

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem informasi berbasis komputer merupakan sekelompok perangkat keras dan perangkat lunak yang dirancang untuk mengubah data menjadi informasi yang bermanfaat. Jenis sistem informasi berbasis komputer

1. Pengolahan Data. Pengolahan data *elektronik – elektronik data processing (EDP)* adalah pemanfaatan teknologi komputer untuk melakukan pengolahan data transaksi-transaksi dalam suatu organisasi. *EDP* adalah aplikasi sistem informasi akuntansi paling dasar dalam setiap organisasi. Sehubungan dengan perkembangan teknologi komputer, istilah pengolahan data mulai dikenal dan mempunyai arti yang sama dengan istilah *EDP*.
2. Sistem Informasi Manajemen (*SIM*), menguraikan penggunaan teknologi komputer untuk menyediakan informasi bagi pengambilan keputusan para manajer.
3. Sistem Pendukung Keputusan – *Decision Support Systems(DSS)*. *DSS* diarahkan untuk melayani permintaan informasi tertentu, khusus, dan tidak rutin dari manajemen. Contoh adalah penggunaan *spreadshee* untuk melakukan analisis “*whatif*” dari data operasi atau anggaran.
4. Sistem *Pakar – expertsystems (ES)* adalah sistem informasi berbasis pengetahuan yang memanfaatkan pengetahuannya tentang bidang aplikasi tertentu untuk bertindak seperti seorang konsultan ahli bagi pemakainya.

5. Sistem Informasi Eksekutif – *executive information systems (EIS)*. *EIS* dibuat bagi kebutuhan informasi strategik manajemen tingkat puncak.
6. Sistem Informasi Akuntansi – sistem berbasis komputer yang dirancang untuk mengubah data akuntansi menjadi informasi (Agustinus ; 2012 : 2).

II.2. Pengertian Informasi

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dsb.

Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi. Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem.

Secara konseptual seluruh sistem organisasional mencapai tujuannya melalui proses alokasi sumber daya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomis pada saat ia mendukung keputusan alokasi sumber daya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan.

Pemakai informasi akuntansi dapat dibagi dalam dua kelompok besar: *ekstern* dan *intern*. Pemakai ekstern mencakup pemegang saham, *investor*, *kreditor*, pemerintah, pelanggan, pemasok, pesaing, serikat pekerja, dan masyarakat. Pemakai intern terutama para manajer, kebutuhannya bervariasi tergantung pada tingkatannya (Agustinus ; 2012 : 1)

II.2.1 Pengertian Sistem Informasi

Sistem informasi bukan merupakan hal yang baru, yang baru adalah komputerisasinya. Sebelum ada komputer, teknik penyaluran informasi yang memungkinkan manajer merencanakan serta mengendalikan operasi telah ada. Komputer menambahkan satu atau dua dimensi, seperti kecepatan, ketelitian dan penyediaan data dengan volume yang lebih besar yang memberikan bahan pertimbangan yang lebih banyak untuk mengambil keputusan.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri ; 2012 : 38)

II.3. Akutansi

Menurut Abubakar (2007:1) dalam bukunya Akutansi Untuk Bisnis Usaha Kecil dan Menengah, Akutansi adalah merupakan proses identifikasi, pencatatan dan komunitas terhadap transaksi ekonomi dari suatu entitas (perusahaan). Jadi secara umum terdapat tiga aktifitas dalam akutansi yaitu :

1. Aktifitas identifikasi (*Identifying*), dalam aktifitas ini akan dilakukan identifikasi terhadap transaksi yang terjadi dalam suatu *entitas* (perusahaan)
2. Aktifitas pencatatan (*recording*), dalam aktifitas ini semua transaksi ekonomi atau transaksi keuangan yang telah diidentifikasi pada tahap pertama akan di catat secara *kronolis* dan *sistematis* dengan ukuran nilai *nometer* tertentu, dan
3. Aktifitas komunikasi (*external user*), misalnya *kreditur*, *investor* dan *fiskus*.

