

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisir, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh Kennet 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 melahirkan konsep-konsep futuristik, antara lain yang terkenal adalah konsep sibernetika (*cybernetics*). Konsep atau dibidang kajian ilmiah ini berkaitan dengan upaya menerapkan berbagai ilmu yaitu ilmu perilaku, fisika, biologi, dan teknik. Oleh karena itu sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan manusia, sehingga melahirkan studi-studi tentang robotika, kecerdasan buatan (*artificial intelegence*). Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*processing*), dan keluaran (*output*).

selain itu, suatu sistem tidak bisa lepas dari lingkungan maka umpan balik (*feed back*) dapat berasal dari lingkungan sistem yang dimaksud. Organisasi dipandang sebagai suatu sistem yang tentunya akan memiliki semua unsur ini (Tata Sutabri; 2012).

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

Model umum sebuah sistem adalah input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem (*Component*)
2. Batas Sistem (*Boundary*)
3. Lingkungan Luar Sistem (*Environment*)
4. Penghubung Sistem (*Interface*)
5. Masukan Sistem (*Input*)
6. Keluaran Sistem (*Output*)
7. Pengolah Sistem (*Proses*)
8. Sasaran Sistem (*Objective*)

### **II.1.2. Klasifikasi Sistem**

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandangannya antara lain :

1. Sistem Abstrak Dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan tuhan, sedangkan sistem fisik merupakan sistem

yang ada secara fisik, misalnya sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia, dan lain sebagainya.

## 2. Sistem Alamiah Dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia dengan mesin, merupakan melibatkan interaksi manusia dengan mesin, yang disebut "*human machine system*". Sistem informasi berbasis komputer merupakan contoh *human machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

## 3. Sistem Deterministik Dan Sistem Probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministic. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem bersifat probabilistik adalah sistem yang mana kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistic.

## 4. Sistem Terbuka Dan Sistem Tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya (Tata Sutabri; 2012).

### II.1.3. Daur Hidup Sistem

Siklus hidup sistem (*system life cycle*) adalah merupakan proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi komputer. Siklus hidup sistem terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem sistem karena tugas-tugas tersebut mengikuti pola yang teratur dan dilakukan secara *top down*. Siklus hidup sistem sering disebut sebagai pendekatan air terjun (*waterfall approach*) bagi pembangunan dan pengembangan sistem.

Pembangunan sistem hanyalah salah satu dari rangkaian daur hidup suatu sistem. Meskipun demikian, proses ini merupakan aspek yang sangat penting. Kita akan melihat beberapa fase/tahapan dari daur hidup suatu system yaitu :

#### 1. Mengenal Adanya Kebutuhan

Sebelum segala sesuatunya terjadi, timbul suatu kebutuhan atau problema yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan dari organisasi dan volume yang meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan ini harus dapat didefinisikan dengan jelas. Tanpa adanya kejelasan dari kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektifitasnya.

#### 2. Pembangunan Sistem

Suatu proses atau seperangkat prosedur yang harus diikuti untuk menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut.

### 3. Pemasangan Sistem

Setelah tahap pembangunan sistem selesai. Sistem kemudian akan dioperasikan. Pemasangan sistem merupakan tahap yang penting pula dalam daur hidup sistem. Peralihan dari tahap pembangunan menuju tahap operasional terjadi pemasangan sistem yang sebenarnya, yang akan merupakan langkah akhir dari suatu pembangunan.

### 4. Pengoperasian Sistem

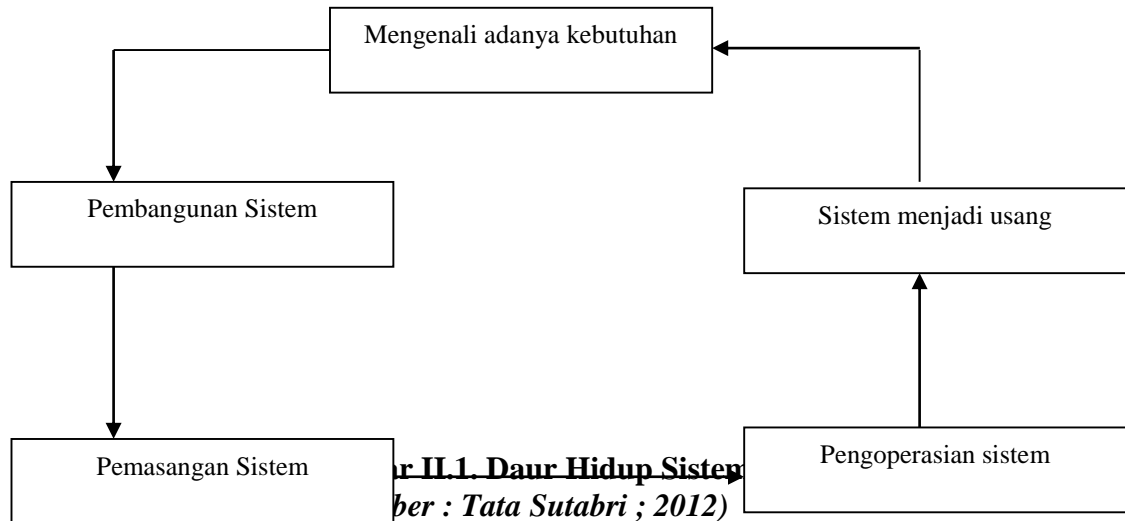
Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis. Sedangkan organisasi ditunjang oleh sistem informasi tadi. Ia selalu mengalami perubahan-perubahan itu karena pertumbuhan kegiatan bisnis, perubahan pengaturan, dan kebijaksanaan ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaiki atau diperbaharui.

