BAB II

TINJAUAN PUSTAKA

2.1. Profil Tempat Penelitian

2.1.1. Sejarah Minimarket Barokah

Terbentuknya Minimarket Barokah yang sebelumnya bernama Minimarket Tubagus karena adanya perpindahan lokasi, kepengurusan dan kepemilikan, yang awalnya milik almarhum H. AHMAD UMAR selanjutnya diwariskan kepada anak ke 3 dari 9 bersaudara yaitu H. KADARUSMAN SOLIHIN (H. KODAR) pada tanggal 04 Oktober 2010.

Minimarket Barokah terbentuk pada tanggal 01 September 2000 hari Jum'at yang sekarang terletak di Jl. Tubagus Ismail Raya No. 40. Maksud dan tujuan terbentuknya Minimarket Barokah saat itu adalah ikut berperan serta dalam menyediakan berbagai kebutuhan masyarakat, baik kebutuhan pokok maupun sehari-hari.

2.1.2. Tempat dan Kedudukan Minimarket Barokah

Minimarket Barokah beralamat di Jl. Tubagus Ismail Raya No. 40 Bandung.

2.1.3. Visi dan Misi

2.1.3.1. Visi

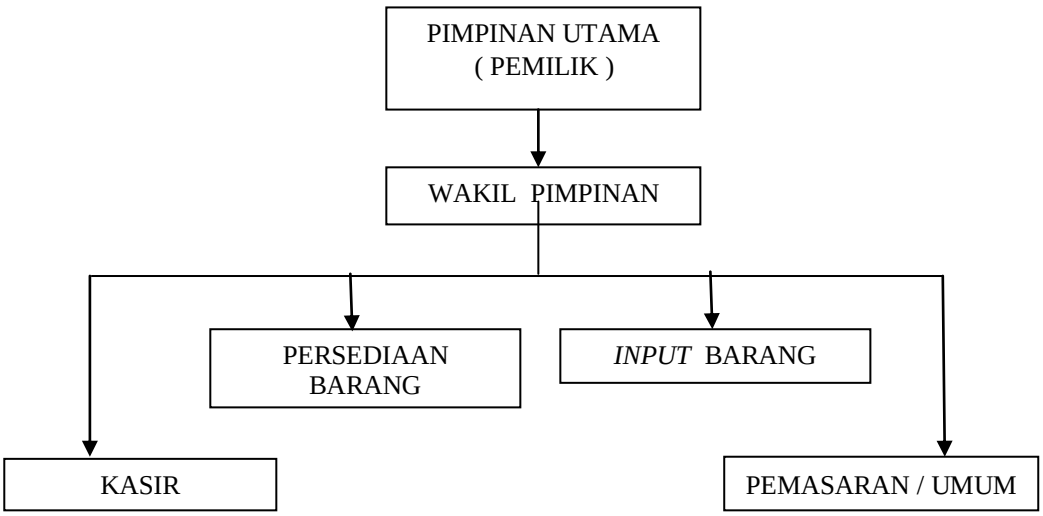
Terwujudnya Pasar Modern dalam bentuk *ritel* yang unggul dalam persaingan dengan dilandasi nilai-nilai islami.

2.1.3.2. Misi

1. Menjadikan Pasar Modern yang bersih, aman, nyaman serta memenuhi kebutuhan barang yang lengkap, murah dan bersaing
2. Memberikan kepuasan kepada konsumen dan pelayanan yang terbaik.

2.1.4. Struktur Organisasi

Minimarket Barokah dalam menjalankan proses bisnisnya selama ini dipimpin oleh seorang Pimpinan Utama dan dibantu oleh Wakil Pimpinan, selain itu juga ada Bagian Kasir, Bagian Persediaan, Bagian *Input* Barang, dan Bagian Pemasaran/umum.



Gambar 2. 1 Struktur Organisasi

Keterangan :

Pimpinan Utama : H. Kadarusman Solihin
Hj. Tati Kartati

Wakil Pimpinan : Yusuf Hendra Saputra, S.Ei

Kasir : Nani Rohaeni

Yani Maryani

Persediaan Barang : Asep Irfan

Input Barang : Sutisna

Pemasaran / Umum : Kamin

Wildan

Deskripsi jabatan pada struktur organisasi di Minimarket Barokah, yaitu :

a. Pimpinan Utama

Mempunyai tugas :

a. Bertanggung atas operasional minimarket secara keseluruhan.
b. Membuat jadwal kerja dan pembagian gaji karyawan.
c. Memeriksa laporan penjualan harian, mingguan dan bulanan.
d. Membuat peraturan dan menindaklanjuti permasalahan yang ada.

b. Wakil Pimpinan

Mempunyai tugas :

a. Membantu Pimpinan dalam melakukan tugasnya.
b. Memimpin pemeriksaan persediaan secara berskala (*stock opname*).
c. Memesan pembelian barang.

c. Kasir

Mempunyai tugas:

a. Melayani konsumen yang akan melakukan transaksi pembelian.
b. Memasukkan data penjualan
c. Membuat laporan kas setiap hari
d. Menghitung jumlah uang kas pada awal jam buka sampai tutup gerai

d. Persediaan Barang

Mempunyai tugas:

a. Memeriksa jumlah persediaan barang setiap hari.
b. Mengajukan permohonan kepada wakil pimpinan untuk melakukan pemesanan barang yang jumlahnya sudah dibatas minimal stok.
c. Bertanggung jawab atas persediaan barang.

e. *Input* Barang

Mempunyai tugas:

a. Mengisi kekosongan barang di rak *display*.
b. Memeriksa setiap saat persediaan barang yang ada di rak *display*.
c. Meminta pengisian kekosongan barang dagangan kepada bagian persediaan barang.

d. Pemasaran/umum

Mempunyai tugas:

a. Merencanakan susunan pemasaran dan operasi serta mengatur pelaksanaan penjualan, serta mengembangkan pelaksanaan sistem dan prosedur yang sesuai dengan aktivitas minimarket.
b. Menyusun program dan pengembangan kebijakan prosedur dan rencana kegiatan dibidang pemasaran seperti promosi.

2.2. Landasan Teori

2.2.1. Konsep Dasar Sistem

2.2.1.1. Pengertian Sistem

Sistem adalah sekumpulan komponen yang mengimplementasikan model dan fungsionalitas yang dibutuhkan. Komponen-komponen tersebut saling berinteraksi di dalam system guna mentransformasi input yang diberikan kepada system tersebut menjadi output yang berguna dan bernilai bagi aktor-nya. (Djon Irawan. *Perancangan Object Oriented Software dengan UML*. 2005*)*

Ada beberapa pendapat yang menjelaskan definisi sistem, yaitu :

1. Menurut Kerz (2008) ”Sistem yaitu gabungan dari sekelompok komponen baik itu gabungan dan/atau bukan manusia (*non-human*) yang saling mendukung satu sama lain serta diatur menjadi sebuah kesatuan yang utuh untuk mencapai suatu tujuan, sasaran bersama atau akhir ”.

2. Menurut Hart (2005) ”Sistem mengandung dua pengertian utama yaitu: (a) Pengertian sistem yang menekankan pada komponen atau subsistem-subsistem yang saling berinteraksi satu sama lain, dimana masing-masing bagian tersebut dapat bekerja secara sendiri-sendiri (*indenpendent*) atau bersama-sama serta saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai secara keseluruhan (b) Definisi yang menekankan pada prosedurnya yaitu merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

3. Menurut Stoa (2008) ”Pengertian dari sistem merupakan gabungan dari keseluruhan langit dan bumi yang saling bekerja sama yang membentuk suatu keseluruhan dan apabila salah satu unsur tersebut hilang dan tidak berfungsi, maka gabungan keseluruhan tersebut tidak dapat lagi kita sebut sistem.”

