

BAB II LANDASAN TEORI

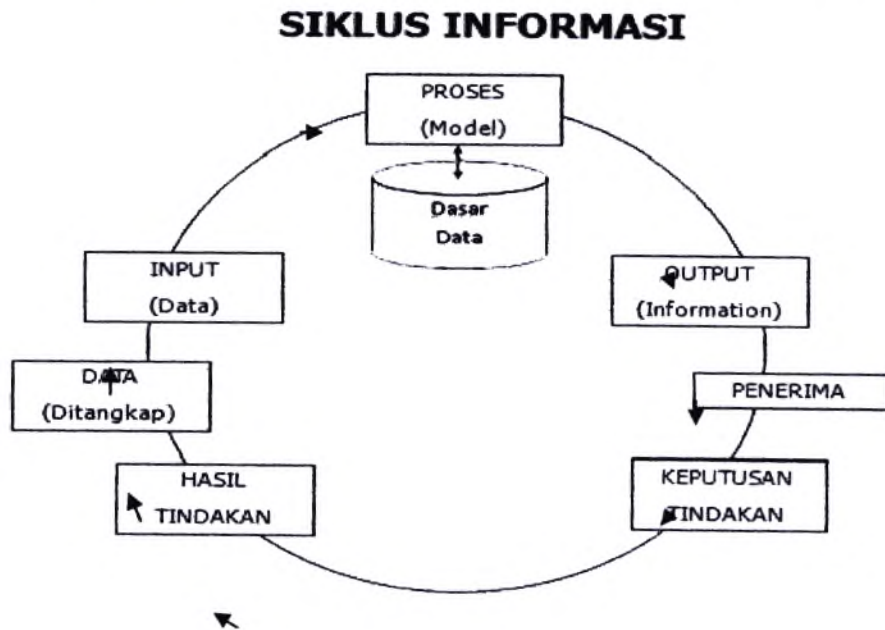
2.1 Definisi Sistem

Sistem menurut Kadir (2003:54) adalah “sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan.” Dalam mendefinisi sistem ada dua pendekatan atau teori menurut Jogiyanto (2005:1) dalam bukunya yang berjudul Analisa dan Desain yaitu sistem yang menekankan pada prosedurnya dan sistem yang menekankan pada elemen atau komponennya. Pengertian sistem pada pendekatan yang lebih menekankan pada prosedur menurut Jerry FitzGerald dalam Jogiyanto (2005:1) mengemukakan bahwa “Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem.

Pengertian sistem pada pendekatan yang lebih menekankan pada elemen atau komponennya menurut Jogiyanto (2005:2) mengemukakan bahwa “Sistem adalah kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”. Kedua kelompok definisi ini adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatannya.

“Suatu sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berintegrasi untuk mencapai suatu tujuan. Sebagai misal, sistem komputer dapat terdiri

tindakan yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data yang diterima lebih lanjutnya menjadi input dan diproses kembali lewat suatu model dan secara berkesinambungan membentuk suatu siklus yang biasa disebut siklus informasi. Dengan aktivitas siklus informasi ini maka setiap informasi dapat menjadi sangat berkembang atau dikembangkan. Siklus informasi tersebut dapat dilihat seperti gambar berikut:



Gambar 2.1 Siklus Informasi

2.3 Definisi Sistem Informasi

Sistem dalam suatu institusi pemerintahan sangatlah penting, karena sistem sangatlah menunjang terhadap kinerja perusahaan atau instansi pemerintah, baik yang berskala kecil maupun besar. Suatu sistem dapat berjalan dengan baik diperlukan kerjasama diantara unsur-unsur yang terkait dalam sistem tersebut.

Terdapat berbagai pendapat yang mendefinisikan definisi sistem, seperti dibawah ini: Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”(*Jogiyanto, 2005:1*).

