

BAB II

TINJAUAN PUSTAKA

II.1 Pengertian Sistem .

Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Tata Sutabri : 2005 : 8).

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

a. Komponen Sistem(*Components*)

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem (Tata Sutabri : 2005 : 11).

b. Batas Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisah-pisahkan (Tata Sutabri : 2005 : 11).

c. Lingkungan Luar Sistem (*Environtment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sitem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. (Tata Sutabri : 2005 : 11).

d. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain (Tata Sutabri : 2005 : 12).

e. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). (Tata Sutabri : 2005 : 12).

f. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi pengeluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain (Tata Sutabri : 2005 : 12).

g. Pengolahan Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran (Tata Sutabri : 2005 : 12).

h. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan (Tata Sutabri : 2005 : 12).

II.1.2. Pengertian Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan (Tata Sutabri : 2005 : 23).

Kualitas suatu informasi tergantung dari 3 (tiga) hal yaitu, informasi harus akurat (*accurate*), tepat waktu (*timelines*), dan relevan (*relevance*). Penjelasan tentang kualitas informasi tersebut akan dijelaskan dibawah ini.

a. Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dapat dari sumber informasi sampai penerima informasi.

b. Tepat waktu (*timelines*)

Informasi yang datang pada si penerima tidak boleh terlambat. Mahalnya informasi disebabkan karena harus cepatnya informasi tersebut dikirim atau didapatkan sehingga diperlukan teknologi mutakhir untuk mendapatkan, mengolah, dan mengirimkannya.

c. Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainnya. Relevansi informasi untuk orang satu dengan yang lain berbeda, misalnya informasi sebab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila ditunjukkan kepada ahli teknik perusahaan.

II.1.3. Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Tata Sutabri : 2005 : 42).

II.1.4. Pengertian Akuntansi

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

II.1.5. Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah aktivitas yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan (Anastasia Diana dan Lilis Setiawati : 2011 : 4).

II.1.6. Pengertian Jasa Service

Jasa service adalah setiap tindakan atau kegiatan yang ditawarkan oleh suatu pihak ke pihak yang lain yang pada dasarnya tidak terwujud dan tidak mengakibatkan kepemilikan apapun. Produksinya dapat dikaitkan atau tidak dikaitkan pada suatu produk fisik.

(http://thesis.binus.ac.id/doc/Bab2/Bab%202_09-151.pdf).

II.1.7. Pengertian Java

Java adalah bahasa suatu jenis teknologi pemrograman yang dikembangkan oleh SUN Microsystems. Teknologi java dapat digunakan untuk membuat aplikasi database, web, jaringan, ataupun grafis. Java merupakan bahasa pemrograman yang sangat andal. Keandalannya terbukti sesuai dengan slogannya yaitu *“Write Once Run Everywhere”* yang mana penulisan kode program dalam bahasa java dapat dijalankan pada berbagai sistem operasi.

(Andi :2012 : 2).

II.1.8. Pengertian NetBeans

Netbeans adalah salah satu Integrated Development Environment (IDE) yang dikembangkan dengan bahasa pemrograman java. Netbeans mempunyai lingkup pemrograman yang diintegrasikan dalam suatu aplikasi pemrograman Graphic User Interface (GUI), suatu kode editor atau text, compiler dan interpreter. (Andi : 2012 : 28)

II.2. Unified Modeling Language (UML)

UML (*Unified sistem Modeling Language*) adalah suatu alat bantu yang sangat handal di dunia pengembangan yang berorientasi objek (Munawar : 2005 : 17). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain. Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan suatu aplikasi, tidak berarti bahwa

semua diagram tersebut akan bisa menjawab persoalan yang ada. Adapun tipe diagram UML yang ada seperti pada Tabel II.1.