II.4. Sistem Informasi Akutansi

Menurut Kusri (2007:10) dalam bukunya *Tuntunan Praktis Membangun Sistem Informasi Akutansi Dengan Visual Basic dan Sql Server*, Sistem informasi akutansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya.

Tujuan dari sistem informasi akutansi 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 akutansi adalah sebagai berikut:

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

Di dalam organisasi, sistem informasi akutansi berfungsi untuk :

- 1.Mengumpulkan dan menyimpan aktifitas yang dilakukan di suatu organisasi, sumber daya yang dipengaruhi oleh aktifitas-aktifitas tersebut dan para pelaku aktifitas tersebut.
- 2.Mengubah data menjadi informasi yang berguna bagi manajemen.

II.5. Pengertian Java

Java Language Specification adalah definisi teknis dari bahasa pemrograman *Java* yang di dalamnya terdapat aturan penulisan *sintaks* dan *semantik Java*. Referensi lengkap tentang *Java Language Specification* dapat anda temui pada website resmi *Java*, yaitu <http://java.sun.com/docs/books/jl> (Wahana Komputer ; 2010 : 3).

Java Memiliki banyak kelebihan, antara lain :

a. Sederhana

Untuk mempelajari *Java* tidaklah sulit. Anda bisa mempelajari dan membuat program secara cepat dengan menggunakan *Java*.

b. Berorientasi Objek

Meskipun banyak bahasa pemrograman yang menyebut dirinya berorientasi objek, *Java* bukan merupakan turunan dari bahasa pemrograman manapun dan sama sekali tidak kompatibel dengan *source-code* bahasa apapun.

c. Aman

Banyak bahasa yang pada awalnya dirancang dengan tingkat keamanan yang hampir tidak ada. Fasilitas-fasilitas yang diberikan seringkali dapat dimanfaatkan untuk disusupi oleh berbagai *virus*. Berbeda dengan *Java* yang disusun sejak awal dengan prinsip keamanan.

d. Interaktif

Java dirancang memenuhi kebutuhan dunia nyata untuk menciptakan program jaringan yang interaktif.

e. Kokoh

Java membatasi anda dari berbagai hal kunci supaya dapat menemukan kesalahan lebih cepat saat mengembangkan program.

f. Terinterpretasi dan Berkinerja Tinggi

Java dapat diterjemahkan oleh sistem apapun yang memiliki program Java di dalamnya. Java dirancang dengan hati-hati sehingga mudah diterjemahkan ke dalam bahasa asli suatu sistem untuk menghasilkan kinerja yang tinggi.

g. Terdistribusi

Java dapat dengan mudah bekerja dalam lingkungan terdistribusi seperti internet/intranet karena kemampuannya menangani *protokol TCP/IP*.

h. Dinamis

Karena berorientasi objek, Java mudah dalam hal pemeliharaan dan pengembangan, karena kita tidak harus membedah isi program untuk mengubah dan mengembangkan program dengan skala lebih besar (Ridwan Sanjaya ; 2005 : 2)

II.5.1. Pengertian NetBeans

NetBeans merupakan salah satu proyek *opensource* yang disponsori oleh *SunMicrosystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu *NetBeans IDE* dan *NetBeans Platform*. *NetBeans IDE* merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-*compile*, mencari kesalahan dan mendistribusikan program. Sedangkan *NetBeans Platform* adalah sebuah modul yang merupakan kerangka awal / pondasi dalam bangun aplikasi desktop yang besar.

NetBeans juga menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi desktop), pemrograman *enterprise*, dan pemrograman perangkat mobile. Saat ini *NetBeans* telah mencapai versi 6.8 (Wahana Komputer ; 2010 : 15)

II.6. Pengertian Database

Database adalah sekumpulan *file* data yang saling berhubungan dan diorganisasi sedemikian rupa sehingga memudahkan untuk mendapat dan memproses data. Lingkungan sistem *database* menekankan data yang tidak tergantung (*idenpendent data*) pada aplikasi yang akan menggunakan data. Data adalah kumpulan fakta dasar (mentah) yang terpisah.

