

BAB II

TINJAUAN PUSTAKA

II.1.Sistem Informasi

II.1.1. Sistem

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

Kata Sistem mempunyai beberapa pengertian, tergantung dari sudut pandang mana kata tersebut didefinisikan. Secara garis besar ada dua kelompok pendekatan, yaitu :

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang dalam hal ini sistem itu didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu aturan tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem, Prosedur (*procedure*) didefinisikan oleh Richard F. Neushl sebagai urutan operasi kerja (tuliskan-menulis), yang biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Pendekatan sistem yang menekankan pada elemen-elemen atau komponen-komponen mendefenisikan sistem sebagai berikut : “Sistem adalah kumpulan dari elemen yang saling berinteraksi untuk mencapai tujuan”. Dengan demikian di dalam suatu sistem, komponen-komponen ini tidak dapat berdiri sendiri-sendiri tetapi sebaliknya saling berhubungan sehingga membentuk satu kesatuan sehingga tujuan sistem itu dapat tercapai.(Kusrini & Andri Koniyo; 2007:5)

II.1.2. Karakteristik Sistem

Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu, antara lain:

1. **Komponen Sistem (*Component*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang saling bekerja sama membentuk suatu komponen sistem atau bagian-bagian dari sistem.

2. **Batasan Sistem (*Boundary*)**

Merupakan daerah yang membatasi suatu sistem dengan sistem yang lain atau dengan lingkungan kerjanya.

3. **Subsistem**

Bagian-bagian dari sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasarannya masing-masing.

4. **Lingkungan Luar Sistem (*Environment*)**

Suatu sistem yang ada diluar dari batas sistem yang dipengaruhi oleh operasi sistem.

5. Penghubung Sistem (*Interface*)

Media penghubung antara suatu subsistem dengan subsistem lain. Adanya penghubung ini memungkinkan berbagai sumber daya mengalir dari suatu subsistem ke subsistem lainnya.

6. Masukan Sistem (*Input*)

Energi yang masuk ke dalam sistem, berupa perawatan dan sinyal. Masukan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat berinteraksi.

7. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

8. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.

9. Sasaran Sistem (*Object*)

Tujuan yang ingin dicapai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran atau tujuan.(Kusrini & Andri Koniyo;2007:6)

II.1.3. Informasi

Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai sedangkan

informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatkannya.(Kusrini&Andri Koniyo; 2007:7)

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. (Jogiyanto HM ; 2005 : 8)

Adapun kualitas dari informasi adalah sebagai berikut :

1. Akurat (*accurate*)
2. Tepat waktu (*timelines*)
3. Relevan (*relevance*)

II.1.4 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.” (Jogiyanto HM ; 2005 : 11)

“Sistem informasi disebut sebagai sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen baik manual ataupun berbasis komputer yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi kepada pihak-pihak yang berkepentingan sebagai pemakai informasi tersebut.” (Anastasia Diana & Lilis Setiawati ; 2011 : 4)

II.1.5. Komponen Sistem Informasi

Komponen-komponen sistem informasi adalah sebagai berikut:

1. Perangkat keras (*hardware*), mencakup berbagai peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang, yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
5. Bais data (*database*), yaitu sekumpulan tabel, hubungan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikas yang data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.(Kusrini & Andri Koniyo ; 2007:9)

II.2. Sistem Informasi Akuntansi

II.2.1 Akuntansi

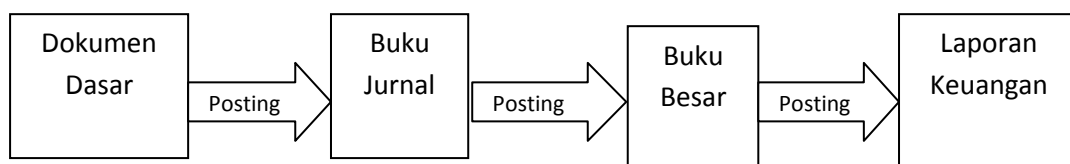
Akuntansi merupakan bahasa dari bisnis. Setiap perusahaan menerapkannya sebagai alat komunikasi. Secara klasik akuntansi merupakan proses pencatatan (*recording*), pengelompokkan (*classifying*), perangkuman

