2. TEORI PENUNJANG

2.1. Ensiklopedia

2.1.1. Latar Belakang

Ensiklopedia adalah salah satu media edukasi yang cukup populer dikalangan para pelajar yang dapat diandalkan keakuratannya, beberapa jenis Ensiklopedia seperti yang berupa buku maupun media elektronik banyak tersedia dimana-mana dewasa ini, namun karena di Indonesia kurang begitu dimasyarakatkan dan mungkin biayanya cukup mahal, maka alat bantu yang bernama ensiklopedia itu jarang digunakan, apalagi yang merupakan buatan anak negeri, jumlahnya hanya beberapa biji saja, dengan adanya *software* ini diharapkan dapat membantu mengembangkan minat pelajar terhadap alat bantu pendidikan yang satu ini.

Pertama-tama, perlu juga diketahui, apa makna dari kata *ensiklopedia* (*encyclopedia*) itu sendiri, *encyclopedia* berasal dari bahasa *Yunani*, yaitu *enkyklios* + *paideia*, yang berarti sebuah hasil kerja yang mengandung informasi dari semua cabang ilmu pengetahuan atau penjelasan secara komprehensif dari cabang ilmu pengetahuan tertentu, biasanya dalam bentu kartikel yang disusun secara alpabet dan terkadang berdasarkan subjek.

Latar Belakang yang melatarbelakangi salah satu media pendidikan yang dinamakan Ensiklopedia ini adalah dikarenakan biasanya sistem pendidikan kurang begitu mengajarkan suatu bidang ilmu secara terperinci, atau biasanya kurang lengkap dan hanya diambil garis besarnya saja, jadi banyak para peneliti, badan penelitian dan para ilmuwan yang sering menyusun sebuah buku yang berisikan data dari bidang ilmu tersebut secara lengkap dan terperinci, bukannya tidak ada media lain yang lengkap, tetapi ensiklopedia bisa dijadikan acuan yang akurat dan terpercaya bahkan bisa dijadikan pegangan yang pasti, jadi Ensiklopedia ibaratnya seperti kompas yang akurat yang mampu memberikan arah yang tepat sehingga informasi yang dicari dapat dijamin keakuratannya.

Sistem Ensiklopedia ini diproses memang untuk memenuhi akan kebutuhan informasi yang sifatnya umum dan kompleks. Berdasarkan uraian diatas, maka jelaslah Ensiklopedia dapat digunakan oleh semua kalangan yang mana informasi tentang suatu bidang ilmu itu dibutuhkan, sehingga informasi yang diperoleh dapat dipastikan keakuratannya karena telah diproses sedemikian rupa oleh para ahli dan badan ilmu pengetahuan yang terkait. Secara global hasil informasi yang diperoleh dari Ensiklopedia akan sangat menunjang dalam proses pembelajaran untuk tercapainya suatu bentuk pola belajar aktif yang lebih baik.

2.1.2. Pengertian dan Perkembangan Ensiklopedia

Ensiklopedia selalu berkembang sesuai dengan tuntutan perkembangan dunia pendidikan dan ilmu pengetahuan, sampai sekarang ada berbagai tipe dari Ensiklopedia itu sendiri, seperti Ensiklopedia umum dan Ensiklopedia khusus serta Ensiklopedia Tema, maksudnya Ensiklopedia yang berisikan data-data tentang suatu bidang ilmu atau suatu bagian informasi, misal Ensiklopedia Fauna seri Mamalia, yang berisikan data-data tentang hewan mamalia secara lengkap dan akurat, begitu pula dengan tema yang lain.

2.1.3. Tujuan Ensiklopedia

Beberapa jenis Ensiklopedia mempunyai tujuan yang khusus, namun pada dasarnya sebuah Ensiklopedia bertujuan untuk memberikan informasi berupa data yang telah diolah sedemikian rupa sehingga mampu mencangkup sebuah bidang ilmu secara mendalam. Misal Ensiklopedia fauna seperti pada proyek skripsi ini, tujuannya yaitu memberikan informasi secara mendetail dan lengkap tentang seluruh jenis fauna sejauh yang telah dikelompokkan oleh para ahli, memang ada jenis ensiklopedia yang memberi batasan-batasan pada informasinya, semisal Ensiklopedia biologis fauna, maka Ensiklopedia itu hanya menjelaskan dunia fauna ditinjau dari segi biologis, sedangkan Ensiklopedia fauna umum seperti pada proyek ini bertujuan untuk menjelaskan garis besar data tentang fauna yang terkait.

2.1.4. Input Ensiklopedia

Ensiklopedia pada umumnya mendapat input data dari penelitian lapangan maupun dari informasi tertulis lainnya yang telah diteliti dan dipastikan keakuratannya melalui banyak metode penelitian dan thesis, apalagi di zaman yang serba modern ini, berbagai peralatan bantu dan teknologi informasi yang canggih dapat mempercepat proses pengumpulan data dan bahkan mengolahnya dengan tepat dan mengagumkan.

2.1.5. Proses Database

Langkah-langkah dasar dalam proses penyusunan database untuk Ensiklopedia ini yaitu antara lain:

- Mencari sumber-sumber yang terkait dan akurat, semisal buku-buku dan ensiklopedia sejenis. Sumber-sumber tersebut harus benar-benar mempunyai dasar yang kuat dan dipercaya, sehingga database yang dibuatpun tidak memberikan informasi yang salah.
- Mengelompokkan data berdasarkan beberapa golongan, misal dalam database ini yaitu golongan mamalia, reptilia, dan lain-lain.
- Menata isi database secermat dan seakurat mungkin sehingga tidak menimbulkan kesalahan informasi dikemudian hari.

Proses database yang dibuat ini melibatkan bahasa SQL yang pada Borland Delphi 5.0 dikenal dalam komponen *ADOQuery*, prosesnya cukup sederhana, yaitu mengakses database dengan mengeset nama database dan mengolahnya dengan bahasa SQL. Sehingga data dapat ditampilkan per-bagian dan terkelompok, semisal ingin menampilkan data hewan dari negara Indonesia, maka gunakan bahasa SQL yang nantinya hasilnya adalah data-data hewan dari negara Indonesia, untuk hal ini akan dijelaskan pada bab selanjutnya tentang penjelasan program.

2.1.6. Output Ensiklopedia

Pada dasarnya semua hal yang berbau sumber informasi tujuannya adalah memberikan informasi tentang apa yang menjadi lahannya, dalam hal ini, informasi yang diberikan oleh Ensiklopedia Fauna ini yaitu data-data/informasi tentang dunia fauna dan pernak-perniknya serta beberapa aplikasi tambahan yang membantu *user* dalam menggunakan informasi yang ada dalam program ini. Jadi secara garis besar, output dari program ini adalah informasi yang akurat tentang dunia fauna dan segala hal yang terkait dengannya, dan terutama data mengenai jenis hewan, lokasi dan data lainnya seperti foto, suara dan video. Sedangkan data biologis dan lainnya tidak dicantumkan disini, karena kurang sesuai dengan pembahasan dan kebutuhan.

2.2. Multimedia

Pada bagian ini akan dibahas pengertian multimedia, perkembangan multimedia, manfaat penggunaan multimedia, dan elemen multimedia terutama pada penggunaan multimedia pada komputer.

2.2.1. Pengertian Multimedia

Multimedia berasal dari kata *multi* yang berarti banyak dan *media* berarti bentuk atau sarana komunikasi. Dalam bidang komputer, multimedia adalah aplikasi atau presentasi berbasis komputer yang menggabungkan dua atau lebih media, seperti teks, suara, gambar, video, dan animasi¹.

Penggunaan multimedia pada bidang komputer menjadikan komputer lebih mudah dipakai, lebih nyaman, lebih menyenangkan dan lebih menarik bagi user. Multimedia dapat memberikan lebih banyak cara dalam menampilkan sebuah data atau informasi.

2.2.2. Sejarah Perkembangan Multimedia

Perkembangan dunia komputer yang begitu luar biasa mengakibatkan *user* dapat menikmati kemampuan audio visual yang semakin sempurna. Komputer terutama *personal computer* semakin berkembang dari hanya sebagai alat bantu para pelaku bisnis menjadi perangkat *edutainment* (*education and entertainment*) yaitu kombinasi antara pendidikan dan hiburan.

