

BAB II

TINJAUAN PUSTAKA

II.1. Landasan Teori

Landasan teori merupakan panduan untuk melaksanakan dan menyesuaikan suatu studi. Dalam hal ini penulis akan mengemukakan beberapa teori yang berkaitan dengan masalah yang akan di bahas. Teori-teori yang akan dikemukakan yang merupakan dasar bagi penulis untuk meneliti masalah yang dihadapi penulis pada saat melaksanakan penelitian di perusahaan.

II.1.1. Pengertian Sistem

Istilah sistem berasal dari kata Yunani yaitu “ *Systema* “. Ditinjau dari sudut katanya sistem berarti sekumpulan objek yang bekerja secara bersama-sama untuk menghasilkan satu kesatuan metode, prosedur, teknik yang digabungkan dan diatur sedemikian rupa sehingga menjadi satu kesatuan yang berfungsi untuk mencapai suatu tujuan. Untuk memahami istilah sistem yang dimaksud secara defenitif akan di jelaskan pengertian sitem yang telah dikemukakan oleh beberapa ahli sebagai berikut :

1. Menurut Kurnisi, 2006 sistem adalah suatu elemen-elemen yang saling berkaitan dan terpadu .
2. Menurut (Kusrini M.Kom; 2009:11) Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*)

Dengan berbagai teori tersebut bahwa suatu sistem yang dirancang tidak akan bisa berdiri sendiri dan akan selalu memiliki ketergantungan terhadap komponen apapun subsistem yang lain.

II.1.2. Karakteristik Sistem

Model umum sebuah sistem adalah input, proses dan output. Hal ini merupakan sebuah konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran.

1. Komponen Sistem (*Components*)

Merupakan suatu sistem yang terdiri dari sejumlah komponen-komponen yang saling berintegrasi, yang artinya saling bekerjasama untuk membentuk satu kesatuan komponen atau elemen yang sistem yang dapat berupa satu sub sistem atau bagian dari sistem.

2. Batasan Sistem

Merupakan batasan daerah yang membatasi antara sub sistem yang lain atau dengan lingkaran luarnya.

3. Lingkaran Luar Sistem (*Environment*)

Adalah hak apapun di luar batas dari sistem yang mempunyai operasi sistem.

4. Penghubung Sistem (*Interface*)

Merupakan suatu media penghubung antara sub sistem satu dengan sub sistem yang lain. Dengan melalui penghubung ini memungkinkan sumber

daya mengalir dari sub sistem ke sub sistem yang lain dengan melalui penghubung.

5. Masuk Sistem

Adalah suatu energy yang di butuhkan dalam sistem masukan dapat berupa masukan perawatan (*maintenance*) dan masukan sinyal (*signal input*)

6. Keluaran Sistem (*Output*)

Merupakan suatu hasil yang terdapat dari energy yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolahan Sistem

Suatu sistem yang dapat mempunyai bagian pengolahan yang akan berupa menjadi luaran.

8. Sasaran Sistem

Suatu sistem yang mempunyai tujuan dan sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka tidak akan ada gunanya. Sistem yang dirancang guna mencapai suatu sasaran desain berdasarkan atau data yang ada dilapisan atau data yang bersangkutan dengan sistem tersebut.

II.1.3. Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Arshar, 2010).

Menurut (Kusrini M.Kom;2009:4) informasi merupakan hasil olahan data, dimana data tersebut sudah diperoleh dan diinterpretasikan menjadi suatu yang bermakna untuk mengambil keputusan. Informasi juga diartikan sebagai

himpunan dari data yang relevan dengan satu atau beberapa orang dalam suatu waktu.

Kegunaan informasi bagi seseorang juga sangat tergantung pada waktu. Pada suatu waktu tertentu informasi tersebut mungkin sangat diperlukan dilain hari, mungkin saja hal tersebut sudah tidak berguna sama sekali. Contohnya, informasi perbandingan harga barang akan sangat dibutuhkan oleh seseorang yang akan membeli barang tersebut. Namun saat ini dia sedang tidak mempertimbangkan untuk membeli barang tersebut, informasi tersebut menjadi kurang bermakna.

