

BAB II

TINJAUAN PUSTAKA

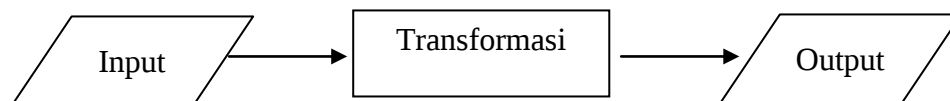
II.1. Pengertian Sistem

Sistem adalah suatu kesatuan usaha yang terdiri dari bagian – bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antar bagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interdependen satu sama lain. Selain itu, dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan yang terus menerus perlu dikembangkan dan dikendalikan. Definisi tersebut menunjukkan bahwa sistem sebagai gugus dari elemen – elemen yang saling berinteraksi secara teratur dalam rangka mencapai tujuan atau subtujuan. Sifat dasar suatu sistem, antara lain :

1. Pencapaian tujuan, orientasi pencapaian tujuan akan memberikan sifat dinamis kepada sistem, memberi ciri perubahan yang terus – menerus dalam usaha mencapai tujuan.
2. Kesatuan usaha, mencerminkan suatu sifat dasar dari sistem, dimana hasil keseluruhan melebihi dari jumlah bagian – bagiannya atau sering disebut konsep sinergi.
3. Keterbukaan terhadap lingkungan, lingkungan merupakan sumber kesempatan maupun hambatan pengembangan. Keterbukaan terhadap lingkungan membuat penilaian terhadap suatu sistem menjadi relatif atau yang dinamakan *equifinality* atau pencapaian tujuan suatu sistem tidak

mutlak harus dilakukan melalui berbagai cara sesuai dengan tantangan lingkungan yang dihadapi.

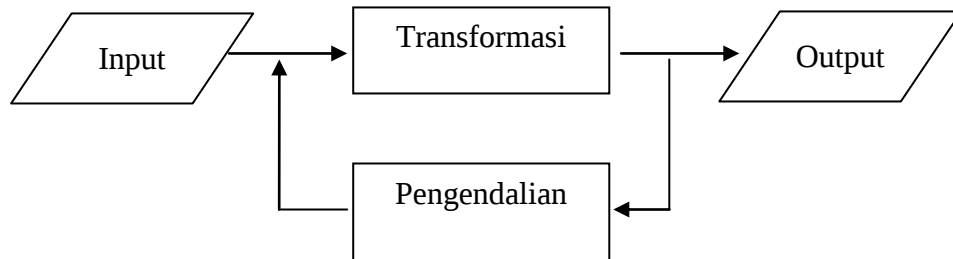
4. Transformasi, merupakan proses perubahan *input* menjadi *output* yang dilakukan oleh sistem, proses transformasi diilustrasikan pada gambar II.1. :



Gambar II.1. Proses Transformasi Input Menjadi Output.

(Sumber : Sistem Informasi Manajemen Sumber Daya Manusia ; 2011 : 2)

5. Hubungan antara bagian, kaitan antara subsistem inilah yang akan memberikan analisis sistem, suatu dasar pemahaman yang lebih luas.
6. Sistem ada berbagai macam, antara lain sistem terbuka, sistem tertutup, dan sistem dengan umpan balik.
7. Mekanisme pengendalian, mekanisme ini menyangkut sistem umpan balik yang merupakan suatu bagian yang memberi informasi kepada sistem mengenai efek dari perilaku sistem terhadap pencapaian tujuan atau pemecahan persoalan yang dihadapi. Skema proses transformasi sistem dengan mekanisme pengendalian dapat dilihat pada gambar II.2.



Gambar II.2. Skema Proses Transformasi Sistem Dengan Mekanisme Pengendalian.

(Sumber : Sistem Informasi Manajemen Sumber Daya Manusia ; 2011 : 3)

II.2. Pengertian 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.

II.2.1. Kualitas Informasi

Informasi yang berkualitas memiliki 3 kriteria, yaitu (Kusrini M.Kom ; 2010 : 7-8). :

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan, tidak bisa 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. Di dalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai. Bila informasi datang terlambat dilakukan., hal itu dapat berakibat fatal bagi perusahaan.

3. Relevan

Informasi yang disampaikan harus mempunyai keterikatan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi bermanfaat bagi pemakaiannya. Disamping karakteristik, nilai informasi juga ikut menentukan kualitasnya. Nilai informasi (*value of information*) ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatkannya.

II.3. 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 didalam 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).

