

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Secara leksikal, sistem berarti susunan yang teratur dari pandangan, teori, asas dan sebagainya. Dengan kata lain, 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 antara bagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interpenden satu sama lain. Selain itu dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan-perubahan yang terus menerus perlu dikembangkan dan dikendalikan. Definisi tersebut menunjukkan bahwa sistem sebagai gugus dan elemen-elemen yang saling berinteraksi secara teratur dalam rangka mencapai tujuan atau subtujuan (Marimin ; 2008 : 1).

II.2. Informasi

Informasi menjadi penting, karena berdasarkan informasi itu para pengelola dapat mengetahui kondisi obyektif perusahaannya. Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan cara tertentu. Informasi disajikan dalam bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan yang dibutuhkan untuk menambah wawasan bagi pemakainya guna mencapai suatu tujuan.

Sebagai contoh, data dapat berupa nama karyawan, jumlah jam kerja, dan lain sebagainya. Jika banyaknya jam kerja dikalikan dengan besarnya upah per jam, maka akan diperoleh gaji kotor. Jika gaji kotor tersebut dijumlahkan, maka akan diperoleh total gaji kotor. Setelah pemrosesan dilakukan terhadap data, maka akan diperoleh informasi yang dapat mengungkapkan tentang gaji kotor per karyawan dan total biaya gaji yang harus disediakan oleh perusahaan.

Informasi tersebut diperlukan sebagai dasar pertimbangan para pengelola organisasi dalam mengambil keputusan manajerial dan strategis.

Pengolahan data menjadi informasi itu merupakan suatu siklus, yang terdiri dari tahap-tahap sebagai berikut :

1. Pengumpulan data. Pada tahap ini dilakukan suatu proses pengumpulan data yang asli dengan cara tertentu, seperti sampling, data transaksi, data *warehouse*, dan lain sebagainya yang biasanya merupakan proses pencatatan data ke dalam suatu *file*.
2. *Input*. Tahap ini merupakan proses pemasukan data dan prosedur pengolahan data ke dalam komputer melalui alat *input* seperti *keyboard*. Prosedur pengolahan data itu merupakan urutan langkah untuk mengolah data yang ditulis dalam suatu bahasa pemrograman yang disebut program.
3. *Pengolahan data*. Tahap ini merupakan tahap di mana data diolah sesuai dengan prosedur yang telah dimasukkan. Kegiatan pengolahan data ini meliputi pengumpulan data, klasifikasi (pengelompokkan), kalkulasi, pengurutan, penggabungan, peringkasan baik dalam bentuk table maupun

grafik, penyimpanan dan pembacaan data dari tempat penyimpanan data (Budi Sutedjo Dharma Oetomo ; 2006 : 12).

II.3. Sistem Informasi

Sistem Informasi dapat didefinisikan sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi. Dengan kata lain, SI merupakan kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi yang akan mendukung pembuatan keputusan dan melakukan kontrol terhadap jalannya perusahaan.

Sistem Informasi juga mampu mendukung para pengelola dan staf perusahaan untuk menganalisa permasalahan, memvisualisasikan ikhtisar analisa melalui grafik-grafik dan tabel-tabel, serta memungkinkan terciptanya produk serta layanan yang baru. Sistem Informasi yang baik tentu memiliki sistematika yang jelas, ringkas, dan sederhana. Mulai dari tahap pemasukan data, pengolahan dengan prosedur yang ditentukan, penyajian informasi yang akurat, interpretasi yang tepat dan distribusinya (Budi Sutedjo Dharma Oetomo ; 2006 : 11).

II.4. Produksi

Bagi orang awam, yang dimaksud dengan kegiatan produksi adalah kegiatan untuk menghasilkan barang. Misalnya, kegiatan membuat kursi, membuat baju, atau merakit mobil. Pengertian tersebut sebenarnya tidak salah, hanya saja kurang lengkap. Selain menghasilkan barang atau jasa, kegiatan

produksi juga harus menambah manfaat atau nilai guna suatu barang dan jasa. Dengan demikian, produksi adalah kegiatan yang dilakukan oleh orang atau lembaga untuk menghasilkan atau menambah manfaat (nilai guna) suatu barang dan jasa. Orang atau lembaga yang melakukan kegiatan produksi disebut produsen. Untuk mendapatkan definisi produksi yang lebih lengkap, hal-hal berikut perlu diperhatikan.

- a. Kegiatan produksi dilakukan oleh perusahaan
- b. Tujuan perusahaan adalah untuk memperoleh laba
- c. Perusahaan mengombinasikan seluruh sumber daya
- d. Barang atau jasa yang dihasilkan perusahaan ditujukan untuk memenuhi kebutuhan manusia.