Perkembangan multimedia dipengaruhi oleh dua jenis teknologi yang saling terkait. Pertama, teknologi perangkat keras seperti teknologi yang digunakan untuk menangani grafik dilayar seperti teknologi monitor, kartu grafis, teknologi untuk

¹ Prabhat K. Andleigh dan Kiran Thakrar. Multimedia Systems Design. Upper Saddle River, New

menangani media penyimpanan seperti *HardDisk* dan *CD-ROM drive*, teknologi untuk menangani suara seperti kartu suara dan *speaker*, dan berbagai teknologi perangkat keras lainnya seperti *processor*, *scanner*, dan lain-lain. Kedua, teknologi perangkat lunak yang digunakan untuk penyimpanan data, pemroses data, dan cara menampilkan data ke layar komputer.

2.2.3. Manfaat Multimedia

Manfaat-manfaat yang dimiliki oleh multimedia antara lain adalah:

- 1. Multimedia menjadikan pemakai lebih mudah dan cepat dalam mempelajari suatu aplikasi.
- 2. Multimedia menjadikan suatu aplikasi menjadi lebih "hidup" karena pengguna dapat berinteraksi dengan sistem.
- Multimedia menjadikan aplikasi menjadi lebih menarik karena dapat memberikan suatu hiburan.

Multimedia memberi lebih banyak pilihan dalam menyampaikan informasi. Sehingga jika salah satu saluran informasi terputus, orang masih dapat mengunakan saluran informasi lainnya.

2.2.4. Elemen Multimedia

Multimedia terdiri dari beberapa elemen, antara lain teks, audio, image, video, dan animasi. Elemen- elemen multimedia tersebut sering kali dikombinasikan untuk mendapatkan hasil yang terbaik. Beberapa elemen multimedia tersebut bahkan merupakan kombinasi yang lainnya. Pada sub bab berikut ini akan dijelaskan tentang elemen-elemen multimedia tersebut.

2.2.4.1. Text

Teks adalah elemen multimedia yang menitikberatkan penggunaan mata sebagai alat untuk menangkap informasi. Teks merupakan jenis penyampaian data yang paling umum dan paling sederhana karena hanya membutuhkan sedikit ruang pada media penyimpanan dibandingkan gambar dan video. Tipe data teks dapat dibuat field- field dalam sebuah database sehingga dapat diindeks, dicari, dan

diurutkan². Sebagai contoh program aplikasi Ensiklopedia ini sebagian berisi teks karena merupakan konversi dari Ensiklopedia seri Fauna yang sebagian berisi teks dan menjadi bentuk yang terkomputerisasi. Penggunaan media teks sebagai sarana komunikasi antara komputer dengan manusia memiliki banyak kelemahan karena mata manusia mudah sekali lelah jika digunakan terlalu lama. Walaupun mengunakan monitor yang telah memenuhi standar kesehatan, radiasi yang berasal dari monitor itu tetap berpengaruh pada kesehatan mata. Penggunaan teks yang terlalu berlebihan akan sangat membosankan bagi penggunanya. Oleh sebab itu penguaan teks dalam presentasi hanya terbatas pada bagian-bagian yang penting. Media Teks seperti media visual lainnya membutuhkan tempat pada layar monitor akan tetapi media teks memiliki kelebihan yaitu dapat digulung keatas-kebawah dan kekiri-kekanan sesuai dengan bagian informasi yang hendak dibutuhkan tanpa mengurangi arti informasi tersebut, sehingga media visual ini dapat dihemat keberadaannya di layar monitor. Media teks dapat digabungkan dengan media lainnya seperti gambar untuk memperjelas arti media tersebut. Misalnya gambar hewan dapat ditambahkan informasi tentang nama ilmiah dan asalnya.

2.2.4.2. Audio dan Suara

Elemen kedua dari Multimedia adalah suara (*audio* dan *voice*), elemen ini menitikberatkan penggunaan telinga sebagai alat utama dalam menangkap informasi. Suara dapat berbentuk musik, suara manusia, perintah dengan suara manusia, percakapan telepon dan lain sebagainya. Suara dapat menjadi pilihan alternatif pengguna komputer terutama jika terjadi gangguan penglihatan. Keuntungan dari media *audio* adalah tidak memakai ruang dalam tampilan monitor, berbeda dengan media visual dimana semakin banyak informasi yang ditampilkan berarti membutuhkan banyak ruang di dalam layar monitor, jika ruang layar monitor sudah tidak mampu menampung lagi maka akan banyak informasi yang tersembunyi. Keuntungan lainnya adalah pengunaan suara dapat mengurangi kontak mata dengan monitor bahkan penguna komputer bahkan dapat meninggalkan komputer dalam jarak yang cukup jauh tanpa takut kehilangan informasi yang dibutuhkan karena informasi berupa media *audio* dapat didengarkan dari segala arah. Oleh sebab itu,

² Ibid, p.33

pengguna tidak perlu memusatkan perhatian ke sumber penghasil informasi audio. Informasi berupa *audio* sangat cocok untuk mengatasi keterbatasan ruang layar tampilan.

Dalam mendisain bunyi yang akan digunakan pada sebuah *interface*, seorang perancang *interface* harus mempertimbangkan faktor psikoakustik. Faktor psikoakustik ini meliputi pengetahuan tentang bunyi dan sistem pendengaran manusia, komponen utama bunyi, lokasi sumber bunyi, faktor-faktor yang dapat mempengaruhi persepsi bunyi, dan pengenalan pola bunyi. Pengetahuan tentang sistem pendengaran manusia sangat berguna untuk merepresentasikan data secara audio karena sifat-sifat sistem pendengaran bisa menunjukkan beberapa keterbatasan yang harus diatasi oleh tampilan *audio* agar tampilan *audio* yang didisain bisa efektif. Menambahkan bunyi pada *interface* tanpa mempertimbangkan faktor psikoakustik bisa menyebabkan *user* tidak dapat membedakan bunyi yang satu dengan bunyi yang lain dan tidak dapat mendengarkan bunyi dengan baik ataupun mengingatnya.

Batas frekuensi maksimum yang masih bisa didengar secara jelas oleh seseorang akan semakin menurun seiring dengan bertambahnya umur. Anak-anak dapat mendengarkan bunyi dari yang berfrekuensi 20 Hz sampai 20 kHz. Setelah mencapai umur 30-an, biasanya seseorang dapat mendengarkan bunyi dengan frekuensi tidak lebih dari 18 kHz, sedangkan batas frekuensi maksimum yang masih dapat didengar setelah mencapai umur 50-an adalah 14 kHz, dan batas frekuensi maksimum yang masih dapat didengar setelah mencapai umur 70-an adalah 10 kHz. Oleh karena itu, agar *interface* yang menggunakan bunyi bisa berguna untuk semua kalangan usia, sebaiknya frekuensi bunyi dijaga supaya tetap berada di bawah 5 kHz dan di atas 130 Hz³.

Penggabungan suara ke dalam sebuah aplikasi multimedia dapat memberi informasi yang tidak mungkin dapat diperoleh dari metode komunikasi yang lain. Sebagai contoh adalah sebuah lagu atau musik hanya dapat dinikmati jika menggunakan suara. Suara juga dapat memperjelas pengertian pemakai tentang

³ Patterson, R.D. (1982). *Guidelines for auditory warning systems on civil aircraft. Civil Aviation Authority*, London.

informasi yang disajikan pada jenis media yang lain. Sebagai contoh, sebuah narasi dapat menggambarkan apa yang terlihat pada sebuah potongan animasi.

Penambahan *audio* pada *interface* dapat memperkaya cara interaksi manusia dengan komputer, dimana manusia bisa memperoleh informasi melalui lebih dari satu indera, tidak hanya melalui sistem visual saja tetapi juga bisa melalui sistem *audio* sehingga interaksi manusia dengan komputer menjadi semakin alami seperti interaksi manusia sehari- hari.

Walaupun berperan penting dalam kehidupan sehari-hari, *non-speech audio*, bunyi yang bukan merupakan hasil pengucapan manusia, belum banyak dimanfaatkan dalam interaksi manusia dengan komputer. Padahal pemakaian *nonspeech audio* yang sesuai bisa meningkatkan fungsionalitas *interface* karena *nonspeech audio* bisa digunakan untuk merepresentasikan *event*, objek atau data komputer.

Salah satu pedoman dalam penggunaan audio terutama *audio non-speech* adalah seperti yang dicetuskan oleh *Deatherage* (1972). Pada pedoman ini, *Deatherage* menunjukkan kapan pemakaian display audio lebih tepat dibandingkan dengan *display* visual⁴. Pedoman ini dapat dilihat pada Tabel 2.1.

