BAB II
TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem

Terdapat dua kelompok dalam mendefinidikan sistem, yaitu menekankan pada prosedurnya dan yang menekankan pada komponen atau elemen.

Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai "*suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang sama*".

Sedangkan pendekatan Sistem yang menekankan pada komponen atau elemen-elemen mendefinisikan bahwa "*Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu*" .JOG[4]

2.1.1 Karakteristik Sistem

1. Komponen Sistem

   Suatu Sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya berkerjasama untuk membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

   Batasan Sistem merupakan daerah yang membatasi diantara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan Sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang menunjukan ruang lingkup sistem itu.

3. Lingkungan Luar Sistem

   Lingkungan Luar Sistem dari sistem adalah diluar batas dari sistem yang mempengaruhi oprasional sistem. Lingkungan Luar Sistem dapat bersifat menguntungkan dan juga dapat bersifat merugikan. Linkungan luar sistem tersebut yang menguntunkan merupakan energi dari Sistem dan dengan demikian harus dijaga serta dipelihara sedangkan lingkungan luar sistem yang merugikan harus ditahan dan dikendalikan, apabila tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

   Penghubung Sistem merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem kesubsistem yang lainnya dengan. Dengan adanya penghubung satu subsistem dapat berinteraksi dengan subsistem yang lainnya dan membentuk suatu satu kesatuan.

5. Masukan Sistem

   Masukan Sistem adalah suatu masukan yang dimasukan kedalam sistem. Masukan dapat berupa masukan perawatan, masukan signal. Masukan perawatan adalah suatu yang dimasukan supaya sistem tersebut dapat beroprasi sedangkan masukan signal adalah suatu masukan yang diproses untuk didapatkannya suatu keluaran.

6. Keluaran Sistem

   Keluaran sistem adalah hasil dari masukan yang diolah dan diklasifikasikan menjadi yang berguna. Keluaran dapat berupa masukan bagi sub sistem lainnya.

7. Pengolahan Sistem

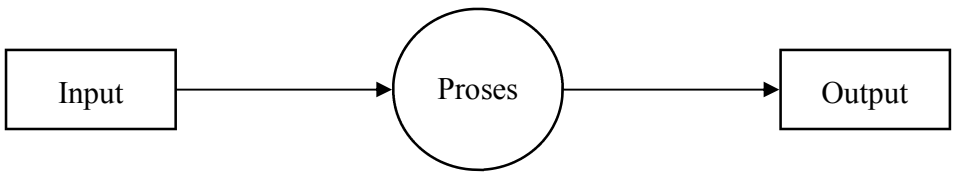
   Sesuatu dapat mempunyai suatu bagian pengolahan atau Sistem itu sendiri sebagai pengolahannya. Pengolahan yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

   Suatu sistem pasti memiliki tujuan/sasaran. Apabila suatu sistem tidak mempunyai sasaran maka oprasi sistem tidak ada gunanya. Sasaran sistem dari sistem dapat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan oleh sistem. Suatu sistem dikatakan berhasil apabila sasaran yang diharapkan memenuhi tujuannya.

2.1.2 Bentuk Uumum Sistem

Pada dasarnya bentuk sistem terdiri atas masukan (*Input*), proses dan keluaran (*Output*), adapun bentuk umum sistem dari suatu sistem adalah sebagai berikut :



Gambar 2.1 Bentuk Umum Sistem
[Sumber : JOG[3]]

2.2 Konsep Dasar Informasi

Informasi adalah faktor terpenting dalam sistem untuk pengembalian suatu keputusan. definisi informasi sendiri dapat didefinisikan sebagai berikut: ”*Informasi adalah data yang diolah menjadi bentuk yang lebih berguna bagi yang menerimanya*” JOG[3]

Sumber dari informasi adalah data, yang kemudian diolah dengan kriteria tertentu untuk menghasilkan informasi yang dibutuhkan. Data adalah suatu kenyataan yang menggambarkan suatu kejadian-kejadian dan menjadi kesatuan yang nyata. Kejadian adalah sesuatu yang terjadi pada saat tertentu yang menyangkut perubahan nilai yang disebut transaksi. Sedangkan kejadian nyata adalah merupakan suatu objek yang nyata dan terjadi pada saat kejadian berlangsung. Data juga dapat diartikan sebagai segala sesuatu yang perlu diolah terlebih dahulu untuk mendapatkan suatu informasi.

Kualitas informasi yang diharapkan tergantung pada tiga hal yaitu :

1. Akurat

   Berarti informasi harus bebas dari kesalahan-kesalahan, tidak biasa,tidak menyesatkan dan mencerminkan maksud dan tujuannya.