Penjelasan di atas menjelaskan bahwa sistem bekerja dalam suatu jaringan kerja dari suatu prosedur yang saling berhubungan satu sama lain untuk menyelesaikan tujuan dan sasaran yang dimaksud. Definisi sistem juga dapat dijelaskan oleh Jogiyanto dalam bukunya *Analisa dan Desain sistem informasi*, menerangkan: "Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu" (*Jogiyanto, 2005:2*).

Secara umum sistem informasi di definisikan sebagai berikut:

"Sekumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan. (*Sutana, 2003:4*).

Sistem juga diartikan sebagai sekumpulan elemen yang bekerja sama dalam suatu kesatuan untuk melaksanakan suatu fungsi yang berguna. Dalam bukunya Jogiyanto sistem dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen.

Sistem informasi menurut Robert A. Leitch dan K. Roscoe Davis adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (*Jogiyanto 1999:11*)

2.4 Pengertian Analisa Sistem Informasi

Menurut Jogiyanto HM (2001:129), 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 :

1. Sistem yang sedang berjalan akan dijadikan dasar untuk perancangan sistem baru atau yang akan dikembangkan.
2. Untuk bahan pertimbangan dalam perancangan sistem baru.
3. Sistem yang sedang berjalan bisa dijadikan bahan acuan untuk menentukan karakteristik sistem yang baru.

Analisa sistem merupakan dasar pelaksanaan perancangan sistem, kesalahan yang dilakukan pada tahap analisis sistem merupakan awal kehancuran dari seluruh sistem yang dibangun.

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

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

2.5 Pengertian Koperasi

Koperasi adalah Asosiasi orang-orang yang bergabung dan melakukan usaha bersama atas dasar prinsip-prinsip koperasi, sehingga mendapatkan manfaat yang lebih besar dengan biaya rendah melalui perusahaan yang dimiliki dan diawasi secara demokratis oleh anggotanya. Asosiasi berbeda dengan kelompok, asosiasi terdiri dari orang-orang yang memiliki kepentingan yang sama, lazimnya yang menonjol adalah kepentingan ekonomi. Sedangkan tujuan koperasi adalah untuk menjadikan kondisi sosial dan ekonomi anggotanya lebih baik dibanding sebelum bergabung dengan koperasi. (Arifin Sitio, *Halomoan Tamb*, 2001:15)

Banyak penelitian koperasi yang disampaikan para pakar dibidang koperasi. Ada beberapa pengertian koperasi di bawah ini:

1. Dr. Muhammad Hatta

Dalam bukunya "*The Movement in Indonesia*" beliau mengemukakan bahwa koperasi adalah usaha bersama untuk memperbaiki nasib penghidupan ekonomi berdasarkan tolong menolong. Mereka didorong oleh keinginan memberi jasa pada kawan "seorang buat semua dan semua buat seorang" inilah yang dinamakan Auto Aktivitas Golongan (Arifin Sitio, *Halomoan Tamb*, 2001:17).

2. International Labour Office (ILO)

Menurut ILO definisi koperasi adalah sebagai berikut :

"Kerjasama adalah asosiasi orang, biasanya sarana yang terbatas, yang secara sukarela bergabung bersama untuk mencapai ekonomi dan melalui pembentukan organisasi bisnis demokratis dikendalikan, membuat kontribusi yang adil terhadap modal yang dibutuhkan dan menerima bagian yang adil dari risiko dan manfaat dari usaha tersebut." (Arifin Sitio, *Halomoan Tamb*, 2001:16).

3. UU No. 25 Tahun 1992 (Perkoperasian Indonesia)

Koperasi adalah Badan usaha yang beranggotakan orang seorang atau badan hukum koperasi dengan melandaskan kegiatannya berdasarkan prinsip koperasi sekaligus sebagai gerakan ekonomi rakyat yang berdasarkan atas dasar asas kekeluargaan (*Arifin Sitio, Halomoan Tamb, 2001:18*).