]	Presentasi audio dipakai apabila:	Presentasi visual dipakai apabila :							
1.	Pesan yang disampaikan sederhana	1. Pesan yang disampaikan kompleks							
2.	Pesan yang disampaikan pendek	2. Pesan yang disampaikan panjang							
3.	Pesan tidak dipakai kemudian	3.	Pesan akan dipakai kemudian						
4.	Berhubungan dengan kejadian pada	4.	Berhubungan dengan lokasi pada						
	suatu saat		ruang						
5.	Membutuhkan tindakan dengan	5.	Membutuhkan tindakan dengan						
	segera		segera						
6.	Sistem visual user terganggu	6.	Sistem pendengaran user						
			terganggu						

Tabel 2.1. Kapan Display Audio dan Display Visual Digunakan

⁴ Deatherage, B.H. (1972). Auditory and other forms of information presentation, p. 124.

Washington D.C.: U.S. Government printing office.

7.	Lokasi terlalu terang atau gelap	7.	Lokasi terlalu bising
8.	Operator bergerak terus menerus	8.	Operator tidak banyak bergerak

Format *audio* antara lain adalah WAV, Midi dan MP3. Pengunaan MP3 semakin digemari karena ukuran filenya sangat kecil dibandingkan dengan WAV.

2.2.4.3.Image

Elemen multimedia yang ketiga adalah gambar (*image*). Gambar memiliki peranan yang sangat penting dalam multimedia karena *image* mampu mewakili ribuan kata dan merupakan jembatan bagi keanekaragaman bahasa yang ada di dunia ini. Kemampuan ini ada karena kemampuan visual bersifat lebih universal. Inilah yang menjadikan gambar memiliki fungsional yang lebih karena gambar mampu memberikan suatu instruksi tanpa kata-kata. Media gambar digunakan jika tidak membutuhkan gerakan, misalnya dengan gambar saja sudah dapat menerangkan bentuk seekor hewan. Seperti media visual lainnya gambar membutuhkan ruang didalam layar monitor. Sebenarnya gambar juga dapat digulung atau dipotong jika informasi yang dibutuhkan hanya sebagian saja.

Media Gambar terdiri dari gambar yang terlihat dan gambar yang tidak terlihat. Kelompok gambar yang terlihat termasuk gambar tangan (*drawings*) seperti cetak biru (*blueprints*), gambar arsitek, dan gambar interior, dokumen yang di-*scan* sebagai gambar, *painting* dari hasil *scanner* atau dari hasil penggambaran mengunakan aplikasi komputer, foto dari hasil *scanner* atau langsung mengunakan sebuah kamera elektronik. Kelompok gambar yang tidak terlihat yaitu gambar yang tidak disimpan sebagai gambar akan tetapi ditampilkan sebagai gambar. Sebagai contoh adalah penunjuk tekanan, penunjuk temperatur dan display meteran lainnya.

Beberapa format gambar seperti JPEG, GIF dan BMP. Format file gambar BMP sangat besar akan tetapi kecepatan aksesnya sangat tinggi. Sebaliknya format file gambar JPEG sangat kecil tetapi waktu aksesnya sangat lama.

File gambar JPEG (Joint Photographic Experts) merupakan kompresi yang menghilangkan sebagian resolusi. Akan tetapi, karena resolusi foto biasanya terlalu tinggi maka penghilangan sebagian resolusi tidak begitu tampak. Biasanya format JPEG digunakan untuk menyimpan gambar foto karena memiliki banyak warna dan memiliki banyak gradasi warna sehingga metode kompresi dari JPEG menjadi efektif. Oleh sebab itu, penggunaan format gambar sangat tergantung kebutuhan.

2.2.4.4.Video

Elemen lain dari multimedia adalah video. Video merupakan gabungan dari media gambar dan suara. Media ini diambil dengan menggunakan alat perekam Video seperti *handycam*. Media ini merupakan media terlengkap dari pada elemen media lainnya. Akan tetapi membutuhkan ruang yang sangat besar untuk menyimpannya.

Video digunakan untuk memberikan cuplikan atau demonstrasi mengenai suatu hal tertentu, misalnya video mengenai kehidupan hewan. Format Video yang sering dipakai adalah MPEG-2 (Moving Picture Experts Group), AVI (Audio Video Interleave) dan Quicktime.

File AVI merupakan standar di dalam sistem operasi Windows, untuk menjalankan file AVI ini membutuhkan biaya yang rendah, dan mendukung beberapa algoritma kompresi, akan tetapi karena memiliki resolusi yang rendah dan ukuran rata-rata dari file ini adalah sepuluh kali lebih besar dari MPEG-2 maka sebaiknya tidak digunakan.

2.2.4.5.Animasi

Selain gambar diam, juga ada gambar yang bergerak yang lebih dikenal dengan istilah animasi. Apa yang sebenarnya dimaksud dengan animasi adalah suatu kumpulan pergerakan yang dilakukan secara berurutan dari suatu rangkaian *frameframe*.

Animasi pada multimedia biasanya terdiri dari gambar atau image non-foto yang ditampilkan berurutan dengan cepat, sehingga menimbulkan kesan bahwa gambar-gambar yang ditampilkan bergerak.

Animasi dalam beberapa hal hampir sama dengan video. Keduanya melibatkan serangkaian gambar yang ditampilkan berurutan dengan cepat untuk menghasilkan gerakan. Perbedaan utama antara keduanya adalah bagaimana cara pembuatannya. Video menggambarkan situasi kehidupan nyata, seperti seseorang yang sedang berjalan atau sebuah pertandingan balap mobil yang sedang berlangsung. Untuk melakukan hal ini diperlukan sebuah kamera video untuk menangkap image. Sedangkan animasi diawali dengan gambar atau *image* non-foto yang digabung dengan proses tertentu.

Di dalam multimedia, animasi ini memegang peranan yang cukup penting, karena dengan adanya animasi maka tentu saja tampilan yang dibuat jauh lebih menarik dan interaktif akan tetapi animasi sebaiknya digunakan dengan bijaksana karena terlalu banyak animasi yang tidak dibutuhkan akan menganggu konsentrasi penguna komputer. Sama halnya dengan gambar diam, gambar bergerak ini juga mempunyai beberapa format penyimpanan, salah satunya yang terkenal adalah GIF Animasi dan FLI.

2.3. Software Aplikasi Penunjang

2.3.1. Borland Delphi 5.0

Pada bagian ini akan dijelaskan pemrograman database dan aplikasi multimedia yang telah diaplikasikan pada proyek Tugas Akhir ini. Penjelasan akan dibagi menjadi dua bagian, yaitu bagian pemrograman database dan pemrograman aplikasi multimedia yang melibatkan komponen MediaPlayer dan sebagainya.

2.3.1.1.Pemrograman Database menggunakan Borland Delphi 5.0

Delphi merupakan program aplikasi database berbasis Windows. Delphi juga merupakan perangkat lunak yang dikembangkan oleh Borland dan Delphi 5.0 merupakan pengembangan versi-versi sebelumnya. Kelebihan Delphi antara lain mempunyai kemampuan untuk menggunakan bahasa SQL yang merupakan bahasa query yang handal. Delphi juga dapat dipakai di Internet. Delphi 5.0 dikembangkan dengan melanjutkan apa yang sudah dikerjakan di Delphi 4 yaitu penambahan fasilitas-fasilitas sehingga mempermudah pengembangan aplikasi, dengan cara mengurangi pengetikan sehingga pengkonsentrasian pada apa yang akan dibuat bukan pada bagaimana cara membuatnya. Delphi 5.0 juga menambahkan hal terpenting dari sebelumnya, yaitu stabilitas.

Borland Delphi 5.0 yang untuk selanjutnya disingkat dengan Delphi merupakan program aplikasi database yang berbasis Object Pascal dari Borland. Selain itu, Delphi juga memberikan fasilitas pembuatan aplikasi visual seperti Visual Basic. Mengapa harus *Delphi*? Apa kelebihan *Delphi* dibandingkan dengan program aplikasi lainnya? Ada dua tipe jawaban untuk pertanyaan-pertanyaan di atas, yaitu jawaban panjang dan jawaban singkat. Jawaban singkat adalah produktifitas. Dengan memakai *Delphi* adalah cara yang paling sederhana untuk membangun aplikasi berbasis *Windows*. Bagi sebagian orang jawaban tersebut kurang memuaskan, oleh sebab itu diperlukan jawaban panjang. Produktifitas dari pengembangan perangkat lunak dibagi menjadi lima atribut penting, yaitu:

- Kualitas dari lingkungan pengembangan visual.
- Kecepatan compiler dibandingkan dengan kompleksitasnya.
- Kekuatan dari bahasa pemrograman dibandingkan dengan kompleksitasnya.
- Fleksibilitas dari arsitektur basis data.
- Pola desain dan pemakaian yang diwujudkan oleh *framework*-nya.

