BAB II

LANDASAN TEORI

II.1. Teori Sistem

Sistem merupakan sekumpulan elemen-elemen yang salin terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. (Sulindawati dkk;2012;375)

Kata sistem mempunyai beberapa pengertian, tergantung dari sudut mana kata tersebut didefenisikan. Secara garis besar ada dua pendekatan yang dilakukan yaitu :

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

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan sistem sebagai sekumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan.

Dengan demikian didalam suatu sistem, komponen-komponen ini tidak dapat berdiri sendiri, tetapi sebaliknya, saling berhubungan hingga berbentuk suatu kesatuan hingga tujuan sistem dapat tercapai. (Kusrini;2010;5)

II.1.1.Karakteristik sistem

Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu yaitu:

1. Komponen sistem (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupaya suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

2. Batasan sistem (*Boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan luar sistem (*Environtment*)

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung sistem (*Interface*)

   Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

5. Masukan sistem (*Input*)

   Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*Output*)

   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem (*Process*)

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

8. Sasaran sistem (*Objective*)

   Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. (Kusrini;2010;6)

II.1.2.Klasifikasi sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya sebagai berikut:

1. Sistem abstrak dan sistem fisik.

   Sistem abstrak adalah sistem yang berisi gagasan atau konsep. Misalnya, sistem teologi yang berisi gagasan tentang hubungan manusia dan Tuhan. Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia.

   Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia). Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tentu.

   Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat. Misalnya, sistem komputer. Sistem tak tentu adalah sistem yang tak dapat diramal dengan pasti karena mengandung unsur probabilitas. Misalnya, sistem arisan dan sistem sediaan.

4. Sistem tertutup dan sistem terbuka

   Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan. Misalnya, reaksi kimia dalam tabung terisolasi. Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan. (Kusrini;2010;7)

II.2. Informasi

Informasi adalah data yang 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 bila dibandingkan biaya untuk mendapatkannya. (Kusrini;2010;7)

II.2.1.Kualitas Informasi

Informasi yang berkualitas memiliki 3 kriteria yaitu :

1. Akurat *(Accurate)*

   Informasi harus bebas dari kesalahan, tidak bias ataupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

2. Tepat pada waktunya (*timeliness)*

   Informasi yang datang pada penerima tidak boleh terlambat. Didalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai. Bila informasi datang terlambat sehingga pengambilan keputusan terlambat dilakukan, hal ini dapat berakibat fatal pada perusahaan.

3. Relevan (*Relevance)*

   Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi harus bermanfaat bagi pemakainya. Disamping karakteristik, nilai informasi juga ikut menentukan kualitasnya. Nilai informasi (*Value of Informatioan)* ditentukan oleh 2 hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaat lebih besar disamping biaya untuk mendapatkannya. (Kusrini ;2010;8)

II.3. Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi

bersifat menajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dan laporan-laporan yang diperlukan.

Defenisi umum sistem informasi adalah suatu sistem yang terdiri atas rangkaian sub sistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan. (Kusrini;2010;8-9)

II.3.1.Komponen Sistem Informasi

Dalam suatu sistem informasi terdapat komponen-komponen sebagai berikut:

1. Perangkat Keras (*hardware),* mencakup sebagai piranti fisik seperti computer dan printer
2. Perangkat lunak (*Sofrware)* atau program, sekumpulan intruksi yang memunggkinkan
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dalam pembangkitan keluaran yang dikehendaki
4. Orang atau semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
5. Basis data *(Database)* yaitu sekumpulan table, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*Resources)* dipakai secara bersama-sama atau diakses oleh sejumlah pemakai. (Kusrini;2010;9)

II.4 Sistem Informasi Akuntansi

Sistem informasi akuntansi merupakan sistem informasi yang mengubah data transaksi bisnis yang 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 pertanggungjawaban

Komponen-komponen yang terdapat dalam sistem informasi akuntansi adalah sebagai berikut :

a. Orang-orang yang mengoperasikan sistem tersebut
b. Prosedur-prosedur, baik manual maupun terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
c. Data tentang proses-proses bisnis
d. Software yang dipakai untuk memproses data organisasi
e. Infrastruktur teknologi informasi

Didalam organisasi sistem informasi akuntansi berfungsi untuk :

a. Mengumpulkan dan menyimpan aktivitas yang dilaksanakan disuatu organisasi, sumber daya yang dipengaruhi oleh aktivitas-aktivitas tersebut dan para pelaku aktivitas tersebut
b. Mengubah data dan informasi yang berguna bagi manajemen
c. Menyediakan pengendalian yang memadai.

Sistem informasi akuntansi merupakan pendukung aktivitas organisasi. Yang termasuk pendukung aktivitas organisasi adalah :

a. Infrastruktur perusahaan, akuntansi, hukum dan administrasi umum
b. Sumber daya manusia : perekrutan, pengontrolan, pelatihan dan kompensasi kepada pegawai.
c. Teknologi : Peningkatan produk dan jasa (Penelitian)
d. Pembelian.

Sementara itu aktivitas utamanya adalah :

a. *Inbount Logistics*, penerimaan, penyimpanan dan distribusi bahan-bahan masukan.
b. Operasi : aktivitas untuk mengubah masukan menjadi barang atau jasa.
c. *Outbount Logistics :* distribusi produk kepelanggan.
d. Pemasaran dan Penjualan.
e. Pelayanan : Dukungan purna jual maintenance.

