

BAB II

LANDASAN TEORI

2.1 Pengertian Penjualan

Penjualan memiliki pengertian ilmu dan seni yang mempengaruhi pribadi yang dilakukan oleh penjual untuk mengajak orang lain agar bersedia membeli barang atau jasa yang ditawarkannya. Penjualan adalah bagian dari pemasaran yang berkonsentrasi pada mempelajari lingkungan usaha. Mendeteksi dan menghasilkan kebutuhan dari konsumen, dan mengenali secara jelas kelompok-kelompok konsumen yang harus yang dilayani kebutuhannya oleh persediaan, saat ini maupun yang akan datang yang dimiliki oleh perusahaan¹.

Penjualan terdiri penjualan secara kredit maupun secara tunai. Kredit memiliki pengertian membayar sesuatu dengan angsuran². Sehingga penjualan secara kredit memiliki pengertian usaha untuk mengajak orang lain untuk membeli barang atau jasa secara angsuran. Sedangkan Tunai memiliki pengertian pembayaran secara kontan atau langsung³. Penjualan Tunai berarti merupakan usaha penjual untuk mengajak pembeli untuk membeli secara kontan.

¹ Derek.J. Moud dan Peter Wooliams, *Business Management An Introduction using Microsample based Simulation* ,1986, Hal:21.

² Ananda Santoso, *Kamus Lengkap Bahasa Indonesia*, 1995, Hal: 192

³ Ibid, Hal: 371

2.2 Pengertian Apartemen

Apartemen adalah tempat tinggal (terdiri atas kamar duduk, kamar tidur, kamar mandi, dan dapur) yang berada pada satu lantai bangunan bertingkat untuk bertempat tinggal yang strategis, efisien, aman⁴. Apartemen merupakan jenis hunian yang menjanjikan kepraktisan dan kemudahan bagi penghuninya, karena hal tersebut saat ini banyak orang, umumnya dari kalangan menengah ke atas yang memiliki jenis hunian ini.

2.3 Pengertian Penjualan Apartemen

Penjualan Apartemen berarti usaha yang dilakukan oleh penjual atau pengusaha untuk mempengaruhi pembeli untuk membeli jenis hunian yang strategis, efisien, dan aman yang berbentuk apartemen. Penjualan apartemen tersebut dapat dilakukan secara kredit, atau secara tunai

2.4 Pengolahan Data

Data adalah kumpulan kejadian yang diangkat dari suatu kenyataan (fakta), dapat berupa angka-angka, huruf, simbol-simbol khusus, atau gabungan dari ketiganya⁵. Data masih belum bisa bercerita banyak sehingga masih perlu diolah lebih lanjut.

Pengolahan data memiliki pengertian proses manipulasi dari data ke dalam bentuk yang lebih berguna dan lebih berarti, yaitu berupa suatu informasi. Dengan

⁴ Ibid, Hal:24.

⁵ Wikibook.org,2007

demikian, informasi adalah hasil dari suatu kegiatan pengolahan data yang memberikan bentuk yang lebih bermakna dari suatu fakta.

2.5 Dasar Perancangan Aplikasi

Sebelum penulis memaparkan gagasan dan ide dari pembuatan aplikasi ini ada baiknya mengetahui dasar dari perancangan sistem. Dasar perancangan aplikasi dirasa perlu dipaparkan, karena tanpa mengetahui hal ini tidak akan diketahui tujuan dari pembuatan program ini.

Aplikasi dapat diartikan sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu⁶.

Perancangan aplikasi adalah tahap setelah analisis dari siklus pengembangan aplikasi dimana pendefinisian dari kebutuhan–kebutuhan fungsional dan persiapan untuk rancang bangun dan implementasi dalam menggambarkan bagaimana suatu aplikasi dibentuk. Alat yang digunakan dalam perancangan aplikasi adalah sebagai berikut:

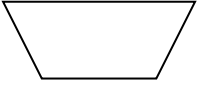
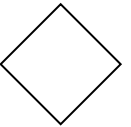
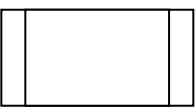
1) Flow Chart

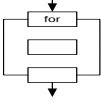
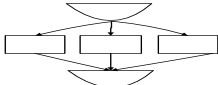
Bagan alir atau flowchart merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Flow chart adalah metode yang digunakan untuk menggambarkan bagaimana proses-proses yang terpisah dikaitkan untuk membentuk suatu sistem. Adapun