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa produksi adalah kegiatan yang dilakukan perusahaan untuk memperoleh laba dengan mengombinasikan seluruh sumber daya ekonomi untuk menghasilkan barang atau jasa yang ditujukan untuk memenuhi kebutuhan manusia (Deliarnov ; 2007 : 45).

II.5. Mebel

Mebel/Furnitur adalah istilah kolektif untuk objek bergerak yang mendukung tubuh manusia (tempat duduk furnitur dan tempat tidur), menyediakan penyimpanan, dan memegang benda pada permukaan horisontal di atas tanah. Penyimpanan furniture digunakan untuk menahan atau berisi objek yang lebih kecil seperti alat-alat, buku, dan barang rumah tangga. Furniture dapat menjadi produk seni dan dianggap sebagai bentuk seni dekoratif.

Pengertian Mebel/Furniture adalah kata benda massa untuk objek bergerak dimaksudkan untuk mendukung berbagai aktivitas manusia seperti tempat duduk dan tidur ditempat tidur , untuk memegang benda pada ketinggian yang nyaman untuk bekerja menggunakan permukaan horisontal di atas tanah, atau untuk menyimpan sesuatu. Penyimpanan furniture seperti meja sering memanfaatkan pintu, laci, rak dan kunci mengandung, mengorganisir atau mengamankan benda-benda kecil seperti pakaian, peralatan, buku, dan barang rumah tangga. Furniture dapat dibuat dari banyak bahan, termasuk logam, plastik, dan kayu. Furniture dapat dibuat menggunakan berbagai sendi kayu yang sering mencerminkan budaya local (Putri Ardhanareshwari ; 2011 : 41).

II.6. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai jenis komputer dan berbagai sistem operasi termasuk telepon genggam. Java dikembangkan suatu teknologi perangkat lunak yang digolongkan *multi platform*. Selain itu, Java juga merupakan suatu *platform* yang memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan suatu program.

Adapun kelebihan Java adalah :

- a. *Multiplatform*. Kelebihan utama Java adalah dapat dijalankan di beberapa *platform*/sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan dimana saja.
- b. OOP (*Object Oriented Programming*-Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di Java adalah objek. Java merupakan salah satu bahasa pemrograman berbasis secara objek.

- c. *Class library* yang lengkap, Java terkenal dengan kelengkapan *library* / perpustakaan (kumpul program yang disertakan dalam pemrograman Java) yang sangat memudahkan dalam penggunaan oleh para programmer untuk membangun aplikasinya.
- d. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrograman C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrograman C++ yang pindah ke Java.
- e. Pengumpulan sampah otomatis, memiliki fasilitas pengatur penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

Sedangkan kekurangan Java adalah :

- a. Tulis sekali, perbaiki di mana saja. Yang dimaksud ungkapan tersebut adalah masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain.
- b. Mudah didekompilasi. Dekompilasi adalah proses mebalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi.
- c. Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar dibandingkan bahasa tingkat tinggi generai sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal) (Wahana Komputer ; 2010 : 1).

II.7. 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.8. MySql

Menurut Wahana Komputer (2010), Mysql pertama kali dirintis oleh seorang programmer *database* bernama Michael Widenius, yang dapat anda hubungi di emailnya monty@analytikerna. Mysql 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.

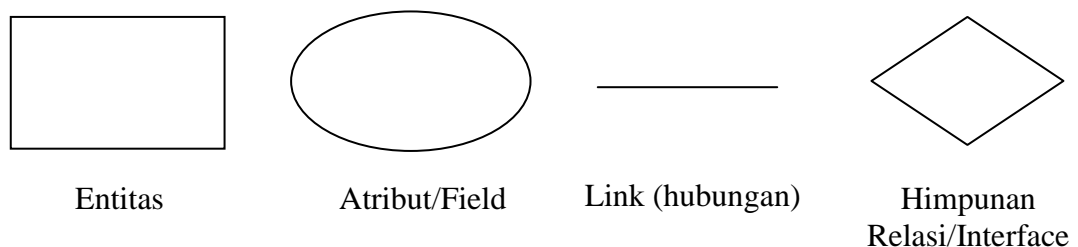
MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. penulis sendiri dalam menjelaskan buku ini

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).

II.9. Entity Relationship Diagram (ERD)

Entity Relationship 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 analis 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 : Yuniar Supardi ; 2010 : 448)

II.10. Kamus Data

Dalam tahapan ini unsur-unsur data yang diperlukan ditentukan untuk selanjutnya dijelaskan lagi dikamus data (*data dictionary*). Pengertian kamus data itu sendiri adalah suatu ensiklopedik dari informasi yang berkaitan dengan data perusahaan atau dapat juga kita katakana bahwa komputer (*computer-based catalog or directory*) yang berisi data perubahan (*metadata*) yang berkenaan dengan tahapan penjelasan data ini adalah sistem kamus data (*data dictionary system/DDS*) dan bahasa pendeskripsi data (*data description language/DDL*).

