Informasi ekonomi yang dihasilkan

oleh akuntansi berguna bagi pihak-pihak di dalam perusahaan itu sendiri maupun di pihak-pihak di luar perusahaan.

Untuk menghasilkan informasi ekonomi, perusahaan perlu menciptakan suatu metode pencatatan, penggolongan, analisis, dan pengendalian transaksi serta kegiatan-kegiatan keuangan, kemudian melaporkan hasilnya. Kegiatan akuntansi meliputi:

1. Pengidentifikasian dan pengukuran data yang relevan untuk suatu pengambilan keputusan.
2. Pemrosesan data yang bersangkutan kemudian pelaporan informasi yang dihasilkan.
3. Pengkomunikasian informasi kepada pemakai laporan.

Kegiatan-kegiatan tadi perlu dirangkaikan dalam suatu sistem yang disebut sistem akuntansi (Soemarso S.R. ; 2009 : 3-4).

## II.6. Persamaan Akuntansi

Secara sepintas disebutkan bahwaselalu ada kesamaan antara kekayaan dan sumber pembelanjaan. Kesamaan itu biasanya dinyatakan dalam suatu persamaan yang di sebut persamaan akuntansi (*accounting equation*) dapat dilihat pada gambar II.2.

$$\text{KEKAYAAN} = \text{SUMBER PEMBELANJAAN}$$

**Gambar II.2. Akuntansi Suatu Pengantar**  
(Sumber : Soemarso S.R ; 2009 : 43)

Biasanya kekayaan yang dimiliki oleh perusahaan disebut aktiva atau harta. Sumber pembelanjaan dapat dibagi menjadi dua, yakni dari kreditur dan pemilik. Sumber pembelanjaan dari kreditur disebut kewajiban dan sumber pembelanjaan dari pemilik disebut modal.

Persamaan akuntansi menunjukkan posisi keuangan perusahaan. Karena transaksi usaha akan mempengaruhi posisi keuangan perusahaan, maka setiap transaksi usaha dapat dinyatakan dalam bentuk efeknya terhadap ketiga unsur persamaan akuntansi (Soemarso S.R; 2009:44).

## **II.7. Proses Akuntansi**

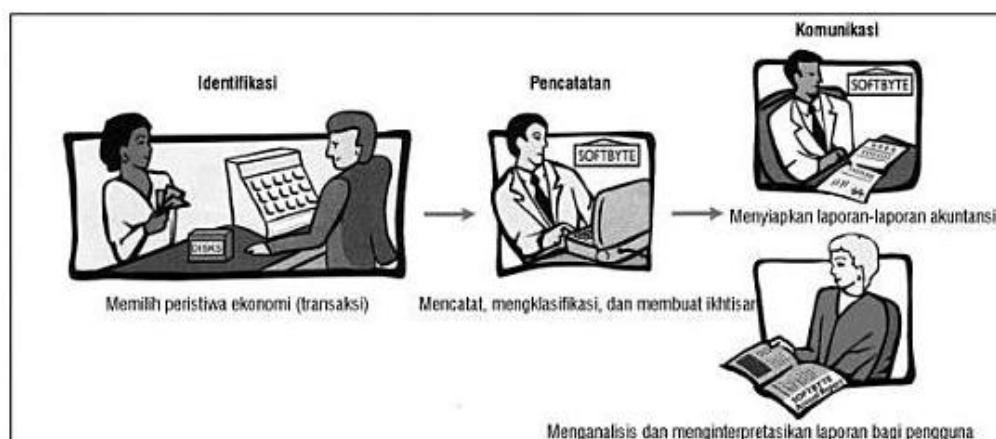
Akuntansi (*accounting*) adalah suatu sistem informasi yang mengidentifikasi, mencatat dan mengkomunikasikan peristiwa-peristiwa ekonomi dari suatu organisasi kepada para pengguna yang berkepentingan. Marilah kita lihat secara lebih dekat ketiga aktivitas tersebut.

