BAB II

LANDASAN TEORI

2.1 Pengertian Sistem Informasi

Informasi merupakan hal yang sangat penting bagi kehidupan sehari-hari dalam pengambilan keputusan. Informasi itu sendiri diperoleh dari sistem informasi (*Information System*).

2.1.1 Sistem

Dari segi Etimologi, kata sistem sebenarnya berasal dari Bahasa Yunani yaitu "*Systema*", yang dalam Bahasa Inggris dikenal dengan "*System*", yang mempunyai satu pengertian yaitu sehimpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan satu keseluruhan yang tidak terpisahkan. Menurut filsuf **Stoa**, bahwa sistem adalah gabungan dari keseluruhan langit dan bumi yang bekerja bersama-sama, sehingga dapat kita lihat bahwa sistem terdiri dari unsur-unsur yang bekerja sama membentuk suatu keseluruhan dan apabila salah satu unsur tersebut hilang atau tidak berfungsi, maka gabungan keseluruhan tersebut tidak dapat lagi kita sebut suatu sistem. Berikut ini adalah definisi kata Sistem menurut beberapa para ahli.

Menurut **Buckley**, Pengertian Sistem didefinisikan sebagai berikut :

"*Sistem adalah suatu kebulatan atau totalitas yang berfungsi secara utuh, disebabkan adanya saling ketergantungan diantara bagian-bagiannya. (A whole that functions as a whole by virtue of interdependence of its parts)*".

Menurut **H. Kerzner**, Pengertian Sistem didefinisikan sebagai berikut :

"*Sistem adalah sekelompok komponen yang terdiri dari manusia dan/atau bukan manusia (non-human) yang diorganisir dan diatur sedemikian rupa sehingga komponen-komponen tersebut dapat bertindak sebagai satu kesatuan dalam mencapai tujuan, sasaran bersama atau hasil akhir. Pengertian ini, mengandung arti pentingnya aspek pengaturan dan pengorganisasian komponen dari suatu sistem untuk mencapai sasaran bersama, karena bila tidak ada sinkronisasi dan koordinasi yang tepat, maka kegiatan masing-masing komponen, sub-sistem, atau bidang dalam suatu organisasi akan kurang saling mendukung"*.

Menurut **B.S. Blanchard (1990)**, Pengertian Sistem didefinisikan sebagai berikut : "*Engineering System adalah aplikasi yang efektif dari usaha-usaha ilmu pengetahuan dan engineering dalam rangka mewujudkan kebutuhan operasional menjadi suatu sistem konfigurasi tertentu, melalui proses yang saling terkait berupa definisi keperluan analisis fungsional, sintesis, optimasi, desain, tes, dan evaluasi"*.

Dalam mendefinisikan suatu sistem, terdapat dua kelompok pendekatan yang menekankan pada prosedurnya dan menekankan pada komponen atau elemennya. Jadi, sistem adalah kumpulan elemen – elemen atau prosedur yang saling berhubungan dan bertanggung jawab memproses data masukan (*input*) sehingga menghasilkan suatu keluaran (*output*). Bagian listing yang paling kecil disebut subsistem. Misalnya sistem komputer yang terdiri dari bagian subsistem perangkat keras dan subsistem perangkat lunak. Masing – masing subsistem dapat terdiri dari subsistem – subsistem atau komponen – komponen yang lebih kecil

lagi. Subsistem – subsistem ini membentuk satu kesatuan sehingga tujuan atau sasaran dari sistem tersebut dapat tercapai. Jadi dengan kata lain, sistem dapat dinyatakan sebagai satu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang saling berinteraksi untuk mencapai suatu tujuan.

Saran dan tujuan sistem adalah untuk menentukan operasi yang akan dilaksanakan. Berkembangnya suatu sistem dapat dipengaruhi oleh informasi yang terdapat didalamnya. Jika berkurangnya informasi, maka suatu sistem lama kelamaan akan berakhir dan tidak dapat digunakan lagi.

2.1.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, anatara lain adalah sebagai berikut :

1. Sistem diklasifikasikan sebagai sistem abstrak (*Abstract System*) dan sistem fisik. Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik adalah sistem yang ada secara fisik.
2. Sistem diklasifikasikan sebagai sistem alamiah (*Natural System*) dan sistem buatan manusia (*Human Made System*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Sistem buatan adalah sistem yang dirancang oleh manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu (*Deterministic System*) dan sistem tak tentu. Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi interaksi diantara bagian-bagian dapat di deteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Sistem tak

tentu adalah sistem yang kondisi masa depannya diprediksi karena mengandung unsur probalitas.

4. Sistem diklasifikasikan sebagai sistem tertutup (*Closed system*) dan sistem terbuka. Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luar. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar.

2.1.3 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, antara lain adalah sebagai berikut :

1. Komponen sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi artinya saling bekerjasama membentuk sistem.

2. Batas sistem

   Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau lingkungan luar.

3. Lingkungan luar sistem

   Lingkungan luar sistem (*enviromment*) dari suatu sistem adalah apapun yang diluar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung sistem

   Penghubung (*interface*) merupakan media penghubung antara satu sub sistem dengan sub sistem yang lain.

5. Masukan sistem

   Masukan sistem (*input*) adalah energi yang dimasukkan ke dalam sistem, masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dpat beroperasi. *Signal input* adalah energi yang diproses untuk di dapatkan keluaran sistem

6. Keluaran sistem

   Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem

   Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah menjadi keluaran.

8. Sasaran sistem

   Suatu sistem pasti mempunyai tujuan (*goal*) dan sasaran (*objective*) suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujuan dari sistem tersebut.

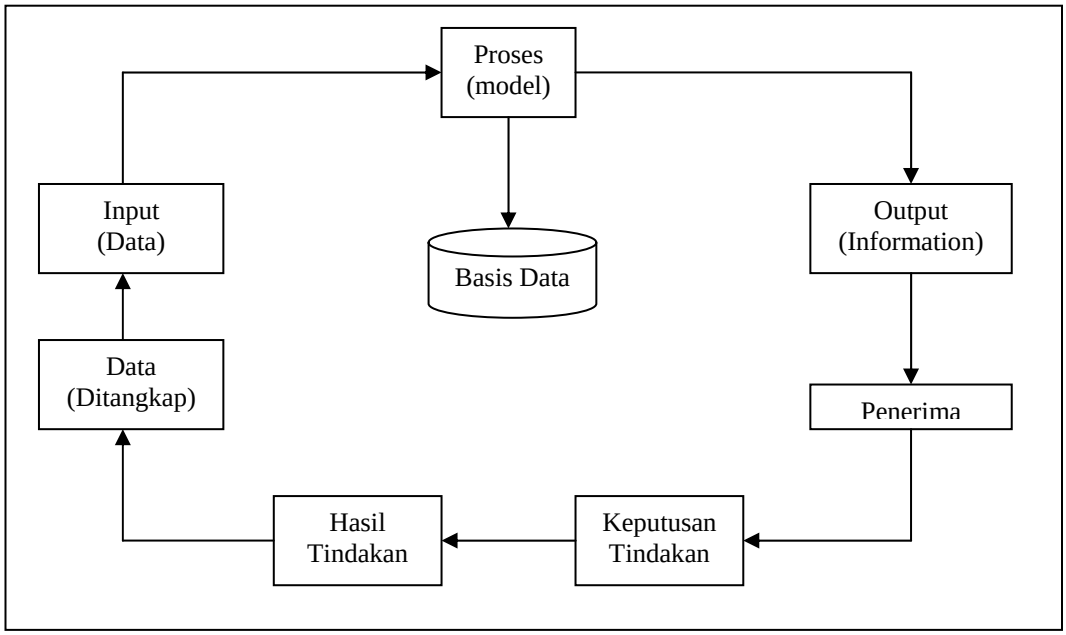
2.1.4 Informasi

Menurut Robert N. Nthony dan John Dearden yang dikutip Jogiyanto HM (1992, 8), pengertian informasi adalah “

“ Informasi adalah data yang diolah menjadi bentuk yang berguna dan lebih berarti bagi yang menerimanya”.

Pengolahan data menjadi informasi memiliki siklus. Siklus tersebut disebut sebagai siklus informasi atau siklus pengolahan data. Gambaran dari

siklus tersebut menurut John Burch yang dikutip dari Jogiyanto HM (1992, 9) adalah sebagai berikut :



Gambar 2.1

Siklus Informasi [Jogiyanto HM 1992, 9]

Informasi yang dihasilkan oleh informasi tesebut dapat dinilai kualitasnya dengan melihat keakuratan, ketepatan waktunya, dan relefansinya. Ketiga kualitas informasi tersebut dikemukakan oleh **John Burch** dan **Gary Grudnitski** yang dikutip dar **Jogiyanto HM (1992, 10)**. **John Burch** dan **Gary Grudnitski** menjelaskan dengan menganggap bahwa informasi merupakan bangunan yang dibangun di atas tiga pilar (penguat), ketiga pilar tersebut yaitu ketiga kualitas informasi tadi.

Data yang telah diproses ke dalam suatu bentuk yang mempunyai arti bagi si penerima dan mempunyai nilai nyata, sehingga dapat dipakai sebagai dasar

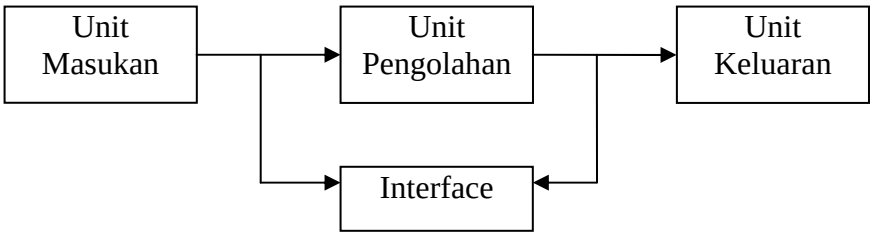
untuk mengambil keputusan, dan terasa bagi keputusan saat itu atau keputusan mendatang.