(*summarizing*), dan pelaporan (*reporting*) dari kegiatan transaksi perusahaan.(
Jogiyanto HM ; 2005 : 17)

Akuntansi adalah aktivitas mengumpulkan, menganalisis, menyajikan dalam bentuk angka, mengklasifikasikan, mencatat, meringkas dan melaporkan aktivitas/transaksi perusahaan dalam bentuk informasi keuangan.(Rudianto; 2009:14).

Untuk sampai pada penyajian informasi keuangan yang dibutuhkan berbagai pihak , maka akuntansi harus melewati suatu proses yang disebut dengan siklus akuntansi. Siklus akuntansi adalah urutan kerja yang harus dibuat oleh akuntan, sejak awal hingga menghasilkan laporan keuangan suatu perusahaan.

Adapun gambaran umum dari siklus akuntansi adalah sebagai berikut:



Gambar II.1. Siklus Akuntansi

Sumber : Pengantar Akuntansi (Rudianto : 2009)

Keterangan :

1. Dokumen Dasar adalah bukti transaksi yang dijadikan dasar oleh akuntan untuk mencatat, seperti: faktur, kuitansi, nota penjualan,dll.

2. Jurnal (*Journal*) adalah aktivitas meringkas dan mencatat transaksi perusahaan berdasarkan dokumen dasar. Tempat untuk mencatat dan meringkas tersebut disebut dengan buku jurnal.
3. *Posting* adalah aktivitas memindahkan catatan di buku jurnal kedalam buku besar sesuai dengan jenis transaksi dan nama perkiraan masing-masing.
4. Buku Besar (*General Ledger*) adalah kumpulan dari semua akun/perkiraan yang dimiliki suatu perusahaan yang saling berhubungan satu dengan lainnya dan merupakan suatu kesatuan.
5. Akun/perkiraan (*Account*) adalah suatu kelas informasi di dalam suatu sistem akuntansi. Atau suatu media yang digunakan untuk mencatat informasi sumber daya perusahaan dan informasi lainnya berdasarkan jenisnya. Misalnya perkiraan kas, perkiraan piutang, akun modal, dan sebagainya.

II.2.2 Sistem Informasi Akuntansi

Menurut Stephen A.Moscove dan Mark G.Simkin Sistem Informasi Akuntansi adalah suatu komponen organisasi yang mengumpulkan, mengklasifikasikan, memproses, menganalisis, mengkomunikasikan informasi pengambilan keputusan dengan orientasi finansial yang relevan bagi pihak-pihak luar dan pihak-pihak dalam perusahaan. Sedangkan Menurut Robert G.Murdick, Thomas C.Fuller dan Joel E.Ross Sistem Informasi Akuntansi adalah kumpulan dari 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.(
Jogiyanto HM ; 2005 : 17)

Dari definisi di atas dapat disimpulkan bahwa sistem informasi akuntansi adalah sistem yang dirancang untuk pengolahan data keuangan (akuntansi) menjadi sebuah informasi.

Sistem Informasi Akuntansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya.

Tujuan dari Sistem Informasi Akuntansi adalah:

1. Mendukung operasi sehari-hari.
2. Mendukung pengambilan keputusan manajemen.
3. Memenuhi kewajiban yang berhubungan dengan pertanggung jawaban.

