BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan dalam mendefinisikan sistem yaitu yang menekankan pada komponen atau elemennya. Menurut Jerry Fitzgerald, Andra F.Frizgerald dan Warren D.Stallings,Jr yang dikutip **JOG[5]** pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sebagai berikut :

“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”.

Sedangkan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem yaitu :

“Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”.

Menurut M.J Alexander yang dikutip **WAH[16],** sistem merupakan suatu group dari elemen-elemen baik yang berbentuk fisik maupun non fisik yang menunjukkan suatu kumpulan saling berhubungan diantaranya dan berinteraksi bersama-sama menuju satu atau lebih tujuan, sasaran atau akhir dari sebuah sistem.

Jadi, dapat disimpulkan bahwa sistem ialah suatu kesatuan atau kumpulan yang utuh yang terdiri dari bagian-bagian atau elemen-elemen yang berbentuk fisik maupun non fisik yang saling berhubungan dan berinteraksi untuk mencapai sasaran atau tujuan tertentu.

2.2 Konsep Dasar Informasi

Beberapa ahli mengemukakan pendapatnya tentang pengertian informasi, diantaranya :

Menurut **JOG[5]** “Informasi yaitu hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya, yang menggambarkan suatu kejadian-kejadian(*events*) yang nyata(*fact*) yang berguna untuk para pengambil keputusan.”

Sedangkan menurut **AMS[1]** “Informasi adalah data yang telah diolah dengan cara tertentu sesuai dengan bentuk yang diperlukan.”

Jadi, dapat disimpulkan bahwa informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan keputusan.

2.3 Konsep Dasar Sistem Informasi

Menurut Robert A. Leith dan K. Roscoe Davis “Sistem Informasi 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”. **JOG[5]**

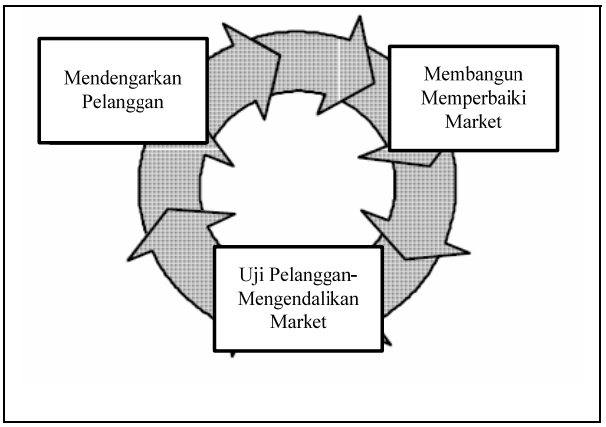
Jadi, dapat disimpulkan bahwa sistem informasi adalah suatu sistem buatan manusia yang bertujuan untuk menyajikan informasi yang dibutuhkan oleh suatu organisasi.

2.4 Sistem Informasi Manajemen

Merupakan penerapan sistem informasi di dalam organisasi untuk mendukung informasi yang dibutuhkan oleh semua tingkatan manajemen. Menurut **JOG[5]**, SIM adalah kumpulan interaksi sistem informasi yang menyediakan informasi baik untuk kebutuhan manajerial maupun kebutuhan operasi. Secara teori, komputer tidak harus digunakan di dalam SIM, tetapi kenyataannya tidaklah mungkin SIM yang kompleks dapat berfungsi tanpa melibatkan elemen non komputer dan elemen komputer. SIM merupakan suatu sistem yang melakukan fungsi-fungsi untuk menyediakan semua informasi yang mempengaruhi semua operasi organisasi. SIM merupakan kumpulan dari sistem informasi.

2.5 Metode Pengembangan Sistem

Dalam melakukan perancangan sistem digunakan metode prototyping. Seperti pada semua metode, prototyping dimulai dari pengumpulan kebutuhan. Dengan perencanaan yang cepat akan dibentuk konstruksi dari prototypenya. Prototype ini dievaluasi oleh pelanggan dan digunakan untuk mengelola kembali kebutuhan dari perangkat lunak yang dikembangkan. Suatu proses iterasi terjadi, setelah prototype disesuaikan dengan kebutuhan pelanggan, sementara pihak pengembang makin mengerti keinginan pemakai.



Gambar 2.1 Model Prototyping

(Sumber : **PRE[11]**)

Kadang-kadang klien hanya memberikan beberapa kebutuhan umum *software* tanpa detail *input*, proses atau detail *output*. Di lain waktu mungkin dimana tim pembangun (*developer*) tidak yakin terhadap efisiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan *form user interface*. Ketika situasi seperti ini terjadi model prototyping sangat membantu proses pembangunan *software*. Proses pada model prototyping yang digambarkan pada gambar 2.1, bisa dijelaskan sebagai berikut:

1. Pengumpulan kebutuhan: *developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan
2. Perancangan : perancangan dilakukan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan prototype.
3. Evaluasi prototype: klien mengevaluasi prototipe yang dibuat dan digunakan untuk memperjelas kebutuhan *software*.

Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. Prototype-prototype dibuat untuk memuaskan kebutuhan klien dan untuk memahami kebutuhan klien lebih baik. Prototype yang dibuat dapat dimanfaatkan kembali untuk membangun *software* lebih cepat, namun tidak semua prototype bisa dimanfaatkan.

2.6 Konsep Objek

Pendekatan pengembangan sistem berorientasi objek adalah cara memandang persoalan menggunakan model-model yang diorganisasikan seputar konsep objek yang mengkombinasikan struktur data dan perilaku suatu entitas. Pada pendekatan ini organisasi perangkat lunak adalah sebagai kumpulan objek diskrit yang saling bekerja sama, berkomunikasi dan berinteraksi menuju sasaran tertentu.

Paradigma *object oriented* terdiri dari 3 bagian, yaitu :

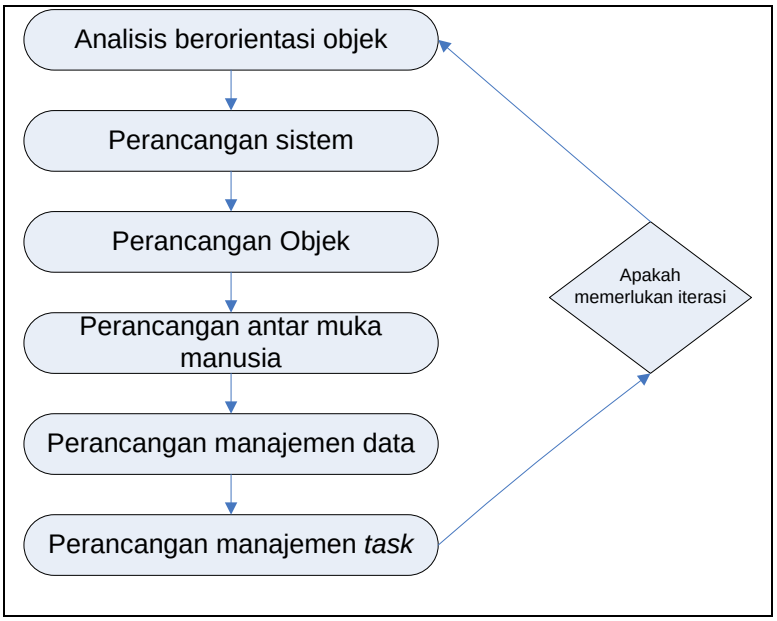
1. Analisis Berorientasi Objek, analisa ini dikembangkan dari konsep yang paling baik dari beberapa disiplin, yaitu Pemodelan informasi, bahasa pemrograman berorientasi objek dan *knowledge-based system.*
2. Desain Berorientasi Objek, desain berorientasi objek mempunyai tiga dasar:

   a. Notasi. Notasi digunakan untuk memberikan gambaran tentang gagasan sistem kepada anggota tim atau pihak lain yang memerlukan.

   b. Strategi. Setiap proyek tidak harus dimulai dari awal. Desain untuk *problem domain* yang biasa ditemui dapat digunakan untuk pemecahan masalah.

c. Kriteria yang baik . Evaluasi suatu desain dapat dilakukan secara objektif dengan melihat apakah diterima, ditolak atau direvisi.

Berikut adalah tahap-tahap perancangan yang diadaptasi dari metode **Coad-Yourdon** yang digambarkan dalam gambar 2.2:



Gambar 2.2 Tahap-tahap perancangan **Coad-Yourdon**

3. Pemrograman Berorientasi Objek adalah inovasi dalam pengembangan perangkat lunak Pada saat pemecahan suatu masalah dengan pemrograman berorientasi objek, tidak dipikirkan bagaimana pemecahan masalah tersebut dapat diimplementasikan kedalam fungsi, tetapi perhatian lebih ditujukan pada pemecahan masalah menjadi objek.

Metode pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama:

1. *Encapsulation* (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas bersam-sama dalam suatu objek, sehingga prosedur atau

fungsi lain dari luar tidak dapat mengaksesnya, kecuali prosedur yang berada dalam objek itu sendiri.

2. *Inheritance* (pewarisan) adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metoda dari induknya langsung.
3. *Polymorphism* yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda, artinya bahwa operasi yang sama mungkin berbeda dalam kelas yang berbeda.