Sistem kamus data berbentuk perangkat lunak yang fungsinya adalah penciptaan dan pemerilahaan serta penyediaan kamus data agar dapat digunakan. Kamus data dapat berbentuk kertas ataupun arsip (*file*) komputer. DDS dapat kita peroleh dalam paket perangkat lunak terpisah ataupun dalam bentuk modul seperti yang ada dalam DBMS (*database management system*) dan CASE (teknik perangkat lunak tambahan komputer / *computer aided software engineering*) (Ian Soomerville ; 2010 : 344).

II.11. Teknik Normalisasi

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*.

Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sesedikit mungkin redundansi.
3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.

4. Mengefisienkan update.
5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal.

II.11.1. 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)

Definisi :

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- 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.

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.

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 :

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

e. Bentuk Normal Tahap Keempat dan Kelima

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 berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

f. Boyce Code Normal Form (BCNF)

- Memenuhi 1st NF
- Relasi harus bergantung fungsi pada atribut superkey (Kusrini ; 2007 : 39-43).

II.12. *Unified Modeling Language (UML)*

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. (Chonoles; 2003 : 6) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan –aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model

untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Herlawati ; 2011 ; 6).

II.12.1. Diagram-Diagram UML

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

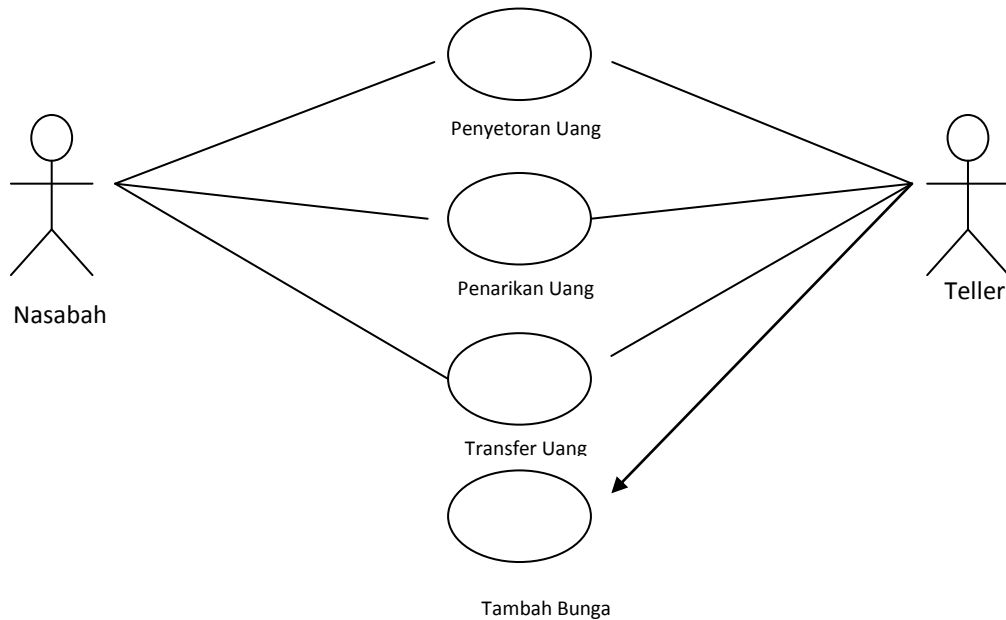
1. Diagram Use Case (use case diagram)

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.

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.

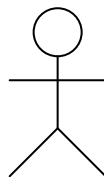
Gambar di bawah ini merupakan salah satu contoh bentuk diagram *use case*.



Gambar II.2. Diagram *Use Case*
 Sumber : Probowo Pudjo Widodo (2011:17)

a. Aktor

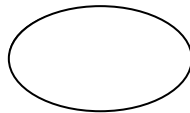
Menurut Chonoles (2003 :17) menyarankan sebelum membuat use case dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.