II.4.1.Siklus 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 akuntasi.

a. Siklus pendapatan

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

b. Siklus pengeluaran kas

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

c. Sikluas konversi

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

d. Siklus Manajemen Sumber Daya Manusia (SDM)

Siklus Manajemen Sumber Daya Manusia (SDM) merupakan siklus yang melibatkan proses penggajian pada karyawan.

e. 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 besarnya.

Didalam sebuah sistem informasi akuntansi, tidak semua siklus harus diimplementasikan.Yang wajib ada dalam sistem tersebut adalah siklus buku besar dan laporan keuangan.Transaksi-transaksi yang termasuk dalam siklus tetapi tidak diimplementasikan, misalnya penggajian, dapat dimasukkan dalam siklus buku besar. (Kusrini;2010;9-11)

II.4.2.Tujuan Sistem Akuntansi

Adapun tujuan sistem akuntansi yaitu :

1. Untuk menyediakan informasi bagi pengelolaan kegiatan usaha baru
2. Untuk memperbaiki informasi yang dihasilkan oleh sistem yang sudah ada, baik mengenai mutu, ketetapan penyajian, maupu struktur informasinya

3. Untuk memperbaiki pengendalian akuntansi dan pengecekan intern, yaitu untuk memperbaiki tingkat keandalan (realibility) informasi akuntansi dan untuk menyediakan catatan lengkap mengenai pertanggungjawaban dan perlindungan kekayaan perusahaan.
4. Untuk mengurangi biaya klerikal dalam penyelenggaraan catatan akuntansi.(Novitasari Ayuningdkk ;2014;3)

II.4.3.Unsur-Unsur Sistem Akuntansi

Adapun unsur – unsur sistem akuntansi yaitu :

1. Klasifikasi rekening

   Klasifikasi rekening adalah penggolongan rekening-rekening yang digunakan dalam sistem akuntansi.Rekening-rekening ini terdiri dari rekening neraca (riel) dan laporan rugi laba (nominal).

2. Buku besar dan buku pembantu

   Buku besar berisi rekening-rekening neraca dan rugi laba yang digunakan dalam sistem akuntansi.

3. Jurnal

   Jurnal adalah catatan transaksi pertama kali (books of original entry). Catatan ini dibuat urut tanggal terjadinya transaksi.Biasanya dibuatkan jurnal-jurnal khusus untuk mencatat transaksi-transaksi yang frekuensinya tinggi.

4. Bukti transaksi

   Bukti transaksi adalah formulir yang digunakan untuk mencatat transaksi pada saat terjadinya (data recording) sehingga menjadi bukti tertulis dari

transaksi yang terjadi seperti faktur penjualan, bukti kas keluar, dan lain-lain.(Novitasari Ayuning dkk;2014;4)

II.4.4.Prinsip Penyusunan Sistem Akuntansi

Adapun prinsip penyusunan sistem akuntansi yaitu :

1. Analisa sistem yang ada. Langkah ini dimaksudkan untuk mengetahui kebaikan dan kelemahan sistem yang berlaku.
2. Merencanakan sistem akuntansi (*system design*). Langkah ini merupakan pekerjaan menyusun sistem baru, atau mengubah sistem lama agar kelemahan-kelemahan yang ada dapat dikurangi atau ditiadakan.
3. Penerapan sistem akuntansi. Langkah ini adalah menerapkan sistem akuntansi yang disusun untuk menggantikan sistem lama. Sebaiknya sistem baru dimulai penggunaanya pada awal periode akuntansi.
4. Pengawasan sistem baru (*follow up*). Langkah ini adalah untuk mengawasi penerapan sistem baru, yaitu mengecek apakah sistem baru ini dapat berfungsi. Apabila ada kesalahan-kesalahan, maka selama masa pengawasan ini perlu diakukan perbaikan-perbaikan. (Novitasari Ayunning dkk ;2014;3)

II.5. Metode Account Payable Procedure (Hutang Dagang)

Account Payable Procedure merupakan catatan hutang yang berupa kartu hutang yang diselenggarakan untuk setiap kreditur , yang memperlihatkan catatan mengenai nomor faktur dari jumlah pemasok, jumlah yang terutang, jumlah pembayaran dan saldo hutang.(Supryanto;2014;3)

Hutang Dagang *Account payable* adalah jumlah uang yang masih harus dibayarkan kepada pemasok, karena perusahaan melakukan pembelian barang atau jasa.

Adapun elemen pengendalian internal terhadap hutang jangka pendek adalah:

1. Diselenggarakannya catatan hutang (dengan kartu atau arsip *voucher* yang belum dibayar) yang secara periodik direkonsiliasi dengan rekening kontrol hutang yang bersangkutan di dalam buku besar.
2. Diadakan pengecekan informasi di dalam buku catatan hutang dagang dengan menggunakan pernyataan piutangditerima dari kreditur.
3. Adanya prosedur yang memberitahu bagian hutang mengenai adanya barang yang dikembalikan kepada penjual.
4. Faktur dari penjual harus dicek pertama kali oleh bagian pembelian sebelum dibayar.
5. *Voucher* dan dokumen pendukungnya dicap “lunas” atau diberi tanda setelah dilaksanakan pembayarannya, untukmengindari pembayaran kedua kalinya.
6. Adanya otorisasi dari pihak berwenang untuk setiap pembayaran hutang jangka pendek.
7. Semua penarikan hutang harus diotorisasi oleh yang berwenang. (Tabita Dian Permata Sari;2012;109)

II.6. Pengertian Basis Data (Database)

Istilah basis data/*database* tersusun atas dua suku kata, yaitu basis dan data (basis data = basis + data). Dalam sistem bilangan biner, kita dapat menuliskan beberapa contoh bilangan sebagai berikut (Edhy Sutanta:2011:25).