III.4. Pengertian Penjualan

Penjualan merupakan tujuan utama dilakukannya kegiatan perusahaan. Perusahaan dalam menghasilkan barang/jasa mempunyai tujuan akhir, yaitu untuk menjual barang/jasa tersebut kepada masyarakat. Oleh karena itu, penjualan memegang peranan penting bagi perusahaan agar produk yang dihasilkan oleh perusahaan dapat terjual dan memberikan penghasilan bagi perusahaan. Penjualan yang dilakukan perusahaan bertujuan untuk menjual barang/jasa yang diperlukan sebagai sumber pendapatan untuk menutup semua ongkos untuk memperoleh laba.

Penjualan adalah pemindahan hak milik atas barang atau pemberian jasa yang dilakukan penjualan kepada pembeli dengan harga yang disepakati bersama dengan jumlah yang dibebankan kepada pelanggan dalam penjualan barang/jasa dalam suatu periode akuntansi. Penjualan merupakan pengalihan hak milik atas barang dengan imbalan uang sebagai gantinya dengan persetujuan untuk menyerahkan barang kepada pihak lain dengan menerima pembayaran. Keberhasilan usaha penjualan dapat dilihat dari *volume* penjualan yang didapat. Dengan kata lain, apakah usaha itu dapat laba atau tidak, sangat tergantung kepada keberhasilan penjualan itu.

II.5. Entity Relationship Diagram (ERD)

Pada dasarnya ERD (*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD di atas. ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan 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. 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 tiap 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 (Wahana Komputer ; 2010 : 30).

Berikut notasi ERD (*Entity Relationship Diagram*) dapat dilihat pada tabel

II.1. :

Tabel II.1. Notasi ERD (*Entity Relationship Diagram*)

Notasi	Keterangan
	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai
	Relasi selalu menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda
	Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah)
	Garis sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut

Sumber : Adi Nugroho(2005:29)

II.6. Bentuk-bentuk Normalisasi

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

b. Bentuk normal tahap pertama (1st Normal Form)

Sebuah table disebut 1NF jika :

1. Tidak ada baris yang duplikat dalam tabel tersebut.
2. Masing-masing cell bernilai tunggal.

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

Berikut ini akan dicontohkan normalisasi dari tabel kuliah yang memiliki atribut : kode_kul, nama_kul, sks, semester, waktu, tempat, dan nama_dos.

Tabel kuliah tersebut tidak memenuhi normalisasi pertama, karena terdapat atribut waktu yang tergolong ke dalam atribut bernilai banyak. Agar tabel tersebut dapat memenuhi 1NF, maka solusinya adalah dengan mendekomposisi tabel kuliah menjadi :

1. Tabel kuliah (kode_kul, nama_kul, sks, semester, nama_dos).
2. Tabel jadwal (kode_kul, waktu, ruang).

c. Bentuk normal tahap kedua (2nd Normal Form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

Sebuah tabel dikatakan tidak memenuhi 2NF, jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari primary key).

Bentuk normal kedua akan dicontohkan berikut.

Misal tabel nilai terdiri dari atribut kode_kul, nim dan nilai. Jika pada tabel nilai. Misalnya kita tambahkan sebuah atribut yang bersifat

redundansi, yaitu nama_mhs, maka tabel nilai ini dianggap melanggar 2NF. Primary key pada tabel nilai adalah (kode_kul, nim).

Penambahan atribut baru (nama_mhs) akan menyebabkan adanya ketergantungan fungsional yang baru yaitu $\text{nim} > \text{nama_mhs}$. Karena atribut nama_mhs ini hanya memiliki ketergantungan parsial pada primary key secara utuh (hanya tergantung pada nim, padahal nim hanya bagian dari primary key). Bentuk normal kedua ini dianggap belum memadai karena meninjau sifat ketergantungan atribut terhadap atribut terhadap primary key saja.

d. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

1. X haruslah superkey pada tabel tersebut.
2. Atau A merupakan bagian dari primary key pada tabel tersebut.

Misalkan pada tabel mahasiswa, atribut alamat_mhs dipecah kedalam alamat_jalan, alamat_kota dan kode_pos. Bentuk ini tidak memenuhi 3NF, karena terdapat ketergantungan fungsional baru yang muncul pada tabel tersebut, yaitu :

1. alamat_jalan, nama_kota – kode_pos.