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa ”Sistem merpakan suatu kumpulan bagian-bagian atau subsistem-subsistem yang disatukan dan dirancang untuk mencapai suatu tujuan”.

2.2.1.2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu sebagai berikut:

1. Komponen-komponen (*components*)

Setiap sistem baik sistem dalam skala besar maupun sistem dalam skala kecil sekalipun memiliki komponen-komponen atau elemen-elemen. Komponen-komponen ini saling berhubungan dan bekerja sama sehingga tercipta satu kesatuan fungsi dari sistem. Sehingga sistem dapat mencapai tujuannya.

2. Penghubung Sistem (*Sistem Interface*)

Penghubung sistem merupakan media perantara antara subsistem yang satu dengan subsistem yang lainnya. Melalui penghubung sistem ini, maka subsistem-subsistem dapat saling meberi dan menerima sumber daya sehingga terjalin kerja sama dan dapat membentuk satu kesatuan fungsi dari sistem.

3. Lingkungan luar (*Environment*)

Lingkungan luar dari sistem adalah segala sesuatu yang berada di luar batas sistem. Lingkungan luar ini bisa juga berupa ekosistem dimana sistem

tersebut berada. Walaupun keberadaannya diluar sistem, tapi lingkungan luar dapat mempengaruhi sistem. Adanya ketidakserasian antara lingkungan luar dengan sistem dapat menyebabkan terganggunya fungsi sistem. Oleh karena itu harus senantiasa tercipta keharmonisan antara sistem dengan lingkungan luarnya.

4. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah pemisah antara satu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memberikan ruang lingkup yang jelas dari suatu sistem. Dengan adanya ruang lingkup yang jelas dari sistem tersebut, maka kita dapat memisahkan dan membedakan satu sistem dengan sistem yang lainnya maupun sistem dengan lingkungan luar.

5. Masukan Sistem (*Sistem Input*)

Masukan adalah bahan atau energi yang dimasukkan kedalam sistem. Energi ini dimasukkan kedalam sistem untuk diproses oleh sistem sesuai dengan fungsi dari sistem agar dapat menghasilkan keluaran.

6. Keluaran Sistem (*Sistem Output*)

Keluaran merupakan hasil dari pengolahan suatu sistem. Keluaran ini tentunya diharapkan dapat berguna sesuai dengan tujuan dari sistem. Selain sebagai hasil akhir, sebagian keluaran bisa juga dijadikan masukan untuk sistem lainnya.

7. Pengolah Sistem (*Sistem Processing*)

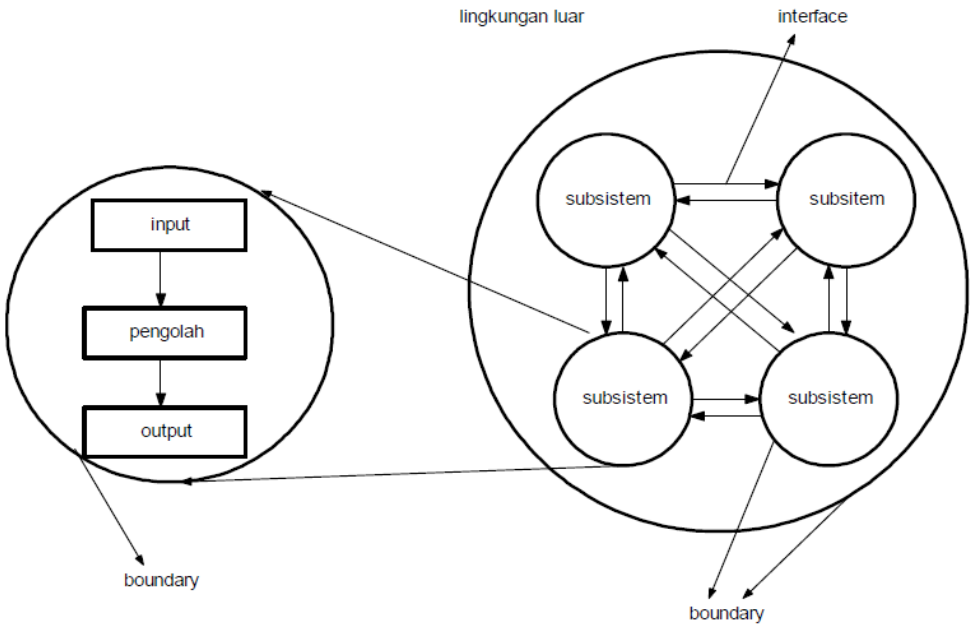
Pengolah sistem adalah mesin atau mekanisme yang digunakan untuk mengubah masukan menjadi keluaran. Pengolah memiliki peranan yang penting,

karena disinilah proses perubahan dan pendayagunaan masukan terjadi sehingga menghasilkan keluaran yang sesuai dengan tujuan sistem.

8. Sasaran dan Tujuan (*goal objective*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Tujuan merupakan hal akhir yang ingin dicapai oleh suatu sistem, sedangkan sasaran merupakan hal-hal yang menjadi objek dan titik fokus untuk meraih tujuan. Suatu sistem bisa dikatakan berhasil menjalankan fungsinya bila berhasil mencapai sasaran dan tujuan dari sistem tersebut.

Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini yang merupakan gambar karakteristik sistem.



Gambar 2. 2 Karakteristik Sistem
(Sumber : Jogiyanto. *Analisis dan Desain Sistem Informasi.* 2005)

2.2.1.3. Klasifikasi Sistem

Dari berbagai sudut pandang, sistem dapat dilasifikasikan menjadi beberapa bagian yaitu :

1. Sistem abstrak dan sistem fisik

Sistem abstrak merupakan sistem yang tidak bisa dilihat secara mata biasa dan biasanya sistem ini berupa pemikiran atau ide-ide. Contoh dari sistem abstrak ini adalah filsafat. Sistem fisik merupakan sistem yang bisa dilihat secara mata biasa dan biasanya digunakan oleh manusia. Contoh dari sistem fisik ini adalah sistem akuntasi, sistem komputer dan sebagainya.

2. Sistem alamiah dan sistem buatan

Sistem alamiah merupakan sistem yang terjadi karena pengaruh alam. Misalnya sistem rotasi bumi, sistem gravitasi dan sebagainya. Sistem buatan merupakan sistem yang dirancang dan dibuat oleh manusia. Misalnya, sistem pengolahan gaji.

3. Sistem tertutup dan sistem terbuka

Sistem tertutup merupakan sistem yang tidak berhubungan dengan bagian luar sistem dan biasanya tidak terpengaruh oleh kondisi di luar sistem. Sedangkan sistem terbuka merupakan sistem yang berhubungan dengan bagian luar sistem.

2.2.2. Konsep Dasar Informasi

2.2.2.1. Pengertian Informasi

Informasi dilihat sebagai salah satu sumber daya yang tersedia bagi manajer dalam memiliki nilai yang sama dengan sumber daya lainnya. Informasi sangatlah berharga karena informasi dapat menunjukan sumber daya lainnya.

Adapun pengertian informasi menurut Edhy Sutanta dalam bukunya *Sistem Informasi Manajemen* yaitu:

"*Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang*".

2.2.2.2. Pengertian Data

Menurut salah seorang pakar dalam bidang penelitian, Donald Cooper**,** data merupakan "kumpulan sejumlah fakta atau kenyataan yang dapat dipercaya kebenarannya sehingga dapat digunakan untuk menarik suatu kesimpulan".