0 → sama dengan 0 dalam sistem bilangan desimal.

1 → sama dengan 1 dalam sistem bilangan desimal.

10 → sama dengan 2 dalam sistem bilangan desimal.

11 → sama dengan 3 dalam sistem bilangan desimal.

100→ sama dengan 4 dalam sistem bilangan desimal.

Istilah basis data dapat dipahami sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data (kalau ada kerangkapan data tersubut harus seminimal mungkin dan terkontrol. (Edhy Sutanta;2011;29)

II.6.1.Tujuan Basis Data

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan informasi. Untuk mencapai tujuan, ada lima aspek penting dalam basis data yaitu :

a. Kerangkapan data (*data redundancy*)

   Kerangkapan data (*data redundancy*) adalah munculnya data –data yang sama secara melimpah (berulang kali) pada file basis data yang semestinya tidak diperlukan. Misalnya ada data mahasiswa yang memuat nim, nama, alamat dan atribut lainya, sementara kita punya data lain tentang data kartu

hasil studi mahasiswa yang isinya terdapat nim, nama, mata kuliah dan nilai. Pada kedua data tersebut kita temukan atribut nama.

b. Inkonsisistensi data (*data inconsistency*)

Inkonsisistensi data (*data inconsistency*) atau data tidak konsisten adalah munculnya data yang tidak konsisten pada medan/kolom yang sama dalam satu atau beberapa file data yang dihubungkan/ direlasikan.

c. Data Terisolasi (*data isolation*)

Data terisolasi disebabkan oleh pemakaian beberapa file basis data dimana program aplikasi tidak dapat mengakses data-data dari file tertentu, kecuali bila program aplikasi diubah/ditambah sehingga seolah-olah ada file yang terpisah/terisolasi terhadap file lain dalam basis data.

d. Keamanan data (*data security*)

Keamanan data (*data security*) adalah merupakan aspek kritis dalam basis data. Prinsip dasar dari keamanan data dalam basis data adlah bahwa data-data dalam basis data merupakan sumber informasi yang bersifat sangat penting dan rahasia.

e. Integritas data (*data integrity*)

Integritas data (*data integrity*) berhubungan dengan kinerja sistem agar dapat melakukan kendali/kontrol pada semua bagian sistem. Integritas dimaksudkan sebagai suatu sarana untuk meyakinkan bahwa data-data yang tersimpan dalam basis data selau berada dalam kondisi yang benar.(Edhy Sutanta;2011;53)

II.6.2.Manfaat/Keuntungan Basis Data

Ada banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/keuntungan basis data diantaranya adalah :

1. Kerangkapan data dapat diminimalkan

   Jika file-file basis data dalam program aplikasi disiptakan oleh perancang yang berbeda pada waktu yang berselang cukup lama maka beberapa bagian data akan mengalami kerangkapan.

2. Inkonsistensi data dapat dihindari

   Basis data yang terbebas dari kerangkapan data akan terhindar dari munculnya data-data yang tidak konsisten.

3. Data dalam basis data dapat digunakan secara bersama (*multiuser*)

   Dalam rangka meningkatkan kinerja meningkatkan kinerja sistem dan untuk memperoleh respons waktu yang cepat, beberapa sistem mengizinkan banyak pemakai untuk dapat meng-*update* data secara bersamaan

4. Standarisasi data dapat dilakukan

   Definisi file basis data di dalam kamus data memungkinkan dilakukannya penerapan standarisasi data dalam basis data.

5. Pembatasan untuk keamanan untuk keamanan data dapat diterapkan

   Data-data dalam basis data dapat diatur sehingga hanya pemakai tertentu yang mempunyai wewenang saja yang dapat mengaksesnya.

6. Integritas data dapat dipelihara.

   Integritas berhubungan dengan kinerja sistem agar dapat melakukan kendali/kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian penuh.