II.1.4. Kualitas Informasi

Kualitas dari suatu informasi tergantung tiga hal yaitu, informasi harus akurat, tepat pada waktunya dan relevan. Yang dimaksud dengan akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Sedangkan tepat pada waktunya berarti, informasi yang datang pada penerima tidak boleh terlambat. Yang terakhir adalah relevan, berarti informasi tersebut mempunyai manfaat untuk pemakaiannya.

II.1.5. Sistem Informasi

Menurut (Kusrini M.Kom; 2009:11) Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan untuk pihak luar.

Berdasarkan dukungan kepada pemakaiannya, sistem informasi dibagi menjadi :

1. Sistem Pemrosesan Transaksi (*Transaction Processing System*) atau TPS
2. Sistem Informasi Manajemen (*Management Informasi System*) atau MTS
3. Sistem Otomasi Perkantoran (*Office Automation System*) atau OAS
4. Sistem Pendukung Keputusan (*Decision Support System*) atau DDS
5. Sistem Informasi Eksekutif (*Executive Informasi System*) atau EIS
6. Sistem Pendukung Kelompok (*Group Support System*) atau GSS
7. Sistem Pendukung Cerdas (*Intelegent Support system*) atau ISS

Mengingat bahwa EIS, DSS, dan MIS digunakan untuk pendukung manajemen, maka ketiga sistem tersebut sering disebut Sistem Pendukung manajemen (*management support system*) atau MSS.

informasi yang datang pada penerima tidak boleh terlambat. Yeng terakhir adalah relevan, berarti informasi tersebut mempunyai manfaat untuk pemakaiannya.

II.2. Sejarah Visual Basic

Menurut Harip Santoso (2006 : 3), diluncurkan *VB.Net 2002* (Pemrograman berbasis teknologi.Net), dapat dianggap sebagai pertanda berakhirnya pengembangan *Visual Studio 6*, karna secara *defacto* VB6 sudah tidak dikembangkan lagi oleh *Microsoft*. Walaupun VB6 tidak dikembangkan lagi, tetapi kesinambungan masa hidup VB6 masih dijaga oleh *Microsoft*. Contohnya dengan diluncurkannya *Service Pack 5* untuk *Visual Studio 6* agar bahasa

pemrograman yang terdapat pada *Visual Studio 6* dapat mengakses *database SQL Server 2005*.

Kini teknologi pemrograman telah dialihkan ke pemrograman berbasis NET, teknologi yang intinya adalah menulis program agar dapat dijalankan pada piranti lunak maupun keras yang berbeda tanpa harus mengubah program yang sudah dikompilasi. Misalnya, program yang dibuat pada sistem operasi *windows*, nantinya dapat dijalankan pada sistem operasi *Unix/Linux* dan sebaliknya, dimana hal ini dimungkinkan dengan diperkenalkannya konsep *.NET Framework*.

Service Pack yang dikeluarkan oleh *Microsoft* lebih pada penyediaan fasilitas piranti tambahan sebagai media untuk mengakses/membuat hubungan dengan piranti *software* yang baru, misalnya *SQL Server 2005*. Jadi walaupun anda menginstal *Service Pack*, tidak berarti *Visual Studio 6 (VB 6)* anda akan lebih canggih. Itulah yang dimaksud dengan VB 6 tidak dikembangkan lagi oleh *Microsoft*.

Visual Basic 6 diluncurkan pada tahun 1998 dan terdiri dari tiga versi, yaitu *Learning Edition*, *Professional Edition*, dan *Enterprise*. Adapun peningkatan yang terdapat pada VB 6 dibandingkan dengan versi sebelumnya adalah, peningkatan akses data, penambahan *tools*, dan control baru untuk mengakses database, misalnya *ADODC*.

Awal sejarah *Visual Basic* dimulai pada pertengahan tahun 1991 dengan diluncurkannya VB versi 1.0. yang merupakan pengembangan dari *Quick Basic*, yaitu bahasa pemrograman *Basic* yang dimiliki oleh *Microsoft* untuk menulis program pada sistem operasi *DOS*. *QuickBasic* masih bersifat teks program, dan

belum bersifat grafik. Singkatnya, piranti mouse belum dapat digunakan dalam kaitan dengan interaksi antara pengguna dan aplikasi program.

BV 2.0 diluncurkan pada tahun 1992, dimana pada masa itu mulai diperkenalkan *ODBC (Open Database Connectivity)* sebagai metode untuk pengaksesan data, perkembangan VB 2.0 tidak terlalu nampak pada masa itu, kerna bahasa pemrograman berbasis *xBase (dBase, Foxpro, Clipper)* mendominasi penulis aplikasi program. Disamping itu sistem operasi DOS masih bergandengan erat dengan sistem operasi *Netware* yang dimiliki oleh *Novell* sebagai sistem operasi dasar pada aplikasi berbasis *LAN (Lokal Area Network)*. (Harip Santoso ;2006 : 4).

II.3. Microsoft SQL Server

Menurut Harip Santoso (2006 : 5), sejarah *SQL Server* berbeda jauh dengan sejarah *Visual Basic*. Bila *Visual Basic* berasal dari pengembangan *QuickBasic* yang juga merupakan produk *Microsoft*, maka *SQL Server* adalah hasil kerja sama antara *Microsoft* dengan *Sybase* untuk memproduksi sebuah software penyimpanan data (database) yang bekerja pada sistem operasi OS/2.

Sistem Operasi OS/2 merupakan sistem operasi baru dari hasil kerja sama antara *Microsoft* dengan *IBM*. Sistem operasi OS/2 mengenal bentuk-bentuk perintah DOS, sekaligus memiliki kemampuan *multitasking*. Untuk mendapat pengakuan pasar, maka *Microsoft* bekerja sama dengan *Ashton-Tate* yang telah dikenal dengan produksi *dBase*. Sayangnya, kerja sama ini tidak berlangsung

lama, karena setelah peluncuran *SQL Server* versi 1.0 pada tahun 1989 kerja sama itu pun berakhir.

Peluncuran *SQL Server* 1.0 dilanjutkan dengan peluncuran *SQL Server* versi 1.1 pada tahun 1990. Adapun fitur terpenting dari produk ini adalah dukungan untuk *platform* baru dari sisi client yang dikenal sebagai sistem operasi Windows 3.1

Pada tahun 1991, *SQL Server* versi 1.11 diluncurkan dan berisi fasilitas perawatan database. Pada tahun yang sama, *Microsoft* mengembangkan sendiri sistem operasi *multi user* yang dikenal sebagai *Windows NT*, *Microsoft SQL Server* versi 4.2 dirilis pada tahun 1992 dan berisi tool administrator database berbasis *GUI-Windows*.

II.4. Database

Menurut Budi Raharjo (2011:3) istilah database banyak memiliki definisi. Untuk sebagai kalangan sederhana *database* diartikan sebagai kumpulan data (buku, nomor telepon, daftar pegawai, dan lain sebagainya). Ada juga yang menyebut *database* dengan definisi lain yang lebih formal dan tegas. *Database* didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil dan secara cepat.

Selain berisi data, *database* juga berisi *metadata*. *Metadata* adalah data yang menjelaskan tentang struktur dari data itu sendiri. Sebagai contoh, anda dapat memperoleh informasi tentang nama-nama kolom dan tipe yang ditampilkan tersebut disebut *metadata*.