Tabel II.1 Tipe Diagram UML

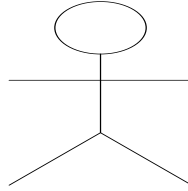
Diagram	Tujuan	Keterangan
Activity	Prilaku prosedural dan parallel	Sudah ada di UML 1
Class	Class, fitur dan relasinya	Sudah ada di UML 1
Communication	Interaksi diantara objek. Lebih menekankan kepada link	Di UML 1 disebut collaboration
Component	Struktur dan koneksi dari komponen	Sudah ada di UML 1
Composite Structure	Dekomposisi sebuah class saat runtime	Baru untuk UML 2
Deployment	Penyebaran/instalasi ke klien	Sudah ada di UML 1
Interaction Overview	Gabungan dari activity dan sequence diagram	Baru untuk UML 1
Object	Contoh konfigurasi instance	Tidak resmi ada di UML 1
Package	Struktur hierarki saat kompilasi	Tidak resmi ada di UML 1
Sequence	Interaksi antara objek. Lebih menekankan pada urutan.	Sudah ada di UML 1
State Machine	Bagaimana event mengubah sebuah objek	Sudah ada di UML 1
Timing	Interaksi antar objek. Lebih menekankan pada waktu	Sudah ada di UML 1
Use Case	Bagaimana user berinteraksi dengan sebuah sistem	Sudah ada di UML 1

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2005 : 23)"

II.3. Notasi Dasar UML

1. Actor

Actor adalah *abstraction* dari orang dan *system* yang lain yang mengaktifkan fungsi dari target *system*. Orang atau *system* bisa muncul dalam beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Berikut notasi actor dalam UML:

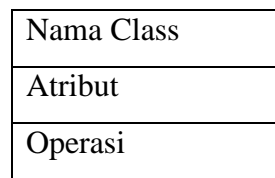


Gambar II.1 : Notasi Actor pada UML

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2005: 41)"

2. Class

Class, dalam notasi UML digambarkan dengan kotak. Nama class menggunakan huruf besar diawal kalimatnya dan diletakkan diatas kotak. Bila class mempunyai nama yang terdiri dari 2 suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar. Berikut notasi class dalam UML:



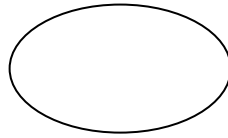
Gambar II.2 : Notasi Class pada UML

Sumber : " Pemodelan Visual dengan UML (Munawar ; 2009 : 41)"

3. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah – langkah yang menerangkan

antara pengguna dan sistem disebut skenario. Notasi *use case* dapat dilihat pada gambar II.3.

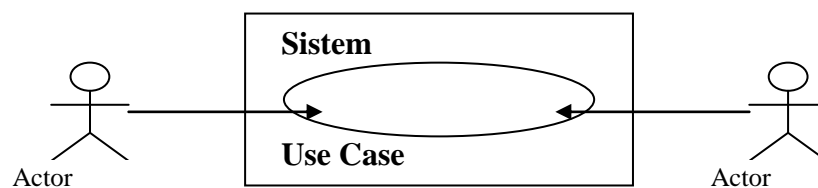


Gambar II.3 : Notasi *Use Case* pada UML

Sumber : ”Pemodelan Visual dengan UML (Munawar : 2009 : 42)”

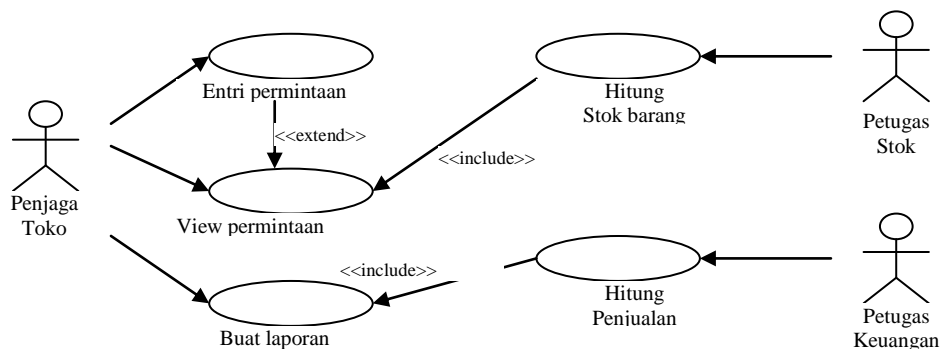
4. *Use Case Diagram*

Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara actor dengan sistem. *Use Case* menggambarkan kata kerja seperti *Login* ke sistem, *maintenance user* dan sebagainya. Model use case seperti gambar 4 dan contoh use case diagram ditunjukkan pada Gambar II.4.



