2. TEORI PENUNJANG

2.1. File WAV

File WAV adalah file audio standar yang digunakan oleh Windows. Suara yang berupa digital audio dalam file WAV disimpan dalam bentuk gelombang, karena itulah file ini memiliki ekstensi .WAV (Wave). File WAV ini dapat dibuat dengan menggunakan berbagai program wave editor maupun wave recorder. Contoh wave recorder adalah Sound Recorder milik Windows atau Sound o'LE milik Soundblaster. Sedangkan contoh wave editor adalah Goldwave, Coolwave, dan lainnya.

Data digital audio dalam file WAV bisa memiliki kualitas yang bermacam-macam. Kualitas dari suara yang dihasilkan ditentukan dari *bitrate*, *samplerate*, dan jumlah *channel*.

2.1.1. Bitrate

Bitrate merupakan ukuran bit tiap samplenya, yaitu 8-bits, 16-bits, 24 bits atau 32 bits. Dalam 8-bits WAV semua samplenya hanya akan memakan sebanyak 1 byte saja. Sedangkan untuk 16-bits akan memakan 2 bytes. WAV 16bits tiap samplenya memiliki nilai antara -32768 sampai 32767, dibandingkan untuk 8-bits yang hanya memiliki nilai antara -128 sampai 127 maka 16-bits WAV file menghasilkan suara yang lebih baik karena datanya lebih akurat daripada 8-bits. Namun, pada kenyataannya telinga manusia umumnya tidak mampu membedakan suara yang dihasilkan.

2.1.2. Samplerate

Samplerate menyatakan banyaknya jumlah sample yang dimainkan setiap detiknya. Samplerate yang umum dipakai adalah 8000Hz, 11025Hz, 22050Hz, dan 44100Hz.

Untuk 8000Hz biasanya disebut sebagai *Telephone quality* karena suara yang dihasilkan menyerupai suara dari telepone. Samplerate 11025Hz sering

digunakan untuk merekam suara manusia karena dinilai sudah cukup. Tetapi untuk music agar dapat memperoleh hasil yang baik maka digunakan samplerate 22050Hz. Sedangkan 44100Hz merupakan *CD quality*, samplerate 44100Hz inilah yang digunakan dalam audio CD karena bagus untuk semua jenis suara.

Berbeda dengan bitrate, perbedaan samplerate dapat dikenali dengan mudah, karena tentu saja suara dari radio dengan suara CD player sangat jauh berbeda.

2.1.3. Jumlah channel

Menentukan suara yang dihasilkan apakah mono atau stereo. Mono memiliki hanya 1 channel, sedangkan stereo 2 channel dan memakan tempat 2 kali lebih banyak daripada mono. Untuk merekam suara manusia, mono sudah cukup memberikan kualitas yang baik, sedangkan *CD quality* menggunakan 2 channel.

Format WAV ini bisa dibilang sangat populer karena hampir setiap program yang bisa membuka dan menyimpan file audio mengenali file WAV. Tetapi dari kepopulerannya itu muncul berbagai masalah pada struktur file WAV, karena beberapa program melakukan kesalahan dalam menyimpan file WAV, dan bila program tersebut telah populer, maka program-program selanjutnya terpaksa harus membuat kode yang bisa membaca file WAV yang salah tersebut. Untungnya beberapa kesalahan yang ada bisa diperbaiki dengan mengkalkulasi ulang nilai-nilai yang ada pada format sample file WAV.

2.2. Struktur File WAV

File WAV menggunakan struktur standar RIFF dengan mengelompokan isi file ke dalam bagian-bagian seperti format WAV dan data digital audio. Setiap bagian memiliki headernya sendiri-sendiri beserta dengan ukurannya.

2.2.1. RIFF

Struktur RIFF (Resource Interchange File Format) ini merupakan struktur yang biasa digunakan untuk data multimedia dalam Windows. Struktur

ini mengatur data dalam file ke dalam bagian-bagian yang masing-masing memiliki header dan ukurannya sendiri dan disebut sebagai *chunk*. Struktur ini memungkinkan bagi program bila tidak mengenali bagian tertentu untuk melompati bagian tersebut dan terus memproses bagian yang dikenal. Data dari suatu bagian bisa memiliki sub-bagian dan seluruh data dalam file berstruktur RIFF selalu merupakan sub-bagian dari suatu bagian yang memiliki header "RIFF". Contoh file yang menggunakan struktur RIFF adalah file WAV dan AVI.