1. Mengidentifikasi peristiwa-peristiwa ekonomi akan melibatkan pemilihan aktivitas-aktivitas ekonomi yang relevan bagi suatu organisasi tertentu. Penjualan keripik oleh PepsiCo, penyediaan jasa oleh Sprint, pembayaran upah oleh Ford Motor Company, dan penerimaan uang tiket dan penyiaran serta pembayaran berbagai biaya oleh tim-tim liga olahraga terkenal adalah contoh dari peristiwa-peristiwa ekonomi.
2. Setelah teridentifikasi, peristiwa-peristiwa ekonomi tersebut kemudian dicatat untuk menjadi alur aktivitas keuangan perusahaan. Pencatatan terdiri atas pembuatan jurnal peristiwa-peristiwa secara sistematis dan kronologis,

yang diukur dalam satuan mata uang dollar dan sen. Di dalam pencatatan, peristiwa-peristiwa ekonomi juga akan diklasifikasikan dan dibuat ikhtisarnya.

3. Aktifitas pengidentifikasian dan pencatatan tidak akan banyak memberikan manfaat, kecuali jika informasi tersebut dikomunikasikan kepada pengguna-pengguna yang berkepentingan. Informasi keuangan akan disampaikan melalui laporan-laporan akuntansi, yang umumnya keuangan yang dilaporkan menjadi bermakna, para akuntan melaporkan data yang tercatat dalam cara yang terstandardisasi. Informasi yang berasal dari transaksi-transaksi yang serupa akan diakumulasikan dan dijumlahkan. Sebagai contoh, seluruh transaksi penjualan PepsiCo akan diakumulasikan selama jangka waktu tertentu dan dilaporkan sebagai satu jumlah dalam laporan keuangan perusahaan. Data seperti itu akan dilaporkan secara agregat. Dengan menyajikan data yang dicatat secara agregat, proses akuntansi akan mampu menyederhanakan banyak transaksi dan membuat serangkaian aktivitas menjadi mudah dipahami dan bermakna (Jerry J.Weygandt, Donald E.Keiso, Paul D.Kimmel; 2007:5).

Salah satu unsur vital dalam mengomunikasikan peristiwa-peristiwa ekonomi adalah kemampuan akuntan untuk menganalisis dan menginterpretasikan informasi yang telah dilaporkan. Analisis meliputi penggunaan rasio, persentase, grafik, dan diagram untuk menyoroti tren-tren dan hubungan keuangan yang signifikan. Proses akuntansi dapat dirangkum menjadi seperti yang disajikan dalam Gambar II.3.



**Gambar II.3. Proses Akuntansi**

(Sumber : Jerry J.Weygandt, Donald E.Keiso, Paul D.Kimm; 2007 : 6)

Akuntansi sebaiknya mempertimbangkan kebutuhan dari para pengguna informasi keuangan. Oleh sebab itu, Anda hendaknya mengetahui siapa pengguna-pengguna tersebut dan kebutuhan-kebutuhan mereka akan informasi (Jerry J.Weygandt, Donald E.Keiso, Paul D.Kimmel ; 2007:6).

## II.8. Arti Debet dan Kredit

Salah satu istilah yang sering mengemuka dalam mempelajari akuntansi adalah debet (atau debit) dan kredit. Debet dan kredit sering dijumpai dalam buku akuntansi manual pada dalam bahasan jurnal, buku besar dan penyusunan neraca lajur (dengan excel tidak perlu ada). Dapat diterangkan bahwa sisi sebelah kiri suatu akun disebut sisi debit, sedang sisi sebelah kanan disebut sisi kredit. Akun adalah formulir untuk mencatat penambahan dan pengurangan yang terjadi dalam pos yang bersangkutan. Oleh karna adanya ketentuan debit dan kredit maka penambahan atau pengurangan yang terjadi dalam akun dapat dinyatakan dalam

debit atau kredit. Aturan untuk mendebit atau mengkredit suatu akun pada umumnya dapat dijelaskan pada Gambar II.4.