⁶ Kamus Lengkap Dunia Komputer, 2001, Hal:20

beberapa simbol-simbol yang digunakan dalam bagan alir sistem akan dijelaskan pada tabel 2.1 dibawah ini:

Tabel 2.1 *Tabel Simbol Flow Chart*

Simbol	Nama	Keterangan
	<i>Manual Operation</i>	Simbol yang digunakan untuk menyatakan proses yang dilakukan secara manual
	<i>Process</i>	Simbol yang digunakan untuk menyatakan proses
	<i>Decission symbol (IF)</i>	Simbol yang digunakan untuk Menunjukkan keputusan dalam bentuk YA atau TIDAK
	<i>One Way Flow</i>	Simbol yang digunakan untuk menyatakan hubungan antara proses yang satu dengan yang lainnya sesuai dengan panah pada garis yang ditunjuk
	<i>Predefined Process</i>	Simbol yang digunakan untuk menyatakan prses yang tak didefinisikan. Atau terdapat proses dalam proses itu.
	<i>Start/Finish</i>	Simbol yang digunakan untuk memulai aliran dan menutupnya.
	<i>Preparation</i>	Simbol untuk memberikan nilai awal

	<i>Document</i>	Simbol yang digunakan untuk menyatakan dokumen yang dihasilkan dari suatu proses
	<i>For loop</i>	Struktur logika yang digunakan untuk pengulangan blok statement
	<i>Case</i>	Struktur logika yang digunakan untuk struktur pemilihan (Case)

2) Normalisasi

Proses normalisasi adalah proses untuk mengorganisasikan file untuk menghilangkan grup elemen yang berulang – ulang.⁷

Bentuk – bentuk normalisasi adalah sebagai berikut :

1. Bentuk Tidak Normal (Unnormalized Form).
Bentuk ini merupakan kumpulan data yang direkam, tidak ada keharusan mengikuti format tertentu dan dapat saja tidak lengkap. Data dikumpulkan apa adanya sesuai dengan kedatangannya.
2. Bentuk Normal Kesatu (1NF / First Normal Form).
Bentuk normal kesatu mempunyai ciri, yaitu setiap data yang dibentuk dalam flat file (file daftar). Data dibentuk dalam record dari record.
3. Bentuk Normal Kedua (2NF / Second Normal Form).
Bentuk normal kedua mempunyai syarat, yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu.

⁷ Jogiyanto, *Analisis dan Disain Sistem Informasi*, 1989, hal 403

4. Bentuk Normal Ketiga (3NF / Third Normal Form).
Untuk menjadi normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua attribut bukan primer tidak punya hubungan yang transitif. Dengan kata lain, setiap attribut bukanlah kunci dan haruslah bergantung pada primary key secara menyeluruh.

3) State Transition Diagram (STD)

State Transition Diagram merupakan alat *modeling behavior*, yaitu suatu sistem yang dipakai pada suatu sistem *real time* dan *interface* manusia dalam suatu sistem *online*. *State transition diagram* digunakan untuk memperlihatkan keadaan dari suatu objek, kejadian yang menyebabkan perubahan dari suatu keadaan ke keadaan lain, akibat dari perubahan tersebut. Dapat disimpulkan bahwa *State Transition Diagram* adalah suatu diagram yang menggambarkan atau menjelaskan secara lambat laun terjadi didalam suatu sistem. Dua elemen yang penting dari suatu *state transition diagram* adalah:

- a. *State*, adalah suatu objek mewakili hasil keseluruhan dari keadaan-keadaan yang telah dialaminya berdasarkan sifat dari obyek itu sendiri. Setiap *state* harus diberi nama dan jika namanya cukup panjang bisa diganti dengan simbol. Nama setiap *state* harus unik dan sesuai.

- b. *State Transition*, suatu kejadian akan mengubah status suatu sistem. Perubahan sistem ini disebut *state transition* yang menghubungkan dua *state*.

Berikut merupakan contoh sederhana dari suatu *state transition diagram*:

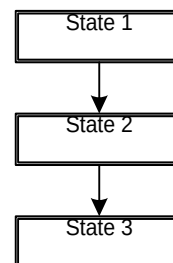


Diagram 2.1 *State Transition Diagram*

4) Spesifikasi Modul (*Pseudocode*)

Spesifikasi modul (*Pseudocode*) adalah kode yang mirip kode program yang sebenarnya. Spesifikasi modul digunakan untuk menggambarkan algoritma yang dikomunikasikan kepada *programmer*.