Untuk memperjelas pengertian dasar informasi dapat dilihat pada siklus informasi di gambar 2.2 berikut :



Gambar 2.2 Model Dasar Sebuah Sistem

Setiap sistem terdiri dari beberapa subsistem dan subsistem terdiri atas beberapa subsistem, masing – masing subsistem saling berkaitan sehingga membentuk jalinan yang disebut interface, dan dapat dilihat pada gambar 2.3 berikut ini :



Gambar 2.3 *Unit* Pengolahan Pusat Sebagai Sistem

Suatu sistem merupakan suatu keseluruhan yang bulat dan utuh, dimana tujuan dari masing – masing bagian yang membentuk sistem akan saling menunjang dan mencapai tujuan dari suatu sistem secara keseluruhan. Berarti

bahwa tujuan yang dicapai dari salah satu bagian tidak dapat mengabaikan pencapaian tujuan dari bagian yang lain.

Definisi umum informasi dalam pemakaian sistem informasi adalah data yang telah diolah menjadi sebuah bentuk yang lebih berguna dan lebih berarti bagi si pemakai dalam mengambil keputusan. Jadi sistem pengolahan informasi yaitu pengolahan data dari bentuk tak berguna menjadi berguna (informasi) bagi penerimanya.

2.1.5 Kualitas Informasi

Kualitas dari suatu informasi tergantung dari tiga hal yaitu informasi harus akurat (*Accurate*), tepat pada waktunya (*Timeliness*), dan relevan (*Relevance*) :

1. Akurat

   Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.

2. Tepat pada waktunya

   Berarti informasi yang datang pada penerima tidak boleh terlambat.

3. Relevan

   Berarti informasi mempumyai manfaat untuk penerimanya.

2.1.6 Sistem Informasi

Informasi merupakan hal yang sangat penting bagi kehidupan sehari-hari dalam pengambilan keputusan. Informasi itu sendiri diperoleh dari sistem informasi (*Information System*).

Sistem informasi adalah sistem buatan manusia yang berisi himpunan yang terintegrasi dari komponen – komponen manual dan komponen – komponen terkomputerisasi yang bertujuan untuk mengumpulkan data, menyimpan data dan menghasilkan informasi untuk pemakai.

Dari definisi di atas dapat dijelaskan bahwa suatu sistem informasi harus mampu mengumpulkan, menganalisa, mengolah dan menyajikan data yang diperoleh menjadi informasi yang berguna, bermanfaat dan mendukung fungsi – fungsi sistem yang ada.

Menurut **Robert A. Leitc** dan **K. Rescoe Davis**, Sistem Informasi didefinisikan sebagai berikut :

"Sistem Informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan informsi harian, mendukung operasi, bersifat material dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan." **(Jogiyanto. HM, 11)**

John Burch dan **Gary Grudnitski** yang dikutip **Jogiyanto HM (1992, 12)**, menyatakan komponen-komponen sistem informasi yang disebut sebagai blok bangunan. Blok bangunan tersebut terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali.

a. Blok masukan merupakan metode-metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-okumen dasar.
b. Blok model adalah kombinasi prosedur, logika dan model matematika yang digunakan untuk memanipulasi masikan dan basis data.

c. Blok keluaran merupakan informasi berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen dan pemakai.
d. Blok teknologi merupakan alat dalam sistem informasi untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirim keluaran, dan membantu mengendalikan sistem secara keseluruhan. Teknologi terdiri dari tiga bagian yaitu *hardware, software, dan brainware*.
e. Blok basis data adalah kumpulan dari data-data yang saling berhubungan satu dengan yang lainnya yang tersimpan dalam *hardware* dan dijalankan dengan *software*. Pengorganisasian basis data dimaksudkan supaya informasi berkualitas dan kapasitas penyimpanannya efisien. Untuk mengakses dan memanipulasi basis data digunakan perangkat lunak yang disebut Database Management System (DBMS).
f. Blok kendali dibangun untuk mengatasi atau mengendalikan hal-hal yang dapat merusak sistem, **(Jogiyanto HM, 1992)**.

2.1.7 Sistem Informasi Manajemen

Sistem Informasi Manajemen adalah sebuah sistem manusia atau mesin yang terpadu untuk menyajikan informasi guna mendukung fungsi operasi manajemen dan pengambilan keputusan dalam suatu organisasi untuk mencapai suatu tujuan yang telah ditetapkan.

Berdasarkan definisi Sistem Informasi Manajemen :

a. Barry E. Cushing

"SIM adalah sekumpulan dari manusia dan sumber daya modal di dalam suatu organisasi yang bertanggung jawab mengumpulkan dan mengolah data untuk menghasilkan informasi yang berguna untuk semua tingkatan manajemen didalam kegiatan perencanaan dan pengendalian." **(Jogiyanto, HM, 12)**

b. Frederick H. Wu

"SIM adalah sekumpulan-sekumpulan dari sifat-sifat yang menyediakan informasi untuk mendukung manajemen." **(Jogiyanto, HM, 14).**

Konsep sistem manuasia dan mesin menyiratkan bahwa sebagian tugas dilakukan oleh manusia dan selebihnya dilakukan oleh mesin.

Pada bagian persoalan, manusia dan mesin membentuk sistem gabungan yang menghasilkan rangkaian interaksi antara mesin dan seorang pemakai (*user*). Dan mesin yang sudah benyak digunakan sebagai alat bantu untuk menyelesaikan penyelesaian manusia adalah mesin komputer. **(Gordon B. davis,8)**

2.2 Model Proses Perangkat Lunak

Aktivitas pertama dalam perencanaan proyek perangkat lunak adalah penentuan ruang lingkup perangkat lunak. Ruang lingkup perangkat lunak menggambarkan fungsi, kinerja, batasan, interface, dan reliabilitas. Funsi-fungsi yang digambarkan dalam statemen ruang lingkup dievaluasi dan dalam banyak

kasus juga disaring untuk memberikan awalan yang lebih detail pada saat estimasi dimulai.

Perencanaan perangkat lunak adalah mengestimasi sumber daya yang dibutuhkan untuk menyelesaikan usaha pengembangan perangkat lunak tersebut.



Gambar 2.4 ***Sumber Daya Pengembangan***

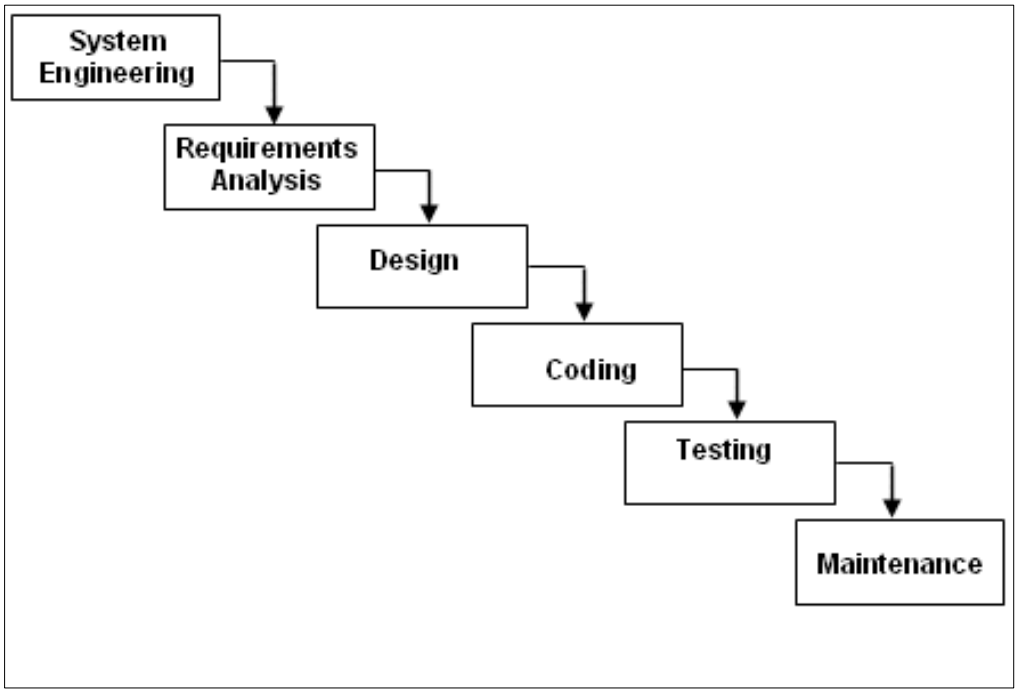
Gambar diatas memperlihatkan sumber daya pengembangan sebagai sebuah piramid.

- Lingkungan pengembangan (piranti perangkat kers dan perangkat lunak) berada pada fondasi piramid sumber daya dan menyediakan infrastruktur untuk mendukung usaha pengembangan.
- Komponen Perangkat Reusable merupakan blok bangunan perangkat lunak yang dapat mengurangi biaya pengembangan secara dramatis dan mempercepat penyampaian.
- Sumber Daya Utama (Manusia) berada pada puncak piramid.

Masing-masing sumber daya ditentukan dengan empat karakteristik yaitu deskripsi sumber daya, statemen ketersediaan, waktu kronologis sumber daya diperlukan , serta durasi waktu sumber daya diaplikasikan.

2.2.1 Metode Waterfall

Metode ini bisa juga disebut dengan classic life cycle. Metode ini membutuhkan pendekatan sistematis dan sekuensial dalam pengembangan perangkat lunak, dimulai dari tingkat sistem dan kemajuan melalui analisis, desain, coding, testing dan pemeliharaan.