2. Tepat Waktu

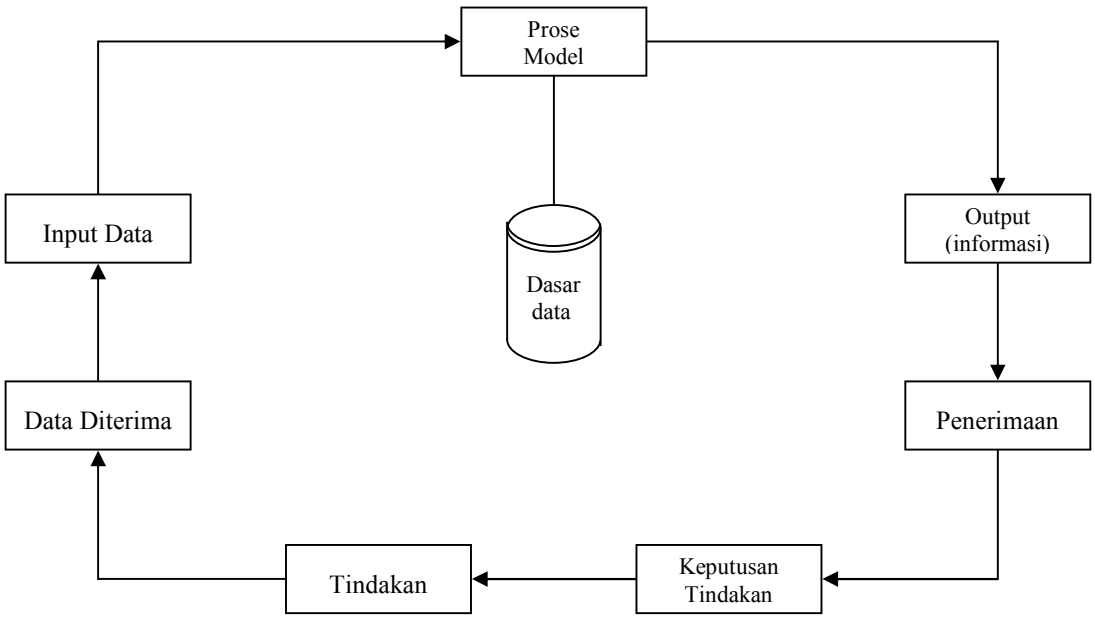
   Berarti informasi yang sampai kepenerima tidak terlambat, karena informasi adalah landasan untuk mengambil keputusan. Untuk itu diperlukan suatu teknologi yang mengolah dan mengirim informasi dengan cepat dan tepat.

3. Relevan

   Berarti informasi mempunyai manfaat dan berguna bagi pemakainya, karena batas relevan seseorang berbeda, maka informasi bisa dikatakan berguna jika benar-benar berguna dan dibutuhkan pemakainya.

2.2.1 Siklus Informasi

Informasi merupakan suatu proses perubahan dunia menjadi informasi, data yang diolah suatu model menjadi informasi bagi penerima, kemudian penerima menginformasikan informasi tersebut dengan membuat suatu keputusan dan melakukan tindakan yang berarti menghasilkan keputusan dan melakukan tindakan yang lain yang akan membuat sejumlah data kembali, untuk lebih jelasnya dapat dilihat pada Gambar 2.2 Siklus Informasi dibawah ini



Gambar 2.2 Siklus Informasi
[Dumber : JOG[4]]

2.3 Konsep Dasar Sistem Informasi

Sistem informasi adalah "*Suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung operasi serta bersifat managerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan*". JOG[4]

Informasi didefinisikan oleh Robert A.Leitch dan K. Rosecoe Davis sebagai berikut :

"*Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi yang mendukung oprasi bersifat managerial dan kegiatan strategi-strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan".*

2.4 Metode Pengembangan Sistem

Pada metode pendekatan sistem ini penulis menggunakan metode *Object Oriented* sebab metode ini tidak memperhatikan sistem sebelumnya, sehingga dengan menggunakan metode ini penulis dapat lebih memfokuskan kepada sistem informasi yang akan dibangun, selain itu dengan metode ini pada konsep objek dalam pemograman dan penggunaan struktur datanya. Adapun alat Bantu yang digunakan adalah UML (*Unifid Modeling Language*) dengan menggunakan *Software* Rational Rose untuk membantu dalam melakukan analisis dan desain *Object Oriented.*

Analisis Object Oriented aktifitas pertama yang dilakukan sebagai bagian dari rekayasa perangkat lunak. Didasarkan pada serangkaian prinsip dasar. Ada lima prinsip dasar untuk membangun model anali, yaitu:

1. Domain Informasi dimodelkan;
2. Fungsi model digambarkan;
3. Tingkah laku model direpesentasikan;
4. Model dipartisi untuk digambarkan, untuk mengekspos detail yang lebih besar;
5. Model awal mepresentasikan inti masalah sementara model selanjutnya memberikan detail implementasi.