Sebuah *database* harus dibuat dengan rapi agar data yang dimasukkan sesuai dengan tempatnya. Sebagai contoh, di sebuah perpustakaan, penyimpanan buku dikelompokkan berdasar jenis atau kategori-kategori tertentu, misalnya kategori buku komputer, buku pertanian, dan lain-lain. Kemudian dikelompokkan lagi berdasarkan abjad judul buku, ini dilakukan agar setiap pengunjung dapat dengan mudah mencari dan mendapatkan buku yang dimaksud (Wahana Komputer ; 2006 ; 1)

II.7. Pengertian MySQL

Mysql pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius, yang dapat anda hubungi di emailnyamonty@analytikernaMysql *database server* adalah RDBMS (*Relasional Database Management System*) yang dapat

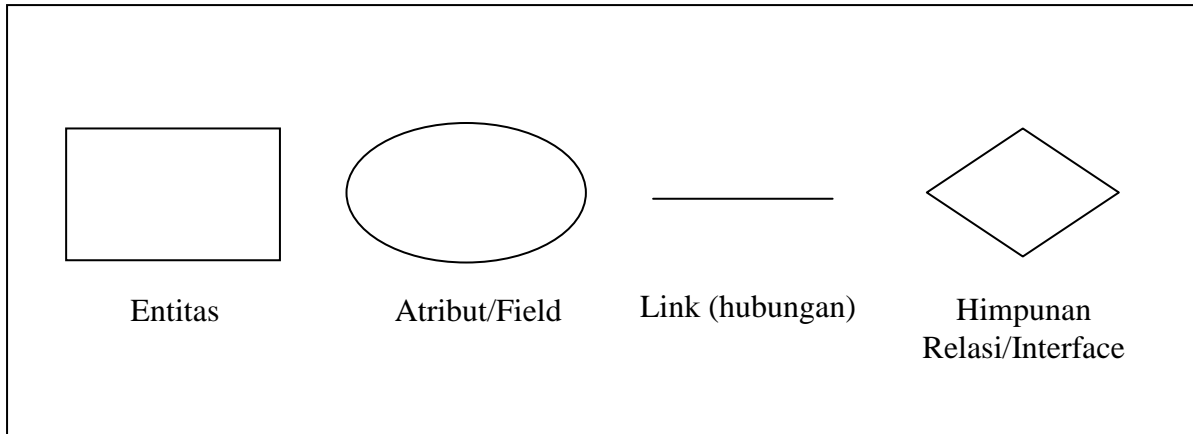
menangani data yang bervolume besar. Meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah *database* yang paling populer diantara *database* yang lain.

MySQL adalah program *database* yang mampumengirimdanmenerima data dengan sangat cepat dan *multiuser*. MySQL memiliki dua bentuk lisensi, yaitu *freeware* dan *shareware*. Penulis sendiri dalam menjelaskan buku ini menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*), yang dapat anda download pada alamat resminya <http://www.mysql.com>. MySQL sudah cukup lama dikembangkan, beberapa *fase* penting dalam pengembangan MySQL adalah sebagai berikut :

- a. MySQL dirilis pertama kali secara internal pada 23 Mei 1995
- b. Versi windows dirilis pada 8 Januari 1998 untuk windows 95 dan windows NT.
- c. Versi 3.23 : beta dari Juni 2000, dan dirilis pada January 2001.
- d. Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (unions) (Wahana Komputer ; 2010)

II.8. EntityRelationship Diagram (ERD)

EntityRelationship Diagram atau ERD merupakan salah satu alat (tool) berbentuk grafis yang populer untuk *desain database*. Tool ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analisis memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secara seksama, tool ini mencapai 2NF (Yuniar Supardi ; 2010 : 448)



Gambar. II.1 Bentuk Simbol ERD
(Sumber : Ir. Yuniar Supardi ; 2010 : 448)

II.9. Kamus Data

Karena DBMS menyimpan kumpulan beberapa *item* data yang terpisah yang dapat digunakan pemakai pada beberapa aplikasi secara bersama-sama, adalah penting bahwa beberapa mekanisme digunakan untuk menyediakan informasi mengenai beberapa *item* data bersangkutan. Itu adalah fungsi dari kamus data.