2.2.2.3. Siklus Informasi

Informasi adalah data yang diolah menjadi bentuk yang berguna. Data yang diolah melalui suatu model informasi. Penerima akan menerima informasi tersebut dan membuat keputusan serta diwujudkan dengan suatu tindakan yang berarti menghasilkan suatu tindakan yang membuat sejumlah data kembali. Data tersebut akan ditanggap sebagai input, diproses kembali lewat suatu model dan seterusnnya sehingga membentuk suatu siklus.

2.2.2.4. Kualitas Informasi

Kualitas dari suatu informasi tergantung dari 3 hal yaitu :

1. Akurat (*accurat*)

Berarti informasi harus bebas dari kesalahan – kesalahan dan informasi harus mencerminkan maksudnya.

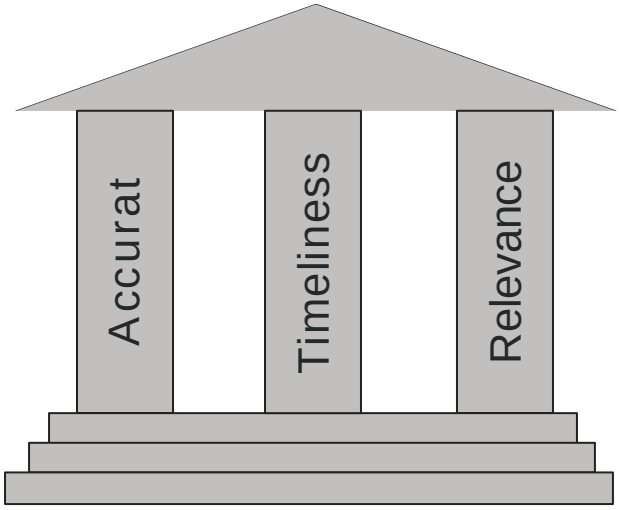
2. Tepat waktu (*time lines*)

Berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan suatu landasan dalam mengambil keputusan.

3. Relevan (*relevance*)

Berarti informasi tersebut mempunyai manfaat oleh pemakai, dimana relevansi untuk tiap–tiap individu tergantung pada yang menerima dan yang membutuhkan.

Berikut merupakan gambar kualitas informasi:



Gambar 2. 3 Kualitas Informasi
(Sumber : Jogiyanto. *Analisis dan Desain Sistem Informasi.* 2005)

2.2.3. Konsep Dasar Sistem Informasi

2.2.3.1. Definisi Sistem Informasi

Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut. Selain itu data juga

memegang peranan yang penting dalam sistem informasi. Data yang akan dimasukkan adalah sebuah sistem informasi dapat berupa formulir-formulir, prosedur-prosedur dan bentuk data lainnya (Andi Kristianto. *Perancangan Sistem Informasi dan Aplikasinya*. 2008).

Menurut Robert A. Leitch, sistem informasi adalah suatu sistem di dalam 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.

2.2.3.2. Komponen Sistem Informasi

Didalam buku yang berjudul Pengenalan Sisitem Informasi dijelaskan komponen sistem informasi terdiri dari beberapa hal sebagai berikut :

a) Perangkat keras (*Hardware*), mencakup berbagai peranti fisik sepeti komputer, dan printer.

b) Perangkat lunak (*Software*), merupakan kumpulan dari perintah yang ditulis dengan aturan tertentu untuk memerintahkan komputer dalam melaksanakan tugas tertentu.

c) Prosedur, sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

d) Orang (*human*), semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.

e) Basis data (*database*), sekumpulan tabel, hubungan , dan lain-lain yang berkaitan dengan penyimpanan data.

f) Jaringan komputer dan komunikasi data, yaitu sistem penghubungan yng memungkinkan sumber *(resources)* di pakai secara bersama atau di akses oleh sejumlah pemakai.

Perhatikan gambar komponen sistem informasi berikut ini:



Gambar 2. 4 Komponen Sistem Informasi
(Sumber : Abdul Kadir. *Pengenalan sistem Informasi.* 2003)

2.2.3.3. Kegiatan Sistem Informasi

Kegiatan yang terdapat dalam sistem informasi adalah sebagai berikut:

1. *Input*, menggambarkan bagaimana suatu kegiatan menyediakan data untuk diproses.
2. Proses (*process*), menggambarkan bagaimana suatu data di proses untuk menghasilkan suatu informasi yang bernilai tambah.
3. *Output*, suatu kegiatan untuk menghasilkan laporan dari proses diatas.
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data .

5. Kontrol, suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

2.2.4. Monitoring

Monitoring (bahasa Indonesia: pemantauan) adalah pemantauan yang dapat dijelaskan sebagai kesadaran (*awareness*) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu. *Monitoring* akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan.

Jadi kegiatan *monitoring* adalah suatu kegiatan memonitor atau mengawasi seluruh aktivitas yang dilakukan oleh seseorang. Kegiatan *monitoring* ini dapat dilakukan secara langsung. *Monitoring* langsung dilakukan dengan cara peninjauan langsung terhadap aktivitas yang sedang berlangsung seperti peninjauan barang yang masuk, barang yang keluar dan lain-lain. Sedangkan *monitoring* tidak langsung dilakukan melalui kegiatan penelaahan laporan tertulis, mencermati laporan lisan atau mewawancarai salah satu dari beberapa orang yang terlibat dalam satu kegiatan.

2.2.5. Peramalan (*Forecasting*)

Peramalan merupakan bagian awal dari suatu proses pengambilan suatu keputusan. Sebelum melakukan peramalan harus diketahui terlebih dahulu apa sebenarnya pesoalan dalam pengambilan keputusan itu. Peramalan adalah pemikiran terhadap suatu besaran, misalnya permintaan terhadap satu atau beberapa produk pada periode yang akan datang. Pada hakekatnya peramalan hanya merupakan suatu perkiraan (*guess*) tetapi dengan menggunakan teknik-teknik tertentu, maka peramalan menjadi lebih sekedar perkiraan. Peramalan dapat dikatakan perkiraan yang ilmiah (*educated guess*). Setiap pengambilan keputusan yang menyangkut keadaan di masa yang akan datang, maka pasti ada peramalan yang melandasi pengambilan keputusan tersebut.

2.2.5.1. Pendefenisian Tujuan Peramalan

Tujuan peramalan dilihat dengan waktu :

a. Jangka pendek (*Short Term*)

Menentukan kuantitas dan waktu dari item dijadikan produksi. Biasanya bersifat harian ataupun mingguan dan ditentukan oleh *Low Management*.

b. Jangka Menengah (*Medium Term*)

Menentukan kuantitas dan waktu dari kapasitas produksi. Biasanya bersifat bulanan ataupun kuartal dan ditentukan oleh *Middle Management*.

c. Jangka Panjang (*Long Term*)

Merencanakan kuantitas dan waktu dari fasilitas produksi. Biasanya bersifat tahunan, 5 tahun, 10 tahun, ataupun 20 tahun dan ditentukan oleh *Top Management*.

2.2.5.2. Beberapa Sifat Hasil Peramalan.

Dalam membuat peramalan atau menerapkan suatu peramalan maka ada beberapa hal yang harus dipertimbangkan yaitu :

1. Ramalan pasti mengandung kesalahan, artinya peramal hanya bisa mengurangi ketidakpastian yang akan terjadi, tetapi tidak dapat menghilangkan ketidakpastian tersebut.
2. Peramalan seharusnya memberikan informasi tentang beberapa ukuran kesalahan, artinya karena peramalan pasti mengandung kesalahan, maka adalah penting bagi peramal untuk menginformasikan seberapa besar kesalahan yang mungkin terjadi.
3. Peramalan jangka pendek lebih akurat dibandingkan peramalan jangka panjang. Hal ini disebabkan karena pada peramalan jangka pendek, faktor-faktor yang mempengaruhi permintaan relatif masih konstan sedangkan masih panjang periode peramalan, maka semakin besar pula kemungkinan terjadinya perubahan faktor-faktor yang mempengaruhi permintaan.