Komponen-komponen yang terdapat dalam Sistem Informasi Akuntansi adalah sebagai berikut :

1. Orang-orang yang mengoperasikan sistem tersebut.
2. Prosedur-prosedur, baik manual maupun yang terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
3. Data tentang proses-proses bisnis
4. *Software* yang dipakai untuk memproses data organisasi
5. Infrastruktur teknologi informasi

Sementara itu aktivitas utamanya adalah :

1. *Indbound Logistics*: penerimaan, penyimpanan dan distribusi bahan masukan.
2. Operasi:aktivitas untuk mengubah masukan menjadi barang atau jasa.
3. *Outbound Logistics*: distribusi produk ke pelanggan
4. Pemasaran dan Penjualan
5. Pelayanan:berupa dukungan purnajual dan *maintenance*.(Kusrini & Andri Koniyo ; 2007:10)

II.2.3 Siklus dalam Sistem Informasi Akuntansi

Sistem Informasi Akuntansi memiliki beberapa sistem bagian (sub-system) yang berupa siklus akuntansi. Siklus akuntansi menunjukkan prosedur akuntansi, mulai dari sumber data sampai ke proses pencatatan/pengolahan akuntansinya. Berikut ini adalah pembagian dari siklus akuntansi.

1. Siklus Pendapatan

Siklus pendapatan merupakan prosedur pendapatan yang dimulai dari bagian penjualan otorisasi kredit, pengambilan barang, penerimaan barang, penagihan sampai dengan penerimaan kas.

2. Siklus Pengeluaran Kas

Siklus pengeluaran kas merupakan prosedur pengeluaran kas yang dimulai dari proses pembelian sampai ke proses pembayaran.

3. Siklus Konversi

Siklus konversi merupakan siklus produksi, dimulai dari bahan mentah sampai barang jadi.

4. Siklus Manajemen Sumber Daya Manusia

Siklus manajemen sumber daya manusia merupakan siklus yang melibatkan proses penggajian pada karyawan.

5. Siklus Buku Besar dan Laporan Keuangan

Siklus ini berupa prosedur pencatatan dan perekaman ke jurnal dan buku besar dan pencetakan laporan keuangan yang datanya diambil dari buku besar. (Kusrini & Andri Koniyo ; 2007 : 11-12)

II.3. Pengertian Arus Kas

Arus Kas (*cash flow*) adalah suatu laporan keuangan yang berisikan pengaruh kas dari kegiatan operasi, kegiatan transaksi investasi dan kegiatan transaksi pembiayaan/pendanaan serta kenaikan atau penurunan bersih dalam kas suatu perusahaan selama satu periode.

Menurut Pernyataan Standar Akuntansi Keuangan (PSAK) No.2 (2002 :5) Arus Kas adalah arus masuk dan arus keluar kas atau setara kas. Laporan arus kas merupakan revisi dari mana uang kas diperoleh perusahaan dan bagaimana merekan membelanjakannya. Laporan arus kas merupakan ringkasan dari penerimaan dan pengeluaran kas perusahaan selama periode tertentu (biasanya satu tahun buku).

Laporan arus kas (*cash flow*) mengandung dua macam aliran/ arus kas yaitu :

1. *Cash inflow* adalah arus kas yang terjadi dari kegiatan transaksi yang melahirkan keuntungan kas (penerimaan kas). Arus kas masuk (*cash inflow*) terdiri dari :
 - a. Hasil penjualan produk/jasa perusahaan
 - b. Penagihan piutang dari penjualan kredit
 - c. Penjualan aktiva tetap yang ada
 - d. Penerimaan investasi dari pemilik atau saham bila perseroan terbatas
 - e. Pinjaman/hutang dari pihak lain
 - f. Penerimaan sewa dan pendapatan lain.
2. *Cash out flow* adalah arus kas yang terjadi dari kegiatan transaksi yang mengakibatkan beban pengeluaran kas. Arus kas keluar (*cash out flow*) terdiri dari :
 - a. Pengeluaran biaya administrasi umum dan administrasi penjualan.
 - b. Pengeluaran biaya bahan baku, tenaga kerja langsung dan biaya lain-lain.
 - c. Pembelian aktiva tetap.
 - d. Pembayaran hutang-hutang perusahaan.
 - e. Pembayaran kembali investasi dari pemilik perusahaan
 - f. Pembayaran sewa, pajak, deviden, bunga dan pengeluaran lain-lain.

