BAB II

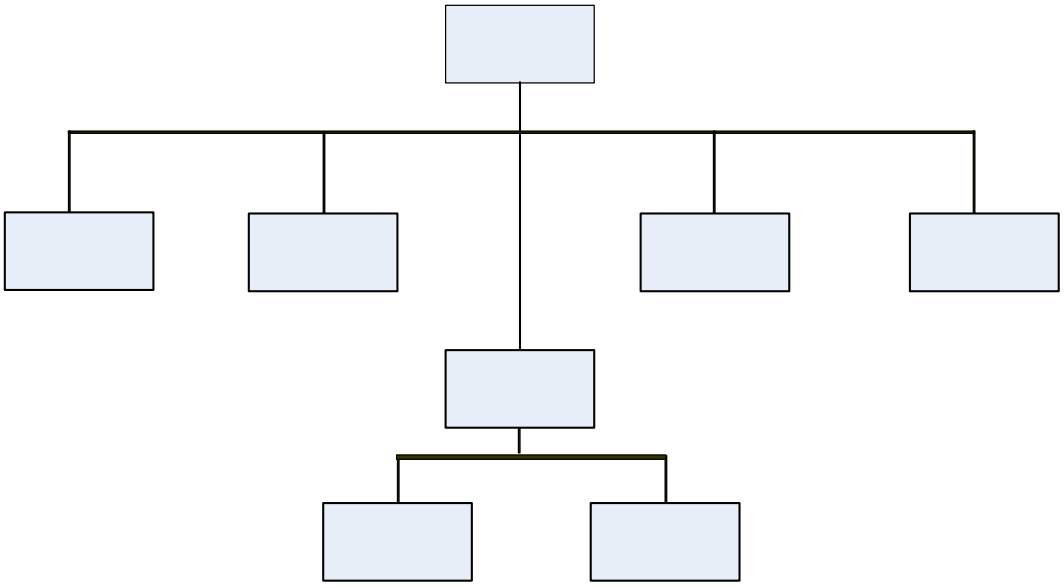
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Sejarah Berdirinya CV. RIDA *computer*.

C.V RIDA *computer* merupakan perusahaan yang memberikan pelayanan dibidang teknologi informasi, penjualan *manufacture*, komputer rakitan dan *computer maintenance* dalam kota maupun luar kota. CV. RIDA *computer* berdiri sejak tahun 1997 dan pada saat itu hanya memiliki beberapa orang karyawan. CV. RIDA *computer* bertempat di Jl Cieunteung No. 61 Tasikmalaya.

2.1.2 Struktur Organisasi



Gambar 2.1 Struktur Organisasi CV. RIDA *computer*

Tugas setiap bagian dari struktur organisasi diatas adalah sebagai berikut :

1. Pimpinan

   Adalah pimpinan di CV. RIDA dan juga sebagai penyandang dana atau modal. Tugas dari pimpinan adalah mengontrol dan mengevaluasi kinerja karyawan serta mengevaluasi perkembangan toko berdasarkan hasil dari laporan-laporan yang diterima dalam kurun waktu tertentu.

2. Administrasi

   Bertugas mencatat laporan hasil transaksi penjualan dari bagian penjualan dan bagian gudang. Hasil dari laporan yang telah di catat akan diserahkan kepada pimpinan.

3. Bagian Gudang

   Bertugas untuk mengontrol stok persediaan barang masuk dan barang keluar.

4. Bagian Penjualan

   Bertugas untuk melayani konsumen dalam penjualan barang

5. Bagian Pembelian

   Bertugas pemesanan persedian barang yang kosong di gudang kepada distributor.

6. Bagian Services

   Bertugas melayani perbaikan perangkat keras komputer.

2.2 Landasan Teori

2.2.1 Pengertian Sistem

Sistem adalah sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan. Menurut Richard F. Neuschel [1], Management by system, (Edisi kedua: New York: McGrawHill, 1960), hal 10. Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendifinisikan sistem sebagai

berikut *"Sistem adalah kumpulan dari elemen-elemen yang berinteraksiuntuk mencapai suatu tujuan tertentu"*. Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu diantaranya :

1. Memiliki komponen.

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut supra sistem, misalnya suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila

perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya.

2. Batas sistem (*boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut

3. Lingkungan luar sistem (*environment*)

   Adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung sistem (*interface*)

   Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

5. Masukan sistem (*input*)

   Merupakan 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 dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah maintanance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

6. Keluaran sistem *(Output)*

   Merupakan hasil dari energi yang diolah oleh sistem.

7. Pengolah sistem *(Process)*

   Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

8. Sasaran sistem

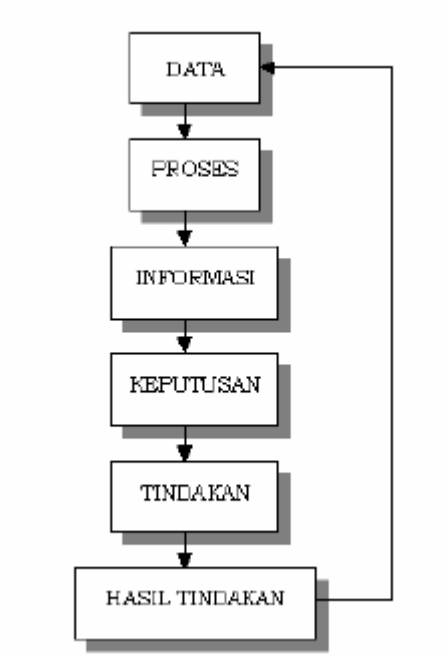
   Kalau sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

2.2.2 Pengertian Informasi

Untuk menuju pada pengertian Sistem Informasi secara utuh, diperlukan pemahaman yang tepat tentang konsep data dan informasi. Keterkaitan data dan informasi sangatlah erat sebagaimana hubungan antara sebab dan akibat. Bahwa data merupakan bentuk dasar dari sebuah informasi, sedangkan informasi merupakan elemen yang dihasilkan dari suatu bentuk pengolahan data.

Sedangkan menurut Gordon B. Davis [2] dalam bukunya *Management Informations System : Conceptual Foundations, Structures, and Development* menyebut informasi adalah *data yang telah diolah menjadi bentuk yang berguna bagi penerimanya dan nyata, berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan*.

Pengolahan data menjadi suatu informasi dapat digambarkan sebagai sebuat siklus yang berkesinambungan seperti berikut :



Gambar 2.2 Siklus Informasi

Secara sederhana dapat dikatakan bahwa data diolah menjadi suatu informasi. Dan pada tahapan selanjutnya, sebuah informasi akan menjadi data untuk terciptanya informasi yang lain.

Setiap Informasi, memiliki beberapa karakteristik yang menunjukkan sifat dari informasi itu sendiri. Karakteristik-karakteristik informasi tersebut antara lain adalah :

1. Benar atau salah.

   Karakteristik tersebut berhubungan dengan sesuatu yang realitas atau tidak dari sebuah informasi.

2. Baru.

   Sebuah informasi dapat berarti sama sekali baru bagi penerimanya.

3. Tambahan.

   Sebuah informasi dapat memperbaharui atau memberikan nilai tambah pada informasi yang telah ada.

4. Korektif.

   Sebuah informasi dapat menjadi bahan koreksi bagi informasi sebelumnya, salah atau palsu.

5. Penegas

   Informasi dapat mempertegas informasi yang telah ada, hal ini masih berguna karena dapat meningkatkan persepsi penerima atas kebenaran informasi tersebut.

Sedangkan Kualitas informasi (*quality of information*) sangat dipengaruhi atau ditentukan oleh 3 hal pokok, yaitu *relevancy*, *accuracy* dan *timelinness*.

1. Relevansi *(relevancy)*

   Informasi dikatakan berkualitas jika relevan bagi pemakainya. Pengukuran nilai relevansi, akan terlihat dari jawaban atas pertanyaan "*how is the message used for problem solving (decision masking)?"* Informasi akan relevan jika memberikan manfaat bagi pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda. Misalnya informasi mengenai hasil penjualan barang mingguan kurang relevan jika ditujukan pada manajer teknik, tetapi akan sangat relevan bila disampaikan pada manajer pemasaran.

2. Akurasi *(accuracy)*

   Sebuah informasi dapat dikatakan akurat jika informasi tersebut tidak bias atau menyesatkan, bebas dari kesalahan-kesalahan dan harus jelas mencerminkan maksudnya. Ketidakakuratan sebuah informasi dapat terjadi karena sumber informasi (data) mengalami gangguan atau kesengajaan sehingga merusak atau merubah data-data asli tersebut.

   Beberapa hal yang dapat berpengaruh terhadap keakuratan sebuah informasi antara lain adalah :

   1. Kelengkapan (*completeness*) informasi.

      "Are necessary message items present ?" Informasi yang lengkap, berarti bahwa informasi yang dihasilkan terdiri dari satu kesatuan informasi yang menyeluruh dan mencakup berbagai hal yang terkait di dalamnya. Karena apabila informasi yang dihasilkan sebagian-sebagian tentunya akan mempengaruhi dalam pengambilan keputusan atau menentukan tindakan secara keseluruhan, sehingga akan berpengaruh terhadap kemampuannya untuk mengontrol atau memecahkan suatu masalah dengan baik.

   2. Kebenaran *(correctness)* informasi.

      "Are message items correct ?" Informasi yang dihasilkan oleh proses pengolahan data, haruslah benar sesuai dengan perhitungan-perhitungan yang ada dalam proses tersebut. Sebagai contoh, jika sebuah informasi menunjukkan total nilai gaji yang harus dibayarkan pada seorang pegawai, maka informasi tersebut haruslah sudah benar dan memuat perhitungan-

perhitungan matematis yang ada di dalam prosesnya seperti perhitungan tunjangan, perhitungan potongan dan sebagainya.

3. Keamanan *(security)* informasi.

   Keamanan sebuah informasi, tergambar dari jawaban atas pertanyaan "*Did the message reach all or only the intended systems users ?* " Sebuah informasiharus aman, dalam arti hanya diakses oleh pihak-pihak yang berkepentingan saja sesuai dengan sifat dan tujuan dari informasi tersebut.

3. Tepat waktu *(timeliness)*

   "How quickly is input transformed to correct output?" Bahwa informasi yang dihasilkan dari suatu proses pengolahan data, datangnya tidak boleh terlambat. Informasi yang terlambat tidak akan mempunyai nilai yang baik, sehingga kalau digunakan sebagai dasar dalam pengambilan keputusan dapat menimbulkan kesalahan dalam tindakan yang akan diambil

2.2.3 Pengertian Sistem Informasi

Istilah sistem informasi menyiratkan suatu pengumpulan data yang terorganisir beserta tatacara penggunaanya yangmencakup lebih jauh daripada sekedar penyakian. Istilah tersebut menyiratkan suatu maksud yang ingin dicapai dengan jalan memilih dan mengatur data serta menyusun tatacara penggunaanya.