2.2.5.3. Metode Peramalan Kualitatif

Metode kualitatif dapat digunakan jika data historis maupun empiris variabel yang diramal tidak ada, tidak cukup atau kurang dipercaya. *Input* utama yang digunakan dalam metode ini adalah *judgement*, opini dan pengalaman. Metode kualitatif dibagi menjadi dua bagian, yaitu metode eksploratoris dan normatif. Metode eksploratoris dimulai dengan masa lalu dan masa kini sebagai titik awalnya dan bergerak ke arah masa depan, seringkali dengan melihat semua kemungkinan yang ada. Metode normatif dimulai dengan menetapkan sasaran dan

tujuan yang akan datang, kemudian bekerja mundur untuk melihat apakah hal ini dapat dicapai berdasarkan kendala, sumber daya dan teknologi yang tersedia. Beberapa metode atau teknik yang termasuk dalam kelompok ini antara lain : *jury of executif* (*expert*)*opinion, delphi method* dan pendekatan hirarki analitik.

2.2.5.4. Metode Peramalan Kuantitatif

Peramalan kuantitatif memiliki sifat yang objektif karena didasarkan pada keadaan aktual (data) yang diolah dengan menggunakan metode-metode tertentu. Metode ini memerlukan data historis atau data empiris sehingga setiap variabel dituntut untuk memiliki satuan ukur atau dapat diukur. Peramalan kuantitatif dapat diterapkan bila terdapat tiga kondisi berikut :

1. Tersedianya informasi tentang masa lalu (data historis).
2. Informasi tersebut dapat dikuantitatifkan dalam bentuk data numerik.
3. Dapat diasumsikan bahwa pola masa lalu akan terus berlanjut di masa mendatang

Dua asumsi pertama merupakan syarat keharusan bagi penerapan metode kuantitatif. Asumsi ketiga merupakan syarat kecukupan, artinya walaupn asumsi ketiga dilanggar oleh model yang dirumuskan masih dapat digunakan, hanya saja akan memberikan kesalahan peramalan yang relatif besar bila perubahan pola data atau bentuk hubungan fungsional tersebut terjadi secara sistematis.

Metode peramalan kuantitatif dapat dibedakan menjadi dua bagian, yaitu :

(1) Metode peramalan dengan menggunakan analisa pola hubungan berdasarkan deret waktu (*time series*). Tujuan metode ini adalah menemukan pola

data deret berskala, mengekstrapolasi pola dalam deret berkala dan mengekstrapolasikan data tersebut ke masa depan.

(2) Metode peramalan dengan menggunakan analisa pola mempengaruhinya yang bukan variabel waktu, disebut metode korelasi atau sebab akibat (*causal method*).

2.2.5.4.1. Metode *Last Period Demand*

Last period demand merupakan metode yang menggunakan data penjualan yang terjadi dari periode sebelumnya. Kelebihan metode ini adalah metode ini sangat mudah dalam perhitungannya, hanya membutuhkan 1 periode penjualan untuk melaksanakan peramalan, dan melacak perubahan dalam data yang berupa *time series* secara cepat. Selain kelebihan tersebut, metode ini juga mempunyai kelemahan yaitu data *time series* dimasa lalu tidak dihiraukan dan kemungkinan kesalahan (*error*) pada metode ini cukup besar apabila terjadi perubahan permintaan yang drastis. Metode ini sangat cocok untuk diterapkan pada kondisi permintaan yang perubahannya tidak terlalu signifikan diantara setiap periodenya.

$$S_{t+1} = S_t \quad \text{.....} \quad (1)$$

Dimana :

$S_{t+1} =$ ramalan barang untuk periode t + 1

$S_t \;\; =$ penjualan yang terjadi pada periode t

Misalkan kita buat peramalan dengan metoda *last period demand* untuk periode ke-5 adalah sebagai berikut:

Penjualan periode ke-1 15 kaleng

periode ke-2 20 kaleng

periode ke-3 12 kaleng

periode ke-4 10 kaleng

Tabel 2. 1 Peramalan dengan Metode *Last Period Demand*

Periode	Permintaan	Peramalan
1	15	-
2	20	15
3	14	20
4	12	14
5	13	12

2.2.5.4.2. Metode *Single Moving Average*

Perhitungan teknik ini diambil dari rata-rata data penjualan yang terjadi selama beberapa periode terakhir [2].

$$S_{t+1} = \frac{X_t + X_{t-1} + \cdots + X_{t-n+1}}{n} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$= \frac{1}{n} \sum \sum_{j=t-n+1}^{t} X_{t-n+1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Dimana :

S_{t+1} = Peramalan untuk periode ke $t + 1$

x_t = Data untuk periode t

n = Jangka waktu $moving\ averages$

Kalau 3 periode *moving averages*

$$S_4 = \frac{X_3 + X_2 + X_1}{3} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Misalkan kita buat peramalan dengan metoda 3 bulan *moving averages* untuk bulan April adalah sebagai berikut:

Penjualan Januari = 20.000 Kg

Februari= 21.000 Kg

Maret = 19.000 Kg

Peramalan bulan April = rata-rata penjualan 3 bulan sebelumnya (mulai Januari sampai Maret) = (20.000 Kg + 21.000 Kg + 19.000 Kg):3 = 20.000 Kg.

Andaikan kenyataan penjualan pada bulan April sebanyak 17.000 Kg maka peramalan untuk bulan Mei = (21.000 Kg + 19.000 Kg + 17.000 Kg) = 19.000 Kg, dan seterusnya. Untuk lebih jelasnya, data penjualan tiap bulan serta peramalan kalau kita memakai cara 3 bulan *moving average* dan 5 bulan *moving average* dapat dilihat pada tabel dibawah ini.

Metode *single moving average* ini biasanya lebih cocok digunakan untuk melakukan peramalan hal-hal yang bersifat *random*, artinya tidak ada gejala *trend* naik maupun turun, musiman, dan sebagainya, melainkan sulit diketahui polanya.

Tabel 2. 2 Peramalan dengan Metode 3 bulan *single moving averages* dan metode 5 bulan *single moving averages*

Nomor	Bulan	Permintaan Konsumen (1.000 Kg)	Peramalan (1.000 Kg)	
			3 periode *moving averages*	5 periode *moving averages*
1	Januari	20	-	-
2	Februari	21	-	-
3	Maret	19	-	-
4	April	17	20,00	-
5	Mei	22	19,00	-
6	Juni	24	19,33	19,80
7	Juli	18	21,00	20,60
8	Agustus	21	21,33	20,00
9	September	20	21	20,40
10	Oktober	23	19,67	21
11	November	22	21,33	21,20

Sifat-sifat *Single Moving Average*, antar lain :

1. Untuk membuat peramalan memerlukan data historis selama jangka waktu tertentu.
2. Semakin panjang jangka waktu *moving average* akan menghasilkan *moving averages* yang semakin halus.