Beberapa hal yang harus dimengerti dan beberapa komponen yang paling sering digunakan dalam pembuatan program aplikasi *database*, antara lain:

Data Module (TDataModule) adalah sebuah klas pada *Delphi* yang dikhususkan untuk menampung komponen-komponen *non visual*, umumnya adalah komponen-komponen milik palet *Data Access*.

Untuk membuat *Data Module* pilih main menu *File/New Data Module*. Jika ingin mengakses satu *Data Module*, maka pada unit yang bersangkutan tambahkan *Data Module* (*File/Use Unit*).

Untuk operasi *database*, *Delphi* mempunyai kemampuan untuk mengakses berbagai jenis data dengan menggunakan *Borland Database Engine (BDE)* atau yang juga disebut dengan *IDAPI*. Tanpa menggunakan *BDE* tersebut, untuk mengakses *database* lokal atau *server* harus menggunakan fungsi dari *vendor* lain.

BDE adalah alat perantara yang memberikan akses yang terpisah atas data yang sedang diakses. *Programmer* tidak perlu menghiraukan lagi bagaimana pengaksesan data sesungguhnya yang dilakukan *BDE*. Dengan *BDE* dapat dihasilkan suatu aplikasi yang memiliki kemampuan akses beberapa jenis data sekaligus, misalkan: aplikasi untuk mengakses data dari *server sybase* (untuk sebuah fungsi) dan *server oracle* (untuk fungsi lain).

Alias adalah nama lain yang diberikan untuk sebuah *database*, baik yang berada pada *disk* lokal ataupun jaringan. Keuntungan menggunakan *alias*, yaitu:

- 1. Kemampuan memisahkan data dan kode program.
- 2. Kemampuan menggunakan sekumpulan data yang berbeda dalam satu *alias*.
- 3. Kemampuan memindahkan data aplikasi dalam *database* lokal ke *database client/server*.

Cara membuat alias, antara lain:

- 1. Menggunakan BDE Configuration Utilitiy atau BDE Administrator.
- 2. Menggunakan Database Desktop.
- 3. Menuliskan *alias* dalam program, kemudian menyimpannya dalam konfigurasi *file*.

Database Desktop (DBD) dapat dianggap sebagai versi mini dari *Paradox* atau *dBase for Windows*. *DBD* menyediakan metode untuk membuat, melihat, mengedit, mengubah struktur, mengindeks, mengurutkan, membuat *query*, dan memanipulasi tabel, serta membuat *alias*.

Komponen DBGrid digunakan untuk menampilkan data dalam bentuk tabel.

Komponen DBNavigator digunakan untuk mendukung kemampuan navigasi dalam operasi tabel. Komponen ini berupa sekelompok tombol.

Komponen DBText digunakan untuk menampilkan data dari sebuah *field* tertentu.

Komponen DBEdit digunakan supaya *user* dapat meng-edit sebuah *field* data.

Komponen DBMemo digunakan untuk memberikan presentasi *visual* dari sebuah *field* memo milik sebuah *database*.

Komponen DBImage digunakan untuk memberikan persentasi *visual* dari sebuah *field* yang menyimpan data grafis.

Komponen DataSource (TDataSource) digunakan untuk menghubungkan komponen *Table* atau *Query* dengan komponen *database visual* (mis: *DBGrid*). Komponen ini mempunyai fungsi utama mengambil data dari tabel yang sudah didefinisikan pada komponen *Table*, *Query*, atau *StoredProc*. Properti *Dataset* akan menghubungkan *DataSource* dengan *Table* atau *Query* yang ada.

Komponen ADOConnection dan ADOQuery yang berada pada palet ADO adalah komponen utama yang diperlukan untuk menghubungkan aplikasi dengan data pada *database* yang penggunaannya tidak memerlukan pengesetan ODBC. Dengan menggunakan *provider Microsoft Jet 4.0 OLEDB provider*, maka database dapat diakses tanpa lagi menggunakan *alias* atau pengesetan ulang seperti pada komponen yang menggunakan ODBC sebagai penghubungnya.

Sedangkan cara-cara penggunaan dan pengaplikasiannya dapat dilihat pada bagian implementasi program dan juga dapat dilihat secara keseluruhan pada bagian lampiran program.

Komponen TextClient dan TextServer yang berada pada palet *dWinsock* adalah beberapa komponen yang digunakan untuk menghubungkan komputer secara LAN(local Area Network), sehingga data dapat diakses oleh beberapa komputer secara bersamaan.

Cara-cara penggunaan dan pengaplikasiannya dapat dilihat pada bagian implementasi program dan juga dapat dilihat secara keseluruhan pada bagian lampiran program.

Komponen Tabel (TTable) dan Query (TQuery) yang berada pada palet Data Access adalah komponen utama yang digunakan untuk menghubungkan aplikasi dengan data. Komponen ini meyediakan akses langsung ke setiap record dan field dari berbagai macam jenis tabel, misalnya: Paradox, dBase, Access, FoxPro, ODBCcompliant, ataupun juga SQL database pada sebuah remote server, seperti InterBase, Oracle, Sybase, MS-SQL Server, Informix, DB2.

Ada beberapa macam cara untuk mengakses nilai *field* dari sebuah tabel, yaitu:

1. NamaTabel['NamaField'], mis:

Table1['kodecust']:= '123';

- 2. NamaTabel.FieldValues['NamaField'], mis: Table1.FieldValues['kodecust']:= `123';
- 3. NamaTabel.FieldByName('NamaField').As<tipedata>, mis: Tabel1.FieldByName('kodecust').AsString:= '123'; Tabel1.FieldByName('kodecust').AsBoolean:= true; Tabel1.FieldByName('kodecust').AsInteger:= 100;

Berikut ini contoh untuk menambahkan data pada akhir record:

```
// set pointer ke akhir table dan siap menambah data
Tablel.Append;
Table1[`Nama'] := Edit1.Text;
Table1[`NRP'] := Edit1.Text;
// menyimpan perubahan yang terjadi
Table1.Post;
```

Ada beberapa macam cara untuk mengakses nilai *field* dari sebuah *query*, yaitu antara lain:

```
1. NamaQuery['NamaField'], mis:
      Query1[`kodecust']:= `123';
2. NamaQuery.FieldValues['NamaField'], mis:
      Query1.FieldValues[`kodecust']:= `123';
3.NamaQuery.FieldByName('NamaField').As<tipedata>, mis:
      Query1.FieldByName('kodecust').AsString:= '123';
      Query1.FieldByName('kodecust').AsBoolean:= true;
      Query1.FieldByName(`kodecust').AsInteger:= 100;
 Berikut ini contoh untuk menambahkan data pada akhir record:
             // Menutup query lebih dulu
             Query1.Close;
             // Menghapus perintah SQL yang ada
             Query1.SQL.Clear;
             // Menambahkan perintah SQL
             Query1.SQL.Add('SELECT * FROM Orders');
             Query1.SQL.Add('WHERE NoOrder < 1000');</pre>
             // Mempersiapkan query
             Query1.Prepare;
             // Mengaktifkan query
```

```
Query1.Open;
```

Komponen *Query* pada *Delphi* dapat memiliki *parameter*. Dengan membuat *query* yang ber-*parameter*, maka tidak perlu lagi menuliskan perintah *SQL* yang sama berulang, tetapi cukup mengganti *parameter*-nya saja. Untuk membuat sebuah perintah *SQL* yang ber-*parameter*, cukup dengan menuliskan perintah *SQL* pada properti *SQL*, dan kemudian mendifinisikan tipe dari *parameter*-nya pada properti *Params*. Variabel *parameter* selalu diawali dengan titik dua (:). Berikut ini contoh perintah *SQL* yang memiliki dua buah parameter *Var_Nama* dan *Var_Umur*:

SELECT * FROM Customer

```
WHERE Nama = :Var_Nama AND
Umur = :Var_Umur
ORDER BY Nama;
```

Ada beberapa macam cara untuk mengakses nilai parameter pada waktu runtime, yaitu:

- NamaQuery.Params.Items[NoIndex].As<tipedata>, mis:
 Query1.Params.Items[0].AsString:= `Budi';
- 2. NamaQuery.Params.ParamValues['NamaParameter'], mis: Query1.Params.ParamValues['Var_Nama']:= 'Budi';
- 3. NamaQuery.ParamByName('NamaParameter').As<tipedata>, mis:
 Query1.ParamByName('Var_Nama').AsString:='Budi';
 Query1.ParamByName('Var_Umur').AsInteger:= 20;

Komponen *Table* dan *Query* mempunyai *method Locate* dan *LookUp* yang digunakan untuk mencari data tertentu. *Method Locate* akan memindahkan posisi *pointer* ke baris pertama dari *record* yang sesuai dengan kriteria pencarian, sedangkan *method LookUp* akan mencari *record* yang sesuai dengan kriteria pencarian, tetapi hanya mengembalikan nilai *field* yang dicari, dan tidak memindahkan posisi *pointer*.

Untuk mengatur relasi antar *database* atau tabel, komponen *Table* mempunyai properti *MasterSource* dan *MasterFields*. Properti *MasterSource* digunakan untuk menentukan nama komponen *DataSource* dari tabel *master* yang akan digunakan oleh tabel detail untuk membentuk sebuah relasi *master-detail*. Pada waktu mendesain, kita bisa memilih nama-nama *DataSource* yang tersedia pada menu *drop-down* dari properti *MasterSource* tersebut.

Setelah nilai properti *MasterSource* diset, maka berikutnya properti *MasterFields* juga harus diset. Properti *MasterFields* akan menspesifikasikan satu atau lebih *field* dalam *master table* yang akan dihubungkan dengan *field-field* pada tabel detail ini untuk membentuk hubungan relasi *master-detail*. Pada waktu mendesain, bisa digunakan *Field Link Designer* untuk membentuk relasi *master-detail* antara dua tabel.

Pada komponen *Query*, untuk pembentukan relasi *database* atau tabel digunakan properti *DataSource* yang akan menentukan komponen *DataSource* mana yang digunakan untuk mengambil atau mengisi *field-field parameter* pada properti

SQL. Properti *DataSource* yang akan diisikan harus menunjuk pada komponen *DataSource* lain yang tidak digunakan untuk komponen *Query* ini.

Jika variabel *parameter* pada *SQL* tidak diset pada waktu desain, maka *Delphi* akan mencoba menset nilai variabel tersebut berdasarkan *field-field* yang terdapat pada tabel atau *query* yang ditunjuk properti *DataSource* yang terdapat pada komponen *Query* tersebut. Properti *DataSource* pada komponen *Query* ini dapat menunjukan pada komponen *Table* atau komponen *Query* yang lain.

2.3.1.2. Pemrograman aplikasi multimedia yang melibatkan komponen MediaPlayer

Multimedia merupakan teknologi komputer yang memungkinkan komputer dapat digunakan untuk menyajikan gambar, tulisan, dan juga suara. Dengan menggunakan multimedia, bahkan bisa membuat aplikasi yang melibatkan video yang terdapat pada VCD. Penjelasan tentang Multimedia ini sedikit banyak telah dijabarkan dibagian depan, jadi tidak akan diulang lagi disini.

Untuk mempermudah penggunaan multimedia, Delphi menyediakan komponen yang disebut MediaPlayer. Komponen ini terdapat pada halaman System pada Component Palette dan dinyatakan dengan ikon

Komponen ini dapat digunakan untuk memainkan:

- Suara yang terdapat pada berkas berekstensi .WAV(Wave audio) maupun .MID(MIDI=Musical Instrument Digital Interface), bahkan file-file MP3,
- Lagu pada CD(Compact Disc),
- Video baik yang berbentuk AVI, MPEG, maupun DAT(Digital Audio Tape).

Pada form, tampilan komponen ini dapat dilihat pada Gambar 2.1.



Gambar 2.1. Contoh Komponen MediaPlayer

Adapun maksud tombol-tombol tersebut dapat dilihat pada Gambar 2.2.



Gambar 2.2. Fungsi tombol-tombol pada MediaPlayer

- Play untuk memainkan
- Pause untuk menghentikan atau meneruskan(bekerja bergantia).
- Stop untuk menghentikan
- Next untuk menuju ke track berikutnya (atau ke akhir media kalau media tidak puya track).
- Prev untuk menuju ke track sebelumnya (atau ke awal media kalau media tidak punya track).
- Step untuk menuju kedepan beberapa frame.
- Back untuk menuju kebelakang beberapa frame.
- Record untuk memulai rekaman.
- Eject untuk mengeluarkan media (misalnya pada CD).

MediaPlayer memiliki sejumlah properti. Beberapa diantaranya dapat dilihat pada Tabel 2.2.

Properti	Keterangan
Display	Menentukan tempat yang digunakan untuk keluaran
	multimedia
DisplayRect	Menentukan ukuran untuk keluaran multimedia
FileName	Menentukan berkas multimedia yang akan diproses
Name	Nama untuk komponen ini

Tabel 2.2. Beberapa properti pada MediaPlayer

Length	Bertipe	Longint.	Menya	ıtakan	panjang	g media
	(diukur	dengan wał	ctu).			
Position	Bertipe Longint. Menyatakan posisi sekarar					
	media	(Misalnya	posisi	suara	yang	sekarang
	dimaink	an).				

Beberapa metode pada MediaPlayer dapat dilihat pada Tabel 2.3.

Metode	Keterangan
Open	Untuk membuka media
Play	Untuk memainkan
Pause	Untuk menghentikan sejenak
Stop	Untuk menghentikan
Next	Untuk menuju ke track berikutnya
Previous	Untuk menuju ke track sebelumnya
Step	Untuk menuju ke depan sejumlah frame(jumlah
	frame ditentukan oleh properti frame)
Back	Untuk menuju ke belakang sejumlah frame(jumlah
	frame ditentukan oleh properti frame)
StartRecording	Memulai perekaman
Eject	Mengeluarkan media(misalnya pada CD)
Resume	Meneruskan proses rekam atau main yang terhenti
Rewind	Meletakkan posisi awal milik media

Tabel 2.3. Beberapa metode pada MediaPlayer

• Membuat Program Sederhana yang Menggunakan MediaPlayer

Untuk mempraktekkan MediaPlayer, mula-mula tutuplah semua berkas yang terbuka (via menu <u>File</u> | Close All). Setelah itu buatlah proyek baru via menu <u>File</u> | New Application. Kemudian perkecil form dan ubahlah Caption milik form menjadi:

Contoh Program MediaPlayer

Ubahlah pula properti Name milik form menjadi FormMedia.

Berikutnya, simpan proyek dan form tersebut melalui <u>File</u> | Save Project As... ke folder tertentu yang telah ditentukan. Namakan Fmedia untuk form dan Media untuk proyek, atau nama lain sesuai keinginan.

Kemudian tambahkan komponen *MainMenu*, *OpenDialog*, dan *MediaPlayer* sebagaimana terlihat pada gambar 2.3.:

Į	ł		P	ro	g	I	ar	n	h	1	eo	li	al	Pl	a	ye	er																							Ę		ļ	×
	Built	1	Ē	-	****		6	2]	うちろう	1	1	>		100	[ţ		1		1000		>	1		ŀ	<		1	Þ		I	-		1	100 M	•	5		4	•		* * * *
	1	2	e.	~	1	-	2	2	*	-	10	1	1	-	1	0	0	3	1	1		-	3	3		2	3	-	1	3	3	1	3	3		1	3	3	1	3	3		
23	8		2	2	2	2	2	2	9	3	3	8	8	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3
			2	2	2	2	2	2	2					2	2	2		2	2	2		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		2
			ŝ.	2	2	2	2	2	2	2				2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
	0		÷.	ř.	×.	×.	×.	x	*							*				x		x	x	×	ž	×	×	×	×	×	×	x	ž	×	x	x	x	x	x	x	x		
	0				*	*	*	*	*							*							*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	×		
		•			*	*	÷	÷	*	1	1	1	1	1	1	×.	1	1	1	1	1	×.	÷	*	*	*	*	÷	*	÷	*	÷	*	*	*	÷	÷	÷	÷	÷	*		*
2.0		•	•	\$	è.	è.	2	÷	Ċ	Ċ	10	1	1	Ċ	Ċ	Ċ	Ċ	Ċ	Ċ	Ċ	Ċ	Ċ.	÷	*	1	1	*	Ċ	1	Ċ	1	Ċ	1	1	1	÷	÷	÷	÷	÷	÷	1	1
1			٥.	٥.	0	0	Ċ	Ċ	٢	Ċ	13	10	10	Ċ	Ċ	Ċ	Ċ	Ċ	Ċ	Ċ.	Ċ	Ċ.	Ċ	1	Ċ	1	1	Ċ	0	٢	0	Ċ	Ċ	1	1	Ċ	Ċ	Ċ	Ċ	Ċ	Ċ	1	1
	2		3	¢,	Ċ,	Ċ,	Ċ,	0	3	3	10	0	0	13	10	3	3	13	13	3	3	3	0	0	0	0	0	3	0	٢	0	3	0	Ċ	0	3	3	3	3	3	1	3	0
			8	9	9	9	9	9	9	3	3	8	8	8	8	9	3	3	3	3	3	3	3	2	9	2	2	9	2	9	2	9	9	9	2	8	9	8	9	9	3	8	9
			2	9	9	2	2	2	9	9	3	9	9	3	3	9	9	3	3	3	9	3	2	2	2	2	2	2	2	9	2	2	2	2	2	2	2	2	2	2	2	9	9
			ŝ.	2	2	2	2	2	2	2	2			2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
	0		ř.	ž.	×.	×.	×.	x															*		x	÷				÷		x	x	÷		÷.	×.	x	x	x	*		
	0	•		*	*	*	*	*	*							*							*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		*
2.0			•	*	*	*	*	*	*	1	1	1	1	1	1	*	1	1	1	1		1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	*
20	2		*	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20			1	0	0	0	0	1	1	1	10	10	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	1	1	1	1	1	3	1
20	22		0	0	0	0	0	0	0	1	10	10	10	1	1	3	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	٢
			3	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	3	0	3	0	0	3	0	3	0	3	0	0	0	3	0	0	0	0	0	0	3	0
			9	9	9	9	9	0	0	3	3	3	3	3	3	9	3	3	3	3	3	3	0	9	0	0	9	0	9	9	9	0	0	0	9	0	0	0	0	0	0	0	0
2			2	2	2	2	9	2	9	9	9	3	3	3	3	3	3	3	3	3	3	3	9	9	9	9	9	9	9	9	9	9	9	2	9	9	9	9	2	9	9	3	2