### 5. Sistem Menjadi Usang

Kadang perubahan yang terjadi begitu drastis, sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Tibalah saatnya secara ekonomis dan teknis sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya.

Sistem informasi kemudian akan melanjutkan daur hidupnya. Sistem dibangun untuk memenuhi kebutuhan yang muncul. Sistem beradaptasi terhadap perubahan-perubahan lingkungannya dinamis. Sampailah pada kondisi dimana sistem tersebut tidak dapat lagi beradaptasi dengan perubahan-perubahan yang ada atau secara ekonomis tidak layak lagi untuk dioperasikan. Sistem yang baru

kemudian dibangun untuk menggantikannya. Untuk dapat menggambarkan daur hidup sistem ini, lihat pada gambar II.1. sebagai berikut (Tata Sutabri; 2012).



**Gambar II.1. Daur Hidup Sistem**  
 Sumber : Tata Sutabri ; 2012)

## II.2. Informasi

Informasi adalah data yang telah diklasifikasikan diolah atau diinterpretasi untuk digunakan dalam proses pengambil keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah dari bentuk tak berguna menjadi berguna bagi penerimanya.

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam pengambil keputusan (Tata Sutabri, 2012).

## II.3. Sistem Informasi

Sistem informasi adalah berupa suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian yang mendukung operasi yang bersifat manajerial dengan kegiatan strategi suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-

laporan yang diperlukan (Tata Sutabri, 2012).**II.3.1. Komponen Dan Jenis Sistem Informasi**

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk satu kesatuan untuk mencapai sasaran :

1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk ke dalam sistem informasi. *Input* di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memainipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan "*tool box*" dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu

pengendalian dari sistem keseluruhan. Teknologi sistem terdiri dari 3 (tiga) bagian yaitu teknisi teknologi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

#### 5. Blok Basis Data (*Database Block*)

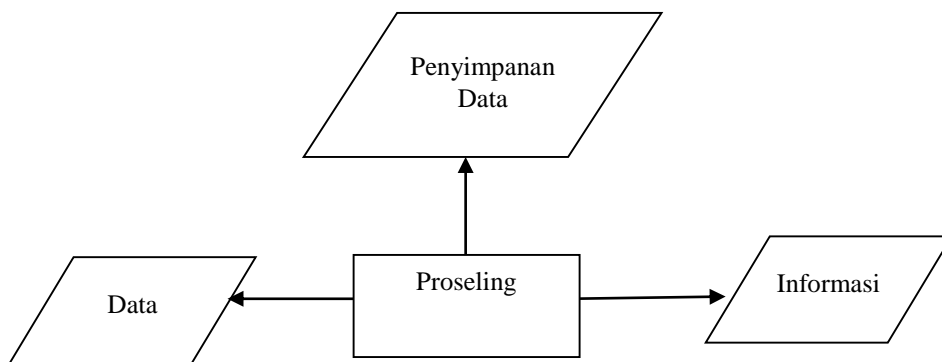
Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*database management system*).

#### 6. Blok Kendali (*Control Block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi (Tata Sutabri, 2012).

## II.4. Data

Mengenai pengertian data, lebih jelas apa yang didefinisikan oleh Drs. Jhon J. Longkutoy dalam bukunya “ Pengenalan Komputer” sebagai berikut : isitilah data adalah suatu istilah majemuk yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan simbol-simbol, gambar-gambar, angka-angka, huruf-huruf, atau simbol-simbol yang menunjukkan suatu ide, objek, kondisi, atau situasi dan lain-lain (Tata Sutabri, 2012).



**Gambar II.2. Pemrosesan Data**  
(Sumber : Tata Sutabri, S. Kom, MM ; 2012)

### II.4.1. Pengolahan Data

Data merupakan bahan mentah untuk diolah, yang hasilnya kemudian menjadi informasi. Dengan kata lain, data yang telah diperoleh harus diukur dan dinilai baik buruknya, berguna atau tidak dalam hubungannya dengan tujuan yang akan dicapai. Pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan penanganan data. Untuk lebih jelasnya akan diuraikan dibawah ini (Tata Sutabri, 2012) :

## 1. Penyimpanan Data (*Data Storage*)

Penyimpanan data meliputi pekerjaan pengumpulan (*filings*), pencarian (*searching*), dan pemeliharaan (*maintenance*). Data disimpan dalam suatu tempat yang lazim dinamakan "*file*". *File* dapat berbentuk map, *ordner*, *disket*, *tape*, *hard disk*, dan lain sebagainya. Sebelum disimpan, suatu data diberi kode menurut jenis kepentingannya. Pengaturan dilakukan sedemikian rupa sehingga mudah mencarinya. Pengkodean memegang peranan penting. Kode yang salah data akan mengakibatkan data masuk ke dalam *file* juga salah, yang selanjutnya akan mengakibatkan kesulitan pencarian data tersebut apabila diperlukan. Jadi *file* diartikan suatu susunan data yang terbentuk dari sejumlah catatan (*record*) yang berhubungan satu sama lain (sejenis) mengenai suatu bidang dalam suatu unit usaha.

