

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

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

Sedangkan pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan operasi di dalam sistem. Prosedur didefinisikan oleh Richard F. Neuschel yang disadur oleh Jogiyanto (2005 : 1) mendefinisikan sebagai berikut :

”Prosedur adalah salah satu urutan operasi klerikal (tulis-menulis) biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis terjadi.”

Menurut Turban (1998:34), sistem merupakan kumpulan dari objek-objek seperti manusia, sumber daya dan prosedur untuk melakukan suatu fungsi atau tujuan. Sistem terbagi menjadi tiga bagian, *input*, proses, dan *output*.

Berdasarkan definisi-definisi tersebut dapat disimpulkan bahwa sistem merupakan suatu jaringan kerja dari suatu prosedur-prosedur yang saling berhubungan dan saling ketergantungan dalam membentuk satu kesatuan atau organisasi untuk menyelesaikan suatu sasaran tertentu. Sistem mengandung komponen yang dapat berupa subsistem / bagian dari sistem yang mempunyai sifat dari sistem untuk menjalankan suatu fungsi dan mempengaruhi proses secara keseluruhan.

Dari beberapa kelompok definisi tersebut adalah benar dan tidak saling bertentangan, yang berbeda adalah cara pendeketannya. Pendekatan sistem yang merupakan kumpulan elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang luas. Definisi ini lebih banyak diterima, karena kenyataanya suatu sistem dapat terdiri dari beberapa subsistem atau sistem bagian.. Sebagai contoh, sistem akuntansi dapat terdiri dari beberapa subsistem-subsistem, yaitu subsistem akuntansi penjualan, subsistem akuntansi pembelian, subsistem akuntansi penggajian, subsistem akuntansi biaya dan lain sebagainya.

2.1.1 Elemen Sistem

Elemen sistem yang terdapat pada sebuah sistem terdiri dari :

a) Masukan (*input*)

Input merupakan data yang masuk dalam suatu sistem.

b) Keluaran (*output*)

Output merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

c) Model

Model merupakan kombinasi prosedur, logika, dan model matematik yang memproses data yang tersimpan di *database* dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

d) Teknologi

Teknologi merupakan alat dalam sistem informasi, teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan output dan membantu pengendalian sistem.

e) Basis Data (*database*)

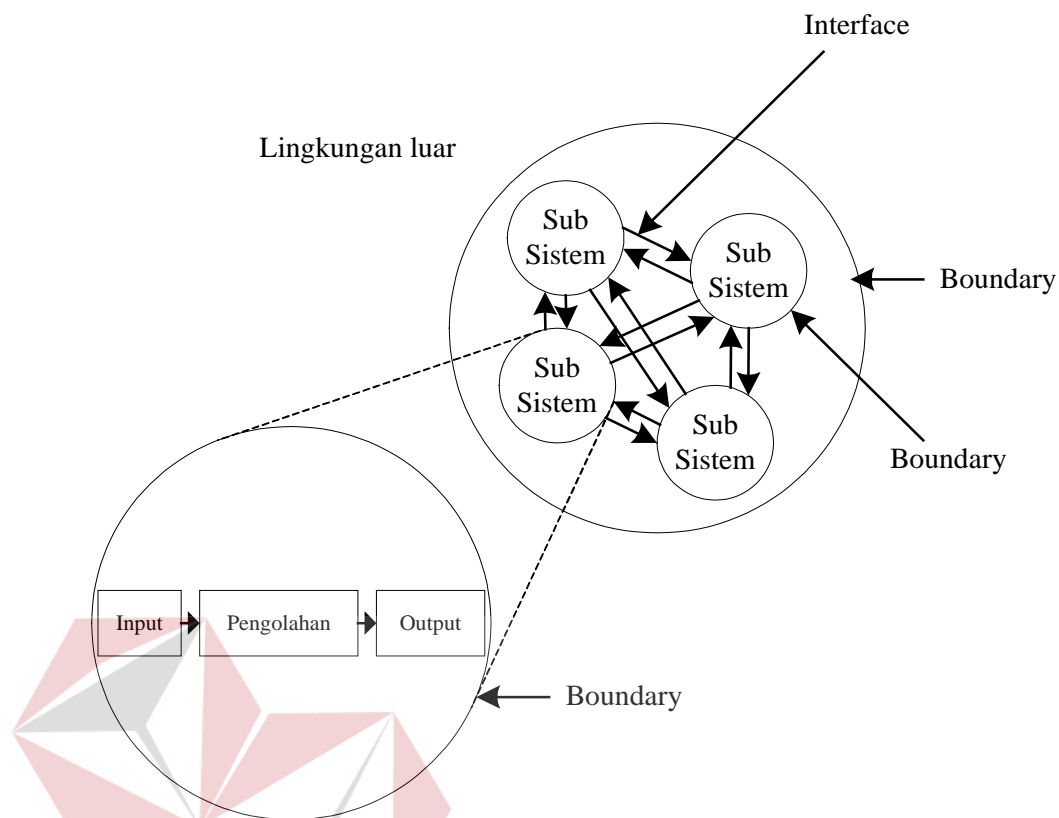
Database merupakan kumpulan data yang saling berhubungan yang tersimpan didalam komputer dengan menggunakan software *database*.