Struktur dasar spesifikasi modul mengikuti struktur dasar dari pemograman terstruktur yang terdiri dari struktur urut, struktur program dan perbaikan kesalahan program.

5) **Entity Relationship Diagram (ERD)**

ERD adalah model konseptual yang mendeskripsikan hubungan antar penyimpanan (dalam DFD) ⁸.

5.1 **Komponen Utama Pembentuk ERD**

Komponen-komponen utama pembentuk *Entity Relationship Diagram* (ERD) adalah :

1. Entitas (*Entity*)

Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain.

2. Atribut (*Attributes/Properties*)

Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (*properti*) dari entitas tersebut. Jenis-jenis atribut :

a. *Key*

Atribut yang digunakan untuk menentukan suatu *entity* secara unik. *Key-key* relational dapat dibagi menjadi:

a) *Super key*

Satu/sekumpulan atribut yang unik yang dapat mengidentifikasi objek secara unik dalam relasi.

b) *Candidate Key*

⁸Husni Iskandar Pohan, *Pengantar Perancangan Sistem*, 1997, Hal 35

Suatu atribut yang memiliki nilai unik.

- *Primary key*

Candidate key yang terpilih.

- *Alternate Key*

Candidate key yang tidak terpilih.

- *Foreign Key*

Suatu key yang pada suatu tabel menjadi *primary key* tapi pada tabel lain sebagai atribut biasa.

b. *Atribut Simpel*, atribut yang bernilai tunggal.

c. *Atribut Multivalue*, atribut yang memiliki sekelompok nilai untuk setiap *instant entity*.

d. *Atribut Composite*, suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.

e. *Atribut Derivatif*, suatu atribut yang dihasilkan dari atribut yang lain.

f. *Weak entity*

3. Relasi (*Relationship*)

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

4. Kardinalitas/Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.

5.2 Simbol Yang digunakan di Dalam ERD

Simbol yang digunakan untuk menggambarkan struktur dan hubungan antar data dalam ERD adalah : ⁹

1. Persegi panjang, menyatakan himpunan Entitas



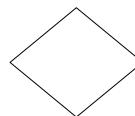
Gambar 2.1 Entitas

2. Lingkaran/Elip, menyatakan Atribut (Atribut yang berfungsi sebagai key digaris bawah).



Gambar 2.2 Atribut

3. Belah ketupat, menyatakan Himpunan Relasi.



Gambar 2.3 Himpunan Relasi

⁹ Fathansyah, *Basis Data*, 2001, Hal 70

4. Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dan Himpunan Entitas dengan Atributnya.

Gambar 2.4 Penghubung

5. Kardinalitas Relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, 1 dan M untuk relasi satu-ke-banyak atau M dan M untuk relasi banyak-ke-banyak).

5.3 Kardinalitas Relasi

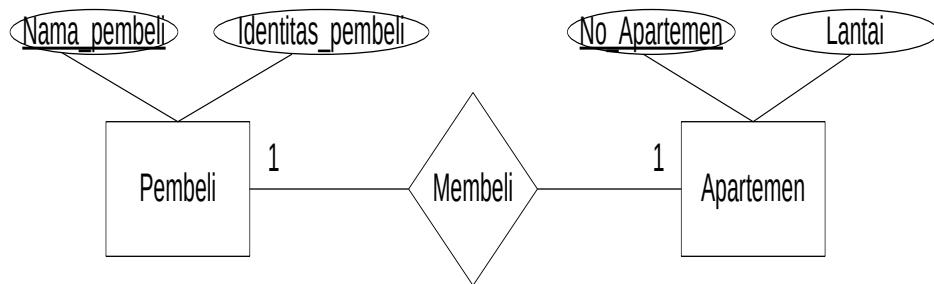
Menurut Fathansyah, berikut adalah contoh penggambaran relasi antar himpunan entitas lengkap dengan kardinalitas relasi dan atribut-atributnya :

1. Relasi satu-ke-satu (*one-to-one*)

Contoh :

Adanya relasi antara himpunan entitas pembeli dengan himpunan entitas apartemen. Himpunan relasinya kita beri nama “Membeli”. Pada relasi ini, setiap pembeli paling banyak membeli satu apartemen dan setiap apartemen pasti dibeli oleh paling banyak satu orang pembeli.