Gambar 2.3. FormMedia pada mode perancangan

Atur properti-properti komponen tersebut sehingga menjadi seperti berikut:

Komponen	Properti	Isi
MainMenu	Name	MainMenuMedia
OpenDialog	Name	OpenDialogMedia
MediaPlayer	Name	MediaPlayer1

Tabel 2.4. Isi dari beberapa Properti-properti Komponen

Untuk membuat menu, klik ganda pada komponen MainMenu yang terdapat pada form. Langkah ini menampilkan Menu Designer. Selanjutnya ketikkan **&Berkas** dan tekan Enter. Lalu ketikkan **B&uka...** dan tekan Enter. Hasilnya adalah sebagaimana terlihat pada gambar 2.4.



Gambar 2.4. Tampilan pada Menu Designer

Klik pada item **Berkas** dan kemudian berikan nama MenuBerkas pada properti **Name**. Untuk memberikan nama pada item menu **Buka...**, klik pada item **Buka...**, dan kemudian berikan nama **MenuBuka** pada properti **Name**.

Untuk menuliskan kode pada kejadian(event) *OnClick* milik item menu **Buka...**, klik ganda pada item ini. Langkah ini menampilkan jendela kode. Tuliskan kode(yang tidak dicetak tebal) berikut sehingga kode lengkap untuk kejadian **OnClick** menjadi seperti berikut:

Akhiri dengan mengklik ikon pada *Menu Designer*. Bentuk form setelah langkah diatas adalah seperti gambar 2.5.



Gambar 2.5. Form mengandung menu

Pada kode yang anda tuliskan di depan, *OpenDialogMedia.Execute* digunakan untuk menjalankan kotak dialog untuk memilih berkas multimedia. Bila *Execute* menghasilkan *True* maka properti *FileName* milik *MediaPlayer* diisi dengan nama berkas yang dipilih oleh pemakai (terdapat pada properti *FileName* milik kotak dialog). Selanjutnya, metode *Open* milik *MediaPlayer* akan membuka berkas multimedia bersangkutan dan membuat tombol-tombol pada *MediaPlayer* dalam keadaan dapat diklik.

Sekarang bisa segera menguji program ini. Jalankan program dan pilih menu Berkas | Buka..... Saat muncul kotak dialog, pilihlah sebuah berkas AVI (Delphi menyediakan dua berkas AVI pada folder Program Files\Borland\ Delphi5\Demos\ Coolstuf. Pilih speedis dan klik tombol Open. Beberapa saat kemudian akan terlihat bahwa tombol-tombol yang semula dalam keadaan mati, sekarang dalam keadaan bisa diklik, Klik tombol "Play" untuk menjalankannya.

Salah satu gambar dari video tersebut terlihat di bawah ini.



Gambar 2.6. Jendela yang berisi video AVI

Terlihat bahwa membuat program untuk memainkan AVI sangatlah mudah. perlu pula Anda praktikkan pada Program ini untuk membuka berkas-berkas berakhiran .MID, .WAV, serta .DAT.

• Meletakkan Video pada Form

Pada program yang telah dibuat, video dimainkan pada jendela tersendiri. Video ini sebenamya dapat diatur agar diletakkan pada form (bisa pada form itu sendiri atau pada komponen lain seperti Panel).

Untuk mempraktikkan hal ini, bukalah program Media.dpr dan kemudian tambahkan komponen panel pada form tersebut, sehingga tampilannya menjadi seperti berikut:



Gambar 2.7. Komponen panel ditambahkan ke form

Selanjutnya lakukan perubahan properti-properti pada *Panel* sehingga menjadi seperti berikut:

Tabel 2.5. Beberapa Properti pada Panel

Properti	Isi
Caption	(Dikosongkan)
Name	PanelMedia

Setelah itu perlu menambahkan kode pada item menu **Buka...** Langkah yang perlu dilakukan adalah:

- 1. Pilih menu View | Toggle Form/Unit.
- 2. Tuliskan kode berikut di bawah baris MediaPlayer1.Open;:

```
MediaPlayer1.Display := PanelMedia;
with PanelMedia do
```

MediaPlayerl.DisplayRect := Rect(0, 0, Width, Height);

Pernyataan pertama digunakan untuk menentukan komponen yang digunakan untuk menampilkan media. Dalam hal ini media akan diletakkan ke *PanelMedia*. Pernyataan berikutnya digunakan untuk menentukan ukuran tampilan media (ditentukan via properti **DisplayRect**). **Rect** digunakan untuk menentukan koordinat pojok kiri atas dan kanan bawah kotak untuk menampilkan media. Pemanggilan:

Rect(0, 0, Width, Height);

berada bahwa pojok kiri atasnya adalah (0,0) dan pojok kanan-bawahnya adalah (PanelMedia.Width, PanelMedia.Height). Dalam hal ini *PanelMedia* tidak Perlu dituliskan mengingat terdapat pernyataan with *PanelMedia*.

Tampilan form saat dijalankan dan video dimainkan akan menjadi seperti berikut:



Gambar 2.8. Video diletakkan pada form

• Menyembunyikan Tombol MediaPlayer

Bila tidak menginginkan kehadiran tombol-tombol yang disediakan oleh *MediaPlayer*, Anda bisa menyembunyikannya dengan memberikan nilai **False** pada properti **Visible** milik *MediaPlayer*. Sebagai pengganti tombol tersebut, dapat dibuat tombol tersendiri. Sebagai contoh diperlihatkan pada gambar 2.9.

₩ Program MediaPlayer Berkas	
	Main Berhenti

Gambar 2.9. Tombol-tombol MediaPlayer disembunyikan

Langkah yang perlu dilakukan untuk membentuk form di atas:

- 1. Bukalah program Media.dpr.
- 2. Klik pada komponen MediaPlayer yang terdapat pada form.
- 3. Melalui Object Inspector, ubahlah isi properti Visible menjadi False.
- 4. Buatlah dua buah tombol dan atur properti-propertinya seperti pada Tabel 2.6.

Tabel 2.6. Ta	abel Properti	Button Main	dan Berhenti
---------------	---------------	--------------------	--------------

Komponen	Properti	Isi
Button teratas	Caption	&Main
	Name	ButtonMain
Button terbawah	Caption	&Berhenti
	Name	ButtonBerhenti

5. Tuliskan kode (yang tidak dicetak tebal) berikut sehingga kode lengkap untuk kejadian OnClick milik tombol Main menjadi seperti berikut:

```
Procedure TformMedia.ButtonMainClick(Sender:Tobject);
    begin
        MediaPlayer1.Play;
    end;
```

 Tuliskan kode (yang tidak dicetak tebal) berikut sehingga kode lengkap untuk kejadian OnClick milik tombol Berhenti menjadi seperti berikut:

```
Procedure TformMedia.ButtonBerhentiClick(Sender:Tobject);
    begin
        MediaPlayer1.stop;
    end;
```

Kode yang anda tuliskan sangat sederhana. *Play* untuk memainkan dan *Stop* untuk menghentikan. Dengan menuliskan kode semacam di atas, dapat diatur sendiri tombol-tombol yang digunakan untuk mengendalikan media.

• Menambahkan Status Main

Untuk mengetahui persentase status main dari media yang sedang dimainkan, monitorlah properti *Position* dan membandingkannya dengan isi properti **Length**. Untuk mempraktikkan hal ini perlu ditambahkan sebuah *Timer* dan sebuah komponen *Gauge* (ada pada halaman *Samples* pada *Component Palette* sebagaimana terlihat dibawah *Panel*.