(Sumber : Pangestu Subagyo. *Forecasting Konsep dan Aplikasi Edisi Kedua.* 1997)

2.2.5.5. Menghitung Kesalahan Peramalan

Untuk menghitung kesalahan (*error*) peramalan biasanya digunakan *mean absolute error* atau *mean squared error*. *Mean absolute error* adalah rata-rata nilai *absolute* dari kesalahan meramal (tidak dihiraukan tanda positif dan negatifnya) atau

$$\text{MAE} = \frac{\sum |X_t - S_t|}{n} \qquad \text{.................................(5)}$$

Sedang *mean squared error* adalah rata-rata dari kesalahan peramalan dikuadratkan atau

$$\text{MSE} = \frac{\sum (X_t - S_t)^2}{n} \qquad \text{.................................(6)}$$

Nilai error yang asli biasanya tidak dirata-rata sebagai ukuran besar kecilnya error, sebab ada yang positif dan ada yang negatif. Sehingga kalau dijumlah error pasti akan kecil, sebab error yang positif akan dikurangi dengan error yang negatif.

2.2.6. Analisis Perancangan Terstruktur

Dalam tahap perancangan suatu sistem diperlukan adanya teknik-teknik mengalir didalam sistem tersebut. Teknik-teknik tersebut adalah Diagram Kontek, *Data Flow Diagram* (DFD), Kamus Data, Normalisasi, dan *Entity Relationship Diagram* (ERD).

2.2.6.1. Diagram Konteks

Diagram konteks merupakan alat pemodelan atau suatu diagram yang menggambarkan sistem berbasis komputer yang dirancang secara global dan merupakan suatu diagram alir data tingkat atas, dimana di dalam diagram konteks ini menggambarkan seluruh jaringan, baik masukan maupun sebuah keluaran sebuah sistem.

Diagram konteks terdiri dari sebuah simbol proses tunggal yang menggambarkan sebuah sistem dan menunjukan data aliran utama untuk dan dari *terminator*. Diagram ini merupakan dasar yang digunakan untuk menentukan aliran data yang mengalir menuju sistem (*input system*) dan keluar dari sistem (*output system*), yang meliputi objek berupa kesatuan luar (*ekseternal entity*). Diagram konteks dapat mendefinisikan jangkauan proses penurunan sistem informasi yaitu menentukan apa yang menjadi bagian dari sistem informasi dan apa yang tidak menjadi bagian sistem informasi.

2.2.6.2. *Data Flow Diagram* (DFD)

Diagram aliran data atau *DFD* merupakan suatu bagan alir data yang digunakan untuk menjelaskan data yang ditransformasikan oleh suatu proses pada

suatu sistem dengan menekankan pada fungsi–fungsi yang ada dalam sistem, cara menggunakan informasi yang tersimpan dan pemindahan informasi antar fungsi di dalam sistem.

Beberapa simbol yang digunakan dalam *Data Flow Diagram* (DFD) antara lain:

1. Proses (*Process)*

Proses adalah simbol pertama data *flow diagram*. Proses dilambangkan dengan lingkaran, dimana proses ini menunjukan bagian dari sistem yang mengubah satu atau lebih *input* dan *output*. Nama proses dituliskan dengan satu kata, singkatan atau kalimat sederhana.

2. Aliran Data (*Flow*)

Aliran data digambarkan dengan tanda panah dari proses. Aliran data juga digunakan untuk menunjukan bagian-bagian informasi dari satu bagian ke bagian lain. Pembagian nama untuk aliran ini menunjukan sebuah arti untuk sebuah aliran. Untuk kebanyakan sistem yang dibuat, aliran data sebenarnya mengambarkan data yakni angka, huruf, pesan, dan macam-macam informasi lainnya.

3. Simpanan Data (*Storage)*

Simpanan data digunakan sebagai penyimpanan bagi paket-paket data. Notasi penyimpanan data digambarkan dengan garis horizontal yang pararel. Simpanan data merupakan simpanan data dari data yang berupa suatu *file* atau *database* di sistem komputer ataupun berupa arsip atau catatan manual. Nama dari simpanan data menunjukan nama *file*-nya.

4. Kesatuan Luar (*External Entity*)

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat berupa orang, Organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau *output* dari sistem.

2.2.6.3. Kamus Data

"Kamus data atau data *directory* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi".

Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap perancangan sistem, kamus data dapat digunakan untuk merancang *input*, *output*, dan merancang *database* program. Kamus data dibuat berdasarkan arus data yang ada.

2.2.6.4. Normalisasi

Suatu *file* yang terdiri dari beberapa grup elemen yang berulang-ulang perlu diorganisasikan kembali. Proses untuk mengorganisasikan *file* untuk menghilangkan grup elemen yang berulan-ulang ini disebut dengan Normalisasi (*Normalization*).

Pada proses normalisasi selalu diuji pada beberapa kondisi, apakah ada kesulitan pada saat menambah (*insert*), menghapus (*delete*), mengubah (*update*), membaca (*retrieve*) pada suatu *database*. Bila ada kesulitan pada pengujian tersebut maka relasi tersebut dapat dipecahkan menjadi beberapa tabel lagi atau

dengan kata lain perancangan yang dilakukan belum mendapatkan suatu *database* yang optimal. Ada beberapa macam kunci (*key function*) yang digunakan untuk proses pencarian, penyaringan, hapus, dan lain sebagainya yang biasa digunakan dalam pengolahan *database*, yaitu sebagai berikut :

a. Kunci Calon (*Candidat Key*)

Kunci kandidat adalah satu atribut atau set minimal atribut yang mengidentifikasi secara unik suatu kejadian yang spesifik dari suatu *entity*. Satu set minimal atribut menyatakan secara tak langsung dimana kita tak dapat membuang beberapa atribut dalam set tanpa merusak kepemilikan unik. Jika satu kunci kandidat berisi lebih dari satu atribut, maka biasanya disebut sebagai kunci gabungan (*composite key*).

b. Kunci Primer (*Primary Key*)

Kunci primer adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasi secara unik suatu kejadian yang spesifik, akan tetapi juga dapat mewakili setiap kejadian dari suatu *entity*. Setiap kunci kandidat punya peluang menjadi *Primary Key*, akan tetapi sebaiknya dipilih satu saja yang dapat mewakili secara menyeluruh terhadap *entity* yang ada.

c. Kunci Alternatif (*Alternative Key*)

Kunci alternatif adalah kunci kandidat yang tidak dipakai sebagai *Primary Key*. Sering kali kunci alternatif ini digunakan sebagai kunci pengurutan dalam pembuatan laporan.

d. Kunci Tamu (*Foreign Key*)

Kunci tamu adalah satu atribut atau set atribut yang melengkapi satu hubungan (*relationship*) yang menunjukan ke induknya. Kunci tamu ditempatkan pada *entity* anak dan sama dengan kunci *primary key* induk yang direlasikan. Hubungan antara *entity* induk dengan anak adalah hubungan satu lawan banyak anak (*one to many relationship*).

e. Kunci Super (*Super Key*)

Kunci super adalah himpunan dari satu atau lebih entitas yang dapat digunakan untuk mengidentifikasi secara unik sebuah entitas dan entitas set.

Teknik normalisasi ini juga merupakan suatu teknik yang menstrukturkan data dalam cara tertentu untuk membantu mengurangi atau mencegah timbulnya masalah yang berhubungan dengan pengolahan *database*. Proses normalisasi menghasilkan struktur *record* yang konsisten secara *logic* yang mudah untuk dimengerti dan sederhana dalam pemeliharaannya. Adapun langkah-langkah pembentukan normalisasi adalah sebagai berikut :

1. Bentuk Tidak Normal (*Unnormalized Form*)

   Bentuk ini merupakan kumpulan data yang direkam, tidak ada keharusan mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau terduplikasi, data dikumpulkan apa adanya sesuai dengan saat pemasukan.