| AKTIVA                        |                                 | = | KEWAJIBAN + MODAL                 |                                   |
|-------------------------------|---------------------------------|---|-----------------------------------|-----------------------------------|
| Akun-akun aktiva              |                                 |   | Akun-akun utang                   |                                   |
| Debit untuk<br>penambahan (+) | Kredit untuk<br>pengurangan (-) |   | Debit untuk<br>pengurangan<br>(-) | Kredit untuk<br>penambahan<br>(+) |
|                               |                                 |   | Akun-akun modal                   |                                   |
|                               |                                 |   | Debit untuk<br>pengurangan<br>(-) | Kredit untuk<br>penambahan<br>(+) |

**Gambar II.4. Akun Neraca**

(Sumber : Soemarso S.R ; 2009 : 70)

Apabila suatu pos aktiva bertambah, maka akun yang bersangkutan didebit, sedangkan apabila suatu transaksi mengakibatkan pas berkurang, maka akunnya dikredit. Sebaliknya, untuk pos-pos kewajiban dan modal, kredit berarti penambahan dan debit berarti pengurangan. Dari uraian ini dapat disimpulkan bahwa mendebit suatu akun dapat berarti penambahan atau pengurangan, tergantung pada jenis akunnya. Demikian juga halnya dengan mengkredit suatu akun (Soemarso S.R; 2009:71).

## II.9. Kamus Data

Kamus data harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan maka Kamus data harus memuat hal-hal berikut ini :

### 1. Nama Arus data

Karena kamus data dibuat berdasarkan arus data yang mengalir di DAD, maka nama dari arus data juga harus dicatat di KD, sehingga mereka yang membaca DAD dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di DAD dapat langsung mencari dengan mudah di KD.

### 2. Alias

Alias atau nama lain dari data 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

Bentuk data yang mengalir adalah :

- Dokumen dasar atau formulir
- Document hasil cetakan komputer
- Laporan cetak
- Tampilan dilayar monitor
- Variable
- Parameter
- Field

#### 4. Arus Data

Arus data menunjukkan dari mana data mengalir dan kemana data akan menuju.

#### 5. Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di Kamus data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

#### 6. Periode

Periode ini menunjukkan kapan terjadinya arus data ini

#### 7. Volume

Volume yang perlu dicatat di KD adalah tentang volume rata-rata dan volume puncak tentang arus data.

#### 8. Struktur Data

Struktur data menunjukkan arus data yang dicatat di KD terdiri dari item-item data apa saja.

### **II.10. Normalisasi**

Normalisasi adalah proses penyusunan tabel-tabel yang tidak redundan (*double*), yang dapat menyebabkan anomaly pada saat operasi manipulasi data, seperti tambah, ubah dan hapus. Sebuah tabel dapat dikategorikan efisien atau normal jika memenuhi tiga kriteria berikut :

1. Jika ada dekomposisi tabel maka dekomposisi itu harus terjamin aman (*lossless join decomposition*). Dekomposisi adalah proses pemisahan satu tabel menjadi dua atau lebih tabel baru.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*dependency preservation*).
3. Tidak melanggar *boyce codd normal form* (BCNF).

Ketiga kategori di atas merupakan kriteria minimal untuk mendapatkan predikat efisien atau normal bagi sebuah tabel. Teori normalisasi dibangun menurut konsep level normalisasi. Level normalisasi atau sering disebut bentuk normal suatu relasi dijelaskan berdasarkan kriteria tertentu pada bentuk normal (Kusrini, Andri Koniyo; 2007:102).

Secara berturut masing-masing level normal tersebut akan dijelaskan seperti dibawah ini.

### **II.10.1. Normalisasi Bentuk Pertama**

Bentuk normal tahap pertama terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (*multivalued attribute*) atau lebih dari satu atribut dengan nilai domain yang sama. Bentuk normalisasi pertama dapat dilihat pada tabel di bawah ini (Kusrini, Andri Koniyo; 2007:104).