Gambar 2.5 *Metode Waterfall*

Pemodelan ini menyangkut aktivitas berikut:

- Rekayasa dan Pemodelan Sistem/Informasi (System/Information Engineering and Modeling). Karena perangkat lunak adalah bagian dari sistem yang lebih besar, pekerjaan dimulai dari pembentukan kebutuhan-

kebutuhan dari semua elemen sistem dan mengalokasikan suatu subset ke dalam pembentukan perangkat lunak. Hal ini penting, ketika perangkat lunak harus berkomunikasi dengan hardware, orang dan basis data. Rekayasa dan pemodelan sistem menekankan pada pengumpulan kebutuhan pada level sistem dengan sedikit perancangan dan analisis. Tahap ini juga kadang disebut dengan Project Definition.

- Analisis Kebutuhan Perangkat Lunak (Software Requirements Analysis). Proses pengumpulan kebutuhan diintensifkan ke perangkat lunak. Harus dapat dibentuk domain informasi, fungsi yang dibutuhkan, performansi dan antarmuka. Hasilnya harus didokumentasikan dan di-review ke pelanggan.
- Desain (Design). Proses desain mengubah kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program. Desain ini harus didokumentasikan dengan baik dan menjadi bagian konfigurasi perangkat lunak.
- Penulisan Program (Coding). Desain tadi harus diubah menjadi bentuk yang dimengerti mesin (komputer). Maka dilakukan langkah penulisan program. Jika desain-nya detil, maka coding dapat dicapai secara mekanis.
- Testing. Setelah kode program selesai dibuat, dan program dapat berjalan, testing dapat dimulai. Testing difokuskan pada logika internal dari perangkat lunak, fungsi eksternal, dan mencari segala kemungkinan kesalahan. Dab memeriksa apakah sesuai dengan hasil yang diinginkan.

- Support/Maintenance. Perangkat lunak setelah diberikan pada pelanggan, mungkin dapat ditemui error ketika dijalankan dilingkungan pelanggan. Atau mungkin pelanggan meminta penambahan fungsi, hal ini menyebabkan faktor maintenance (pemeliharaan) ini menjadi penting dalam penggunaan metode ini. Pemeliharaan ini dapat berpengaruh pada semua langkah yang dilakukan sebelumnya.

Kelebihan :

Metode ini masih lebih baik digunakan walaupun sudah tergolong "kuno", daripada menggunakan pendekatan asal-asalan. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

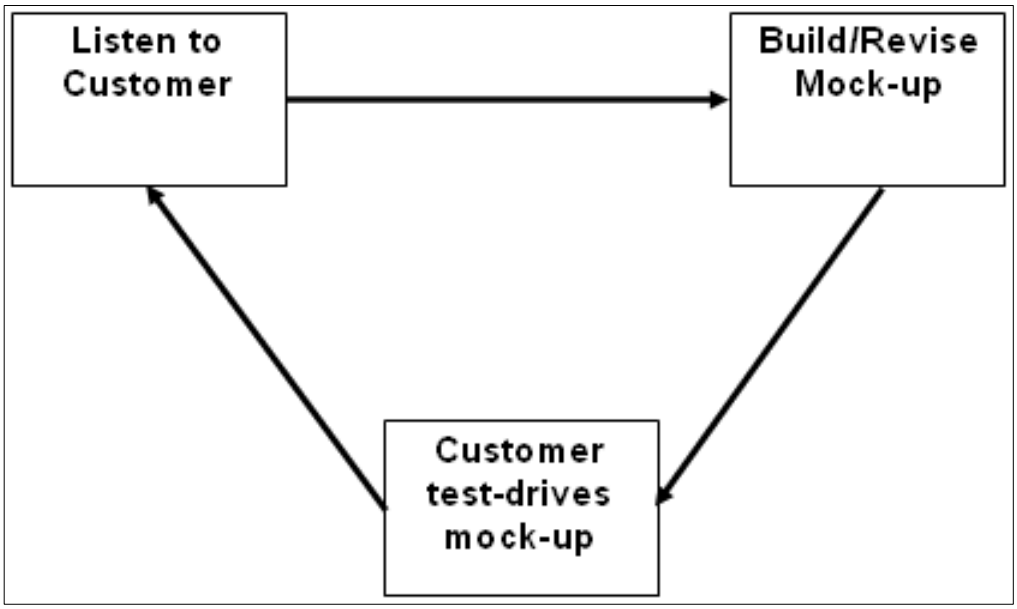
Kekurangan :

- Metode ini jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.
- Sulit bagi pelanggan untuk menentukan semua kebutuhan secara eksplisit.
- Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
- Kesalahan di awal tahap berakibat sangat fatal pada tahap berikutnya.

2.2.2 Metode Prototyping

Prototyping adalah proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk model dari perangkat lunak yang harus dibuat. Model tersebut dapat berupa tiga bentuk:

- Bentuk prototype di atas kertas/model berbasis komputer yang menggambarkan interaksi manusia yang mungkin terjadi.
- Working prototype, yang mengimplementasikan sebagian dari fungsi yang ditawarkan perangkat lunak.
- Program jadi yang melakukan sebagian atau seluruh fungsi yang akan dilakukan, tapi masih ada fitur yang masih dikembangkan.



Gambar 2.6 ***Metode Prototyping***

Urutan kejadian dari metode ini dapat dilihat pada gambar. Seperti pada semua metode, prototyping dimulai dari pengumpulan kebutuhan. Dengan perencanaan yang cepat akan dibentuk konstruksi dari prototipenya. Prototipe ini

dievaluasi oleh pelanggan dan digunakan untuk mengelola kembali kebutuhan dari perangkat lunak yang dikembangkan. Suatu proses iterasi terjadi, setelah prototipe disesuaikan dengan kebutuhan pelanggan, sementara pihak pengembang makin mengerti keinginan pemakai.

Kelebihan :

- Metode ini cukup efektif sebagai paradigma dalam rekayasa perangkat lunak. Kuncinya adalah mendapatkan kebutuhan dan aturan yang jelas yang disetujui pelanggan dan pembuat perangkat lunak. Walaupun pada umumnya prototipe akan dihilangkan dan dibuat perangkat yang sebenarnya.
- Ketika diperlihatkan working version, pelanggan bisa langsung merasakan seakan-akan itu adalah sistem yang sebenarnya.

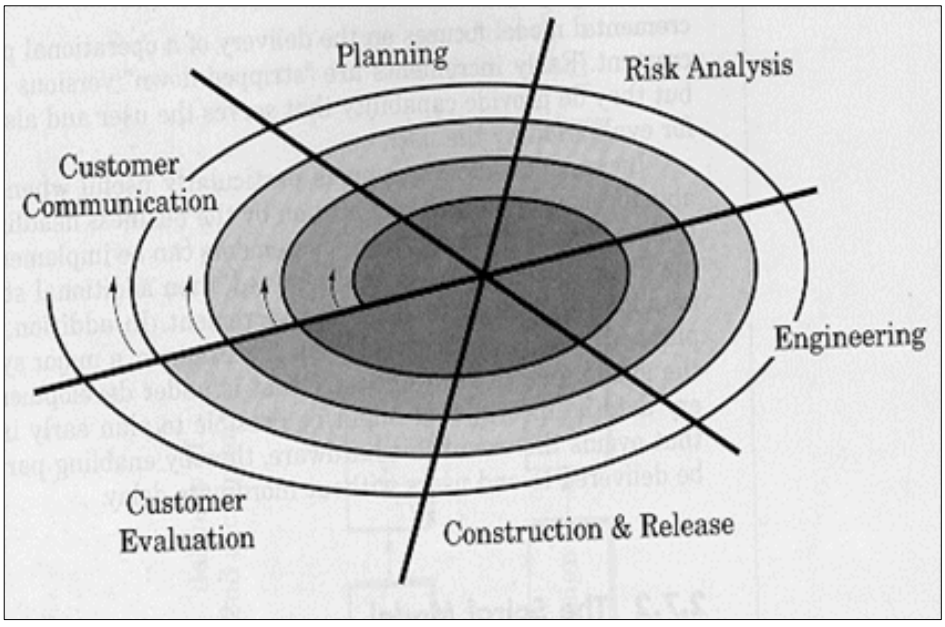
Kekurangan :

- Pelanggan yang melihat working version dari model yang diminta/diperlihatkan tidak menyadari bahwa mungkin saja prototype dibuat terburu-buru dan rancangan tidak tersusun dengan baik.
- Pengembang kadang-kadang membuat implementasi sembarang, karena ingin working version selesai dengan cepat.
- Karena sudah melihat prototype-nya, pelanggan menjadi tidak sabar untuk menunggu versi jadinya. Karena pelanggan merasa program jadinya tidak akan lama lagi selesai.

2.2.3 Metode Spiral

Model ini mengambil fitur penting dari model waterfall dan prototyping, dengan menambah elemen baru yaitu analisa resiko (risk analysis). Model ini memiliki 6 aktivitas penting, yaitu:

- Customer Communication; komunikasi antara pengembang dengan pelanggan.
- Planning; penentuan tujuan, alternatif dan batasan.
- Risk Analysis; analisa alternatif dan identifikasi/pemecahan resiko.
- Engineering; pengembangan level berikutnya dari produk.
- Construction and release; testing, instalasi, dan menyediakan support termasuk dengan training pada user dan pembuatan dokumentasi.
- Customer Evaluation; penilaian terhadap hasil engineering.



Gambar 2.7 ***Metode Spiral***

Bentuk spiral memberikan gambaran bahwa makin iterasinya membesar, maka menunjukkan makin lengkapnya versi dari perangkat lunak yang digunakan. Selama awal sirkuit, objektif, alternatif dan batasan didefinisikan serta resiko diidentifikasi dan dianalisa. Jika analisa resiko menunjukkan ada ketidakpastian terhadap kebutuhan, maka prototyping harus dibuat pada kuadran engineering. Simulasi dan pemodelan lain dapat digunakan untuk mendefinisikan masalah dan memperbaiki kebutuhan.