2. Bentuk Normal Kesatu (1NF / *First Normal Form*)

   Suatu relasi jika dan hanya jika sifat dari setiap relasi atributnya bersifat atomik, yaitu jika tidak dipecah maka tidak memiliki sifat induknya.

3. Bentuk Normal Kedua (2 NF / *Second Normal Form*)

   Bentuk mormal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama atau *primary key*. Sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci *field*. Kunci *field* haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

4. Bentuk Normal Ketiga (3 NF / *Third Normal Form*)

   Untuk menjadi normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak mempunyai hubungan yang transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan pada *primary key* secara menyeluruh.

5. *Boyce Codd Normal Form* (BCNF)

   BCNF mempunyai paksaan yang lebih kuat dari bentuk normal ketiga, dan relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut *super key*.

6. Bentuk Normal Keempat (4 NF)

   Relasi R adalah bentuk 4 NF jika dan hanya jika relasi tersebut juga termasuk BCNF dan semua ketergantungan *multivalue* adalah juga ketergantungan fungsional.

7. Bentuk Normal Kelima (5 NF)

   Disebut juga PJNF (*Projection Join Normal Form*) dari 4 NF dilakukan dengan menghilangkan ketergantungan *join* yang bukan merupakan kunci kandidat.

2.2.6.5. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) adalah suatu model diagram yang menyatakan keterhubungan suatu *entity* dengan *entity* yang lain. Atau juga dapat dikatakan sebagai sebuah teknik untuk menggambarkan informasi yang dibutuhkan dalam sistem dan hubungan antar data-data tersebut.

Secara terjemahan dalam bahasa Indonesia, *Entity Relationship Diagram* adalah diagram relasi atau keterhubungan entitas. Dari model *Entity Relationship Diagram* akan didapatkan data-data yang dibutuhkan sistem. Dengan begitu maka akan didapatkan pula kejelasan aktivitas yang dilakukan dalam sistem.

Didalam *Entity Relationship Diagram* (ERD) dikenal beberapa komponen, yaitu sebagai berikut :

a. Entitas (*Entity*)

Adalah suatu objek yang memiliki hubungan dengan objek lain. Dalam ERD digambarkan dengan bentuk persegi panjang.

b. Hubungan (*Relationship*)

Dimana entitas dapat berhubungan dengan entitas lain, hubungan ini disebut dengan *entity relationship* yang digambarkan dengan garis.

Ada empat bentuk relasi dasar pada *database*, yaitu :

a. One-to-One

Artinya satu data memiliki satu data pasangan.

b. One-to-Many

Artinya satu data memiliki beberapa data pasangan.

c. Many-to-One

Artinya beberapa data memiliki satu data pasangan.

d. Many-to-Many

Artinya beberapa data memiliki beberapa data pasangan.

c. Atribut

Adalah elemen dari entitas yang berfungsi sebagai deskripsi karakter entitas dan digambarkan dengan bentuk *elips*.

2.2.7. *Database*

Biasanya aplikasi berbasis komputer yang digunakan pada berbagai institusi menggunakan *database* atau basis data. Basis data merupakan suatu bentuk pengelolaan data yang ditujukan agar pengaksesan terhadap data dapat dilakukan dengan mudah. Sistem yang ditujukan untuk menangani basis data biasa disebut DBMS (*Database Management System*). Dengan menggunakan DBMS, pemakai dapat melakukan hal-hal berikut dengan mudah, seperti :

a. Menambah data.

b. Menghapus data.

c. Mengubah data.

d. Mencari data.

e. Menampilkan data dengan kriteria tertentu.

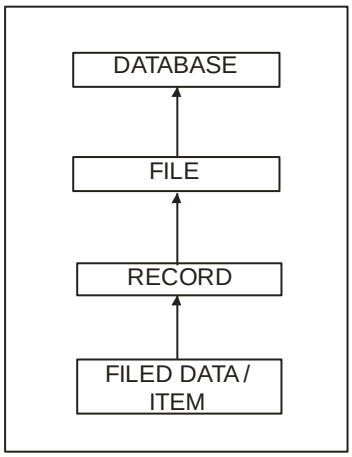
f. Mengurutkan data.

Sistem basis data adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan dengan yang lain dan membuat

menjadi beberapa aplikasi dalam suatu organisasi. Adapun keuntungan dari basis data adalah :

a. Salah satu komponen penting dalam sistem informasi.
b. Menentukan kualitas informasi yang bersifat akurat.
c. Mengurangi duplikasi data (*data redundancy*).
d. Hubungan data dapat ditingkatkan (*data relatability*).
e. Mengurangi pemborosan tempat simpanan luar sampai dengan membentuk suatu basis data.

Dalam basis data memiliki hirarki data, hal itu terlihat pada gambar berikut :



Gambar 2. 5 Hirarki Data

2.2.5.6. Elemen-Elemen Sistem *Database*

1. Karakter (*character*)

Merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf, ataupun karakter khusus (*special characters*) yang membentuk suatu item data atau *field*.

2. *Field*

Mewakili suatu atribut dari *record* yang menunjukan suatu *item* dari data, seperti misalnya nama, alamat, dan lain sebagainya. Kumpulan dari *field* membentuk suatu *record*.

a. Field name, harus diberi nama untuk membedakan *field* yang satu dengan yang lainnya.

b. Field representation, tipe *field* dapat berupa karakter, teks, tanggal, angka, dan sebagainya. Lebar *field* yaitu ruang maksimum yang dapat diisi dengan karakter data.

c. *Field value* adalah isi dari *field* untuk masing-masing *record*.

3. *Record*

Record adalah Kumpulan dari *field* membentuk suatu *record*. *Record* mengambarkan suatu unit data individu yang tertentu. Kumpulan dari *record* menbentuk suatu *file*.

4. File

File terdiri dari *record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya *file* mata pelajaran berisi data tentang semua mata pelajaran yang ada.

5. *Database*

Kumpulan dari *file* atau tabel yang membentuk suatu *database*.

2.2.8. Perangkat Lunak Pendukung

2.2.8.1. Visual Basic 6.0

Visual Basic (VB) selain disebut sebagai bahasa pemrograman (*Language Program*), juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis Windows. Dalam pengembangan aplikasi,Visual Basic menggunakan pendekatan visual untuk merancang *user interface* dalam bentuk *form*, sedangkan untuk kodingnya menggunakan bahasa *basic* (standar) yang cenderung mudah dipelajari bagi penggunanya.

Visual Basic memiliki lingkungan pengembangan terpadu bagi *programmer* dalam mengembangkan aplikasinya atau dikenal dengan IDE (*Integreted Development Environment*). Dengan menggunakan IDE *programmer* dapat membuat *user interface*, melakukan koding, melakukan *testing* dan *debuging* serta mengkompilasi program. Penguasaan yang baik akan IDE akan sangat membantu *programmer* dalam mengefektifkan tugas-tugasnya sehingga dapat bekerja dengan efisien.

2.2.8.1.1. Tampilan Visual Basic 6.0

Visual Basic memiliki beberapa komponen yang dapat digunakan dalam membuat suatu aplikasi. Komponen-komponen tersebut adalah :

1. *Control Menu*

Control Menu adalah menu yang dapat digunakan untu memanipulasi layar Visual Basic. Di menu ini bisa digunakan untuk merubah suatu ukuran, memindahkan, dan menutup layar Visual Basic. Di menu ini juga memiliki beberapa perintah.

