

PEMBUATAN PERANGKAT LUNAK WAVE MANIPULATOR UNTUK MEMANIPULASI FILE WAV

Ibnu Gunawan, Kartika Gunadi

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra

e-mail: ibnu@petra.ac.id, kgunadi@petra.ac.id

ABSTRAK: File WAV adalah file audio standar yang digunakan oleh Windows. Manipulasi file WAV bisa dilakukan untuk mendapatkan hasil suara yang berbeda-beda. Proses manipulasi file WAV tersebut akan dikemas menjadi suatu aplikasi yang akan mampu melakukan perubahan-perubahan pada file WAV. WAVE Manipulator adalah nama dari suatu aplikasi audio yang akan dibuat untuk mengubah hasil suara dari file-file audio seperti file WAV. Tujuan dari pembuatan perangkat lunak ini adalah untuk membangun suatu aplikasi yang dapat digunakan untuk mengubah atau memberi efek pada file WAV yang merupakan salah satu file audio yang sangat dikenal dengan menggunakan Borland Delphi 7. Hasil dari penelitian ini adalah sebuah perangkat lunak yang bernama WAVE Manipulator yang akan berguna untuk melakukan manipulasi pada file WAV dilengkapi dengan kemampuan untuk menentukan bagian dari sample yang akan dimanipulasi dan lainnya untuk memudahkan pemakai program dalam melakukan manipulasi file WAV.

Kata kunci: manipulasi, WAV, WAVE

ABSTRACT: *WAV File is a standard audio file used in Windows. Manipulation of WAV File can be done to obtain different output from the WAV File. These manipulation processes are made into an application that is capable to make changes on WAV File. WAVE Manipulator is an audio application used to change voice from audio files like WAV file. The objective of this project is to develop an application that can be used to change or give effect to the popular audio file known as WAV File using Borland Delphi 7. Application is designed using standard view to make it easier to use. The output of this research is a WAVE Manipulator Software that function as a tool to manipulate WAV File with the ability to choose part of sample for the manipulation, so that user can easily manipulate a WAV File.*

Keywords: *manipulate, WAV, WAVE.*

PENDAHULUAN

Dalam teknologi perangkat lunak dikenal adanya file yang berekstensi WAV. File WAV ini merupakan file audio standar yang digunakan oleh Windows. WAV file ini memungkinkan suara direkam dalam berbagai kualitas, seperti 8-bit atau 16-bit samples dengan rate 11025Hz, 22050Hz atau 44100Hz. Untuk kualitas yang baik, yaitu: 44100Hz, 16-bit akan memakan kapasitas sekitar 150Kbytes setiap detik.

File WAV banyak dipakai dalam pembuatan game. Biasanya untuk suara-suara efek dan musik. WAV sendiri memang cenderung memiliki ukuran yang besar, tetapi hal itu dikarenakan format file WAV yang tidak dikompresi sehingga memiliki kelebihan, yaitu waktu loading yang lebih cepat.

Contoh lain yaitu pada waktu merekam suatu lagu ke komputer, terkadang hasil rekaman yang diperoleh kurang memuaskan. Dalam hal ini bisa saja dilakukan perekaman ulang, tetapi hal tersebut tidak selalu bisa dilakukan, terutama bila sumber rekaman tersebut dari radio.

Untuk itu maka dibuatlah suatu perangkat lunak yang bisa memanipulasi file WAV yaitu sebuah WAVE Manipulator. Perangkat lunak ini nantinya akan bisa membuka file WAV lalu merubah file WAV tersebut, seperti menaikkan atau menurunkan kualitas, memberikan efek fading, merubah pitch, mengkonversi mono ke stereo, dan lainnya. Dengan demikian maka dapat menghasilkan file WAV yang benar-benar cocok dengan 'telinga' masing-masing.

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)[1]. 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*[1].

Bitrate merupakan ukuran bit tiap sampelnya, yaitu 8-bits, 16-bits, 24 bits atau 32 bits. Dalam 8-bits WAV semua sampelnya hanya akan memakan sebanyak 1 byte saja. Sedangkan untuk 16-bits akan memakan 2 bytes. WAV 16-bits tiap sampelnya 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[1].

Samplerate menyatakan banyaknya jumlah sample yang dimainkan setiap detiknya. *Samplerate* yang umum dipakai adalah 8000Hz, 11025Hz, 22050Hz, dan 44100Hz[1].