Sistem yang umum dalam penyimpanan data (*filings*) ialah berdasarkan lembaga, perorangan, produksi, atau lain-lainya, tergantung dari sifat organisasi yang bersangkutan. Kadang-kadang dijumpai kesulitan apabila menghadapi suatu data dalam bentuk surat misalnya yang menyangkut ketiga klasifikasi tadi. Metode yang terbaik adalah "referensi silang" (*cross reference*) antara *file* yang satu dengan *file* lain. Untuk memperoleh kemudahan dalam pencarian data (*searching*) di dalam *file*, maka *file* dibagi menjadi 2 (dua) jenis yaitu :

### a. *File* Induk (*Master File*)

*File* induk ini berisi data-data permanen yang biasanya hanya dibentuk satu kali saja dan kemudian digunakan untuk pengolahan data selanjutnya.

Contoh : *file* kepegawaian, *file* gaji

b. *File Transaksi (Detail File)*.

*File* transaksi berisi data-data temporer untuk suatu periode untuk suatu bidang kegiatan atau suatu periode yang dihubungkan dengan suatu bidang kegiatan.

Contoh : *file* lembur perminggu, *file* mutasi harian.

Pemeliharaan *file (file maintenance)* juga meliputi “peremajaan data” (*data updating*), yaitu kegiatan menambah catatan baru pada suatu data, mengadakan perbaikan, dan lain sebagainya. Misalnya, dalam hubungan dengan *file* kepegawaian, sudah tentu sebuah organisasi, entah itu perusahaan atau jawatan, akan menambah pegawainya. Sementara itu, ada pula pegawai yang pensiun atau berhenti bekerja atau putus hubungan dengan organisasi. Dengan demikian, data mengenai pegawai yang bersangkutan akan dikeluarkan dari *file* tersebut. Tidak jarang pula harus dilakukan perubahan data terhadap data seorang pegawai, misalnya kenaikan pangkat, kenaikan gaji berkala, menikah, pindah alamat, dan lain sebagainya.

2. Penanganan Data (*Data Handling*)

Penanganan data meliputi berbagai kegiatan, seperti pemeriksaan (*verifying*), perbandingan (*comparing*), pemilihan (*sorting*), peringkasan (*extracting*), dan penggunaan (*manipulating*). Pemeriksaan data mencakup pengecekan data yang muncul pada berbagai daftar yang berkaitan atau yang datang dari berbagai sumber, untuk mengetahui berbagai sumber dan perbedaan atau ketidaksesuaian. Pemeriksaan ini dilakukan dengan kegiatan pemeliharaan *file (file maintenance)*.

Pemilihan atau sorting dalam rangka kegiatan penanganan data mencakup pengaturan ke dalam suatu urutan yang teratur, misalnya daftar pegawai menurut pangkatnya, dari pangkat yang tertinggi sampai yang terendah atau daftar pelanggan dengan menyusun namanya menurut abjad dan lain sebagainya. Peringkasan merupakan kegiatan lain dalam penanganan data. Ini mencakup keterangan pilihan, misalnya daftar pegawai yang telah mengabdikan dirinya kepada organisasi/perusahaan lebih dari 10 tahun atau daftar yang memesan beberapa hasil produksi sekaligus dan lain-lain.

Penggunaan data atau informasi “*data manipulation*” merupakan kegiatan untuk menghasilkan informasi. Kegiatan ini meliputi kompilasi tabel-tabel, statistik, ramalan mengenai perkembangan, dan lain sebagainya. Tujuan manipulasi ini adalah menyajikan informasi yang memadai mengenai apa yang terjadi pada waktu lampau guna menunjang manajemen, terutama membantu menyelidiki alternatif kegiatan mendatang. Jadi, hasil pengolahan data itu merupakan data untuk disimpan bagi penggunaan di waktu yang akan datang, yakni informasi yang akan disampaikan kepada yang memerlukan atau mengambil keputusan mengenai suatu hal (Tata Sutabri, 2005).

## **II.5. Pengertian Persediaan**

Menurut Donald E, Kieso yang diterjemakan oleh Emil Salim,SE (2009), “Persediaan (inventory) adalah pos-pos aktiva yang dimiliki oleh perusahaan untuk dijual dalam operasi bisnis normal, atau barang yang akan digunakan atau dikonsumsi dalam membuat kecermatan.investasi dalam persediaan biasanya

merupakan aktiva lancar paling besar dari perusahaan dagang (ritel) dan manufaktur”.