(UU No. 25 tahun 1992 tentang perkoperasian indonesia)

1. Keanggotaanya sukarela dan terbuka. Yang keanggotaanya bersifat sukarela, dan bersedia menerima tanggung jawab keanggotaan.
2. Pengawasan oleh anggota secara Demokratis. Anggota yang secara aktif menetapkan kebijakan dan membuat keputusan. Dalam koperasi primer, anggota memiliki hak suara yang sama (satu anggota satu suara).
3. Partisipasi anggota dalam kegiatan ekonomi. Anggota menyetorkan modal mereka secara adil dan melakukan pengawasan secara demokratis. Sebagian dari modal tersebut adalah milik bersama.

2.6 Peralatan Pendukung (*Tools*)

2.6.1 Pengenalan UML (*Unified Modeling Language*)

UML singkatan dari Unified modeling Language yang berarti bahasa permodelan standar. (Chonoles 2003 : bab 1) mengatakan sebagai bahasa, berarti UML memiliki syntax dan semantik. Ketika kita menggunakan konsep UML ada aturan-aturan yang yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu pada system, bagaimana transaksinya? Bagaimana

sistem mengatasi error yang terjadi? Bagaimana keamanan system yang akan kita buat? Dan sebagainya dapat dijawab dengan UML.

Model-Model Diagram Dalam UML

Beberapa literature menyebutkan bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram perwaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, Antarmuka-Antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan system berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.
2. Diagram paket (Package Diagram). Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.
3. Diagram Use Case. Bersifat statis. Diagram ini memperlihatkan himpunan Use Case dan Aktor-Aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram interaksi dan Sequence (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

3. UU No. 25 Tahun 1992 (Perkoperasian Indonesia)

Koperasi adalah Badan usaha yang beranggotakan orang seorang atau badan hukum koperasi dengan melandaskan kegiatannya berdasarkan prinsip koperasi sekaligus sebagai gerakan ekonomi rakyat yang berdasarkan atas dasar asas kekeluargaan (*Arifin Sitio, Halomoan Tamb, 2001:18*).

(UU No. 25 tahun 1992 tentang perkoperasian indonesia)

1. Keanggotaanya sukarela dan terbuka. Yang keanggotaanya bersifat sukarela, dan bersedia menerima tanggung jawab keanggotaan.
2. Pengawasan oleh anggota secara Demokratis. Anggota yang secara aktif menetapkan kebijakan dan membuat keputusan. Dalam koperasi primer, anggota memiliki hak suara yang sama (satu anggota satu suara).
3. Partisipasi anggota dalam kegiatan ekonomi. Anggota menyetorkan modal mereka secara adil dan melakukan pengawasan secara demokratis. Sebagian dari modal tersebut adalah milik bersama.

2.6 Peralatan Pendukung (*Tools*)

2.6.1 Pengenalan UML (*Unified Modeling Language*)

UML singkatan dari Unified modeling Language yang berarti bahasa permodelan standar. (Chonoles 2003 : bab 1) mengatakan sebagai bahasa, bearti UML memiliki syntax dan semantik. Ketika kita menggunakan konsep UML ada aturan-aturan yang yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu pada system, bagaimana transaksinya? Bagaimana

sistem mengatasi error yang terjadi? Bagaimana keamanan system yang akan kita buat? Dan sebagainya dapat dijawab dengan UML.

Model-Model Diagram Dalam UML

Beberapa literature menyebutkan bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram perwaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, Antarmuka-Antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan system berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.
2. Diagram paket (Package Diagram). Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.
3. Diagram Use Case. Bersifat statis. Diagram ini memperlihatkan himpunan Use Case dan Aktor-Aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram interaksi dan Sequence (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

5. Diagram Komunikasi (Communication Diagram). Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi structural dari objek-objek yang menerima serta mengirim pesan.
6. Diagram Statechart (Statechart Diagram). Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (interface), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram aktivitas (Activity Diagram). Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas yang lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (Component Diagram). Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka, serta kolaborasi-kolaborasi.
9. Diagram Deployment (Deployment Diagram). Bersifat statis. Diagram ini memperlihatkan konfigurasi saat konfigurasi dijalankan (run-time). Memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. Diagram Deployment berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat

aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*Distributed Computing*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semua dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram yang lainnya (misalnya *Data Flow Diagram*, *Entity Relationship Diagram* dan sebagainya).