Gambar II.4 : Model *Use Case* pada UML

Sumber : ” Pemodelan visual dengan UML (Munawar ; 2009 : 43)”



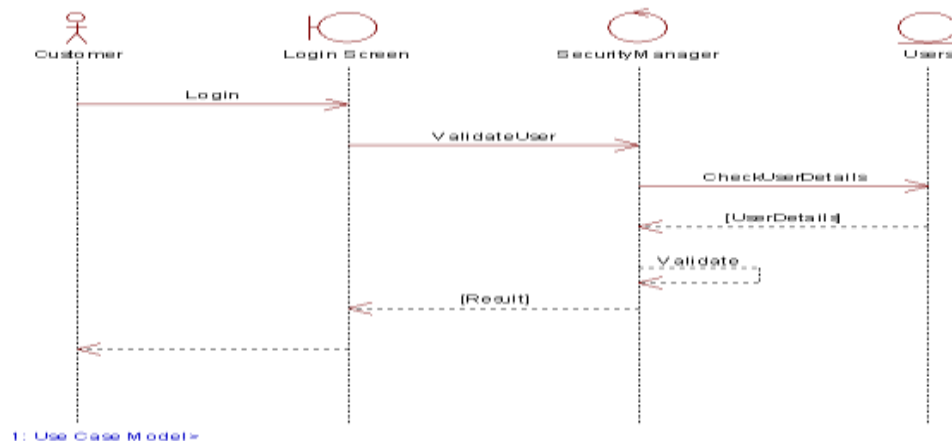
Gambar II.5 : Use Case diagram pada penjualan CD

Sumber : " *Pemodelan visual dengan UML (Munawar : 2009 : 43)* "

4. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini dalam *use case*.

Komponen utama squence diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. Message dengan tanda panah dan waktu yang ditunjukkan dengan progress vertical. Berikut Contoh *sequence diagram* :



Gambar II.6 : Sequence Diagram

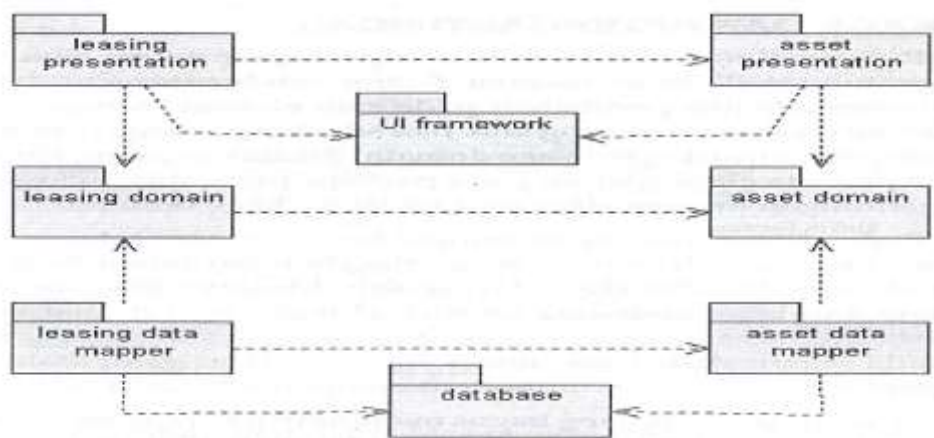
Sumber : " *Pemodelan visual dengan UML (Munawar : 2009 : 44)* "

6. Class Diagram

Sama seperti *class*, maka class diagram merupakan diagram yang selalu ada di pemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Class diagram umumnya tersusun dari elemen *class*, *interface*, *dependency*, *Generalization* dan *Association*. Relasi *dependency* menunjukkan bagaimana terjadi ketergantungan antar *class* yang ada. Relasi *Generalization* menunjukkan bagaimana suatu *class* menjadi *superclass* dari *class* lainnya dan *class* tersebut menjadi *subclass* dari *class* tersebut. Relasi *Association* menggambarkan navigasi antar *class*, berapa banyak objek lain bisa berhubungan dengan satu objek (*multiplicity* antar *class*), dan apakah satu *class* menjadi bagian dari *class* lainnya (*agregation*). *Class diagram* digunakan untuk menggambarkan disain statis dari sistem yang sedang dibangun.