### **II.5.1. Jenis Persediaan**

Menurut Stice dan Skousen yang diterjemakan oleh Ali Akbar (2009) kata persediaan (atau persediaan barang dagangan) secara umum ditunjukkan untuk barang-barang yang dimiliki oleh perusahaan dagang, baik berupa usaha grosir maupun ritel, ketika barang-barang tersebut telah dibeli dan ada kondisi siap untuk dijual.

1. Bahan Baku
2. Barang dalam proses
3. Barang jadi
- 4.

### **II.5.2. Sistem Pencatatan Persediaan**

Menurut Mulyadi (2001) ada dua macam metode pencatatan persediaan yaitu metode mutasi persediaan (*perpetual inventory method*) dan metode persediaan fisik (*physical inventory method*).

Dalam metode mutasi persediaan, setiap mutasi persediaan dicatat dalam kartu persediaan. Dalam metode persediaan fisik, hanya penambahan persediaan dari pembelian saja yang dicatat, sedangkan mutasi berkurangnya persediaan karena pemakaian tidak dicatat dalam kartu persediaan. Untuk mengetahui berapa harga pokok persediaan yang dipakai atau dijual, harus dilakukan dengan perhitungan fisik sisa persediaan yang masih ada digudang pada akhir periode akuntansi.

Sedangkan menurut Donal E, Kieso yang diterjemakan oleh Emil Salim, SE (2009) dalam pencatatan persediaan ada dua metode pencatatan, perusahaan menggunakan salah satu dari dua jenis sistem agar pencatatan persediaan tetap aktual yaitu :

#### 1. Sistem Perpetual

Sistem persediaan perpetual (perpetual inventory system) secara terus-menerus melacak perubahan akun persediaan. Yaitu, semua pembelian dan penjualan (pengeluaran) barang dicatat secara langsung ke akun persediaan pada saat terjadi. Karakteristik akuntansi dari sistem persediaan perpetual adalah :

- a. Pembelian barang dagang untuk dijual atau pembelian bahan baku untuk produksi didebet ke persediaan dan bukan ke pembelian
- b. Biaya transportasi masuk, retur pembelian dan pengurangan harga, serta diskon pembelian didebet ke persediaan dan bukan ke akun terpisah.
- c. Harga pokok penjualan diakui untuk setiap penjualan dengan mendebet akun harga pokok penjualan, dan mengkredit persediaan.
- d. Persediaan merupakan akun pengendalian yang didukung oleh buku besar pembantu yang berisi catatan persediaan individual.

#### 2. Sistem Periodik

Menurut sistem persediaan periodik (periodic inventory system), kuantitas persediaan di tangan ditentukan, seperti yang tersirat oleh namanya, secara periodik. Semua pembelian persediaan selama periode akuntansi dicatat dengan mendebet akun pembelian. Total akun pembelian pada akhir periode akuntansi

ditambahkan ke biaya persediaan di tangan pada awal periode untuk menentukan total biaya barang yang tersedia untuk dijual selama periode berjalan.

### **II.5.3. Metode Penilaian Persediaan**

Menurut Stice dan Skousen yang diterjemakan oleh Ali Akbar (2009) pada akhir priode, total biaya persediaan harus dialokasikan dalam persediaan yang masi ada (untuk dilaporkan di neraca sebagai aset) dan dalam persediaan yang terjual selama priode tersebut (untuk dilaporkan di laporan laa rugi sebagai beban "harga pokok penjualan"). Metode - metode yang paling umum adalah :

#### **1. Identifikasi Khusus**

Pada metode ini, biaya dapat dialokasikan ke barang yang terjual selama periode berjalan dan ke barang yang ada di tangan pada akahir periode berdasarkan biaya aktual dari unit tersebut. Metode ini diperlukan untuk mengidentifikasi biaya historis dari unit persediaan. Dengan indenfikasi khusus, arus biaya yang dicatat disesuaikan dengan arus fisik barang.

#### **2. Metode Biaya Rata-rata (*Average*)**

Metode ini membebankan biaya rata-rata yang sama ke setiap unit. Metode ini didasarkan pada asumsi bahwa barang-barang yang terjual seharusnya dibeli pada tiap harga. Metode rata-rata mengutamakan yang mudah terjangkau untuk dilayani, tidak peduli apakah barang tersebut masuk pertama atau masuk terakhir.

### 3. Metode Masuk Pertama, Keluar Pertama (FIFO)

Metode ini didasarkan pada asumsi bahwa unit yang terjual adalah unit yang terlebih dahulu masuk. FIFO dapat dianggap sebagai sebuah pendekatan yang logis dan realitas terhadap arus biaya ketika penggunaan metode identifikasi khusus adalah tidak memungkinkan atau tidak praktis. FIFO mengasumsikan bahwa arus biaya yang mendekati paralel dengan arus fisik dari barang yang terjual. Beban dikenakan pada biaya yang dinilai melekat pada barang yang terjual. FIFO memberikan kesempatan kecil untuk memanipulasi keuntungan karena pembebanan biaya ditentukan oleh urutan terjadinya biaya. Selain itu, di dalam FIFO unit yang tersedia pada persediaan akhir adalah unit yang paling terakhir dibeli, sehingga biaya yang dilaporkan akan mendekati atau sama dengan biaya penggantian diakhir periode.

### 4. Metode Masuk Terakhir, Keluar Pertama (LIFO)

Metode ini didasarkan pada asumsi bahwa barang yang paling barulah yang terjual. Metode LIFO sering dikritik secara teoritis tetapi metode ini adalah metode yang paling baik dalam pengaitan biaya persediaan dengan pendapatan. Apalagi metode LIFO digunakan selama periode inflasi atau harga naik, LIFO akan menghasilkan harga pokok yang lebih tinggi, jumlah laba kotor yang lebih rendah dan persediaan akhir yang lebih rendah. Dengan demikian LIFO cenderung memberikan pengaruh yang stabil terhadap margin laba kotor, karena pada saat terjadi kenaikan harga LIFO mengaitkan biaya yang tinggi saat ini dalam perolehan barang-barang dengan harga jual yang meningkat, dengan menggunakan LIFO, persediaan dilaporkan dengan menggunakan biaya dari

pembelian awal. Jika LIFO digunakan dalam waktu yang lama, maka perbedaan antara nilai saat ini dengan biaya LIFO akan semakin besar.

## **II.6. Sistem Informasi Akuntansi (SIA)**

Menurut Murdick (1984, dalam Jogiyanto, 2007) SIA adalah kumpulan kegiatan-kegiatan dari organisasi yang bertanggung jawab untuk menyediakan informasi keuangan dan informasi yang didapatkan dari transaksi data untuk tujuan pelaporan internal kepada manajer untuk digunakan dalam pengendalian dan perencanaan sekarang dan operasi masa depan serta pelaporan eksternal kepada pemegang saham, pemerintah, dan pihak-pihak luar lainnya. Dari definisi tersebut dapat diambil kesimpulan bahwa SIA merupakan suatu kegiatan input, proses, dan output data yang dilakukan oleh perusahaan. Hasil data akhir yang telah di proses SIA bertujuan sebagai pelaporan bagi pihak internal dan eksternal guna melakukan pengendalian terhadap perusahaan tersebut.

## **II.7. *Unified Modeling Language (UML)***

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. Menurut Chonoles (2003) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan - aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem,

bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

*UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

*UML* telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

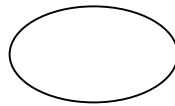
Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Dan Herlawati; 2011).

### II.7.1. Diagram - Diagram UML

Model - model diagram UML dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

#### 1. *Use Case Diagram*

Menurut Pilone (2005) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. Sedangkan menurut Whitten (2004) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval*.



**Gambar II.3. Simbol Use Case**  
(Sumber : Prabowo Pudjo Widodo Dan Herlawati ; 2011)

*Use case* sangat menentukan karakteristik sistem yang kita buat, oleh karena itu Chonoles (2003) menawarkan cara untuk menghasilkan *use case* yang baik yakni :

- a. Pilihlah Nama Yang Baik.
- b. Ilustrasikan Perilaku Dengan Lengkap.
- c. Identifikasi Perilaku Dengan Lengkap.
- d. Menyediakan Use Case Lawan (*Inverse*)
- e. Batasi Use Case Hingga Satu Perilaku Saja

## 2. *Diagram Aktivitas (Activity Diagram)*

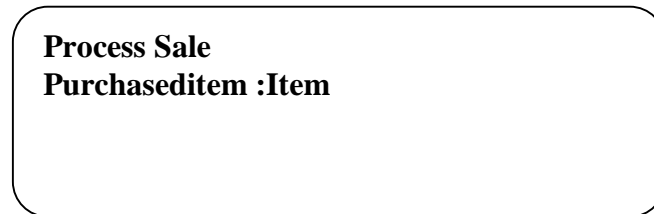
Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore (Probowo Pudji Widodo, Dan Herlawati; 2011).

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi nelakukan langka sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

- a. Fungsi Matematika
- b. Pemanggilan Perilaku
- c. Pemrosesan Data

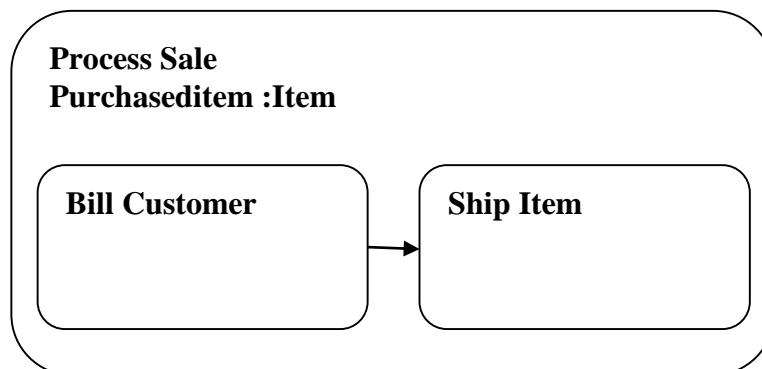
Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classifier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan dengan model proses bisnis, informasi itu biasanya disebut *process-relevant data*. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya *specific* dan digunakan hanya untuk aktivitas tertentu.

Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



**Gambar II.4. Aktivitas Serderhana Tanpa Rincian**  
(Sumber : Prabowo Pudjo Widodo Dan Herlawati ; 2011)

Detail aktivitas dapat dimasukan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.

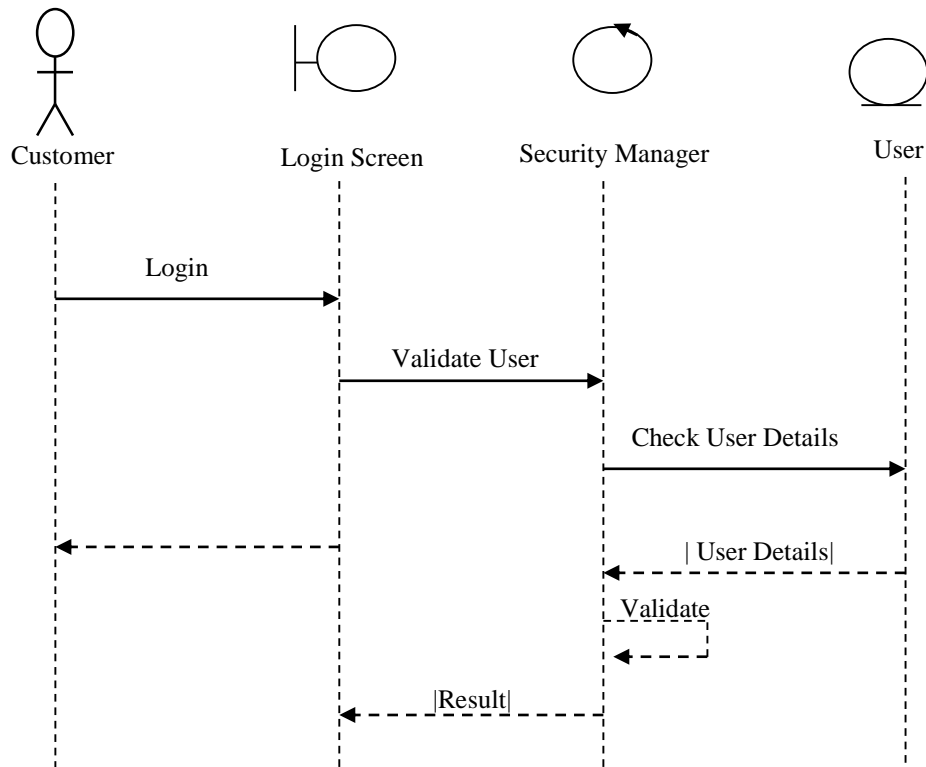


**Gambar II.5. Aktivitas Dengan Detail Rincian**  
(Sumber : Prabowo Pudjo Widodo Dan Herlawati ; 2011)

### 3. Diagram Urutan (*Sequence Diagram*)

Menurut Douglas (2004) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*).

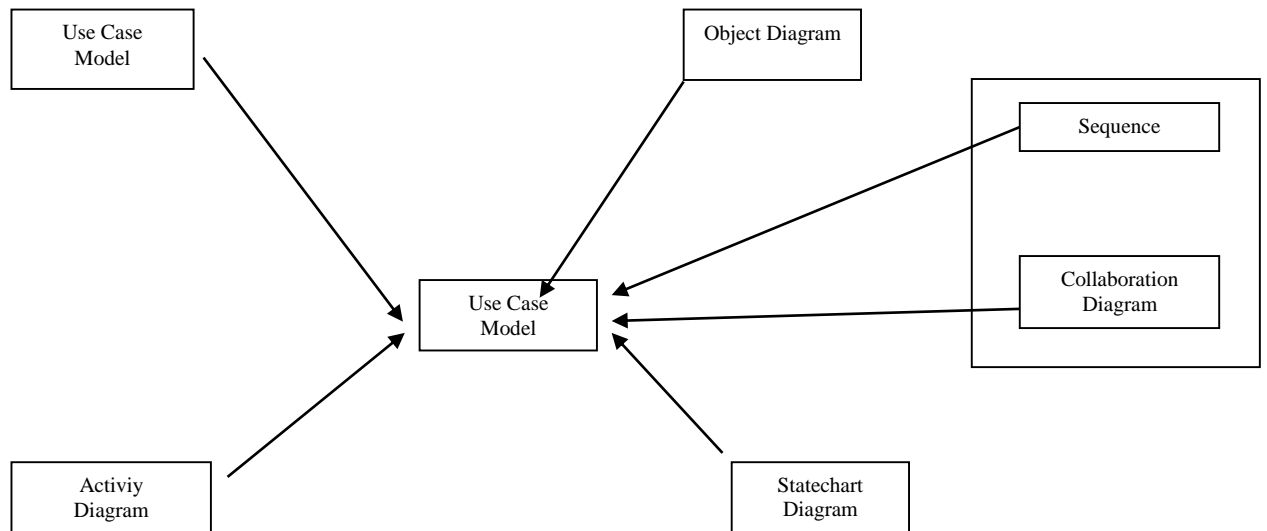
Menurut Pilone (2005) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.7. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya (Prabowo Pudjo Widodo Dan Herlawati; 2011).