Maka penggambarannya adalah :

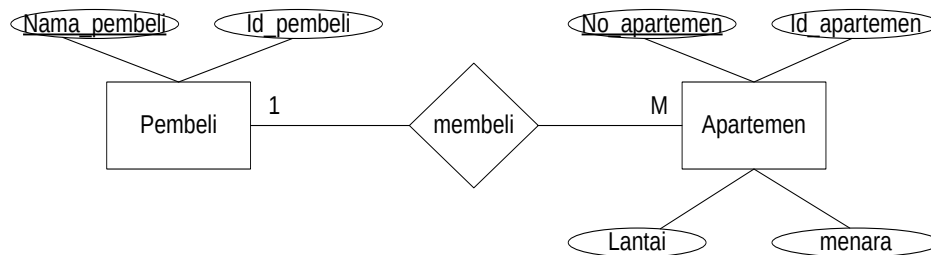


Gambar 2.5 Relasi satu-ke-satu

2. Relasi satu-ke-banyak (*one-to-many*)

Contoh :

Adanya relasi antara himpunan entitas pembeli dengan himpunan entitas apartemen. Himpunan relasinya kitat beri nama “membeli”. Pada relasi ini, setiap pembeli dapat membeli lebih dari satu apartemen, sedang setiap apartemen dibeli hanya oleh paling banyak satu orang pembeli. Maka penggambarannya adalah :

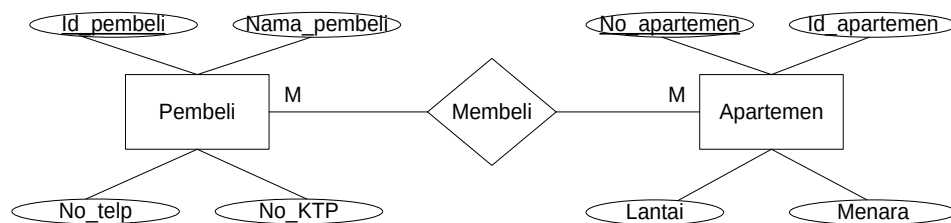


Gambar 2.6 Relasi satu-ke-banyak

3. Relasi banyak-ke-banyak (*many-to-many*)

Contoh :

Adanya relasi antara himpunan entitas pembeli dengan himpunan entitas apartemen. Himpunan relasinya kita beri nama “membeli”. Pada relasi ini, setiap pembeli dapat membeli lebih dari satu apartemen, demikian juga sebaliknya setiap apartemen dapat dipelajari oleh lebih dari satu orang pembeli. Maka penggambarannya adalah :



Gambar 2.7 Relasi banyak-ke-banyak

2.6 Basis Data (*Database*)

2.6.1 Definisi Basis Data

Basis Data (*Database*) adalah sekumpulan *file* yang saling berhubungan memungkinkan para pemakai atau pemogram lain untuk mengakses atau memanipulasi *file-file* tersebut¹⁰.

Tujuan pertama merancang suatu *database* dalam membuat suatu aplikasi adalah untuk memenuhi informasi yang berisikan kebutuhan-

¹⁰ Fathansyah, *Op cit*, hal 9

kebutuhan *user* secara khusus dan dalam aplikasi-aplikasinya. Kemudian yang kedua adalah untuk memudahkan pengertian struktur informasi. Dan yang ketiga adalah untuk mendukung kebutuhan-kebutuhan pemrosesan data dan beberapa obyek penampilan (*processing time*, *response time* dan *storage space*).

Database Management System (DBMS) adalah kumpulan *file* yang saling terkait bersama dengan program untuk pengolahan tersebut. *Database* adalah kumpulan datanya, dan program pengolahannya sendiri berdiri dalam satu paket program komersial untuk membaca data, mengisi data, menghapus data, melaporkan data dalam *database*.¹¹ Definisi *database* terdiri dari:

a. *Entity*

Entity adalah orang, tempat, kejadian atau konsep yang informasinya direkam.

b. *Attribute*

Setiap *Entity* mempunyai *attribute* atau sebutan untuk mewakili suatu *Entity*. *Attribute* disebut juga sebagai data elemen, data *field*, data *item*.