2. *Menu*

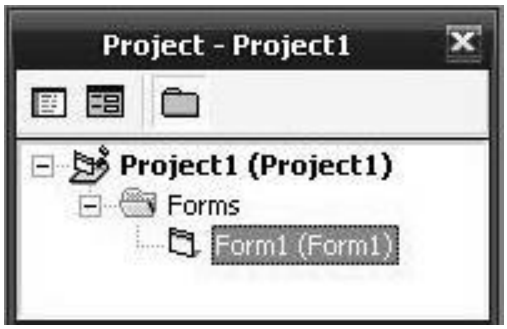
Menu dalam layar Visual Basic berisikan perintah tertentu untuk melakukan tugas tertentu. Sebagian dari isi menu ini hampir sama dengan program-program windows yang lainya, seperti menyimpan *project*, membuka *project*, dan lain-lain.

3. *ToolBar*

ToolBar adalah tombol-tombol yang mewakili suatu perintah dari Visual Basic. Biasanya tombol-tombol ini yang sering digunakan oleh *user* dalam Visual Basic.

4. *Project Explorer*

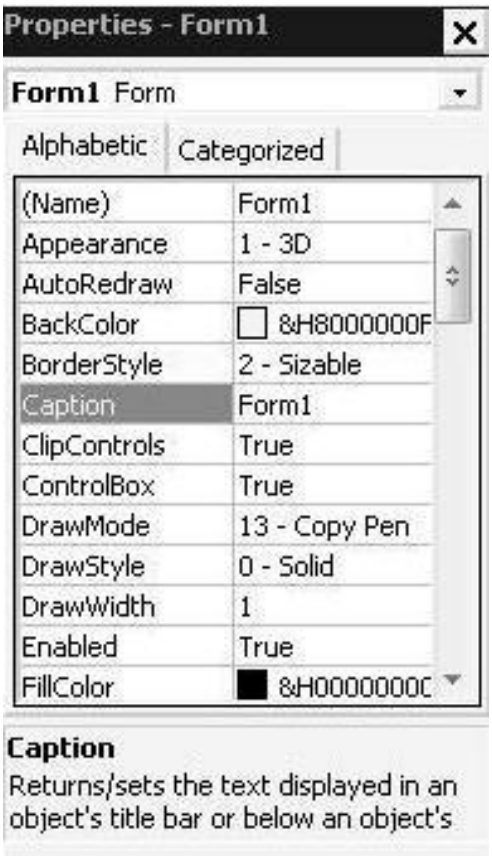
Project Explorer adalah yang mengandung semua *file* di dalam aplikasi Visual Basic. Dalam *project explorer* ditampilkan semua *file* yang terdapat pada aplikasi yang sedang dibuat. Berikut gambar *project explorer*.



Gambar 2. 6 *Project Explorer* Visual Basic

5. Properties

Properties adalah jendela yang mengandung semua informasi mengenai objek yang terdapat di dalam aplikasi Visual Basic. Biasanya di dalam *properties* terdapat properti yang dapat diatur langsung dari jendela *properties*. Berikut adalah tampilan *properties*.



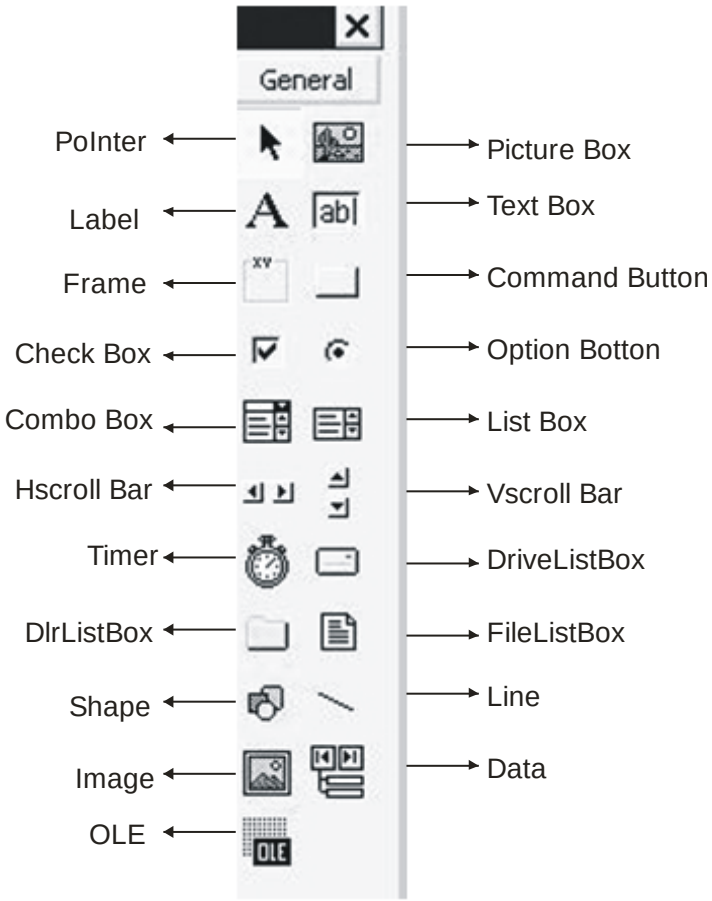
Gambar 2. 7 *Properties* Visual Basic

6. *Form Layout*

Form layout adalah jendela yang akan menunjukan bagaimana *form* bersangkutan ditampilkan ketika *runtime*. Didalam *form layout* juga terdapat menu-menu yang akan muncul jika mengklik kanan pada *mouse.*

7. *ToolBox*

Toolbox adalah layar yang berisikan komponen-komponen yang dibutuhkan untuk membuat aplikasi. Dari *toolbox* ini dapat mengambil komponen-komponen (*object*) yang akan dibutuhkan pada *form* untuk dibuat dalam aplikasi Visual Basic. Berikut gambar *toolbox* dalam Visual Basic.



Gambar 2. 8 *ToolBox* Visual Basic

Fungsi dari kontrol-kontrol tersebut, adalah :

a. *Pointer* berfungsi untuk memilih kontrol yang sudah berada pada *form*.

b. *PictureBox (Pic)* berfungsi untuk menampilkan *image* dengan format: BMP, DIB (*bitmap*), ICO (*icon*), CUR (*cursor*), WMF (*metafile*), EMF (*enhanced metafile*), GIF dan JPEG atau JPG.

c. *Label* berfungsi untuk menampilkan teks tetapi pemakai tidak dapat berinteraksi dengannya atau mengubahnya.

d. *TextBox* berfungsi untuk menampilkan teks dimana pemakai dapat mengisi atau melihat teks yang ditampilkan sebagai *output* suatu proses.

e. *Frame* berfungsi untuk mengidentifikasi sebuah grup pengontrol.

f. *Command Button* (cmd) berfungsi untuk memberikan sebuah perintah atau tindakan ketika digunakan.

g. *CheckBox* berfungsi untuk memberikan pilihan benar/salah (*True/False*) atau ya/tidak (*Yes/No*).

h. *OptionButton* berfungsi untuk memberikan pilihan lebih dari satu sebagai pilihan terhadap beberapa *option* yang hanya dapat dipilih satu.

i. *ListBox* berfungsi untuk menampilkan daftar item yang dapat dipilih salah satu.

j. *ComboBox berfungsi untuk unit kontrol* kombinasi antara *TextBox* dengan *ListBox* dimana pemasukkan data dapat dilakukan dengan pengetikkan maupun pemilihan.

k. *HScrollBar* atau *Horizontal Scroll Bar* berfungsi untuk menampilkan balok gulung horizontal.

l. *VScrollBar* atau *Vertical Scroll Bar* berfungsi untuk menampilkan balok gulung vertikal.

m. *Timer* berfungsi untuk mengeksekusi waktu kejadian pada rutin program termasuk selang waktu (*interval*).