## a. Tabel Pembelian

| NoNotaBeli | Tanggal    | Kd_Pemasok | Kd_BahanBaku | HargaBeli | JumlahBarang | .. |
|------------|------------|------------|--------------|-----------|--------------|----|
| NB01       | 28/02/2007 | TK01       | A01          | 27500     | 10           | .. |
| NB01       | 28/02/2007 | TK01       | A02          | 50000     | 10           | .. |
| NB02       | 28/02/2007 | UD01       | B03          | 10500     | 50           | .. |

| .. | DisconBeli | CaraBayar | JthTempo   | UangMuka | JmlHutang |
|----|------------|-----------|------------|----------|-----------|
| .. | 20000      | Tunai     | -          | -        | -         |
| .. | 20000      | Tunai     | -          | -        | -         |
| .. | 30000      | Kredit    | 28/03/2007 | 200000   | 600000    |

**Tabel II.1. Tabel Pembelian Normal 1**  
(Sumber : Kusriani, Andri Koniyo; 2007:104)

## b. Tabel Penjualan

| NoNotaJual | TglJual    | TglOder    | TglSelesai | Kd_Pelanggan | NamaBarang/adi | .. |
|------------|------------|------------|------------|--------------|----------------|----|
| NJ01       | 21/03/2007 | 14/03/2007 | 20/03/2007 | P01          | Pintu Pagar    | .. |
| NJ01       | 22/03/2007 | 14/03/2007 | 20/03/2007 | P01          | Pintu Pagar    | .. |
| NJ01       | 23/03/2007 | 14/03/2007 | 20/03/2007 | P01          | Pintu Pagar    | .. |
| NJ02       | 22/03/2007 | 15/03/2007 | 21/03/2007 | P02          | Pintu Dorong   | .. |
| NJ02       | 22/03/2007 | 15/03/2007 | 21/03/2007 | P02          | Pintu Dorong   | .. |

| .. | Kd_BahanBaku | JmlBarang | UpahTenagaKerja | BiayaOverhead | UangMuka | .. |
|----|--------------|-----------|-----------------|---------------|----------|----|
| .. | A01          | 5         | 1000000         | 250000        | 100000   | .. |
| .. | A02          | 4         | 1000000         | 250000        | 100000   | .. |
| .. | B03          | 3         | 1000000         | 250000        | 100000   | .. |
| .. | A02          | 2         | 500000          | 350000        | 100000   | .. |
| .. | B03          | 2         | 500000          | 350000        | 100000   | .. |

| .. | Disjual | CaraBayar | JumlahBiaya | JthTempo   | SisaBayar | JmlPiutang |
|----|---------|-----------|-------------|------------|-----------|------------|
| .. | 20000   | Tunai     | 1625000     | -          | 625000    | -          |
| .. | 20000   | Tunai     | 1625000     | -          | 625000    | -          |
| .. | 20000   | Tunai     | 1625000     | -          | 625000    | -          |
| .. | 30000   | Kredit    | 2000000     | 28/03/2007 | 1500000   | 1500000    |
| .. | 30000   | Kredit    | 2000000     | 28/03/2007 | 1500000   | 1500000    |

**Tabel II.2. Tabel Penjualan Normal 1**  
(Sumber : Kusriani, Andri Koniyo; 2007:105)

## II.10.2. Normalisasi Bentuk Kedua

Bentuk normal tahap kedua terpenuhi jika normalisasi tahap pertama terpenuhi dan semua atribut tidak termasuk dalam kunci primer secara utuh. Dengan demikian untuk membentuk normal kedua haruslah sudah ditentukan

kunci-kunci fieldnya. Kunci fieldnya harus unik dan dapat mewakili atribut lain yang menjadi anggotanya. Sebuah tabel dikatakan tidak memenuhi normalisasi bentuk kedua jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari *primary key*). Bentuk normalisasi kedua dapat dilihat pada tabel di awah ini (Kusrini, Andri Koniyo; 2007:106).

| <b>Pemasok</b> |                | <b>BahanBaku</b> |                    | <b>Pelanggan</b> |                 |
|----------------|----------------|------------------|--------------------|------------------|-----------------|
| *              | Kd_Pemasok     | *                | Kd_BahanBaku       | *                | Kd_Pelanggan    |
|                | NamaPemasok    |                  | NamaBahanBaku      |                  | NamaPelanggan   |
|                | C_Person       |                  | Ukuran             |                  | C_Person        |
|                | AlamatPemasok  |                  | Satuan             |                  | AlamatPelanggan |
|                | TeleponPemasok |                  | HargaJualBahanBaku |                  | TeleponPemasok  |
|                |                |                  | StockAwal          |                  |                 |
|                |                |                  | StockMin           |                  |                 |
|                |                |                  | StockMax           |                  |                 |

| <b>Pembelian</b> |             | <b>Detail Pembelian</b> |                    | <b>Penjualan</b> |                 |
|------------------|-------------|-------------------------|--------------------|------------------|-----------------|
| *                | NoNotaBeli  | *                       | NoNotaBeli         | *                | NoNotaJual      |
|                  | Tanggal     | *                       | Kd_BahanBaku       |                  | Kd_Pelanggan    |
|                  | Kd_Pemasok  |                         | HargaBeliBahanBaku |                  | NamaBarangJadi  |
|                  | CaraBayar   |                         | JumlahBarang       |                  | TglOrder        |
|                  | JthTempo    |                         |                    |                  | TglSelesai      |
|                  | DiscontBeli |                         |                    |                  | TglJual         |
|                  | UangMuka    |                         |                    |                  | UpahTenagaKerja |
|                  |             |                         |                    |                  | BiayaOverhead   |
|                  |             |                         |                    |                  | DiscontJual     |
|                  |             |                         |                    |                  | UangMuka        |
|                  |             |                         |                    |                  | CaraBayar       |
|                  |             |                         |                    |                  | JthTempo        |

**Tabel II.3. Normalisasi Bentuk Kedua**  
(Sumber : Kusrini, Andri Koniyo; 2007:107)

### II.10.3. Normalisasi Bentuk Ketiga

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak mempunyai hubungan

transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* secara menyeluruh. Bentuk normalisasi ketiga dapat dilihat pada tabel di bawah ini (Kusrini, Andri Koniyo; 2007:108).

| <b>Pemasok</b> |  | <b>BahanBaku</b>   |  | <b>Pelanggan</b> |  |
|----------------|--|--------------------|--|------------------|--|
| * Kd_Pemasok   |  | * Kd_BahanBaku     |  | * Kd_Pelanggan   |  |
| NamaPemasok    |  | NamaBahanBaku      |  | NamaPelanggan    |  |
| C_Person       |  | Ukuran             |  | C_Person         |  |
| AlamatPemasok  |  | Satuan             |  | AlamatPelanggan  |  |
| TeleponPemasok |  | HargaJualBahanBaku |  | TeleponPemasok   |  |
|                |  | StockAwal          |  |                  |  |
|                |  | StockMin           |  |                  |  |
|                |  | StockMax           |  |                  |  |

| <b>Pembelian</b> |  | <b>Detail_Pembelian</b> |  | <b>Penjualan</b> |  |
|------------------|--|-------------------------|--|------------------|--|
| * NoNotaBeli     |  | * NoNotaBeli            |  | * NoNotaJual     |  |
| Tanggal          |  | * Kd_BahanBaku          |  | Kd_Pelanggan     |  |
| Kd_Pemasok       |  | HargaBeliBahanBaku      |  | NamaBarangJadi   |  |
| CaraBayar        |  | JumlahBarang            |  | TglOrder         |  |
| JthTempo         |  |                         |  | TglSelesai       |  |
| DiscontBeli      |  |                         |  | TglJual          |  |
| UangMuka         |  |                         |  | UpahTenagaKerja  |  |

|  |               |
|--|---------------|
|  | BiayaOverhead |
|  | DiscontJual   |
|  | UangMuka      |
|  | CaraBayar     |
|  | JthTempo      |