¹¹Kristanto, Hariyanto, *Konsep dan Perancangan Database*, 1996, hal 1

c. *Data Value* (Nilai atau Isi Data)

Data Value adalah data aktual atau informasi yang disimpan pada tiap elemen atau *attribute*.

d. *Record/Tuple*

Kumpulan elemen-elemen saling berkaitan menginformasikan tentang suatu *entity* secara lengkap.

e. *File*

Kumpulan *record-record* sejenis yang mempunyai panjang elemen, *attribute* yang sama, namun berbeda-beda *data value*-nya.

Basis data dapat didefinisikan dalam sejumlah sudut pandang seperti¹² :

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan *file/tabel/arsip* yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

¹² Fathansyah, *Op cit*, Hal 2

2.6.2 Operasi Dasar Basis Data

Operasi-operasi dasar yang dapat dilakukan berkenaan dengan basis data dapat meliputi¹³:

- a. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru.
- b. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus beserta isinya, jika ada).
- c. Pembuatan file/tabel baru ke suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
- d. Penghapusan file/tabel dari suatu basis data (*drop tabel*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
- e. Penambahan/pengisian data baru ke sebuah file/tabel di sebuah basis data (*insert*), yang identik dengan penambahan lembaran arsip ke sebuah map arsip.
- f. Pengambilan data dari sebuah file/tabel (*retrieve/search*), yang identik dengan pencarian lembaran arsip dari sebuah map arsip.
- g. Pengubahan data dari sebuah file/tabel (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
- h. Penghapusan data dari sebuah file/tabel (*delete*), yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

¹³ *Ibid* ,Hal 4

2.6.2.1 Kegunaan Basis Data

Pemanfaatan sebuah basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) seperti berikut ini¹⁴ :

1. Kecepatan Dan Kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, dari pada kita menyimpan data secara manual.

2. Efisiensi Ruang Penyimpanan (*Space*)

Karena keterkaitan yang erat antara kelompok data dalam sebuah basis data, maka redundansi (pengulangan) data pasti akan selalu ada. Banyaknya redundansi ini tentu akan memperbesar ruang penyimpanan yang harus disediakan. Dengan basis data, efisiensi penggunaan ruang penyimpanan data dapat dilakukan dengan penekanan jumlah redundansi data, yaitu dengan menerapkan pengkodean atau dengan membuat relasi antar kelompok data yang saling berhubungan.

¹⁴*Ibid*, Hal 5

3. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (*constraint*) tipe data, domain data, keunikan *dll* yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

4. Ketersediaan (*Availability*)

Pertumbuhan data sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu kita gunakan. Karena itu kita dapat memilah adanya data utama/*master*/referensi, data transaksi, data histori hingga data kadaluarsa. Data yang sudah jarang atau tidak pernah digunakan dapat dilakukan penghapusan atau dipindahkan ke media penyimpanan. Disisi lain, karena kepentingan pemakaian data, sebuah basis data dapat memiliki data yang disebar di banyak lokasi geografis. Dengan pemanfaatan teknologi jaringan komputer data yang berada disuatu lokasi, dapat juga diakses (menjadi tersedia/*available*) bagi lokasi lain

5. Kelengkapan (*Completeness*)

Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat

menambah *record-record* data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan field-field baru pada suatu tabel.

6. Keamanan (*Security*)

Dalam sistem pengelolaan basis data kita dapat menerapkan aspek keamanan basis data dengan menentukan siapa-siapa (pemakai) yang boleh menggunakan basis data beserta objek-objek didalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukannya.

7. Kebersamaan Pemakaian (*Shareability*)

Pemakaian basis data sering kali tidak terbatas pada satu pemakai saja atau disatu lokasi saja atau oleh satu sistem aplikasi saja.

2.6.2.2 Bahasa Basis Data

Cara berinteraksi/berkomunikasi antara pemakai dengan basis data tersebut diatur dalam sebuah bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS (*Database Management System*). Bahasa itu dapat kita sebut sebagai bahasa basis data yang terdiri atas sejumlah perintah (*statement*) yang diformulasikan dan dapat diberikan *user* dan dikenali/diproses oleh DBMS untuk

melakukan suatu aksi/pekerjaan tertentu.¹⁵ Contoh-contoh bahasa basis data adalah *SQL*, *dbase*, *QUEL* dan sebagainya.