Untuk 8000Hz biasanya disebut sebagai *Telephone quality* karena suara yang dihasilkan menyerupai suara dari telepon. *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[1].

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

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*.

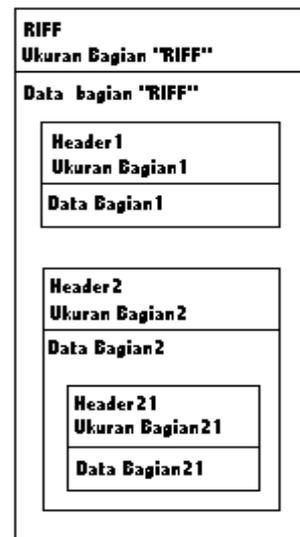
Struktur File WAV

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

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[2].

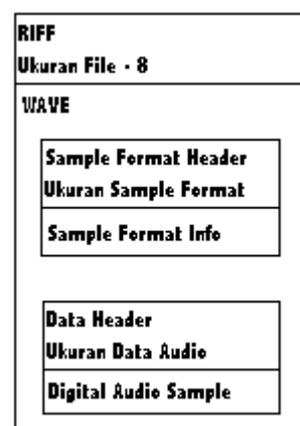


Gambar 1. Struktur file RIFF

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 sub-bagian dari bagian RIFF lalu diikuti sub-bagian data audionya.



Gambar 2. Struktur file WAV

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

Tabel 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 seperti yang ada pada tabel 1.

'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. Untuk lebih jelasnya ada di tabel 2.

Tabel 2. Detail Bagian Sample Format

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 - 0xFFFFFFFF
0x10	4	Average bytes per second	1 - 0xFFFFFFFF
0x14	2	Block align	1 - 65,535
0x16	2	Significant bits per sample	2 - 65,535

Tabel 3. Beberapa kode kompresi

Code	Description
0 (0x0000)	Unknown
1 (0x0001)	PCM/uncompressed
2 (0x0002)	Microsoft ADPCM
6 (0x0006)	ITU G.711 a-law
7 (0x0007)	ITU G.711 μ-law
17 (0x0011)	IMA ADPCM
20 (0x0016)	ITU G.723 ADPCM (Yamaha)
49 (0x0031)	GSM 6.10
64 (0x0040)	ITU G.721 ADPCM
80 (0x0050)	MPEG
65,536 (0xFFFF)	Experimental

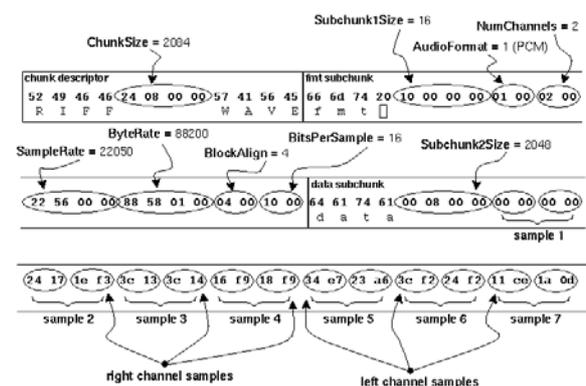
Kompresi kode menempati 2 byte berikutnya dengan nilai 1 untuk PCM. Dua byte berikutnya menyatakan jumlah channel dari file WAV, lalu 4 byte menyatakan sample rate dan 4 byte lagi menyatakan rata-rata byte tiap detiknya. Dua byte setelahnya merupakan *Block Align* yang menyatakan ukuran data untuk satu 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. Detail dari hal ini ada di tabel 3[1].

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. Detailnya ada di tabel 4.

Tabel 4. Detail Bagian Data Audio

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

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 2.3 menunjukkan contoh dari representasi data ke 4 tabel tadi (tabel 1, 2., 3 dan 4) pada file WAV.



Gambar 3. Contoh 72 byte awal dari file WAV

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 <http://thunder.prohosting.com/~jwcs/dap/algor/index.html>.

Sedangkan contoh diambil dengan menggunakan program GoldWave, yaitu program untuk melakukan manipulasi WAV.