Gambar II.3. Aktor
 Sumber : Probowo Pudjo Widodo (2011:17)

b. Use Case

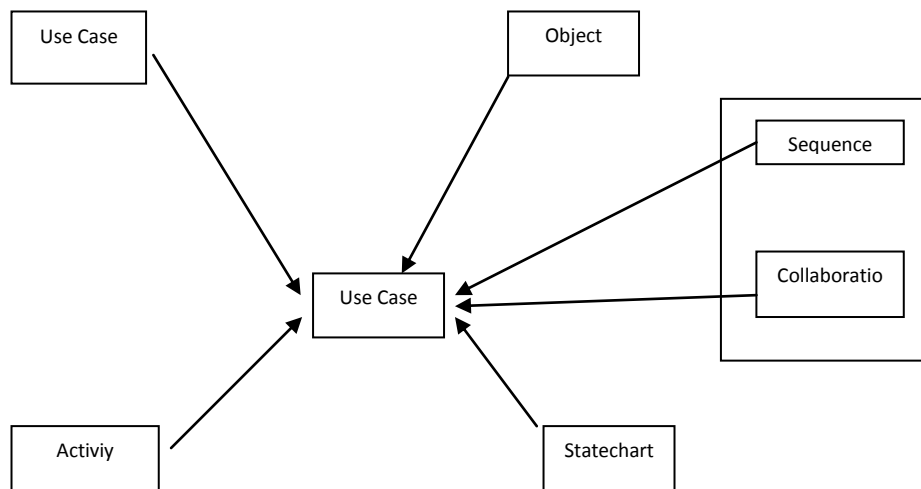
Menurut Pilone (2005 : 21) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. Sedangkan menurut Whitten (2004 : 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual,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



Gambar II.4. Simbol *Use Case*
Sumber : Probowo Pudjo Widodo (2011:22)

2. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini *forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model (Probowo Pudji Widodo; 2011 : 37).



Gambar II.5. Hubungan Diagram Kelas Dengan Diagram UML lainnya

Sumber : Probowo Pudjo Widodo (2011 : 38)

3. 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 pengajian tiap jumat sore (Probowo Pudji Widodo ;2011 : 143-145).

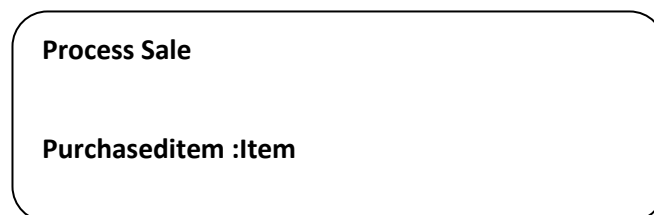
Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

- 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.

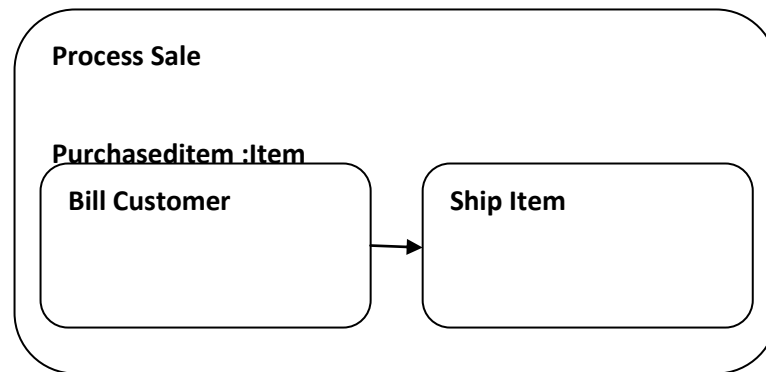
Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.6. Aktivitas serderhana 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.

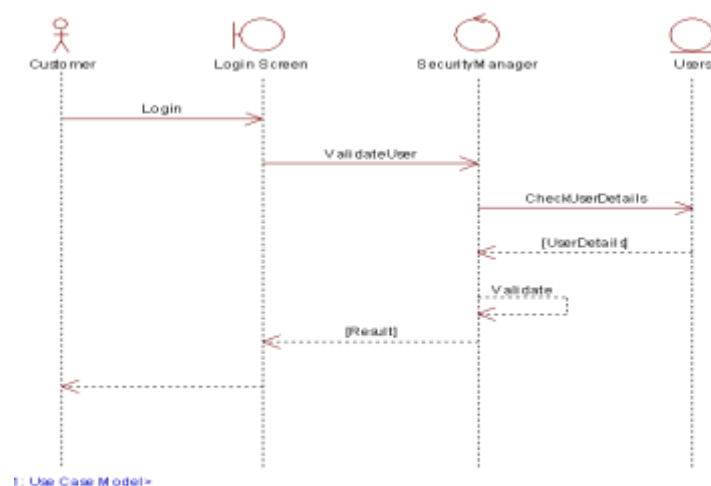


Gambar II.7. Aktivitas dengan detail rincian
 Sumber : Probowo Pudjo Widodo (2011:145)

4. *Sequence Diagram*

Menurut Douglas (2004 : 174) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*).

Menurut Pilone (2005 : 174) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.8. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya.



Gambar II.8. Diagram Urutan
 Sumber : Probowo Pudjo Widodo (2011:175)