Pendekatan pengembangan sistem berorientasi objek mempunyai beberapa keunggulan yaitu sebagai berikut :

1. Pendekatan objek menuntun penggunaan ulang (*re-use*) komponen-komponen program sebelumnya. Penggunaan kembali komponen-komponen sebelumnya ini membuat pengembangan perangkat lunak menjadi lebih cepat dan berkualitas lebih tinggi.
2. Perangkat lunak yang dikembangkan dengan berorientasi objek mempermudah pemeliharaan.
3. Sistem berorientasi objek lebih mudah diadaptasi dan diskala menjadi sistem yang lebih besar, karena system yang besar dibangun dengan merakit subsistem-subsistem yang dapat digunakan secara berulang-ulang.
4. Pendekatan objek bekerja yang mendekati *kognisi* manusia.
5. Pendekatan objek menghasilkan sistem yang dibangun diatas bentuk-bentuk antara yang stabil sehingga lebih mampu untuk mengikuti perubahan.

6. Pendekatan objek dapat digunakan tidak hanya pada perancangan perangkat lunak saja tetapi juga seluruh proses pengembangan perangkat lunak.

2.7 Pengenalan UML (*Unified Modeling Language*)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). UML Menggabungkan konsep berarah objek seperti Booch, OMT, OOSE, dsb sehingga menjadi bahasa pemodelan tunggal. UML menekankan pada apa yang dapat dikerjakan dengan metode-metode tersebut.

UML memiliki bagian-bagian yang dapat mempermudah pembuatan sistem diantaranya :

a. *Use case diagram*, menggambarkan interaksi antara *actor* dengan proses atau sistem yang dibuat.
b. *Class diagram*, menggambarkan interaksi antar *class* serta atribut-atribut yang melekat pada *class* tersebut.

c. *Sequence diagram*, merupakan interaksi antara objek-objek dalam suatu sistem dan terjadi komunikasi yang berupa pesan (*message*) serta parameter waktu.

d. *Collaboration Diagram*, *Collaboration Diagram* hampir sama dengan *sequence diagram* tetapi berbeda pada objek yang di titik tekankan , *collaboration* lebih menekankan pada pemunculan objek itu sendiri sedangkan *sequence diagram* lebih pada penyampaian *message* dengan parameter waktu.

e. *Activity diagram*, menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti.

f. *Statechart Diagram***,** yaitu diagram yang menggambarkan perubahan *state* dari *state* yang satu ke *state* lainnya.

g. *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*.

h. *Deployment/physical diagram*, menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana

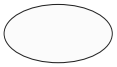
kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.

2.8 IBM Rational Rose

*IBM Rational Ros*e menurut **NUG[9]** adalah kakas (tools) pemodelan visual untuk pengembangan sistem berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam analisis dan perancangan sistem. Dalam *Rational Rose,* pemodelan adalah cara melihat sistem dari berbagai sudut pandang. Ia mencakup semua diagram yang dikenal dalam UML, aktor-aktor yang terlibat dalam sistem, *use case,* objek-objek, kelas-kelas, komponen-komponen, serta simpul-simpul penyebaran (*deployment node*).

1. *Use case Diagram*

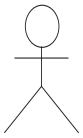
Use Case Diagram menggambarkan sejumlah *eksternal actors* dan hubungannya dengan *use case* yang diberikan oleh sistem. Dengan kata lain, *use case* menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem. *Use* case menurut **RIY[13]** adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagram.* Dalam *Use case Diagram* terdapat *use case* yang dapat digambarkan sebagai berikut :



Gambar 2.3 *Use Case*

(Sumber : **NUG[9]**)

Dalam pemodelan sistem dengan UML, aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan. Aktor dapat digambarkan sebagai berikut :

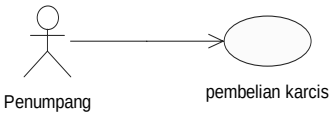


Gambar 2.4 Aktor

(Sumber : **NUG[9]**)

Use case dan aktor masing-masing tidak berdiri sendiri. Mereka saling terhubung dengan apa yang dinamakan relasi. Dalam hal ini, ada berbagai relasi yang dikenal dalam model UML, yaitu :

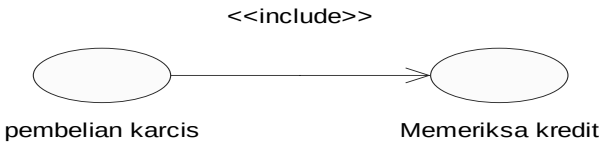
1. Relasi Asosiasi, yaitu relasi yang terjadi antara aktor dengan *use case* biasanya berupa asosiasi. Relasi Asosiasi Dalam UML dapat dicontohkan sebagai berikut:



Gambar 2.5 Relasi Asosiasi

(Sumber : **NUG[9]**)

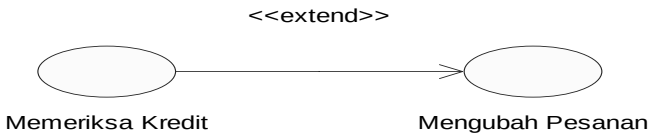
2. Include Relationship, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana kondisi ini sebuah *use case* adalah bagian *use case* lainnya.