Sebuah bahasa basis data biasanya dapat dipilah ke dalam 2 bentuk, yaitu:

a. *Data Definition Language (DDL)*

Struktur/skema basis data yang menggambarkan/mewakili desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yang disebut *Data Definition Language (DDL)*. Dengan bahasa inilah kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel, dan sebagainya.

b. *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 :

- a) Penyisipan/penambahan data baru kesuatu basis data (*Insert*).
- b) Pengubahan data dari suatu basis data (*Update*).
- c) Penghapusan dari suatu data (*Delete*).

¹⁵*ibid*, Hal 14

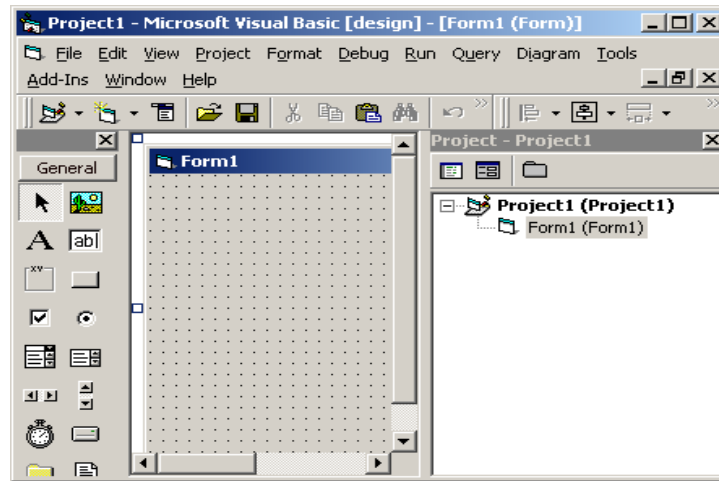
2.7 Visual Basic

Visual Basic adalah bahasa pemrograman yang merupakan pengembangan dari bahasa pemrograman Basic (pemograman prosedural), sebagai sarana untuk menghasilkan aplikasi-aplikasi berbasis windows. Visual Basic adalah bahasa pemrograman yang berorientasi objek. Salah satu keistimewaanya adalah VB memiliki compiler yang handal sehingga dapat membuat file executeable yang lebih cepat dan efisien, tambahan-tambahan Kontrol baru yang canggih dan juga sarana pengaksesan database yang lebih cepat¹⁶.

2.7.1 Lingkungan Visual Basic

Lingkungan pemrograman Visual Basic mengandung semua sarana yang anda butuhkan untuk membangun program-program windows dengan cepat dan tepat, yang terdiri dari Baris Menu, Toolbar, Toolbox, Jendela Form, Jendela Project Container, Jendela Project, Jendela Properties, Jendela Form Layout dan Jendela Immediate.

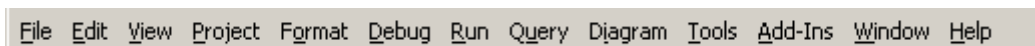
¹⁶Ariyo Suryo Kusumo, *Buku Latihan Microsoft Visual Basic 6.0*, 2003,hal 10



Gambar 2.8 Lingkungan Visual Basic (IDE)

a. Baris Menu

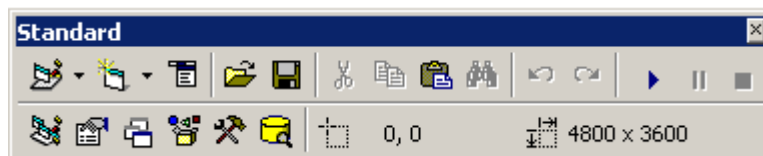
Baris menu adalah kumpulan perintah-perintah yang digunakan untuk mengatur penggunaan aplikasi Visual Basic, mulai dari pengorganisasian file, peng-editan objek dan lain sebagainya.



Gambar 2.9 Baris Menu

b. Toolbar

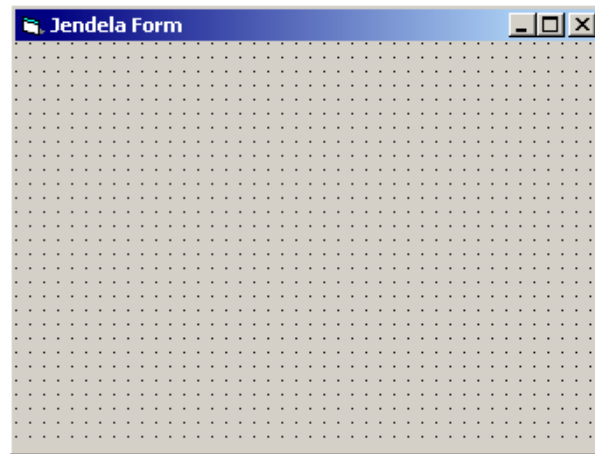
Toolbar adalah kumpulan perintah-perintah yang sering digunakan, dalam bentuk icon.