Pelanggan mengevaluasi hasil engineering (kuadran customer evaluation) dan membuat usulan untuk perbaikan. Berdasarkan masukan dari pelanggan, fase berikutnya adalah planning dan analisis resiko. Setelah analisis resiko, selalu diperiksa apakah proyek diteruskan atau tidak, jika resiko terlalu besar, maka proyek dapat dihentikan.

Kelebihan :

- Model spiral ini adalah pendekatan yang paling realistik untuk sistem skala besar. Metode ini menggunakan pendekatan evolusioner, sehingga pelanggan dan pengembang dapat mengerti dan bereaksi terhadap suatu resiko yang mungkin terjadi. Model ini membutuhkan konsiderasi langsung terhadap resiko teknis, sehingga diharapkan dapat mengurangi terjadinya resiko yang lebih besar. Pada setiap fase evolusi, bisa digunakan prototyping.

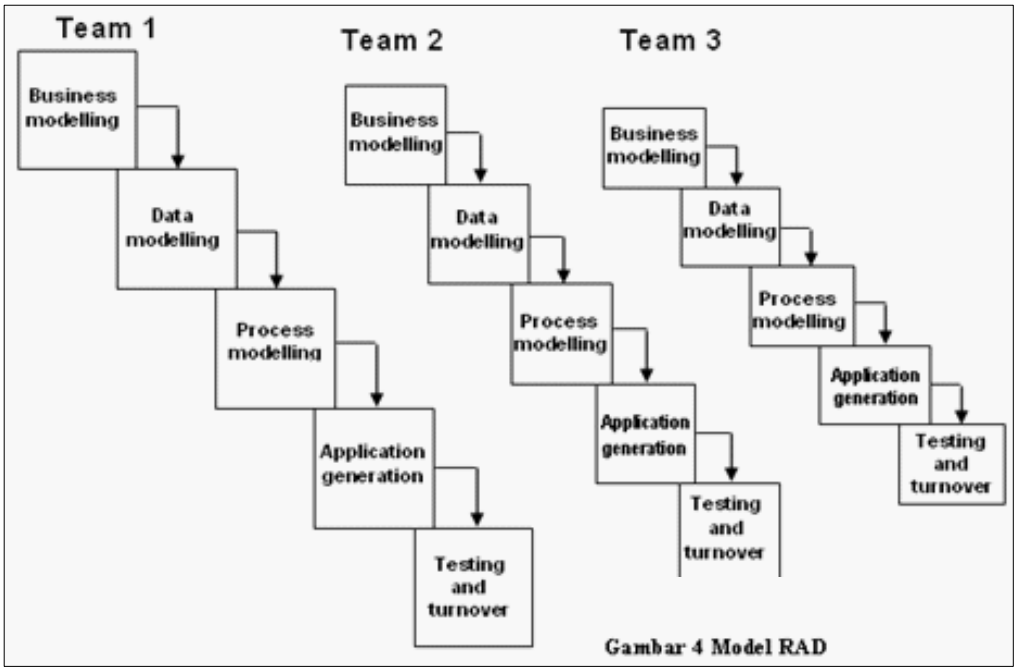
Kekurangan :

- Mungkin akan agak sulit untuk meyakinkan pelanggan besar, bahwa pendekatan evolusioner ini dapat diatur. Hal ini membutuhkan keahlian

tersendiri. Selain itu, jika resiko utama tidak ditemukan, maka masalah bisa muncul kemudian. Sehingga membutuhkan kemampuan manajemen dan perkiraan resiko (risk assessment) yang cukup tinggi.

2.2.4 Metode RAD (Rapid Application Development)

Model RAD merupakan model inkremental dari proses pengembangan perangkat lunak yang menekankan pada sedikitnya siklus pengembangan. Model ini memecah suatu proyek menjadi bagian-bagian kecil yang mana tiap bagiannya dibangun dengan model yang mirip dengan Waterfall. Tujuan utama model ini adalah menyelesaikan suatu proyek per bagian, sehingga proses perencanaannya pun per bagian (walaupun pada awalnya melakukan perencanaan secara global).



Gambar 2.8 ***Metode RAD (Rapid Application Development)***

Model RAD menekankan pada fase-fase berikut:

- Business modeling. Pada tahap ini, aliran informasi (information flow) pada fungsi-fungsi bisnis dimodelkan untuk mengetahui informasi apa yang mengendalikan proses bisnis, informasi apa yang hasilkan, siapa yang membuat informasi itu, kemana saja informasi mengalir, dan siapa yang mengolahnya.
- Data modeling. Aliran informasi yang didefinisikan dari business modeling, disaring lagi agar bisa dijadikan bagian-bagian dari objek data yang dibutuhkan untuk mendukung bisnis tersebut. Karakteristik (atribut) setiap objek ditentukan beserta relasi antar objeknya.
- Process modeling. Objek-objek data yang didefinisikan sebelumnya diubah agar bisa menghasilkan aliran informasi untuk diimplementasikan menjadi fungsi bisnis. Pengolahan deskripsi dibuat untuk menambah, merubah, menghapus, atau mengambil kembali objek data.
- Application generation. RAD bekerja dengan menggunakan fourth generation techniques (4GT). Sehingga pada tahap ini sangat jarang digunakan pemrograman konvensional menggunakan bahasa pemrograman generasi ketiga (third generation programming languages), tetapi lebih ditekankan pada reuse komponen-komponen (jika ada) atau membuat komponen baru (jika perlu). Dalam semua kasus, alat bantu untuk otomatisasi digunakan untuk memfasilitasi pembuatan perangkat lunak.

- Testing and turnover. Karena menekankan pada penggunaan kembali komponen yang telah ada (reuse), sebagian komponen-komponen tersebut sudah diuji sebelumnya. Sehingga mengurangi waktu testing secara keseluruhan. Kecuali untuk komponen–komponen baru.

Kelebihan :

- RAD memang lebih cepat dari waterfall jika kebutuhan dan batasan proyek sudah diketahui dengan baik. Juga jika proyek memungkinkan untuk dimodularisasi.

Kekurangan :

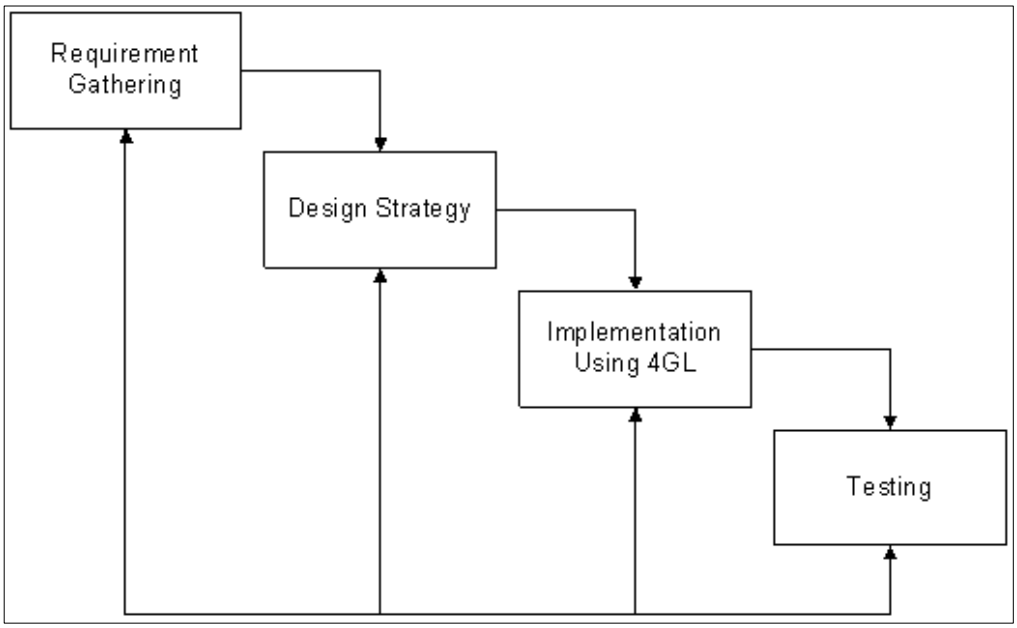
- Tidak semua proyek bisa dipecah (dimodularisasi), sehingga belum tentu RAD dipakai pada semua proyek.
- Karena proyek dipecah menjadi beberapa bagian, maka dibutuhkan banyak orang untuk membentuk suatu tim yang mengerjakan tiap bagian tersebut.
- Membutuhkan komitmen antara pihak pengembang dan pelanggan.
- Karena dibuat dengan reuse komponen-komponen yang sudah ada, fasilitas-fasilitas pada tiap komponen belum tentu digunakan seluruhnya oleh program yang me-reuse-nya sehingga kualitas program bisa menurun.

2.2.5 Metode Fourth-Generation Techniques (4GT)

Istilah Fourth-Generation (generasi keempat) mengarah ke perangkat lunak yang umum yaitu, tiap pengembang perangkat lunak menentukan beberapa karakteristik perangkat lunak pada level yang tinggi. Tool akan otomatis menghasilkan sumber berdasarkan spesifikasi tersebut. Teknik 4GT ini

menekankan pada kemampuan menentukan perangkat lunak pada level mesin dengan bahasa yang lebih alami atau notasi yang lebih memiliki arti.

Saat ini, pengembangan perangkat lunak yang mendukung 4GT, berisi tool-tool berikut: Bahasa non-prosedural untuk query basis data, report generation, data manipulation, interaksi layar dan definisi, code generation, kemampuan grafik level tinggi, kemampuan spreadsheet, dan automated generation of HTML serta bahasa-bahasa yang mirip yang digunakan untuk membuat suatu website dengan menggunakan perangkat lunak tertentu. Tiap tool ini ada, tapi hanya untuk aplikasi khusus.



Gambar 2.9 ***Metode Fourth-Generation Techniques (4GT)***

Metode 4GT ini dimulai dari pengumpulan kebutuhan. Idealnya pelanggan akan menjelaskan kebutuhannya, yang akan langsung ditranslasikan ke prototipe operasional. Tapi, prototipe ini tidak bekerja. Pelanggan mungkin tidak pasti akan hal yang dibutuhkannya atau tidak dapat menentukan informasi yang

dapat ditangani tool 4GT. Tool 4GT yang sudah ada tidak cukup canggih untuk mengakomodasikan bahasa alami. Pada saat ini, dialog antara pelanggan dan pengembang yang ada pada metode sebelumnya tetap menjadi bagian penting dari teknik 4GT.

Implementasi menggunakan 4GL (Fourth-Generation Language) dapat dihasilkan dari program kode yang sesuai. Tetapi struktur data dengan informasi lainnya harus ada dan dapat diakses oleh 4GL. Untuk aplikasi kecil, adalah mungkin untuk langsung berpindah dari pengumpulan kebutuhan ke implementasi menggunakan bahasa non-prosedural (Fourth-Generation Language - 4GL).

Kelebihan:

- Karena 4GT menggunakan 4GL yang “notabene” merupakan bahasa pemrograman yang khusus dirancang dengan tujuan tertentu (spesifik), maka untuk permasalahan yang tertentu dengan 4GL tertentu pula sangat tepat menggunakan 4GT. Bahkan ada 4GL yang bisa meng-generate sistem dari output yang dihasilkan oleh CASE tools.

Kekurangan:

- Untuk usaha yang besar, dibutuhkan pengembangan strategi desain untuk sistem, walau digunakan bahasa 4GL. Penggunaan 4GT tanpa perencanaan matang (untuk proyek besar) akan menyebabkan kesulitan yang sama (kualitas dan pemeliharaan yang jelek, ketidakpuasan pelanggan) seperti dengan metode konvensional.
- 4GL tidak selalu berhasil menghasilkan sistem yang diinginkan.

4GL merupakan bahasa pemrograman yang notasinya lebih mendekati pada bahasa yang dimengerti manusia daripada high-level programming language. Semua 4GL dirancang untuk mengurangi aktivitas pemrograman, waktu dan biaya pengembangan perangkat lunak.

Pada umumnya 4GL banyak digunakan untuk mengakses database. Contoh bahasa pemrograman yang banyak dipakai adalah:

- Database query languages: SQL, Oracle SQL*Plus, Progress 4GL.
- Report Generators: Oracle Reports, LINC, GEMBase, BuildProfessional, Informix-4GL, Focus, Metafont, RPG-II, S, IDL-PV/WAVE, Gauss, Mathematica.
- Data-stream languages: AVS, APE, Iris Explorer.
- Screen painters and generators: Oracle Forms
- GUI creators: Visual Basic, MATLAB's GUIDE

Generasi bahasa komputer lainnya adalah sebagai berikut:

- First Generation: Bahasa Mesin
- Second Generation: Bahasa Assembly
- Third Generation: High-level programming languages, seperti Pascal, C, C++, dan Java.
- Fifth Generation: Digunakan untuk artificial intelligence dan neural networks, seperti Prolog, OPS5 dan Mercury.

2.3 Metode Pendekatan Sistem

Metode pendekatan sistem menggunakan analisis dan perancangan testruktur, adapun tahapan-tahapan yang ada di dalam perncangan suatu sistem adalah :

1. Perancangan proses (flow map, DFD, kamus data)
2. Perancangan basis data (ERD, normalisasi, tabel relasi, struktur file)
3. Perancangan program (perancangan input/output, struktur program, kebutuhan sistem)

2.4 Alat dan Teknik Perancangan Sistem

Untuk mempermudah dalam melakukan langkah langkah yang sesuai dengan yang diberikan dalam metodologi pengembangan sistem yang terstruktur diperlukan suatu alat dan teknik untuk melaksanakan perancangan sistem. Alat-alat yang digunakan umumnya berbentuk gambar, digram, atau grafik. Alat-alat yang digunakan untuk analisis dan perancangan adalah sebagai berikut :

2.4.1 *Flow Map*

Merupakan suatu diagram yang menggambarkan sistem yang di dalamnya terdapat subsistem-subsistem. Di dalam subsistem-subsistem tersebut terdapat dokumen-dokumen yang mengalir yang menghubungkan antara subsistem- subsistem yang ada pada sistem tersebut.

2.4.2 *Contex Diagram*

Contex diagram adalah diagram tingkat atas yang tidak detail dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar *entitas eksternal*. Dalah hal ini aliran data yang keluar disebut *terminator*, dapat berupa sistem lain, suatu perangkat keras, orang atau organisasi.

2.4.3 *Data Flow Diagram*

Merupakan diagram yang di presentasikan dengan lambang-lambang tertentu. Dan suatu sistem yang menunjukan proses, aliran data, entitas dan aliran data yang terdapat pada sistem tersebut.

Beberapa simbol yang digunakan di DFD antara lain:

a. Kesatuan Luar (*Entity Eksternal*)

Setiap sistem pasti mempunyai batasan sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima iput dan menghasilkan output kepada lingkungan luarnya. Kesatuan luar (*entity eksternal*) merupakan kesatuan dilingkungan luar sistem, dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan input atau menerima output dari sistem. Kesatuan luar ini kebanyakan adalah salah satu dari yang berikut ini:

1. Suatu kantor, departemen atau divisi dalam perusahaan tetepai diluar sistem yang sedang dikembangkan.
2. Orang atau sekelompok orang di organisasi tetapi diluar sistem yang sedang dikembangkan.

Kesatuan luar digambarkan dengan simbol persegi panjang.

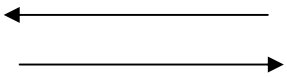


Gambar 2.10 External Entity

b. Aliran Data (*Data Flow*)

Aliran data si DFD diberikan symbol suatu panah. Aliran data ini mengalir diantara proses (*process*), simpanan data (*data storage*) dan kesatuan luar (*entity eksternal*). Aliran data ini menunjukan aliran dari data yang dapat berupa masukan dri sistem atau hasil dari proses sistem.

Arus data digambarkan dengan simbol tanda panah.

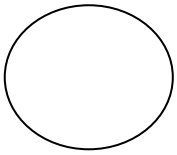


Gambar 2.11 Data Flow

c. Proses (*Process*)

Proses adalah kegiatan yang dilakukan oleh orang, organisasi, mesin atau computer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses yang digambarkan secara umum. Suatu proses dapat ditunjukan dengan simbol lingkaran atau denagn symbol empat persegi panjang tegak dengan sudut- sudutnya yang tumpul.

Proses digambarkan dengan simbol lingkaran.



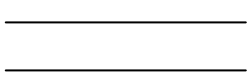
Gambar 2.12 Proses

d. Berkas / Simpanan Data (*Data Storage*)

Berkas atau simpanan data merupakan simpanan dari data yang dapat berupa:

1. Suatu file atau database disistem computer.
2. Suatu arsip atau catatan manual.
3. Suatu kotak tempat data di meja seseorang.
4. Suatu table acuan manual.
5. Suatu agenda-agenda atau buku.

Digambarkan dengan simbol dua garis horizontal.



Gambar 2.13 File.

2.4.4 Kamus Data

Merupakan katalog fakta tentang data dalam kebutuhan-kebutuhan informasi dari suatu sistem informasi. Arus data dibuat berdasarkan arus data dari Data Flow Diagram (DFD).

2.5 Basis Data

Database meupakan suatu koleksi data yang terintegrasi, diorganisasikan dan disimpan dalam suatu cara yang memudahkan pengambilan kembali, banyak file dapat terintegrasi secara logis dan itu merupakan konsep *database.*

Database adalah kumpulan data yang disimpan dalam suatu file atau beberapa file atau secara operasiaonal database adalah daftar yang terdiri dari beberapa kolom yang masing-masing kolom berisikan satu jenis (item) data.

Dalam konsep database data memiliki tingkatan yang dikenal sebagai heirarki data yang terdiri dari ;

1. *Bit* adalah unit terkecil dari data yang menggambarkan sinyal 0 dan 1 dimana 0 artinya tidak memiliki arus dan 1 artinya memiliki arus
2. *Byte* adalah kumpulan bit yang membentuk sebuah karakter
3. *Field* atau elemen data adalah kumpulan karakter yang membentuk suatu kata atau sekelompok kata atau angka
4. *Record* merupakan kompulah dari field yang saling berhubungan.
5. *Fie* kumpulan record yang berhubungan dengan suatu subjek tertentu
6. *Database* merupakan kumpulan dari data yang tersimpan dalam file-file

2.5.1 Sistem Pengelola Basis Data*(Database Manajemen System/DBMS)*

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus/spesifik. Perangkat lunak inilah (sebagai DBMS) yang akan menentukan bagaimana data diorganisasikan, disimpan, diubah dan diambil kembali. Ia juga

menerapkan mekanisme pengamanan data, pemakaian data secara bersama, keakuratan data dan sebagainya.

Perangkat lunak yang termasuk DBMS seperti dBase III+, dBase IV, FoxBase, Ms-Access, Borland Paradox, Ms-SQL Server, Oracle Borland-interbase dan lain sebagainya. Salah satu tujuan DBMS adalah untuk menyediakan fasilitas/antar muka (*interface*) dalam melihat data (yang lebih ramah atau *user friendly*) kepada pemakai.

2.5.2 Langkah-Langkah Desain Basis Data *(Database)*

Untuk tahap desain database secara umum, yang perlu dilakukan oleh analisis sistem adalah mengidentifikasikan terlebih dahulu file-file yg diperlukan oleh sistem informasi, langkah-langkahnya sebagai berikut :

1. Menentukan kebutuhan file database untuk sistem yang baru. File yang dibutuhkan dapat ditentukan dari diagram arus data dari sistem baru yang telah dibuat.
2. Menentukan parameter dari file database setelah file-file dibutuhkan telah dapat ditentukan, maka parameter dari file selanjutnya juga dapat ditentukan. Parameter ini meliputi :
    a. Tipe dari file
    b. Organisasi dari file
    c. Field kunci dari file

2.5.3 Kegunaan Basis Data *(Database)*

Penyusunan suatu database digunakan untuk mengatasi masalah-masalah pada penyusunan data, yaitu :

a. Redudansi dan Inkonsistensi data
b. Kesulitan pengaksesan data
c. Isolasi data untuk standarisasi
d. Multiple user (*banyak pemakai*)
e. Masalah keamanan (*Security*)
f. Masalah Integrasi (*kesatuan*)
g. Masalah dan Independence (*Kebebasan Data*)

2.5.4 Bahasa Basis Data *(Database)*

DBMS merupakan perantara bagi pemakai, dengan basis data dalam disk. Cara berinteraksi/berkomunikasi antar pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang diterapkan oleh perusahaan pembuat DBMS. Bahasa ini dapat kita sebut sebagai bahasa basis data yang terdiri dari sejumlah perintah yang diformulasikan dan dapat diberikan user serta dapat dikenali/diproses oleh DBMS untuk melakukan suatu aksi atau pekerjaan tertentu. Sebuah bahasa basis data dapat dipilah dalam 2 bentuk yaitu :

1. *Data Definition Language* (DDL)
2. *Data Manipulation Language* (DML)

Struktur atau skema basis data yang menggambarkan desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yang disebut *Data Definition Language* (DDL). Dengan bahasa inilah dapat dibuat tabel baru, membuat indeks, merubah table, menentukan struktur penyimpanan table dan sebagainya, yang mana hasil dari kompilasi perintah *Data Definition Language*

(DDL) adalah kumpulan table yang disimpan dalam file khusus yang disebut kamus data (*Data Dictionary*).

Sedangkan *Data Manipulation Language* (DML) merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa:

1. Penyisipan atau penambahan data baru dari suatu basis data.
2. Penghapusan data dari suatu basis data.s
3. Pengubahan dari suatu basis data.

Data Manipulation Language (DML) merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data.

2.5.5 Perancangan Basis Data

Dalam perancangan basis data yang harus dilakukan yaitu :

1. *Entity Relationship Diagram* (ERD)

   ERD merupakan notasi grafik dalam pemodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relative kompleks. Dengan ERD kita dapat menguji model dengan menjabarkan proses yang harus dilakukan. ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada 3 macam simbol yang digunakan yaitu :

a. Entitas

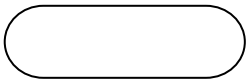
Merupakan suatu objek yang dapat diidentifikasikan dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat



Gambar 2.14 Entity

b. Atribut

Entitas mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakter entitas.



Gambar 2.15 Atribut

c. Relationship

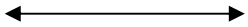
Relationship sebagaimana halnya entitas maka dalam hubungan pun harus dibedakan antar hubungan atau bentuk hubungan antar entitas dengan isi dari hubungan itu sendiri.

Relasi antar 2 tabel atau 2 file dapat dikategorikan/dibedakan menjadi 3 macam yaitu :

- *One To One Relationship*

  Yang berarti, entitas pada himpunan entitas A berhubungan paling banyak dengan 1 entitas B, dan begitu juga sebaliknya setiap entitas pada

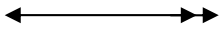
himpunan B berhubungan paling banyak 1 dengan entitas pada himpunan entitas B.



Gambar 2.16 One to one relationship

- *One To Many Relationship*

  Yang berarti, entitas pada himpunan entitas A berhubungan dengan banyak pada himpunan entitas B, tetapi tidak sebaliknya setiap entitas pada himpunan entitas B, hubungan paling banyak dengan satu entitas pada himpunan entitas A.



Gambar 2.17 One to many relationship

- *Many To Many Relationship*

  Yang berarti, entitas pada himpunan entitas A berhubungan dengan banyak pada himpunan entitas B dan begitu juga sebaliknya, setiap entitas pada himpunan entitas B berhubungan banyak dengan himpunan entitas B.



Gambar 2.18 Many to many relationship

2. Normalisasi

   Normalisasi merupakan proses yang menggunakan pendekatan formal untuk menelaah dan kemudian mengelompokan data item / field / atribut ke bentuk yang lebih baik dalam menghadapi perubahan-perubahan bisnis dimasa mendatang serta meminimumkan pengaruh perubahan pada sistem aplikasi atau program. Ada beberapa bentuk nomalisasi, yaitu :

1. Bentuk Normal Pertama

   Aturan bentuk normal I sebagai relasi yang tidak boleh mengandung kelompok berulang (*Repeating Group*).

2. Bentuk Normal Kedua

   Aturan normalisasi kedua berbunyi bahwa setiap field yang tidak bergantung sepenuhnya (fungsionil) pada kunci primer harus dipindahkan ke tabel lain. Bergantung secara fungsionil adalah sebuah istilah matematika, yang berarti bahwa sebuah field sangat ditentukan oleh kunci.

3. Bentuk Normal Ketiga

   Aturan normalisasi ketiga berbunyi bahwa tidak boleh ada kebergantungan antara field field non kunci.

4. Bentuk Normal Keempat

   Suatu relasi disebut dalam bentuk normal IV, jika relasi tersebut sudah dalam bentuk normal III dan seluruh attribut yang bukan primary key tidak tergantung bernilai banyak (*Multivalued Dependencies*) kepada primary key-nya.

2.6 Sekilas Microsoft Visual Basic 6.0

Perangkat lunak yang digunakan untuk membuat implementasi dari sistem pada tugas akhir ini adalah Bahasa Pemograman *Visual Basic 6.0*. Bahasa Pemograman *Visual Basic 6.0* mrupakan Bahasa Pemograman yang digunakan untuk membuat aplikasi *Windows* yang berisi grafis (*GUI – Graphical User*

Interface), atau pemograman yang menggunakan tampilan grafis sebagai alat komunikasi dengan pemakainya.

Sejarah pekembangan *Visual Basic* adalah sebagai berikut :

a. *Visual Basic* pertama kali diperkenelkan pada tahun 1991 yaitu program *basic* untuk *Dos* dan *Windows*
b. *Visual Basic 3.0* dirilis tahun 1993
c. *Visual Basic 4.0* dirilis pada akhir tahun 1995
d. *Visual Basic* yang digunakan untuk membuat aplikasi adalah versi 6.0 yang dirilis pada akhir tahun 1998.

Pada *Visual Basic* , langkah – langkah untuk mengembangkan aplikasi :

1. Membuat *user interface*/tampilan
2. Mengatur properti
3. Menulis kode program

Lingkungan pemograman *Visual Basic 6.0* mengandung sarana pemograman yang dapat membantu Anda dalam membuat program *Visual Basic* diantaranya :

a. Baris Menu

   Baris menu menyediakan akses kepada sebagian besar perintah yang mengendalikan lingkungan pemograman ini. Menu dan perintah sama fungsinya seperti pada pemograman – pemograman berbasis *windows* lainnya, Anda dapat mengaksesnya dengan menggunakan *keyboard* atau mouse.

b. *Toolbar*

Toolbar yaitu sekumpulan tombol yang berfungsi sebagai tombol cepat untuk menjalankan perintah dan mengendalikan lingkungan pemograman *Visual Basic.*

c. *Taskbar windows*

Taskbar Windows digunakan untuk berpindah antara berbagai komponen Visual Basic dan untuk mengaktifkan program Windows lainnya.

d. Toolbox

Toolbox biasanya terletak di sisi sebelah kiri layar. *Toolbox* mengandung kontrol – kontrol yang biasa digunakan untuk menambah gambar, label, tombol, kotak daftar, baris pengulung, menu, serta bentuk – bentuk geometris antar muka.

e. *Project Windows*

Jendela *Project* biasanya terletak di sisi sebelah kiri atas layar. Jendela *Project* berisi semua file yang digunakan pada proses pemograman dan menyediakan akses ke file tersebut menggunakan dua tombol khusus : *View Code* dan *View Object.*

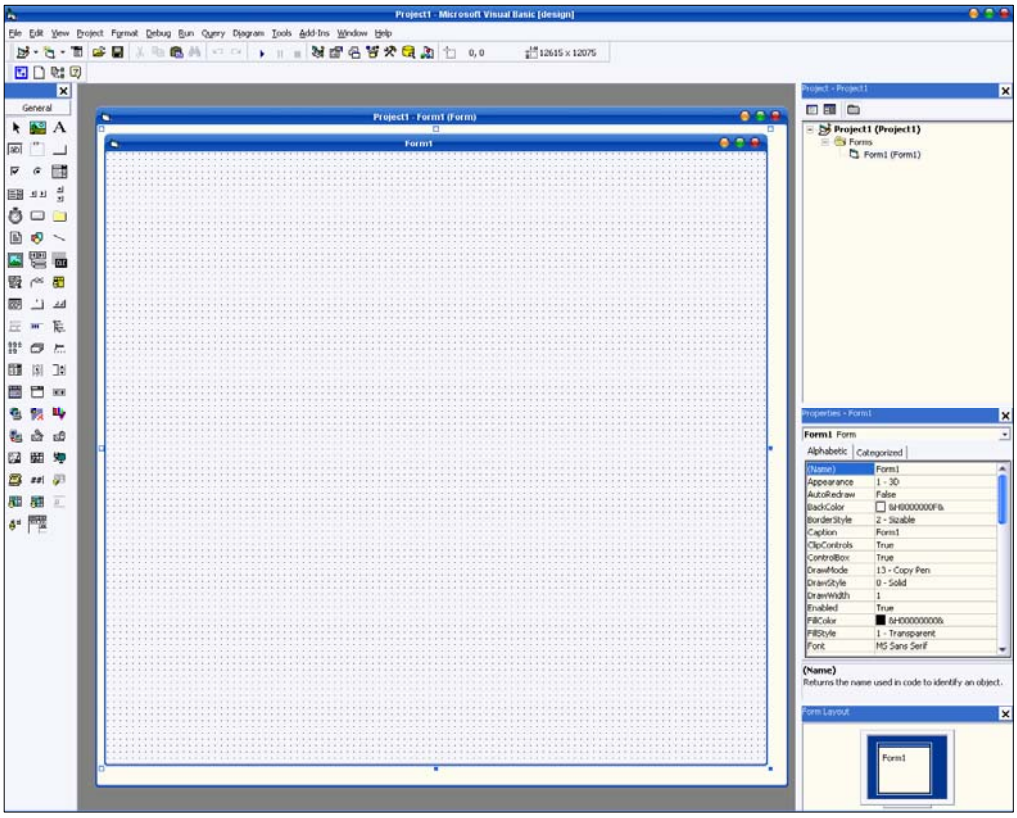
f. *Propertis Windows*

Jendela Propetis biasanya terletak di sisi kanan tengah layar. Jendela propertis digunakan untuk mengubah karakteristik atau setting propertis dari elemen yang terdapat pada form. Setting propertis adalah kualitas objek – objek yang terdapat pada antarmuka.

g. *Form Layout Windows*

Jendela *Form Layout* biasanya terletak di sisi kanan bawah layar. Jendela *form layout* digunakan untuk melihat jumlah Object dalam layar yang kita tampilkan, juga kita dapat menampilkan garis relasi dengn memilih *Resolution Guides*.

Mengenal tampilan Visual Basic 6.0 :



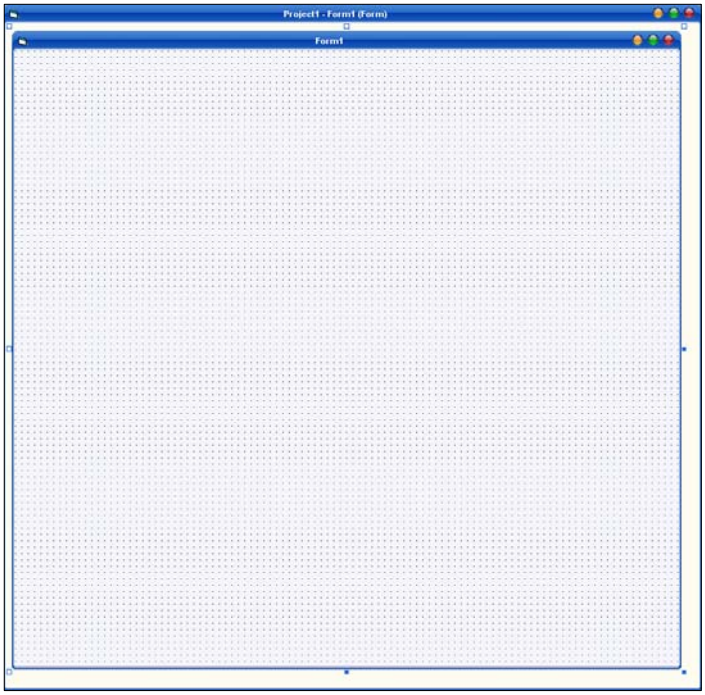
Gambar 2.19 Tampian Awal Visual Basic 6.0

1. *Main Window* (jendela umum) terdiri dari *title bar* (baris judul), *menubar* dan *toolbar*.



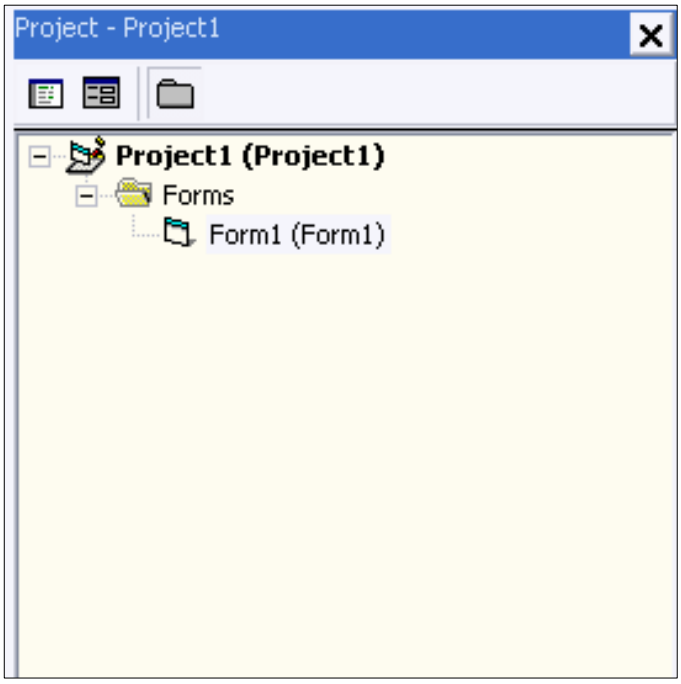
Gambar 2.20 *Main Window, Menubar* dan *Toolbar*

2. *Form Window* (jendela form) adalah pusat dari pengembangan aplikasi visual basic.



Gambar 2.21 *Form Window* (Jendela Form)

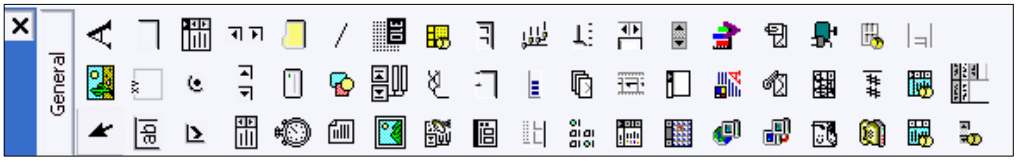
3. *Project Window* (*jendela proyek*) menampilkan daftar *form* dan modul proyek. Proyek merupakan kumpulan dari modul *form*, modul *class*, modul standart, dan *file* yang membentuk suatu aplikasi.



Gambar 2.22 *Project Window* (Jendela Proyek)

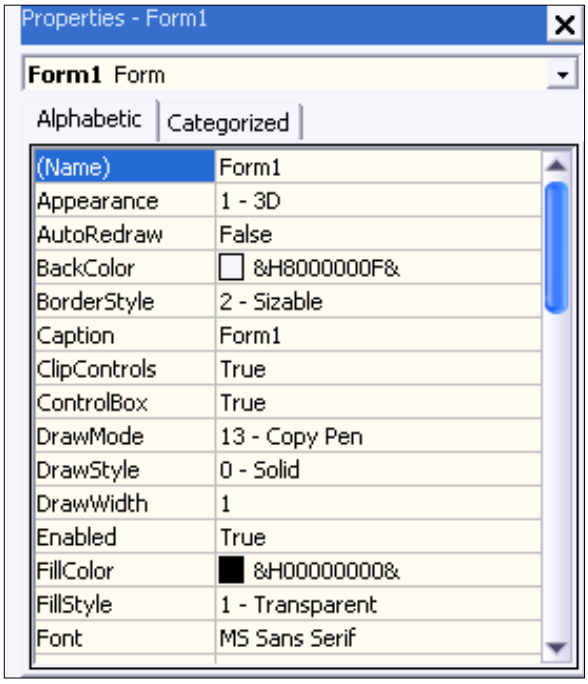
4. *Toolbox*

   Kumpulan dari objek yang digunakan untuk membuat *user interface* serta kontrol bagi program aplikasi.



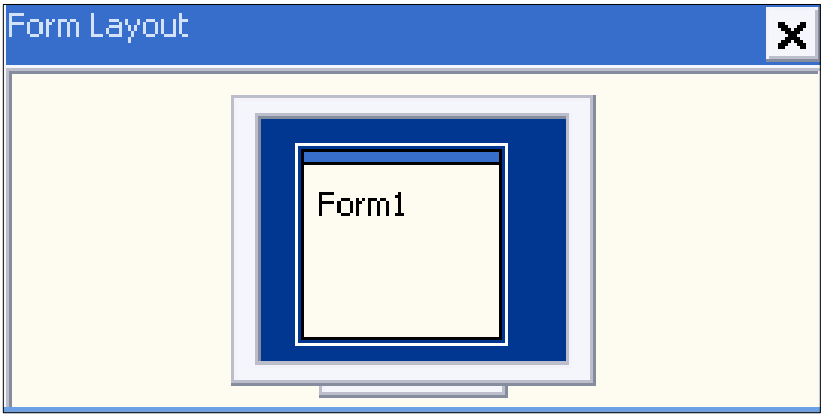
Gambar 2.23 *Toolbox*

5. *Properties Window* (jendela property) berisi daftar struktur seting *property* yang digunakan pada sebuah abjek terpilih.



Gambar 2.24 *Properties Windows* (Jendela Properti)

6. *Form Layout Window* (jendela layout Form) menampilkan posisi relatif terhadap layar monitor.



Gambar 2.25 *Layout Window* (Jendela Layout)

2.6.1 Mengenal *Integrated Development Environment (IDE) VB 6.0*

Bagian-bagian utama di dalam IDE VB 6 berikut ini :

1. Keterangan
2. Menubar
3. Toolbar
4. Toolbox
5. Jendela Form
6. Jendela Code
7. Project Explorer

2.6.2 Elemet Visual Basic 6.0

2.6.2.1 Proyek

Dalam Visual Basic, pembuatan sebuah program aplikasi harus dikerjakan dalam sebuah proyek. Proyek tersebut berisi kumpulan file-file yang digunakan untuk membuat aplikasi.

Sebuah proyek dapat terdiri dari satu file proyek (.vbp), satu file *form* (.frm), satu file data *binary* untuk setiap *form* (.frx), satu file untuk setiap modul *class* (.cls), satu file untuk setiap modul standar (.bas), atau file *resource* tunggal (.res).

Selain modul dan file, beberapa tipe komponen dapat juga dimasukkan kedalam proyek, seperti satu/lebih file yang terdiri dari kontrol ActiveX (.ocx), Insertable Object seperti objek Worksheet Exel, Reference yang ditambahkan ke eksternal ActiveX, ActiveX Designer untuk merancang class pada objek, serta Standard Control seperti Command Button.