Gambar 2.6 Include Relationship

(Sumber : **NUG[9]**)

3. Extend Relationship, yaitu kelakukan yang hanya berjalan di bawah kondisi tertentu. Extend Relationship memungkinkan suatu *use case* memiliki kemungkinan untuk memperluas fumgsionalitas yang disediakan *use case* yang lainnya.



Gambar 2.7 Extend Relationship

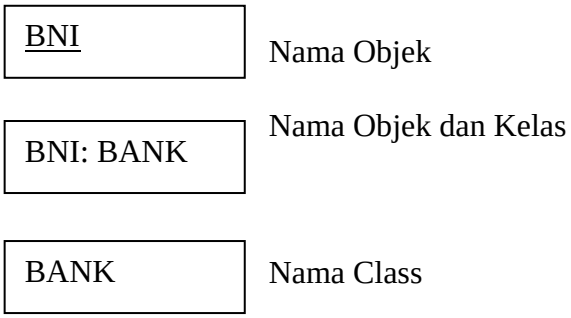
(Sumber : **NUG[9]**)

2. *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antara sejumlah *object* dalam urutan waktu. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antar *object* juga interaksi antar *object* yang terjadi pada titik tertentu dalam ekseskusi sistem.

Dalam UML, *object* pada *sequence diagram* digambarkan dengan segi empat yang berisi nama dari *object* yang digarisbawahi. Pada *object* terdapat 3 cara untuk menamainya yaitu : nama *object* , nama *object* dan *class* serta nama *class.*

Contoh :



Gambar 2.8 Penamaan Objek

(Sumber : **RIY[13]**)

Dalam *sequence diagram* setiap *object* hanya memiliki garis yang digambarkan garis putus – putus ke bawah. Pesan antar *object* digambarkan dengan anak panah dari objek yang mengirimkan pesan ke *object* yang menerima pesan.

3. *Collaboration Diagram*

Collaboration Diagram merupakan cara alternatif untuk menggambarkan skenario dari sistem. Diagram ini menggambarkan interaksi objek yang diatur objek sekelilingnya dan hubungan antara setiap objek dengan objek yang lainnya. *Collaboration diagram* berisi :

1. Objek yang digambarkan dengan segiempat
2. Hubungan antara objek yang digambarkan dengan garis penghubung
3. Pesan yang digambarkan dengan teks dan panah dari objek yang mengirim pesan ke penerima pesan

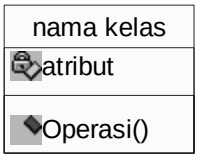
Perbedaan Antara *Sequence Diagram* dengan *Collaboration Diagram* adalah sebagai berikut :

Sequence Diagram memberikan cara untuk melihat skenario dari sistem berdasarkan waktu (apa yang terjadi pertama kali, apa yang terjadi selanjutnya). *User* akan lebih mudah membaca dan mengerti tipe diagram ini. Karenanya, sangat berguna pada fase analisis awal.

Sedangkan *Collaboration Diagram* cenderung untuk memberikan gambaran besar dari skenario selama kolaborasi disusun dari *object* sekelilingnya dan hubungan antar *object* yang satu dengan lainnya. Diagram ini akan nampak digunakan pada pengembangan tahap desain, ketika kita merancang implementasi dari hitungan.

4. *Class Diagram*

Kelas adalah sesuatu yang membungkus (*encapsulate*) informasi atau atribut dan perilaku atau operasi dalam dirinya. Diagram kelas menurut **NUG[8],** adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang sedang dikembangkan.



Gambar 2.9 Notasi Kelas Dalam UML

(Sumber: **NUG[9]**)

5. *Activity Diagram*

Activity Diagram menurut **NUG[9]**, adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case. Activity Diagram* digunakan untuk memodelkan aspek dinamis dari sistem. *Activity Diagram* mirip diagram alir (*flowchart*), memperlihatkan aliran kendali dari suatu aktifitas ke aktifitas lainnya.

6 Indikator *Multiplicity*

Walaupun *multiplicity* ditentukan untuk *class*, *multiplicity* menentukan banyaknya *object* yang terlibat dalam relasi. *Multiplicity* menentukan banyaknya *object* yang terhubung satu dengan yang lainnya. Indikator *multiplicity* terdapat pada masing – masing akhir garis relasi, baik pada asosiasi maupun agregasi. Beberapa contoh *multiplicity* adalah :

1	Tepat Satu
0..*	Nol atau lebih
1..*	Satu atau lebih
0..1	Nol atau Satu
5..8	range 5 s.d. 8
4..6,9	Range 4 s.d. 6 dan 9

2.9 Basis Data (*database*)

Menurut **AMS[1]** *Database* adalah “Kumpulan semua data yang disimpan dalam satu *file* atau beberapa *file*. Secara operasional *database* adalah daftar yang terdiri dari beberapa kolom yang masing-masing kolom berisikan satu jenis (*item*) data”.