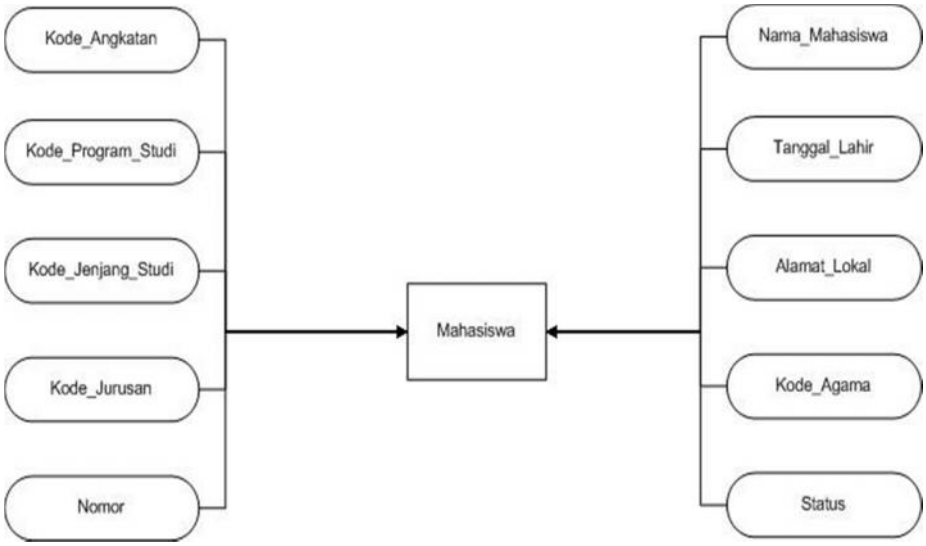
7. Perbedaan kebutuhan data dapat diseimbangkan

   Setiap pemakai dalam sistem memiliki kebutuhan yang berbeda-beda. Pengembangan basis data yang benar mampu menyeimbangkan perbedaan-perbedaan kebutuhan tersebut karena secara konseptual akan menggunkan basis data yang sama. (Edhy Sutanta;2011;49-50)

II.6.3.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 dalam sebuah *database* yang baik.

Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen – komponen data (atribut atau *field*) yang membuatnya dapat dibedakan dari obyek yang lain. Hal ini berarti sebuah *entity* memiliki himpunan yang diperlukan sebuah *primary key* untuk membedakan anggota–anggota dalam himpunan tersebut. Berikut ini contoh penggambaran entitas Mahasiswa yang ditunjukan pada Gambar II.1.



Gambar : II.1 Gambar ERD

Sumber : (Edhy Sutanta;2011;100)

Ada beberapa jenis kerelasian antar entitas (*Relationship*) yaitu:

a. Kerelasian jenis1-ke-1/satu ke satu (*one to one*)

Kerelasian jenis ini terjadi jika kejadian atau transaksi diantara dua entitas yang berhubungan hanya memungkinkan terjadi sebuah kejadian atau transaksi pada kedua entitas. Secara lebih teknis, jika nilai yang digunakan sebagai penghubung pada entitas pertama hanya dimungkinkan muncul satu kali saja pada entitas kedua yang saling berhubungan.

b. Kerelasian jenis n-ke-1/banyak ke satu (*many to one*) atau 1-ke-n/satu ke banyak (*one to many*)

Kerelasian jenis ini terjadi jika kejadian atau transaksi diantara dua entitas yang berhubungan hanya memungkinkan terjadi satu kali dalam entitas pertama dan dapat terjadi lebih dari satu kali kejadian atau transaksi pada entitas kedua. Secara lebih teknis, jika nilai yang digunakan sebagai

penghubung pada entitas pertama hanya dimungkinkan muncul lebih dari satu kali pada entitas kedua yang saling berhubungan.

c. Kerelasian jenis n-ke-n/banyak ke banyak (*many to many*)

Kerelasian jenis ini terjadi jika kejadian atau transaksi diantara dua entitas yang berhubungan hanya memungkinkan terjadi lebih dari satu kali dalam entitas pertama dan entitas kedua. Secara lebih teknis, jika nilai yang digunakan sebagai penghubung pada entitas pertama hanya dimungkinkan muncul lebih dari satu kali, baik pada entitas pertama maupun pada entitas kedua yang saling berhubungan, dan sebaliknya. (Edhy Sutanta ;2011;102-103)

II.7. Normalisasi

Normalisasi diartikan suatu teknik yang menstrukturkan/mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data.Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomallies*) yang terjadi akibat kerangkapan data dalam relasi dan in-efisiensi pengolahan. Suatu basis data dikatakan tidak normal jika terjadi 2 (dua) anomali berikut :

1. *Insertion Anomaly*

   Anomali yang terjadi jika ada data yang tidak bisa disisipkan kedalam tabel.

2. *Update/Modification anomaly*

   Anomali yang terjadi jika ada perubahan pada suatu item data maka harus mengubah lebih dari satu baris data.

   Langkah-langkah normalisasi sampai pada bentuk 3NF yaitu :

1. Relasi bentuk tidak normal (*un normalizedform*/UNF)

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

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

Tabel II.1 Relasi supplier dalam bentuk UNF

Kode_Supplier	Status	Kota	Kode_Barang	Jumlah_Barang
S01	10	Jakarta	B01	100
			B02	150
			B03	200
S02	20	Surabaya	B02	250
			B04	200
S03	30	Yogyakarta	B05	150
			B06	100

Sumber : Edhy Sutanta:2011:179

2. Relasi bentuk normal pertama (*first nornform*/1NF)

   Relasi disebut sebagai 1NF jika memenuhi kriteria sebagai berikut:

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

Permasalahan dalam INF adalah sebagai berikut:

a. Tidak dapat menyisipkan informasi parsial.

b. Terhapusnya informasi ketika menghapus sebuah record.

c. Pembaharuan atribut nonkunci mengakibatkan sejumlah record harus diperbaharui.

Mengubah relasi UNF menjadi 1NF dapat dilakukan dengan cara yaitu :

a. Melengkapi nilai-nilai dalam atribut.

b. Mengubah struktur relasi.