Kamus data adalah suatu *file* yang terpisah yang menyimpan informasi seperti :

- Nama setiap *item*/jenis/kolom data.
- Struktur data untuk tiap *item*.
- Program yang menggunakan tiap *item*
- Tingkat keamanan untuk setiap *item* (Zulkifli Amsyah ; 2005 : 382).

Pemakai yang perlu memperoleh informasi dari *database* dapat menuju ke kamus data untuk mendapatkan nama dari *item* data yang digunakan pada penelusuran (*search*). Dan yang mendesain aplikasi dapat menggunakan kamus untuk menentukan apakah *item* data sudah disimpan di komputer, dan kalau sudah dengan nama apa *item* data tersebut dapat dipanggil dan aplikasi apa yang digunakan.

Kamus data berguna khusus bagi perlindungan timbulnya kelebihan data. Tanpa kamus data, pemakai dari lain bagian mungkin menyimpan versi identik dari *item* data yang sama pada beberapa lokasi, dimana masing-masing *item* data mempunyai nama yang berbeda.

Operasional komputer dalam organisasi dewasa ini banyak yang sudah menggunakan model kerja jaringan (*network*). Dengan menggunakan data dasar yang sama untuk keperluan informasi masing-masing unit dan manajemen organisasi secara keseluruhan.

II.10. Teknik Normalisasi

Proses normalisasi menyediakan cara sistematis untuk meminimalkan terjadinya kerangkapan data diantara relasi dalam perancangan logikal basis data. Format normalisasi terdiri dari lima bentuk, yaitu:

II.10.1. Bentuk-bentuk Normalisasi

a. Bentuk normal tahap pertama (*1st Normal Form*)

Suatu tabel dikatakan sudah *INF* jika telah memenuhi ketentuan sebagai berikut:

- Tidak ada atribut mempunyai nilai berulang atau nilai array
- Tidak mempunyai baris yang rangkap

Bentuk unnormal mengijinkan nilai-nilai pada suatu atribut dapat berulang.

b. Bentuk normal tahap kedua (*2nd normal form*)

Relasi dapat dikatakan format normal kedua jika sudah dalam format normal pertama dan diikuti kondisi sebagai berikut:

- Key terdiri dari atribut tunggal
- Setiap atribut nonkey ketergantungan fungsional pada semua key atau tidak terjadinya ketergantungan pada *keycomposite*.

Misalnya tabel UNIV berada dalam normal kedua dengan mengasumsikan DNO sebagai key, kecuali CRSE. Jika ditentukan CNO dan SECNO sebagai *keycomposite*, atribut nonkey CNAME tergantung hanya pada CNO, bukan pada SECNO, sehingga CNAME tidak secara ketergantungan fungsional penuh terhadap key (CNO, SECNO).

c. Bentuk normal tahap ketiga (*3rd normal form*)

Relasi dikatakan format normal ketiga jika sudah dalam format normal kedua dan tidak ada ketergantungan transitif diantara atribut. Misalnya tabel STUDNT mempunyai atribut SSNO sebagai *key* (2NF). Ketergantungan transitif terjadi diantara DNO dan COLREG. Saat DNO determinan COLREG tanpa melibatkan *key* SSNO. Contohnya, DNO='CS' termasuk COLREG='Arts/Sc.' tidak tergantung oleh atribut SSNO, sehingga STUDNT belum termasuk 3NF. Yang menjadi catatan, ketergantungan transitif tidak akan terjadi jika ada ketergantungan fungsional diantara atribut-atribut *non-key* yang melibatkan *key*.

d. *BoyceCode Normal Form (BCNF)*

BCNF menentukan setiap determinan adalah kunci kandidat (*candidatekey*). Misalnya UNIV mempunyai dua determinan yaitu DNO dan DNAME yang merupakan kunci kandidat sehingga termasuk ke dalam BCNF. Di lain pihak

CRSLST dalam 3NF tetapi tidak dalam BCNF. Atribut komposisinya (CNO, SECNO, SID, OFRNG) sebagai kunci-kunci kandidat dan tidak ada ketergantungan transitif, sehingga CRSLST termasuk ke dalam 3NF. Namun atribut CNO adalah determinan saat SECNO tergantung penuh secara fungsional terhadap CNO, walaupun CNO bukan kunci kandidat, sehingga CRSLST belum termasuk BCNF.

e. Bentuk Normal Tahap Keempat dan Kelima

Bentuk ini adalah bentuk normal ketiga atau BCNF dengan nilai atribut tidak tergantung pada nilai banyak (*multivaluedependency*). Konsep pada bentuk ini adalah ketergantungan pada gabungan beberapa atribut (*joindependency*) (HaidarDzacko ; 2007 : 12).

II.11. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : k4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language*(UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

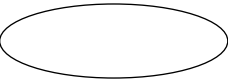
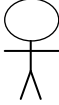


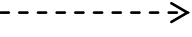
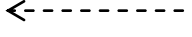
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. Usecase Diagram

Usecase diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *usecase* diagram, yaitu :

Tabel 1. Simbol Use Case




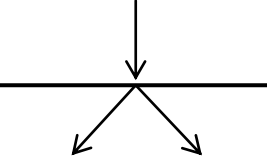
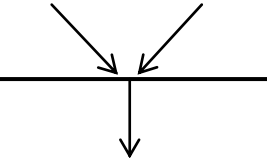
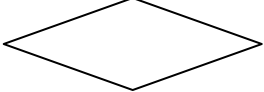

Gambar	Keterangan
	<i>Usecase</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>usecase</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>usecase</i> , tetapi tidak memiliki control terhadap <i>usecase</i> .
	Asosiasi antara aktor dan <i>usecase</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>usecase</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>usecase</i> lain (<i>required</i>) atau pemanggilan <i>usecase</i> oleh <i>usecase</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>usecase</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata : 2013;4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu

Tabel 2. Simbol *Activity Diagram*

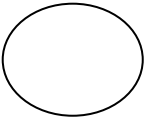
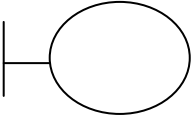
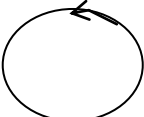

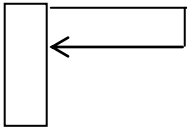


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>Endpoint</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>DecisionPoints</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activitydiagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata;2013;4)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel 3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>BoundaryClass</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata;2013;4)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel 4.1 *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata;2013;4)

II.12. **Pendapata Jasa**

Pendapatan adalah arus masuk bruto dari manfaat ekonomi yang timbul akibat aktivitas normal perusahaan selama satu periode, arus masuk itu mengakibatkan kenaikan modal (ekuitas) dan tidak berasal dari kontribusi penanaman modal. Arus masuk dimaksud adalah hasil dari penjualan produk perusahaan.

Jasa adalah hasil penggunaan produk atau fasilitas perusahaan perusahaan berupa produk tidak berwujud. Untuk memproduksi jasa, perusahaan juga menggunakan bahan baku berupa tenaga kerja dan masukan modal. Biaya untuk karyawan, asuransi, kegiatan administrasi adalah contoh jasa antara lain penyewaan properti, mobil atau aset lain perusahaan.

Arus masuk bruto atau pendapatan adalah hasil dari penjualan produk yang hanya diterima dan dapat diterima oleh perusahaan. Jumlah yang ditagih atas nama pihak ketiga, seperti pajak pertambahan nilai, bukan merupakan manfaat ekonomi yang mengalir ke perusahaan dan tidak mengakibatkan kenaikan modal, dan karena itu harus dikeluarkan dari pendapatan.