Gambar 2.10. FormMedia dilengkapi dengan Timer dan Gauge

Kemudian atur properti-properti milik kedua komponen tersebut sehingga, menjadi seperti berikut:

Komponen	Properti	Isi
Timer	Enabled	False
	Interval	100
	Name	TimerMedia
Gauge	Name	GaugeMedia

Tabel 2.7. Tabel Komponen Timer dan Gauge

Langkah selanjutnya, tambahkan kode berikut:

```
GaugeMedia.MinValue := 0;
GaugeMedia.MaxValue := MediaPlayer1.Length;
GaugeMedia.Progress := MediaPlayer1.Position;
```

di bawah:

with PanelMedia do

```
MediaPlayerl.DisplayRect := Rect(2, 2, WIdth-4, Height-4);
Kode diatas digunakan untuk mengisikan keadaan awal komponen Gauge.
```

Length untuk memperoleh panjang media, **Position** menyatakan posisi sekarang dari media. Perlu diketahui, pada komponen *Gauge*:

- MinValue menyatakan nilai terendah yang diwakili gauge.
- MaxValue menyatakan nilai tertinggi yang diwakili gauge.
- Progress menyatakan nilai sekarang pada gauge.

Lalu tuliskan kode berikut pada kejadian **OnTimer** milik *Timer*:

GaugeMedia.Progress := MediaPlayer1.Position;

Selanjutnya, ubahlah kode pada kejadian OnClick milik tombol Main sehingga kode selengkapnya adalah seperti berikut (yang dicetak tebal adalah bagian yang telah ada sebelumnya):

Kode diatas digunakan untuk memeriksa nilai properti DeviceID. Properti ini tidak berisi nol kalau ada media yang telah dibuka. Pada kode di atas, kalau properti ini berisi nol maka akan dimunculkan suatu pesan kemudian prosedur diakhiri. Bila tidak bernilai nol, maka pernyataan:

TimerMedia.Enabled := True;

akan dijalankan. Tujuannya adalah mengaktifkan *timer* sehingga posisi sekarang media dapat diketahui (pada Gauge).

Adapun kode yang perlu Anda tambahkan pada kejadian *OnClick* milik tombol "Berhenti" adalah sebagai berikut:

Yang diletakkan untuk mematikan Timer.

2.3.2. Database(Microsoft Access 2000)

Database dipakai untuk setiap aplikasi sederhana sampai aplikasi rumit yang melibatkan beberapa *user*. Oleh sebab itu *database* dibagi-bagi sesuai dengan kompleksitasnya. Ada tiga model *database*, yaitu:

• Database yang berdiri sendiri (stand-alone)

Database stand-alone adalah database yang sangat sederhana karena disimpan di sistem *file* lokal dan mesin database mengakses pada mesin yang sama. Disainer database tidak perlu kuatir akan terjadi konkurensi karena database hanya dipakai untuk satu aplikasi.

• Database terbagi (File Share)

Database ini hampir sama dengan database stand-alone tetapi diakses oleh beberapa user. Misalnya sebuah database karyawan dipakai oleh beberapa bagian. Jika satu bagian mengubah data, bagian lain juga dapat merasakan perubahan tersebut. Database ini akan mengalami masalah jika aplikasi memerlukan banyak perhitungan dan pada saat yang bersamaan ada pengaksesan ke dalam database.

• Database Client/Server

Database ini merupakan database level tertinggi. Database ini memerlukan sebuah mesin khusus(server) yang dipakai untuk melayani sekelompok user. Misalnya database karyawan diimplementasikan dengan database client/server. Seorang user akan menampilkan semua karyawan yang gajinya lebih dari satu juta rupiah. Pada database terbagi, sistem akan terhenti pada saat proses ini dikerjakan. Pada database client/server, komputer akan menanyakan apakah user akan menunggu hasilnya atau akan mengerjakan pekerjaan lain sambil proses tersebut dikerjakan.

Aplikasi *Delphi* dirancang untuk kompatibel dengan setiap *database*. *Delphi* dapat memakai tabel yang sudah dibuat oleh beberapa program aplikasi misalnya: *Access, Paradox, dBase, InterBase, SyBase, Oracle*, dan sebagainya. Untuk hal-hal selanjutnya akan dipakai *Paradox* atau *dBase*. Alasannya karena sudah terintegrasi di dalam *Delphi*. Tabel 2.8. berikut menunjukkan perbandingan kemampuan antara tabel *Paradox* dan tabel *dBase*.

Attribut	Paradox	DBase
Record maksimum	2 milyar	1 milyar
Banyaknya field per tabel	255	1024
Byte per record	32.750	32.767

Tabel 2.8. Tabel Perbandingan antara tabel Paradox dan dBase

Dengan kemampuan seperti di atas, kedua *database* dapat dipakai untuk aplikasi pada umumnya. *Paradox* memberikan fleksibilitas lebih tinggi dibanding *dBase*. *Paradox* mengizinkan validasi data pada *level database*. Sekuritas *password* juga dapat diintegrasikan ke dalam *Paradox*.

2.3.3. Adobe Photoshop 5.0

Adobe Photoshop adalah salah satu software desain grafis yang sangat populer dikalangan para desainer. Kemampuan dan keunggulannya sudah tidak perlu disangsikan dan telah diakui banyak kalangan. Berikut ini adalah beberapa penjelasan mengenai dasar-dasar pemakaian software ini.

2.3.3.1. Palet

Palet dalam Photoshop digunakan untuk mengontrol sifat dan cara kerja dari peranti(*tool*) yang ada di dalam Toolbox. Pilihan **Window** pada menu menunjukkan jenis-jenis palet yang ada. Jika memilih salah satu palet melalui menu ini, palet yang dipilih akan muncul sebagai window yang mengambang(*floating window*) dalam ruang kerja(*workspace*). Lihat Gambar 2.11.

2.3.3.2. Bekerja dengan Palet

Berikut ini adalah beberapa hal yang akan sering dilakukan dalam bekerja dengan palet dalam Photoshop.

Menampilkan dan Menyembunyikan Palet: Untuk menampilkan palet, gunakan pilihan Window pada menu dan pilih palet yang dikehendaki. Nama palet yang dapat ditampilkan diawali dengan Show. Sebagai contoh, dapat menggunakan pilihan menu Window > Show Layers untuk menampilkan Layers Palette. Untuk menyembunyikan palet, gunakan pilihan Window pada menu dan pilih palet yang ingin disembunyikan. Nama palet yang dapat disembunyikan diawali dengan Hide.



Gambar 2.11. Menu utama Adobe Photoshop versi 5.5

Memilih Palet: Photoshop biasanya mengelompokkan beberapa palet dalam satu window. Sebagai contoh, jika memilih **Window > Show Layers**, Photoshop menampilkan Layer Palette, Chennels Palette, dan Paths Palette dalam satu window. Tab yang ada di belakang palet menunjukkan nama-nama palet yang ada dalam kelompok tersebut. Untuk memilih salah satu palet yang ada dalam satu kelompok, klik tab nama palet tersebut.





Memindahkan Palet: Untuk memindahkan sebuah palet, klik bagian atas window kemudian geser ke lokasi yang diinginkan.

Mengubah Ukuran Palet: Ukuran palet dapat diperkecil dengan mengklik tombol *minimize* yang terdapat di sudut kanan atas dari window yang memuat palet tersebut. Untuk mengembalikannya ke ukuran semula dapat diklik tombol *maximize* yang terdapat di sudut yang sama.

Mengakses Menu Palet: Setiap palet memiliki sejumlah menu. Pengaksesan menu suatu palet dapat dengan mengklik segitiga yang terdapat di sebelah kanan atas dari window yang memuat palet tersebut, dan dapat juga diakses melalui menu utama.

2.3.3.3. Jenis-jenis Palet

Layers Pallete: Layers Palette menunjukan semua layer yang ada dalam sebuah citra. Layer yang baru dibuat ditempatkan di atas layer yang telah ada sebelumnya. Dengan menggunakan Layer Palette, dapat diubah ukuran layer, menghapus, membuat yang baru atau duplikat dari sebuah layer, menampilkan atau menyembunyikan sebuah layer, dan sebagainya. Lihat gambar 2.13.

Brushes Palette: Palette ini menunjukan ukuran dan bentuk dari peranti gambar seperti pen, pencil, paintbrush, airbrush, dan eraser. Dengan mengakses Palette Menu, dapat membuat, menghapus, atau mengubah ukuran dan bentuk dari kuas yang ada. Lihat gambar 2.14.



