

BAB III

LANDASAN TEORI

3.1 DATABASE

Database atau basis data di dalam buku Simarmata & Paryudi (2006:1), sebagai berikut:

- a. Menurut Stephens dan Plew (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data.
- b. Menurut silberschatz, dkk (2002) mendefenisikan basis data sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan.
- c. Menurut Ramakrishnan dan Gehrke (2003) menyatakan basis data sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan.
- d. Menurut McLeod, dkk (2001), adalah kumpulan seluruh sumber daya berbasis komputer milik organisasi.

Jadi dapat disimpulkan *database* adalah koleksi dari data - data yang terkait secara logis dan deskripsi dari data - data tersebut, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi.^[12]

“Database is a self-describing collection of integrated tables”, yang berarti *database* adalah sebuah koleksi data yang menggambarkan integrasi antara tabel yang satu dengan tabel yang lainnya. *“Database is a self-describing”*, disini dijelaskan bahwa struktur data saling terintegrasi dalam suatu tempat yang dikenal sebagai kamus data atau metadata.

Jadi, *database* adalah suatu koleksi data yang saling berhubungan secara logis dan menggambarkan integrasi antara suatu tabel dengan tabel lainnya, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi. Berikut ini merupakan alasan dari penggunaan database :

- Padat.

Tidak perlu lagi membuat arsip kertas dalam ukuran besar.

- Kecepatan

Mesin dapat mendapatkan kembali dan mengubah data jauh lebih cepat daripada yang manusia yang dapat lakukan.

- Mengurangi pekerjaan yang membosankan

Rasa bosan dari proses memelihara arsip – arsip berupa kertas dapat dikurangi.

- Aktual.^[2]

Informasi yang terbaru dan akurat selalu tersedia disetiap waktu ketika dibutuhkan. Terdapat delapan keuntungan dengan menggunakan pendekatan *database*, yaitu:

- Redundansi dapat dikurangkan.
- Ketidak konsistenan dapat dihindari.
- Data dapat dibagikan.
- Standar - standar dapat diselenggarakan.
- Pembatasan keamanan dapat diselenggarakan.
- Integritas dapat dipertahankan.
- Keperluan yang bertentangan dapat diseimbangkan.

- Tersedianya dukungan untuk transaksi.

Pada umumnya data dalam *database* bersifat *integrated* dan *shared*. Maksud dari *integrated* adalah *database* merupakan penggabungan beberapa file data yang berbeda, dengan membatasi pengulangan baik keseluruhan file ataupun sebagian. Pengertian *shared* artinya adalah data individu dalam *database* dapat digunakan secara bersamaan antara beberapa pengguna yang berbeda. ^[4]

3.2 Microsoft Access

Microsoft Access adalah suatu program aplikasi basis data komputer relasional yang digunakan untuk merancang, membuat dan mengolah berbagai jenis data dengan kapasitas yang besar. Aplikasi ini menggunakan mesin basis data *Microsoft Jet Database Engine*, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah Microsoft Office Access 2007 yang termasuk ke dalam *Microsoft Office System 2007*.

Microsoft Access dapat menggunakan data yang disimpan di dalam format *Microsoft Access*, *Microsoft Jet Database Engine*, *Microsoft SQL Server*, *Oracle Database*, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna / *programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para *programmer* yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek. ^[5]

3.2.1 Komponen Utama (Object)

1. Tabel Data

Tabel data adalah objek utama dalam *database* yang digunakan untuk menyimpan sekumpulan data sejenis dalam sebuah objek. Table terdiri atas :

- a. *Field Name* : atribut dari sebuah tabel yang menempati bagian kolom.
- b. *Record* : Isi dari *field* atau atribut yang saling berhubungan yang menempati bagian baris.

2. Query (SQL / Structured Query Language)

Query adalah bahasa untuk melakukan manipulasi terhadap *database*. Digunakan untuk menampilkan, mengubah, dan menganalisa sekumpulan data.

Query dibedakan menjadi 2, yaitu :

- a. *DDL (Data Definition Language)*

Digunakan untuk membuat atau mendefinisikan obyek - obyek *database* seperti membuat tabel, relasi antar tabel dan sebagainya.

- b. *DML (Data Manipulation Language)*

Digunakan untuk manipulasi *database*, seperti : menambah, mengubah atau menghapus data serta mengambil informasi yang diperlukan dari *database*.

3. Form

Form digunakan untuk mengontrol proses masukan data (*input*), menampilkan data (*output*), memeriksa dan memperbaharui data.

4. *Report*

Form digunakan untuk menampilkan data yang sudah dirangkum dan mencetak data secara efektif.

5. **Tipe Data**

Field - field dalam sebuah tabel harus ditentukan tipe datanya. Ada beberapa tipe data dalam Access, yaitu :

1. *Text*

Text digunakan untuk *field alphanumeric* (misal : nama, alamat, kode pos, telp), sekitar 255 karakter tiap fieldnya.

2. *Date/Time*

3. *Currency*

4. *Auto Number*

5. *Yes/No* dll.

6. **Tabel Relationship**

Table Relationship adalah relasi atau hubungan antara beberapa tabel. Relasi antar tabel dihubungkan oleh *primary key* dan *foreign key*. Untuk membuat *relationship* maka masing - masing tabel harus memiliki *primary key* dan *foreign key* untuk dapat menghubungkan antara tabel induk dengan tabel anak. Sehingga diperlukan teknik normalisasi terlebih dahulu sebelum membuat *relationship* antar tabel. Normalisasi merupakan proses pengelompokkan data elemen menjadi tabel yang menunjukkan entitas dan relasinya.

a. Primary Key

Tabel memiliki *primary key*, yaitu suatu atribut yang tidak hanya mengidentifikasi secara unik suatu kejadian tetapi juga mewakili setiap kejadian dari suatu entitas.

b. ForeignKey

Foreign key adalah atribut yang melengkapi *relationship* dan menunjukkan hubungan antara tabel induk dengan tabel anak. *Foreign key* ditempatkan pada tabel anak.^{[1][6]}

Microsoft Access memiliki keterbatasan, yaitu :

Tabel 3.1 Tabel Kemampuan Microsoft Access

<i>File size</i>	2 gigabytes, termasuk semua <i>object</i> di database (data, forms, reports, indices, macros, modules, etc.)
Jumlah tabel	32,768.
<i>Table size</i>	2 gigabytes, dikurangi size dari <i>system objects</i>
<i>Number of characters in a table name</i>	64
<i>Number of characters in a field name</i>	64
<i>Number of fields in a table</i>	255
<i>Number of open tables</i>	2048 Note that this includes internal tables opened by Access.
<i>Number of characters in a text field</i>	255
<i>Number of characters in a memo field</i>	65,535 when text is entered through the UI
	1 gigabyte if text is entered

	<i>programmatically</i>
<i>Number of tables in a query</i>	32
<i>Number of levels of nested queries</i>	50 ^[9]

3.3 Komunikasi Serial

Komunikasi serial ialah pengiriman data secara serial (data dikirim satu persatu secara berurutan), sehingga komunikasi serial lebih lambat daripada komunikasi paralel. Komunikasi Serial dapat digunakan untuk menggantikan Komunikasi Paralel jalur data 8-bit dengan baik. Tidak saja memakan biaya yang lebih murah, namun dapat digunakan untuk menghubungkan dua peralatan yang sangat jauh. Misalnya menumpang pada kabel telpon.

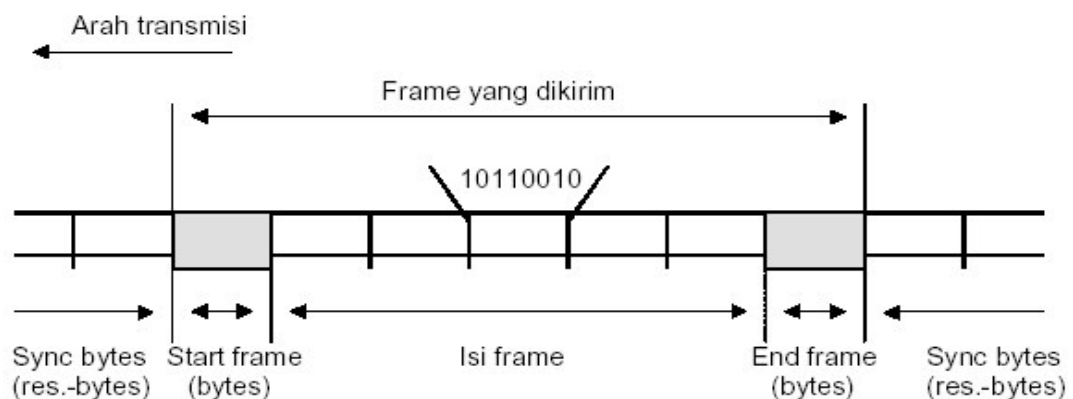
Agar komunikasi serial dapat bekerja dengan baik, data *byte* harus diubah kedalam bit - bit serial menggunakan peralatan yang disebut *shift register parallel-in serial-out*, kemudian data dikirimkan hanya dengan satu jalur data saja. Hal yang serupa dikerjakan pada penerima, dimana penerima harus mengubah bit - bit serial yang diterimanya menjadi data *byte* yang persis seperti data semula pada pengirim, dengan menggunakan *shift register serial-in parallel-out*. Tentu saja jika data serial tersebut dikirim menumpang jalur telpon, maka dibutuhkan peralatan pengubah status digital 0s atau 1s menjadi sinyal suara audio. Peralatan seperti ini kemudian disebut modem (modulator/demodulator).

Modulator sebagai pengubah sinyal digital menjadi sinyal audio, sebaliknya Demodulator adalah sebagai mengubah kembali sinyal audio menjadi sinyal digital. Pada jarak yang sangat dekat, kita dapat menggunakan komunikasi

serial sederhana dan tidak perlu modulasi. Seperti yang dapat kita lihat pada hubungan komputer kita dengan keyboard atau mouse.^[11]

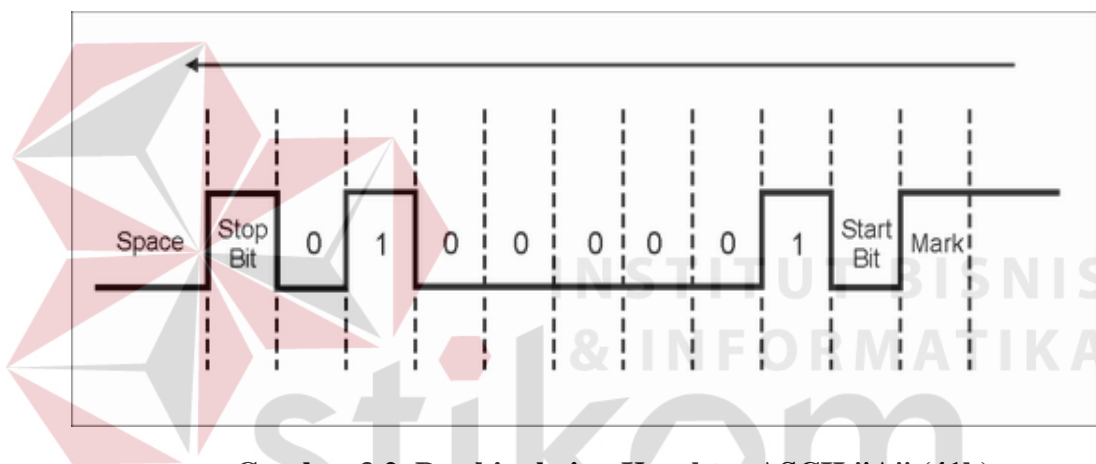
3.3.1 Komunikasi Serial Synchronous dan Asynchronous

Komunikasi data serial mengenal dua buah metode, yaitu *synchronous* dan *asynchronous*. Metode *synchronous* mengirimkan datanya beberapa *byte* atau karakter (atau disebut blok data atau *frame*) sebelum meminta konfirmasi apakah data sudah diterima dengan baik atau tidak. Sementara metode *asynchronous* data dikirim satu *byte* setiap pengiriman. Biasanya tidak dibutuhkan konfirmasi penerimaan data. Dari kedua jenis metode tersebut dapat dipilih dan dilakukan lewat program. Tentu saja dibutuhkan program yang baik dan teliti untuk melakukannya. Namun dewasa ini proses pengiriman data serial tersebut sudah dilakukan oleh sebuah chip tersendiri (*Hardware*). Salah satu chip disebut UART (*Universal Asynchronous Receiver Transmitter*) dan satunya lagi disebut USART (*Universal Synchronous Asynchronous Receiver Transmitter*). Dalam protokol berbeda, *synchronous* memerlukan sinyal tambahan yang digunakan untuk sinkronisasi setiap denyut dari proses transfer.



Gambar 3.1. Sinyal Transmisi Sinkron

Komunikasi data serial *asynchronous* sekarang sudah digunakan demikian luas untuk transmisi yang berorientasi karakter, sementara metode *synchronous* digunakan untuk transmisi yang berorientasi blok. Pada mode *asynchronous*, setiap karakter ditempatkan berada diantara bit start dan bit stop. Bit start selalu satu bit, tapi stop bit bisa satu bit atau dua bit. Start bit selalu 0 (*low*) dan stop bit selalu 1 (*high*). Contohnya, pada gambar 3.2 di mana karakter A (01000001 biner) dibingkai (dikurung) oleh start bit dan satu stop bit.



Gambar 3.2. Pembungkai Karakter ASCII "A" (41h)

Perlu diperhatikan Gambar 3.2 kalau keluar dari standard serial RS232 maka nilai (1) tegangan (-) sedang nilai 0 tegangan (+). Pada komunikasi serial *asynchronous*, peralatan dan modem dapat diprogram untuk menggunakan lebar data 7 atau 8-bit. Tentu saja ditambah dengan Stop bit. Dahulu, sistem karakter ASCII masih terbatas pada data 7-bit, namun sekarang ASCII *extended* sudah lazim menggunakan lebar data 8-bit. Pada peralatan kuno, dengan komponen - komponen yang lambat pula, dibutuhkan stop bit yang agak panjang, hal ini dimaksudkan untuk memberi kesempatan peralatan untuk menangani data yang

telah diterimanya, dan untuk mempersiapkan diri untuk menerima byte berikutnya. Namun sekarang modem PC kita dewasa ini biasanya hanya menggunakan satu bit stop. Jika kita hitung, dengan menggunakan satu bit stop total kita memiliki 10-bit untuk setiap karakter 8-bit. Dengan kata lain setiap karakter 8-bit dikirim bersama-sama start dan stop bit sehingga total menjadi 10-bit, yang artinya ada proses transfer lebih panjang 20% setiap karakternya.^[11]

3.3.2 Data Transfer Rate

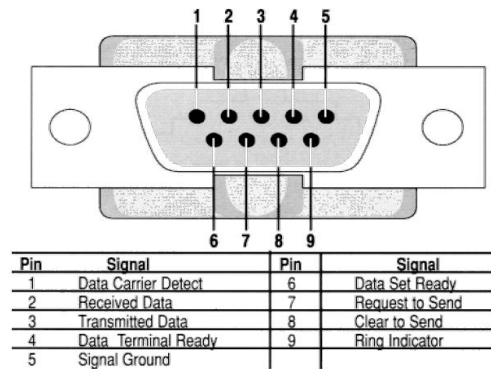
Kecepatan *transfer* data pada komunikasi data serial diukur dalam satuan BPS (bits *persecond*). Sebutan terkenal lainnya adalah *baud rate*. Namun *baud rate* dan bps tidak serta merta adalah sama. Hal ini mengacu kepada fakta bahwa *baud rate* adalah terminology modem dan diartikan sebagai perubahan *signal* dalam satuan bit signal setiap detik. Sedang data *transfer rate* penamaannya mengacu pada jumlah bit dari *byte* data yang ditransfer setiap detik. Sementara itu kecepatan transfer data (data *transfer rate*) pada komputer tergantung pada jenis komunikasi yang diberlakukan atasnya. Seperti contoh, komputer PC-IBM model kuno dapat mentransfer data mulai dari 100 s/d 9600 bps. Namun pada saat sekarang kecepatan komunikasi serial menjadi sangat pesat. 56.000 bps kemudian menjadi standar kecepatan pada modem. Namun para perancang komputer sepakat untuk membatasi kecepatan pada komunikasi serial *asynchronous* hanya setinggi 100.000 bps. Untuk kecepatan yang lebih tinggi mode *synchronous* kemudian menjadi pilihan.^[11]

3.3.3 Port Serial

Komputer PC-IBM dan sejenisnya yang berbasis mikro prosesor x86 (8086, 186, 286, 386, 486, dan pentium) memiliki dua buah (setidaknya satu) port COM. Kedua port COM memiliki konektor standar RS232. Terminal atau konektor yang digunakan untuk mengkonversi level tegangan biasa disebut DB9. Port-port COM tersebut diberi nama dengan COM1 dan COM2. Biasanya port COM ini sekarang digunakan untuk modem (modem sekarang juga sudah banyak menggunakan USB).^[10]

3.3.4 Prinsip Kerja Port Serial

Perangkat yang menggunakan kabel serial untuk komunikasinya dibagi ke dalam dua kategori. Yaitu DCE (*Data Communications Equipment*) dan DTE (*Data Terminal Equipment*). *Data Communications Equipment* adalah perangkat seperti modem, TA adapter, plotter dan lain-lain, sedangkan *Data Terminal Equipment* adalah komputer anda atau terminal. Untuk menjamin terjadinya sebuah transfer data yang cepat dan *realible* antara 2 peralatan, lalu lintas data harus dikoordinasi dengan baik. Tidak seperti printer yang selalu mencetak setiap karakter yang diterimanya. Namun dalam komunikasi serial, bisa saja peralatan tidak memiliki lagi tampungan data yang diterimanya. Sehingga dia harus memberitahukan PC untuk tidak lagi mengirim data. Hingga modem selesai mengerjakan semua tugasnya. Dan kembali memberitahukan PC untuk kembali mengirim data berikutnya setelah modem siap.



Gambar 3.3. Konektor Serial RS232 DB-9

1. DTR (*Data Terminal Ready*)

Saat komputer kita pertama dinyalakan dan *Operating System*-nya telah siap untuk melakukan tugasnya, kemudian mengirimkan sinyal DTR untuk memberitahukan pada peralatan yang mungkin terhubung dengannya (misalnya modem), bahwa komputer telah siap berkomunikasi. Jika terjadi masalah dengan port COM, maka sinyal ini tidak diaktifkan. Sinyal ini adalah aktif rendah.

2. DSR (*Data Set Ready*)

Seperti juga Komputer dengan DTR-nya, saat modem pertama dihidupkan, dan siap berkomunikasi dia akan menyalakan DSR ini menuju ke komputer. Sehingga komputer dapat segera tahu bahwa saat itu ada modem yang terhubung dan siap untuk digunakan. Ini adalah sinyal dari modem ke PC, dan berjenis aktif *low*. Sinyal ini tidak akan dikeluarkan modem, jika modem dalam masalah atau rusak.

3. RTS (*Request To Send*)

Saat komputer kita hendak mengirimkan data kepada *device* seperti modem, maka komputer akan mengirimkan RTS ini menuju modem terlebih

dahulu. Hal tersebut agar modem tahu, bahwa akan ada pengiriman data dari komputer kepadanya, dan modem segera bersiap-siap untuk menerima data. RTS adalah sinyal aktif low dari komputer ke device seperti modem.

4. CTS (*Clear To Send*)

Dalam merespon RTS, ini modem menyalakan sinyal CTS. Saat komputer mengirimkan RTS, maka modem bersiap - siap untuk menerima data, dan jika ruang untuk menerima data tersebut sudah ada dan cukup, baru sinyal tersebut dikirimkan pada komputer untuk segera mengirimkan datanya.

5. DCD (*Data Carrier Detect*)

Ini adalah sinyal yang dikeluarkan oleh modem kepada PC, untuk menginformasikan PC bahwa modem mendeteksi adanya *carrier* yang valid dan adanya kontak / hubungan dengan modem lain yang jauh. (Biasanya sinyal ini diberikan setelah komunikasi dengan modem lain tersambung).

6. RI (*Ring Indikator*)

Sinyal ini diberikan oleh modem kepada komputer untuk menginfomasikan bahwa saluran telah dihubungi dan berbunyi. Sinyal ini muncul bersamaan saat telpon berbunyi. Sinyal ini seperti saat telpon kita sedang tidak digunakan, dan seseorang dari sana menelfon,lalu pesawat telfon kita berbunyi, meminta kita untuk segera mengangkatnya, dan berkomunikasi. Dalam modem bunyi tersebut digantikan dengan sinyal. Saat komputer kita dalam keadaan mati, setting BIOS dalam komputer

memungkinkan komputer untuk hidup sendiri saat adanya panggilan semacam ini.

7. RXD (*Receive Data*)

Sinyal RXD adalah sinyal data yang diterima dari perangkat lainnya. Diakhir perangkat lain, sinyal ini didapat dari sinyal TXD (Transmit data). Harus diketahui bahwa sinyal TXD dan RXD, bersama ground, mereka hanya menghendaki saluran untuk komunikasi data. Semua saluran yang lain digunakan untuk mengontrol atau handshaking.

8. TXD (*Transmit Data*)

Sinyal TXD adalah sinyal data actual yang dikirimkan dari satu perangkat ke perangkat lainnya. Sinyal ini masuk ke RXD yang berhubungan dengan pin RXD konektor itu.^[10]

3.4 Visual Basic 6.0

Visual basic 6.0 adalah salah satu aplikasi untuk membuat *system* informasi *database*. Visual basic 6.0 ini adalah terusan dari Visual Basic 1 sampai 5. Walau agak tertinggal, tapi aplikasi *software* ini masih digunakan untuk pengimplementaisannya di dalam pembuatan sistem *database*. Visual basic 6.0 ini pun berkembang menjadi Visual Studio, Visual Basic 2008 sampai Visual Basic. net.

Visual Basic 6.0 ini sering juga di gunakan dalam perhitungan gaji, penjualan barang dan lain-lain. Visual Basic ini *support* dengan *operating system* *windows*, dan untuk *database*-nya bisa menggunakan Microsoft Access, SQL Server dan Oracle.

Dalam menu Visual Basic terdapat komponen - komponen yang tentu saja di perlukan untuk menentukan *coding* atau *syntax* yang akan digunakan. karena salah titik atau koma dalam Visual Basic akan mengakibatkan program sistem *debug* atau *error*. Dalam pengimplementasian aplikasi program visual basic ada 5 pokok yang penting di dalam komponennya. yaitu :

A. *Project* : *Project* adalah nama file yang akan dikenal dalam pemanggilan program aplikasi.

B. Sub Form/MDI *Form* : *Form* utama dalam Microsoft Visual Basic. karena MDI *form* ini mewakili *form* - *form* yang ada di dalam aplikasi *database* yang kita buat

C. *Form* : untuk pengimplementasian sistem yang dapat melakukan beberapa perintah eksekusi sesuai yang di inginkan. dalam *form* ini juga kita dapat mengkodekan aplikasi untuk memanggil *database*, menampilkan data, menghapus data, mengupdate data, mengedit data dan mencetak data yang telah kita eksekusi.

D. *Coding/syntax* : perintah-perintah dalam bahasa program aplikasi Visual Basic yang telah ditentukan. Dalam *coding* ini juga yang mempengaruhi sistem dapat berjalan atau tidak.

E. *Report* : Hasil cetak yang diinginkan *user* yang digunakan sebagai *output* dari program.

F. *Module* : Suatu perintah untuk mengkoneksikan antara program Visual Basic dengan *database* yang akan di gunakan di dalam mengaplikasikan program.^{[3][7]}

3.5 Crystal Report

Menurut Daryanto mengemukakan bahwa Crystal Report, yaitu: “*Crystal Report merupakan tool yang sering digunakan untuk membuat laporan*”.^[3]

Menurut Madcom dalam bukunya yang berjudul Database Visual Basic 6.0 Dengan Crystal Report menyebutkan bahwa : “*Crystal repots merupakan program khusus untuk membuat laporan yang terpisah dari program Microsoft Visual Basic tetapi keduanya dapat dihubungkan (linkagge)*”.

Dari uraian diatas dapat disimpulkan bahwa Crystal Report merupakan program khusus untuk membuat laporan yang terpisah dari program Microsoft Visual Basic, tetapi keduanya dapat dihubungkan. Pada standar *reprts expert* terdiri dari delapan langkah yaitu:

a. Tab Data

Tab data ini harus menentukan tabel atau *query* yang akan digunakan untuk membuat laporan.

b. Tab *Fields*

Tab ini dapat menentukan *field - field* atau kolom - kolom yang akan ditampilkan dalam laporan.

c. Tab *Sort*

Tab ini dapat mengurutkan atau mengelompokkan data - data yang akan ditampilkan dalam laporan.

d. Tab total

Tab ini dapat menampilkan subtotal berdasarkan kriteria atau pengelompokan berdasarkan *fields* yang telah ditentukan pada tab sebelumnya.

e. Tab TopN

Tab ini dapat menentukan data yang akan ditampilkan dengan cara memilih N terbesar dari data yang ada.

f. Tab *Graph*

Tab ini dapat membuat dan memilih *type* grafik yang diinginkan.

g. Tab *Select*

Tab ini dapat menyaring atau memfilter *record - record* yang akan ditampilkan dalam laporan.

h. Tab *Style*

Tab ini dapat memilih bentuk tampilan laporan dan menuliskan judul laporan yang diinginkan. Kelebihan yang terdapat dalam Crystal Report adalah hasil cetakan Crystal Report lebih baik dan lebih mudah, karena pada Crystal Report banyak tersedia objek maupun komponen yang mudah digunakan.

Hubungan crystal report dengan visual basic yaitu percetakan/ hasil akhir dilakukan dalam sebuah form maka objek crystal report yang terdapat didalam form harus dihubungkan dengan file crystal report yang sudah jadi. Berkaitan dengan tampilan laporan ada beberapa tombol yang mendukung, yaitu:

1. Tab *Design*

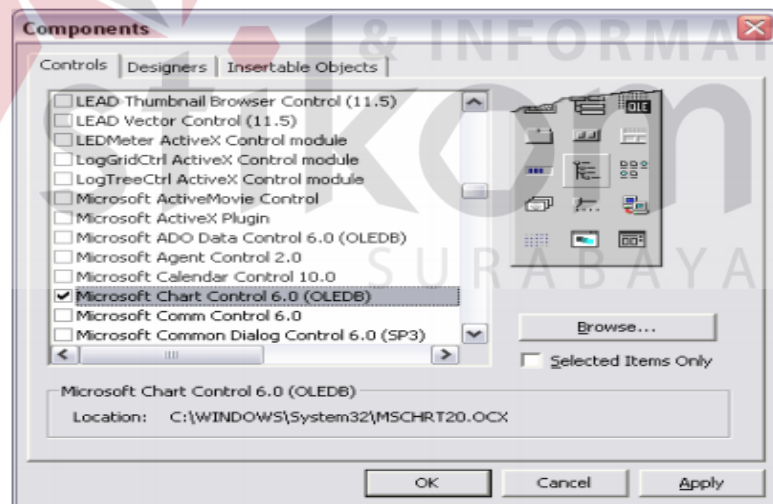
Fungsi dari tab desain ini adalah untuk menampilkan desain laporan yang dibuat.

2. Tab *Preview*

Fungsi dari tab *preview* adalah untuk menampilkan bentuk / hasil dari laporan.^[8]

3.6 MSChart

Adalah sebuah komponen yang terdapat pada Visual Basic 6.0 yang digunakan untuk menampilkan data dalam bentuk grafik. Pengambilan komponen MSChart dapat dilakukan dengan cara menambah komponen MSChart pada *Project*, menu *Project >> Components*. Pilih Microsoft Chart Control 6.0 (OLEDB) seperti gambar 3.7 dibawah ini.



Gambar 3.4. Penambahan komponen MSChart pada VB 6.0

Terdapat beberapa *property* yang harus diatur kedalam MSChar untuk membuat sebuah grafik. *User* hanya perlu mengaturnya dalam properti MSChart

atau mengaturnya didalam program. Properti – properti yang harus diatur diantaranya adalah :

1. *Column Count*

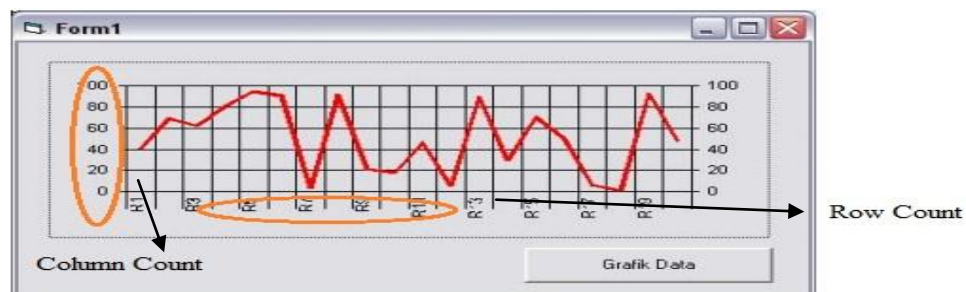
Digunakan sebagai acuan nilai dari data yang akan dimasukkan kedalam sebuah grafik. Misalnya saja tahun dan banyaknya penduduk sebuah provinsi. (misalnya saja dalam grafik kepadatan penduduk pulau Jawa. Yang dijadikan sebagai *column count* adalah jumlah kepadatan penduduk). Tapi hal ini juga tergantung dari kebutuhan *user* dalam mengolah data.

2. *Row Count*

Digunakan sebagai acuan penambahan data. (misalnya saja dalam grafik kepadatan penduduk pulau Jawa. Yang dijadikan sebagai *row count* adalah tahun dari data). Tapi hal ini juga tergantung dari kebutuhan *user* dalam mengolah data.

3. *Chart Type*

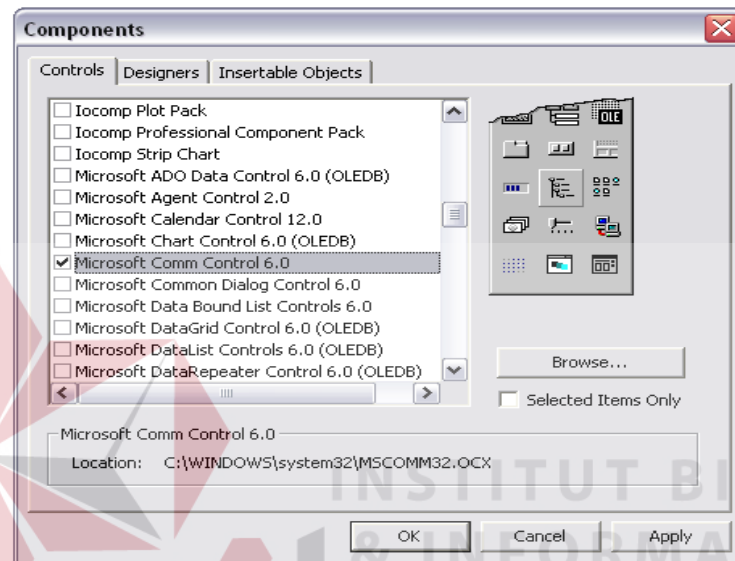
Terdapat beberapa tipe grafik yang dapat dipilih di MSChart, misalnya saja grafik batang, grafik garis, grafik point, diagram lingkaran, panah, dan masih banyak lagi yang dapat dipilih oleh *user* sesuai dengan kebutuhan *user*. Bentuk dari chart juga dapat dalam bentuk 3D maupun 2D.



Gambar 3.5. Hasil grafik pada MSChart

3.7 Komunikasi Serial Visual Basic 6.0

Dalam berkomunikasi dengan “dunia luar” atau perangkat lain di luar komputer, *Visual Basic 6.0* menyediakan komponen **MS Comm Control 6.0**, sebagai media komunikasi. Untuk menambahkan komponen ini pada *Visual Basic*, pilih **Project** → **Components**. Setelah itu akan muncul tampilan sebagai berikut :



Gambar 3.6. Penambahan Komponen *oncomm* pada Visual Basic 6.0

Fungsi dari komponen ini adalah sebagai berikut :

- a. Mengadakan hubungan dengan *serial port* PC.
- b. Berhubungan dengan alat komunikasi lain (contoh: *modem*).
- c. Melakukan pertukaran data.
- d. Memonitor dan merespon *event* dan *error* yang terjadi pada hubungan *serial*.

Untuk mengadakan suatu komunikasi serial antara 2 peralatan, kita harus melakukan beberapa langkah.

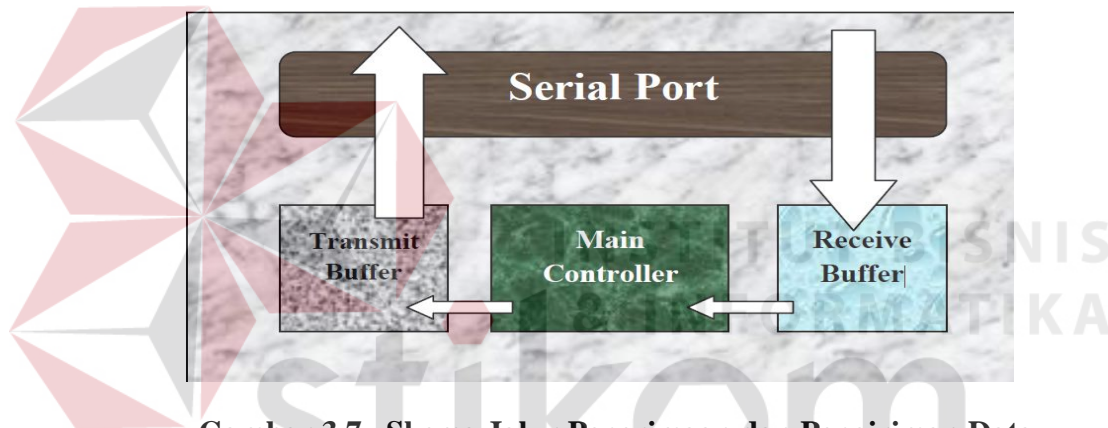
1. Membuka *serial port*.
2. Mengatur *serial device*.

3. *Setting Receive and Transmit Buffer Properties.*

4. *Managing Receive and Transmit Buffer.*^[10]

3.7.1 Membuka Serial Port

Pada komunikasi serial, bit - bit data yang masuk dari dunia luar ke dalam komputer melalui *serial port* akan ditampung dulu di *receive buffer* sebelum akan dieksekusi oleh *main controller*. Demikian pula sebelum dikirimkan ke luar, data akan ditampung dulu di *transmit buffer*. Gambar skema lengkapnya dapat dilihat pada gambar di bawah ini.



Gambar 3.7. Skema Jalur Penerimaan dan Pengiriman Data

Sebelum membuka *serial port*, dilakukan pengaturan protokol komunikasi serial dengan *property* *MSComm*. Menentukan nomor *port* komunikasi menggunakan *CommPort* dan menentukan *baud rate*, *parity*, data *bits*, *stop bits* menggunakan *property setting*. Sedangkan untuk membuka *serial port* cukup menggunakan perintah *PortOpen*. Sehingga kode program akan tertulis sebagai berikut :

```
MSComm1.ComPort = 2
MSComm1.Settings = "9600,N,8,1"
MSComm1.PortOpen = True
```

3.7.2 Mengatur Serial Device

Pada tahap ini kita perlu memastikan bahwa pengaturan protokol komunikasi serial yang digunakan pada peralatan lain yang kita akses, sesuai dengan pengaturan pada komputer yang kita pakai.

3.7.3 Setting Receive dan Transmit Buffer Properties

Ada beberapa *property* dari *receive buffer* dan *transmit buffer* (*property* dari **MSComm**) yang perlu kita atur.

- a. *InBufferSize* : mengatur ukuran *receive buffer*
- b. *OutBuffer Size* : mengatur ukuran *transmit buffer*
- c. *Rthreshold* : menentukan jumlah karakter yang diterima oleh *receive buffer* sebelum *OnComm event* dipicu
- d. *Sthreshold* : menentukan jumlah karakter yang diterima oleh *transmit buffer* sebelum *OnComm event* dipicu. Nilai 0 berarti tidak pernah dipicu, sedangkan nilai 1 berarti dipicu setiap satu karakter.
- e. *InputLen* : menentukan jumlah karakter yang dibaca CPU dari *receive buffer*
- f. *InputMode* : menentukan tipe data input yang akan dibaca CPU.
comInputModeText untuk data *string*/teks dan *comInput Mode Binary* untuk data biner.

3.7.4 Managing Receive dan Transmit Buffer

Untuk menampilkan data dari peralatan lain ke dalam aplikasi VB, digunakan properti *input*, sedangkan untuk mengirim data dari aplikasi VB ke peralatan lain digunakan properti *output*. Contoh struktur kode untuk *input* dan *output*.^[11]

```

TxtDisplay.Text = MSComm1.Input ( Contoh Input )
MSComm1.Output = "Data String" ( Contoh Output )

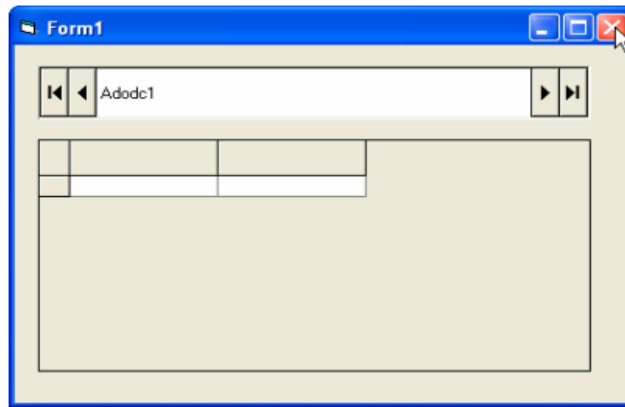
```

3.8 Koneksi antara Database dengan Visual Basic 6.0

Terdapat beberapa cara dalam membuat koneksi antara *database* terutama dalam hal ini adalah Microsoft Access dengan Visual Basic. Diantaranya adalah dengan menggunakan komponen ADODC yang terdapat pada Visual Basic, atau menggunakan koneksi ODBC tanpa melalui DSN, juga dapat melalui model DSN dengan menggunakan objek ODBC. *User* dapat menentukannya sesuai dengan kebutuhan dan juga keefektifan sebuah aplikasi. Karena setiap cara memiliki kelemahan dan kelebihan masing – masing.

1. Menggunakan ADODC

Dengan cara menambahkan komponen ADODC yang terdapat pada Visual Basic 6.0. Biasanya *database* dapat terkoneksi langsung pada komponen datagrid yang ada pada Visual basic dengan cara mengaturnya pada properti datagrid. Pengaturan dilakukan pada properti ADODC.



Gambar 3.8. Contoh penggunaan ADODC

2. Menggunakan koneksi ADODB tanpa DSN

Penggunaan metode ADODB tanpa DSN dapat dilakukan dengan cara membuat program koneksi pada *database* pada modul Visual Basic agar dapat diakses oleh semua *form* pada Visual Basic.

```
Sub koneksi(flag As Boolean)
Set conn = New ADODB.Connection

On Error GoTo Handler
If flag = True Then
    conn.Open "PROVIDER=MICROSOFT.JET.OLEDB.4.0;PERSIST
              SECURITY INFO=FALSE;DATA SOURCE=" & App.Path &
              "\batan.mdb"
    conn.CursorLocation = adUseClient
Else
    conn.Close
End If
Exit Sub

Handler:
    MsgBox Err.Description, vbCritical, "Error"
End Sub
```

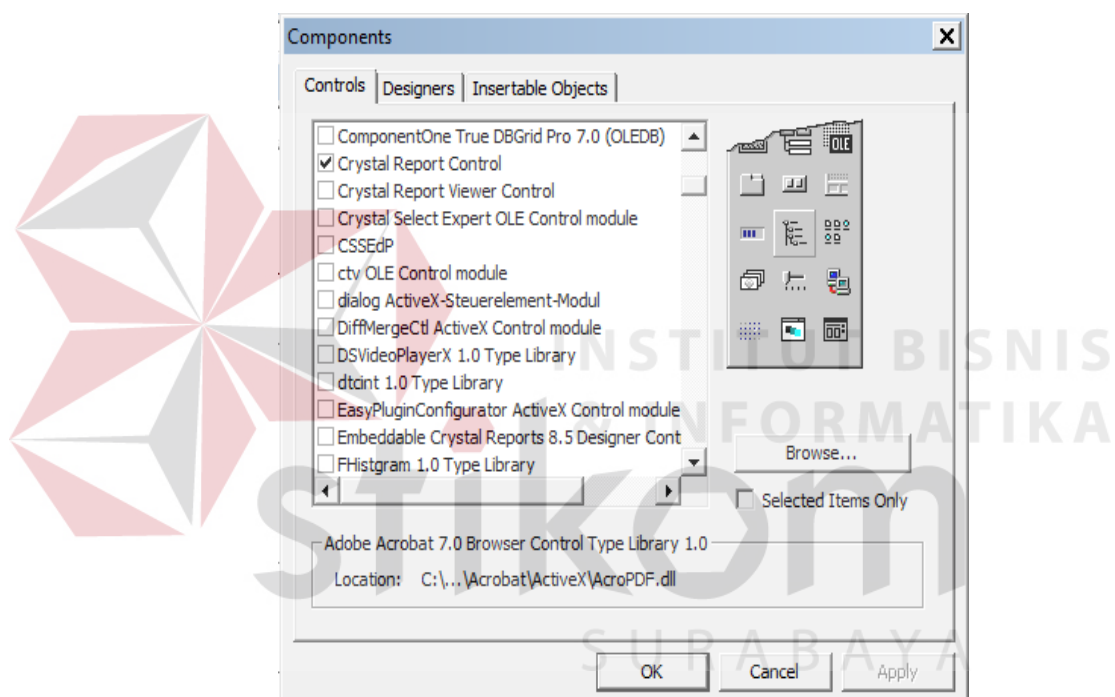
3. Menggunakan koneksi ADODB tanpa DSN

Metode ini memerlukan pengaturan yang terdapat pada Administrative Tool→*Data Source* (ADODC). Dan koneksi dapat dilakukan pada program Visual Basic.

3.9 Koneksi antara Crystal Report dengan Visual Basic 6.0

Koneksi Crystal Report dengan Visual Basic dilakukan dengan menambahkan komponen Crystal Report yang ada pada Visual Basic 6. Selanjutnya diprogram sesuai dengan kebutuhan *user*. Pengambilan komponen *Crystal Report* pada Visual Basic dapat dilakukan dengan cara :

Klik kanan pada *toolbox* → *components* → pilih Crystal Report Control seperti pada gambar 3.9 dibawah ini.^[8]



Gambar 3.9 Pengambilan Component Crystal Report pada VB 6.0