Pengertian penjualan jasa bisa menyangkut pelaksanaan tugas yang secara kontraktual telah disepakati untuk dilakukan selama suatu periode waktu. Jasa tersebut dapat diserahkan selama satu periode atau lebih. Penghasilan bersih dari penjualan jasa adalah penghasilan setelah dikurangi semua komisi penjualan, retur, rabat, diskon dan sebagainya (Kuswadi ; 2010 : 58). Contoh :

Harga penjualan kotor 100 unit @ Rp. 10.000 =		Rp. 1.000.000
Komisi penjualan 5 %	Rp. 50.000	
Diskon 10%	Rp. 100.000	
Retur 4 unit	Rp. 40.000	Rp. 190.000
Penjualan bersih dari 96 unit yang terjual		Rp. 810.000

**Gambar. II.2 Contoh Pendapatan Jasa
(Sumber : Kuswadi;2010;58)**

II.13. Biaya Operasional

Biaya operasional atau biaya operasi adalah biaya-biaya yang tidak berhubungan langsung dengan produk perusahaan tetapi berkaitan dengan aktivitas operasional perusahaan sehari-hari.

Secara umum, biaya operasional diartikan sebagai biaya yang terjadi dalam kaitannya dengan operasi yang dilakukan perusahaan dan diukur dalam satuan uang. Dimana biaya operasi sering disebut juga sebagai *operational cost* atau biaya usaha.

Adapun maksud dari semua biaya – biaya ini dijalankan oleh pihak perusahaan, karena biaya ini mempunyai hubungan langsung dari kegiatan utama perusahaan. Tujuan biaya operasi adalah:

1. Mengkoordinasikan dan mengendalikan arus masukan (*input*) dan keluaran (*output*), serta mengelola penggunaan sumber – sumber daya yang dimiliki agar kegiatan dan fungsi operasional dapat lebih efektif.
2. Untuk mengambil keputusan, akuntansi biaya menyediakan informasi biaya masa yang akan datang (*future cost*) karena pengambilan keputusan berhubungan dengan masa depan. Informasi biaya masa yang akan datang tersebut jelas tidak diperoleh dari catatan karena memang tidak dicatat, melainkan diperoleh dari hasil peramalan. Proses pengambilan keputusan khusus ini sebagian besar merupakan tugas manajemen perusahaan dengan memanfaatkan informasi biaya tersebut.
3. Digunakan sebagai pegangan atau pedoman bagi seorang manajer di dalam melakukan kegiatan – kegiatan perusahaan yang telah direncanakan perusahaan (Ira Kridsiyanti ; 2012 : 5).

II.14. Metode Multiple Step

Laporan laba rugi *multiple step*, adalah laporan laba rugi yang menghitung penghasilan dan beban berdasarkan kriterianya masing-masing, yaitu pendapatan dan beban operasional serta pendapatan dan beban lain-lain. (Ryan Ariefiansyah ;2013 : 14).

Penyusunan laporan laba-rugi dalam bentuk ini disusun secara bertahap mulai dari kelompok pendapat andan beban usaha, pendapatan luar usaha dan beban luar usaha. Sampai dengan kelompok pendapatan lain-lain dan beban lain-lain. Bentuk multi step ini banyak digunakan di perusahaan dagang atau perusahaan industri. Contoh bentuk laporan laba-rugi multi step sebagai berikut:

"Bengkel XYZ"
Laporan Laba-Rugi
Periode Berakhir 31 Desember

1.	Pendapatan Usaha		Rp. xxxxx
2.	Beban Usaha:		
-	Rp. xxxxx	
-	Rp. xxxxx	
-	Rp. xxxxx (+)	
	Total Beban Usaha		<u>Rp. xxxxx (-)</u>
	Laba Usaha		Rp. xxxxx
3.	Pendapatan Luar Usaha:		
-	Rp. xxxx	
-	Rp. xxxx (+)	
	Rp. xxxxx		
4.	Beban Luar Usaha:		
-	Rp. xxxx	
-	Rp. xxxx (+)	
	Rp. xxxxx (-)		
			Rp. xxxxx
5.	Pendapatan Lain-lain:		
-		<u>Rp. xxxxx (+)</u>
	Laba sebelum pajak		Rp. xxxxx
	Pajak Penghasilan		<u>Rp. xxxxx (-)</u>
	Laba Bersih setelah pajak		<u>Rp. xxxxx</u>

Gambar. II.3 Contoh Multiple Step
(Sumber : Ryan Ariefiansyah ;2013 : 14)