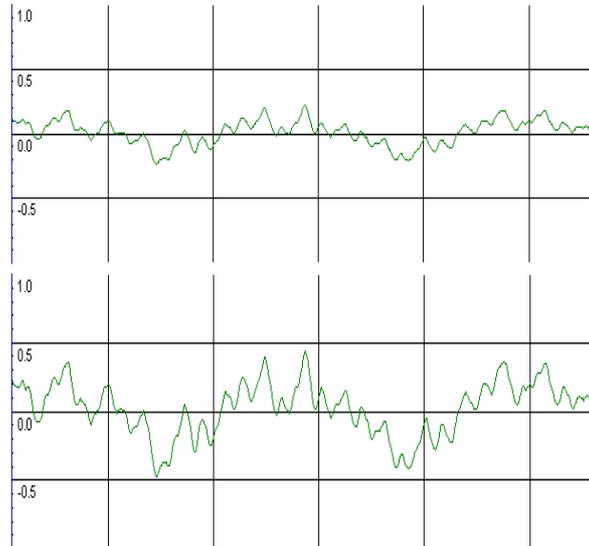
Manipulasi Volume

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 mengubah 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.

Aplikasi 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.

Mengubah Volume

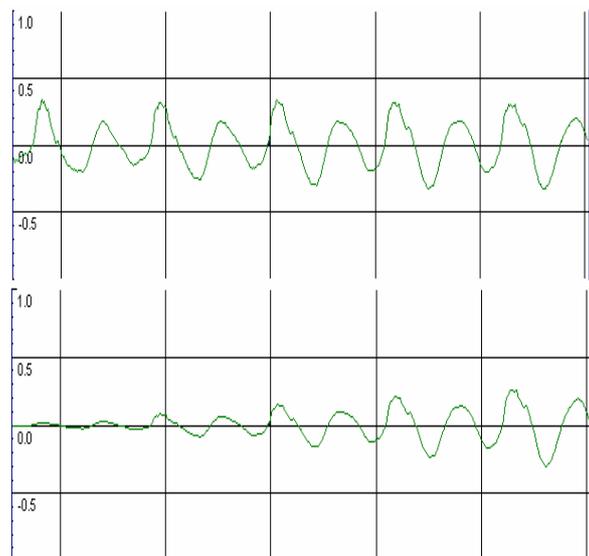
Untuk mengubah volume yang dilakukan adalah proses perkalian dan tidak akan pernah ada proses penjumlahan ataupun pengurangan. Karena misalnya bila dilakukan proses pengurangan, maka akan ada kemungkinan amplitudo melewati nol sehingga akan mengacaukan bentuk gelombangnya. Sedangkan alasan untuk pemakaian perkalian daripada pembagian adalah untuk menghindari pembagian dengan nol. Jadi untuk mengubah volume, maka setiap sample akan dikalikan dengan sebuah angka. Gambar 4 adalah gambar contoh gelombang suara sumber dengan yang telah diperbesar 2 kali.



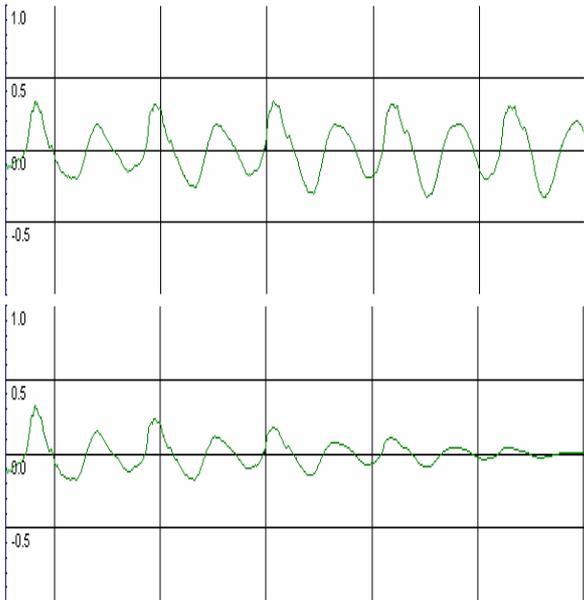
Gambar 4. Contoh gelombang suara diperbesar 2 kali

Fadein dan Fadeout

Fadein dan Fadeout merupakan aplikasi 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 5 merupakan contoh dari gelombang suara yang di fadein sedangkan Gambar 6 adalah gelombang suara yang di fadeout



Gambar 5. Contoh gelombang suara diberi efek fadein



Gambar 6. Contoh gelombang suara diberi efek fadeout

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.

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 awalnya 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.