Laporan arus kas ini memberikan informasi yang relevan tentang penerimaan dan pengeluaran kas dari perusahaan dari suatu periode tertentu, dengan mengklasifikasikan transaksi berdasarkan pada kegiatan operasi, investasi dan pendanaan.

Menurut Pernyataan Standar Akuntansi Keuangan (PSAK) No.2 (2002:9) Laporan arus kas harus melaporkan arus kas selama periode tertentu yang diklasifikasikan menurut aktivitas operasi, investasi, dan pendanaan.(<http://www.ilmu-ekonomi.com/2012/04/pengertian-arus-kas-cash-flow.html>)

II.4. Visual Basic

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 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 pembuatan, penggunaan aplikasim 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 mengembangkan aplikasi.

Lingkungan kerja *Visuaal Basic* atau disebut *Integrated Development Environment (IDE)* adalah suatu lingkungan kerja tempat programmer melakukan pemograman yang didukung oleh *compiler*, *editor* baik editor grafis maupun kode, dan lain sebagainya untuk memudahkan pemograman.

II.5. SQL Server

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. (Wahana Komputer; 2010 :2)

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk *Microsoft*. Bahasa kueri utamanya *Transact SQL* yang merupakan implementasi dari *SQL* standar ANSI/ISO yang digunakan oleh *Microsoft* dan *Sybase*. Umumnya *SQL Server* digunakan di dunia bisnis yang memiliki basis data bersekala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya *SQL Server* pada basis data besar.

Microsoft SQL Server dan *Sybase/ASE* dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (*Tabular Data Stream*). Selain dari itu, *Microsoft SQL Server* juga mendukung *ODBC* (*Open Database Connectivity*), dan mempunyai *driver JDBC* untuk bahasa pemrograman *Java*. Fitur yang lain dari *SQL Server* ini adalah kemampuannya untuk membuat basis data *mirroring* dan *clustering*. Pada versi sebelumnya, *Microsoft SQL Server 2000* terserang oleh cacing komputer *SQL Slammer* yang mengakibatkan kelambatan akses internet pada tanggal 25 Januari 2003.

SQL Server 2005 (sebelumnya diberi kode “Yukon”) dirilis pada Oktober 2005. Ini termasuk dukungan asli untuk mengelola data XML, selain data relasional. Untuk tujuan ini, mendefinisikan sebuah tipe data xml yang dapat digunakan baik sebagai tipe data dalam kolom database atau sebagai literal dalam permintaan.

SQL Server 2005 memungkinkan untuk mengintegrasikan dengan NET Framework. SQL Server 2005 juga memperkenalkan “MARS” (Multiple Active Hasil Set), sebuah metode yang memungkinkan penggunaan koneksi database untuk beberapa tujuan. (http://id.wikipedia.org/wiki/Microsoft_SQL_Server)

II.6. Konsep UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membandu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP). (Martin Fowler ; 2005:1)

Unified Modeling Language (UML) adalah salah satu alat bantu yang sangat handal di dunia perkembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi perkembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan dengan yang lain (Munawar ; 2005:17).

II.6.1 Diagram – diagram pada Metode UML

1. Use Case Diagram

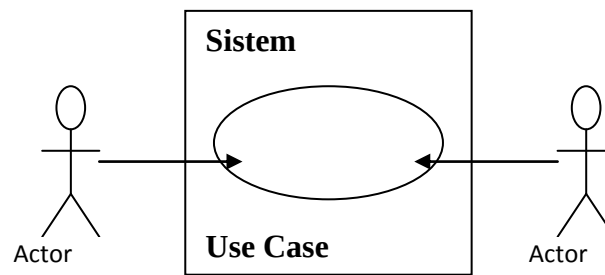
Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna.

Use case bekerja dengan cara deskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem yang disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras dan urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian *scenario* yang digabungkan bersama-sama oleh pengguna tujuan umum pengguna.

Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan *actor*. *Actor* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

Model *use case* adalah bagian dari model *requirement*. Termasuk disini adalah problem domain object dan penjelasan tentang *user interface*. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif user.