2.6.2 Entity Relationship Diagram (ERD)

A. Pengertian ERD

ERD menurut Ladjamudin (2005:142) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan relationship data.

Elemen-elemen Diagram Hubungan Entitas antara lain :

a. Entity

Digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. *Entitas* diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat *unsure* waktu didalamnya).

b. Relationship

Digambarkan dengan sebuah bentuk belah ketupat, *Relationship* adalah hubungan alamiah yang terjadi antara *entitas*.

c. *Relationship Degree*

Relationship Degree atau *Derajat Relationship* adalah jumlah entitas yang berpartisipasi dalam satu *Relationship*. Ada 3 *Derajat relationship* yang sering digunakan di dalam ERD :

1. *Unary Relationship*

Adalah model *Relationship* yang terjadi diantara *entity* yang berasal dari *entity set* yang sama. Disebut juga sebagai *Recursive Relationship* atau *Reflective Relationship*.

2. *Binary Relationship*

Adalah model *Relationship* antara *instance-instance* dari suatu tipe *entitas* (dua *entity* yang berasal dari *entity* yang sama).

3. *Ternary Relationship*

Merupakan *relationship* antara *instance-instance* dari tiga tipe *entitas* secara sepihak.

d. *Atribut*

Adalah sifat atau karakteristik dari tiap *entitas* maupun tiap *relationship*. Maksudnya adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud *entitas* maupun *relationship*, sehingga sering dikatakan *atribut* adalah elemen dari setiap *entitas* dan *relationship*.

Jenis-jenis atribut antara lain :

Atribut *Key* adalah atribut yang digunakan untuk menentukan suatu *entity* secara unik

1. Atribut *Simple* adalah atribut yang bernilai tunggal
2. Atribut *multivalued* adalah atribut yang memiliki sekelompok nilai untuk setiap *instant entity*.
3. Atribut *Composite* adalah atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.
4. Atribut *Derivative* adalah atribut yang dihasilkan dari atribut lain.

e. Kardinalitas (*Cardinality*)

Menunjukkan jumlah maksimum *tupelo* yang dapat berelasi dengan *entitas* pada *entitas* yang lain.

Terdapat 3 macam kardinalitas relasi, yaitu :

1. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada *entitas* pertama, hanya mempunyai satu hubungan dengan satu kejadian pada *entitas* yang kedua dan sebaliknya.

2. *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama banyak ke satu. Untuk satu kejadian pada *entitas* yang pertama dapat mempunyai banyak hubungan dengan kejadian pada *entitas* yang kedua dan sebaliknya.

3. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah *entitas* akan mempunyai banyak hubungan dengan kejadian pada *entitas* lainnya. Baik dilihat dari sisi *entitas* yang pertama, maupun dilihat dari sisi *entitas* yang kedua.

2.6.4 Bahasa Pemrograman Visual Basic

Visual Basic menurut Kusumo (2005:4) “*Visual Basic* adalah bahasa pemrograman tercepat dan termudah untuk membuat suatu aplikasi dalam *Microsoft Windows*. Dengan menggunakan metode *Graphical User Interface* (GUI). *Visual Basic* memudahkan pemrograman untuk berinteraksi langsung dengan elemen-elemen untuk setiap bentuk pemrograman. *Visual Basic* juga merupakan pemrograman terkendali kejadian (*event-driven programming*), artinya program menunggu sampai dengan adanya respon dari pemakai berupa *event* atau kejadian tertentu (seperti tombol diklik, menu dipilih, dan lain-lain). Dan ketika *event* terdeteksi kode yang berhubungan dengan *event* tersebut (*procedure event*) akan dijalankan”.