Tabel II.2 Relasi supplier_1 dalam bentuk 1NF

Kode_Supplier	Status	Kota	Kode_Barang	Jumlah_Barang
S01	10	Jakarta	B01	100
S01	10	Jakarta	B02	150
S01	10	Jakarta	B03	200
S02	20	Surabaya	B02	250
S02	20	Surabaya	B04	200
S03	30	Yogyakarta	B05	150
S03	30	Yogyakarta	B06	100

Sumber: Edhy Sutanta;2011;180

3. Relasi bentuk normal kedua (*second nornform*/2NF)

Relasi disebut sebagai 2NF jika memenuhi kriteria sebagai berikut :

a. Jika memenuhi kriteria INF.

b. Jika semua atribut nonkunci FD pada PK.

Permasalahan dalam 2NF adalah sebagai berikut :

a. Kerangkapan data (*data redundancy*).
b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data (*data inconsistency*).
c. Proses pembaharuan data tidak efisien.
d. Penyimpangan pada saat penyisipan, penghapusan, dan pembahruan.

Kriteria tersebut mengindikasikan bahwa antara atribut dalam 2NF masih mungkin TDF.Selain itu, relasi 2NF menuntut telah didefinisikan atribut PK dalam relasi. Mengubah relasi INF menjadi bentuk 2NF dapat dilakukan dengan cara :

a. Identifikasikan FD relasi 1NF (jika perlu gambarkan diagram ketergantungan datanya).
b. Berdasarkan informasi tersebut, dekomposisi relasi 1NF menjadi relasi-relasi baru sesuai FD-nya. Jika menggunakan diagram maka simpul-simpul yang berada pada puncak diagram ketergantungan data bertindak sebagai PK pada relasi baru.

Tabel II.3 Relasi supplier_2 dalam bentuk 2NF

Kode_Supplier	Status	Kota
S01	10	Jakarta
S02	20	Surabaya
S03	30	Yogyakarta

Tabel II.4 Relasi Barang

Kode_Supplier	Kode_Barang	Jumlah_Barang
S1	P1	100
S1	P1	95
S2	P2	75
S2	P2	70
S3	P3	60
S3	P3	50

Sumber : Edhy Sutanta;2011;181

4. Relasi bentuk normal ketiga (*thrid nornform*/3NF)

   Suatu relasi disebut 3NF jika memenuhi kriteria sebagai berikut :

   a. Jika memenuhi kriteria 2NF
   b. Jika setiap atribut nonkunci tidak TDF (*non transitive depency*) terhadap PK.

   Permasalahan dalam 3NF adalah keberadaan penentu yang tidak merupakan bagian dari PK menghasilkan duplikasi rinci data pada atribut yang berfungsi sebagai FK (duplikasi berbeda dengan kerangkapan data).

   Mengubah relasi 2NF menjadi bentuk 3NF dapat dilakukan dengan mengubah struktur relasi dengan cara :

   a. Identifikasikan TDF relasi 2NF (jika perlu gambarkan diagram ketergantungan datanya).
   b. Berdasarkan informasi tersebut, dekomposisi relasi 2NF menjadi relasi-relasi baru sesuai TDF-nya. Jika menggunkan diagram maka simpul-

simpul yang berada pada puncak diagram ketergantungan data berrtindak sebagai PK pada relasi baru. (Edhy Sutanta;2011;174-178)

Tabel II.5 Relasi supplier_3 dalam bentuk 3NF

Kode_Supplier	Status
S01	10
S02	20
S03	30

Tabel II.6 Relasi kota

Status	Kota
10	Jakarta
20	Surabaya
30	Yogyakarta

Sumber : Edhy Sutanta;2011;182

II.7.1.Kamus Data (*data dictionary*)

Dalam suatu rancangan *database*, *data dictionary* digunakan untuk menjelaskan atau mendeskripsikan kolom-kolom pada masing-masing tabel yang akan dibuat kedalam *database.* Deskripsi kolom yang dimaksud disini meliputi tipe data, lebar karakter atau digit serta keterangan tentang kunci relasi. (Budi Raharjo;2011;50)

Tabel II.7 Kategori

Nama Kolom	Tipe Data	Lebar	Null ?	Kunci
Kategori_id	INT	11	Not Null	Primary Key
Kategori_nama	VARCHAR	25		

Tabel II.8Pengarang

Nama Kolom	Tipe Data	Lebar	Null ?	Kunci
Pengarang_id	CHAR	3	Not Null	Primary Key
Pengarang_nama	VARCHAR	30		

Tabel II.9Penerbit

Nama Kolom	Tipe Data	Lebar	Null ?	Kunci
Pengarang_id	CHAR	3	Not Null	Primary Key
Pengarang_nama	VARCHAR	30		

Sumber : Budi Raharjo;2011;50

II.7.2 SQL Server 2008

Bahasa *query* merupakan bahasa khusus yang digunakan untuk melakukan manipulasi dan menanyakan pertanyaan (*query*) yang berhubungan dengan data dalam basis data. Bahasa query tidak sama dengan bahasa pemrograman, dimana bahasa query tidak memiliki kemampuan untuk menyelesaikan banyak masalah seperti bahasa pemrograman pada umumnya.