f) Kontrol (*control*)

Pengendalian atau *control* yang dirancang untuk menanggulangi gangguan terhadap sistem informasi.

2.1.2 Karakteristik Sistem

Definisi sistem menurut Jogiyanto (1999, 1) adalah kumpulan elemen-elemen yang saling berkaitan, bekerja sama untuk mencapai suatu tujuan. Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :



Gambar 2.1 Karakteristik Suatu Sistem

a. Komponen Sistem (*component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

b. Batas Sistem (*boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas sistem juga menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem (*environment*)

Lingkungan luar dari suatu sistem adalah apapun yang ada diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem dan harus tetap dijaga serta dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, karena kalau tidak maka akan mengganggu kelangsungan hidup dari sistem tersebut.

d. Penghubung Sistem (*interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya.

e. Masukan Sistem (*input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan signal (*signal input*). *Maintenance input* adalah energy yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energy yang diproses untuk didapatkan keluaran. Sebagai contoh di dalam sistem computer, program

adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

f. Keluaran Sistem (*output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat berupa masukan untuk subsistem yang lain atau kepada supra sistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna merupakan hasil sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

g. Pengolahan Sistem (*proses*)

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku atau bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

h. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objecyive*). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akana dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuannya.

2.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

a) Sistem abstrak (*abstract sistem*) dan sistem fisik (*physical sistem*)

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalkan sistem tologia, yaitu sistem berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalkan, sistem komputer, sistem penggajian, sistem produksi, sistem absensi, dan lain sebagainya.

b) Sistem tertentu (*deterministic sistem*) dan sistem tak tentu (*probabilistic sistem*)

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer sebagai contoh sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

c) Sistem alamiah (*natural sistem*) dan sistem buatan (*human made sistem*)

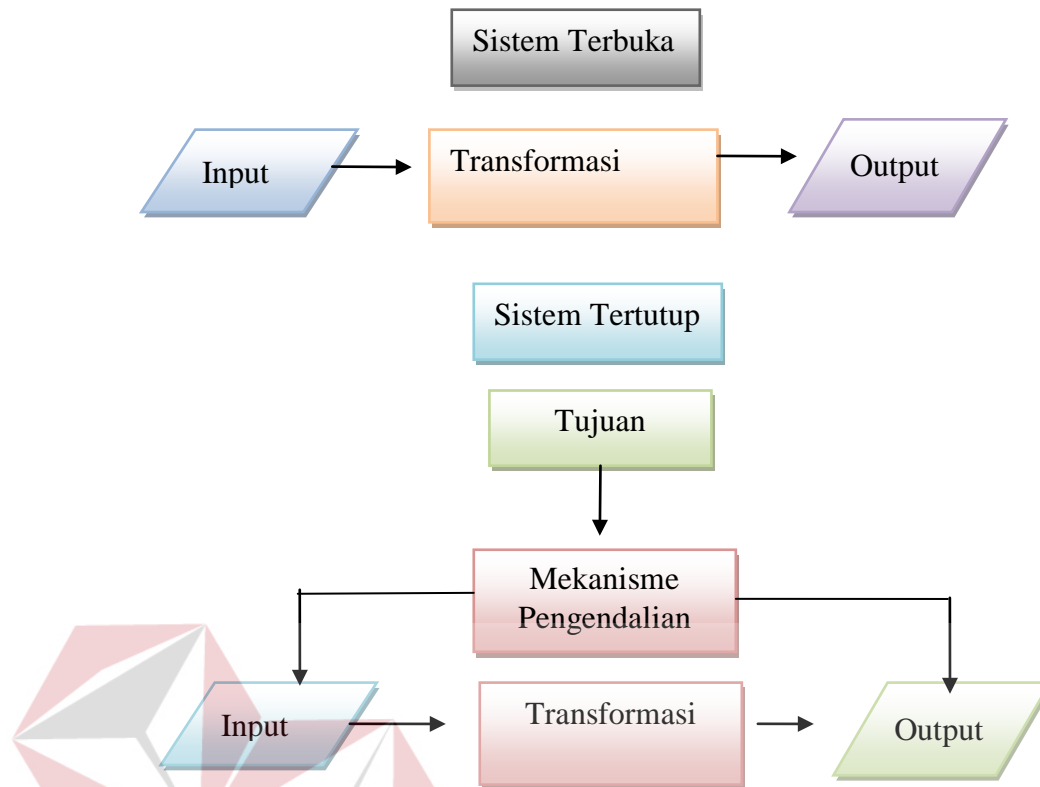
Sistem alamiah adalah sistem yang terjadi melalui proses alam yang telah melalui proses alam yang telah ditentukan oleh Tuhan, tidak dibuat manusia. Misalkan sistem tata surya. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human machine sistem* atau *man-*