Notasi *use case* menunjukkan 3 aspek dari sistem yaitu *actor use case* dan *system / sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*. Ilustrasi *actor*, *use case* dan *system* ditunjukkan pada gambar II.2.



Gambar II.2. Use Case Diagram

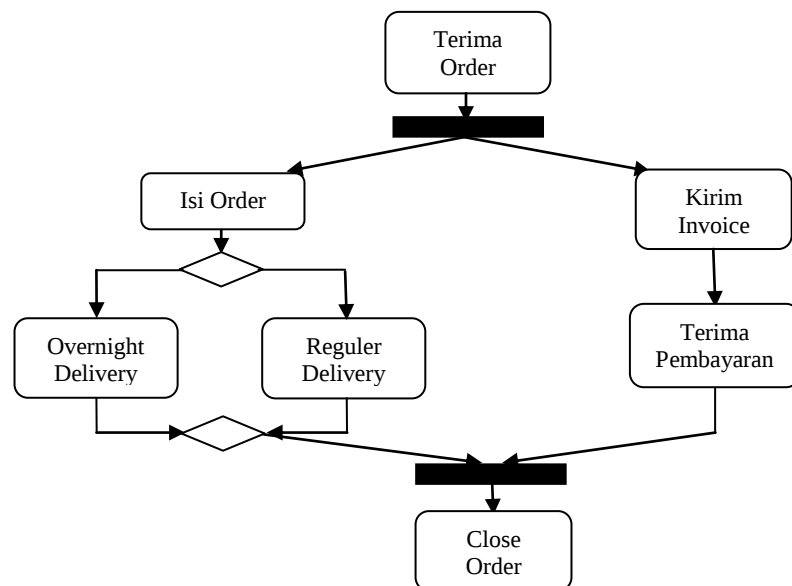
Sumber : (Munawar:2005 : 64)

Untuk mengidentifikasi *actor*, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. *Actor* adalah *abstraction* dari orang dan sistem yang lain mengaktifkan fungsi dari target sistem. Orang atau sistem bila muncul dalam beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*.

Use case adalah abstraksi dari interaksi antara sistem dan *actor*. Oleh karena itu sangat penting untuk memilih abstraksi yang cocok. *Use case* dibuat berdasarkan keperluan *actor*. *Use case* harus merupakan 'apa' yang dikerjakan *software* aplikasi, bukan 'bagaimana' *software* aplikasi mengerjakannya. Setiap *use case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan *actor*. Namun *use case* boleh terdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama (Munawar ; 2005 : 63-66).

2. Activity diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa (Munawar ; 2005 : 109). Berikut gambar *activity diagram* sederhana.



Gambar II.3.Contoh Activity Diagram Sederhana

Sumber : (Munawar ; 2005 : 111)

3. Class Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Kelas memiliki apa yang disebut *Atribut* dan metode atau operasi :

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Susunan kelas suatu sistem yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main, kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem, kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
3. Kelas yang diambil dari pendefinisian *use case*, kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.
4. Kelas yang diambil dari pendefinisian data, kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

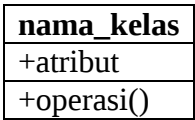
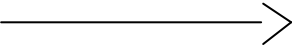
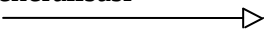
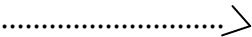
Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohension* dan *coupling*. *Cohension* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan yang lain dalam sebuah kelas. Sebagai aturan

secara umum maka sebuah metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah (Rosa A.S dan M. Shalahuddin ; 2011 : 122-123).

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel II.1. Struktur Tabel Simbol Class Diagram

Simb	Deskripsi
kelas 	Kelas Pada struktur sistem
Antarmuka / <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>assosiation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus)
Keberuntungan / <i>dependency</i> 	Relasi antar kelas dengan makna ketergantungan antar kelas
Agregasi / <i>agregation</i>	Relasi antar kelas dengan makna

Sumber : (Rosa A.S dan M. Shalahuddin ; 2011 : 23)

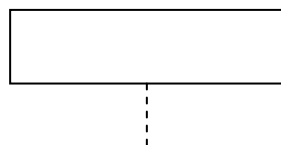
4. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sebuah contoh objek dan pesan yang diletakkan diantara objek-objek ini didalam *use case*.