Gambar 2.1 Struktur file RIFF

2.2.2. Little Endian vs Big Endian

Little Endian adalah cara yang dipakai untuk menyimpan data yang berukuran lebih dari 1 byte, yaitu dengan cara disimpan secara terbalik. Contoh: misalnya 2 byte data bernilai 7321 memiliki nilai hexadesimal 1C 99. Dengan cara Little Endian maka data ini akan disimpan sebagai 99 1C. Secara umum PC menggunakan metode ini. Sedangkan kebalikan dari Little Endian adalah Big Endian, dalam Big Endian maka data contoh tadi akan disimpan sebagai 1C 99. Big Endian dipakai oleh Macintosh. Penyimpanan data dalam file WAV juga memakai metode Little Endian.

2.2.3. PCM

File WAV sendiri ada berbagai macamnya dan diklasifikasikan berdasarkan tipe kompresinya, seperti PCM, Microsoft ADPCM, IMA ADPCM, dan lainnya. PCM sendiri merupakan tipe yang tidak terkompresi.

2.2.4. File WAV

Sesuai dengan struktur file RIFF, file WAV diawali dengan 4 byte yang berisi 'RIFF' lalu diikuti oleh 4 byte yang menyatakan ukuran dari file tersebut dan 4 byte lagi yang berisi 'WAVE' yang menyatakan bahwa file tersebut adalah file WAV. Berikutnya adalah informasi dari format sample yang menjadi subbagian dari bagian RIFF lalu diikuti sub-bagian data audionya.

Rii Uk	FF uran File - 8	
W.	RVE	
	Sample Format Header Ukuran Sample Format	
	Sample Format Info	
	Data Header	
	Ukuran Data Audio	
	Digital Audio Sample	

Gambar 2.2 Struktur file WAV

Bagian RIFF merupakan bagian utama dari semua file yang memakai format RIFF.

Tabel	2.1	Detail	Bagian	RIFF
-------	-----	--------	--------	------

Offset	Size	Description	Value	
0x00	4	Chunk ID	"RIFF" (0x52494646)	
0x04	4	Chunk Data Size	(file size) - 8	
0x08	4	RIFF Type	"WAVE" (0x57415645)	
0x0c	Wave chunks			

Isi dari 4 byte pertama adalah 'RIFF', 4 byte berikutnya adalah ukuran dari bagian RIFF yang nilainya sama dengan ukuran dari file dikurangi 8.

'WAVE' menempati 4 byte berikutnya dan digunakan sebagai penentu jenis dari file tersebut, dalam hal ini adalah file WAV. Setelah itu barulah informasi format dan data dari file WAV disimpan.

Bagian format sample berisi informasi-informasi mengenai bagaimana data disimpan dan memainkannya. Bagian ini dimulai dengan ID 'fmt ', lalu diikuti dengan 4 byte yang merupakan panjang dari informasi dan bernilai 16 untuk PCM.

Offset	Size	Description	Value
0x00	4	Chunk ID	"fmt " (0x666D7420)
0x04	4	Chunk Data Size	16 + extra format bytes
0x08	2	Compression code	1 - 65,535
0x0a	2	Number of channels	1 - 65,535
0x0c	4	Sample rate	1 - OxFFFFFFFF
0x10	4	Average bytes per second	1 - OxFFFFFFFF
0x14	2	Block align	1 - 65,535
0x16	2	Significant bits per sample	2 - 65,535

Tabel 2.2 Detail Bagian Sample Format

Code	Description
0 (0×0000)	Unknown
1 (0x0001)	PCM/uncompressed
2 (0x0002)	Microsoft ADPCM
6 (0×0006)	ITU G.711 a-law
7 (0×0007)	ITU G.711 μ-law
17 (0x0011)	IMA ADPCM
20 (0×0016)	ITU G.723 ADPCM (Yamaha)
49 (0x0031)	GSM 6.10
64 (0×0040)	ITU G.721 ADPCM
80 (0×0050)	MPEG
65,536 (0xFFFF)	Experimental

