BAB II
LANDASAN TEORI

2.1 Perangkat Keras

2.1.1 RFID

RFID (*Radio Frequency Identification*) merupakan sebuah metode identifikasi dengan menggunakan sarana *Tag RFID* untuk mengambil data secara jauh menggunakan frekuensi radio. Data yang dikirimkan oleh *RFID* berupa kode-kode yang bertujuan untuk mengenali objek. *RFID* ini terdiri dari dua bagian yakni *RFID Reader* atau pembaca *RFID* dan *Tag RFID*.

2.1.1.1 RFID Reader

RFID Reader dapat juga disebut *interrogator* yaitu alat pembaca *Tag RFID* yang kompatibel. Ketika *Tag RFID* didekatkan maka *Tag* yang terbaca didekatnya akan memberikan respon dengan mengirimkan identitas *Tag*. Identitas *Tag* yang dibaca tersebut dikirimkan ke komputer melalui *port* serial komputer.

Pembaca *Tag* ini berkomunikasi dengan *Tag* tanpa *line of sight*. Jadi pembaca berkomunikasi dengan *Tag* menggunakan gelombang radio. Menurut bentuknya, *reader* dibedakan menjadi 2, yaitu *reader* bergerak seperti peralatan genggam yang dapat berpindah-pindah dan stasioner seperti peralatan pendeteksi barang yang berada di *super market* yaitu *point-of-sale*. Selain itu pembaca juga dapat dibedakan berdasarkan kapasitas penyimpanannya, kemampuan pemrosesan serta frekuensi pembacaannya.



Gambar 2.1 *RFID Reader*

2.1.1.2 Tag RFID

Sistem *RFID* merupakan suatu sistem identifikasi otomatis yang bertujuan untuk memungkinkan data ditransmisikan oleh peralatan *portable* yang disebut *Tag*. Kemudian *Tag* tersebut dibaca oleh suatu *reader RFID* dan diproses menurut kebutuhan dari aplikasi tertentu. Data yang ditransmisikan oleh *Tag* dapat menyediakan in*form*asi identifikasi atau lokasi atau hal-hal khusus tentang produk-produk ber *Tag*, seperti harga, warna, tanggal pembelian dan lain-lain.

Sebuah *Tag RFID* atau dapat disebut *transponder* terdiri dari sebuah *microchip* dan sebuah antena. *Microchip* itu sendiri dapat berukuran kecil. *Chip* tersebut menyimpan nomor seri yang unik atau in*form*asi lainnya tergantung kepada tipe memorinya. Tipe memori digolongkan menjadi *read-only*, *read-write* atau *write-onceread-many*. Antena yang terpasang pada *microchip* mengirimkan in*form*asi ke *reader RFID*. Antena dengan rentang yang lebih besar tingkat pembacaan yang lebih jauh. *Tag* tersebut terpasang atau tertanam dalam objek yang akan diidentifikasi. *Tag* dapat di-*scan* dengan *reader RFID* bergerak maupun stasioner.

Tag RFID sangat bervariasi dalam bentuk dan ukuran. Untuk bentuk itu sendiri ada yang terbuat dari plastik keras, biasanya digunakan untuk identifikasi pengenal karyawan. Selain itu ada pula yang berbentuk gantungan kunci, dapat digunakan untuk kunci kendaraan. Untuk ukuran *Tag* yang kecil dapat ditanamkan di bawah kulit manusia ataupun dibawah kulit hewan.



Gambar 2.2 *Tag RFID*

Terdapat beberapa jenis *Tag* menurut cara kerjanya, yaitu:

a. *Tag* pasif yaitu *Tag* paling sederhana, sehingga *Tag* ini bekerja dengan induksi listrik pada antena yang disebabkan adanya frekuensi radio *scanning* yang masuk. Induksi tersebut cukup untuk mengirimkan respon balik. Karena tidak menggunakan asupan daya maka *Tag* ini sederhana dan memiliki ukuran yang kecil dan tipis. Saat ini ukuran *Tag* terkecil sebesar 4 mm. Karena *Tag* ini sederhana maka *Tag* hanya mengirimkan data berupa nomor saja. Jarak pembacaannya pun tidak begitu jauh.

b. *Tag* semi-aktif yaitu *Tag* yang telah disuplai daya tetapi tidak dapat menginisiasi dengan *reader*. asupan daya yang dipakai *Tag* ini digunakan untuk melakukan fungsi yang lain seperti mamantau keadaan lingkungan dan mencatu bagian elektronik internal *Tag* serta memfasilitasi penyimpanan informasi. *Tag* ini tidak secara aktif memberikan sinyal ke *reader*, *Tag* ini tetap diam hingga menerima sinyal dari *reader*.

c. *Tag* Aktif mempunyai asupan daya yang sama dengan *Tag* semi-aktif. Pengiriman sinyal dilakukan secara kontinyu. Untuk jarak pembacaannya, *Tag* ini dapat dibaca hingga jarak yang cukup jauh. *Tag* aktif mempunyai memori yang besar yang dapat dibaca dan ditulis sehingga dapat memuat banyak informasi ketika terbaca.

2.1.2 Frekuensi RFID

Pemilihan frekuensi radio merupakan kunci karakteristik operasi sistem *RFID*. Frekuensi sebagian besar ditentukan oleh kecepatan komunikasi dan jarak baca terhadap *Tag*. Secara umum tingginya frekuensi mengindikasikan jauhnya jarak baca. Frekuensi yang lebih tinggi mengindikasikan jarak baca yang lebih jauh.

Pemilihan tipe frekuensi juga dapat ditentukan oleh tipe aplikasinya. Aplikasi tertentu lebih cocok untuk salah satu tipe frekuensi dibandingkan dengan tipe lainnya karena gelombang radio memiliki perilaku yang berbeda-beda menurut frekuensinya. Sebagai contoh, gelombang LF memiliki kemampuan penetrasi terhadap dinding tembok yang lebih baik dibandingkan dengan gelombang dengan frekuensi yang lebih tinggi, tetapi frekuensi yang lebih tinggi memiliki laju data (*data rate*) yang lebih cepat. Kombinasi dari level-level frekuensi dan daya yang dibolehkan menentukan rentang fungsional dari suatu aplikasi tertentu seperti keluaran daya dari *reader*.

Berikut ini adalah empat frekuensi utama yang digunakan oleh sistem *RFID*.

a. Band LF (*Low Frequency*) berkisar dari 125 KHz hingga 134 KHz. *Band* ini paling sesuai untuk penggunaan jarak pendek (*short-range*) seperti sistem anti pencurian, identifikasi hewan dan sistem kunci mobil.

b. Band HF (*High Frequency*) beroperasi pada 13.56 MHz. Frekuensi ini memungkinkan akurasi yang lebih baik dalam jarak tiga kaki dan karena itu dapat mereduksi resiko kesalahan pembacaan *Tag*. Sebagai konsekuensinya *band* ini lebih cocok untuk pembacaan pada tingkat item (*item-level reading*). *Tag* pasif dengan frekuensi 13.56 MHz dapat dibaca dengan laju 10 to 100 *Tag* perdetik pada jarak tiga kaki atau kurang. *Tag RFID* HF digunakan untuk pelacakan barang-barang di

perpustakaan, toko buku, kontrol akses gedung, pelacakan bagasi pesawat terbang, pelacakan item pakaian.

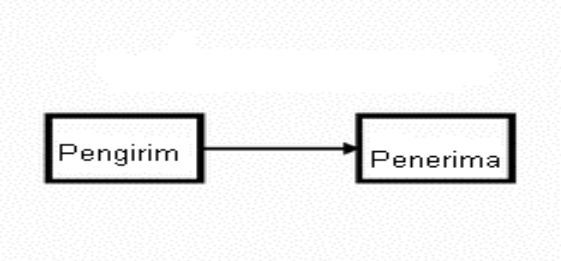
c. *Tag* dengan band UHF (*Ultra High Frequency*) beroperasi disekitar 900 MHz dan dapat dibaca dari jarak yang lebih jauh dari *Tag* HF, berkisar dari 3 hingga 15 kaki. *Tag* ini lebih sensitif terhadap faktor-faktor lingkungan daripada *Tag-Tag* yang beroperasi pada frekuensi lainnya. Band 900 MHz muncul sebagai *band* yang lebih disukai untuk aplikasi rantai supply disebabkan laju dan rentang bacanya. *Tag* UHF pasif dapat dibaca dengan laju sekitar 100 hingga 1.000 *Tag* perdetik. *Tag* ini umumnya digunakan pada pelacakan kontainer, truk, trailer, terminal peti kemas, serta telah diadopsi oleh peritel besar dan Departemen Pertahanan Amerika Serikat. Sebagai tambahan, di Amerika Serikat, *band* MHz digunakan untuk mengidentifikasi isi kontainer dalam area komersial dan industri untuk meningkatkan ketepatan waktu dan akurasi transmisi data. Menurut FCC penggunaan semacam itu menguntungkan perusahaan pengapalan komersial dan memberikan manfaat keamanan yang signifikan dengan dimungkinkannya seluruh isi kontainer teridentifikasi dengan mudah dan cepat serta dengan dapat diidentifikasinya kerusakan selama pengapalan.

d. *Tag* yang beroperasi pada frekuensi gelombang mikro, biasanya 2.45 dan 5.8 GHz, mengalami lebih banyak pantulan gelombang radio dari obyek-obyek didekatnya yang dapat mengganggu kemampuan *reader* untuk berkomunikasi dengan *Tag*. *Tag RFID* gelombang mikro biasanya digunakan untuk manajemen rantai *supply*.