Microsoft Visual Basic adalah suatu program aplikasi yang termasuk pada bahasa pemrograman tingkat tinggi dan berbasis OOP.

Dukungan terhadap metode pemrograman yang berorientasi obyek (*object orientation programming*) adalah salah satu nilai tambah yang menjadi pertimbangan sehingga bahasa pemrograman ini banyak digunakan. Selain itu kelebihan lain yang dimiliki oleh *Microsoft Visual Basic versi 6.0* ini adalah sebagai berikut :

- a. Memiliki *compiler* andal yang dapat menghasilkan file *executable* yang lebih cepat dan efisien.
- b. Kemampuan membuat *ActiveX* dan fasilitas internet, sarana akses data yang lebih cepat dan handal untuk membuat aplikasi *database*.

BASIC (*Beginner's All Purpose Symbolic Instruction Code*) adalah suatu bahasa pemrograman “kuno” yang merupakan awal dari bahasa-bahasa

pemrograman tingkat tinggi lainnya. *BASIC* dirancang tahun 1950-an dan ditunjukkan untuk dapat digunakan oleh para *programmer* pemula. *Visual Basic* masih dapat mempertahankan sintaks atau format penulisan yang pernah dipakai oleh *BASIC*.

Visual Basic mempunyai beberapa kelebihan dari pemrograman linear (*BASIC*), diantaranya seperti :

a. Lebih Cepat (*Faster*)

Tidak seperti pemrograman linear, *Visual Basic* tidak perlu mengetikkan kode program untuk setiap objek. Di dalam *Visual Basic* hanya perlu memiliki dan mengklik pilihan-pilihan yang disediakan dengan menggunakan *mouse*, sehingga waktu untuk membuat program aplikasipun menjadi lebih singkat.

b. Resiko Kesalahan Kecil (*Small Risk On Error*)

Karena lebih sedikit mengetik, secara otomatis kesalahan juga relatif lebih kecil. Sintaks pemrograman juga tidak perlu dihafalkan lagi, karena semuanya sudah disediakan didalam menu pilihan. Demikian juga waktu *debugging* akan terpotong sangat besar, karena setiap objek tidak perlu di *debug* setiap kali digunakan.

c. Daur Ulang (*Program Recycle*)

Setiap objek dapat digunakan berulang-ulang dalam program yang sama maupun yang lain.

d. Perawatan Lebih Mudah (*Easier Maintenance*)

Dalam *BASIC* seandainya ingin merubah sifat dari sebuah modul program, tentunya harus merubah seluruh program dan semua yang ikut dipengaruhi oleh modul tersebut. Tetapi dalam *Visual Basic*, perubahan pada suatu tidak

perlu melakukan perubahan besar pada program secara keseluruhan, karena setiap objek berdiri sendiri dan tidak perlu berhubungan dengan lingkungannya.

Selain dari bermacam kemampuan dan keuntungan, *Visual Basic* juga handal dalam mengolah dan mengelola basis data atau *database*. *Database* merupakan kumpulan data yang terdiri dari satu atau lebih tabel yang terintegrasi satu sama lain, dimana setiap *user* diberi wewenang (*otoritas*) untuk dapat mengakses (mengubah, menganalisa, menghapus, memperbaiki) data dalam tabel tersebut.

Adapun sistem yang mengatur sebuah *database* dinamakan *Database Manajemen* (DBMS), yaitu kumpulan *file* yang saling berkaitan bersama dengan program untuk pengelolaannya. *Database* adalah kumpulan data, sedang program pengelolaannya berdiri sendiri dalam satu paket program komersial untuk membaca, mengisi, menghapus dan melaporkan dalam *database*.

Pembuatan program tentunya tidak terlepas dari tahapan-tahapan yang harus dikerjakan secara terstruktur untuk membantu program dalam menyelesaikan programnya dengan baik.