Dalam pemrograman basis data, salah satu bahasa yang harus kita kuasai adalah SQL. SQL merupakan bahasa komputer standar yang digunakan untuk

berkomunikasi dengan sistem manajemen basis data relasional (RDBMS).SQL sering disebutkan sebagai singkatan dari *Stuctrured Query Language*. (Ema dan Anggit Dwi Hartanto;2012:62)

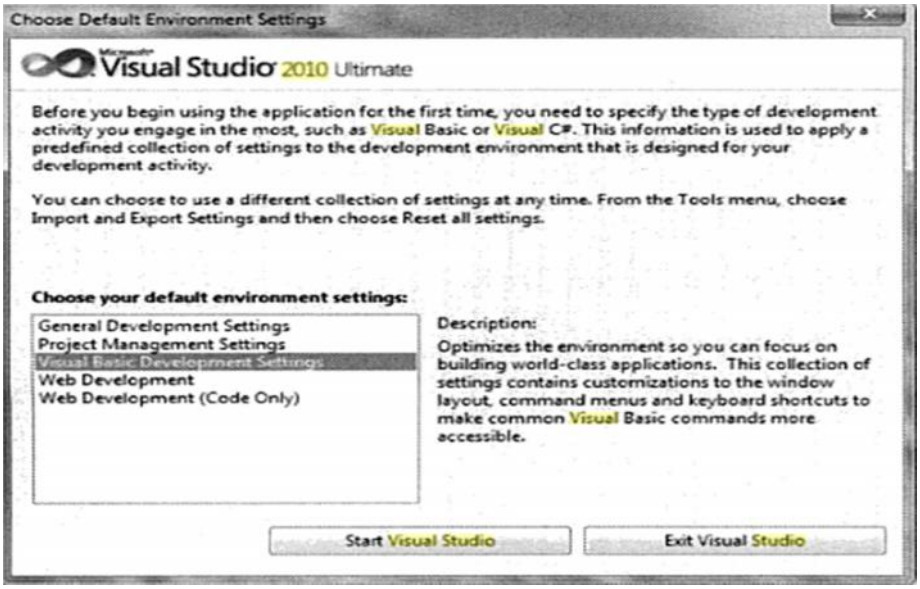
II.8. Visual Basic 2010

Visual Basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio* 2010. Sebagai produk lingkungan penembangan terintegrasi atau IDE andalan yang dikeluarkan oleh *Microsoft*.

*Visual Studio*merupakan produk pemrograman andalan *Microsoft Corporation*, yang di dalamnya berisi beberapa jenis IDE pemrograman seperti *Visual Basic, Visual C++, Visual Web Developer. Visual C#, dan Visual F#.*Semua IDE pemrograman tersebut sudah mendukung penuh implementasi *.Net Framework* terbaru, yaitu *.Net Framework* 4.0 yang merupakan pengembangan dari *.Net Freamework* 3.5. Adapun *database* standar yang disertakan adalah *Microsoft SQL Server 2008 Expres.* (Wahana Komputer ;2010:2)

II.8.1.Antar Muka Visual Basic 2010

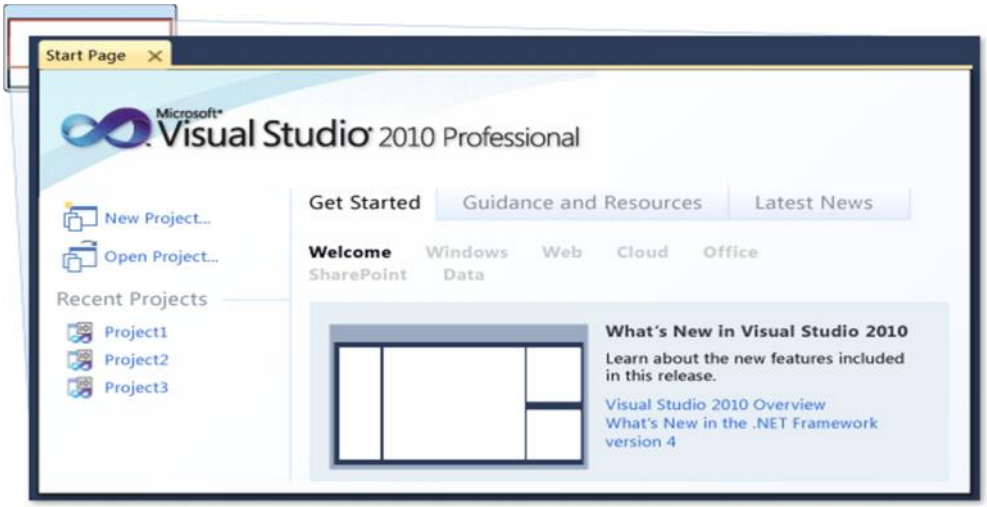
Saat menjalankan *Visual Basic* 2010 pertama kali muncul jendela *choose default environtment settings.* Disini bisa memilih apakah ingin memilih antar muka di *Visual Studio*. Untuk *programmer Visual Basic* lebih baik memilih *Visual Basic Development Centre.*