Contoh *class diagram* :



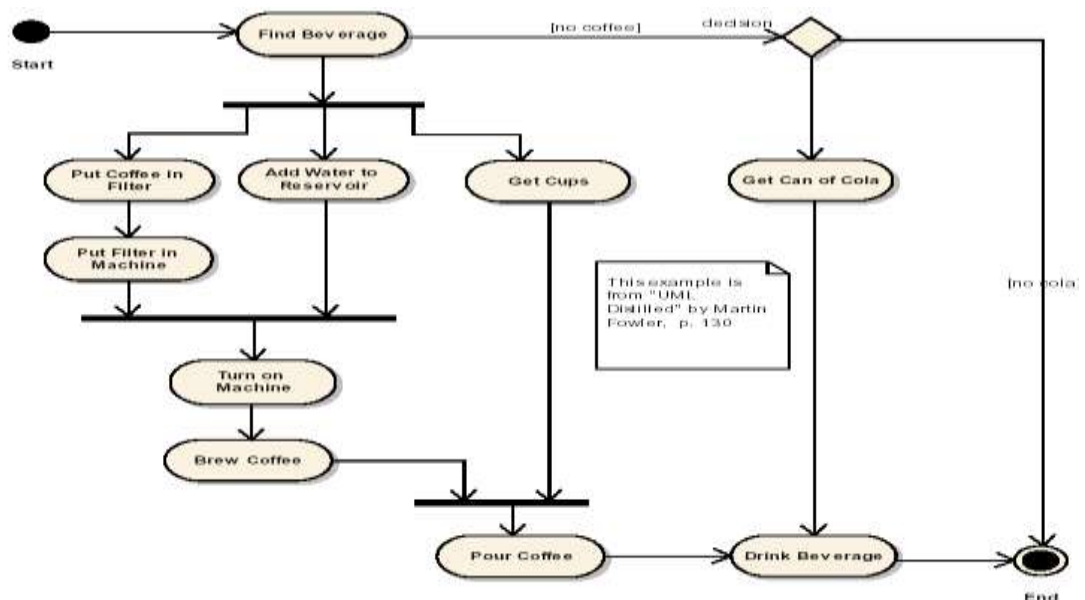
Gambar II.7 : Class Diagram

Sumber : " *Pemodelan visual dengan UML (Munawar : 2009 : 45)* "

7. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem dan interaksi antar subsistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standart UML menggunakan segi empat dengan

sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu, digambarkan dengan simbol belah ketupat. Untuk mengilustrasikan proses-proses paralel (*fork and join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Adapun contoh dari Activity Diagram dapat di lihat pada Gambar II.8.



Gambar II.8 : Activity Diagram

Sumber : " Pemodelan visual dengan UML (Munawar : 2009 : 47)"

II.4. Pengertian Database

Suatu kumpulan data terhubungan (interrelated data) yang disimpan secara bersama-sama pada suatu media tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data(controlled redundancy) dengan cara tertentu sehingga mudah digunakan oleh satu atau lebih program aplikasi secara optimal data

tersimpan tanpa mengalami ketergantungan pada program yang akan digunakan. (Tata Sutabri : 2005 : 161).

Dari pengertian tersebut dapat disimpulkan bahwa sistem database mempunyai beberapa karakteristik yang penting yaitu:

- a. Bersifat data oriented dan bukan program oriented.
- b. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah databasenya.
- c. Dapat berkembang dengan mudah baik volume maupun strukturnya.
- d. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah.
- e. Dapat digunakan dengan cara-cara berbeda.
- f. Kerengakapan data (*data redundancy*) minimal.

II.5. MySQL

MySQL adalah RDBMS (*Relational Database Management System*) yang didistribusikan secara gratis dibawah ini lisensi GPL (*General Public Lisensi*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* (komresial).

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). Sql adalah sebuah konsep pengoprasian database, terutama untuk pemilihan/seleksi dan pemasukan

data yang memungkinkan pengoprasian data dikerjakan dengan mudah secara otomatis.