n. *DriveListBox* berfungsi untuk menampilkan daftar *drive.*

o. *DirListBox* berfungsi untuk menampilkan daftar direktori pada *drive* terpilih.

p. *FileListBox* berfungsi untuk menampilkan daftar *file* pada *drive* dan direktori terpilih.

q. *Shape* berfungsi untuk membuat lingkaran, *elips,* dan kotak pada *form.*

r. *Line* berfungsi untuk membuat garis lurus pada *form.*

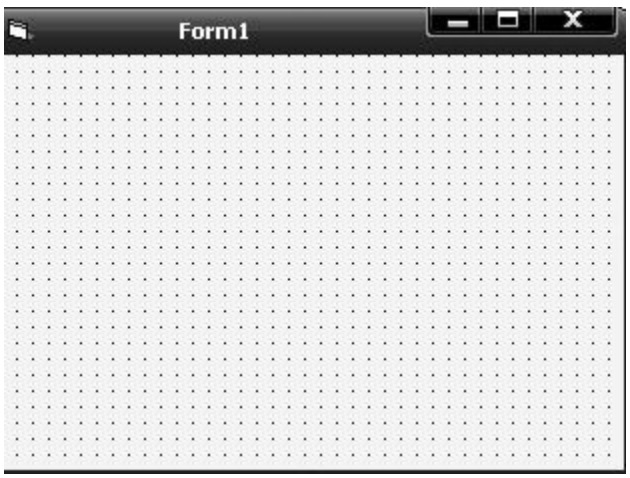
s. *Image* berfungsi untuk menampilkan gambar : .bmp, .gif, .jpg, .wmf, dan .ico. Perbedaan dengan *picture box* adalah *control* ini memberikan perintah seperti *command button* saat pemakai mengklik gambar dengan menggunakan *mouse*.

t. *Data* berfungsi untuk menampilkan *database* pada suatu *form*.

u. *OLE* atau *Object Linking and Embedding* berfungsi untuk memungkinkan pemakai menempilkan suatu objek dari aplikasi yang mendukung OLE ke aplikasi Visual Basic.

8. Form

Form adalah daerah kerja utama yang berguna untuk merancang program aplikasi dalam Visual Basic. *Form* ini bisa diletakkan berbagai macam objek seperti teks, gambar, dan sebagainya yang terdapat di *toolbox* Visual Basic. Berikut gambar *form* dalam Visual Basic.



Gambar 2. 9 *Form* Visual Basic

2.2.8.2. XAMPP

XAMPP adalah sebuah *software web server* apache yang didalamnya sudah tersedia *database server* mysql dan *support php programming*. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia Apache Web Server, MySQL Database Server, PHP Support (PHP 4 dan PHP 5) dan beberapa *module* lainnya. Hanya bedanya kalau yang versi untuk Windows sudah dalam bentuk instalasi grafis dan yang Linux dalam bentuk *file* terkompresi tar.gz. Kelebihan lain yang berbeda dari versi untuk Windows adalah memiliki fitur untuk mengaktifkan sebuah *server* secara grafis, sedangkan Linux masih berupa perintah-perintah di dalam *console.* Oleh karena itu yang versi untuk Linux sulit untuk dioperasikan. Dulu XAMPP untuk Linux dinamakan LAMPP, sekarang diganti namanya menjadi XAMPP FOR LINUX.



Gambar 2. 10 Logo XAMPP

XAMPP merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server* Apache, PHP dan MySQL secara manual.

XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi.

2.2.8.3. MySQL

SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan/seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Salah satu aplikasi *database* yang menggunakan bahasa SQL adalah MySQL. MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public Lisence*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source*/komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama SQL dalam *database* sejak lama.



Gambar 2. 11 Logo MySQL

MySQL memiliki beberapa kelebihan diantaranya :

1. MySQL sebagai *Data Management System* (DBMS) dan *Relation Databese Management System* (RDBMS).
2. MySQL adalah *software database* yang *OpenSource*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.

3. MySQL merupakan *database server*, jadi dengan menggunakan *database* ini dapat menghubungkan ke media *internet* sehingga dapat diakses jarak jauh.
4. MySQL merupakan sebuah *database client*. Selain menjadi *server* yang melayani permintaan, MySQL juga dapat melakukan *query* yang mengakses *database* pada *server*, jadi MySQL dapat juga berperan sebagai *Client*.
5. MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*.
6. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *Gigabyte* sekalipun.
7. MySQL didukung oleh driver ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi maupun Visual Basic.
8. MySQL adalah *database* menggunakan enkripsi *password*. Jadi *database* ini cukup aman karena memiliki *password* untuk mengaksesnya.
9. MySQL dapat menciptakan lebih dari 16 kunci per tabel dan dalam satu kunci memungkinkan berisi belasan *field*. Serta MySQL mendukung *field* yang dijadikan sebagai kunci primer dan kunci unik atau (*Unique*).
10. MySQL memiliki kecepatan dalam pembuatan tabel maupun peng-*update*-an tabel.

SQL (*Structure Query Language*) dibagi menjadi 2 bentuk *Query*, yaitu :

1. DDL (*Data Definition Language*)

DDL adalah sebuah metode *query* SQL yang berguna untuk mendefinisikan data pada sebuah *database*, adapun *query* yang dimiliki adalah :

a. *CREATE* : untuk melakukan pembuatan tabel dan *database*

b. *DROP* : untuk melakukan penghapusan tabel maupun *database*

c. *ALTER* : untuk melakukan pengubahan struktur tabel yang telah dibuat, baik menambah *field* (*add*), mengganti mana *field* (*change*) ataupun menamakannya kembali (*rename*), serta menghapus (*drop*).

2. DML (*Data Manipulation Language*)

DML adalah sebuah metode *query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *query* ini adalah untuk melakukan pemanipulasian *database* yang telah ada atau telah dibuat sebelumnya.

Adapun *query* yang termasuk didalamnya adalah :

a. *INSERT* : untuk melakukan peng-*input*-an/pemasukan data pada tabel *database*.

b. *UPDATE* : untuk melakukan perubahan/peremajaan terhadap data yang ada pada tabel.

c. *DELETE* : untuk melakukan penghapusan data pada tabel. Penghapusan dapat dilakukan secara sekaligus (seluruh isi tabel) maupun hanya beberapa *recordset*.

Perintah-Perintah SQL

1. Menambah Data (*Insert*)

Perintah SQL yang digunakan untuk menambah data ke *database* adalah perintah *INSERT*. Perintah ini mempunyai bentuk umum sebagai berikut:

```
INSERT [INTO] nama_tabel [(daftar field/kolom)] VALUES (daftar_data)
```

2. Memilih/Mengakses Data (*Select*)

Perintah SQL yang digunakan untuk memilih data adalah perintah *SELECT*. Bentuk umum perintah ini adalah sebagai berikut:

```
SELECT daftar_select FROM daftar_tabel
[WHERE kondisi_pencarian]
[GROUP BY daftar_group_by]
[HAVING kondisi_pencarian]
[ORDER BY daftar_order[ASC | DESC ]]
```

3. Mengedit/Mengubah Data (*Update*)

Perintah SQL yang digunakan untuk mengubah data adalah perintah *UPDATE SET*. Perintah ini mempunyai bentuk umum sebagai berikut:

```
UPDATE nama_table SET field1=databaru1 [,data2=databaru2]
[WHERE kondisi_update]
```

4. Menghapus Data (*Delete*)

Perintah SQL yang digunakan untuk menghapus data adalah perintah *DELETE FROM*. Perintah ini mempunyai bentuk umum sebagai berikut:

```
DELETE FROM tabel_hapus WHERE kondisi_hapus
```