Tabel 2.3 Beberapa kode kompresi

Kompresi kode menempati 2 byte berikutnya dengan nilai 1 untuk PCM. Gambar 2.5 menunjukan beberapa kompresi yang ada pada file WAV. Dua byte berikutnya menyatakan jumlah channel dari file WAV, lalu 4 byte menyatakan sample rate dan 4 byte lagi menyatakan rata-rata byte tiap detiknya. Dua byte setelahnya merupakan *Block Align* yang menyatakan ukuran data untuk satu sample penuh dalam byte. Yang dimaksud dengan satu sample penuh adalah satu sample yang mewakili nilai dari sample pada semua channel pada suatu waktu. Dua byte terakhir dari bagian sample format ini menyatakan *bitrate* dari data yang disimpan, bernilai 8, 16, 24 atau 32.

Beberapa hal yang perlu diperhatikan:

- Nilai rata-rata byte tiap detik bisa dihitung dengan rumus:

AvgBytesPerSec = SampleRate x BlockAlign

- Nilai Block Align dapat dihitung dengan rumus:
 BlockAlign = Significantbitpersample / 8 x NumofChannel
- Untuk WAV 8-bit, maka data disimpan dalam bentuk *unsigned* (tak bertanda) sedangkan untuk 16-bit keatas disimpan dalam bentuk *signed*.

Tabel 2.4 Detail Bagian Data Audio

Offset	Length	Туре	Description	Value
0x00	4	char[4]	chunk ID	"data" (0x64617461)
0x04	4	dword	chunk size	depends on sample length and compression
0x08				sample data

Bagian berikutnya adalah bagian data audio. Di bagian inilah sample digital audio disimpan. Bagian ini dimulai dengan ID 'data' dan diikuti dengan 4 byte yang menyatakan besarnya data dalam byte, lalu selebihnya adalah data digital audio-nya. Untuk WAV dengan jumlah channel lebih dari satu, maka data disimpan secara berselingan. Misalnya untuk WAV stereo maka data disimpan seperti: channel1, channel2, channel1, channel2, channel1, ... dst. Gambar 2.7 menunjukan contoh dari data pada file WAV.

Time	Channel	Value
0	1 (left)	0x0053
	2 (right)	0x0024
1	1 (left)	0x0057
	2 (right)	0x0029
2	1 (left)	0x0063
	2 (right)	0x003C

Tabel 2.5 Contoh data pada WAV stereo

Berikut adalah contoh dari 72 byte awal dari suatu file WAV beserta dengan penjelasan setiap byte-nya.

52 49 46 46 24 08 00 00 57 41 56 45 66 6d 74 20 10 00 00 00 01 00 02 00 22 56 00 00 88 58 01 00 04 00 10 00 64 61 74 61 00 08 00 00 00 00 00 00 24 17 1e f3 3c 13 3c 14 16 f9 18 f9 34 e7 23 a6 3c f2 24 f2 11 ce 1a 0d



Gambar 2.3 Contoh 72 byte awal dari file WAV

2.3. Manipulasi File WAV

Dengan mengetahui struktur dari file WAV maka manipulasi file WAV dapat dilakukan dengan cukup mudah. Yang dimaksud dengan manipulasi file WAV disini adalah suatu proses yang dilakukan untuk mengubah hasil suara yang dari file WAV dengan cara mengubah isi dari file WAV tersebut baik pada bagian sample format maupun pada bagian data. Ada 3 jenis manipulasi yang akan dibahas disini yaitu: manipulasi Volume, Frekuensi dan Channel. Manipulasi-manipulasi ini diambil dari web JWCS Digital Audio Programming yaitu di <u>http://thunder.prohosting.com/~jwcs/dap/algor/index.html</u>. Sedangkan contoh diambil dengan menggunakan program GoldWave, yaitu program untuk melakukan manipulasi WAV.