Dalam hal ini (alamat_jalan, nama_kota) bukan superkey sementara kode_pos juga bukan bagian dari primary key pada tabel mahasiswa. Jika

tabel mahasiswa didekomposisi menjadi tabel mahasiswa dan tabel alamat, maka telah memenuhi 3NF. Hal itu dapat dibuktikan dengan memeriksa dua ketergantungan fungsional pada tabel alamat tersebut, yaitu :

1. alamat_jalan, nama_kota – kode_pos
2. kode_pos – nama_kota

Ketergantungan fungsional yang pertama tidak melanggar 3NF, karena (alamat_jalan, nama_kota) merupakan superkey (sekaligus sebagai primary key) dari tabel alamat tersebut. Demikian juga dengan ketergantungan fungsional yang kedua meskipun (kode_pos) bukan merupakan superkey, tetapi nama_kota merupakan bagian dari primary key dari tabel alamat. Karena telah memenuhi 3NF, maka tabel tersebut tidak perlu di-dekomposisi lagi.

e. Boyce Code Normal Form (BCNF)

1. Memenuhi 3rd NF.
2. Relasi harus bergantung fungsi pada atribut superkey.

f. Bentuk Normal Tahap Keempat (4th Normal Form)

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat (4NF) berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional.

g. Bentuk Normal Tahap Kelima (5th Normal Form)

Adapun bentuk normal tahap kelima (5NF) merupakan nama lain dari *Project Join Normal Form* (PJNF) (Kusrini, M.Kom ; 2010 : 41-43).

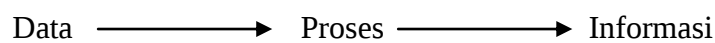
II.7. Pengertian Database

Banyak sekali definisi tentang database yang diberikan oleh para pakar dibidang ini. Database terdiri dari dua penggalan kata yaitu data dan base, yang artinya berbasiskan pada data. Tetapi secara konseptual, database diartikan sebuah koleksi atau kumpulan data yang saling berhubungan (*relation*), disusun menurut aturan tertentu secara logis, sehingga menghasilkan informasi. Sebuah informasi yang berdiri sendiri tidaklah dikatakan database. Contoh : Nomor telepon seorang pelanggan, disimpan dalam banyak tempat apakah itu di file pelanggan, di file alamat dan di lokasi yang lain. Antara file yang satu dengan file yang lainnya tidak saling berhubungan, sehingga apabila salah seorang pelanggan berganti nomor telepon dan anda hanya mengganti di file pelanggan saja, akibatnya akan terjadi ketidakcocokan data, karena di lokasi yang lain masih tersimpan data telepon yang lama.

Dalam sistem database hal ini tidak boleh dan tidak bisa terjadi, karena antara file yang satu dengan file yang lain saling berhubungan. Jika suatu data yang sama anda ubah, data tersebut di file yang lain akan otomatis berubah juga. Sehingga mampu menjadi informasi yang diinginkan dan dapat dilakukan proses pengambilan, penghapusan, pengeditan, terhadap data secara mudah dan cepat (Efektif, Efisien dan Akurat).

Data adalah fakta, baik berupa sebuah objek, orang dan lain – lain yang dapat dinyatakan dengan suatu nilai tertentu (angka, simbol, karakter tertentu, dan lain – lain). Sedangkan informasi adalah data yang telah diolah sehingga bernilai guna dan dapat dijadikan bahan dalam pengambilan keputusan (Yuhefizard ; 2010 : 2).

Hubungan data dan informasi dapat dilihat pada gambar II.3.



Gambar II.3. Data dan Informasi.

**(Sumber : Database Management Menggunakan Microsoft Access 2003;
2010 : 2)**

III.8. Mengenal Visual Basic

Visual Basic dibuat oleh microsoft, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, Visual Basic juga cukup handal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi database. Visual basic merupakan bahasa pemrograman event drive, dimana program akan menunggu sampai ada respons dari user/pemakai program aplikasi yang dapat berupa kejadian atau event, misalnya ketika user mengklik tombol atau menekan enter. Jika kita membuat aplikasi dengan visual basic maka kita akan mendapatkan file yang menyusun aplikasi tersebut, yaitu :

1. File Project (*.vbp)

File ini merupakan kumpulan dari aplikasi yang kita buat. File project bisa berupa file *.frm, *.dsr atau file lainnya.

2. File Form (*.frm)

File ini merupakan file yang berfungsi untuk menyimpan informasi tentang bentuk form maupun interface yang kita buat (Edy Winarto ; 2010 : 1).

