

BAB II

LANDASAN TEORI

2.1 Definisi Sistem

Sistem dapat didefinisikan dalam dua kelompok pendekatan yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Menurut Jerry FitzGerald, Ardra F. FitzGerald, Warren D. Stallings dalam Jogiyanto (2005) pendekatan sistem yang menekankan pada prosedurnya mendefinisikan suatu Sistem itu adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Sedangkan prosedur itu sendiri oleh Richard F. Neuschel (2005) didefinisikan sebagai suatu urutan operasi klerikal (tulis-menulis), biasanya melibatkan beberapa orang didalam satu atau lebih departemen yang ditetapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.

Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponennya menurut Jogiyanto (2005), mendefinisikan Sistem itu adalah kumpulan dari elemen-elemen yang saling berintegrasi untuk mencapai suatu tujuan tertentu.

Suatu sistem dapat terdiri dari beberapa sub-sub sistem, dan sub-sub sistem tersebut dapat pula terdiri dari sub-sub sistem yang lebih kecil lagi. Dari beberapa definisi sistem diatas memiliki maksud dan tujuan yang sama

tetapi berbeda dalam pendekatannya. Para ahli lebih banyak menggunakan definisi sistem dengan menekankan pada komponen-komponen atau elemen-elemen.

2.1.1 Karakteristik Sistem

Menurut Jogiyanto HM (2005 : 3) Suatu sistem memiliki karakteristik atau sifat tertentu, yaitu memiliki komponen-komponen, batasan sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolahan, sasaran atau tujuan. Karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem (*Boundary*)

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

3. Lingkungan Luar Sistem (*Environment*).

Lingkungan luar dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem (*Interface*).

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke sistem lain.

5. Masukkan Sistem (*Input*).

Masukkan adalah energi yang dimasukkan ke dalam sistem atau segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk di proses .

6. Keluaran Sistem (*Output*).

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan bisa juga dikatakan hasil dari suatu pemrosesan.

7. Pengolahan Sistem (*Process*).

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran atau proses juga merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna.

8. Sasaran Sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan (*goal*) dan sasaran (*objective*). Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukkan yang dibutuhkan sistem dalam keluaran yang dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuannya

2.1.2 Klasifikasi Sistem

Menurut Jogiyanto HM (2005 : 6), Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

Sistem Abstrak dan Sistem Fisik (Physical System)

Sistem Abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik dapat berupa gagasan atau konsep. Sistem fisik adalah sistem yang ada secara fisik dan dapat dilihat oleh mata kita.

Sistem Alamiah (Natural System) dan Sistem Buatan Manusia (Human Made System)

Sistem Alamiah adalah sistem yang terjadi melalui proses alam, tanpa ada campur tangan manusia. Sedangkan Sistem buatan manusia adalah sistem yang terjadi melalui proses yang dirancang dan dilakukan oleh manusia.

Sistem Tertentu (Deterministic System) dan Sistem Tak Tentu (Probabilistic System)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

Sistem Tertutup (Closed System) dan Sistem Terbuka (Open System)

Sistem Tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya, secara teoritis sistem tertutup ini ada tetapi kenyataan ini tidak ada sistem yang benar-benar tertutup yang ada hanyalah *Relatively Closed System* (secara relatif tertutup, tidak benar-benar tertutup). Sistem Terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem lainnya.

2.2 Definisi Informasi

Menurut Jogiyanto HM (2005, Hal 8), pengertian dari informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima dan membutuhkannya.

Informasi yang baik harus memenuhi kriteria sebagai berikut :

1. Akurat

Informasi yang diperoleh harus bebas dari kesalahan-kesalahan dan tidak boleh menyesatkan serta harus mencerminkan suatu maksud. Informasi diharuskan akurat karena dari informasi yang tidak akurat akan banyak timbul gangguan yang dapat merubah atau merusak informasi.

2. Tepat Waktu

Informasi yang sampai pada yang membutuhkan tidak boleh terlambat, informasi yang telah usang tidak akan mempunyai nilai, hal ini disebabkan karena informasi merupakan landasan dalam pengambilan keputusan, maka akan berakibat fatal pada suatu organisasi, instansi maupun perusahaan.

3. Relevan

Informasi tersebut harus bermanfaat bagi yang membutuhkannya. Relevansi informasi untuk tiap-tiap orang adalah berbeda. Informasi yang baik hanya akan dihasilkan oleh data yang baik dengan pemrosesan data yang tepat.

2.3 Definisi Sistem Informasi

Sistem informasi menurut Abdul Kadir (2003) adalah *Computer Bases Information* (CBIS) atau yang dalam Bahasa Indonesia disebut juga Sistem Informasi Berbasis Komputer merupakan sistem pengolah data menjadi

sebuah informasi yang berkualitas dan dipergunakan untuk suatu alat bantu pengambilan keputusan. Sistem Informasi yang akurat dan efektif, dalam kenyataannya selalu berhubungan dengan istilah “*Computer-based*” atau pengolahan informasi yang berbasis pada komputer. Sistem Informasi berbasis komputer mengandung arti bahwa komputer memainkan peranan penting dalam sebuah sistem informasi.

Secara teori sebuah Sistem Informasi memang tidak harus menggunakan komputer dalam kegiatannya. Tetapi pada prakteknya tidak mungkin sistem informasi yang sangat kompleks itu dapat berjalan dengan baik tanpa adanya komputer. Sistem Informasi merupakan sistem pembangkit informasi. Dengan integrasi yang dimiliki antar subsistemnya, sistem informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat dan akurat sesuai dengan manajemen yang membutuhkannya.

2.3.1 Pengertian Analisa Sistem Informasi

Menurut Jogiyanto HM (2005 : 129), Analisa Sistem Informasi dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Untuk mendapatkan hasil yang baik dalam merancang suatu sistem, maka diperlukan terlebih dahulu adanya sistem informasi tentang sistem yang

sedang berjalan, meliputi deskripsi prosedur dan analisa dokumen yang digunakan.

Beberapa alasan perlunya menganalisa sistem yang sedang berjalan adalah sebagai berikut :

Sistem yang sedang berjalan akan dijadikan dasar untuk perancangan sistem baru atau yang akan dikembangkan.

Untuk bahan pertimbangan dalam perancangan sistem yang baru.

Sistem yang sedang berjalan bisa dijadikan bahan acuan untuk menentukan karakteristik sistem yang baru.

Dalam analisa sistem informasi terdapat pula langkah-langkah dasar yang harus dilakukan oleh analisa sistem sebagai berikut :

- a) *Identify*, yaitu mengidentifikasi masalah
- b) *Understand*, yaitu memahami kerja dari sistem yang ada
- c) *Analyze*, yaitu menganalisa system
- d) *Report*, yaitu membuat laporan hasil analisis.

2.3.2 Manajemen Data

Manajemen data adalah bidang studi yang luas, tapi pada dasarnya adalah proses pengelolaan data sebagai sumber daya yang berharga bagi suatu organisasi atau bisnis. Manajemen Data merupakan proses pengembangan arsitektur data, praktek dan prosedur yang berhubungan dengan data dan kemudian mengeksekusi aspek-aspek ini secara teratur. Aspek yang penting dalam manajemen data adalah kelengkapan identitas data.

Manajemen Data yang merupakan bagian dari manajemen sumber daya informasi mencakup semua kegiatan yang memastikan bahwa data:

- a. Data Akurat
- b. Up to Date (Mutakhir)
- c. Aman
- d. Tersedia bagi pemakai (user)

2.4 UML

2.4.1 Definisi UML

UML (*Unified Modelling Language*)

Menurut Munawar (2005 : 17) *Unified Modelling language* (UML) adalah suatu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek, UML menyediakan pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain.

Dengan menggunakan UML, dapat membuat model untuk semua jenis aplikasi peranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, dan dapat digunakan juga untuk mendefinisikan notasi dan *syntax*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram-diagram piranti lunak, setiap bentuk mempunyai makna tertentu dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

UML adalah salah satu bentuk notasi atau bahasa yang sama yang digunakan oleh professional dibidang *software* untuk menggambarkan atau memodelkan sebuah sistem *software*. Sebelumnya ada banyak notasi atau bahasa lain untuk mencapai keperluan yang sama misalnya DFD (*Data Flow Diagram*). Tetapi sejak matang dan populemnya teknologi pemrograman, perancangan, dan analisis berorientasi objek, UML telah menjadi *de facto standard language*.

Diagram-diagram yang terdapat dalam UML antara lain: *use case diagram*, *class diagram*, *statechart diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, *deployment diagram*.

2.4.2 Model – Model Diagram dalam UML

2.4.2.1 Use Case Diagram

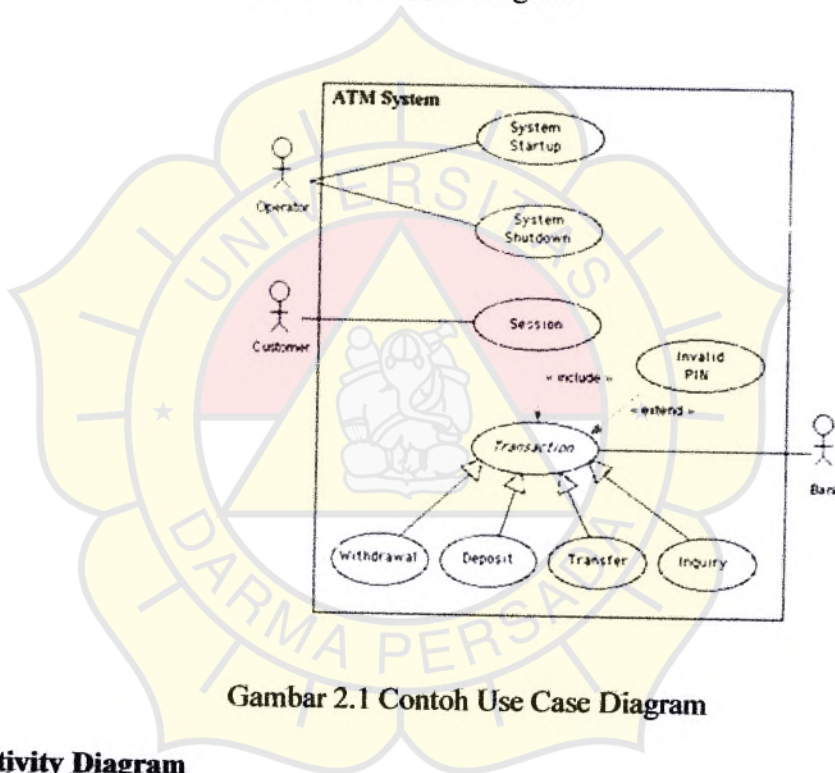
Munawar (2005 : 109) menjelaskan *Use case diagram* merupakan deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara memdeskripsikan tipikal interaksi antar *user* (pegguna) sebuah sistem dengan sistem-nya sendiri bagaimana sistem tersebut digunakan atau “apa” yang diperbuat sistem dan bukan “bagaimana”. *Use case* dapat dipresentasikan sebuah interaksi antara *actor* dengan sistem .

Use case Diagram dapat sangat membantu bila sedang menyusun *requirement* (Jacobsoonet all, 1992). Diagram ini menunjukkan tiga aspek dari sistem yaitu *actor*, *use case*, dan sistem atau subsistem.

Dilihat dari segi pandang Romi Satria Wahono dan Sri Dharyanti sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari 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 antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

Contoh Use Case Diagram



Gambar 2.1 Contoh Use Case Diagram

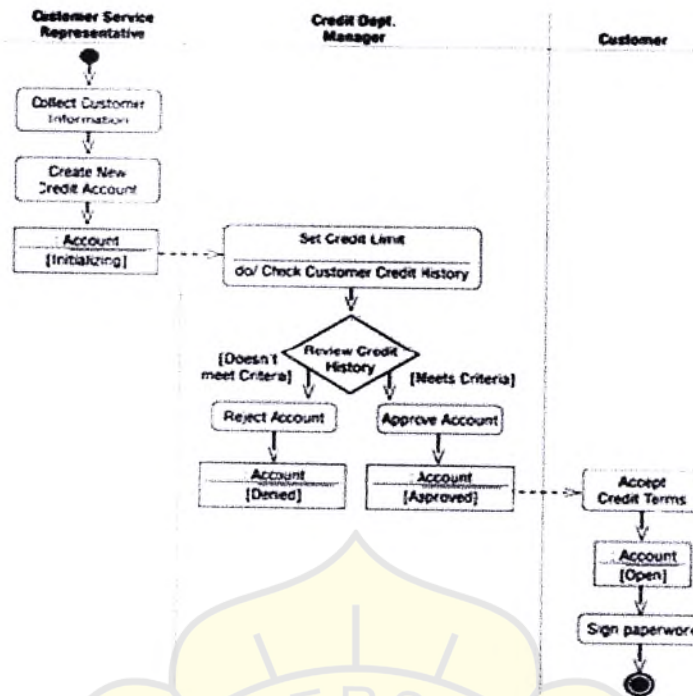
2.4.2.2 Activity Diagram

Romi Satria Wahono *Activity diagrams* merupakan yang 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.

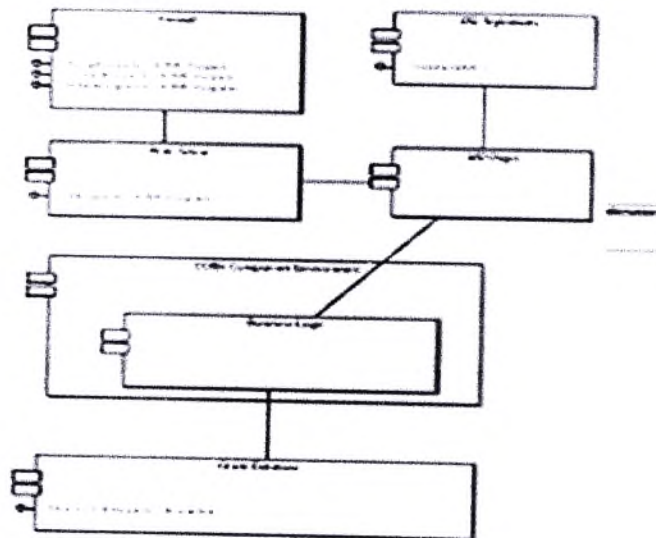
Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Contoh *activity diagram*:



Gambar 2.2 Contoh Activity Diagram

2.4.2.3 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Contoh component diagram:



Gambar 2.4 Contoh Component Diagram

2.5 TEORI PENDUKUNG

2.5.1 Pengertian Penilaian Hasil Belajar Siswa Sekolah

Penilaian hasil belajar siswa sekolah dapat diartikan sebagai proses pengumpulan dan pengolahan informasi untuk menentukan pencapaian hasil belajar peserta didik yang berkaitan dengan cara memenuhi syarat – syarat dari sekolah untuk proses pembelajaran siswa dan untuk mengetahui nilai siswa siswi dari pembelajaran selama sekolah.

2.5.2 Visual Basic

Visual Basic adalah sebuah sarana pembuat program yang lengkap namun mudah. Sejarah Visual Basic dimulai dari pengembangan bahasa basic di Dartmouth College, Amerika Serikat, pada awal tahun 1960-an. Jadi siapa pun yang biasa menggunakan Windows, ia pasti membuat program dengan Visual Basic. Anda hanya perlu tahu cara menggunakan mouse, memanipulasi jendela, serta logika pemrograman untuk membuat sebuah aplikasi Visual Basic (Cipto Mangkusumo, 2000). Microsoft Visual Basic pertama kali membuat program

Visual Basic untuk Dos dan untuk Windows pada tahun 1991. Kemudian membuat program lagi dalam versi 3.0 pada tahun 1993. Setelah membuat versi 3.0, Microsoft membuat program Visual Basic dalam versi 4.0 pada tahun 1995 begitu juga dengan versi 5.0, dan akhirnya Microsoft membuat program Visual Basic 6.0 pada tahun 1998 yang menggunakan teknologi Graphical User Interface (GUI) sehingga lebih memaksimalkan kualitas program yang dihasilkan. Program ini pun mampu menampilkan tampilan-tampilan yang menarik. Sehingga kita tidak bersusah payah untuk mengatur tampilan desain dari program ini. Karena pada program ini menyediakan fasilitas - fasilitas yang dapat digunakan untuk membuat aplikasi dengan tombol sebagai control dari aplikasi yang kita kerjakan serta kotak perintah sehingga program aplikasi dapat dijalankan. Visual Basic dilengkapi dengan Form, ToolBox, Window Properties, Form Layout, Jendela Kode dan lain - lain. Form adalah daerah kerja utama, dimana Anda akan membuat program-program aplikasi Visual Basic. ToolBox adalah perlengkapan didalam pemrograman Visual Basic dan digunakan oleh aplikasi untuk mendapatkan input dari pemakai agar ditampilkan sebagai output. ToolBox terdiri dari menu - menu yang isinya adalah:

- a) Pointer digunakan untuk mengatur ukuran, posisi dan memblok objek.
- b) PictureBox berfungsi untuk menampilkan file - file atau objek gambar.
- c) Label berfungsi untuk memberikan teks atau keterangan.

- d) TextBox digunakan untuk menampilkan text tetapi pemakai dapat mengisinya atau pun melihat text yang ditampilkan sebagai output suatu proses.
- e) Frame merupakan unit untuk mengidentifikasi sebuah group pengontrolan.
- f) CommandButton unit kontrol ini akan memberikan sebuah perintah atau tindakan.
- g) CheckBox pilihan yang digunakan untuk memberikan perintah benar dan salah.
- h) OptionButton unit ini merupakan bagian dari OptionButton yang lain yang akan membentuk group pilihan.
- i) ComboBox dengan unit ini memungkinkan kita untuk dapat mengetikkan pilihan atau bisa juga memilih item lewat Drop-Drown List.
- j) ListBox kontrol ini digunakan untuk menampilkan daftar item yang dapat dipilih oleh pemakai.
- k) Horizontal ScrollBar untuk menggerakkan objek kearah horizontal.
- l) Vertical ScrollBar untuk menggerakkan objek kearah Veritical.
- m) Timer unit ini digunakan untuk pembuatan animasi dan hasilnya akan menjadi berkelip – kelip.
- n) FileListBox kontrol ini berfungsi untuk menampilkan daftar file.
- o) Shape untuk membuat lingkaran, persegi empat, ellips pada form.
- p) Line untuk pembuatan garis.
- q) Image unit ini berfungsi untuk menampilkan objek gambar.

- r) Data digunakan untuk mengakses file – file DBMS.
- s) Ole digunakan untuk memasukkan objek OLE.

Jendela Properties adalah jendela mengandung semua informasi mengenai objek yang terdapat pada aplikasi Visual Basic anda. Form layout Windows adalah jendela yang menggambarkan posisi dari form yang ditampilkan pada layer monitor. Jendela Kode adalah salah satu jendela yang sangat penting di dalam Visual Basic. Sedangkan VisualBasic.net adalah generasi selanjutnya dari visual basic. VisualBasic.Net memungkinkan kita untuk membangun aplikasi database client atau server performa tinggi dan sangat cocok didampingkan dengan perangkat lunak SQL Server 2000. Pemilihan VisualBasic.Net sebagai pengembang sistem ini adalah merupakan salah satu program aplikasi dibawah platform .net Framework. Junindar (2008;3-5).

Di dalam vb terbaru terdapat Windows Control Library : yang membuat kontrol sendiri dan memasukan berbagai fungsi yang anda inginkan di dalam kontrol tersebut. Fasilitas untuk membuat kontrol tersebut adalah Windows Control Library. Kontrol ini sama dengan ActiveX Control (.ocx) dalam pemrograman VB6.

2.5.3 Sejarah SQL Server

SQL Server 2005 atau disebut juga dengan SQL Server 9 merupakan pengembangan dari SQL Server versi 2000. Pada SQL Server 2005 disediakan beberapa tools yang dapat digunakan oleh para developer, yaitu:

- a. SQL Server Management Studio
- b. SQL Computer Manager

c. Sqlcmd (SQL Command)

d. SQL Management Object

SQL Server 2005 merupakan aplikasi database produk Microsoft yang memiliki fitur-fitur baru sehingga membuatnya menjadi platform database yang sempurna. SQL Server adalah RDMS (Relational Database Management System) yang dikembangkan oleh Microsoft. SQL Server terdiri dari beberapa versi antara lain yaitu:

1. SQL Server 2005 Enterprise Edition : Merupakan edisi yang terlengkap dari SQL Server 2005 dan hanya dapat diinstall pada system operasi jenis server, seperti Windows 2003 Server atau Windows 2008 Server.

Dalam SQL Server yaitu Database merupakan tempat penyimpanan data, yang memudahkan pengolahan dan pencarian data. SQL Server 2008 merupakan DBMS (Database Management System) yang powerfull untuk mengolah data. Di samping memiliki user interface yang mudah digunakan,. SQL Server 2008 juga memiliki fitur tambahan yang bisa diimplementasikan untuk meningkatkan performa dan mengurangi tempat penyimpanan. SQL Server 2008 mempunyai beberapa fitur baru yang ditambahkan dari Versi sebelumnya, Diantaranya :

1. User Instance

Di mana memberikan kemampuan untuk memperlakukan database seperti file biasa. Dengan ini, database lokal bisa dipindahkan, disalin, atau dikirimkan melalui email berikut aplikasi yang menggunakannya. Di lokasi yang baru, aplikasi yang menggunakan database tersebut bisa dijalankan tanpa perlu melakukan konfigurasi apapun.

2. SQL Service Broker

Di mana komunikasi antar aplikasi terjadi melalui kontrak pertukaran pesan secara peer-to-peer yang disebut sebagai dialog.

3. FOR XML

Fitur ini dapat digunakan untuk menggabungkan beberapa baris (*row*) di sebuah tabel menjadi satu baris (*row*) tunggal. Soetam (2008 : 2)