Prinsip tersebut menjadi pondasi bagi penulis untuk mengembangkan Sistem yang akan bangun.

2.5 Pengenalan UML

UML merupakan gabungan metode *Booch. Rumbaugh* (OMT) dan *Jacobson.* Tetapi UML ini akan mencakup lebih dari pada OOA&D. pada pertengahan pengembangan UML dilakukan Standarisasi proses dengan OMG (Object Management Group) dengan harapan UML akan menjadikan standar pemodelan pada masa yang akan dating [PRM[6]]

UML disebut *sebagai bahasa pemodelan, kebayakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan*

(sebagian besar grafik) merupakan narasi dari metode yang digunakan untuk mendisain model secara cepat .[PRM[6]]

Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka anda hayah akan membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desasin.

UML merupakan bahasa standar untuk penulisan *blueprint software* yang digunakan untuk visualisasi, spesipikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

UML dimulai secara resmi pada oktober 1994, ketika Rumbaugh bergabung dengan Booch pada Relational Soefware Corporation. Proses ini memfokuskan pada penyatuan metode Booch dan OMT. UML Versi 0.8 merupakan metode penyatuan yang dirilis pada bulan oktober 1995. dalam waktu yang sama, Jacobson bergabung dengan Reational dan cakupan UML semakin luas sampai si luar perusahaan OOSE. Dokumentasi UML versi 0.9 akhirnya dirilis pada bulan juni 1996. meskipun pada tahun 1996 melihat dan menerima *feedback* dari komunikasi *Software Engenering*. Dalam waktu tersebut, menjadi lebih jelas bahwa beberapa organisasi perangkat lunak melihat UML sebagai strategi dari bisnisnya. Kemudian dibangunlah UML *Consrtium* dengan beberapa organisasi yang akan meyumbangkan sumberdayanya untuk bekerja, mengembangkan, dan melengkapi UML.

2.5.1 Pengertian UML

UML adalah bahasa untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifacts* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak lainnya , selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi *Object.* UML dibuat Oleh *Grade Booch, James Rumbaungh dan iIvan Jacobson* dibawah *Rational Software Corp*. UML menyediakan notasi-notasi yang membangun memodelkan perangkat lunak, namun hamper dalam semua bidang yang membutuhkan pemodelan.

UML (*Unified Modeling Language*) dapat menggambarkan yang akan kita gambarkan dengan secara elektronik (dengan menggunakan Komputer) lewat sarana perangkat lunak Rationsl Rose. Dengan pemodelan menggunakan UML ini, pengembang dapat melakukan tinjauan umum bagi arsitektur Sistem secara Keseluruhan, penelaahan bagaimana objek-objek dalam sistem saling mengirimkan pesan, menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang diharapkan/seharusnya, dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu dimasa yang akan dating [PRM[6]]

2.5.2 Bagian-bagian UML

Bagian-bagian utama dari UML adalah *View,Diagram, Model element dan general mechanism.*

1. *View*

*View*digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grapik, tapi merupakan suatu abstraksi yang berisi sejunlah diagram.

Beberapah jenis *View* dalam UML antara lain adalah *use case view, logical view, component view, concurrency view, dan deploypment view.*

a. *Use Case View*

Mendeskrifsikan fungsionalitas sistem yang seharunya dilakukan sesuai yang diinginkan oleh *external actors. Actor* yang beriteraksi dengan sistem dapat berupa user atau sistem lainnya. *View* ini digambarkan dalam *Use case diagrams* dan kadang-kadang dengan *activity diagram.*

View ini digunakan terutama untuk pelanggan, perancang (*designer*) pengembang (*developer*), dan penguji sistem (*tester*).

b. *Logical View*

Mendeskrifsikan bagaimana fungsionalitas dari sistem, struktur stati (*class, object* dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu.

View ini digabarkan dalam *class diagrams* untuk struktur statis dan dalam *state. Sequence, collaboration,* dan *activity diagrams* untuk model

dinamisnya. View ini digunakan untuk perancangan (*designer*) dan pengembang (*developer*).

c. *Component View*

Mendeskrifsikan implementasi dan ketergantungan modu komponen yang merupakan tipe lainnya dari *code modul* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan inpormasi administrasi lainnya.