machine sistem. Sistem informasi penggajian merupakan contoh *man-machine system*, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

d) Sistem tertutup (*closed sistem*) dan sistem terbuka (*open sistem*)

Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luarnya. Namun pada sistem tertutup ini ada juga yang kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relative closed sistem* (sistem relative tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistem sifat terbuka dan terpengaruh oleh lingkungan luar, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik.

Klasifikasi sistem terbuka dan sistem tertutup dapat digambarkan sebagai berikut :



Gambar 2.2 Klasifikasi Sistem

2.2 Pengertian Pembelian

Istilah purchasing atau pembelian sinonim dengan procurement atau pengadaan barang. Berikut adalah definisi procurement menurut Bodnar dan Hopwood (2001:323), yaitu: "Procurement is the business process of selecting a source, ordering, and acquiring goods or services". Pendapat tersebut kurang lebih mempunyai arti: bahwa pengadaan barang adalah proses bisnis dalam memilih sumber daya-sumber daya, pemesanan dan perolehan barang atau jasa.

Menurut Mulyadi (2001:299), pembelian adalah suatu usaha yang dilakukan untuk pengadaan barang diperlukan oleh perusahaan. Jenis pemasok pembelian pun di bedakan berdasarkan pemasok, yaitu :

1. Pembelian lokal adalah pembelian dari pemasok yang berasal dari dalam negeri
2. Pembelian impor adalah pembelian dari pemasok yang berasal dari luar negeri

Jenis pembelian pun dibedakan berdasarkan transaksi, yaitu :

1. Transaksi pembelian tunai adalah jenis transaksi dimana pembayarannya dilakukan secara langsung pada saat barang diterima.
2. Transaksi pembelian kredit adalah jenis transaksi dimana pembayarannya tidak dilakukan secara langsung pada saat barang diterima, tetapi dilakukan selang beberapa waktu setelah barang diterima, sesuai perjanjian kedua belah pihak.

Pembelian adalah serangkaian tindakan untuk mendapatkan barang dan jasa melalui pertukaran, dengan maksud untuk digunakan sendiri atau dijual kembali. Pembelian merupakan sistem aplikasi siklus pengeluaran yang umum. Sistem aplikasi pembelian mencakup prosedur-prosedur pemilihan pemasok, permintaan, pembelian, penerimaan, dan pembayaran kepada pemasok. Model sistem aplikasi pembelian mencakup pemisahan fungsi-fungsi sebagai berikut ini: permintaan, pembelian, penerimaan, gudang, hutang dagang, dan buku besar. Prosedur pemilihan pemasok yang memadai merupakan faktor penting dalam keterpaduan seluruh sistem aplikasi pembelian.

Fungsi pembelian sering dianggap sebagai bagian yang paling penting dan berpengaruh, bahkan bisa dikatakan sebagian besar proses bisnis berasal dari kegiatan pembelian. Alasan yang sangat fundamental untuk membahas fungsi pembelian ialah karena dalam bidang ini pemborosan mudah terjadi, baik karena perilaku yang disfungsional maupun karena kurangnya pengetahuan dalam berbagai aspek pembelian bahan, sarana, prasarana dan suku cadang yang diperlukan perusahaan.

Pandangan ini menurut Siagian (2001:192) mudah dipahami karena dalam proses produksi perusahaan memerlukan bahan baku. Tidak banyak perusahaan

yang menguasai sendiri bahan baku yang diperlukan untuk diolah lebih lanjut menjadi produk jadi, sehingga bisa disimpulkan bahwa tidak ada satupun bentuk atau jenis perusahaan yang tidak terlibat dengan fungsi pembelian. Pengalaman banyak perusahaan bahwa biaya untuk menghasilkan suatu produk mungkin mencapai sekitar lima puluh persen (50%) dari harga jual produk, menjadikan fungsi pembelian sebagai sumber pemborosan apabila tidak diselenggarakan dengan baik dan sumber penghematan yang akan memperbesar laba perusahaan apabila dilakukan dengan teliti dan cermat.