Gambar 2.13. Penjelasan Panel Layer



Gambar 2.14. Penjelasan Panel Brush

Option Palette: Option Palette menunjukkan opsi yang tersedia untuk peranti yang sedang digunakan. Palet ini mengontrol sifat dan kelakuan peranti tersebut. Jika memilih sebuah peranti dari Toolbox, Option Palette akan menunjukkan semua opsi yang tersedia untuk peranti tersebut.



Gambar 2.15. Panel Linear Gradient Option

Channels Palette: Channel berfungsi untuk menyimpan informasi mengenai warna. Dengan menggunakan channel, dapat disimpan dan dimuat sebuah seleksi. Channels Palette memungkinkan untuk membuat dan mengatur channel-channel yang ada.



Gambar 2.16. Panel Channels

Swatches Palette: Swatches Palette menunjukkan table warna atau *Color Look Up Table (CLUT)* untuk file citra yang dibuka.

Color Palette: Color Palette menunjukkan nilai dari warna foreground dan background. Pengubahan warna foreground dan background dapat dilakukan dengan mengubah nilai warna atau dengan mengklik sebuah warna dari *color bar* yang terdapat di bagian Color Palette.



Gambar 2.17. Panel Color

Navigator Palette: Navigator Palette berisi thumbnail dari file citra yang dibuka dan bagian dari citra yang sedang dikerjakan. Dengan menggunakan Navigator Palette, dapat dengan mudah me-zoom ke dalam atau ke luar dengan menggunakan slider yang terdapat di bagian bawah window Navigator Palette.



Gambar 2.18. Panel Navigator

History Palette: History Palette adalah salah satu fasilitas baru yang terdapat pada Photoshop. Palet ini menyimpan status perubahan yang dilakukan terhadap sebuah citra. Dengan History Palette, dapat dilakukan *undo* terhadap perubahan yang dilakukan.



Gambar 2.19. Panel History

Action Palette: Photoshop dapat mengotomatisasi perkerjaan dengan mengelompokan perintah-perintah ke dalam sebuah aksi. Actions Palette berguna untuk merekam dan memainkan aksi. Melalui Action Palette dapat juga menyunting suatu aksi.

2.3.3.4.Toolbox

Toolbox berisi berbagai peranti yang dapat digunakan untuk memanupulasi dan menyunting sebuah citra. Penggunaan peranti yang ada dalam toolbox untuk melakukan seleksi, membuat teks, atau membuat gambar. Untuk menampilkan Toolbox, gunakan pilihan **Window > Show Tools**. Toolbox tampil sebagai window mengambang di dalam ruang kerja.



Perlu diperhatikan bahwa beberapa dari peranti yang terlihat dalam Toolbox mempunyai segitiga kecil di sudut kanan bawah. Peranti ini memiliki peranti-peranti lain yang tersembunyi di belakangnya. Untuk mengakses peranti-peranti yang tersembunyi ini harus tetap dengan menekan mouse pada saat mengklik peranti yang terlihat di dalam Toolbox.

Berikut ini adalah peranti-peranti dari Toolbox yang dapat digunakan.

[]]	\bigcirc	e=3		4
-----	------------	-----	--	----------

Rectangular Marquee Elliptical Marquee Single Row Marquee

Single Column Marquee

PPP

Lasso Polygon Lasso Magnetic Lasso



Pencil Line

Crop



Туре





Dodge

Type Mask Vertical Type

Vertical Type Mask



Linear Gradient Radial Gradient Angle Gradient Diamond Gradient Reflected Gradient Burn Sponge



Eyedropper Color Sampler



Pen Tool Magnetic Pen Freeform Pen Add Anchor Point Delete Anchor Point Direct Selection Convert Anchor Point Rubber Stamp Pattern Stamp

2

Å

• Peranti untuk Membuat Seleksi



Rectangular Marquee

Rectangular Marquee digunakan untuk membuat seleksi yang berbentuk segi empat.

 \bigcirc

Elliptical Marquee

Elliptical Marquee digunakan untuk membuat seleksi yang berbentuk elips. Peranti ini juga dapat digunakan untuk membuat lingkaran, yaiktu dengan menekan tombol SHIFT

sambil menggeser mouse.

5 23	Single Row Marquee	Peranti ini digunakan untuk membuat seleksi horizontal setinggi satu pixel.
	Single Column Marquee	Peranti ini digunakan untuk membuat seleksi vertical selebar satu pixel.
¥	Сгор	Crop berguna untuk membuang area pada citra yang tidak diinginkan. Perhatikan bahwa peranti ini mempengaruhi semua layer yang ada baik aktif
R	Lasso	maupun tidak. Lasso digunakan untuk membuat seleksi dengan menggambar bebas (<i>freehand</i>). Jika menggunakan peranti ini, pada saat mouse dilepaskan
		titik awal dan akhir akan terhubungkan. Gunakan anti- alias pada Option Palette untuk menghaluskan seleksi.
$\left\langle \right\rangle$	Polygon Lasso	Polygon Lasso digunakan untuk membuat seleksi berupa poligon atau segi-banyak yang berupa garis patah-patah. Jika menggunakan peranti ini untuk membuat seleksi poligon,

akhirilah pada titik yang sama dengan titik waktu memulai. Gunakan anti-alias pada Option Palette unutk menghaluskan seleksi.

Magnetic Lasso berguna untuk membuat seleksi yang melekat pada perbatasan dari pixel yang dipilih. Gunakan anti-alias pada Option Palette untuk menghaluskan seleksi.

> Peranti ini membuat seleksi atas pixel-pixel yang ada di dalam kawasan warna tertentu. Toleransi dapat ditetapkan melalui Option Palette. Gunakan anti-alias pada Option Palette untuk menghaluskan seleksi.

Paintbrush digunakan untuk membuat guresan warna dengan kuas. Warna yang digunakan adalah warna foreground dan kuas yang digunakan adalah kuas yang dipilih dari Brushes Palette.

Airbrush digunakan untuk

 \wp

Magnetic Lasso

*

Magic Wand

Peranti untuk Menggambar



Paintbrush

Airbrush

dipilih da

L

membuat goresan dengan pinggiran yang halus. Sama seperti Paintbrush, warna yang digunakan adalah warna foreground dan kuas yang digunakan adalah kuas yang dipilih dari Brushes Palette.



Paint Bucket



Pencil



Line

Eraser

Ø

digunakan adalah kuas yang dipilih dari Brushes Palette. Paint Bucket mengisi seleksi

dengan warna foreground.

Pencil digunakan untuk membuat goresan secara bebas. Warna yang digunakan adalah warna foreground dan kuas yang digunakan adalah kuas yang dipilih dari Brushes Palette.

Line digunakan untuk membuat garis. Peranti ini dapat juga digunakan membuat garis dengan mata panah pada salah satu atau kedua ujungnya. Gunakan anti-alias pada Option Palette untuk membuat garis yang halus.

Eraser digunakan untuk menghapus area pada layer yang aktif. Jika *Layer Transparency* pada layer

ditandai, akan eraser menghapus dengan menggunakan warna foreground. Dapat dipilih salah dari jenis satu penghapus Pencil, Airbrush, atau Paintbrush Option melalui Palette.

• Peranti untuk Membuat Teks



Type Mask

Type

Peranti untuk membuat teks.

Peranti ini digunakan untuk membuat *masker type* yaitu seleksi berupa teks. Peranti ini dapat juga digunakan untuk membuat outline teks. Peranti bahwa berbeda dengan peranti Type, Type Mask tidak menempatkan teks pada layer tersendiri

Vertical Type digunakan untuk membuat teks pada sumbu vertikal.

l™ ∭

 $|\mathbf{T}|$

Vertical Type Mask

Vertical Type

Vertical Type Mask digunakan untuk membuat masker type pada sumbu vertikal. • Peranti lainnya



setelah melalui pembesaran.

2.3.3.5. Layer Type

Layer Type ditunjukkan oleh huruf "T" pada Layers Palette. Layer Type dapat diubah seperti halnya layer biasa, dapat pula menyunting teks, mengubah jenis dan ukuran font, dan mengubah warna foreground teks yang terdapat pada Layer Type.

Sebelum dapat menggunakan filter pada Layer Type, Layer Type harus terlebih dahulu diubah menjadi layer biasa. Untuk mengubah layer Type menjadi layer biasa, dapat digunakan perintah **Layer** > **Type** > **Render Layer**. Perlu diingat bahwa setelah menjadi layer biasa, penyuntingan atau pengubahan jenis dan ukuran font dari teks yang terdapat pada Layer Type tidak akan dapat dilakukan lagi.