View ini digambarkan dalam *component view* dan digunakan untuk pengembang (*develover*).

d. *Concurrency View*

Mengambil sistem kedalam proses dan prosesor. View ini digambarkan dalam diagram dinamis (*state, sequence, collaboration* dan *activity diagram*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembangan(*devwloper*), pengintegrasi (*integrator*) dan penguji (*tester*)

e. *Deploypment View*

Mendeskripsikan fisik dari sistem seperti Komputer dan perangkat (*nodes*) dan bagaimana hubungannya denan yang lain. View ini digambarkan dalam *deployment diagrams* dan digunakan untuk pengembangan (*developer*) pengintregrasian (*Intergrator*) dan penguji (*tester*).

2. Diagram

Diagram berbentuk grafik yang menunjukan symbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *View* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain:

a. *Use Case Diagram*

Menggambarkan sejumlah *external Actor* dan hubungannya ke *use case* yang diberikaj\n oleh sistem. *Use case* adalah deskrifsi fungsi yang disediakan oleh sistem kedalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagram.*

Use case digambarkan haya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

b. *Class Diagram*

Menggambarkan struktur satatis *class* di dalam sistem. *Class mempresentasikan sesuatu yang ditangani oleh sistem. Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialed* (satu *Class* merupakan spesialisasi dari *class* lainnya) atau *package* (grup bersama sebagai satu unit).

c. *State Diagram*

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *Class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan.

State Class tidak digambarkan untuk semua *class,* hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

d. *Sequence Diagram*

Menggambarkan kolaborasi dinamis antara sejumlah *object* . kegunaannya untuk menunjukan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object,* sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama pesan diwakili oleh garis dengan tanda padah dan waktu yang ditunjukan dengan proses vertical.

e. *Collaboration Diagram*

Menggambarkan kolaborasi dinamis seperti *sequence Diagrams.* Dalam menunjukan pertukaran pesan. *Collaboration diagrams* menggambarkan oject dan hubungannya (mengacu kekonteks). Jika penekanannya pada waktu atau urutan maka gunakan *sequence Diagrams,* tapai jika pendekatannya pada konteks gunakan *collaboration diagrams.*

f. *Aktivity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskrifsikan aktifitas yang dibentuk dalam suatu oprasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi

g. *Componentt Diagram*

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *ssource code.* Komponen biner atau *executeable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component View.*

h. *Deployment Diagram*

Menggambarkan arsitektur fisik dari perangkat keras danperangkat lunak sistem. Menunjukan hubu hubungan dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Didalam *nodes, executeable component dan object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *nodes* tertentu dan ketergantungan komponen.

2.5.3 *Indikator MultiPlicity*

MultiPlicity pada kasus asosiasi menunjukan bahwa ada sejumlah objek pda sebuah *class* yang berhubungan dengan sebuah objek pada sebuah asosiasi *Class.* Untuk menggambarkan hal tersebut bias dilihat pada fakta bahwa sebuah

tim mungkin terdiri dari beberapa pemain. Untuk menunjukan hal itu angka harus diletakan didekat *Class* yang dimaksud

Ada bayak *MultiPlicity* yang memungkinkan untuk dipakai. Sebuah *class* dapat berhubungan dengan yang lainnya dalam satu kesatuan (*one-to-one*) satu kebayak (*one-to-many*), satu kesatu atau lebih (*one-to-one or more*), satu ke nol atau satu ke sebuah interval tertentu (contoh satu ke lima sampai dengan sepuluh), satu kesejumlah n atau satu keserangkaian pilihan (contoh satu ke sembilan atau sepuluh).

Pada notasi UML, untuk menampilkan lebih atau bayak digunakan (*) untuk menunjukan 'atau' digunakan titik dua (..) seperti contohnya 1..* (berate satu atau lebih). Notasi 'atau' bias juga digunakan tanda koma (,).

2.6 Basis Data

"*Basis data adalah kumpulan atau koleksi dari data-data yang disimpan pada alat-alat penyimpanan tertentu dengan struktur penyimpanan yang khas dan digunakan perangkat lunak untuk menampilkannya*" FAT[3]

Basis data merupakan kumpulan *file-file* yang memiliki hubungan antara satu *file* dengan *file* yanglainya sehingga membentuk satu kesatuan data yang memberikan informasi tentang satu organisasi. Basis data dapat dijalankan dengan menggunakan konpigurasi, salah satu yang terkenal adalah metode *client/server.*

Prinsip utama suatu basis data adalah pengaturan data atau arsip, sedangkan tujuan utama basis data adalah kemudahan dan kecepatan dalam pencarian data atau arsip yang kita inginkan.