Berikut adalah beberapa alasan mengapa pembelian merupakan area yang penting yang dikemukakan Brown, dkk (2001:131), yaitu :

1. Fungsi pembelian memiliki tanggung jawab untuk mengelola masukan perusahaan pada pengiriman, kualitas dan harga yang tepat, yang meliputi bahan baku, jasa dan *sub-assemblies* untuk keperluan organisasi.
2. Berbagai penghematan yang berhasil dicapai lewat pembelian secara langsung direfleksikan pada lini dasar organisasi. Dengan kata lain, begitu penghematan harga dibuat, maka akan mempunyai pengaruh yang langsung terhadap struktur biaya perusahaan. Sehingga sering dikatakan bahwa penghematan pembelian satu persen (1%) ekuivalen dengan peningkatan penjualan sebesar sepuluh persen (10%).
3. Pembelian dan suplai material mempunyai kaitan dengan semua aspek operasi manajemen.

Cara sebuah perusahaan dalam mengendalikan strategi pengadaan barangnya akan mempunyai pengaruh langsung terhadap bagaimana perusahaan tersebut menjalankan bisnisnya. Pembelian yang baik juga perlu menjadi

perhatian untuk organisasi-organisasi non-profit dan pemerintah. Berbagai tekanan yang berkaitan dengan kurangnya dana yang tersedia dan besarnya biaya, mendorong organisasi-organisasi tersebut untuk beroperasi seefisien mungkin dengan biaya seminimum mungkin.

Dengan begitu apapun jenis dan ukuran perusahaannya, pembelian yang dilaksanakan dengan ekonomis dan efektif amat diperlukan dalam upaya mencapai kondisi perusahaan yang sehat karena pembelian merupakan kegiatan yang memerlukan pengalokasian sumber daya dalam jumlah besar.

2.3 Sistem Pembelian

Pengertian sistem pembelian menurut Mulyadi, (2001, 299) yaitu, Sistem pembelian adalah prosedur kegiatan yang digunakan dalam perusahaan untuk pengadaan barang yang diperlukan oleh perusahaan.

Tahapan-tahapan prosedur sistem pembelian menurut Mulyadi (2001:301) sebagai berikut :

1. Prosedur permintaan pembelian

Dalam prosedur ini fungsi gudang mengajukan permintaan pembelian dalam formulir surat penerimaan pembelian kepada fungsi pembelian. Jika barang tidak disimpan di gudang, misalnya untuk barang-barang yang langsung dipakai, fungsi yang memakai barang mengajukan permintaan pembelian langsung ke fungsi pembelian dengan menggunakan surat permintaan pembelian.

2. Prosedur Permintaan Penawaran Harga Pemilihan Pemasok

Dalam prosedur ini fungsi pembelian mengirimkan surat penawaran harga kepada para pemasok untuk memperoleh informasi mengenai harga barang dan berbagai syarat pembelian yang lain, untuk memungkinkan pemilihan pemasok yang akan ditujukan sebagai pemasok barang yang diperlukan oleh perusahaan.

3. Prosedur Order Pembelian

Dalam prosedur ini fungsi pembelian mengirimkan surat order pembelian kepada pemasok yang dipilih dan memberitahukan kepada unit-unit organisasi lain dalam perusahaan (misalnya fungsi penerimaan, fungsi yang meminta barang dan fungsi pencatat utang) mengenai order pembelian yang sudah dikeluarkan oleh perusahaan.

4. Prosedur Penerimaan Barang

Dalam prosedur ini fungsi penerimaan melakukan pemeriksaan mengenai jenis, kualitas dan mutu barang yang diterima dari pemasok dan kemudian membuat laporan penerimaan barang untuk menyatakan barang dari pemasok tersebut.

5. Prosedur Pencatatan Utang

Dalam prosedur ini fungsi akuntansi memeriksa dokumen-dokumen yang berhubungan dengan pembelian (surat order pembelian, laporan penerimaan barang dan faktur dari pemasok) dan menyelenggarakan pencatatan utang atau mengarsipkan dokumen sumber sebagai catatan utang.

6. Prosedur Distribusi Pembelian

Prosedur ini meliputi distribusi rekening yang didebet dari transaksi pembelian untuk kepentingan pembuatan laporan manajemen.