Gambar II.2 Form Chose Default Environtment Settings

Sumber : Wahana Komputer;2010;11

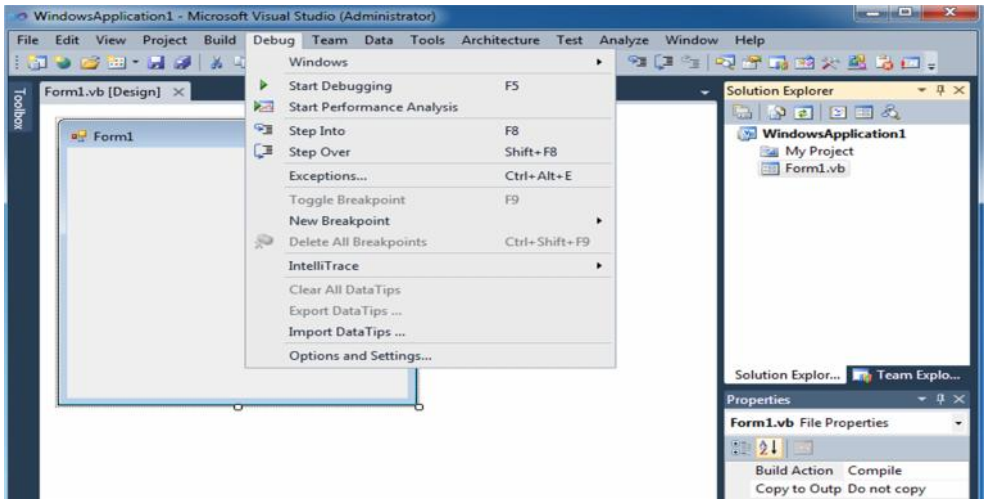
Dibagian awal *visual basic*, bisa memilih *Start Page*. *Start Page* adalah halaman yang mencantumkan informasi-informasi seputar program dan juga informasi RSS dari sumber tertentu. Jika tidak ingin menampilkan hal ini hilangkan tanda centang pada *Show Page On Startup.*



Gambar II.3 Start Page Visual Basic 2010

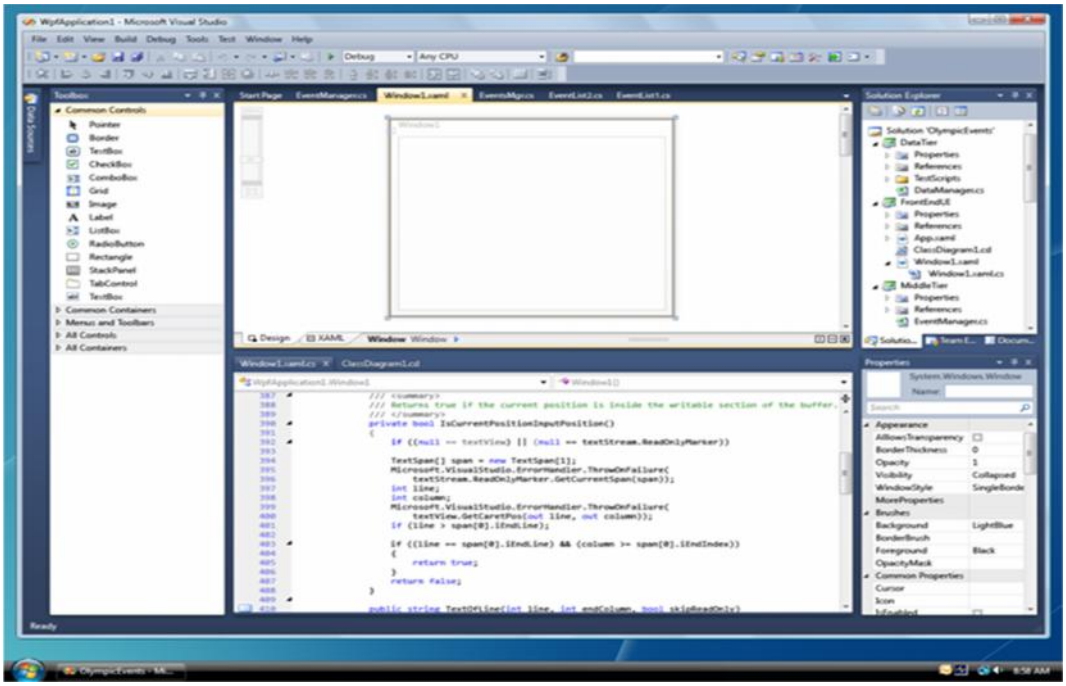
Sumber : Wahana Komputer;2010;12

Jika *start page* ditutup terlihat tampilan sebagai berikut :



Gambar II.4 Tampilan IDE (*Integrated Development Environtment)* setelah *Start Page* ditutup
Sumber : Wahana Komputer;2010;13

Jika ada sebuah form yang terlihat, tampilan lengkap IDE seperti gambar berikut :



Gambar II.5 Tampilan lengkap IDE

Sumber : Wahana Komputer;2010;14

Komponen-komponen dari IDE adalah :

1. Dibagian kiri terdapat toolbox yang menampilkan semua objek tool yang bisa dimasukkan kedalam form untuk membuat program.
2. Dibagian tengah terdapat tempat meletakkan form dan kode, baik disaat desain ataupun pada saat program dijalankan.
3. Dibagian kanan terdapat solution explorer yang merupakan explorer untuk melihat file-file disebuah objek.
4. Dikanan bawah terdapat propertis untuk melihat properti dari nilai-nilai pada objek yang dipilih dibagian tengah.