Jadi, *database* merupakan kumpulan file, tabel atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis. Basis data atau lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama, yaitu kemudahan dan kecepatan dalam pengambilan kembali data atau arsip.

2.9.1 *Database Management System* (DBMS)

Menurut **AMS[1]** DMBS adalah “Suatu rangkaian program-program yang mengelola sebuah *database* dan menyediakan mekanisme-mekanisme melalui mana jenis-jenis data dapat disimpan, dicari kembali, dan diubah.

Dengan DBMS terbentuklah hubungan antar *file* sehingga data dalam *file* tersebut dapat dihubungkan satu sama lain, misalnya untuk keperluan pembuatan laporan. DBMS dapat mengelola berbagai data yang disimpan pada *file* yang berbeda, karena itu perlu dipahami perbedaan antara pandangan fisik, yaitu

bagaimana data disimpan dalam komputer, dengan pandangan logis atau pandangan pendesain dana pemakai terhadap data.

Jadi, *database* manajemen sistem merupakan alat untuk mengeloloa berbagai data yang disimpan pada file berbeda sehingga data dalam file tersebut dapat dihubungkan satu sama lain.

2.9.2 Sistem Manajemen *Database*

Terdapat banyak jenis sistem manajemen *database* yang membuat paket perangkat lunak ini sebagai komponen yang mempunyai nilai khusus pada sistem informasi. Dengan mengetahui karakteristik kita dapat mengetahui bagaimana DBMS bekerja dan bagaimana DBMS dapat digunakan. DBMS dapat digunakan untuk keperluan sebagai berikut :

1. Memelihara koleksi data yang dapat dipakai secara bersama.
2. Membentuk hubungan antara *item* data.
3. Memelihara data yang kelebihan (*redudancy*).
4. Memelihara independesi antara program dengan data.
5. Menyediakan data lengkap untuk pembuatan laporan.
6. Menyediakan cara pencarian data dan pengawasan terhadap penyimpanan data.
7. Memungkinkan dilakukannya pengembangan aplikasi.

2.9.3 Jenis *Database*

Alasan untuk membedakan antara struktur dan struktur jaringan adalah bahwa beberapa paket perangkat lunak DBMS dapat menyediakan suatu struktur jaringan, dimana lain-lainnya dapat menyediakan struktur pohon. Pada

kenyataannya, itu merupakan salah satu karakteristik yang membedakan tiga jenis sistem manajemen :

1. Database Hierarki

Sistem *database* hierarki menyediakan data yang dapat ditampilkan dengan struktur pohon, yaitu aplikasi dimana orang tua boleh mempunyai banyak anak, tetapi seorang anak hanya dapat mempunyai satu orang tua.

2. Database Jaringan

Sistem manajemen *database* jaringan dapat mempersiapkan struktur jaringan, yaitu suatu aplikasi dimana seorang anak dapat mempunyai lebih dari satu orang tua.

3. Database Relasional

Sistem manajemen *database* relasional lebih fleksibel dibandingkan dengan kedua DBMS hierarki dan jaringan, sering mudah didesain dan digunakan.

Dengan menggunakan terminologi *database* relasional disebutkan bahwa data diorganisasikan dalam bentuk tabel dua dimensi yang sederhana. Tabel ini merupakan dasar suatu penampilan grafik data dalam suatu *file*. Dan tiap jenis data dapat dinyatakan dengan cara tersebut, dengan resiko terjadi beberapa kelebihan.

2.10 GSM (*Global System for Mobile Communication*)

GSM pertama kali diperkenalkan di benua Eropa tahun 1991. menjelang akhir tahun 1993, beberapa negara non-eropa di Amerika Selatan, Asia, dan Australia telah mengadopsi GSM dan turunannya secara teknis : DCS 1800, yang menangani layanan komunikasi personal, disebut *Personal Communication*

Service (PCS) di pita 1,8 GHz sampai 2,0 GHz. Di Indonesia sendiri, sampai saat ini *operator* GSM di pita 900 MHz ada tiga, yakni PT.Telkomsel, PT.Satelindo dan PT. Excelcomindo Pratama, sedangkan pita 1800 MHz adalah PT. Indosat dengan IM3-nya (*Indosat MultiMedia Mobile*).

GSM adalah sistem komunikasi seluler standar generasi kedua yang dikembangkan untuk mengatasi masalah sistem yang terpisah-pisah pada sistem seluler generasi pertama di Eropa. GSM merupakan sistem seluler pertama di dunia yang memiliki rincian modulasi digital dan arsitektur di tingkat jaringan serta layanan. Awalnya, GSM dikembangkan dengan tujuan supaya dapat berfungsi sebagai layanan seluler di seluruh wilayah Eropa, dan menjanjikan layanan jaringan yang merentang luas melalui penggunaan ISDN.