Menurut John Burch dan Gary Grudnitski: [3] " *komponen sistem informasi terdiri dari:*

1. *Blok Masukan (Input Block) yang terdiri atas data mentah informasi yang akan dijadikan sumber daya informasi.*
2. *Blok Model (Model Block) yang terdiri atas kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang ada di basis data dengan cara tertentu untuk hasil yang diinginkan.*
3. *Blok Keluaran (Output Block) adalah produk keluaran yang dihasilkan sistem informasi berupa informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta user.*
4. *Blok Teknologi (Technology Block). Teknologi merupakan ”kotak alat” atau tool box dalam sistem informasi yang digunakan untuk menerima input, menjalankan model, menyimpan dan akses data, memperoleh dan mengirim output serta kontrol dari sistem secara keseluruhan komponen teknologi (brainware, hardware dan software)*
5. *Blok Basis Data (Database Block) adalah komponen data yang saling berhubungan tersimpan di hardware dan dimanipulasi pada software*
6. *Blok Kendali (Control Block) adalah blok pengendali sistem informasi”.*

2.2.4 Pengertian Inventory

Inventory merupakan kata lain dari persediaan. Persediaan tergantung pada jenis usaha yang ditekuni oleh perusahaan yang bersangkutan. Pada perusahaan yang bergerak dibidang penjualan produk, persediaan barang merupakan salah satu unsur yang paling efektif dalam operasional perusahaan, yang secara berkelanjutan

digunakan dalam kegiatan penjualan barang harian yang dapat disajikan dalam bentuk laporan persediaan barang.

Menurut Hartono Jogianto [4] *inventory* adalah " *serangkaian kebijaksanaan dan pengendalian yang memonitor tingkat persediaan dan menentukan tingkat persediaan yang harus dijaga, kapan persediaan harus diisi, dan berapa besar pesanan harus dilakukan* ".

2.2.5 Pengertian Sistem Inventory

Berdasarkan dari pengertian sistem dan *inventory* maka penulis dapat menarik kesimpulan bahwa pengertian dari sistem *inventory* adalah sistem yang mampu mengelola pengadaan atau persediaan barang digudang dan dapat memberikan layanan tentang informasi yang berupa data dalam menentukan kapan pemesanan persediaan barang atau stok terhadap *supplier* dilakukan sebelum stoknya habis.

Sistem perhitungan persedian dapat dilakukan :

a. *Secara bersambungan (perpetual).* Dalam hali ini nilai persediaan dihitung dan ditentukan sehabis tiap transaksi penerimaan atau pengeluaran barang (pembelian, penerimaan barang untuk diolah, penyerahan produk jadi yang selesai dikerjakan, pemakaian peralatan, dan penjualan produk jadi). Akuntansi sistem persediaan bersambungan (*perpetual inventory system*) merupakan ciri akuntansi biaya yang baik, karena dengan mempergunkan cara tersebut nilai persediaan dapat diketahui dan ditentukan setiap saat dan harga pokok penjualan dapat ditentukan pada saat penjualan.

b. *Secara berkala (periodik).* Dalam hal ini nilai persediaan ditentukan berdasarkan perhitungan fisik secara berkala. Oleh sebab itu sistem pengeluaran berkala *(periodic inventory system)* juga disebut sebagai sistem persediaan fisik *(physical inventory system).* Didalam sistem ini pembelian-pembelian akan ditambahkan pada nilai persediaan awal, persediaan akhir dihitung dan ditentukan nilainya, kemudian perbedaan antara persediaan awal + pembelian-pembelian dengan nilai persediaan akhir dibukukan sebagai Harga Pokok Penjualan.

Tujuan utama kedua sistem perhitungan itu adalah untuk mentukan angka-angka yang cukup teliti dan dapat dipertanggung jawabkan, yang dapat dipergunakan untuk keperluan analisis dan pengawasan, karena perhitungan harga pokok penjualan dan penentuan nilai persediaan merupakan bagian yang penting dalam penyusunan neraca dan perkiraan Laba-Rugi perusahaan.

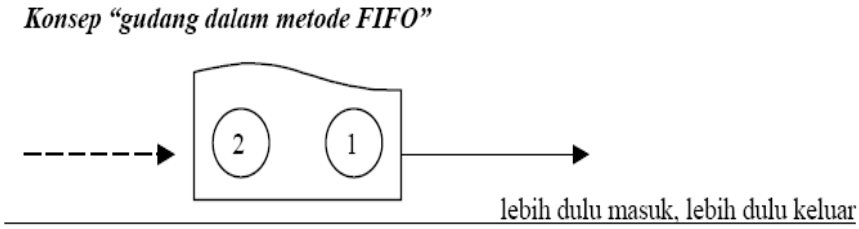
Metode-metode hitung pokok yang biasa dan banyak dipergunakan adalah:

1. *First-In First-Out* (FIFO)
2. *Last-In First-Out* (FIFO)
3. *Average Cost*

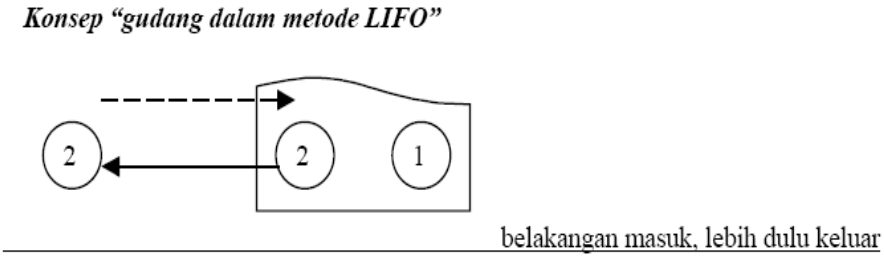
Metode-metode hitung pokok ini berkenaan dengan anggapan (asumsi) yang dipergunakan untuk menentukan arus biaya, dan secara konkretnya asumsi mengenai urutan pengeluaran barang dari gudang.

Metode First-In First-Out mengasumsikan bahwa barang-barang yang lebih dulu masuk kedalam gudang, juga akan dikeluarkan lebih dulu dari gudang.

Sebaliknya, metode Last -In First-Out, mengasumsikan bahwa barang-barang yang belakangan masuk ke dalam gudang, akan dikeluarkan lebih dulu dari gudang.



Gambar 2.3 Konsep gudang dalam metode FIFO



Gambar 2.4 Konsep gudang dalam metode LIFO

Urutan pengeluaran barang yang sebenarnya mungkin berbeda dari asumsi yang dipergunakan. Dapat saja terjadi bahwa barang yang dalam urutan pembelian, termasuk dalam kelompok yang kedua dari tiga urutan kelompok barang yang dibeli, dikeluarkan lebih dahulu, misalnya karena letak barang tersebut dekat dengan pintu keluar gudang. Asumsi-asumsi itu hanya dipergunakan untuk tujuan hitung pokok dan tidak menentukan atau ada sangkut- pautnya dengan urutan pengeluaran sebenarnya.

2.2.6 Model Proses Perangkat Lunak

Model proses perangkat lunak terdiri atas:

1. *Waterfall/Sequential linear* yaitu pendekatan pengembangan perangkat lunak yang sistematik dan sekuensial
2. *Prototype* yaitu model proses berupa pengumpulan kebutuhan perangkat lunak
3. *Rapid Aplication Development (RAD)* yaitu model *waterfall* yang menekankan siklus perkembangan yang sangat pendek dan terdiri dari fase-fase sebagai berikut:
    1. Fase *business modeling*
    2. Fase *data modeling*
    3. Fase *Process Modeling*
    4. Fase *aplikation generation*
    5. Fase *testing* dan *turnover*
4. *Evolutioner* yaitu model *iteratif* berupa pengembangan perangkat lunak sedikit demi sedikit dan yang termasuk ke dalam model ini adalah:
    1. Model pertambahan yaitu penggabungan elemen-elemen model *sequential linear* dengan filosofi *prototype iteratif*
    2. Model Spiral yaitu model proses perangkat lunak yang evolusioner yang merangkai sifat *iteratif* dari prototipe dengan cara kontrol dan aspek sistematis dari model *secuential linear*

3. Rakitan komponen yaitu model *iteratif* yang terdiri dari gabungan komponen spiral
4. Perkembangan konkuren/rekayasa konkuren sebagai paradigma bagi pengembangan aplikasi klien server
5. Model Formal mencakup aktifitas yang membawa pada spesifikasi matematis perangkat lunak komputer.
6. Model Teknik generasi ke-4 mencakup serangkaian bantu perangkat lunak yang luas yang secara umum memiliki satu hal, masing-masing memungkinkan perekayasa perangkat lunak untuk mengkhususkan beberapa karakteristik perangkat lunak pada satu tingkat yang tinggi.

2.2.7 Pemodelan Rekayasa Perangkat Lunak

2.2.7.1 Flow Map

Flow Map digunakan untuk menggambarkan prosedur kerja secara fisik dan mendefinisikan hubungan antara bagian (pelaku proses), proses (manual atau berbasis komputer) dan aliran data dalam bentuk dokumen keluaran dan masukan.

2.2.7.2 Entity-Relationship Diagram (ERD)

Entity atau entitas adalah individu yang mewakili sesuatu yang nyata dan dapat dibedakan dengan sesuatu yang lain. *Entity-Relationship Diagram* hanya berfokus pada data, dengan menunjukan jaringan data yang ada untuk suatu sistem yang diberikan. *Entity-Relatioship Diagram* mulanya diusulkan Peter Chen untuk

desain sistem *database relational* dan telah dikembangkan oleh yang lainnya. Komponen utama ERD antara lain:

1. Objek data
2. Atribut
3. Hubungan
4. Berbagai tipe indikator

Tujuan utama *ERD* adalah untuk mewakili objek data dan hubungannya dengan objek data yang lain. Dengan *ERD* kita dapat menguji model dengan mengabaikan proses yang dilakukan. Hubungan antar entitas, yaitu :

1. Satu ke Satu (*One To One Relationship*)

   Hubungan antara entity pertama dengan entity kedua adalah satu berbanding satu.



Gambar 2.5. ERD *One To One*

2. Satu ke Banyak *(One To Many)*

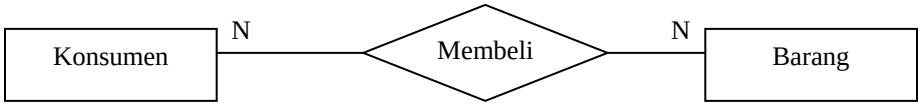
   Himpunan entity pertama dapat berhubungan satu atau lebih entity pada himpunan kedua, tetapi tidak sebaliknya.



Gambar 2.6. ERD *One To Many*

3. Banyak ke Banyak *(Many To Many)*

   Setiap entity pada himpunan entity pertama dapat berhubungan banyak entity pada himpunan entity kedua, begitu pula sebaliknya.



Gambar 2.7. ERD *Many To Many*

2.2.7.3 Diagram Konteks

Konteks diagram merupakan proses yang mewakili proses dari semua sistem. Konteks diagram menggambarkan hubungan *input* dan *output* antara sistem dengan entitas luarnya. Menurut Hartono Jogianto [4] Diagram Konteks adalah "*hubungan antara sistem dengan entitas luarnya yang berfungsi sebagai transformasi dari suatu proses yang melakukan transformasi data input menjadi data output*".

2.2.7.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) atau Diagram Aliran Data (DAD) memperlihatkan hubungan fungsional dari nilai yang dihitung oleh sistem termasuk nilai masukan, keluaran serta tempat penyimpanan internal. Diagram Aliran Data (DAD) merupakan gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam objek kemudian melewati proses yang mentransformasinya ke tujuan yang lain. *Data Flow Diagram (DFD)* digunakan pada metode pengembangan sistem yang terstruktur *(structured analysis and design).*

Simbol *Data Flow Diagram (DFD)*

1. *External Entity* (kesatuan luar)/*buondary* (batas sistem). Setiap sistem mempunyai *boundary* yang memisahkan suatu sistem dengan lingkaran luar. Sistem menerima *input* dan mengluarkan *output* pada lingkungan luarnya. *External Entity* merupakan kesatuan *(entity)* di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain di lingkungan luar yang akan memberi *input* atau menerima *output* dari sistem.
2. *Data Flow* atau arus data pada *Data Flow Diagram (DFD)* diberi simbol panah yang mengalir diantara proses, simpanan data *(data store)* dan kesatuan luar *(External Entity).*
3. *Process* adalah kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Setiap proses harus diberikan penjelasan lengkap meliputi identifikasi proses, nama proses dan pemroses.
4. *Data Store* (simpanan luar) adalah simpanan dari data yang dapat berupa file/database dari sistem komputer, suatu arsip atau catatan manual, suatu kotak tempat data di meja seseorang, suatu nilai acuan manual dan suatu agenda atau buku.

2.2.7.5 Kamus Data/ Data Dictionary (DD)

Kamus Data atau *Data Dictionary (DD)* atau istilah *system data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan Kamus Data analisis sistem dapat mendefinisikan data yang mengalir dari sistem dengan lengkap. Kamus Data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun perancangan sistem.

Pada tahap analisis, Kamus Data digunakan sebagai alat komunikasi antara analisis sistem dengan *user*. Sedangkan pada tahap perancangan sistem, Kamus Data digunakan untuk merancang *input*, laporan-laporan dan basis data. Kamus Data dibuat berdasarkan arus data yang ada di *Data Flow Diagram (DFD)* yang sifatnya global. Kamus Data harus dapat mencerminkan keterangan tentang data yang dicatat, maka kamus data harus memuat nama arus data, alias, bentuk data, arus data, penjelasan, periode, volume dan struktur data.

Tabel 2.1 Notasi Kamus Data

No	Notasi	Arti Notasi Kamus Data
1	=	artinya terbentuk dari *(is composed)* atau terdiri dari *(consist of)* atau sama dengan *(is equivalent of)*
2	+	*And* (dan)
3	[]	salah satu dari/memilih salah satu dari elemen-elemen data dari dalam kurung berikut ini
4	I	sama dengan simbol ’[]’, pemisah pilihan dari tanda []

5	N{}M	iterasi mulai nilai minimum N kali dan nilai maksimum M kali
6	()	*optional* (*parenthesis* sifatnya *optional*, dapat ada atau tidak)
7	*	keterangan setelah tanda ini adalah komentar
8	@	Petunjuk (key field)

2.2.8 Database Management System (DBMS)

Database Management System (DBMS) merupakan perangkat lunak yang di desain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, misalnya penyimpanan data dalam file dan menulis kode aplikasi yang spesifik untuk pengaturannya.

Penggunaan DBMS untuk suatu aplikasi tergantung pada kemampuan dan dukungan DBMS yang beroperasi secara efisien. Sehingga agar bisa menggunakan DBMS dengan baik, perlu diketahui cara kerja dari DBMS tersebut.

Pendekatan yang dilakukan untuk menggunakan DBMS secara baik meliputi implementasi DBMS dan arsitektur secara detail untuk dapat memahami desain dari suatu basis data.

Operasi basis data melalui DBMS dapat berupa: penambahan data, pencarian data, pengubahan data, penghapusan data, pengurutan data, penggabungan data, dan penyimpulan/pengelompokon data.

2.2.9 Sistem Komputer

Sistem yang berbasis komputer sangat diperlukan dalam pengolahan data untuk menghasilkan informasi yang dibutuhkan. Elemen yang mendukung dalam sistem komputer yaitu :

1. *Hardware* (perangkat keras), yaitu peralatan yang ada pada sistem komputer secara fisik.
2. *Software* (perangkat lunak), yaitu program-program yang berisis perintah-perintah untuk melakukan pengolahan data, dimana program dibuat sesederhana mungkin.
3. *Brainware, yaitu manusia yang dapat mengoperasikan, serta mengatur sistem komputer*.

2.2.10 Jaringan Komputer

Jaringan komputer dapat diartikan sebagai suatu sistem yang terbentuk dari interkoneksi fasilitas-fasilitas yang dirancang untuk membawa trafik dari beragam sumber telekomunikasi. Jaringan komputer merupakan sumber daya yang dapat digunakan bersama (*shared*) oleh sejumlah *end user* untuk berkomunikasi dengan *user* lain yang lokasinya berjauhan.

LAN (*Local Area Network*) adalah jaringan yang dibatasi oleh area yang relatif kecil seperti perkantoran disebuah gedung atau sebuah sekolah dan biasanya tidak lebih dari 1 kilometer persegi. LAN digunakan untuk menghubungkan komputer yang berada di dalam suatu area kecil yang jarak antar komputernya bisa

mencapai 5 sampai 10 km. LAN biasanya bekerja pada kecepatan mulai 10 Mbps sampai 100 Mbps.

Model hubungan pada LAN terbagi menjadi dua yaitu :

1. *Peer to Peer*

   Model hubungan ini memungkinkan user membagi sumber daya yang ada di komputernya baik berupa *file* dan layanan *printer* serta dapat mengakses sumber daya yang terdapat pada komputer lain. Dalam model hubungan *peer to peer* ini, seluruh komputer mempunyai kemampuan yang sama untuk memakai sumber daya yang tersedia di dalam jaringan dan model ini di desain untuk jaringan berskala kecil dan menengah.

2. *Client Server*

   Model hubungan ini memungkinkan jaringan untuk mensentralisasi fungsi dan aplikasi pada satu atau dua *dedicated file server*. Sebuah *file server* menjadi jantung dari seluruh sistem. Model hubungan ini menyediakan mekanisme untuk mengintegritaskan seluruh komponen yang ada pada jaringan dan memungkinkan banyak pengguna secara bersama-sama memakai sumber daya pada *file server*.

2.2.10.1 Sistem Client Server

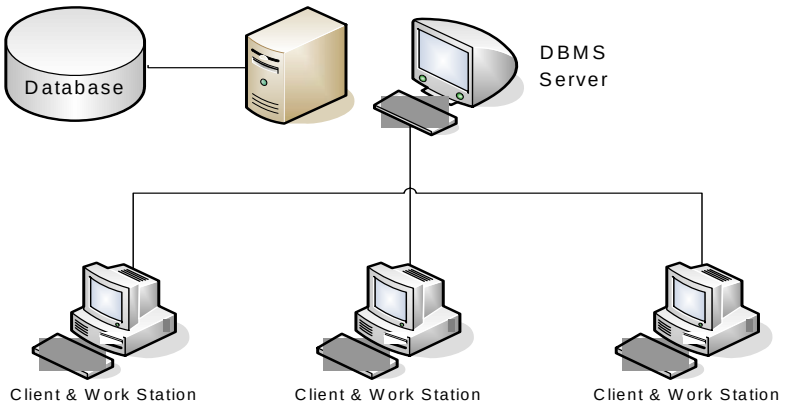
Sistem *client server* terdiri dari dua komponen utama, yaitu *client* dan *server*. Menurut WIL [5] *client server* adalah "*model konektivitas LAN (Local Area Network) yang mengenal adanya client dan server dimana masing-masing memiliki fungsi yang*

berbeda satu sama lain, dan dalam pelaksanaan tugasnya server sebagai pembagi data pada client ".

Kelebihan model hubungan *Client Server* yaitu :

1. Terpusat (sumber daya dan keamanan data di kontrol melalui server)
2. Skalabilitas
3. Fleksibel
4. Teknologi baru dapat dengan mudah terintegrasi ke dalam sistem.
5. Keseluruhan komponen (*client/network/server*) dapat bekerja sama.

Client berisi aplikasi basis data dan *server* berisi DBMS dan basis data. Setiap aktivitas akan ditangani terlebih dahulu oleh *client*. Jika ada proses yang harus melibatkan data yang tersimpan pada basis data, maka *client* harus melakukan *request* dengan *server*. Pada sistem c*lient server* untuk memenuhi kebutuhan tersebut, *client* akan mengirimkan perintah (*message*) *query* pengambilan data. Kemudian *server* yang menerima *message* tersebut akan menjalankan *query* tersebut dan hasilnya akan dikirimkan kembali ke *client*. Dengan begitu, transfer datanya akan menjadi lebih efisien. Adapun bentuk dari Sistem *Client Server* yang sederhana adalah sebagai berikut :



Gambar 2.8. Sistem *Client Server* Sederhana

2.2.11 Topologi Star (Bintang)

Topologi *Star* merupakan topologi yang setiap *node* (*file server*, *workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator*. Data yang dikirim ke jaringan lokal akan melewati *concentrator* sebelum sampai ke tempat tujuan. *Concentrator* akan mengatur dan mengendalikan keseluruhan fungsi jaringan dan bertindak sebagai repeater (penguat aliran data).

Menurut JOG [4] topologi *star* adalah topologi dimana "*beberapa node dihubungkan dengan suatu node pusat (central node) yang membentuk seperti bintang, dimana semua komunikasi diatur langsung oleh central node, yang mana central node ini mengatur semua arus keluar masuk informasi*".

Topologi ini mudah dikembangkan, baik dalam penambahan maupun pengurangan terminal. Banyak terminal yang dapat terhubung tergantung pada jumlah port yang tersedia pada hub yang digunakan. Pada port star ini, hub yang

digunakan akan menjadi titik kritis, sehingga perlu adanya perhatian dan pemeliharaan terhadap hub tersebut.

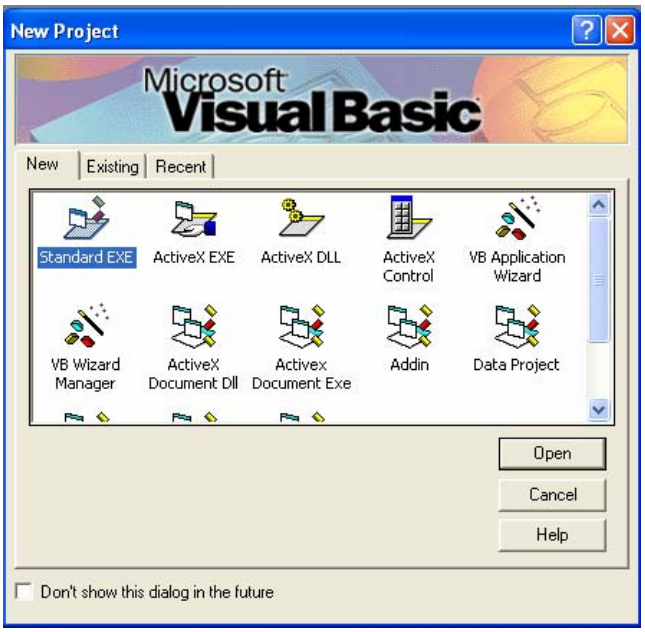
2.2.12 Visual Basic dan MySQL

Microsoft Visual Basic adalah bahasa pemrograman yang digunakan untuk membuat palikasi *Windows* yang berbasis grafis (GUI–*Graphical User Interface*). Visual Basic merupakan *event-driven programming* (pemrograman terkendali kejadian) artinya program menunggu sampai adanya respon dari pemakai berupa *event*/kejadian tertentu (tombol diklik, menu dipilih, dan lain-lain). Ketika *event* terdeteksi, kode yang berhubungan dengan *event* (prosedur *event*) akan dijalankan.

2.2.12.1 IDE Visual Basic

Langkah awal dari belajar Visual Basic adalah mengenal IDE *(Integrated Developement Environment)* Visual Basic yang merupakan Lingkungan Pengembangan Terpadu bagi programmer dalam mengembangkan aplikasinya. Dengan menggunakan IDE programmer dapat membuat user interface, melakukan koding, melakukan testing dan debuging serta menkompilasi program menjadi executable. Penguasaan yang baik akan IDE akan sangat membantu programmer dalam mengefektifkan tugas-tugasnya sehingga dapat bekerja dengan efisien

Sesaat kita aktif di IDE Visual Basic, maka kita akan dihadapkan kepada suatu pilihan terhadap jenis Project yang ingin kita buat sebagaimana mestinya. Sebagai langkah awal dari proses belajar adalah memilih project Standard EXE.



Gambar 2.9. *Dialog box New Project* ditampilkan sesaat anda menjalankan IDE Visual Basic 6.

2.2.12.2 Jendela IDE

IDE Visual Basic 6 menggunakan model MDI *(Multiple Document Interface).* Berikut ini adalah gambar yang menunjukan bagian-bagian dan nama-nama jendela yang dapat tampil pada IDE Visual Basic. Mungkin pada IDE anda hanya ditampilkan sebagian jendela dibandingkan dengan Gambar 1-3 anda tidak perlu terkejut, karena penampilan dari jendela-jendela tersebut dapat diatur dengan perintah pada menu View. Sebagai langkah awal dari proses belajar, tidak semua jendela akan kita gunakan, tetapi hanya beberapa yang penting, sedangkan yang lainnya bersifat khusus.

Adapun jendela-jendela yang perlu kita perhatikan adalah sebagai berikut:

1. *Menu Bar*, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dll
2. *Main Toolbar, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.*
3. *Jendela Project*, jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi anda. Anda dapat menggunakan icon Toggle Folders untuk menampilkan modul-modul dalam jendela tersebut secara di group atau berurut berdasarkan nama. Anda dapat menggunakan Ctrl+R untuk menampilkan jendela project, ataupun menggunakan icon Project Explorer.
4. *Jendela Form Designer,* jendela ini merupakan tempat anda untuk merancang user interface dari aplikasi anda. Jadi jendela ini menyerupai kanvas bagi seorang pelukis.
5. *Jendela Toolbox*, jendela ini berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan user interface.
6. *Jendela Code*, merupakan tempat bagi anda untuk menulis koding. Anda dapat menampilkan jendela ini dengan menggunakan kombinasi Shift-F7.
7. *Jendela Properties*, merupakan daftar properti-properti object yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan *(foreground)* dan warna latar belakang *(background).* Anda dapat menggunakan F4 untuk menampilkan jendela properti.
8. *Jendela Color Palette*, adalah fasilitas cepat untuk mengubah warna suatu object.

9. *Jendela Form Layout*, akan menunjukan bagaimana form bersangkutan ditampilkan ketika runtime.
10. *Jendela Toolbox* merupakan jendela yang sangat penting bagi anda. Dari jendela ini anda dapat mengambil komponen-komponen (object) yang akan ditanamkan pada form untuk membentuk user interface.

Adapun secara garis besar fungsi dari masing-masing intrinsic kontrol tersebut adalah sebagai berikut :

a. *Pointer* bukan merupakan suatu kontrol; gunakan icon ini ketika anda ingin memilih kontrol yang sudah berada pada form.
b. *PictureBox* adalah kontrol yang digunakan untuk menampilkan image dengan format: BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, dan JPEG.
c. *Label* adalah kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.
d. *TextBox* adalah kontrol yang mengandung string yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal, atau banyak baris.
e. *Frame* adalah kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.
f. *CommandButton* merupakan kontrol hampir ditemukan pada setiap form, dan digunakan untuk membangkitkan event proses tertentu ketika pemakai melakukan klik padanya.
g. *CheckBox* digunakan untuk pilihan yang isinya bernilai yes/no, true/false.

h. *OptionButton* sering digunakan lebih dari satu sebagai pilihan terhadap beberapa option yang hanya dapat dipilih satu.
i. *ListBox* mengandung sejumlah item, dan user dapat memilih lebih dari satu (bergantung pada property MultiSelect).
j. *ComboBox* merupakan konbinasi dari TextBox dan suatu ListBox dimana pemasukkan data dapat dilakukan dengan pengetikkan maupun pemilihan.
k. *HScrollBar dan VScrollBar* digunakan untuk membentuk scrollbar berdiri sendiri.
l. *Timer* digunakan untuk proses *background* yang diaktifkan berdasarkan interval waktu tertentu. Merupakan kontrol non-visual.
m. *DriveListBox, DirListBox, dan FileListBox* sering digunakan untuk membentuk dialog box yang berkaitan dengan file.
n. *Shape dan Line* digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan, oval.
o. *Image* berfungsi menyerupai image box, tetapi tidak dapat digunakan sebagai kontainer bagi kontrol lainnya. Sesuatu yang perlu diketahui bahwa kontrol image menggunakan resource yang lebih kecil dibandingkan dengan PictureBox
p. *Data* digunakan untuk data binding
q. *OLE* dapat digunakan sebagai tempat bagi program eksternal seperti Microsoft Excel, Word, dll.

2.2.12.3 Jenis File Pada Visual Basic

Proyek merupakan cikal bakal dari aplikasi dan terdiri dari beberapa *file* yang saling terkait satu sama lain, *file-file* tersebut mempunyai ekstensi yang berbeda-beda sesuai dengan isi dan fungsinya. Sebuah *Proyek* dapat terdiri atas:

a. File. VBF untuk tiap proyek
b. File FRM untuk tiap form aplikasi pada sebuah proyek.
c. File. VBX atau OCX untuk tiap custom pada sebuah proyek.
d. File. BAS untuk tiap modul standar pada proyek.
e. File CLS untuk tiap modul class pada sebuah proyek.
f. File. FRX untuk tiap form pada sebuah proyek yang memiliki control berproperti menggunakan data *binary*.
g. File. RES yaitu sebuah *resource file* pada sebuah proyek.

Dalam hal ini tidak perlu membuat seluruh *file* di atas. Karena apa saja yang dibutuhkan tergantung dari proyek yang sedang dikerjakan.

2.2.12.4 Komponen Pada Visual Basic

Komponen-komponen yang sering dipakai diantaranya:

a. Modul *Form* (File berekstensi .FRM).

   Merupakan kumpulan *form* yang berisi deskripsi secara grafis mengenai tampilan serta kontrol yang digunakan dalam sebuah proyek. Termasuk di dalamnya pengaturan properti yang digunakan oleh masing-masing kontrol. Disampimg itu rutin yang ditulis untuk masing-masing *obyek* pada *form* juga

tersimpan pada *file* ini. Untuk aplikasi yang terdiri dari banyak *form*, masing-masing *form* tersimpan dalam sebuah *file* berekstensi.FRM.

b. Modul Class (File Berekstensi .CLS)

Fungsinya sama dengan modul *form* tetapi terlihat dalam bentuk tampilan. Melalui modul ini bisa dicoba membuat sendiri obyek yang ingin ditampilkan. Masing-masing obyek yang dibuat harus mempunyai properti dan modul tersendiri.

c. Modul Standar (File berekstensi. BAS).

Modul ini berisi deklarasi tipe, konstanta, variabel eksternal *procedure* (prosedur yang menangani kejadian) dan *public procedure*. Biasanya *fil*e ini berisikan perhitungan atau logika yang kompleks.

d. File Recourse (File berekstensi.RES).

File ini berisi *bitmap*, text string, atau data lainnya yang dapat diubah tanpa perlu mengedit kembali rutin. Biasanya *file* ini digunakan untuk mengubah bahasa pada user *interface* alokasi untuk pemakai berbahasa asing (selain yang digunakan oleh si pembuat aplikasi).

e. OLE Custom Control dan Insertable Objects (File berekstensi. OCX)

File ini berisi *custom* dan *Insertable Objects* yang ditambahkan pada proyek anda.

f. VBX Custom Controls (File berekstensi. VBX).

Sama dengan *file* berekstensi .OCX namun hanya digunakan pada Visual Basic 4.0 versi 16 *bit*.

2.2.12.5 *Structure Query Language* (SQL)

SQL merupakan bahasa yang memungkinkan pemakai untuk melakukan akses data ke *DBMS* dan pemakai dapat menggambarkan data yang ingin diperoleh (Hoffman, 2001). SQL juga memungkinkan pemakai untuk melakukan manipulasi data (*input*, *edit*, dan *delete*). Dalam perkembangannya, SQL telah menjadi bahasa standar dalam kegiatan manipulasi data pada *database* dari berbagai *vendor* yang ada di dunia. Pada subbab ini akan diberikan penjelasan tentang dasar penggunaan dari perintah SQL.

Beberapa Perintah yang umum dipergunakan dalam SQL adalah sebagai berikut:

1. SELECT, merupakan perintah untuk melakukan pengambilan data yang berada dalam tabel
2. Conditional, merupakan perintah untuk mengambil data dengan kondisi tertentu, menggunakan “WHERE” dan relational operator seperti “<”, “> “,”=”, ”<>”.
3. Joins, merupakan bentuk perintah untuk melakukan pengambilan data dari dua atau lebih tabel sekaligus.
4. *Aggregate Function* seperti *Sum(), Count(), Avg(), Max(), Min()*. Bila ingin mengetahui berapa jumlah “id_barang” dengan id_kelompok tertentu pada tabel barang.
5. *CREATE TABLE*, perintah ini dipergunakan untuk membuat sebuah tabel baru dengan *field*-nya.

6. *ALTER TABLE*, perintah ini berfungsi melakukan perubahan pada struktur tabel (bukan isi tabel).
7. *INSERT*, merupakan perintah yang dipergunakan untuk menambahkan data pada tabel.
8. *UPDATE*, berfungsi untuk mengubah data pada tabel.
9. *DELETE*, merupakan perintah untuk menghapus data pada tabel baik satu atau seluruhnya. Untuk menghapus data tertentu maka dapat dipergunakan WHERE sedangkan untuk menghapus seluruh isi tabel maka *WHERE* tidak dipergunakan lagi.