II.9. *Unified Modeling Language* (*UML)*

*Unified Modeling Language (*UML) adalah sebuah "bahasa" yang telah menjadi standar industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi procedural dalam VB atau C. (Yuni SugiartiS.T,M.Kom;2013:34)

II.9.1.Pengenalan UML

UML itu adalah salah bentuk language atau bahasa.UML didefinisikan sebagai bahsa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem.Karena tergolong bahasa visual, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem yang sedang dimodelkan.

UML adalah salah satu bentuk notasi atau bahsa yang sama digunakan oleh professional dibidang *software* untuk mengambarkan atau memodelkan sebuah sistem. Sebelumnya ada banyak notasi atau bahasa lain untuk mecapai keperluan misalnya DFD (*Data Flow Diagram*) dan Booch Diagram. UML telah menjadi *de facto standard language.*

UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melaui sejumlah elemen grafis yang bias dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasikan berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. (Yuni Sugiarti S.T, M.Kom;2013:36)

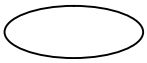
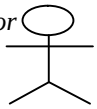
Adapun jenis-jenis dari tipe diagram *UML* adalah sebagai berikut :

1. *Use Case Diagram*

   Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Dengan kata lainnya, use case digunakan untuk mengetahui fungsi apa saja yang berhak menggunkan fungsi-fungsi itu.

Use Case Diagram 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*. (Rossa A.S, M.Shalaluddin;2013;155)

Tabel II.10 Simbol-simbol *Use Case Diagram*

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan menggunakan kata kerja diawal frase nama *use case* .
Actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Association	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor .
Extend <<extends>>	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu mirip dengan prinsip inheritance pada pemrograman berorientasi objek.
Include <<include>>	Menjelaskan bahwa *use case* termasuk dalam *use case* lain.

Sumber : Rosa.A.S, M,Shalaluddin;2013;156-157

2. *ClassDiagram*

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.

a. Atribut merupakan variable yang dimiliki oleh suatu kelas.
b. Atribut mendeskripsikan property dengan sebaris teks didalam kotak kelas tersebut.
c. Operasi atau metode adalalh fungsi-fungsi yang dimiliki oleh suatu kelas tersebut.

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara mereka.Diagram kelas juga menunjukan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

Diagram kelas menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

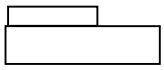
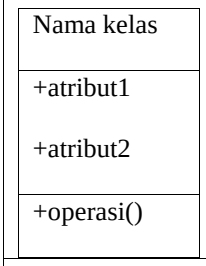
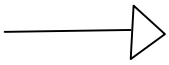
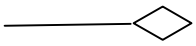
Kelas memiliki tiga area pokok :

a. Nama (dan *stereotype*)
b. Atribut
c. Metoda

Contoh kelas Manusia :

- Atribut : nama ,usia, tanggal lahir.
- Method/operasi :berjalan, makan, minum. (Yuni Sugiarti S.T,M.Kom;2013;58)

Tabel II.11 Simbol-simbol *Class Diagram*

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem.
Interface/Antarmuka 	Sama dengan konsepinterface dalam pemrograman berorientasi objek.
Asosiasi	Relasi antar kelas dengan makna umum asosiasi biasanya juga disertai dengan *multiplicity*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi(umum khusus)
Agregasi 	Relasi antar kelas dengan makna semua-bagian (*whole-part*)

Sumber : Yuni Sugiarti S.T,M.Kom;2013;59

3. *Statechart Diagram*

Statechart diagram atau State machine diagram menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. *State machine diagram* merupakan pengembangan dari diagram *Finite State Automata* dengan penambahan beberapa fitur dan konsep baru. *Statechart diagram* cocok digunakan untuk menggambarkan alur interaksi pengguna dengan sistem. (Rosa A.S, M.Shalaluddin;2013;163)

4. *ActivityDiagram*

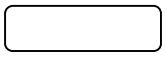
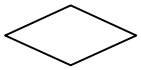
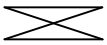
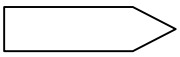
Activity Diagram atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa *Activity Diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokkan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujiannya.

Activity Diagram menggambarkan berbagai aliraktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing berawal, decision yang mungkin terjadi, dan bagaimana proses parallel yang mungkin terjada pada beberapa eksekusi. (Yuni Sugiarti S.T,M.Kom;2013;80)

Tabel II.12. Simbol Yang Ada Pada *Activity Diagram*

Simbol	Deskripsi
	Status awal aktivitas sistem,sebuah diagram aktivitas memiliki sebuah status awal.
	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah satus akhir.
	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan (*decision*) dimana jika ada pilihan aktivitas lebih dari satu.
	*Join* : kegiatan penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	*Rake* ; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman

Sumber : Rosa A.S, M.Shalaluddin;2013;162

5. *SequenceDiagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*.

6. *CollaborationDiagram*

   Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram* , tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message.*

7. *DeploymentDiagram*

   Deployment Diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware.*Sistem terdiri dari node-node dimana setiap node diwakili untuk sebuah kubus.Garis yang menghubungkan antara 2 kubus menunjukkan hubungan di antara kedua node tersebut.Tipe node bisa berupa *device* yang berwujud *hardware* dan bisa juga *processor*. (Rosa A.S, M.Shalaluddin;2013;165-169)