2.10.1 Layanan GSM

Menurut **ROS[14]** secara umum layanan *mobile* data pada GSM dibedakan menjadi dua jenis :

1. *Message Service*
    a. *Short Message Service* (SMS) merupakan fasilitas untuk mengirim dan menerima pesan singkat dalam bentuk teks dalam jaringan ponsel.
    b. *Enhanced Message Service* (EMS) merupakan fasilitas pengirim pesan yang dapat berupa bentuk teks biasa, gambar dan suara pada ponsel.
    c. *Multimedia Message Service* (MMS) merupakan fasilitas pengiriman pesan berupa teks biasa, foto, suara, gambar bergerak

(animasi multimedia), gambar berwarna bahkan klip audio video pada jaringan ponsel.

2. *Mobile Internet*, layanan-layanan yang ditawarkan antara lain *Mobile Chatting*, *e-mail Browsing* dengan metode *connect* WAP, GPRS dan *Dial Up.*

2.10.2 SMS (*Short Message Service*)

SMS (*Short Message Service*) ialah layanan pesan singkat yang terdapat pada telepon seluler. SMS menurut **ROS[14]**, adalah sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, memungkinkan dilakukannya pengiriman pesan dalam *alphanumeric* antara terminal pelanggan atau antara terminal pelanggan dengan sistem eksternal seperti *email*, *paging*, *voice mail*,dan lain-lain. Isu SMS pertama kali muncul di belahan Eropa sekitar tahun 1991 bersama sebuah teknologi komunikasi *wireless* yang saat ini cukup banyak penggunanya, yaitu *Global System for Mobile Communication* (GSM). Dipercaya bahwa pesan pertama yang dikirim menggunakan SMS dilakukan pada bulan desember 1992, dikirimkan dari sebuah *personal computer* (PC) ke telepon *mobile* (bergerak) dalam jaringan GSM milik Vodafone Inggris.

Perkembangannya kemudian merambah ke benua Amerika, dipelopori oleh beberapa operator komunikasi bergerak berbasis digital seperti BellSouth Mobility, PrimeCo, Nextel dan beberapa operator lain. Teknologi *digital* yang digunakan bervariasi dari yang berbasi GSM, *Time Division Multiple Access* (TDMA), hingga *Code Division Multiple Access* (CDMA).

Dalam sistem SMS, mekanisme utama yang dilakukan dalam sistem adalah melakukan pengiriman pesan singkat dari satu terminal pelanggan ke terminal lain. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem SMS yang bernama *Short Message Service Center* (SMSC) disebut juga *Message Center*. SMSC merupakan sebuah perangkat yang merupakan tugas *store and forward* trafik *short message.* Di dalamnya termasuk penentuan atau pencarian rute tujuan akhir dari *short message.*

Tabel 2.1 Daftar Alamat SMS Center

No	Operator GSM	Nomor SMS Center
1.	Satelindo	+6216124
2.	Excelcomindo	+6218445009
3.	Telkomsel	+6281100000
4.	IM3	+62855000000

2.10.3 SMS Premium

SMS Premium merupakan bisnis SMS yang mempunyai tariff minimal Rp. 2000,00 dengan menggunakan *short number.* Secara singkat proses SMS premium dari pihak-pihak yang terlibat yaitu pelanggan mengirimkan SMS ke sebuah nomor pendek (*short number*) empat digit, yang dikenal juga dengan istilah AND (*abbreviated dialing number*), SMS tersebut diterima oleh SMS center milik operator, oleh operator SMS tersebut akan diarahkan ke CP. SMS tersebut diolah oleh CP sesuai dengan jenis layanan yang diminta, dan selanjutnya dikirim kembali ke operator. Kemudian, operator meneruskan hasil dari CP kepada pelanggan.

Model layanan untuk SMS premium dapat dikelompokkan menjadi dua bagian yaitu :

1. SMS *pull*

   Jenis layanan ini identik dengan konten atas dasar permintaan pelanggan secara tidak berlangganan. Contohnya, info kerja, info kelulusan SPMB, dan info lain yang sejenisnya. Pelanggan yang menggunakan metode ini setiap ingin dikirim konten diharuskan mengirimkan *keyword.*

2. SMS *push*

   Jenis layanan ini identik dengan konten berlangganan seperti mengetik "REG QURAN" yang dikirimkan ke *short number* tertentu, maka *subscriber* akan dikirimi konten SMS secara rutin entah satu hari sekali, dua hari sekali tergantung layanan yang disediakan oleh CP. Pernyataan berhenti bisa dengan mengirim *keyword* semisal "UNREG QURAN" dikirim ke *short number* yang sama ketika menyatakan berlangganan.