**Tabel II.4. Normalisasi Bentuk KeTiga**  
(Sumber : Kusrini, Andri Koniyo ; 2007 : 108)

## II.11. UML

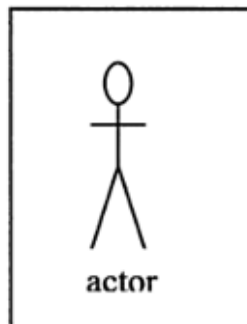
Pemodelan perangkat lunak bekerja dengan cara ygng cukup serupa layaknya seorang arsitek atau insinyur teknik sipil yang akan membuat sebuah bangunan, ia biasanya membuat denah-denang yang menggambarkan bentuk jadi

dari bangunan. Sebagai seorang perancang sistem perangkat lunak juga bertindak dengan cara yang serupa, hanya saja yang dirancang bukan bangunan, melainkan sistem perangkat lunak. Menggambarkan komponen sistem perangkat lunak dalam bentuk-bentuk geometri tertentu dalam aplikasi.

UML adalah sekumpulan symbol dan diagram untuk memodelkan software. Dengan menggunakan UML, desain software dapat diwujudkan dalam bentuk symbol dan diagram. Desain dalam bentuk symbol dan diagram, kemudian dapat diterjemahkan menjadi kode program. Telah tersedia tools yang dapat membuat kode program diagram UML dapat menggunakan bahasa pemrograman apa saja dengan syarat bahasa pemrograman tersebut harus mendukung pemrograman berorientasi obyek (Julius Hermawan ; 2007 : 14).

#### **II.11.1. Actor**

Actor adalah segala sesuatu yang berinteraksi dengan system aplikasi komputer. Jadi actor ini bisa berupa orang perangkat keras, atau mungkin juga obyek lain dalam system yang sama. Biasanya yang dilakukan oleh actor adalah memberikan informasi pada system dan atau memerintahkan system untuk melakukan sesuatu (Julius Hermawan, 2007;14). *Actor* dapat dilihat pada Gambar II.5.

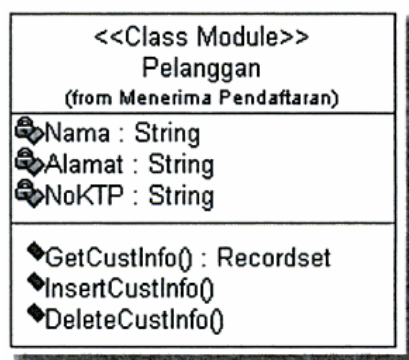


**Gambar II.5. Actor**

(Sumber : Julius Hermawan; 2007:14)

### II.11.2. Class

Class merupakan pembentuk utama dari sistem berorientasi obyek karena class menunjukkan kumpulan obyek yang memiliki atribut dan operasi yang sama. Class digunakan untuk mengimplementasikan interface (Julius Hermawan, 2007;14). *Class* dapat dilihat pada Gambar II.6.



**Gambar II.6. Class**

(Sumber : Julius Hermawan;2007:14).

*Class* digunakan untuk mengabstraksikan elemen-elemen dari system yang sedang dibangun. *Class* bisa untuk merepresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata.

Notasi *class* berbentuk persegi panjang berisi 3 bagian persegi paling atas untuk nama class, persegi panjang paling bawah untuk operasi, dan persegi di tengah untuk atribut. Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas merepresentasikan informasi yang disimpan di dalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh obyek (Julius Hermawan, 2007;14).

### II.11.3. *Interface*

*Interface* digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya interaction ini dilengkapi juga dengan teks bernama operation signature yang tersusun dari nama operasi parameter yang dikirim dan tipe parameter yang dikembalikan (Julius Hermawan, 2007;18). Simbol daripada *interface* dapat dilihat pada Gambar II.7.



**Gambar II.7. *Interface***  
(Sumber : Julius Hermawan;2007:18).

### II.11.4. *Use Case*

*Use case* menjelaskan urutan kegiatan yang dilakukan actor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namun use case hanya menjelaskan apa yang dilakukan oleh actor dan sistem, bukan

bagaimana actor dan system melakukan kegiatan tersebut. *Use case* dapat dilihat pada Gambar II.8.



**Gambar II.8. Simbol *Use Case***  
(Sumber : Julius Hermawan; 2007 : 16)

Di dalam *use case* terdapat teks untuk menjelaskan urutan kegiatan yang disebut *Use case specification*. *Use case specification* terdiri dari :

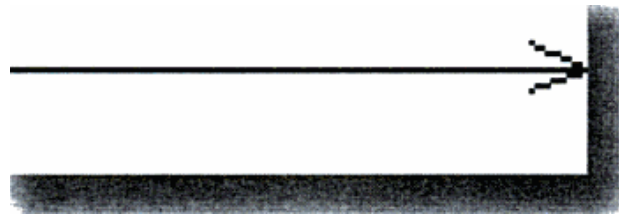
1. Nama *Use case*, mencantumkan nama dari *use case* yang bersangkutan. Sebaiknya diawali dengan kata kerja untuk menunjukkan suatu aktivitas.
2. Deskripsi singkat (*Brief Description*), menjelaskan secara singkat dalam 1 atau 2 kalimat tentang tujuan dari *use case* ini.
3. Aliran normal (*Basic Flow*), ini adalah jantung dari *use case*. Menjelaskan interaksi antar actor dan system dalam kondisi normal, yaitu segala sesuatu berjalan dengan lancar, tiada halangan atau hambatan dalam mencapai tujuan dari *use case*.
4. Aliran alternative (*Alternate Flow*), merupakan pelengkap dari basic flow karena tidak ada yang sempurna dalam setiap kali *use case* berlangsung. Di dalam alternate flow ini dijelaskan apa yang akan terjadi bila alternate flow ini dijelaskan apa yang akan terjadi bila suatu halangan atau hambatan terjadi sewaktu *use case* berlangsung. Ini terutama berhubungan dengan error yang

mungkin terjadi, misalnya karena system kekurangan dan untuk diolah (usia pegawai belum diinput), terjadi masalah eksternal (printer belum di turn-on).

5. *Special Requiremen*, berisi kebutuhan lain yang belum tercakup dalam aliran normal dan alternatif. Biasanya secara tegas dibedakan bahwa *basic flow* dan *alternate flow* menangani kebutuhan fungsional dari *use case*, sementara *special requirement* yang tidak berhubungan dengan kebutuhan fungsional, misalnya kecepatan transaksi maksimum berapa cepat dan berapa lama, kapasitas akses yaitu jumlah user yang akan mengakses dalam waktu bersamaan.
6. *Pre Condition*, menjelaskan persyaratan yang harus dipenuhi sebelum *use case* bisa dimulai.
7. *Post Condition*, menjelaskan kondisi yang berusaha atau terjadi saat *use case* selesai dieksekusi (Julius Hermawan; 2007:16-17).

#### **II.11.5. Interaction**

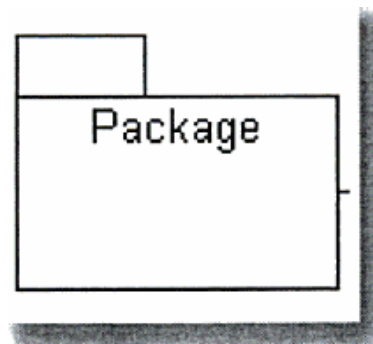
*Interaction* digunakan untuk menunjukkan baik aliran pesa atau informasi antar obyek maupun hubungan antar obyek. Biasanya *interaction* ini dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan. *Interaction* dapat dilihat pada Gambar II.9.



**Gambar II.9. Interaction**  
(Sumber : Julius Hermawan;2007:18)

### II.11.6. Package

*Package* adalah container atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan (*visibility*) dari model yang sedang dibangun.



**Gambar II.10. Package**  
(Sumber : Julius Hermawan;2007:19).

### II.11.7. Dependency

*Dependency* merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen member pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah. Dapat dilihat pada Gambar II.11.

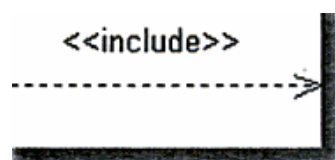


**Gambar II.11. *Dependency Extends***

(Sumber : Julius Hermawan; 2007 : 20)

Terdapat 2 stereotype dari dependency, yaitu include dan extend. Include menunjukkan bahwa suatu bagian dari elemen memicu eksekusi bagian dari elemen lain, misalnya untuk notasi A -> B operasi yang ada di class A memicu dieksekusinya operasi yang ada di class B.

*Extend* menunjukkan bahwa suatu bagian dari elemen di garis tanpa panah bisa disisipkan ke dalam elemen yang ada di garis dengan panah, misalnya untuk notasi A -> B suatu fungsi dari use case A bisa disisipkan ke dalam use case B atau dengan kata lain A optional untuk B. Kedua *stereotype* ini direpresentasikan dengan menambahkan text include atau extend di notasi *dependency*. Dapat dilihat pada Gambar II.12.



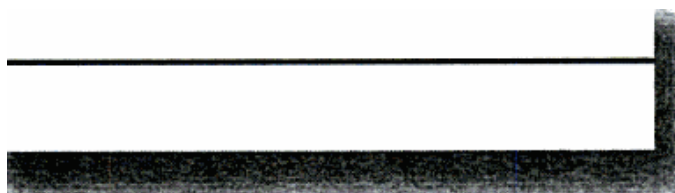
**Gambar II.12. *Depedency Include***

(Sumber : Julius Hermawan; 2007 : 20)

### **II.10.8. *Association***

*Association* menggambarkan navigasi antar class (*Navigation*), berapa banyak obyek lain yang bisa dihubungkan dengan satu obyek (*Multiplicity* antar *class*), dan apakah suatu class menjadi bagian dari class lainnya (*Aggregation*).

*Navigation* dilambangkan dengan penambahan tanda panah di akhir garis. Bidirectional navigasi menunjukkan bahwa dengan mengetahui salah satu class bisa didapatkan informasi dari class lainnya. Sementara dengan *UniDirectional* navigation hanya dengan mengetahui class di ujung garis association tanpa panah kita bisa mendapatkan informasi dari class di ujung dengan panah, tetapi tidak sebaliknya. Simbol *Association* dapat dilihat pada Gambar II.12.



**Gambar II.13. Simbol Association**

(Sumber : Julius Hermawan; 2007 : 21)

## **II.11. Microsoft Visual Basic 2010**

Visual Basic merupakan salah satu bahasa pemrograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi Windows. Visual Basic 2008 atau Visual Basic 9 adalah versi terbaru yang telah diluncurkan oleh Microsoft bersama C#, visual C++, dan Visual Web Developer dalam satu paket Visual Studio 2008 (Wahana Komputer; 2010: 2).

Visual Basic 2008 merupakan aplikasi pemrograman yang menggunakan teknologi .NET Framework. Teknologi .NET Framework merupakan komponen Windows yang terintegrasi serta mendukung pembuata, penggunaan aplikasi, dan halaman web. Teknologi .NET Framework mempunyai 2 komponen utama, yaitu CLR (Common Language Runtime) dan Class Library, CLR digunakan untuk menjalankan aplikasi yang berbasis .NET, sedangkan Library adalah kelas pustaka

atau perintah yang digunakan untuk membangun aplikasi (Wahana komputer; 2010:2).

## **II.12. SQL Server 2008**

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang database. SQL Server adalah sebuah DBMS (Database Management System) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server 2008 dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

Microsoft merilis SQL Server 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut :

1. Versi 32 bit (x86), yang biasanya digunakan untuk komputer dengan single processor (Pentium 4) atau lebih tepatnya processor 32 bit dan sistem operasi Windows XP.
2. Versi 64 bit (x64), yang biasanya digunakan untuk komputer dengan lebih dari satu processor (Misalnya Core 2 Duo) dan system operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7 (Wahana Komputer; 2010:3).