Komponen utama *Sequence diagram* terdiri dari atas objek yang dituliskan dengan kotak segiempat bernama. *Messege* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*(Munawar ; 2005 : 87).

a. Objek / *participant*

Objek diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram. Setiap *participant* dihubungkan garis titik-titik yang disebut *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*. *Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*. *Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*. Bentuk *participant* dapat dilihat pada gambar II.4



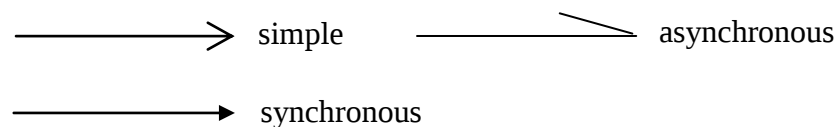
Gambar II.4. Bentuk *Participant*

Sumber : (Munawar ; 2005 : 88)

b. *Messege*

Sebuah *messege* bergerak dari suatu *participant* ke *participant* yang lain dan dari *lifeline* ke *lifeline* yang lain. Sebuah *participant* bisa mengirim sebuah *message* kepada dirinya sendiri.

Sebuah *message* bisa jadi *simple*, *synchronous* atau *asynchronous*. *Message* yang *simple* adalah sebuah perpindahan (transfer), contoh dari satu *participant* ke *participant* yang lainnya. Jika suatu *participant* mengirimkan sebuah *message* tersebut akan ditunggu sebelum di proses dengan urusannya. Namun jika *message asynchronous* yang dikirimkan, maka jawabannya atas *message* tersebut tidak perlu ditunggu. Simbol *message* pada *sequence diagram* dapat dilihat pada gambar II.5



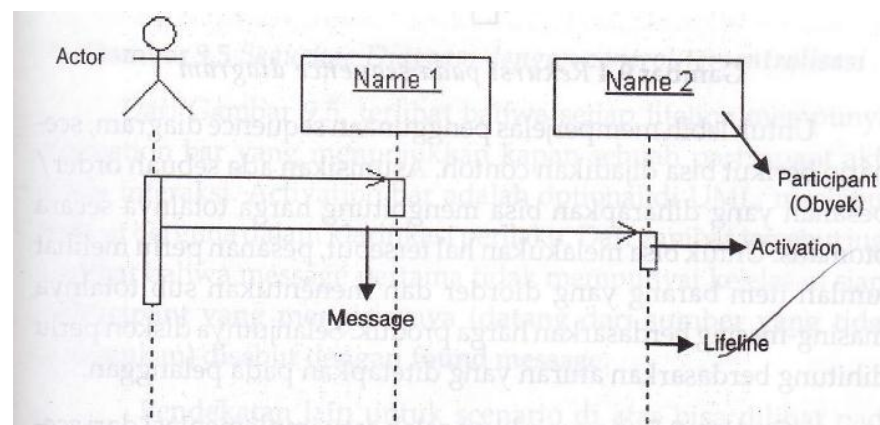
Gambar II.5. Bentuk Message

Sumber : (Munawar ; 2005 : 88)

c. *Time*

Time adalah diagram yang mewakili waktu pada arah vertikal. Waktu dimulai dari atas ke bawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibanding *message* yang lebih dekat ke bawah.

Terdapat dua dimensi pada *sequence diagram* yaitu dimensi dari kiri ke kanan menunjukkan tata letak *participant* dan dimensi dari atas ke bawah menunjukkan lintasan waktu. Simbol-simbol yang ada pada *sequence diagram* ditunjukkan pada gambar II.6



Gambar II.6.Bentuk Time

Sumber : (Munawar ; 2005 : 89)

II.7. Konsep Sistem Basis Data (*Database*)

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer, data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. *Database* sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu. Misalnya dari data nama siswa dan tanggal lahir siswa untuk bisa mendapatkan informasi siswa yang berulang tahun pada hari ini. Tentu saja informasi tersebut akan didapatkan dari *software* pemroses *database* dengan cara memberikan perintah dalam bahasa tertentu yaitu *SQL (Structured Query Language)*.