2.6.5 Microsoft Access 2007

Microsoft Access (atau *Microsoft Office Access*) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi *Microsoft Office*, selain tentunya *Microsoft Word*, *Microsoft Excel*, dan *Microsoft PowerPoint*.

Aplikasi ini menggunakan mesin basis data *Microsoft Jet Database Engine*, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah *Microsoft Office Access 2007* yang termasuk ke dalam *Microsoft Office 2007*.

Access juga dapat digunakan sebagai sebuah basis data untuk aplikasi Web dasar yang disimpan di dalam server yang menjalankan *Microsoft Internet Information Services* (IIS) dan menggunakan *Microsoft Active Server Pages* (ASP).

Meskipun demikian, penggunaan *Access* kurang disarankan, mengingat telah ada *Microsoft SQL Server* yang memiliki kemampuan yang lebih tinggi.

Beberapa pengembang aplikasi profesional menggunakan *Microsoft Access* untuk mengembangkan aplikasi secara cepat (digunakan sebagai *Rapid Application Development/RAD tool*), khususnya untuk pembuatan purwarupa untuk sebuah program yang lebih besar dan aplikasi yang berdiri sendiri untuk para salesman.

Microsoft Access kurang begitu bagus jika diakses melalui jaringan sehingga aplikasi-aplikasi yang digunakan oleh banyak pengguna cenderung menggunakan solusi sistem manajemen basis data yang bersifat klien/server.

Meskipun demikian, tampilan muka *Access* (*form, report, query, dan kode Visual Basic*) yang dimilikinya dapat digunakan untuk menangani basis data yang sebenarnya diproses oleh sistem manajemen basis data lainnya, seperti halnya *Microsoft Jet Database Engine* (yang secara default digunakan oleh *Microsoft Access*), *Microsoft SQL Server*, *Oracle Database*, dan beberapa produk lainnya yang mendukung ODBC.

1. Fitur *Microsoft Access*

Salah satu keunggulan *Microsoft Access* dilihat dari perspektif programmer adalah kompatibilitasnya dengan bahasa pemrograman *Structured Query Language (SQL)*; query dapat dilihat dan disunting sebagai statemen-statemen SQL, dan statemen SQL dapat digunakan secara langsung di dalam Macro dan VBA Modul untuk secara langsung memanipulasi tabel data dalam Access.

Para pengguna dapat mencampurkan dan menggunakan kedua jenis bahasa tersebut (VBA dan Macro) untuk memprogram form dan logika dan juga untuk mengaplikasikan konsep berorientasi objek.

Microsoft SQL Server Desktop Engine (MSDE) 2000, yang merupakan sebuah versi mini dari *Microsoft SQL Server 2000*, dimasukkan ke dalam *Office XP Developer Edition* dan dapat digunakan oleh *Microsoft Access* sebagai alternatif dari *Microsoft Jet Database Engine*.

Tidak seperti sebuah sistem manajemen basis data relasional yang komplit, *Microsoft JET Database Engine* tidak memiliki fitur *trigger* dan *stored procedure*. Dimulai dari *Microsoft Access 2000* yang menggunakan *Microsoft Jet Database Engine* versi 4.0, ada sebuah sintaksis yang mengizinkan pembuatan kueri dengan beberapa parameter, dengan sebuah cara seperti halnya sebuah *stored procedure*, meskipun prosedur tersebut dibatasi hanya untuk sebuah pernyataan tiap prosedurnya.

Access juga mengizinkan form untuk mengandung kode yang dapat dieksekusi ketika terjadi sebuah perubahan terhadap tabel basis data, seperti halnya *trigger*, selama modifikasi dilakukan hanya dengan menggunakan form tersebut, dan

merupakan sesuatu hal yang umum untuk menggunakan kueri yang akan diteruskan (*pass-through*) dan teknik lainnya di dalam Access untuk menjalankan *stored procedure* di dalam RDBMS yang mendukungnya.