II.9. SQL Server 2008

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang di *develop* oleh *Microsoft*, yang digunakan untuk menyimpan dan mengolah data. Pada SQL Server 2008, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada SQL Server 2008, kita bisa membuat *object – object* yang sering digunakan pada aplikasi bisnis, seperti membuat *database, table, fuction, stored procedure, trigger* dan *view*. Selain *object*, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data (Cybertron Solution ; 2010 : 101).

II.10. UML (*Unified Modeling Language*)

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 industry perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML adalah sebagai berikut :

1. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut (Windu Gata ; 2013 : 4).

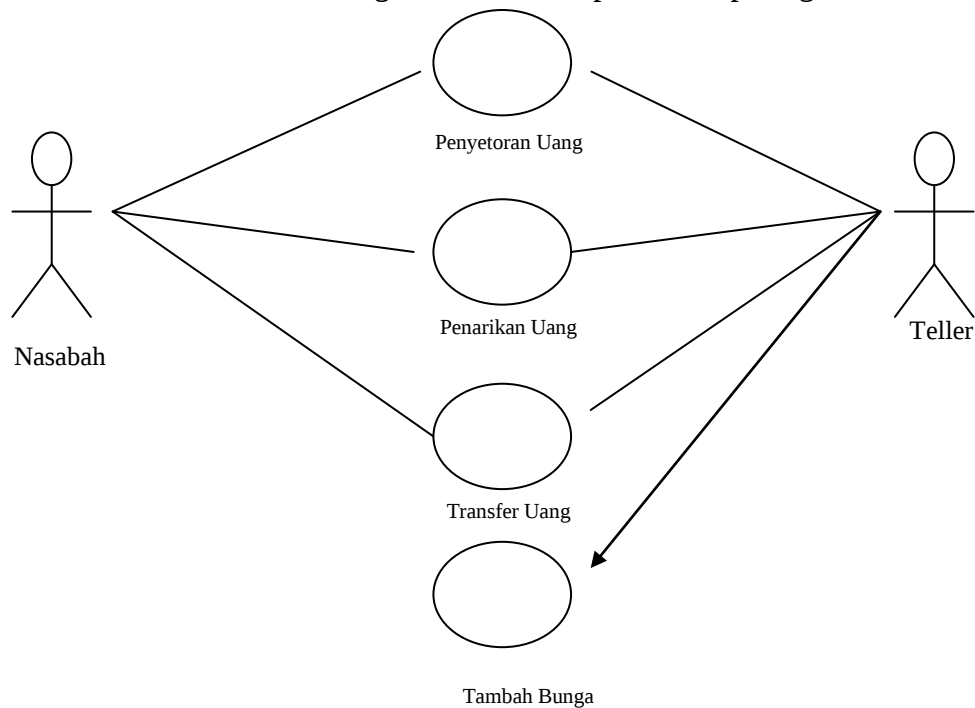
Diagram Use Case (use case diagram)

Use Case menggambarkan *external view* dari sistem yang akan kita buat modelnya. Menurut Pooley (2005:15) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram.

Komponen pembentuk diagram *use case* adalah :

- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use Case*, aktivitas/ sarana yang disiapkan oleh bisnis/sistem.
- c. Hubungan (*Link*), aktor mana saja yang terlibat dalam *use case* ini.

Salah satu contoh bentuk diagram *use case* dapat dilihat pada gambar II.4. :

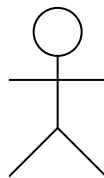


Gambar II.4. Diagram *Use Case*

Sumber : Probowo Pudjo Widodo (2011:17)

1. Aktor

Menurut Chonoles (2003 :17) menyarankan sebelum membuat use case dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.

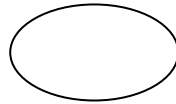


Gambar II.5. Aktor

Sumber : Probowo Pudjo Widodo (2011:17)

2. *Use Case*

Menurut Pilone (2005 : 21) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. Sedangkan menurut Whitten (2004 : 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval*

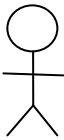
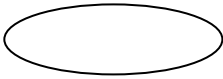
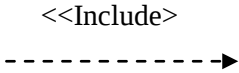
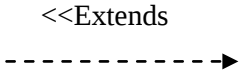


Gambar II.6. Simbol *Use Case*

Sumber : Probowo Pudjo Widodo (2011:22)

Berikut Simbol-simbol *Use Case* dapat dilihat pada tabel II.2. :

Tabel II.2. Tabel *Simbol-simbol Use case*

NAMA	KETERANGAN	SIMBOL
<i>Actor</i>	Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang kita kembangkan	
<i>Use Case</i>	Peringkat Tertinggi dari fungsional yang dimiliki sistem.	
<i>Relasi Asosiasi</i>	Relasi yang terjadi antara aktor dengan use case biasanya berupa asosiasi.	
<i>Include Relationship</i>	Relasi cakupan memungkinkan suatu use case untuk menggunakan fungsionalitas yang disediakan oleh use case lainnya.	
<i>Extends Relationship</i>	Memungkinkan suatu use case memiliki kemungkinan untuk memperluas fungsional yang disediakan use case yang lainnya.	

Sumber : Adi Nugroho(2005:49)

2. Diagram Aktivitas (*Activity Diagram*)

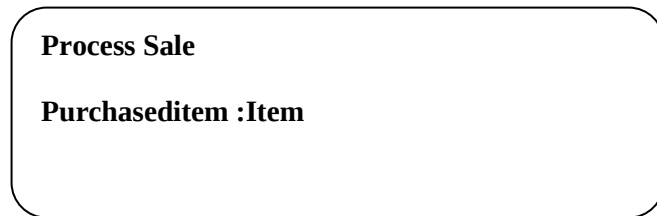
Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore (Probowo Pudji Widodo ; 2011 : 143-145).

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

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

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

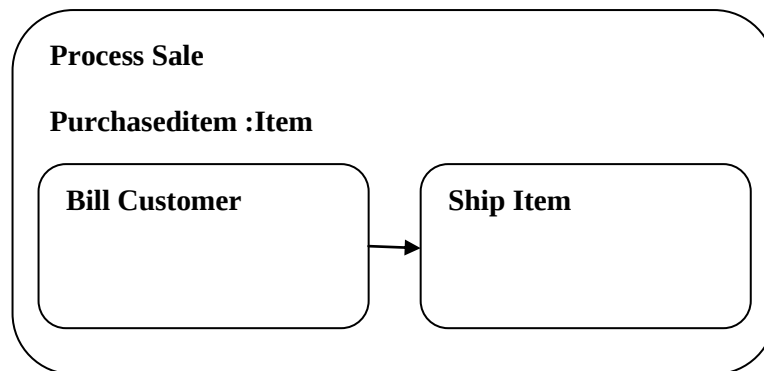
Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.7. Aktivitas sederhana tanpa rincian

Sumber : Probowo Pudjo Widodo (2011:145)

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



Gambar II.8. Aktivitas dengan detail rincian

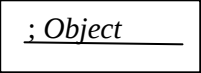
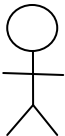
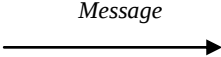
Sumber : Probowo Pudjo Widodo (2011:145)

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 (Windu Gata ; 2013 : 7).

Berikut notasi sequen diagram dapat dilihat pada tabel II.3. :

Tabel II.3. Notasi Sequence Diagram

Object	<i>Object</i> merupakan instance dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama obyek di dalamnya yang diawali dengan sebuah titik koma.	
Actor	<i>Actor</i> juga dapat berkomunikasi dengan objek, maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>Actor</i> sama dengan simbol pada <i>Actor Use Case Diagram</i> .	
Lifeline	<i>Lifeline</i> mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah obyek.	
Activation	<i>Activation</i> dinotasikan sebagai kotak segi empat yang digambar pada sebuah <i>Lifeline</i> . <i>Activation</i> mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.	
Message	<i>Message</i> digambarkan dengan anak panah horizontal antara <i>activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object-object</i> .	

Sumber : Adi Nugroho (2005 : 92)

4. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggungjawab 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 (Windu Gata ; 2013 : 8).

Diagram kelas mempunyai dua jenis yaitu *domain class diagram* dan *design class diagram*. Fokus *domain class diagram* adalah pada sesuatu dalam lingkungan kerja pengguna, bukan pada *class* perangkat lunak yang nantinya akan anda rancang. Sedangkan *design class diagram* tujuannya adalah untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun.



Gambar II.9. Notasi *Domain Diagram Class*

Sumber : E. Triandini dan G. Suardika (2012 : 49 -50)



Gambar II.10. Notasi *Design Diagram Class*

Sumber : E. Triandini dan G. Suardika (2012 : 49 -50)