Penyusunan suatu basis data digunakan untuk mengatasi masalah pada penyusunan data yaitu :

a. Redudansi dan inkonsitensi data

b. Kesulitan pengaksesan data.

c. Isolasi data untuk standarisasi.

d. *Multiple user* (bayak pemakai).

e. masalah keamanan (*security*)

f. Masalah integrasi (kesatuan).

g. Masalah data *independence* (kebebasan data)

Proses dasar yang dimiliki oleh database ada 4, yaitu :

1. Pembuatan data-data baru (*create database*)
2. Penambahan data (*inseret*)
3. Mengubah data (*up date*)
4. Menghapus data (*delete*)

Sistem manajemen data base (*Database Management system*) merupakan sistem pengoprasian dan sejumlah data pada komputer. Dengan sistem ini dapat merubah data, memperbaiki data yang salah dan menghapus data yang tidak dapat dipakai.

2.7 Pengertian Jaringan Komputer

Jaringan komputer adalah "*merupakan sekelompok komputer otonom yang saling berhubungan antara satu dengan yang lainnya menggunakan protocol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, program-program penggunaan bersama perangkat keras seperti printer, hardisk dan sebagainya."* AND[2].

Selain itu jaringan Komputer bisa diartikan sebagai kumpulan sejumlah terminal computer yang berada diberbagi lokasi yang terdiri dari lebih satu Komputer yang salaing berhubungan.

Tujuan dibangunnya suatu jaringan Komputer adalah membawa informasi dari Komputer pengirim (*transmiter*) menuju Komputer penerima (*reciver*) melaui media komunikasi.

2.8 Pengertian *Client/Server*

Client server merupakan Komputer yang memangfaatkan sumber daya dalam jaringan yang disediakan oleh Komputer lainnya, yang disebut dengan server. Client juga merupakan sebuah aplikasi atau proses yang meminta pelayanan dari Komputer atau proses lainnya. Sever merupakan perangkat khusus dalam jaringan Komputer yang menjadi tempat bagi semua nodes didalam jaringan yang memisahkan secara jelas mana yang dapat memberikan layanan jaringan (*server*) dan yang mana yang hanya menerima layanan (*client*). MIL[3]

2.9 Perangkat Lunak Pendukung

Suatu perangkat lunak (*Software*) pendukung sangat berperan penting dalam membangun suatu aplikasi yang baik, karena berguna untuk melengkapi kebutuhan aplikasi yang akan dibangun. Adapun perangkat lunak yang digunakan yaitu : Borland Delphi 6.0 dan SQL Server 2000.

2.9.1 Sekilas Borland Delphi 6.0

"Delphi adalah aplikasi yang digunakan untuk pengembangan dengan memangfaatkan keistimewaan konsep-konsep antar muka grafis dalam Microsoft windows."

Delphi adalah software buatan bholand yang sangat terkenal. Berbeda dengan software windows umumnya, Delphi bukan lah software aplikasi seperti MS Office atau Software lainnya. Dlphi adalah suatu bahasa pemograman. *Developher Language,* aplikasi untuk membuat aplikasi. Delphi digunakan untuk membangun aplikasi windows, aplikasi grafis, aplikasi visual, bahkan aplikasi jaringan (*Client/Derver*) dan lain sebagainya.

Delphi merupakan program yang bersifat terstruktur danprosedural, Delphi hamper sama dengan pascal sangat baik bagi pemograman. Jadi Delphi sangat cocok dan tepat dalam membangun aplikasi ini

2.9.2 SQL Server 2000

Micdosoft SQL *server* 2000 adalah perangkat *lunak perangkat lunak Realition Management Sistem (RDBMS) yang handal.* Untuk mendukung proses taransaksi yang besar seperti akutansi atau manufaktur.

[Sumber : MIL[3]]

Sistem kan dipergunakan oleh beberapah computer (*client/server*) maka data base yang digunakan adalah merupakan data base yang berfungsi untuk menunjang hal tersebut untuk itu dalam penulisan ini dipakai database Microdoft SQL Server 200, selain itu database ini juga *compatible* dengan bahasa pemograman yang digunakan

2.10 Kasus Yang Dianalisis/Dirancang

Kasus yang dianalisis tau dirancang penulis adalah menyangkut proses pelayanan kesehatan dibagaian Administrasi, karena proses pencatatan masih dilakukan secara manual dimana bayak kesalahan-kesalahan yang tidak disengaja. Untuk itu penulis mengambil kasus Pelayanan kesehatan Administrasi pada Puskesmas Lembang agar masalah yang menyangkut Administrasi dapat diatasi.