2.3.1. Manipulasi Volume

Manipulasi volume dilakukan untuk mengubah kerasnya suara yang dihasilkan oleh file WAV. Perlu diketahui bahwa data pada file WAV sebenarnya merupakan amplitude dari gelombang suara pada suatu waktu. Untuk mengubah atau memanipulasi volume maka yang perlu dilakukan adalah mengubang amplitude dari gelombang suara tersebut. Untuk itu maka manipulasi ini dapat dilakukan dengan mudah, misalnya sample pada suatu file WAV 16-bit yaitu: 1000, 2000, 3000, 4000, 2500. Untuk mendapatkan hasil suara yang diperbesar 2 kali, maka nilai dari masing-masing sample tersebut dikalikan dengan 2, sehingga menjadi: 2000, 4000, 6000, 8000, 5000.

Applikasi dari manipulasi volume ini yaitu contohnya adalah pembuatan efek *fadein* dan *fadeout*. Yang dimaksud dengan *fadein* adalah pembuatan efek dimana intensitas suara yang dihasilkan semakin lama semakin meninggi. Sedangkan *fadeout* merupakan kebalikan dari *fadein* yaitu intensitas suara yang semakin lama semakin rendah. Atau dengan kata lain *fadein* bisa digambarkan dengan pendengar berjalan mendekati sumber suara sedangkan *fadeout* pendengar menjauhi sumber suara.

2.3.1.1. Mengubah Volume

Untuk mengubah volume yang dilakukan adalah proses perkalian dan tidak akan pernah ada proses penjumlahan ataupun pengurangan. Karena misalnya bila dilakukan proses pengurangan, maka akan ada kemungkinan amplitudo melewati nol sehingga akan mengacaukan bentuk gelombangnya. Sedangkan alasan untuk pemakaian perkalian daripada pembagian adalah untuk menghindari pembagian dengan nol. Jadi untuk mengubah volume, maka setiap sample akan dikalikan dengan sebuah angka. Berikut adalah gambar contoh gelombang suara sumber dengan yang telah diperbesar 2 kali.



Gambar 2.4 Contoh gelombang suara diperbesar 2 kali

2.3.1.2. Fadein dan Fadeout

Fadein dan Fadeout merupakan applikasi dari mengubah volume dan dapat dengan mudah dilakukan hanya dengan mengubah angka pengali untuk setiap sample. Untuk Fadein maka angka pengali tersebut bernilai dari nol sampai satu sedangkan untuk Fadeout maka angka pengali tersebut bernilai dari satu sampai nol.



Gambar 2.5 Contoh gelombang suara diberi efek fadein



Gambar 2.6 Contoh gelombang suara diberi efek fadeout

2.3.1.3. Panning

Panning adalah pengaturan seberapa besar suara pada speaker kiri dibanding dengan speaker kanan. Panning ini hanya bisa dilakukan untuk WAV stereo saja. Untuk Panning dari kiri ke kanan, maka channel kiri dilakukan fadeout sedangkan kanan fadein dan sebaliknya. Hasilnya seolah-olah suara bergerak dari kiri ke kanan untuk panning kiri ke kanan dan dari kanan ke kiri untuk panning kanan ke kiri.

2.3.2. Manipulasi Frekuensi

Frekuensi dalam file WAV menyatakan jumlah sample yang dimainkan setiap detiknya. Frekuensi juga menentukan kualitas dari suara yang dihasilkan. Manipulasi Frekuensi dilakukan dengan mengubah nilai sample rate yang terdapat pada bagian sample format pada file WAV. Perlu diperhatikan bahwa jika mengubah nilai sample rate, maka nilai rata-rata byte per detik juga perlu dikalkulasi ulang dengan mengalikan samplerate dan blockalign.

Jika manipulasi hanya dilakukan pada bagian sample format saja, maka suara yang dihasilkan oleh file WAV tersebut akan menjadi terlalu cepat atau terlalu pelan. Hal ini dikarenakan jika menaikkan frekuensi maka jumlah sample yang dimainkan setiap detiknya bertambah, sehingga waktu memainkan file WAV pun menjadi lebih pendek. Untuk mengatasi hal ini, maka jumlah sample yang terdapat dalam bagian data harus disesuaikan jumlahnya.