2.4 Pengertian Persediaan

Pada umumnya didalam laporan tahunan, unsur harta terbesar pada neraca yaitu persediaan. Persediaan merupakan harta perusahaan yang paling sensitif sehingga perlu diamankan dari kemungkinan pencurian, pemborosan, kerusakan dan sebab lain yang menyebabkan kerugian.

Istilah persediaan dikemukakan oleh Donal E. Kieso, Jerry J. Weygant dan Terry D. Warfield yang diterjemahkan oleh Emil Salim (2007:402), Persediaan adalah pos-pos aktiva yang dimiliki oleh perusahaan untuk dijual dalam operasi bisnis normal atau barang yang akan digunakan atau dikonsumsi dalam membuat barang yang akan dijual.

Persediaan didefinisikan sebagai barang jadi yang tersimpan atau digunakan untuk dijual pada periode mendatang, yang dapat berbentuk bahan baku yang disimpan untuk proses, barang dalam proses manufaktur dan barang jadi yang disimpan untuk dijual maupun diproses.

Dari definisi di atas, dapat disimpulkan bahwa persediaan adalah material yang berupa bahan baku, barang setengah jadi, atau barang jadi yang disimpan dalam suatu tempat atau gudang dimana barang tersebut menunggu untuk diproses atau diproduksi lebih lanjut.

2.5 Testing dan Implementasi Sistem

Menurut Standar ANSI/IEEE 1059, Testing adalah proses menganalisa suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*defects/error/bugs*) dan mengevaluasi fitur-fitur dari entitas *software*.

Menurut Romeo (2003:3), *Testing software* adalah proses mengoperasikan software dalam suatu kondisi yang dikendalikan untuk :

1. Verifikasi.

Apakah telah berlaku sebagaimana yang ditetapkan (menurut spesifikasi)?

2. Mendeteksi error.

3. Validasi.

Apakah spesifikasi yang ditetapkan telah memenuhi keinginan atau kebutuhan pengguna yang sebenarnya?

Menurut Romeo (2003:33), *Test Case* merupakan tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi, ataupun hasil yang telah ditentukan sebelumnya. Metode testing ini dibagi menjadi dua, yaitu :

2.5.1 White Box Testing

White box testing atau *glass box testing* atau *clear box testing* adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain procedural. Metode desain *test case* ini dapat menjamin :

1. Semua jalur (path) yang independen/terpisah dapat dites setidaknya sekali tes.
2. Semua logika keputusan dapat dites dengan jalur yang salah atau jalur yang benar.
3. Semua *loop* dapat dites terhadap batasannya dan ikatan operasionalnya.
4. Semua struktur internal data dapat dites untuk memastikan validasinya.

2.5.2. Black Box Testing

Black box testing atau *behavioral testing* atau *specification-based testing*, *input/output testing* atau *functional testing* dilakukan tanpa sepengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black box testing* berfokus

pada kebutuhan fungsional pada software, berdasarkan spesifikasi kebutuhan dari *software*.

Menggunakan *black box testing*, perancang *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program. Kategori *error* dapat diketahui melalui *black box testing*, antara lain :

1. Fungsi yang hilang atau tidak benar.
2. Error dari antar-muka.
3. *Error* dari struktur data atau akses *eksternal database*.
4. *Error* dari kinerja atau tingkah laku.
5. *Error* dari inisialisasi dan terminasi.

2.6 Microsoft SQL Server

SQL (*Structured Query Language*) server adalah suatu sistem *Database Management Sistem* (DBMS) yang memberikan suatu cara untuk menyimpan dan mengelola informasi SQL server merupakan prosuk database relasional karena SQL server mengijinkan kita untuk menghubungkan data dari sumber-sumber yang berbeda. Dalam SQL server tabel-tabel yang digunakan untuk menyimpan informasi dan objek-objek tambahan yang mewakili informasi dan bekerja sebagai bagian dari database. Berbeda dengan sistem database yang lain seperti dBase yang menyimpan data secara terpisah antara tabel, *report*, dan *form*, SQL server menyimpan ketiga elemen atau tipe data tersebut dalam satu database.

SQL server menyimpan data di dalam tabel yang dikelompokkan dengan baris-baris dan kolom-kolom. Suatu database SQL server dapat berisi satu tabel

atau lebih. Dengan menggunakan *interface* yang berbasis grafik karena ada dalam lingkungan sistem operasi windows, SQL server membuat pembuatan suatu aplikasi pengelolaan suatu sistem database menjadi cukup mudah.

SQL server adalah manager database relational yang berbasis windows yang mengeksploitasi lingkungan windows. SQL server bukan tulisan ulang dari database DOS yang ada sebelumnya SQL server murni didesain untuk lingkungan windows.

Pada umumnya semua engine database (termasuk MySQL) mengadopsi bahasa standar SQL yaitu bahasa yang digunakan untuk memanipulasi dan memperoleh data dari sebuah database relasional. SQL membuat seorang developer atau administrator database melakukan hal-hal berikut :

- a. Mengubah struktur sebuah database,
- b. Mengubah pengaturan keamanan sistem,
- c. Memberikan hak akses kepada pengguna untuk mengakses database atau table,
- d. Memperoleh informasi dari database.

Perintah-perintah SQL secara umum dapat dikelompokkan menjadi lima macam, yaitu :

1. *Data Definition Language (DDL)*

Adalah perintah SQL yang digunakan untuk menjelaskan objek database. Dengan kata lain DDL digunakan untuk mendefinisikan kerangka database.

Prinsipnya adalah :

- a. *Create* : untuk membuat/menciptakan objek database
- b. *Alter* : untuk memodifikasi/objek database
- c. *Drop* : untuk menghapus objek database

d. Objek *database* yang dimaksud terdiri dari database, tabel, *index*, dan *view*

2. *Data Manipulating Language (DML)*

Adalah perintah yang digunakan untuk mengoperasikan atau memanipulasi isi database. SQL menyediakan 4 perintah DML :

- a. *Select* : digunakan untuk mengambil data dari database
- b. *Delete* : digunakan untuk menghapus data pada database
- c. *Insert* : menambahkan data ke database
- d. *Update* : memodifikasi data ke database

3. *Security*

Adalah perintah-perintah yang digunakan untuk menjamin keamanan data.

Antara lain terdiri atas :

- a. *Grant* : member akses kepada user tertentu untuk akses ke database
- b. *Revoke* : mencabut hak akses dari user

4. *Integrity*

Adalah perintah-perintah yang digunakan untuk menjaga kesatuan data.

Contoh : *recover table* : untuk memperbaiki tabel pada database

5. *Auxilliary*

Adalah perintah-perintah pelengkap atau tambahan seperti : *unload* dan *rename*.

2.7 Entity Relationship Diagram

ERD digunakan untuk menginterpretasikan, menentukan dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan database.

ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari pemakai.

Adapun elemen-elemen dari ERD ini adalah :

1. *Entitas*, adalah sesuatu yang dapat diidentifikasi di dalam lingkungan kerja pemakai, suatu yang penting bagi pemakai dari sistem yang akan dikembangkan
2. *Atribut*, entitas memiliki atribut yang berfungsi untuk menjelaskan karakteristik dari entitas.
3. Pengidentifikasi, data-data entitas memiliki nama yang berfungsi untuk mengidentifikasikan mereka. Sebuah identifikasi dapat bersifat unik atau tidak unik.
4. Relasi atau hubungan yang berfungsi untuk menunjukkan hubungan satu entitas dengan entitas yang lain. Hubungan ini boleh memiliki atribut. Banyaknya entitas dalam suatu relasi menunjukkan tingkat dari relasi bersangkutan, namun yang banyak digunakan dalam aplikasi-aplikasi adalah model yang menggunakan relasi tingkat 2 atau yang disebut dengan hubungan biner. Hubungan biner ini memiliki tiga tipe yaitu :
 - a) hubungan biner satu ke satu (*one to one*) yaitu suatu record pada satu tabel dihubungkan ke suatu *record* pada tabel ke kedua.
 - b) hubungan biner satu ke banyak (*one to many*) yaitu suatu *record* pada satu table dihubungkan ke suatu record atau lebih pada tabel ke dua.
 - c) hubungan biner banyak ke banyak (*many to many*) yaitu setiap *record* pada tabel pertama berhubungan dengan setiap *record* pada tabel kedua.

2.8 Data Flow Diagram

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan professional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD sering juga disebut juga dengan nama *Bubble Chart*, *Bubble Diagram*, model proses, diagram alur kerja, atau model fungsi.

Data Flow Diagram (DFD) ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks daripada data yang dimanipulasi oleh sistem. Dengan kata lain, *Data Flow Diagram* adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. *Data Flow Diagram* ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisis maupun rancangan sistem yang mudah dikomunikasikan oleh professional sistem kepada pemakai maupun pembuat program. *Data Flow Diagram* terdiri dari *context diagram* dan diagram rinci (*Data Flow Diagram Levelled*). *Context diagram* berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. *Data Flow Diagram levelled* menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi.