Pada era kemajuan teknologi seperti sekarang ini, nilai informasi sangatlah penting, terlebih bagi kemajuan perusahaan. Oleh karena itu, penggunaan dan penguasaan *database* sangat penting. Dalam *database* ada sebutan-sebutan untuk satuan data yaitu :

1. Karakter, ini adalah satuan data terkecil. Data terdiri atas susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
2. *Field*, adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir, dan lain-lain. Dalam dunia perancangan *database*, *field* juga disebut atribut. Bila dipandang dari sudut pemrograman berorientasi obyek maka sebuah *field* akan memiliki dua properti utama yaitu properti *name* dan properti *type*, Properti *name* atau nama adalah properti dari *field* yang berisi nama *field* yang mewakili data sejenis yang disimpannya. Sedangkan Properti *type* adalah properti yang mengatur tipe data dari data yang akan ditampungnya. Misalnya nama *field*nya adalah nama siswa maka tipe datanya adalah *char*, bila nama *field*nya adalah tanggal lahir maka tipe datanya adalah *date*. *Field* dilihat seperti kolom.
3. *Record*, adalah kumpulan dari *field*. Pada *record* dapat ditemukan banyak sekali informasi penting dengan cara mengombinasikan *field-field* yang ada.
4. Tabel, adalah sekumpulan dari *record-record* yang memiliki kesamaan *entity* dalam dunia nyata. Kumpulan dari tabel adalah *database*, wujud fisik sebuah *database* dalam komputer adalah sebuah *file* yang didalamnya terdapat berbagai tingkatan data yang telah disebutkan diatas. (Wahana Komputer ; 2010 : 24)

II.7.1. *Entity Relationship Diagram (ERD)*

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

Entity dapat berarti sebuah objek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen-komponen data (atribut atau *field*) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia *database entity* memiliki atribut yang menjelaskan karakteristik dari *entity* tersebut. Ada dua macam atribut yang dikenal dalam *entity* yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif. Hal ini berarti setiap *entity* memiliki himpunan yang diperlukan sebuah *primary key* untuk membedakan anggota-anggota dalam himpunan tersebut.

Atribut dapat memiliki sifat-sifat sebagai berikut :

1. *Atomic*, atomik adalah sifat dari atribut yang menggambarkan bahwa atribut tersebut berisi nilai yang spesifik dan tidak dapat dipecah lagi. Contoh dari sifat atomik adalah *field* status dari tabel karyawan yang hanya berisi menikah atau *single*.
2. *Multivalued*, sifat ini menandakan atribut ini bisa memiliki lebih dari satu nilai untuk tiap *entity* tertentu. Misalnya adalah *field* hobi, hobi dari setiap karyawan mungkin dan hampir pasti lebih dari satu. Misalnya karyawan A memiliki hobi; membaca, nonton TV, dan bersepeda.

3. *Composite*, atribut yang bersifat komposit adalah atribut yang nilainya adalah gabungan dari beberapa atribut yang bersifat atomik. Contohnya adalah atribut alamat yang dapat dipecah menjadi atribut atomik berupa alamat, kode pos, no telepon, dan kota.