Gambar 2.10 Standard Toolbar

c. Jendela Form

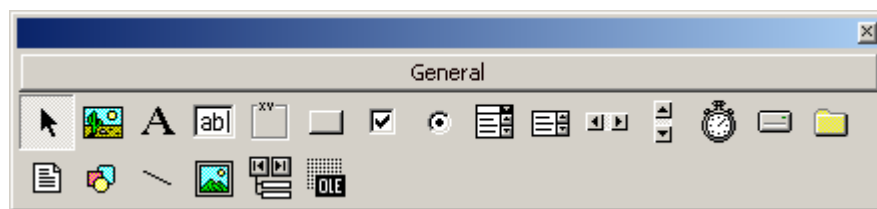
Pada Visual Basic form adalah jendela yang bisa diubah-ubah, untuk membuat antar muka program (tempat objek-objek), tempat untuk menampung objek-objek yang lain.



Gambar 2.11 Jendela Form

d. Toolbox

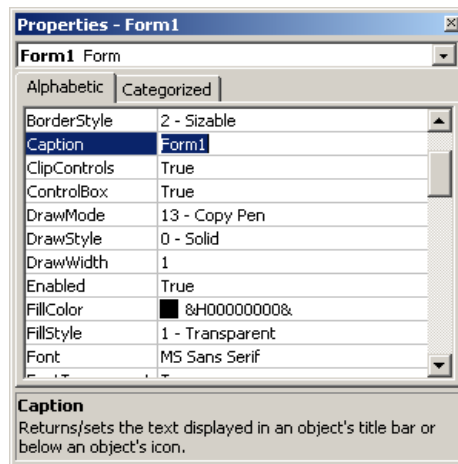
Untuk menambahkan elemen antarmuka (objek) program kedalam form kita membutuhkan tool atau kontrol yang terdapat di dalam toolbox. Yang berisikan label, textbox, image, shape dan lain-lain.



Gambar 2.12 Toolbox

e. Jendela Properties

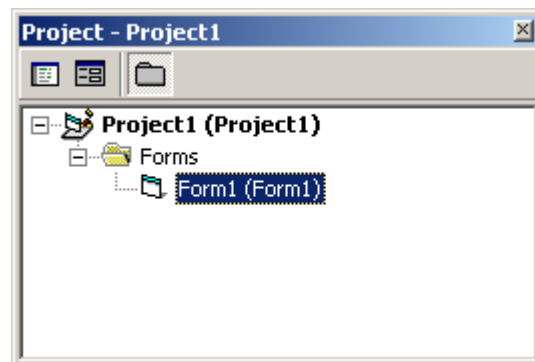
Jendela properties digunakan untuk mengubah karakteristik atau seting dari objek yang terdapat pada form.



Gambar 2.13 Jendela Properties

f. Jendela Project

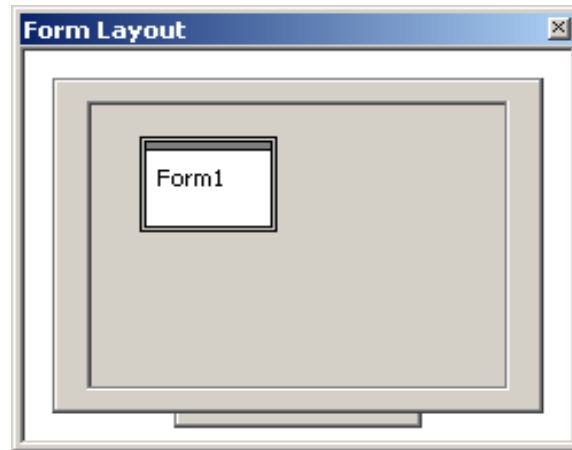
Jendela Project berisikan semua file yang digunakan pada proses pemograman dan menyediakan akses ke file tersebut menggunakan tombol khusus View Code dan View Object serta menunjukkan objek-objek yang dibuat atau digunakan.