Manipulasi Channel

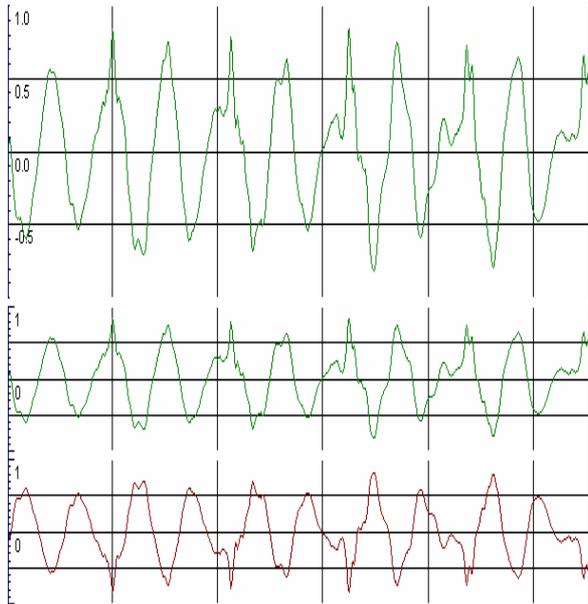
Manipulasi channel dilakukan dengan mengubah nilai jumlah channel pada bagian sample format, namun tentunya pengubahan ini perlu diikuti juga dengan penambahan atau pengurangan data. Misalnya untuk mengubah WAV mono (1 channel) menjadi stereo (2 channel) maka pada bagian data perlu ditambahkan sample 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 sampelnya. 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 sampelnya 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.

Mengubah Mono ke Stereo

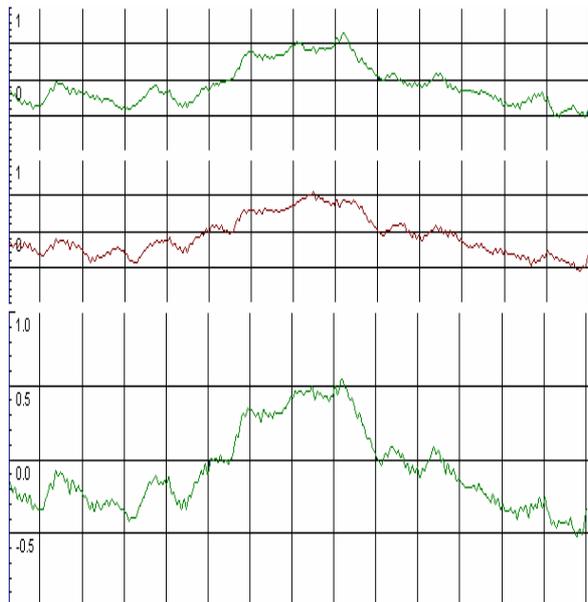
File WAV yang awalnya hanya memiliki satu channel bisa diubah menjadi dua channel. Channel pertama tidak akan diubah-ubah, sedangkan untuk channel kedua diisi dengan membalik fase gelombang pada channel pertama. Dengan melakukan ini maka channel satu maupun dua akan menghasilkan suara yang sama, karena pengubahan mono ke stereo ini tidak akan bisa menghasilkan suara sebaik stereo yang aslinya. Hal ini terlihat pada Gambar 7.



Gambar 7. Contoh gelombang suara diubah dari mono ke stereo

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 sampelnya. Hal ini terlihat pada Gambar 8.



Gambar 8. Contoh gelombang suara diubah dari stereo ke mono

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.

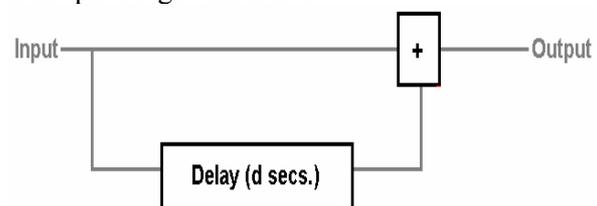
Time Warp

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 penambahan sample untuk memanjangkan file WAV dan pengurangan sample untuk memendekkan 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.