Jika melakukan penurunan frekuensi, maka jumlah sample tiap detiknya juga harus dikurangi, dan jika menaikkan frekuensi, maka harus diadakan penambahan sample. Contoh jika mengubah file WAV 22050Hz menjadi 44100Hz, maka sample per detik yang asalnya berjumlah 22050 sample harus ditambah menjadi 44100 sample. Sebaliknya jika mengubah file WAV 22050Hz menjadi 11025Hz maka sample per detiknya harus dibuang separuhnya. Pengubahan jumlah sample ini harus diikuti dengan pengubahan ukuran bagian data disetarakan dengan jumlah sample yang ada, tergantung dari *bitrate* dan jumlah channel dari file WAV.

2.3.3. Manipulasi Channel

Manipulasi channel dilakukan dengan mengubah nilai jumlah channel pada bagian sample format, namun tentunya pengubahan ini perlu diikuti juga dengan penambahan atau pengurangan data. Misalnya untuk mengubah WAV mono (1 channel) menjadi stereo (2 channel) maka pada bagian data perlu ditambahkan sample sebanyak sample yang sudah ada. Sample yang asli akan menjadi channel pertama sedangkan untuk channel kedua diisi dengan kebalikan dari channel pertama.

Lain halnya untuk mengubah stereo menjadi mono, maka sample untuk mono dapat diperoleh dengan menggabungkan channel 1 dan channel 2. Penggabungan kedua channel ini dapat dilakukan dengan mengambil nilai rata-rata dari setiap samplenya. Misalnya suatu WAV stereo memiliki sample: 1000,2000,3000 untuk channel 1 dan 5000,2000,4000 untuk channel 2, maka jika diubah menjadi mono maka nilai setiap samplenya akan menjadi: 3000,2000,3500.

Applikasi lain dari manipulasi channel ini adalah penukaran channel atau yang dikenal dengan istilah *reverse stereo*. Hal semata-mata dilakukan hanya dengan menukar nilai setiap sample pada channel 1 dengan sample pada channel 2 dan sebaliknya.

2.3.3.1. Mengubah Mono ke Stereo

File WAV yang asalnya hanya memiliki satu channel bisa diubah menjadi dua channel. Channel pertama tidak akan diubah-ubah, sedangkan untuk channel kedua diisi dengan membalik fase gelombang pada channel pertama. Dengan melakukan ini maka channel satu maupun dua akan menghasilkan suara yang sama, karena pengubahan mono ke stereo ini tidak akan bisa menghasilkan suara sebaik stereo yang aslinya.



Gambar 2.7 Contoh gelombang suara diubah dari mono ke stereo

2.3.3.2. Mengubah Stereo ke Mono

Mengubah WAV stereo menjadi mono pada intinya bisa dilakukan dengan menggabungkan isi channel satu dan channel dua menjadi satu channel. Untuk menggabungkan dua buah gelombang bisa dilakukan dengan menjumlahkan nilai amplitude dari setiap samplenya.



Gambar 2.8 Contoh gelombang suara diubah dari stereo ke mono

2.3.4. Manipulasi Lain

Manipulasi yang tidak termasuk dalam ketiga manipulasi diatas. Manipulasi ini juga bisa berupa gabungan dari manipulasi yang ada. Yang akan dibahas disini adalah time warp dan efek echo.

2.3.4.1. Time Warp

Time Warp akan mengubah panjang dari file WAV. Yang dimaksud dengan panjang disini adalah panjang dari waktu memainkan file WAV bukan ukuran filenya. Proses time warp mirip dengan mengubah frekuensi, yaitu dilakukan panambahan sample untuk memanjangkan file WAV dan pengurangan sample untuk memendekan file WAV, tetapi informasi sample rate tidak diubah sehingga dengan penambahan jumlah sample maka WAV akan bertambah panjang dan bertambah pendek dengan pengurangan jumlah sample.

2.3.4.2. Efek Echo

Efek echo ini adalah suara yang seolah-olah memantul sehingga akan terdengar lagi setelah jangka waktu tertentu dengan volume yang berbeda.