**Gambar II.6. Diagram Urutan**  
(Sumber : Prabowo Pudjo Widodo Dan Herlawati ; 2011)

#### 4. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini *forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model (Prabowo Pudjo Widodo Dan Herlawati; 2011).



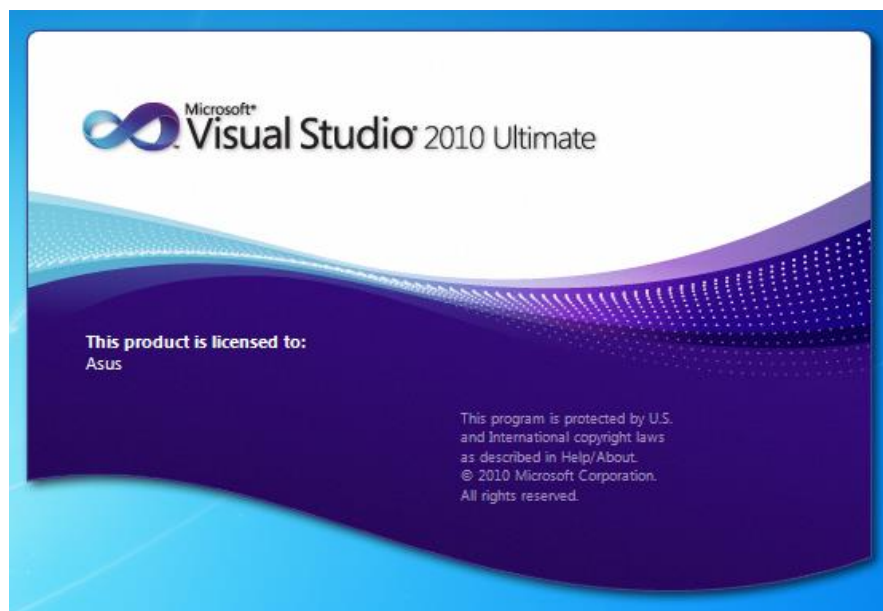
**Gambar II.7. Hubungan Diagram Kelas Dengan Diagram UML lainnya**  
 (Sumber : Prabowo Pudjo Widodo Dan Herlawati ; 2011)

## II.8. Bahasa Pemrograman Microsoft Visual Studio 2010

Microsoft Visual Studio merupakan aplikasi pemrograman yang menggunakan teknologi .NET Framework. Teknologi .NET Framework merupakan komponen *Windows* yang terintegrasi serta mendukung pembuatan, penggunaan aplikasi dan halaman web. Teknologi .NET Framework mempunyai 2 komponen utama, yaitu CLS (*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).

Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++,

Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*).



**Gambar II.8. Tampilan Awal Microsoft Visual Studio 2010**  
(Sumber : Erick Kurniawan ; 2011)

Visual Studio kini telah menginjak versi *Visual Studio 9.0.21022.08*, atau dikenal dengan sebutan *Microsoft Visual Studio 2008* yang diluncurkan pada 19 November 2007, yang ditujukan untuk platform *Microsoft .NET Framework 3.5*. Versi sebelumnya, *Visual Studio 2005* ditujukan untuk platform *.NET Framework 2.0 dan 3.0*. *Visual Studio 2003* ditujukan untuk *.NET Framework 1.1*, dan *Visual Studio 2002* ditujukan untuk *.NET Framework 1.0*. Versi-versi tersebut di atas

kini dikenal dengan sebutan *Visual Studio .NET*, karena memang membutuhkan *Microsoft .NET Framework*. Sementara itu, sebelum muncul *Visual Studio .NET*, terdapat *Microsoft Visual Studio 6.0 (VS1998)*.

## II.9. *MYSQL*

*Mysql* adalah salah satu software sistem manajemen *database* (DBMS) *structured Query Language (SQL)* yang bersifat *open source*. *SQL* adalah bahasa standart untuk mengakses *database* dan didefenisikan dengan standart *ANSI/ISO SQL*. *MYSQL* dikembangkan, disebarluaskan, dan didukung oleh *MYSQL AB*. *MYSQL AB* adalah perusahaan komersial yang didirikan oleh pengembang *MYSQL*. *MYSQL* merupakan aplikasi *Relational Database Management System* (RDBMS) yang digunakan untuk aplikasi *client server* atau sistem *embedded*.

*Mysql* mempunyai beberapa sifat yang menjadikannya sebagai salah satu software *database* yang banyak digunakan oleh pemakai diseluruh dunia. Sifat-sifat yang dimiliki oleh *MYSQL* antara lain :

1. *Mysql* merupakan DBMS (*Database Management System*)
2. *Database* adalah kumpulan data yang terstruktur. Data dapat berupa daftar belanja, kumpulan gambar, atau yang lebih luas yaitu informasi jaringan perusahaan. Agar dapat menambah, mengakses, dan memproses data tersimpan pada sebuah komputer *database*, kita membutuhkan sistem manajemen *database* (DBMS) seperti *MYSQL Server*. Sejak komputer sangat baik menangani sejumlah besar data, *sistem manajemen database* (DBMS) memainkan peran utama dalam perhitungan baik sebagai peralatan yang berdiri sendiri maupun bagian aplikasi.

3. *Mysql* merupakan RDBMS (*Relational Database Management System*).
4. *Database relational* menyimpan data pada table-tabel yang terpisah, bukan menyimpan data dalam ruang penyimpanan yang besar. Hal ini menambah kecepatan *fleksibilitas*.
5. *Mysql* merupakan *software open source*.
6. *Open source* berarti setiap orang dapat menggunakan dan mengubah software yang bersangkutan. Setiap orang dapat mendownload *software MYSQL* dari internet dan menggunakan tanpa membayar. Bahkan jika mengkehendaknya anda bisa mempelajari kode sumber dan mengubah sesuai yang anda butuhkan (Wahana Komputer; 2008).

## **II.10. Normalisasi**

Menurut Edhy Sutanta (2011) normalisasi diartikan sebagai suatu teknik yang menstrukturkan / mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomalles*) yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi penolahan.

Proses normalisasi menghasilkan relasi yang optimal yaitu :

1. Memiliki struktur *record* yang konsisten secara logic.
2. Memiliki struktur *record* yang mudah untuk dimengerti.
3. Memiliki struktur *record* yang sederhana dalam pemeliharaan.
4. Memiliki struktur *record* yang mudah ditampilkan kembali untuk memenuhi kebutuhan pengguna.

5. Meminimalisasi kerangkapan data guna meningkatkan kinerja system.

### **II.10.1. Level Normalisasi**

Menurut Edhy Sutanta (2011) bentuk normal yang dikenal hingga saat ini meliputi bentuk 1NF, 2NF, 3NF, BCNF, 4NF, 5NF, DKNF, dan RUNF. Secara berturut-turut level normal tersebut akan dibahas berikut ini, dimulai dari bentuk tidak normal.

#### **1. Relasi bentuk tidak Normal (*un normalized* / UNF)**

Relasi-relasi yang dirancang tanpa mengindahkan batasan dalam definisi basis data dan karakteristik RDBM menghasilkan relasi UNF. Bentuk ini harus dihindari dalam perancangan relasi dalam basis data. Relasi UNF mempunyai kriteria sebagai berikut :

- a. Jika relasi mempunyai bentuk *non flat file* (dapat terjadi akibat data disimpan sesuai dengan kedatangannya, tidak memiliki struktur tertentu, terjadi duplikasi atau tidak lengkap).
- b. Jika relasi memuat set atribut berulang (*non single value*).
- c. Jika relasi memuat atribut *non atomic value*.

#### **2. Relasi bentuk normal pertama (*first norm form* / 1NF)**

Relasi disebut sebagai 1NF jika memenuhi kriteria sebagai berikut :

- a. Jika seluruh atribut dalam relasi bernilai atomic (*atomic value*).
- b. Jika seluruh atribut dalam relasi bernilai tunggal (*singlevalue*).
- c. Jika relasi tidak memuat set atribut berulang.
- d. Jika semua *record* mempunyai sejumlah atribut yang sama.

### 3. Bentuk normal kedua (*second norm form / 2NF*)

Relasi disebut sebagai 2NF jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria 1NF.
- b. Jika semua atribut nonkunci FD dan PK.

### 4. Bentuk normal ketiga (*third norm form / 3NF*)

Suatu relasi disebut 3NF jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria 2NF.
- b. Jika setiap atribut non kunci tidak TDF (*non transitive dependency*) terhadap PK.

### 5. Bentuk normal *Boyke-Codd* (*Boyce-Codd norm form / BCNF*)

Berikut normal BCNF dikemukakan oleh R.F. Boyce dan E.F. Codd.

Suatu relasi disebut sebagai BCNF jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria 3NF.
- b. Jika semua atribut penentu (determinan) merupakan CK.

### 6. Bentuk normal keempat (*forth norm form / 4NF*)

Relasi disebut sebagai 4NF jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria 3NF.
- b. Jika setiap atribut di dalamnya tidak mengalami ketergantungan pada banyak nilai atau dengan kalimat lain, bahwa semua atribut yang mengalami ketergantungan pada banyak nilai adalah bergantung secara fungsional (*functionally dependency*).

### 7. Bentuk normal kelima (*fifth norm form / 5NF*)

Suatu relasi memenuhi kriteria 5NF jika keserasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang sederhana.

**8. Bentuk normal kunci domain (*domain key norm form* / **DKNF**)**

Suatu relasi disebut sebagai DKNF jika setiap batasan dapat disimpulkan secara sederhana dengan mengetahui sekumpulan atribut pada kuncinya. Bentuk DKNF ini dikemukakan oleh R. Fagin pada 1981 dan bersifat sangat spesifik, artinya tidak semua relasi dapat mencapai level ini.