Dalam berkas *Access Database Project* (ADP) yang didukung oleh *Microsoft Access* 2000 dan yang selanjutnya, fitur-fitur yang berkaitan dengan basis data berbeda dari versi format/struktur data yang digunakan Access (*.MDB), karena jenis berkas ini dapat membuat koneksi ke sebuah basis data MSDE atau *Microsoft SQL Server*, ketimbang menggunakan *Microsoft JET Database Engine*. Sehingga, dengan menggunakan ADP, adalah mungkin untuk membuat hampir semua objek di dalam server yang menjalankan mesin basis data tersebut (tabel basis data dengan *constraints dan trigger, view, stored procedure, dan UDF*). Meskipun demikian, yang disimpan di dalam berkas ADP hanyalah form, report, macro, dan modul, sementara untuk tabel dan objek lainnya disimpan di dalam server basis data yang membelakangi program tersebut.

2. Pengembangan dengan Access

Access mengizinkan pengembangan yang relatif cepat karena semua tabel basis data, kueri, form, dan report disimpan di dalam berkas basis data miliknya (*.MDB). Untuk membuat *Query*, *Access* menggunakan *Query Design Grid*, sebuah program berbasis grafis yang mengizinkan para penggunanya untuk membuat query tanpa harus mengetahui bahasa pemrograman SQL.

Di dalam *Query Design Grid*, para pengguna dapat memperlihatkan tabel basis data sumber dari query, dan memilih field-field mana yang hendak dikembalikan oleh proses dengan mengklik dan menyeretnya ke dalam grid. Join juga dapat dibuat dengan cara mengklik dan menyeret field-field dalam tabel ke dalam field

dalam tabel lainnya. Access juga mengizinkan pengguna untuk melihat dan memanipulasi kode SQL jika memang diperlukan.

Bahasa pemrograman yang tersedia di dalam *Access* adalah *Microsoft Visual Basic for Applications* (VBA), seperti halnya dalam beberapa aplikasi *Microsoft Office*. Dua buah pustaka komponen *Component Object Model* (COM) untuk mengakses basis data pun disediakan, yakni *Data Access Object* (DAO), yang hanya terdapat di dalam *Access 97*, dan *ActiveX Data Objects* (ADO) yang tersedia dalam versi-versi *Access* terbaru.

2.6.6 Crystal Report

Crystal Reports dirancang untuk membuat laporan yang dapat digunakan dengan bahasa pemrograman berbasis Windows, seperti *Borland Delphi*, *Visual Basic*, *Visual C/C++*, dan *Visual Interdev*. Menurut Hadi, (2003) ada beberapa kelebihan dari *Crystal Reports* ini adalah :

- a. Dari segi pembuatan laporan, tidak terlalu rumit yang memungkinkan paraprogrammer pemula sekalipun dapat membuat laporan yang sederhana tanpamelibatkan banyak kode pemrograman.
- b. Integrasi dengan bahasa-bahasa pemrograman lain yang memungkinkan dapatdigunakan oleh banyak programmer dengan masing-masing keahlian.
- c. Fasilitas impor hasil laporan yang mendukung format-format populer seperti *Microsoft Word*, *Excel*, *Access*, *Adobe Acrobat Reader*, HTML dan sebagainya.

dari subsistem perangkat keras dan subsistem perangkat lunak. Subsistem perangkat keras dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar”.

2.2 Definisi Informasi

Informasi menurut Jogiyanto (2005:8) adalah “data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya”. Sumber dari informasi adalah data dan data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu.

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna bagi penerimanya yang menggambarkan suatu kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali.

1. Siklus Informasi

Informasi merupakan keluaran (*output*) dari proses data. *Output* ini harus sudah tersusun dengan baik dan mempunyai arti bagi yang menerimanya sehingga fungsi dapat dirasakan secara utuh dan relevan guna mencapai suatu keputusan yang diinginkan.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan