

BAB II

MULTIMEDIA DAN WINDOWS API

2.1. Pengertian Multimedia

Multimedia merupakan suatu konsep dan teknologi baru dalam bidang teknologi informasi dimana bentuk teks, gambar, suara, animasi dan video disatukan dalam komputer untuk disimpan, diproses dan disajikan baik secara linear maupun interaktif.

Ada beberapa aspek dalam multimedia yang perlu diketahui yaitu meliputi:

- a. *Still picture* adalah gambar yang diam, hal-hal ini dapat dikelola dengan adanya berbagai perangkat lunak seperti Fractal Design, Photoshop, Paint Shop Pro, dan lain-lain. Jenis-jenis gambar diam yang banyak digunakan antara lain BMP, TIF, JPEG, GIF, dan lain sebagainya. Gambar diam dapat diperoleh dengan cara melukis secara langsung menggunakan paket pengolahan citra atau piranti input seperti scanner, kamera digital.
- b. *Animated picture* adalah gambar bergerak yang merupakan kumpulan dari berbagai gambar diam yang ditampilkan secara bergantian sehingga membentuk kesan gambar yang bergerak. Istilah lain untuk gambar bergerak yang dikenal dalam dunia multimedia adalah *Video*. Berbagai jenis *file* yang digunakan antara lain: AVI, FLI, dan sebagainya. Pengolahan gambar video dapat dikelola dengan Adobe Premiere, Photoshop, dan lain-lain.
- c. *Sound* adalah ada dua jenis pembagian pada aspek ini yaitu suara yang mendukung gambar bergerak dan yang tidak atau individu. Pada jenis yang terakhir inilah yang sedang terjadi perubahan besar terutama pada format *file* dan teknik kompresinya. Ada beberapa format suara jenis ini yaitu: WAV, MP3, SND, MIDI, RMI, AIFF ,dan sebagainya

Adapun kebutuhan perangkat keras yang dibutuhkan untuk membangun aplikasi multimedia yaitu komputer Macintosh dari Apple dan IBM PC. Bagi komputer PC yang digunakan untuk membuat aplikasi multimedia digolongkan dengan *Multimedia Personal Computer* (MPC). MPC tidak hanya terdiri dari unit komputer saja, tetapi mempunyai standar spesifikasi terendah yaitu MPC level 1 atau MPC level 2. Standar ini ditetapkan oleh Multimedia PC Marketing Council yang berkedudukan di Washington, USA.

MPC Level 1 mempunyai standar spesifikasi yaitu:

- a. Mikroprosesor 386SX,
- b. RAM minimal 2 MB,
- c. HardDisk 30 MB,
- d. CD-ROM Drive,
- e. VGA Video (16 warna)
- f. Sound Card 8 bit, speaker atau headphones,
- g. Keyboard dan mouse

Spesifikasi terendah level 1 ini memang tidak dapat secara penuh mendukung pembuatan aplikasi multimedia. MPC level 2 dengan standar spesifikasi terendah dikeluarkan tahun 1993. Walaupun spesifikasi terendah ini berasal dari MPC level 2, tetapi tidak disarankan untuk menjadi sistem konfigurasi secara penuh. Spesifikasi minimum dari MPC level 2 adalah:

- a. Mikroprocessor 486 SX
- b. RAM 4 MB
- c. HardDisk 160 MB
- d. CD-ROM Drive
- e. SVGA (256 warna)
- f. Sound Card 16 bit ,speaker atau headphones,
- g. Keyboard dan mouse

Seiring dengan perkembangan teknologi komputer saat ini standar MPC tersebut mungkin sudah berubah, sehingga dapat disimpulkan bahwa semakin baik perangkat multimedia, maka semakin baik pula aplikasi yang dibuat.

Sedangkan perangkat lunak yang digunakan untuk menangani aspek multimedia ada beberapa macam salah satunya adalah MS Media Player yang dapat menjalankan *file* multimedia dalam bentuk *audio* atau *video*. Kemampuan yang dimiliki perangkat lunak lain sejenis ini sekarang semakin berkembang dan makin banyak fasilitas tambahan lainnya misalnya saja untuk merekam *file* suara dari perangkat keras, mengedit, mengolah serta menyimpannya dalam berbagai macam format kompresi file multimedia.

Lahirnya teknologi multimedia ini merupakan hasil kemajuan teknologi elektronik, komputer dan perangkat lunak. Kemampuan penyimpanan dan pengolahan gambar digital dalam belasan juta warna dengan resolusi tinggi serta reproduksi suara maupun *video* dalam bentuk digital merupakan terobosan yang mengagumkan dari teknologi ini.

2.1.1. File Audio

Dalam sistem operasi Windows *file audio* merupakan *file* yang berisi rekaman suara secara digital yang direkam melalui perangkat lunak Wave for Windows dan perangkat keras kartu suara (*Sound Card*). Adapun bentuk *file audio* ini pertama kali diperkenalkan oleh Microsoft dalam paket MS Windows versi 3.1 berupa file “.WAV” (*Waveform Sound File*). Teknologi *file audio* saat ini sudah cukup maju dimana ukuran, kompresi data dan kualitas suara menjadi dasar perkembangan teknologi perekaman suara secara digital.

Berikut ini bahasan dua teknologi *file audio* yang cukup terkenal, yaitu:

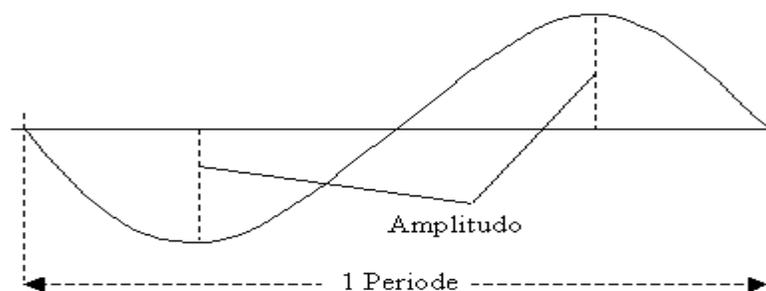
Waveform Sound File

Waveform Audio adalah teknik rekaman suara secara digital yang diperkenalkan oleh Multimedia Windows pada MS Windows versi 3.1. Proses perekaman suara secara digital dikenal dengan nama *sampling*.

Sampling adalah proses memecah suara menjadi bagian-bagian kecil informasi yang kemudian diubah menjadi deretan angka-angka. Karena jumlahnya begitu banyak maka diperlukan bantuan komputer untuk memprosesnya. Karena rumitnya *file waveform* membutuhkan tempat penyimpanan yang besar.

Sebuah *file waveform* dengan *digital waveform audio* dapat menyimpan dan memainkan kembali segala macam suara, tak terbatas pada musik.

Suara terjadi karena getaran diudara yang menstimulasi syaraf telinga manusia sehingga menghasilkan sensasi pendengaran. Telinga manusia mengubah energi kedalam jarak suara yang didengar. Suara umumnya dinyatakan sebagai bentuk gelombang analog. Bentuk gelombang suara yang sederhana dinyatakan dalam bentuk gelombang sinus, sebagai berikut:



Gambar 2.1 Gelombang sinus

Amplitudo (jarak antara titik puncak sinus dengan garis mendatar) menunjukkan volume suara. Titik puncak amplitudo menunjukkan keras

sekali dan titik terendah amplitudo menunjukkan pelan. Garis mendatar menunjukkan diam. Semua bentuk gelombang dibagi menjadi periode.

Periode adalah jarak antara dua puncak berurutan dalam bentuk gelombang. Frekuensi bentuk gelombang dinyatakan dengan jumlah periode yang terjadi dalam satu detik. Satu periode per detik setara dengan satu Hertz (Hz). Seratus periode setara dengan satu Kilo Hertz (KHz). Bentuk gelombang suara secara fisik dapat dilihat pada alur yang ada pada permukaan piringan. Dengan meletakkan ujung jarum stylus keatas alur piringan dan memutar piringan tersebut akan menghasilkan arus listrik kemudian diubah menjadi gelombang suara yang setelah melalui penguat suara disambung ke speaker dan menghasilkan suara.

Ada tiga karakteristik yang menentukan kualitas dan besarnya *file digital waveform* , yaitu:

Frekuensi - makin tinggi frekuensi sampel suara, makin besar kapasitas disk yang dibutuhkan. Ada tiga *sampling* frekuensi standar yaitu: 11.025 khz, 22.025 khz, dan 44.1 khz.

Banyaknya Informasi - yang disimpan pada tiap sampel menunjukkan tingkat presisi dari sampel yang diukur. Kita dapat membuat sampel dalam resolusi 8 bit atau 16 bit. Informasi tiap sampel diperoleh dengan membagi tiap *waveform* menjadi beberapa bagian yang sama atau dikenal dengan istilah *equal unit*. Untuk sampel 16 bit, tiap sampel dibagi menjadi 65.536 equal unit. Makin tinggi resolusinya makin presisi sampel yang dihasilkan. Dengan demikian suara yang dihasilkan makin mendekati aslinya.

Channel/Saluran – suara menunjukkan proses perekaman satu channel Waveform yang dikenal dengan istilah mono atau dua channel yang dikenal istilah stereo. Suara stereo hasilnya jauh lebih baik

dibandingkan dengan mono namun membutuhkan tempat penyimpanan dua kali lebih banyak dibandingkan dengan mono.

Masalah yang sering terjadi pada saat proses merekam *file digital Waveform* sedang berlangsung tiba-tiba tempat untuk menyimpan dalam disk tidak mencukupi. Ini terjadi karena *Waveform* membutuhkan begitu banyak tempat untuk menyimpan sampel data digital. Berikut ini adalah daftar yang dapat dijadikan patokan untuk mengetahui banyaknya tempat yang dibutuhkan untuk merekam *Waveform* dengan berbagai resolusi dan sampling rate selama satu menit.

| SAMPLING (kHz) | Bit/Sampel | Channel | Byte/Menit |
|----------------|------------|---------|------------|
| 11.025 | 8 bits | 1 | 66 k |
| 11.025 | 8 bits | 2 | 1.32 MB |
| 11.025 | 16 bit | 1 | 1.32 MB |
| 11.025 | 16 bit | 2 | 2.62 MB |
| 22.05 | 8 bits | 1 | 1.32 MB |
| 22.05 | 8 bits | 2 | 2.62 MB |
| 22.05 | 16 bit | 1 | 2.62 MB |
| 22.05 | 16 bit | 2 | 5.29 MB |
| 44.1 | 8 bits | 1 | 2.62 MB |
| 44.1 | 8 bits | 2 | 5.29 MB |
| 44.1 | 16 bit | 1 | 5.29 MB |
| 44.1 | 16 bit | 2 | 10.58 MB |

Tabel 2.2 Kebutuhan Ruang Penyimpanan dalam 1 menit

MP3

Merupakan kependekan dari **MPEG Layer 3**, yaitu suatu format penyimpanan *file audio* secara digital dengan kompresi data yang akhirnya dapat mengurangi ukuran *file audio* tersebut dan sedikit reduksi kualitas. Format *file* ini dikeluarkan suatu organisasi yang menangani standarisasi pembuatan *digital video* dan kompresi *audio*, yaitu *Moving Pictures Expert Group* (MPEG). Organisasi ini bekerja dibawah naungan *International Standards Organization* (ISO).

File MP3 merupakan *file* yang berisikan data *audio* yang dikompresi. Setiap blok data *audio* yang dikompresi memiliki *header* untuk menyimpan informasi yang digunakan untuk proses dekompresi

pada saat file ini dimainkan. Data *audio file* MP3 akan diakhiri informasi tambahan dalam beberapa bit seperti “*copyright*” bit dan “*private*” bit. Beberapa bit terakhir inilah yang dimanfaatkan oleh **Eric Kemp** untuk diisi dengan data tambahan yang lain berupa teks yang menginformasikan isi *file audio*, seperti judul lagu, artis, dan sebagainya. Peletakkan Informasi ini dikenal dengan istilah *Tag*. Untuk memudahkan pengenalan *Tag* ini maka ukuran *Tag* yang dipilih adalah 128 Byte.

Berikut ini adalah ukuran data ID3 Tag versi 1.0 dan layoutnya:

| | |
|-------------------------|---------------|
| Song Title (Judul Lagu) | 30 Characters |
| Artist | 30 Characters |
| Album | 30 Characters |
| Year (Tahun) | 4 Characters |
| Genre (Jenis Musik) | 1 byte |
| Comment (Komentar) | 30 Characters |

Tabel 2.3 Data ID3 Tag



Gambar 2.1 Layout ID3 Tag

Pada tabel diatas jika dijumlahkan ukuran byte diatas adalah 125 byte dan bukan 128 byte. Tiga byte terakhir ini sebenarnya digunakan oleh kalimat “TAG” yang menandakan bahwa *file* MP3 memiliki ID3Tag. Kalimat ini dapat dilihat pada setiap akhir *file* MP3 yang memiliki ID3 Tag

Versi ID3 Tag saat ini sudah mencapai versi 2.0 dan tentu saja ada tambahan informasi seperti *Track* pada ID3 Tag versi 1.1. Informasi yang terdapat pada *file audio* MP3 ini sangat membantu bagi para pendengar music secara digital untuk memanajemen *file* musik yang ada di *Disk*.

2.1.2. File Video

Merupakan elemen yang berisi rekaman dari kaset video atau yang lainnya yang biasanya bersifat analog, kemudian dimasukkan dalam lingkungan komputer dan berubah menjadi gambar-gambar digital yang tidak jauh berbeda dengan animasi yang bersuara.

Terdapat beberapa jenis format *file video* saat ini diantaranya : AVI, IVF, WMV, MPEG , dan sebagainya.

File Audio Video Interleaved (AVI) merupakan sebuah *file* yang berisi *waveform audio* dan *digital video*. Teknik perekaman video ini mirip dengan gambar bioskop, yaitu dengan cara mengelabui pandangan mata manusia. Bila mata melihat sederetan gambar (disebut *frame*) yang digerakkan dengan cepat perubahan gambar akan nampak seperti berjalan. Mata mulai nampak dikelabui ketika gambar bergerak 16 *frame* per detik. Gambar bioskop misalnya ditampilkan setiap 24 *frame* per detik, sedang televisi menggunakan 30 *frame* per detik. Gambar yang direkam dalam *file* berakhiran .AVI memiliki standar resolusi 160 * 200 pixel dengan 15 *frame* per detik sedang *audio track* diatur pada 11 k *sampling rate* dengan 8 bit per sampel.

Karena perkembangan teknologi komputer yang pesat saat ini, maka tidak dapat dipungkiri lagi bahwa proses perekaman, pengeditan, penyimpanan *file video*, dan sebagainya dapat dipastikan akan melibatkan komputer.

2.2. Sistem operasi Windows

Microsoft Windows merupakan suatu perangkat lunak yang populer untuk mengoperasikan komputer *Personal Computer* (PC) yang dengan segala kemampuannya dapat memikat para user untuk menggunakannya terutama tampilan grafiknya yang mudah digunakan (*user friendly*) dan kemudahan instalasinya. Ada beberapa release untuk sistem operasi ini baik untuk *Personal* maupun *Network* yaitu MS Windows 3.0, 3.1, 3.11, 95, 98, NT, 2000 dan yang terakhir masih dalam tahap beta test yaitu MS Windows Millenium Edition. Adapun kemampuan dari sistem operasi yang menjadikan *software* ini banyak digunakan antara lain:

- a. Kerja sama antar aplikasi yang dinamik dan unik. Ciri ini dengan drastis dapat meningkatkan produktivitas pemakai.
- b. Sistem operasi yang *multitasking*, sehingga beberapa program dapat dijalankan pada satu saat.
- c. Dukungan aplikasi yang beragam.
- d. Mengakses memori lebih besar, sehingga pemakai tidak lagi dibatasi jumlah memori.
- e. *Driver* standar untuk setiap aplikasi windows.
- f. Investasi yang menguntungkan.
- g. Windows menerapkan operasi pemakaian baku untuk setiap aplikasi.

2.2.1. Pemrograman Windows

Perusahaan Microsoft sebagai pembuat sistem operasi Windows telah membuat perlengkapan (*tool*) pemrograman windows berupa referensi manual yakni *Microsoft Windows Software Development Kit* (SDK) dan *Device Driver Kit* (DDK).

Pada SDK dan DDK tersebut tercantum hal-hal dasar yang harus diikuti pada pembuatan program berbasis windows misalnya API, sumber-

sumber, teknik-teknik pemrograman, serta contoh program. Sayangnya bahasa pemrograman yang tersedia adalah terutama untuk bahasa C dan bahasa Assembly. Hal ini akan menimbulkan kesulitan bagi orang awam yang baru ingin belajar bahasa pemrograman langsung menuju ke pembuatan program aplikasi berbasis Windows. SDK dan DDK windows tersebut ditujukan hanya bagi orang yang telah mahir kedua bahasa pemrograman tersebut.

Tetapi, tidak lama kemudian muncul bahasa pemrograman *visual*, dimana pembuatan aplikasi berbasis windows akan menjadi mudah. Adapun hal-hal yang memudahkan pemrograman *visual* antara lain:

a. Pemrograman yang digerakkan oleh pesan (*message driven*)

Semua kegiatan dari program windows dilakukan melalui pesan. Pesan merupakan informasi ketika terjadi perubahan pada window. Setiap kali window dipindahkan, terjadi penekanan tombol, dan kejadian-kejadian lainnya, windows akan selalu memberikan respon terhadap pesan tersebut, dan akan melakukan tindakan-tindakan yang sesuai.

Untuk mempermudah pemrograman, maka kode-kode pesan telah dibakukan oleh microsoft dengan tanda “WM_” pada awal simbol nama setiap pesan. Sebagai contoh “WM_RBUTTONDOWN” jika tombol kanan mouse ditekan lalu “WM_RBUTTONUP” jika tombol kanan mouse dilepas.

Dalam pelaksanaannya walaupun pesan-pesan yang dikirimkan oleh windows banyak tetapi tidak semua pesan tersebut diproses oleh aplikasi, hanya pesan-pesan yang tertentu saja yang diproses yang berhubungan dengan tujuan program yang dirancang.

b. Pemakaian sumber secara bersama (*resource sharing*)

Setiap aplikasi windows menggunakan sumber-sumber secara bersama. Sumber-sumber maksudnya disini adalah memori, layar,

harddisk, keyboard, dan lainnya. Supaya sumber-sumber disini dapat digunakan bersama, maka setiap aplikasi windows harus menggunakan fungsi-fungsi API. Aplikasi harus meminta terlebih dahulu (dari Windows) sumber yang akan diaksesnya, dan kemudian jika tidak digunakan dikembalikan lagi, supaya dapat dimanfaatkan oleh aplikasi lainnya.

c. Manajemen Memori (*memory management*)

Pada dasarnya manajemen memori aplikasi windows merupakan pengembangan manajemen memori yang diterapkan pada program-program DOS. Pengembangan tersebut terletak pada masalah penggunaan memori secara bersama misalnya karena beberapa aplikasi dapat dibuka beberapa kali, maka tidak mungkin salinan aplikasi tersebut ditempatkan dibagian memori setelah bagian memori ditempati oleh aplikasi yang sebelumnya dibuka. Hal ini tentu saja akan menyebabkan memori akan cepat habis. Windows mengatasinya dengan menerapkan pengelolaan memori yang baru yaitu dengan memindahkan memori ke disk atau ke bagian (alamat) memori yang lain. Bahkan jika dimungkinkan, memori yang telah ditempati data atau kode dapat dikosongkan untuk mengatasi kekurangan memori.

d. Hasil keluaran (*output*) grafik

Sebelum munculnya windows, program-program aplikasi yang hebat dan laku dipasaran, seperti Lotus 123, Word Perfect, dBase, menggunakan modus teks sebagai media penghubung dengan pemakai, akan tetapi dengan semakin diperlukannya suatu program yang lebih mudah digunakan, dan lebih menyenangkan untuk digunakan, maka banyak pengembang perangkat lunak yang beralih ke modus grafik.

Untuk mendapatkan hasil tampilan grafik, Windows menggunakan pustaka yang disebut *Graphics Device Interface* (GDI). Tombol-tombol (*buttons*), icon, bitmap, scrollbar, atau semua tampilan yang ada pada windows memanfaatkan GDI.

GDI merupakan perangkat yang berdiri sendiri, dengan pengertian proses penggambaran ke layar juga berlaku untuk penggambaran ke printer. Dalam pemrograman, GDI cukup diberitahu ke peralatan apa penggambaran dilakukan, proses selanjutnya diserahkan ke GDI. Untuk membantu GDI melakukan tugas sesuai dengan peralatan yang digunakan, maka GDI bekerja sama dengan perangkat pengendali (*device driver*) untuk masing-masing peralatan. Sebagai contoh terdapat file VGA.DRV yang merupakan perangkat pengendali untuk monitor vga dan EPSON.DRV untuk printer merek epon.

Selain layar dan printer, GDI mengenal dua buah perangkat lagi yaitu **BITMAP** dan **METAFILE**. Kedua perangkat yang disebutkan terakhir merupakan perangkat samaran (*pseudo device*), dimana tujuannya untuk penyimpanan pada memori atau disk.

e. Objek hubungan dengan pemakai (*User-Interface Object*)

Windows kaya dengan objek-objek dalam melakukan hubungan dengan pemakai. Tujuan dari objek-objek ini adalah untuk mempermudah pemakaian program aplikasi Windows. Menu merupakan salah satu objek yang paling sering digunakan. Selain menu objek Windows lainnya seperti window, Icon, Scrollbar, kotak dialog.

2.2.2. Perangkat lunak yang digunakan untuk pembuatan aplikasi

Microsoft Visual Basic adalah bahasa pemrograman yang digunakan untuk membuat aplikasi Windows berbasis grafis (*GUI-Graphical User*

Interface). Visual Basic merupakan pemrograman terkendali kejadian (*event-driven programming*) artinya program menunggu sampai adanya respon dari pemakai berupa kejadian tertentu (tombol diklik, menu dipilih, dan lain-lain). Ketika *event* terdeteksi, kode yang berhubungan dengan *event* (*Procedure Event*) akan dijalankan.

Sejarah singkat Visual Basic

Berikut ini adalah point-point penting dalam sejarah perkembangan Visual Basic, sebagai berikut:

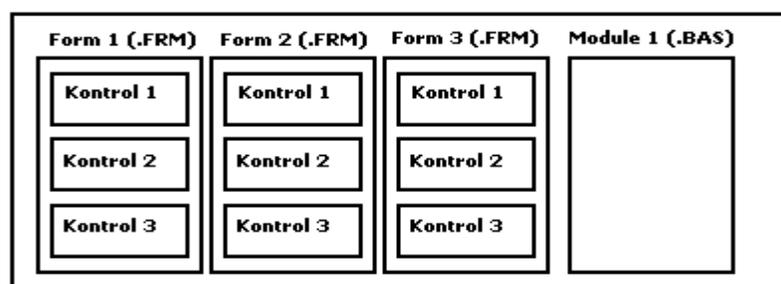
- a. Visual Basic pertama kali diperkenalkan tahun 1991 yaitu program Visual Basic untuk DOS dan untuk Windows.
- b. Visual Basic 3.0 dirilis tahun 1993.
- c. Visual Basic 4.0 dirilis pada akhir 1995.
- d. Visual Basic terbaru adalah versi 6.0 yang dirilis pada akhir tahun 1998.

Microsoft umumnya membuat tiga edisi Visual Basic yaitu:

- a. Standard Edition merupakan produk dasar.
- b. Professional Edition berisi tambahan Microsoft Jet Data Access Engine (database) dan pembuatan server OLE automation.
- c. Enterprise Edition adalah edisi client-server.

Struktur aplikasi MS Visual Basic

Proyek (.VBP, .MAK)



Tabel 2.4 Struktur Aplikasi Visual Basic

Aplikasi (proyek) terdiri dari:

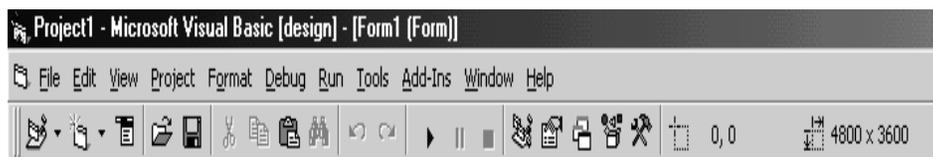
- a. **Form**-Windows atau jendela untuk membuat tampilan.
- b. **Kontrol** (*Control*)-Tampilan berbasis grafis yang dimasukkan pada form untuk membuat interaksi dengan pemakai. Contoh: TextBox, Label, Scroll bar , dan lain-lain.
- c. **Properti** (*Properties*)-Nilai yang dimiliki oleh sebuah objek Visual Basic. Contoh: Name, Caption, Size, dan lain-lain.
- d. **Metode** (*Methods*)-Serangkaian perintah yang sudah tersedia pada suatu objek yang dapat diminta untuk mengerjakan tugas khusus.
- e. **Prosedur kejadian** (*Event Procedures*)-Kode yang berhubungan dengan suatu objek. Kode ini akan dieksekusi ketika ada respon dari pemakai berupa event tertentu.
- f. **Prosedur Umum** (*General Procedures*)-Kode yang tak berhubungan dengan suatu objek. Kode ini harus diminta oleh aplikasi.
- g. **Modul** (*Module*)-Kumpulan dari prosedur umum, deklarasi variable dan definisi konstanta yang digunakan oleh aplikasi.

Langkah-langkah untuk mengembangkan aplikasi:

1. Membuat tampilan.
2. Mengatur nilai properti.
3. Menulis kode program.

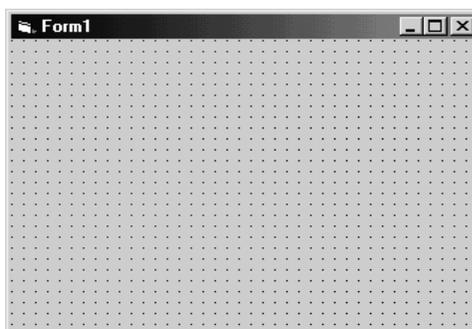
Tampilan Layar Visual Basic

- a. **Jendela Utama** (*Main Windows*) terdiri dari title bar (baris judul), Menu bar, dan Toolbar. Baris judul berisi nama proyek, mode operasi Visual Basic dan form yang aktif. Menubar & Toolbar merupakan pengontrol operasi dalam lingkungan Visual Basic.



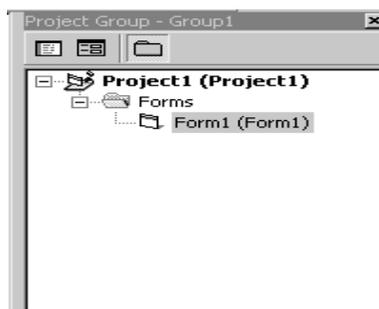
Gambar 2.2 Jendela utama Visual Basic

- b. **Jendela Form** (*Form Window*) pusat dari pengembangan aplikasi.



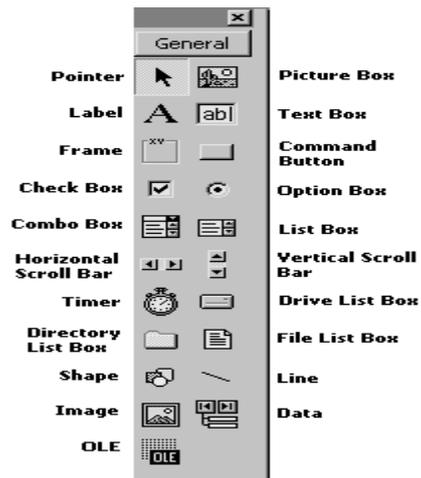
Gambar 2.3 Jendela Form

- c. **Jendela Proyek** (*Project Window*) menampilkan daftar form dan modul proyek. Proyek merupakan kumpulan dari modul form, modul class, modul standar, dan file sumber yang membentuk suatu aplikasi.



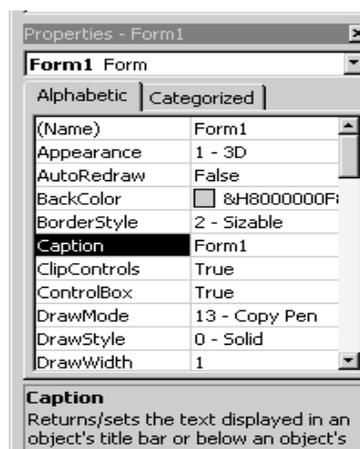
Gambar 2.4 Jendela Proyek

- d. **Toolbox** adalah kumpulan dari objek yang digunakan untuk membuat tampilan serta kontrol bagi program aplikasi.



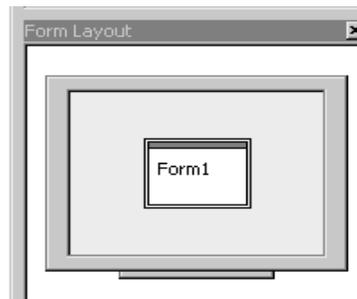
Gambar 2.5 Toolbox

- e. **Jendela Properti** (*Properties Window*) berisi daftar struktur pengaturan properti yang digunakan pada sebuah objek terpilih. Ada dua tab tampilan: *Alphabetic* (urut abjad) dan *Categorized* (urut berdasar kelompok). Di bagian bawah kotak terdapat properti dari objek terpilih.



Gambar 2.6 Jendela Properti

- f. **Jendela layout form** (*Form Layout Windows*) menampilkan posisi relatif terhadap layar monitor.



Gambar 2.7 Jendela layout form

Karena kemudahan, fasilitas serta unsur kedekatannya dengan sistem operasi windows, maka bahasa pemrograman inilah yang nantinya akan digunakan untuk membuat Player Multimedia.

2.3. Windows API

Seperti dijelaskan diatas bahwa MS Windows memberikan kesempatan bagi pengguna atau programmer untuk membuat aplikasi sendiri. Sebagai sistem operasi komputer Windows menyediakan sub rutin penghubung atau lebih dikenal dengan Windows API, dimana dengan rutin ini dapat dikembangkan suatu aplikasi atau utiliti yang berjalan di lingkungan Windows. Windows API merupakan pustaka yang berisi kumpulan fungsi yang dibuat dengan bahasa C. Pada Windows API terdapat lebih dari 500 fungsi yang diletakkan di beberapa file, antara lain:

- a. GDI.EXE atau GDI32.DLL, kumpulan fungsi GDI.
- b. USER.EXE atau USER32.DLL, kumpulan fungsi untuk antarmuka pemakai seperti penanganan mouse, cursor, icon, dan sebagainya.
- c. KERNEL (KRNL286.EXE, KRNL386.EXE, KERNEL32.DLL)
- d. Kumpulan fungsi untuk operasi level rendah (*Low Level Operation*), seperti manajemen memori, penanganan *multitasking*, dan sebagainya.
- e. LZEXPAND.EXE, kumpulan fungsi untuk menangani kompresi *file* dengan metode Lempel-Ziv.

- f. MMSYSTEM.DLL, WINMM.DLL merupakan kumpulan fungsi untuk multimedia seperti memainkan file WAV, MID, AVI, dan sebagainya.
- g. dan lain-lain.

File-file API yang berakhiran “.EXE” merupakan *file* khusus yang menjadi dasar bagi sistem operasi windows seperti GDI.EXE & USER.EXE, sedangkan yang berakhiran “.DLL” merupakan file API pendukung seperti MMSYSTEM.DLL & WINMM.DLL. Semua *file* API ini diletakkan didalam direktori sistem windows. DLL terakhir yang disebutkan diatas (WINMM.DLL) merupakan pustaka yang sangat berperan dalam pembuatan aplikasi player multimedia nantinya.

2.3.1. Komponen-komponen dalam Windows API

Dalam Windows API terkadang fungsi-fungsi API memerlukan suatu bagian atau objek yang dianggap satu kesatuan dari fungsi tersebut. Objek-objek ini termasuk fungsi API yang merupakan komponen pembentuk Windows API. Berikut ini adalah komponen-komponen tersebut, yaitu:

- a. **Fungsi** (*Function*) – merupakan komponen inti pembentuk Windows API. Fungsi ini tersimpan dalam file DLL dan dapat dengan mudah diakses kapanpun oleh sebarang aplikasi Windows.
- b. **Struktur** (*Structure*) – merupakan kombinasi kumpulan dari beberapa variable individu, yang sering digunakan oleh Windows API untuk menyimpan kumpulan informasi yang saling berkaitan. Kebanyakan fungsi API membutuhkan struktur untuk menghemat ruang parameter. Untuk menggunakan struktur ini harus dideklarasikan dahulu.
- c. **Konstanta** (*Constant*) – penamaan bagi nomor isoteric yang berguna bagi Windows API. Masing-masing konstanta mempunyai fungsi sendiri-sendiri.

- d. **Fungsi CallBack** (*CallBack Function*) – merupakan definisi aplikasi yang memanfaatkan fungsi API.
- e. **Pesan** (*Messages*) – merupakan tipe spesial dari konstanta.

2.3.2. Memanggil fungsi API di Visual Basic

Untuk menggunakan fungsi API, deklarasikan fungsi tersebut di kode sumber kemudian memanggilmnya sama seperti pada fungsi yang lain di visual basic.

Deklarasi fungsi API

Karena fungsi API bukan bagian internal Visual Basic, maka sebelum menggunakannya harus mendeklarasikan secara eksplisit. Pendeklarasian ditempatkan pada bagian (*declarations*) dalam form atau modul. Jika dideklarasikan dalam form maka *keyword* (kata kunci) “*private*” digunakan sebagai awalan didepan nama fungsi, agar fungsi hanya bisa diakses dari dalam form, sedangkan dalam modul deklarasi dapat dilakukan secara “*public*” dan “*private*” karena deklarasi fungsi API pada modul dapat digunakan baik dari dalam modul maupun form. Berikut aturan penulisan deklarasi fungsi:

```
[{Public / Private}] Declare Function nama_fungsi Lib
"nama_DLL" [Alias "nama_alias_fungsi"] (daftar_argumen)
As tipe_data
```

Keterangan:

nama_fungsi - API merupakan referensi yang dapat dipanggil dari dalam kode sumber.

nama_DLL - merupakan pustaka dari fungsi tanpa mencantumkan lokasi file, karena letak semua file API diasumsikan oleh Visual Basic pada direktori sistem direktori Windows.

nama_alias_fungsi - (opsional) merupakan nama alias untuk membedakan dua fungsi yang berbeda yaitu versi *ANSI* (Inggris) dan *Uni-Code Wide Character* (China). Versi *ANSI* mempunyai nama alias fungsi dengan berakhiran huruf A, sedangkan versi *Uni-Code Wide Character* berakhiran dengan huruf W.

daftar_argumen - merupakan parameter yang digunakan fungsi.

tipe_data - tipe data nilai balik dari fungsi.

Daftar argumen

Merupakan daftar parameter berupa variabel yang isinya akan digunakan oleh fungsi untuk diproses. Berikut ini ilustrasi sintaks argumen:

```
[ {ByVal | ByRef} ] nama_argumen As tipe_data, ...
```

Keterangan:

nama_argumen – merupakan parameter yang dipakai fungsi, Biasanya bentuk nama argumen mewakili dari kegunaannya dan tipe data.

tipe_data – tipe data dari isi argumen. Tipe data dapat berupa kata kunci “*Any*”, yang dapat menampung semua jenis tipe data atau dapat berupa tipe struktur yang merupakan kumpulan variable dan tipe data.

Tipe data

Telah diketahui bahwa tipe data mempengaruhi ukuran dan format variable argumen yang ditujukan pada fungsi. Fungsi API hanya mendukung tipe data yang terbatas dibandingkan fungsi lain yang tersedia pada Visual Basic. Berikut ini tipe data yang dapat digunakan fungsi API:

- a. *Byte* - tipe data yang terdiri dari 8 bit integer.
- b. *Integer* - tipe data yang terdiri dari 16 bit integer.
- c. *Long* - tipe data yang terdiri dari 32 bit integer.

- d. *String* - tipe data yang terdiri dari kumpulan karakter.

Tipe Struktur

Tipe struktur dalam pemrograman merupakan kumpulan/group variable yang saling berhubungan. Sebuah struktur terdiri dari satu atau lebih variabel yang masing-masing dapat memiliki tipe data yang berbeda dan atau sama (termasuk tipe struktur lainnya). Berikut ini adalah sintaks tipe struktur:

```
[(Public | Private)] Type nama_tipe
    variable1 As tipe_data1
    variable2 As tipe_data2
    ...
End Type
```

Keterangan:

nama_tipe – adalah nama tipe struktur.

variable1, variable2, ... - nama variable yang menjadi anggota suatu tipe.

tipe_data1, tipe_data2, ... - jenis tipe data yang digunakan variable.

Jenis tipe data yang harus digunakan dalam struktur adalah *byte*, *integer*, *long*, dan *string*.

Tipe struktur biasanya digunakan untuk menyimpan informasi yang saling berkaitan, sehingga dalam API terdapat bermacam-macam tipe struktur yang siap digunakan oleh fungsi-fungsi API.

Pengubah (Modifier) ByVal & ByRef

Kata kunci *ByVal* dan *ByRef* merupakan suatu metode pengubah default parameter yang berguna untuk melewati parameter pada fungsi API. Kata kunci tersebut keduanya sangat berbeda dan tidak dapat diganti kedudukannya dengan kata lain argumen/parameter yang memiliki prefix *ByVal* tidak dapat diganti

dengan *ByRef*. Jika terjadi kesalahan dalam penggunaan keyword ini, akan dapat menyebabkan aplikasi macet.

ByVal (*By Value*)-merupakan modifier untuk melewati nama_argumen/variable berdasarkan isi.

ByRef (*By Reference*)-merupakan modifier untuk melewati nama_argumen/variable berdasarkan referensi.

Keyword ini pada umumnya jarang sekali digunakan, karena dianggap kurang begitu penting dan membingungkan dalam pemanggilan fungsi API.

Cara mudah pendeklarasian fungsi API

Untuk menghindari kesalahan penulisan terutama masalah besar-kecilnya huruf, oleh Microsoft pada paket Visual Basic 5.0 disertakan utiliti API Viewer untuk menayangkan dan menyisipkan daftar panjang berisi konstanta API, yang berkaitan dengan tipe struktur dan deklarasi fungsi. Hal ini akan sangat membantu programmer karena pendeklarasian atau pemanggilan fungsi API dalam Visual Basic sangat memperhatikan besar-kecilnya huruf (*case-sensitive*).

2.3.3. Contoh-contoh pendeklarasian fungsi API di Visual Basic

Berikut ini beberapa contoh sederhana dari cara pendeklarasian fungsi

API :

- a. Deklarasi fungsi yang tidak memiliki parameter dengan nilai balik 32 bit integer.

```
Declare Function GetDesktopWindow Lib "user32.dll" () As Long
```

- b. Deklarasi fungsi yang memiliki parameter dengan prefix ByVal.

```
Declare Function Sleep Lib "kernel32.dll" (ByVal dwMilliseconds As Long)
```

- c. Deklarasi fungsi dengan nama fungsi alias versi ANSI.

```
Declare Function CharUpper Lib "user32.dll" Alias "CharUpperA" (ByVal lpsz As String) As String
```

- d. Deklarasi tipe struktur

```
Type RECT
    left As Long
    top As Long
    right As Long
    bottom As Long
End Type
```

- e. Deklarasi fungsi yang menggunakan tipe struktur RECT sebagai parameter.

```
Declare Function CreateRectRgnIndirect Lib "gdi32.dll" (lpRect As RECT) As Long
```

2.4. Media Control Interface (MCI)

MCI merupakan seperangkat perintah yang dapat bekerja secara mandiri yang digunakan dalam aplikasi Windows untuk mengendalikan perangkat multimedia seperti *Audio Compact Disc*, *Video disc*, *Waveform sound file*, dan *Midi sequencer*. MCI menggunakan metode bahasa manusia (dalam bahasa Inggris) untuk mengakses perangkat multimedia berupa perintah-perintah MCI (*MCI Commands Strings*).

2.4.1. Komunikasi dengan perangkat MCI

Untuk dapat menggunakan atau memerintahkan perangkat melalui MCI, maka dibutuhkan perintah-perintah MCI. Semua perintah ini akan ditangani oleh pengatur perangkat (*Device Handler*). Adapun untuk mengirimkan perintah MCI digunakan fungsi MCI API yaitu:

- a. **mciSendString**

```
Declare Function mciSendString Lib "winmm.dll" Alias "mciSendStringA" (ByVal lpszCommand As String, ByVal lpszReturnString As String, ByVal cchReturnLength As Long, ByVal hwndCallback As Long) As Long
```

Keterangan:

lpzCommand – parameter yang digunakan untuk menampung perintah MCI.

lpzReturnString – parameter yang digunakan untuk menampung keluaran/output dari perintah MCI. Parameter ini harus diinisialisasi dengan panjang karakter minimal 128 karakter untuk menampung keluaran berupa string.

CchReturnLength – panjang karakter yang digunakan.

hwndCallback – Handle Window yang memanggil fungsi ini.

Sedangkan untuk mendapatkan deskripsi jika terjadi kesalahan dalam perangkat MCI digunakan perintah yaitu:

b. **mciGetErrorString**

```
Declare Function mciGetErrorString Lib "winmm.dll" Alias
"mciGetErrorStringA" (ByVal fdwError As Long, ByVal
lpzErrorText As String, ByVal cchErrorText As Long) As
Long
```

Keterangan:

fdwError – merupakan kode kesalahan MCI yang digunakan fungsi *mciGetErrorString* untuk mendapatkan deskripsi kesalahan.

lpzErrorText – parameter yang menampung deskripsi kesalahan. Parameter ini harus dapat menampung minimal 128 karakter. Jika kode kesalahan tidak ditemukan maka parameter ini bernilai Null.

cchErrorText – panjang karakter yang digunakan *lpzErrorText*.

Nilai balik dari kedua fungsi ini jika berhasil akan mengembalikan nilai 0, jika tidak akan mengembalikan kode kesalahan MCI dan bukan nilai 0. Tambahan lagi yaitu kedua fungsi ini dapat digunakan pada hampir semua versi Windows yaitu Windows 95, 98, NT 3.1 dan 2000.

2.4.2 Perintah-perintah MCI (MCI Command Strings)

Perintah MCI merupakan perintah dalam bahasa manusia yang telah dibakukan oleh Microsoft dalam *Microsoft Multimedia Development Kit Reference Manual* dan tersusun secara abjad. Perintah ini dibagi dalam empat kategori dan hanya dapat digunakan oleh fungsi `mciSendString`, yakni;

a. Perintah-perintah System (*System Commands*)

Perintah yang berhubungan dengan sistem perangkat yaitu: **break, sound, sysinfo.**

b. Perintah-perintah Penting (*Required Commands*)

Perintah yang dapat didukung oleh semua perangkat MCI, diantaranya: **capability, close, info, open, status.**

c. Perintah-perintah dasar (*Basic Commands*)

Merupakan perintah-perintah opsional yang digunakan oleh perangkat MCI. Jika perintah dasar ini tidak dapat didukung oleh suatu perangkat maka perangkat akan menyampaikan pesan “*Action not available for this device*” yang berarti bahwa perangkat yang menerima perintah tidak mendukung perintah yang dimaksud. Perintah dasar ini diantaranya: **load, pause, play, record, resume, save, seek, status, stop.**

d. Perintah-perintah perluasan (*Extended Commands*)

Merupakan perintah spesifik yang hanya dapat bekerja pada suatu jenis perangkat. Misalnya: **set.**

2.4.3 Jenis-jenis perangkat MCI

Ada beberapa jenis perangkat MCI yang dapat didukung oleh standar MCI, yaitu:

- a. `cdaudio` (*CD Audio player*)
- b. `dat` (*Digital audio tape player*)
- c. `digitalvideo` (*Digital video*)

- d. MMMovie (*Multimedia Movie Player*)
- e. other (perangkat MCI tidak terdefinisi)
- f. overlay (Overlay device)
- g. scanner (Image scanner)
- h. sequencer (MIDI sequencer)
- i. vcr (Videotape player or recorder)
- j. videodisc (Videodisc player)
- k. waveaudio (perangkat *audio* yang dapat memainkan file wave. Seperti Kartu suara Soundblaster).
- l. MPEGVideo

Jenis perangkat MCI yang terinstalasi dalam sistem Windows akan didokumentasikan dalam *file* konfigurasi *driver* SYSTEM.INI pada bagian [mci] dan juga nama file driver yang digunakan oleh perangkat tersebut. Sebagai contoh:

```
[mci]
cdaudio=MCICDA.DRV
```

Sedangkan asosiasi file yang didukung oleh perangkat tersebut disimpan dalam file konfigurasi WIN.INI pada bagian [mci ekstensions].

Sebagai contoh:

```
[mci ekstensions]
cda=cdaudio
```

Perlu diketahui bahwa setiap perubahan terhadap konfigurasi driver perangkat MCI (SYSTEM.INI & WIN.INI) yang digunakan akan sangat berpengaruh terhadap perangkat itu sendiri.

2.4.4 Contoh-contoh penggunaan perintah MCI di Visual Basic

Berikut ini contoh penggunaan fungsi MCI di Visual Basic:

- a. Fungsi untuk membuka file multimedia menggunakan mciSendString:

```
Err_code = mciSendString("open C:\Music\canyon.mid alias canyon", "MPEGVideo", 0, 0)
```

- b. Fungsi untuk memainkan file multimedia

```
Err_code = mciSendString("play mpeg")
```

- c. Fungsi untuk berhenti sejenak memainkan file multimedia

```
Err_code = mciSendString("pause mpeg")
```

- d. Fungsi untuk berhenti memainkan file multimedia

```
Err_code = mciSendString("stop mpeg")
```

- e. Fungsi untuk menutup file multimedia

```
Err_code = mciSendString("close mpeg")
```

- f. Fungsi untuk mencari deskripsi kesalahan

```
Err_code = mciGetErrorString(Err_code,Deskripsi,128)
```