Untuk membuat sebuah program aplikasi, kita terlebih dahulu harus membuat *form*, baru kemudian membuat file dan modul lain. Setelah seluruh komponen dipadukan dan kode telah ditulis, langkah selanjutnya adalah membuat proyek kita menjadi file yang bisa dieksekusi.

2.6.2.2 Menu

Visual Basic menyediakan sarana untuk membuat menu tanpa terlebih dahulu membuat dasar programnya.

Pembuatan menu mutlak diperlukan, mengingat program yang sudah siap pakai bisa menjadi lebih efektif dan praktis. Selain itu, bentuk menu yang cantik bisa menjadi daya tarik bagi konsumen program aplikasi komputer.

2.6.2.3 Prosedur

Dalam membuat prosedur harus diperhatikan beberapa aspek yang menjadi sifat setiap jenis prosedur.

Adapun jenis-jenis prosedur, yaitu:

- Sub Procedure
- Function Procedure

2.6.2.4 Objek/Kontrol Standar

Objek/kontrol merupakan objek hubungan dengan pemakai dimana tujuanna untuk interaksi antara aplikasi/program dengan pemakai. Objek ini dapat berupa suatu tools yang digunakan untuk masukkan maupun keluaran, atau hanya sebagai pesan yang ditampilkan kelayar.

Setiap objek/kontrol memiliki properti seperti warna atau tulisan yang diletakkan pada objek/kontrol tersebut.

2.6.2.5 Kontrol Standar Visual Basic 6.0

Kontrol-kontrol diletakkan pada windows toolbox untuk dipilih dan digambarkan pada *form.* Berikut adalah beberapa kontrol standar yang telah disediakan oleh visual basic.

1. Kontrol Label

   Kontrol Label digunakan untuk menampilkan teks/tulisan pada *form* dan tidak dapat diedit oleh user, teks/tulisan ini dapat ditulis langsung pada saat perancangan atau saat aplikasi dijalankan.

2. Kontrol Command Button

   Kontrol Command Button atau disebut kontrol button dapat menyebabkan tejadinya suatu tindakan berupa perintah dieksekusi ketika kontrol tersebut diklik.

3. Kontrol Textbox

   Kontrol Textbox digunakan untuk menampilkan tekxt/tulisan yang diketikkan oleh user saat *run time* atau untuk menampilkan teks yang ditunjuk oleh properti teks saat *design time.*

4. Kontrol Option Button / Radio Button

   Kontrol Option Button / Radio Button digunakan untuk melakukan pemilihan yang hanya boleh memilih salah satu pilihan saja. Bagi sebagian programmer juga disebut dengan radio button. Keadaan radio button dapt diketahui dengan menggunakan properti *value,* yaitu nilai *true* bila control dalam keadaan terpilih (ditandai dengan lingkaran hitam didalam control) dan *false* juka tidak terpilih (lingkran hitam dihilangkan).

Pada sebuah *form*, objek Option Button dianggap sebuah group, artinya pemilihan sebuah onjek Option Button pada sebuah *form* akan mematikan objek Option Button lainnya pada *form* yang sama.

Untuk mengelompokkan objek Option Button diperlukan sebuah control yang tujuannya mengelompokkan objek/control. Kontrol tersebut diberi nama *frame.* Kontrol ini harus digambarkan terl;ebih dahulu, dilanjutkan dengan menggambarkan beberapa kontrol Option Button atau control lainnya tepat didalam frame tersebut. Proses ini akan secara otomatis mengelompokkan kontrol-kontrol yang berada didalamnya.

5. Kontrol Check Box

   Kontrol Check Box digunakan untuk melakukan pemilihan lebih dari satu pilihan. Pemrogramannya tidak berbeda dengan Option Button, hanya ketika sebuah control tipe ini diklik atau properti *value* dirubah ke *true* tidak akan menyebabkan control lainnya tipe ini berubah nilai properti *value*-nya.

6. Kontrol List Box

   Kontrol List Box digunakan untuk menampilkan daftar pilihan, dimana *user* dapat memilih pilhan dengan cara mengklik pada teks/item yang diinginkan. Jika jumlah pilihan terlalu banyak maka tidak semua item ditampilkan, tetapi dapat ditmpilkan dengan melakukan *srolling* terhadap Listbox. Properti *stored* dapat digunakan untuk meletakkan item dalam keadaan terurut bila diatur nilai *true*.

7. Kontrol Horizontal dan Vertikal Scrollbar

   Kedua kontrol ini menghsilkan posisi kotak gulung dalam proyek baris gulung. Kita dapat menaikkan posisi kotak gulung.

8. Kontrol Image

   Kontrol Image memungkinkan kta menempatkan gambar dilokasi tertentu pada sebuah *form*. Format gambar yang dapat ditampilkan antara lain: *bitmap, icon, metafile, enchanced metafile, JPEG* atau *GIF*.

9. Kontrol Shape

   Kontrol Shape digunakan untuk membuat objek *Shape* berupa *rectangle, square, oval, circle, rounded square.*

10. Kontrol Picture Box

    Kontrol Picture Box digunakan untuk menampilkan gambar yang berasal dari *imap, icon, metafile, enchanced metafile, JPEG* atau *GIF.*

2.7 Sekilas Mengenai *Microsoft Access*

Bila kita mempunyai daftar berisi nama dan alamat mahasiswa berarti kita telah mempunyai *database*. Demikian pula juga kita memiliki buku telepon di rumah, buku itu pun sebuah *database*. Ada banyak hal yang tidak mungkin atau sukar sekali dilakukan secara manual. Disinilah perlunya program komputer seperti *Microsoft Access* untuk mengelola *database*.

Access memungkinkan anda untuk mengumpulkan data, menyinpan data, mengatur informasi seperti halnya membuat laporan yang mengarah pada

kesimpulan akhir. Berikut ini beberapa hal yang dapat dilakukan dengan access, diantaranya :

1. Mengetik langsung ke dalam database atau mengimpornya langsung ke program lain
2. Mngurutkan, mengindeks, dan mengatur data dengan cara yang diinginkan
3. Scara cepat dapat membuat laporan menggunakan semua data atau sebagian data
4. Mngkostumasi form-form entry data sehingga mempermudah pemakai komputer yang kurang berpengalaman untuk memasukan informasi baru ke dalam database.

2.7.1 Jenis Data di dalam *Microsoft Access 2003*

Jenis data yang dimiliki *Microsoft Access* adalah sebagai berikut :

1. *Text*

   Jenis data *text* (teks) digunakan untuk menyimpan data seperti nama, alamat, atau bilangan yang tidak membutuhkan perhitungan, seperti nomor telepon, kode pos, atau nomor serial produk.

2. *Memo*

   Jenis data *memo* digunakan untuk menyimpan data teks yang panjangnya lebih dari 225 karakter.

3. *Number*

   Jenis data *number* (bilangan) digunakan untuk menyimpan nilai numerik untuk perhitungan matematis, kecuali perhitungan yang menyangkut uang.

4. *Date/Time*

   Digunakan bagi *field* yang menampung data tanggal atau jam.

5. *Currency*

   Digunakan untuk menampung bilangan. Kelebihan *field* jenis ini dari *field number* adalah bahwa bilangan yang disimpan dalam *field* ini tidak akan kena pembulatan pada saat perhitungan.

6. *AutoNumber*

   Digunakan untuk membuat *field* yang secara otomatis akan memasukkan bilangan pada saat kita menambahkan sebuah *record*.

7. *Yes/No*

   Membuat *field* untuk menampung dua jenis keadaan , ya atau tidak. *Field* ini menyimpan bilangan -1 untuk entri *Yes* dan 0 untuk entri *No*.

8. *OLE Object*

   Membuat *field* untuk menyimpan objek *OLE*, file dokumen, gambar, dan file lainnya yang dibuat dalam program lain.

9. *Hiperlink*

   Teks atau gabungan teks dan bilangan yang disimpan sebagai teks dan digunakan sebagai alamat *hiperlink*.

10. *Lookup Wizard*

    Membuat *field* yang menampilkan salah satu dari dua macam daftar yang memudahkan pemasukan data. Kedua macam daftar itu adalah :

    a. Daftar *Lookup* yang menampilakn nilai-nilai yang dirujuk dari sebuah tabel atau *query* yang telah ada.

b. Daftar nilai yang menampilkan kumpulan nilai yang tidak bisa diubah yang kita masukkan saat kita membuat *field* ini.

2.7.2 Pembuatan *Database* baru dalam *Microsoft Access 2003*

Di jendela *database*, ada enam tab yang bisa diplih, yaitu *Tables*, *Queries*, *Forms*, *Reports*, *Pages*, *Macros* dan *Modules*.

a. Tables

Untuk menyimpan data yang kita buat.

b. Queries

Select Query adalah jenis *query* yang paling umum. Dengan *select query* bisa didapatkan data dari satu atau beberapa tabel dan menampilkan hasilnya dalam sebuah *datasheet* dimana dapat juga memperbaharui *record-record* dengan beberapa pembatas.

c. Forms

Sebuah *form* membutuhkan sumber data yang berasal dari tabel atau *query*.

d. Report

Report adalah tampilan untuk melihat data pada tabel atau *query*. Berbeda dengan *form* yang biasa menerima masukan data, *report* hanya menampilkan data saja.

e. Pages

Pages adalah tampilan untuk membuka data *access pages* di dalam *microsort access database* atau *microsoft access project* yang digunakan

untuk membuka atau mengubah *page* yang dipilih atau membuat *page* baru.

f. Macros

Makros bisa megotomasi beberapa tindakan seperti membuka *form* atau membuka *report*.

g. Modules

Moduls adalah tampilan untuk di dalam *microsoft access database* atau *microsoft access project* yang digunakan untuk mengubah dengan memilih modul atau membuat modul baru.