Ada beberapa derajat relasi yang dapat terjadi yaitu :

1. *One to one*, menggambarkan bahwa antara 1 anggota *entity* A hanya dapat berhubungan dengan 1 anggota *entity* B. Biasanya derajat relasi ini digambarkan dengan simbol 1-1.
2. *One to many*, menggambarkan bahwa antara 1 anggota *entity* A dapat memiliki hubungan dengan lebih dari 1 anggota *entity* B. Biasanya derajat relasi ini digambarkan dengan simbol 1-N.
3. *Many to many*, menggambarkan bahwa lebih dari satu anggota A dapat memiliki hubungan dengan lebih dari satu anggota *entity* B. Simbol yang digunakan adalah N-N. (Wahana Komputer ; 2010 :30-31)

II.7.2. Normalisasi

Normalisasi adalah proses dimana tabel-tabel pada database dites dalam hal kesalingketergantungan di antara *field-field* pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu *field* dalam tabel tersebut , maka tabel tersebut harus dipecah menjadi banyak tabel. Banyaknya tabel pecahannya bergantung pada seberapa banyak ketergantungannya. Tiap tabel hanya boleh memiliki sebuah *field* kunci yang menjadi ketergantungan dari *field* lainnya dalam tabel tersebut.

Ada beberapa langkah dalam normalisasi tabel, yaitu :

1. *Decomposition*, dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenai dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di dekomposisi. Dekomposisi akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat-syaratnya.
2. Bentuk tidak normal, pada bentuk ini semua data yang ada pada tiap *entity* (diambil atributnya) masih ditampung dalam satu tabel besar. Data yang ada pada tabel ini masih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak tertata rapi.
3. *Normal Form* pertama (*1st Normal Form*), pada tahapan ini tabel di dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomik.
4. *Normal Form* kedua (*2nd Normal Form*), pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer tabel tersebut. Dalam hal ini semua tabel telah memenuhi bentuk *normal form* yang kedua.
5. *Normal Form* ketiga (*3rd Normal Form*), setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum

efisien. Biasanya penggunaan bentuk normal (normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang telah dihasilkan telah memiliki kualitas untuk membentuk sebuah *database* yang dapat diandalkan. (Wahana Komputer ; 2010: 32-35).

II.7.3 Kamus Data

Menurut Jogiyanto (2005 : 725) Kamus Data (KD) atau *Data Dictionary* (*DD*) atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan – kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap perancangan sistem, KD digunakan untuk merancang *input*, merancang laporan-laporan dan *database*. KD dibuat berdasarkan arus data yang ada di Diagram Alir Data (DAD). Arus data DAD sifatnya adalah global, hanya ditunjukkan nama arus datanya saja.

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

1. Nama Arus Data

Karena KD 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 mencarinya 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

Telah diketahui bahwa arus data dapat mengalir :

- a. Dari kesatuan luar ke suatu proses, data yang mengalir ini biasanya tercatat di suatu dokumen atau formulir.
- b. Hasil dari suatu proses ke kesatuan luar, data yang mengalir ini biasanya terdapat di media laporan atau *query* tampilan layar atau dokumen hasil cetakan komputer.
- c. Hasil suatu proses ke proses yang lain, data yang mengalir ini biasanya dalam bentuk variabel atau parameter yang dibutuhkan oleh proses penerimanya.
- d. Hasil suatu proses yang direkamkan ke simpanan data, data yang mengalir ini biasanya berbentuk suatu variabel.
- e. Dari simpanan data dibaca oleh suatu proses, data yang mengalir ini

biasanya berupa suatu *field* (item data).

4. Arus data

Arus data menunjukkan dari mana data mengalir dan ke mana data akan menuju. Keterangan arus data ini perlu dicatat di KD supaya memudahkan mencari arus data ini di DAD.

5. Penjelasan

Bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut. Sebagai misalnya nama dari arus data adalah tembusan permintaan persediaan, maka dapat lebih dijelaskan sebagai tembusan dari faktur penjualan untuk meminta barang dari gudang.

6. Periode

Periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat di KD karena dapat digunakan untuk mengidentifikasi kapan input data harus dimasukan kesistem, kapan proses dari program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

7. Volume

Volume yang perlu dicatat di KD adalah tentang volume rata-rata dan *volume* puncak dari arus data. *Volume* ini digunakan untuk mengidentifikasi besarnya simpanan luar yang akan digunakan, kapasitas dan jumlah dari alat *input*, alat pemroses dan alat output.

8. Struktur data

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