2.1.3 Transmisi Data

2.1.3.1 Transmisi Data Serial

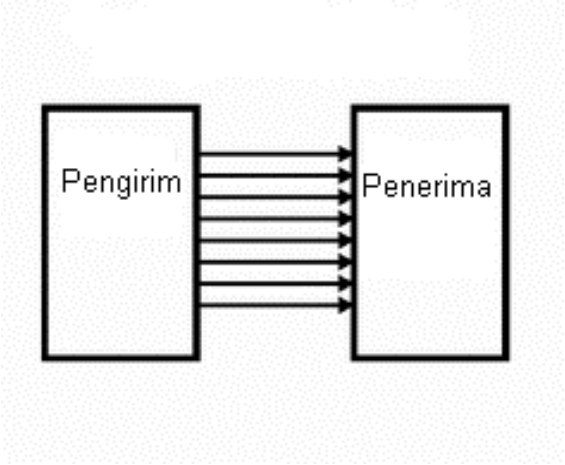
Transmisi serial dapat menggantikan transmisi paralel. Dengan menggunakan transmisi serial pengiriman data jarak jauh menjadi lebih efektif dibanding dengan transmisi paralel. Saluran serial mengirimkan setiap karakter per elemen sehingga hanya diperlukan satu atau dua penghantar, yaitu kirim data (Txd) dan terima data (Rxd). Pada transmisi serial ini membutuhkan waktu yang relatif lebih lama dibanding transmisi serial karena pengiriman data per bit secara beruntun.



Gambar 2.3 Transmisi data serial

2.1.3.2 Transmisi Data paralel

Pada transmisi paralel semua saluran atau penghantar digunakan untuk mengirimkan data dari perangkat luar ke komputer atau dari komputer ke perangkat luar bersamaan menggunakannya. Transmisi paralel dapat mengirim data dengan cepat, namun jarak menjadi hambatan karena tidak mungkin menggunakannya dengan jarak yang jauh atau panjang. Untuk mengirim data dengan jarak yang jauh maka digunakanlah transmisi serial.



Gambar 2.4 Transmisi data paralel

2.2 Perangkat Lunak

2.2.1 Visual Basic 6.0

2.2.1.1 Pengenalan Visual Basic 6.0

Visual Basic 6.0 merupakan salah satu bahasa pemrograman yang dapat digunakan untuk menyusun dan membuat program aplikasi pada lingkungan sistem operasi *Windows*. Program aplikasi dapat berupa *program database*, program grafis dan lain sebagainya. Di dalam *Visual Basic 6.0* sudah terdapat komponen-komponen yang sangat membantu program aplikasi.

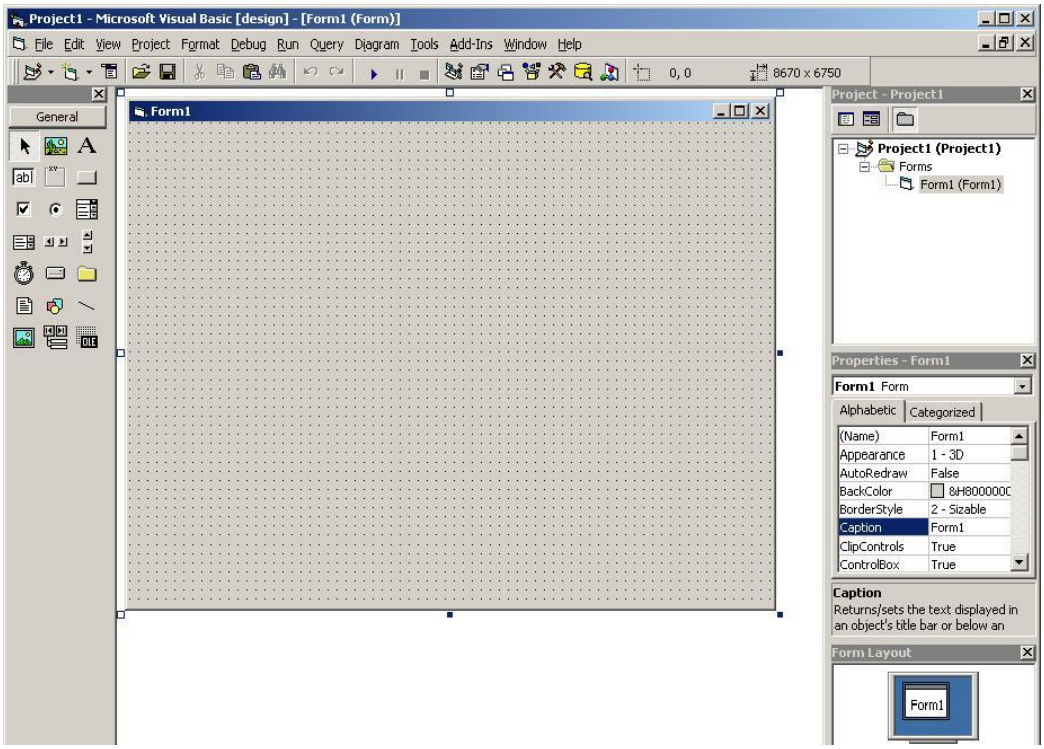
Untuk dapat menyusun dan membuat suatu program aplikasi dari *Visual Basic 6.0*, tentunya *user* harus mengetahui fasilitas-fasilitas yang disediakan oleh *Visual Basic 6.0* agar proses pembuatan program aplikasi dapat berjalan dengan lancar.

a. Pembuatan Project

Dalam pembuatan sebuah program aplikasi harus diperhatikan dalam langkah-langkah pembuatannya. Pada saat membuka *Visual Basic 6.0* tampilan pertama adalah memilih *project* yang digunakan, yaitu *standard.exe*. Berikut ini gambar pada saat *Visual Basic 6.0* dijalankan:



Gambar 2.5 Pilihan jenis New Project



Gambar 2.6 Tampilan *IDE Visual Basic*

b. Menu

Visual Basic 6.0 mempunyai beberapa fasilitas yang dapat digunakan untuk membantu pengguna dalam membuat program aplikasi, salah satunya adalah menu. *Visual Basic 6.0* mempunyai tigabelas menu yang dapat digunakan dan masing-masing menu mempunyai fungsi yang berbeda.



Gambar 2.7 *Menu Visual Basic*

c. Toolbar

Pada prinsipnya *toolbar* berfungsi sama dengan menu, hanya saja berbeda tampilan. Untuk dapat menggunakan fasilitas menu, harus memilih menu yang dipilih terlebih dahulu dan selanjutnya memilih sub menu yang akan digunakan. Pada toolbar, pengguna tinggal memilih *icon* yang terdapat pada masing-masing *toolbar*.



Gambar 2.8 *Toolbar Visual Basic*

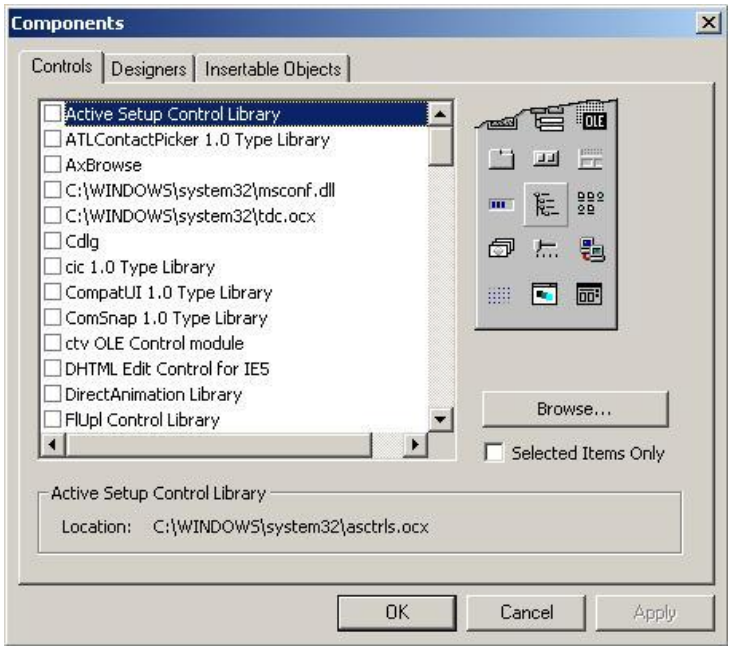
d. Toolbox

Toolbox merupakan tempat kontrol-kontrol yang akan digunakan untuk membantu pembuatan program aplikasi. Secara *default*, pada *toolbox* hanya terdapat kontrol-kontrol seperti yang terdapat pada gambar di bawah ini:



Gambar 2.9 *Toolbox Visual Basic*

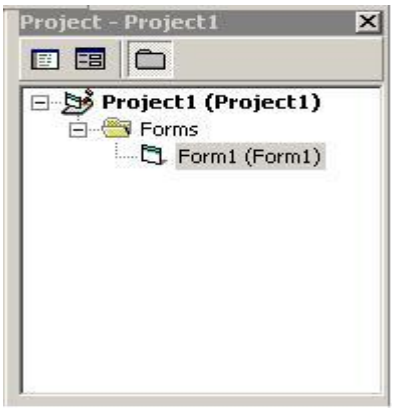
Kontrol-kontrol pada *toolbox* dapat ditambah. Untuk dapat menambah kontrol-kontrol pada *toolbox,* klik kanan bagian *toolbox* lalu memilih *components.* Lalu pengguna dapat menambahkan kontrol yang sesuai dengan kebutuhan.



Gambar 2.10 Komponen Objek Kontrol

e. Project Explorer

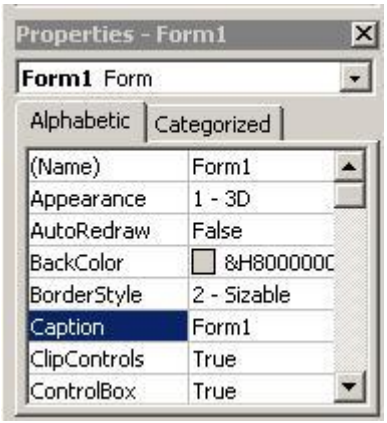
Project Explorer merupakan tempat yang digunakan untuk melihat daftar *forms*, *modules*, *class modules,* dan *designers*. Di samping itu, *project explorer* dapat digunakan untuk menambah *forms*, *modules*, *class modules* dan *designers*.



Gambar 2.11 *Project Explorer*

f. Properties Window

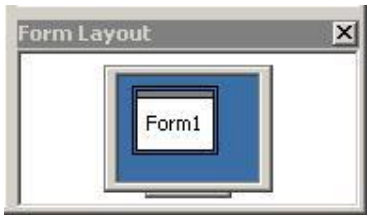
Properties Window merupakan tempat yang digunakan untuk mengatur properti dari setiap *object control*. Pada *properties* ini semua *object control* dapat diatur sesuai dengan program aplikasi yang akan dibuat.



Gambar 2.12 *Properties Window*

g. *Form Layout Window*

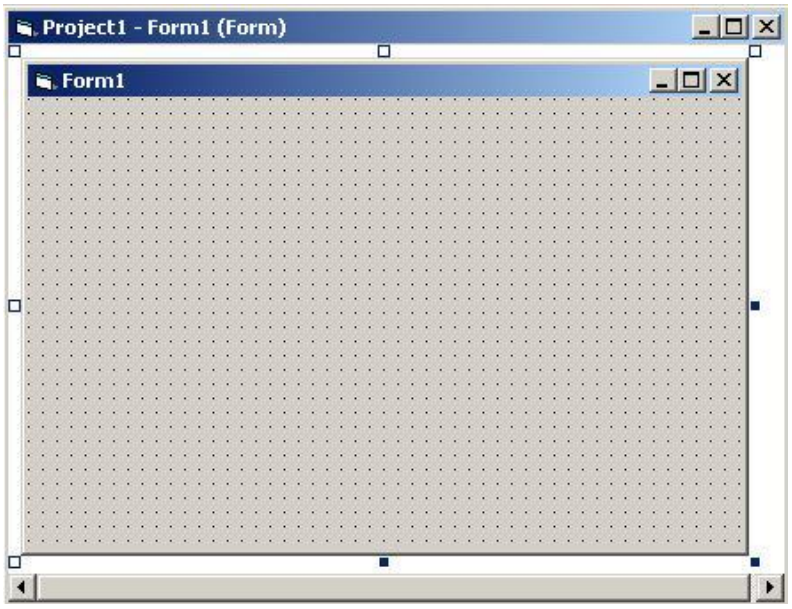
Form layout window ini berfungsi untuk melihat atau mengetahui posisi tampilan pada saat dieksekusi atau dijalankan. Untuk dapat mengubah ukuran *form*, dapat digunakan fasilitas *properties window*. Pengguna dapat mengatur posisi *form* sesuai dengan kebutuhan.



Gambar 2.13 *Form layout Window*

h. *Form* Objek

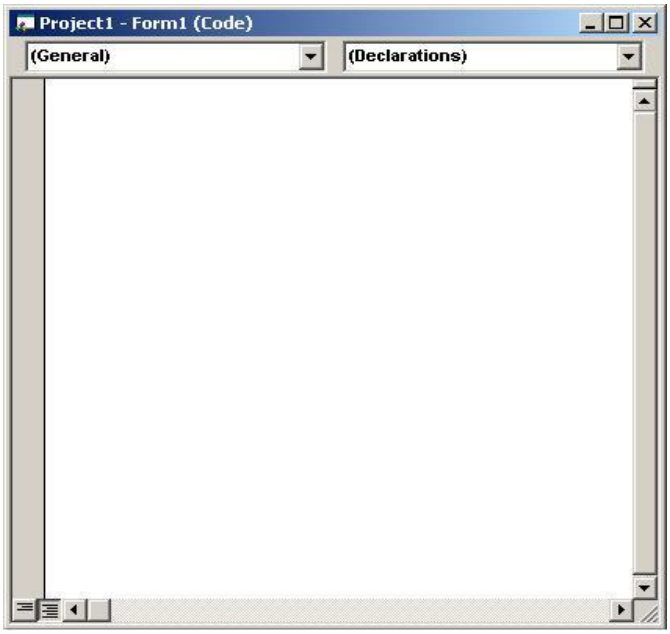
Form objek digunakan untuk menempatkan atau meletakan objek dari kontrol-kontrol yang akan digunakan untuk merancang dan membuat program aplikasi.



Gambar 2.14 Tampilan *form object*

i. *Form* Kode

Form code digunakan sebagai tempat untuk menulis kode-kode program aplikasi. Berikut tampilan *form code* pada *Visual Basic 6.0.*

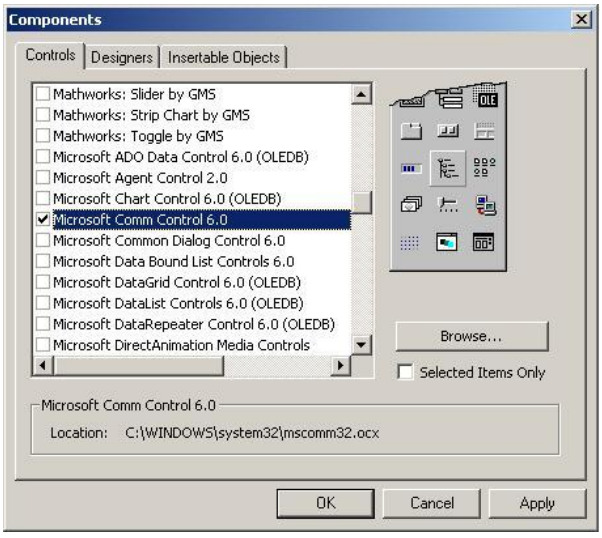


Gambar 2.15 Tampilan *Form* Kode

2.2.1.2 Komunikasi serial pada Visual Basic 6.0

Program aplikasi *Visual basic 6.0* telah menyediakan komponen untuk berkomunikasi dengan peralatan luar seperti mikrokontroler, modem, ataupun dengan perangkat lainya.

Secara default pada *toolbox* tidak disediakan, untuk menambahkan komponen tersebut dengan menambahkan komponen seperti telihat pada gambar di bawah ini:



Gambar 2.16 Penambahan komponen *Microsoft comm control 6.0* pada *visual Basic 6.0*

Fungsi dari *Microsoft comm control* ini adalah sebagai berikut:

1. Mengadakan komunikasi *serial port* pada komputer
2. Untuk bertukar data dengan peralatan luar
3. Memonitor dan merespon *event* dan *error* yang terjadi pada hubungan serial.

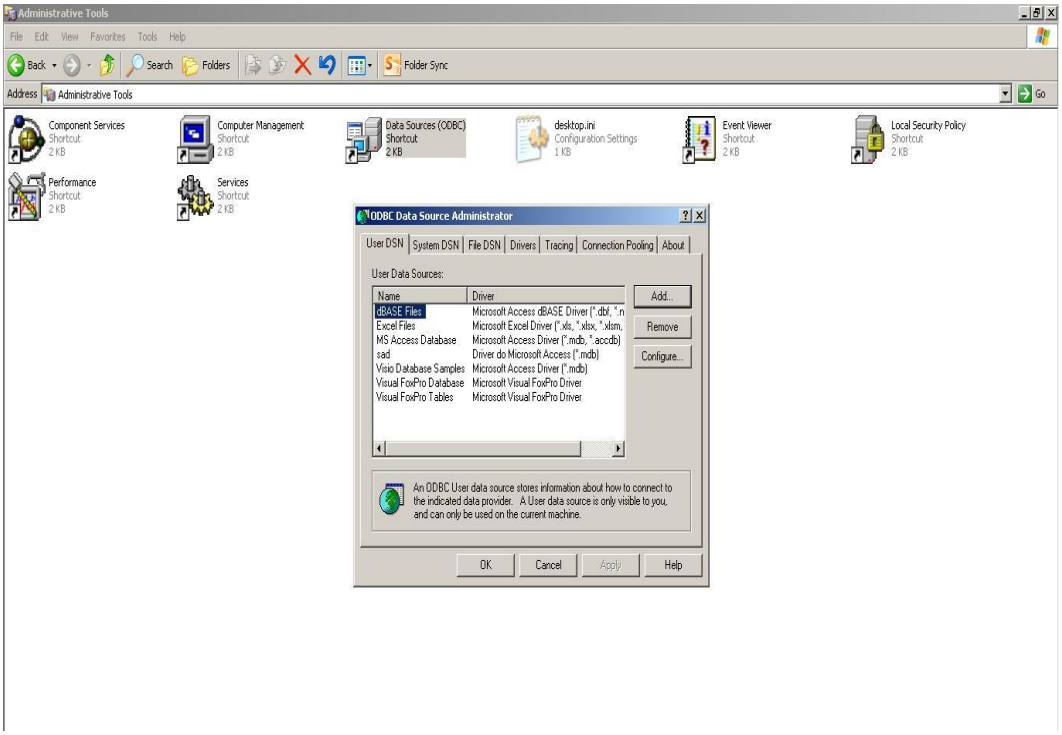
Karena pogram ini berkomunikasi dengan peralatan luar, untuk dapat menjalankanya ada beberapa langkah pengaturan, antara lain:

1. Membuka *serial port*.
2. Mengatur *receive* dan *transmit buffer properties*.
3. Managing *receive* dan *transmit buffer*.

2.2.1.3 Koneksi Visual Basic 6.0 dengan *database* Microsoft Acces menggunakan ODBC Driver

Untuk memudahkan koneksi antara *Visual Basic 6.0* dengan *Microsoft acces* maka dapat membuat koneksi dengan *administrative Tools* yang terdapat pada *control panel* yang telah disediakan oleh sistem operasi *Windows* sendiri.

Berikut gambar *control panel* untuk pembuatan koneksi antara *Microsoft acces* dengan *visual basic 6.0*:



Gambar 2.17 tampilan pada saat membuat koneksi untuk *Microsoft Access* dengan *Visual Basic 6.0*

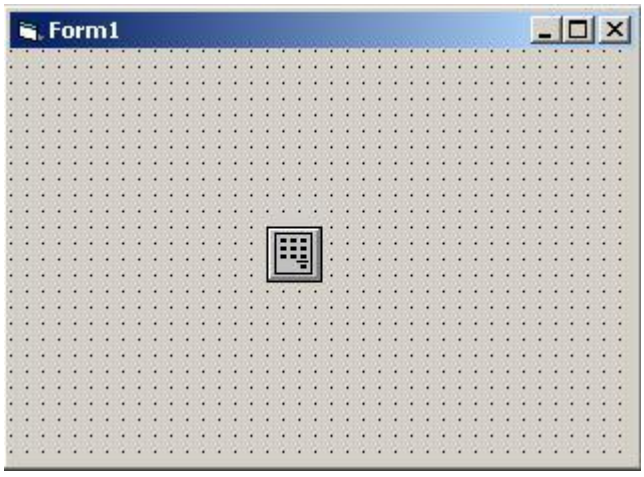
Sebelum membuat koneksi untuk *Visual basic 6.0*, pastikan data atau *file* telah dibuat sebelumnya karena dalam pembuatan koneksi akan diminta *file* mana yang akan dikoneksikan. Karena program yang dibuat akan dikoneksikan dengan *Microsoft acces* maka dipilih *driver do Microsoft acces*(*.mdb). Setelah itu maka berikanlah nama untuk memudahkan dalam pencarian pada saat menghubungkan pada *Visual Basic 6.0.*

2.2.1.4 Koneksi Visual basic 6.0 dengan Crystal Report

Banyak fasilitas koneksi yang telah disediakan oleh *Visual Basic 6.0* untuk menunjang program yang akan dibuat dengan aplikasi ini. Diantaranya adalah menghubungkan dengan crystal report yang berfungsi untuk mencetak laporan.

Namun sebelum menghubungkan perlu disiapkan terlebih dahulu program aplikasi *crystal report*.

Apabila program aplikasi telah di-*install* maka selanjutnya menambahkan komponen tambahan ke *toolbox* dan kemudian digunakan di *form object*. Berikut merupakan komponen *crystal report* yang telah ditambahkan ke *form* object:

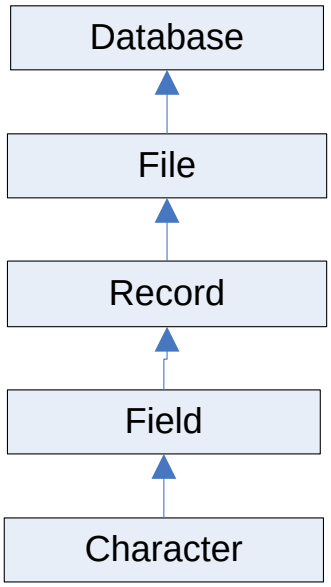


Gambar 2.18 Tampilan component *Crystal Report* pada *form object*

2.2.2 *Database*

Basis data atau yang sering disebut *database* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting dalam sistem informasi karena merupakan basis dalam menyediakan informasi disebut dengan *database* sistem.

Sistem basis data adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi pendekatan database. Untuk menghasilkan sebuah *database*, suatu data mempunyai jenjang atau tahapan prosesnya, dimulai dari karakter-karakter (*characters*), item data (*field*), *record* dan kemudian *database*. Jenjang ini dapat digambarkan sebagai berikut:

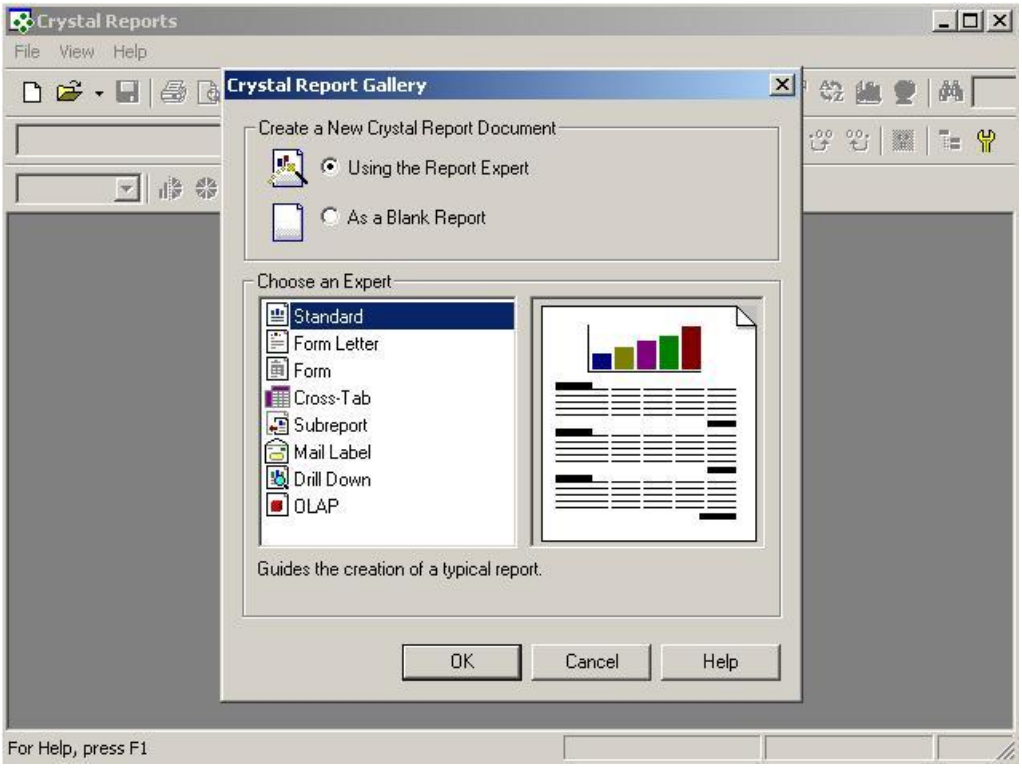


Gambar 2.19 Jenjang proses pembuatan *database*

1. *Field* yaitu menggambarkan suatu atribut dari *record* yang menunjukkan suatu item data, seperti nama, nip, bagian dan lain sebagainya. Kumpulan *field* ini membentuk suatu *record.*
2. *Record* yaitu dibentuk dari kumpulan *field*. *Record* menggambarkan suatu individu yang tertentu. Kumpulan *record* membentuk satu *file*.
3. *File* yaitu kumpulan dari *record.*
4. *Database* yaitu kumpulan dari *file* yang membentuk suatu *database*.

2.2.3 Crystal Report

Crystal report merupakan program khusus untuk membuat laporan yang terpisah dari program *microsoft Visual Basic 6.0*, tetapi keduanya dapat dihubungkan. Berikut tampilan utama ketika program dibuka dan akan membuat rancangan laporan:



Gambar 2.20 Tampilan *Crystal Report* pada saat memulai perancangan

Crystal report ini dapat membuat laporan-laporan yang terhubung dengan *database* seperti *Microsoft access*, *My Sql*, ataupun dengan *Sql Server*. Pemilihan *database* yang dihubungkan biasanya dipilih pada saat akan membuat laporan, setelah itu rancangan laporan yang dibuat akan dihubungkan dengan *Visual Basic 6.0* dengan komponen tambahan yang terdapat pada *Visual Basic 6.0*.