Dalam *Data Flow Diagram levelled* akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. Jadi dalam *Data Flow Diagram levelled* bisa dimulai dari *Data Flow Diagram* level 0 kemudian turun ke *Data Flow Diagram* level 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level x harus berhubungan dengan aliran data yang masuk dan keluar pada level $x+1$ yang mendefinisikan proses pada level x tersebut. Proses yang tidak dapat diturunkan/dirinci lagi dikatakan primitif secara fungsional dan disebut sebagai proses primitif.

Data Flow Diagram berfungsi untuk menggambarkan proses aliran data yang terjadi di dalam sistem dari tingkat yang tertinggi sampai yang terendah, yang memungkinkan untuk melakukan dekomposisi, mempartisi, atau membagi sistem ke dalam bagian-bagian yang lebih kecil dan yang lebih sederhana.

Tempat penyimpanan data digunakan untuk menyimpan data hasil proses maupun menyediakan data yang akan diproses. Proses adalah kegiatan pemrosesan suatu sistem informasi *input* dan *output*, suatu proses harus memiliki input dan output. Tanda panah berfungsi untuk menunjukkan aliran data diantara proses penyimpanan, dan entitas. Sedangkan entitas adalah elemen-elemen yang menunjukkan asal dari masukkan ke dalam sistem dan menerima keluaran dari sistem.

2.9 Visual Basic .Net

Visual Basic 2005 adalah salah satu bahasa pemrograman yang ditargetkan dalam .NET Framework. Seperti bahasa sehari-hari, Visual Basic sintaks dan beberapa kata-kata yang valid yang bisa digunakan dalam membuat aplikasi. Visual Basic merupakan pilihan yang populer bagi yang mulai belajar pemrograman karena sintaks penulisan kodenya mudah dibandingkan dengan bahasa pemrograman yang lain.

Microsoft Visual Basic .Net adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.Net Framework*, dengan menggunakan bahasa *Basic*. Dengan menggunakan alat ini, para programmer dapat membangun aplikasi Windows Forms, aplikasi *website* berbasis ASP.NET dan juga aplikasi *command-line*. Bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic versi sebelumnya yang diimplementasikan di atas .NET Framework.

2.10 Konsep Dasar Basis Data

2.10.1 Database

Menurut Marlinda (2004:1), *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Penyusunan satu *database* digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu redundansi dan inkonsistensi data, kesulitan pengaksesan data, isolasi data untuk standarisasi, *multiple user* (banyak pemakai), masalah keamanan (*security*), masalah integrasi (kesatuan), dan masalah data *independence* (kebebasan data).

2.10.2 Sistem Basis Data

Menurut Marlinda (2004:1), sistem basis data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan atau merekam serta memelihara dan operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Pada sebuah sistem basis data terdapat komponen-komponen utama yaitu perangkat keras (*hardware*), Sistem Operasi (*Operating Sistem*), Basis Data (*Database*), Sistem (Aplikasi atau Perangkat Lunak) Pengelola Basis Data (DBMS), Pemakai (*User*), dan Aplikasi (Perangkat Lunak) lain (bersifat opsional).

Keuntungan sistem basis data adalah :

1. Mengurangi kerangkapan data, yaitu data yang sama disimpan dalam berkas data yang berbeda-beda sehingga *update* dilakukan berulang-ulang.
2. Mencegah ketidakkonsistenan.
3. Keamanan data dapat terjaga, yaitu data dapat dilindungi dari pemakai yang tidak berwenang.
4. Integritas dapat dipertahankan.
5. Data dapat dipergunakan bersama-sama.

6. Menyediakan *recovery*.
7. Memudahkan penerapan standarisasi.
8. Data bersifat mandiri (*data independence*).
9. Keterpaduan data terjaga, memelihara keterpaduan data berarti data harus akurat. Hal ini sangat erat hubungannya dengan pengontrolan kerangkapan data dan pemeliharaan keselarasan data.

Kerugian sistem basis data adalah :

1. Diperlukan tempat penyimpanan yang besar.
2. Diperlukan tenaga yang terampil dalam mengolah data.
3. Perangkat lunaknya mahal.
4. Kerusakan sistem basis data dapat mempengaruhi departemen yang terkait.

2.10.3 Database Management Sistem

Menurut Marlinda (2004:6), *Database Management Sistem (DBMS)* merupakan kumpulan file yang saling berkaitan dan program untuk pengelolanya. Basis Data adalah kumpulan datanya, sedangkan program pengelolanya berdiri sendiri dalam suatu paket program yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data.

Bahasa-bahasa yang terdapat dalam DBMS adalah :

1. *Data Definition Language (DDL)*

Pola skema basis data dispesifisikan dengan satu set definisi yang diekspresikan dengan satu bahasa khusus yang disebut DDL. Hasil kompilasi perintah DDL adalah satu set tabel yang disimpan di dalam file khusus yang disebut *data dictionary/directory*.

2. *Data Manipulation Language (DML)*

Bahasa yang memperbolehkan pemakai mengakses atau memanipulasi data sebagai yang diorganisasikan sebelumnya model data yang tepat.

3. *Query*

Pernyataan yang diajukan untuk mengambil informasi. Merupakan bagian DML yang digunakan untuk pengambilan informasi.

DBMS memiliki fungsi sebagai berikut :

1. *Data Definition*

DBMS harus dapat mengolah pendefinisian data.

2. *Data Manipulation*

DBMS harus dapat menangani permintaan-permintaan dari pemakai untuk mengakses data.

3. *Data Security dan Integrity*

DBMS dapat memeriksa *security* dan *integrity* data yang didefinisikan oleh DBA.

4. *Data Recovery dan Concurrency*

- a. DBMS harus dapat menangani kegagalan-kegagalan pengaksesan basis data yang dapat disebabkan oleh kesalahan sistem, kerusakan disk, dan sebagainya.
- b. DBMS harus dapat mengontrol pengaksesan data yang konkuren yaitu bila satu data diakses secara bersama-sama oleh lebih dari satu pemakai pada saat yang bersamaan.

5. *Data Dictionary*

DBMS harus menyediakan data *dictionary*.

2.11 SQL Server Data Access Componen (SDAC)

SQL Server Data Access Components (SDAC) adalah library komponen yang menyediakan akses ke database Microsoft SQL Server. SDAC menghubungkan ke SQL Server langsung melalui OLE DB, yang merupakan antarmuka SQL Server asli. Perpustakaan SDAC ini dirancang untuk membantu programmer mengembangkan aplikasi server database SQL lebih cepat dan bersih. SDAC adalah pengganti solusi lengkap untuk server standar konektivitas SQL dan menyajikan alternatif yang efisien dengan BDE (*Borland Database Engine*) untuk mengakses SQL Server. SDAC Palette Keuntungan Teknologi SDAC. SDAC adalah database wrapper konektivitas langsung dibangun khusus untuk server SQL Server. SDAC menawarkan jangkauan yang luas dari fitur SQL Server mengatur dan menekankan strategi akses data dioptimalkan luas cakupan fitur SQL Server.

Dengan memberikan akses ke fungsionalitas database yang paling canggih, SDAC memungkinkan pengembang untuk memanfaatkan kemampuan penuh dari SQL Server dan aplikasi database mengoptimalkan mereka. SDAC menyediakan dukungan lengkap bekerja dengan SQL Server 2005 Compact Edition, Jasa Broker teknologi, interface IRowsetFastLoad, bekerja dengan informasi metadata, dan MARS kode dioptimalkan. Tujuan dari SDAC adalah untuk memungkinkan pengembang untuk menulis aplikasi database yang efisien dan fleksibel. Perpustakaan SDAC diimplementasikan dengan menggunakan algoritma akses data lanjutan dan teknik optimasi. Kelas dan komponen menjalani tes kinerja yang komprehensif dan dirancang untuk membantu menulis kinerja tinggi, lapisan data ringan akses, kompatibilitas dengan metode konektivitas lain.

Antarmuka SDAC mempertahankan kompatibilitas dengan standar komponen akses data VCL seperti BDE (*Borland Database Engine*). Ada aplikasi berbasis BDE (*Borland Database Engine*) dapat dengan mudah bermigrasi ke SDAC dan ditingkatkan untuk memanfaatkan fitur SQL Server-spesifik. Proyek migrasi dapat diotomatisasi dengan BDE Migrasi / ADO Wizard. SDAC terhubung ke SQL Server melalui OLE DB, yang merupakan terendah didokumentasikan antarmuka SQL Server. SDAC menghubungkan melalui OLE DB melalui satu set COM interface berbasis. SDAC dirancang untuk menjadi ringan dan terdiri dari lapisan minimal antara kode Anda dan database SQL Server. Sebagai perbandingan, *Borland Database Engine* (BDE) menggunakan beberapa lapis untuk mengakses SQL Server, dan memerlukan software tambahan akses data yang akan diinstal pada mesin klien.