2.11 Pengertian Undian

Undian yaitu, tiap-tiap kesempatan yang diadakan oleh suatu badan atau perusahaan untuk mereka yang telah memenuhi syarat-syarat tertentu dapat ikut serta memperoleh hadiah berupa uang atau benda yang akan diberikan kepada peserta-peserta yang ditunjuk sebagai pemenang dengan jalan undi atau dengan cara lain menentukan untung yang tidak dapat dipengaruhi oleh peserta itu sendiri.

Menurut UU Nomor 22 Tahun 1954 undian berhadiah diijinkan. Adapun persyaratan undian yaitu, peserta tidak dapat ikut bermain atau ikut serta menentukan untuk menjadi pemenang, peserta lebih bersikap pasif dalam penentuan pemenang dan penentuan pemenang dilakukan dalam waktu satu kali (dalam satu waktu atau periode yang relative berjangka lama).

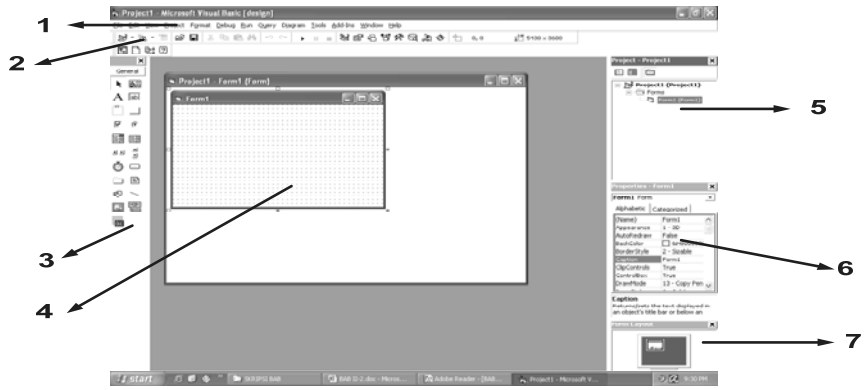
2.12 Perangkat Lunak Pendukung

Dalam pembuatan aplikasi undian dengan SMS ini, menggunakan software atau perangkat lunak yaitu : Microsoft Visual Basic 6.0 Enterprise Edition, Microsoft Access 2003, Komponen *Active-X Control* Mobile FBUS Lite Version, Komponen Active Skin 4.0, IBM Rational Rose 7.0 Enterprise Edition dan Crystal Report 8.5.

2.12.1 *Microsoft Visual Basic 6.0 Enterprise Edition*

Menurut **NOV[7]**, menyebutkan program ialah kumpulan pernyataan-pernyataan yang disusun dan disimpan dalam suatu dokumen agar dapat dijalankan dalam satu kesatuan waktu (bersamaan). Sedangkan *Visual Basic* adalah sebuah bahasa pemrograman yang digunakan untuk membuat program aplikasi berbasis orientasi objek atau *object oriented program* (OOP). *Visual Basic* merupakan pengembangan dari bahasa BASIC yang pernah poluper. Dengan *Visual Basic* versi terbaru 6.0 ini perancangan sebuah program akan lebih mudah karena didukung oleh komponen-komponen pelengkap yang memiliki standar sistem operasi *Windows.*

Lingkungan kerja Visual Basic 6.0 adalah sebagai berikut :



Gambar 2.10 Lingkungan Kerja Visual Basic 6.0

(Sumber : **NOV[7]**)

Keterangan Gambar :

1. *Menu bar* berfungsi sebagai perintah untuk menjalankan tugas tertentu. Misalnya kalau menyimpan suatu *project* maka bisa memilih menu *file* dan memilih *save project*.
2. *Toolbar* adalah tombol-tombol yang mewakili suatu perintah tertentu, dalam *toolbar* ini terdapat *icon-icon* (perintah bergambar).
3. *Toolbox* berisi *icon-icon* yang nantinya akan dipakai dalam pembangunan program aplikasi.
4. *Form Window* merupakan suatu objek yang dipakai sebagai tempat bekerja program aplikasi. *Form window* secara otomatis akan tersediakan oleh *visual basic 6.0* secara bersamaan ketika akan membuat sebuat *project* baru. Melalui *form window* ini dapat meletakkan berbagai macam objek interaktif, misalnya teks, gambar, tombol dan sebagainya.
5. *Project Explorer* adalah jendela yang mengandung semua *file* di dalam aplikasi *visual basic*. Setiap aplikasi *visual basic* disebut dengan istilah *project,* dan setiap *project* dapat mengandung lebih satu *file* misalnya *form, modul, class* dan sebagainya.
6. *Jendela Properties* adalah jendela yang mengandung semua informasi tentang objek yang terdapat pada aplikasi *visual basic*. Fungsi *properties* ini sebagai pengatur dan karakteristik dari suatu objek misalnya untuk merubah tinggi objek, warna objek dan sebagainya.
7. *Form Layout Window* berfungsi untuk menggambarkan posisi *form* terhadap layar monitor ketika sedang di jalankan (*run).*

Dalam pemrograman berbasis obyek (OOP), khususnya dalam *visual basic* ada beberapa istilah yaitu :

1. *Object* ialah komponen di dalam sebuah program.
2. *Property* ialah karakteristik yang dimiliki objek.
3. *Method* ialah aksi yang dapat dilakukan oleh objek.
4. *Event* ialah kejadian yang dapat dialami oleh objek.