Gambar 2.9 Ilustrasi efek echo

Cara kerjanya terlihat pada gambar diatas, yaitu setiap sample dari input akan langsung menuju output. Suara yang langsung menuju output adalah suara asli., sedangkan untuk suara echo akan menuju ke kotak delay, dimana dalam kotak tersebut suara akan ditahan selama waktu tertentu baru kemudian dilepaskan. Ketika dilepaskan maka suara echo akan bercampur dengan suara asli, jadi dilakukan pencampuran suara antara suara asli dengan suara echo. Untuk mencampur suara tersebut dapat dilakukan dengan proses penjumlahan. Perlu diingat bahwa suara echo yang dilepas dari delay box bisa memiliki volume yang berbeda dengan suara asli, jadi suara echo ini akan diubah dulu volumenya baru dicampur dengan suara asli.



Gambar 2.10 Contoh gelombang suara diberi efek echo dengan delay 5ms dan volume 50%

2.3.5. Clipping

Yang dimaksud dengan *clipping* adalah hal yang mungkin terjadi pada saat melakukan manipulasi pada file WAV karena sample yang dimanipulasi menghasilkan nilai yang melebihi batas atas atau batas bawah dari nilai yang telah ditentukan. Untuk file WAV 8-bit batas atas dan bawah adalah -128 dan 127, sedangkan untuk file WAV 16-bit batas atas dan bawahnya adalah -32768 dan 32767.

Hal ini bisa terjadi ketika menambahkan nilai sample dengan suatu nilai. Umumnya terjadi pada saat mengubah volume (memperbesar volume), dan bisa juga terjadi ketika mengubah WAV stereo menjadi mono. Untuk masalah memperbesar volume hal ini bisa diobati dengan langsung memberikan nilai batas pada sample-sample yang melebihi batas. Sedangkan untuk penggabungan WAV atau channel dilakukan langkah pencegahan terhadap clipping dengan cara mengambil nilai rata-rata dari dua sample yang digabungkan.

2.4. Aplikasi MDI pada Borland Delphi 7

Aplikasi MDI adalah aplikasi yang memakai satu form sebagai tempat untuk dibukanya form-form lain. Contoh dari Aplikasi MDI (Multi Document Interface) ini adalah MS Word. Aplikasi MDI memungkinkan untuk membuka lebih dari 1 file sekaligus.

Dalam Borland Delphi untuk membuat Aplikasi MDI ini cukuplah mudah. Pertama-tama mulailah dengan projek yang baru, lalu tambahkan mainmenu dan toolbar kedalam form utama dan yang paling penting adalah udah property formstyle dari form menjadi fsMDIForm.

Berikutnya adalah membuat form anak dari form utama. Pilih new form dari menu file lalu ubah property formstyle menjadi fsMDIChild. Untuk mencegah agar form anak tidak dibuat pada awal program dijalankan maka dapat dilakukan melalui menu Project \rightarrow Option pada delphi, lalu masukkan form anak ke bagian available form.

Pada form utama tambahkan menu File, New, dan toolbutton pada toolbar untuk membuka form baru. Pada event Onclick dari button dan menu File \rightarrow New yang baru dibuat ketikkan baris perintah seperti:

```
procedure Tform1.New1Click(Sender: TObject);
var newform: tform2;
begin
    newform:=tform2.create(application);
end;
```

Jangan lupa untuk menambahkan form anak kedalam daftar uses. Ini adalah program MDI Application yang paling sederhana. Untuk berpindah dari satu form anak ke lainnya bisa digunakan perintah Next dan Previous. Untuk mengatur posisi window form anak bisa digunakan perintah Tile dan Cascade seperti pada Windows. Variable ActiveMDIChild akan selalu menunjuk ke form anak yang sedang aktif. Berikut adalah gambar jika program diatas dijalankan.

🌠 Form1		-o×
Eile		
7 Form2		
	7 / Form2	
	-	

Gambar 2.11 Contoh Program MDI Application Sederhana Pada Delphi

Dalam contoh ini, form utama adalah form1 sedangkan form2 adalah form anak yang akan muncul setiap ada file yang dibuka.