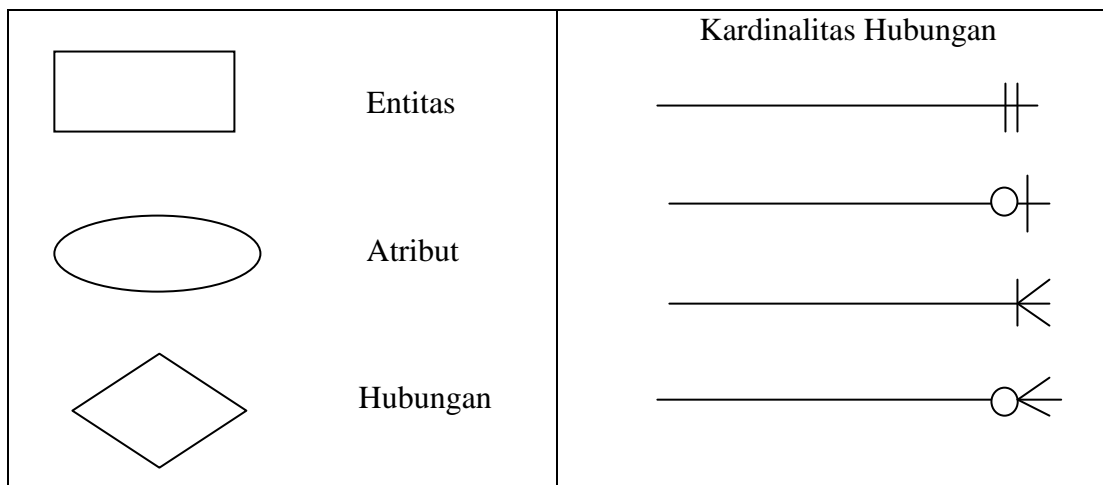
II.4.1. Pemodelan Data

Menurut Abdul Kadir (2009:30) pada perancangan konseptual diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antara data. Hubungan tersebut dapat dinyatakan dalam bentuk E-R, karena model E-R adalah dasar penting dalam merancang database maka akan dijelaskan tentang gambaran tentang model E-R, penjelasan mengenai komponen-komponen yang menyusun model E-R, hingga cara penyusunan model E-R.

II.4.2. Model E-R

Menurut Abdul Kadir (2009:30) Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antara entitas. Huruf E sendiri menyatakan entitas dan R menyatakan hubungan (dari kata *Relationship*). Model ini dinyatakan dalam bentuk diagram, itulah sebabnya model E-R sering disebut sebagai diagram E-R.

Model E-R melibatkan sejumlah notasi, beberapa notasi dasar dalam E-R ditunjukkan pada gambar II.2, notasi-notasi tersebut diberikan hanya untuk memberikan suatu pengetahuan dasar.



Gambar II.2 : Sejumlah notasi pada model E-R

Sumber : Abdul Kadir (2009:31)

1. Entitas

Yang dimaksud dengan entitas adalah sesuatu dalam dunia nyata yang keberadaannya tidak bertanggung jawab pada yang lain. Sebagai contoh, setiap pegawai dalam sebuah organisasi adalah sebuah entitas. Entitas dapat berupa sesuatu yang nyata abstrak (berupa suatu konsep). Secara lebih rinci dijelaskan bahwa entitas dapat berupa seseorang, sebuah objek, sebuah kejadian atau suatu konsep.

2. Atribut

Setiap entitas dinyatakan dalam sejumlah atribut. Atribut adalah property atau karakteristik yang terdapat pada setiap entitas.

3. Hubungan (*Relationship*)

Hubungan (*Relationship*) menyatakan ketertarikan antara beberapa tipe entitas.

II.5. Unified Modeling Language (UML)

Menurut Rosa A.S & M. Shalahudin (2011 : 118) pada perkembangan teknik pemrograman objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metode berorientasi objek.

Menurut prabowo pudjo Widodo & Herlambang (2011 : 6) UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

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

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

bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dengan mengaplikasikan beragam sistem. Intinya UML merupakan alat komunikasi yang konsisten dalam mendukung para pengembangan sistem saat ini.

II.5.1. Diagram UML

II.5.1.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang di tekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara actor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang teks case untuk semua feature yang ada pada sistem. Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dalam proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah use

case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antara use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

II.5.1.2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (*atribut/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metoda/fungsi*). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, dan lain-lain. Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

II.5.1.3. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state

sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behavior internal sebuah sistem (dan interaksi antara subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebagai aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana actor menggunakan sistem untuk melakukan aktivitas. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behavior pada kondisi tertentu. Untuk mengilustrasikan proses-proses parallel (fork dan 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.

II.5.1.4. Sequence Diagram

Sequence diagram menggambarkan interaksi antara objek didalam dan disekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri antar dimensi vertical(waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram bisa digunakan untuk menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang meng-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk actor, memiliki lifeline vertical. Message

digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metode dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebagai message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity.