

BAB II

LANDASAN TEORI

Dalam merancang dan mengimplementasikan sistem ini serta pembuatan tugas akhir ini di gunakan beberapa landasan teori, yaitu :

2.1 Microsoft Visual Basic 6.0

Pada tahun 1980-an sistem operasi DOS cukup populer dikalangan pemakai PC karena di dalamnya disertakan bahasa BASIC yang dikenal dengan QBASIC (*QuickBasic*). Diera *windows*, Microsoft menciptakan Visual Basic yang merupakan pengembangan dari bahasa pemrograman BASIC (*Beginners All-purpose Symbolic Instruction Code*). Kemudian berkembang versi – versinya sampai pada versi Microsoft Visual Basic 6.0. Visual Basic adalah bahasa pemrograman berbasis windows, yang sangat interaktif dan lebih *compatible* dengan sistem operasi windows yang umum digunakan untuk menghasilkan aplikasi yang bekerja pada sistem operasi Windows 95 atau Windows NT 4. Banyak keunggulan dimilikinya, yang paling menonjol adalah kemudahan pemakaian. Microsoft Visual Basic 6.0 menyediakan fasilitas yang memungkinkan untuk menyusun sebuah program dengan memasang objek-objek grafis dalam sebuah *form* dengan penggunaan aplikasi GUI (*Graphical User Interface*).

Pada bidang *database* Microsoft Visual Basic 6.0 memberikan kemudahan seperti Oracle, Microsoft SQL Server, AS400, MySQL, DB2 dan lain sebagainya. Dengan adanya ODBC (*Open Database Connectivity*), DAO (*Data Access Object*), ADO (*ActiveX Data Object*) maupun RDO (*Remote Data Object*)

Microsoft Visual Basic 6.0 memberikan lebih banyak kemudahan. Microsoft Visual Basic 6.0 menyediakan tiga macam *interface* yang bisa digunakan untuk merancang aplikasi sesuai dengan kebutuhan. Selain MDI (*Multi Document Interface*) dan SDI (*Single Document Interface*), kini tersedia *Explorer Document Interface* dengan gaya seperti Windows Explorer.

2.2 COM: Common Object Model

Dalam Visual Basic, secara prinsip, obyek COM yang dipakai adalah ActiveX Component. ActiveX merupakan salah satu jenis obyek COM yang berdasarkan teknologi OLE (*Object Linked and Embedded*). Yang paling penting membedakan antara ActiveX dan COM, semua ActiveX adalah obyek COM, tetapi tidak semua obyek COM adalah ActiveX.

COM merupakan standar yang dihasilkan dari pendekatan konsep OOP. Enkapsulasi dan penggunaan ulang *code* adalah keuntungan penggunaan COM. Selain itu masih terdapat beberapa keuntungan yang lain, dan merupakan ciri sekaligus kekuatan COM, yaitu:

1. Interoperabilitas. Karena COM merupakan suatu standar biner, maka komponen yang berdasarkan standar COM bersifat *language independent*. Artinya, suatu komponen *software* bisa ditulis dari bahasa pemrograman apa pun, asal bahasa tersebut mengikuti konsep dasar COM.
2. Sistem penerapan versi. Saat membuat suatu versi baru dari komponen COM yang sudah ada, interface dapat ditambahkan, namun tidak dapat menghilangkan atau memodifikasi *interface* yang

sudah ada. Aturan ini sangat penting berdasarkan aturan dasar COM tentang prinsip penggunaan ulang dalam banyak aplikasi. Untuk memastikan aplikasi yang sedang menggunakan suatu komponen tidak rusak saat komponen yang lebih baru di-*install*, maka komponen yang baru tersebut harus mendukung semua fungsionalitas dari versi yang lalu (*backward compatibility*).

3. Standarisasi. Salah satu cara yang dapat membuat pengguna merasa lebih senang dengan aplikasi baru adalah dengan mengembangkan *software* yang familiar bagi pengguna. Dengan menggunakan ulang komponen standar *user interface* seperti ActiveX Control, suatu program dapat dibuat kelihatan sama dan beroperasi mirip dengan Windows (standar).

2.3 ActiveX

Visual Basic dapat digunakan untuk membuat 3 jenis dasar komponen ActiveX, yaitu:

1. ActiveX Control yang merupakan komponen *software* yang dibuat dari elemen visual, yang disebut *User Control*. ActiveX Control berjalan di kapling memori aplikasi klien (*In-Process*).
2. ActiveX Code Component yang merupakan library atau pustaka yang terdiri dari kelas yang dapat digunakan oleh aplikasi klien untuk membuat obyek. Terdapat 2 macam ActiveX Code Component, yaitu (1) DLL, *In-Process*, dan (2) EXE, *Out-of-Process*. ActiveX Code Component lebih cocok untuk mengerjakan pekerjaan yang ada di

balik layar. Suatu komponen dapat digunakan untuk mengerjakan suatu pekerjaan sementara program masih tetap responsif.

3. ActiveX Document yang merupakan komponen yang dikompilasi sedemikian rupa sehingga bisa digunakan melalui OLE *Container*, seperti Microsoft Word atau Internet Explorer.