Gambar 2.14 Jendela Project

g. Jendela Form Layout

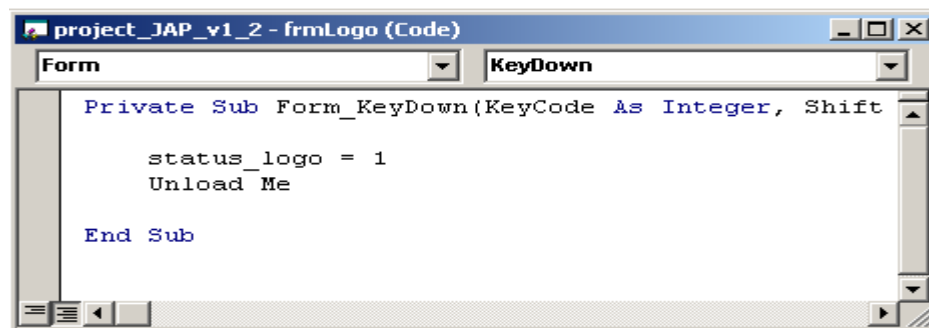
Jendela Form Layout adalah tempat untuk mengatur posisi atau letak dari form di layar monitor dan membandingkan letak tersebut dengan berbagai resolusi layar monitor.



Gambar 2.15 Jendela Form Layout

h. Window Code

Pada Visual Basic statement-statement dituliskan di window kode (gambar 2.17) dimana setiap objek dengan tiap event mempunyai tempat tersendiri untuk penulisan statement-statement pemrograman tergantung kepada kebutuhan penggunanya.



Gambar 2.16 Window Code

2.7.2 Cara Kerja Visual Basic

Sebagai bahasa yang berbasiskan objek, maka penulisan program dibagi menjadi bagian-bagian kecil yang disebut objek. Setiap objek memiliki entiti yang terpisah dengan entiti objek-objek lain dalam lingkungannya. Objek-objek terpisah ini dapat diolah sendiri-sendiri, dan setiap objek memiliki sekumpulan sifat dan metode yang melakukan fungsi tertentu sesuai dengan yang telah kita programkan kepadanya.

Setiap objek mengandung tiga hal utama seperti dibawah ini :

a. Properti atau Atribut

Properti adalah karakteristik atau sifat dari sebuah objek. Misalnya property warna untuk teks adalah hitam, ukurannya 1x2 unit, dan sebagainya.

b. Metode

Metode adalah serangkaian prosedur yang dimiliki oleh suatu objek yang akan dijalankan sesuai dengan respon yang diberikan oleh suatu perintah atau kejadian. Misalnya, objek tombol exit memiliki metode untuk keluar dari aplikasi.

c. Event

Event adalah “kejadian” atau segala sesuatu yang dapat dialami oleh sebuah objek. Sebagai contoh, meletakkan mouse di atas objek, mengklik tombol mouse pada sebuah tombol mengetik pada kotak teks, dan sebagainya.

Kontrol, Kontrol adalah salah satu objek di dalam Visual basic. Kontrol menyediakan interface (antarmuka atau penghubung) antara user (pemakai

akhir) dengan program aplikasi yang dibuat. Setiap Kontrol yang terdapat di Visual Basic 6.0 disediakan oleh jendela Toolbox. Kontrol-kontrol yang ditampilkan pada toolbox tersebut adalah Kontrol standar yang terdapat pada semua versi Visual Basic 6.0 seperti gambar 2.13.

Project, Project adalah sebuah file pengendali atau file pusat yang mengelola atau mencatat seluruh file-file yang dibutuhkan untuk membentuk suatu aplikasi. File proyek ini akan menyimpan semua komponen, perubahan, serta apa saja yang berhubungan dengan aplikasi anda. Proyek ini akan disimpan pada file dengan akhiran VBP. Untuk melihat dan mengelola semua file yang mendukung aplikasi yang anda buat, disediakan sebuah jendela Project Explorer. Project Explorer ditampilkan dalam bentuk struktur pohon. Pada bagian atas Project Explorer terdapat tiga buah icon yang berfungsi untuk memanipulasi komponen-komponen pada aplikasi Visual Basic yaitu View Code, View Object dan Toggle Folders.