2.12.2 Microsoft Access 2003

Microsoft Acces menurut **NOV[7]** merupakan DBMS (*database management system*) bertipe relasional untuk *platform* sistem operasi *windows*. Ms.Access adalah RDBMS (*relational database management system*) yang cukup baik digunakan untuk aplikasi-aplikasi berukuran kecil hingga menengah.

2.12.3 Komponen Active-X Control Mobile FBUS 1.5 Lite Version

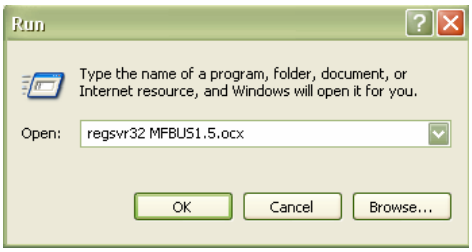
Active-X Control menurut **NOV[7]** adalah objek-objek control yang disertakan pada *visual basic* seperti *textbox, label, picturebox, command button* dan lainnya. Mobile FBUS 1.5 Lite Version merupakan suatu komponen *Active-X Control* yang dibuat oleh softwarecave yang berfungsi untuk menghubungkan konfigurasi ponsel dengan komputer. *Active-X Control* ini di *download* secara gratis dari www.softwarecave.com dan ada juga yang versi professional yang harus bayar apabila ingin di *download.*

Cara menginstalasi *Active-X Control* cukup mudah dengan cara meng-*copy* semua komponennya ke windows\system32, yaitu :

1. gjfbus15.dll
2. MFBUS15.ocx
3. MFBUS15.oca

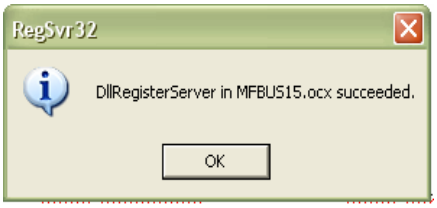
4. REGSVR32.EXE

Setelah itu, buka Menu Run terus ketikkan regsvr32 MFBUS15.ocx lalu klik OK seperti gambar 2.11.



Gambar 2.11 Menu Run MFBUS

Setelah itu, akan muncul pesan seperti gambar 2.12 dibawah ini.



Gambar 2.12 Pesan Komponen Mobile FBUS sudah didaftarkan

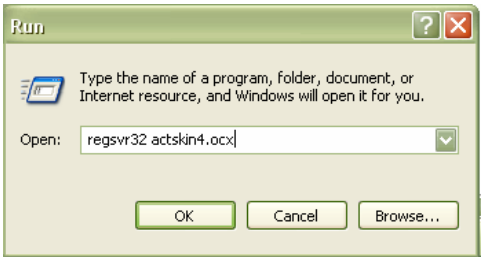
2.12.4 Komponen Active Skin 4.0

Komponen Active Skin 4.0 merupakan software yang dapat memperindah tampilan form dengan cara meletakkan skin kedalam form. Cara menginstalasi Active Skin cukup mudah dengan cara meng-*copy* semua komponennya ke windows atau system32, yaitu :

3 actskin4.ocx

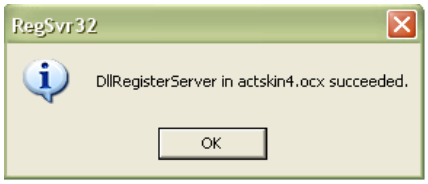
4 winaqua.skn

setelah itu, buka Menu Run terus ketikkan actskin4.ocx lalu klik OK seperti gambar 2.13.



Gambar 2.13 Menu Run Active Skin

Setelah itu, akan muncul pesan seperti gambar 2.14 dibawah ini



Gambar 2.14 Pesan Komponen Active Skin sudah didaftarkan

2.12.5 Cristal Report 8.5

Crystal Report menurut **IMA[4],** adalah salah satu *software* yang handal dari Seagate dalam hal pembuatan laporan. Dengan memanfaatkan *tools* yang dimiliki crystal report dapat dihasilkan laporan yang bervariasi hasilnya.

Jadi, pemilihan dalam pembuatan laporan pemenang undian bunda bakery menggunakan Crystal Report 8.5 karena pencetakan hasil laporan sangatlah bervariasi tergantung keinginan pengguna.