Efek Echo

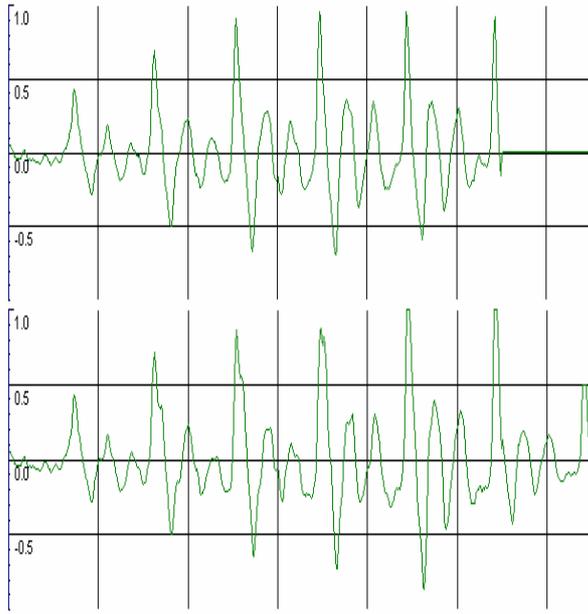
Efek echo ini adalah suara yang seolah-olah memantul sehingga akan terdengar lagi setelah jangka waktu tertentu dengan volume yang berbeda.

Cara kerjanya terlihat pada Gambar 9, 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 9. Ilustrasi efek echo

Hasil penerapan dari efek echo ini pada gelombang suara dapat dilihat pada Gambar 10.



Gambar 10. Contoh gelombang suara diberi efek echo dengan delay 5ms dan volume 50%

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.

DESAIN PROGRAM

Aplikasi yang akan dibuat ini menerapkan sistem MDI, sehingga akan memungkinkan untuk membuka lebih dari satu file WAV secara bersamaan.

Program *Wave Manipulator* ini akan bisa membuka maupun menyimpan kembali file WAV. Namun file WAV yang disupport adalah yang bertipe PCM, yaitu file WAV yang tidak terkompresi. File WAV yang dibuka akan digambarkan dalam bentuk

spektrum-nya. File WAV yang dibuka tersebut akan bisa dimanipulasi sesuai dengan pilihan-pilihan manipulasi yang tersedia. Selain itu file WAV tersebut juga bisa dimainkan.

Dalam melakukan manipulasi, user bisa memilih untuk memanipulasi sebagian saja dari file WAV tersebut. Namun ada beberapa manipulasi seperti mengganti sample rate yang harus dilakukan pada file WAV secara keseluruhan. Sama halnya untuk memainkan file WAV tersebut, user bisa memainkannya mulai dari waktu tertentu sampai dengan waktu tertentu.

Untuk masalah penggambaran spektrum, normalnya akan digambarkan dalam ukuran berskala. Namun user bisa melakukan pembesaran sampai dengan ukuran satu banding satu.

Dalam hal memanipulasi file WAV, program WAVE Manipulator ini tidak menggunakan komponen khusus untuk menangani file WAV, semua akan dilakukan secara prosedural. Namun untuk memainkan file WAV yang dibuka, akan digunakan komponen yaitu Window Media Player.

Analisa Kebutuhan

Secara garis besar, aplikasi WAVE Manipulator ini akan berjalan dalam 3 tahap, yaitu:

1. Membuka file WAV

Tahap pertama adalah membuka file WAV dari disk lalu menggambarkan isi file WAV tersebut dalam bentuk spektrum. Untuk menggambarkan spektrum ini harus diusahakan agar proses penggambaran tidak memakan waktu yang lama, demikian pula ketika membaca file tersebut dari disk.

2. Memanipulasi file WAV

Dalam memanipulasi file WAV, program perlu diberikan fasilitas untuk melakukan pembatalan atau yang biasa disebut *undo*. Selain itu perlu juga diberikan fasilitas untuk memainkan file WAV agar user bisa mendengarkan hasil dari manipulasi. Hal lain yang tidak kalah penting adalah proses manipulasi harus bisa berjalan dalam waktu yang sesingkat mungkin.

3. Menyimpan file WAV

Setelah selesai dengan manipulasi, maka user akan diberikan pilihan untuk dapat menyimpan file WAV tersebut kembali ke disk. Untuk penyimpanan ke disk ini bisa dibayangkan tidak akan ada masalah, yang diperlukan hanya kecepatan saja. Sebaiknya program juga bisa menyimpan file dengan nama lain.

Cara Kerja Program

Berdasarkan analisa kebutuhan yang telah dijelaskan, maka dirancang suatu desain cara kerja program WAV Manipulator yang akan dibuat.

Membuka File Wav

Untuk membuka file WAV, pertama-tama akan dilakukan pembacaan mengenai informasi dari file WAV tersebut, yaitu Compression Code, Number of Channel, Sample Rate, Average Bytes per Second, Block Align, Significant bits per Sample, dan Data Size. Setelah itu dihitung jumlah sample yang ada pada file WAV tersebut dengan rumus $TotalSample = DataSize \div BlockAlign$. Yang dimaksud dengan total sample ini adalah jumlah sample penuh.

Untuk membaca data samplanya, pertama-tama ditentukan besar buffer. Semakin besar buffer akan semakin cepat proses pembacaan dari file, dalam hal ini penulis membatasi ukuran buffer sebanyak 1 juta byte. Pada waktu membaca sample, perlu dipastikan agar jumlah byte yang terbaca selalu merupakan suatu sample penuh.

Langkah-langkah membaca sample adalah pertama ambil data dari file dan simpan dalam Buffer. Data dalam buffer ini masi merupakan data yang belum diolah. Untuk mengambil nilai sample dalam buffer perlu diperhatikan nilai Significant bit per Sample-nya atau bitrate-nya. Untuk bitrate 8 maka setiap byte dalam buffer menyatakan satu sample sedangkan untuk 16 bit maka setiap 2 byte menyatakan satu sample, dan seterusnya. Untuk 16 bit ke atas, maka perlu dilakukan penyatuan byte menjadi nilai sample, untuk melakukan ini dapat dilakukan dengan rumus:

$$Sample = byte-1 + byte-2 \times 16 + byte-3 \times 256 + byte-4 \times 65536 + \dots$$

atau

$$Sample = byte-1 + byte-2 \times 16^1 + byte-3 \times 16^2 + byte-4 \times 16^3 + \dots$$

Penyatuan byte menjadi sample ini akan memakan waktu yang banyak, mengingat banyaknya jumlah sample dalam satu file WAV. Untuk itu maka proses perkalian digantikan dengan proses penggeseran biner atau yang disebut *binary shifting*. *Binary shifting* ini bisa dilakukan untuk semua perkalian dengan angka 2 pangkat (2^n). Mengingat bahwa 16 adalah 2^4 maka perkalian dengan 16 ini juga bisa dilakukan dengan *binary shifting*. Proses *binary shifting* ini jauh lebih cepat daripada perkalian. Contoh *binary shifting*, misalnya akan mengalikan angka 3 dengan angka 4 (2^2), maka hanya perlu menggeser nilai biner dari angka 3 ke kiri sebanyak 2

kali. Nilai biner dari angka 3 adalah 11 setelah digeser 2 kali ke kiri akan menjadi 1100. 1100 ini sama dengan 12 dalam desimal. *Binary shifting* ini juga dipakai ketika membaca informasi dari file WAV. Untuk proses pembagian dengan 2^n , juga bisa dilakukan dengan *binary shifting* tetapi penggeseran dilakukan ke kanan.

Setelah memperoleh nilai samplanya maka perlu dilihat apakah nilai tersebut melebihi nilai batas atas dari range yang ditentukan. Untuk 8-bit batas atasnya adalah 127, untuk 16-bit adalah 32767. Untuk WAV 8-bit data dalam file disimpan dalam bentuk *unsigned* (tak bertanda), sehingga semua sample yang diperoleh nilainya perlu dikurangi dengan 128, karena nilai diam pada WAV 8-bit adalah 128. Sedangkan untuk 16-bit ke atas data disimpan dalam bentuk *signed* (bertanda). Untuk 16-bit maka hanya sample yang bernilai di atas 32767 saja yang perlu dikurangi dengan 65536 (panjang range sample untuk 16-bit).

Menyimpan File WAV

Untuk menyimpan file WAV harus mengerti struktur dari file WAV. Pertama-tama tulis 'RIFF', lalu 4 byte berikutnya adalah ukuran dari RIFF atau filesize dikurangi 8, lalu 'WAVE'. Berikutnya adalah 'fmt', diikuti dengan ukuran format yaitu 16 untuk PCM, dan diikuti oleh informasi format dari file WAV. Setelah itu barulah tulis 'data' beserta DataSize ke file.

Setelah selesai semua itu, barulah data audio akan diproses. Pertama-tama hitung besar buffer yang akan digunakan sama seperti ketika membuka file WAV. Variable BLoaded dipakai untuk melihat apakah Buffer sudah terisi sesuai dengan BSize, sedangkan CurrentSample dipakai untuk menghitung apakah semua sample sudah ditulis ke file. Prinsipnya adalah memasukkan setiap sample ke dalam buffer, lalu jika buffer sudah terisi sebanyak BSize, maka dilakukan penulisan dari buffer ke file.

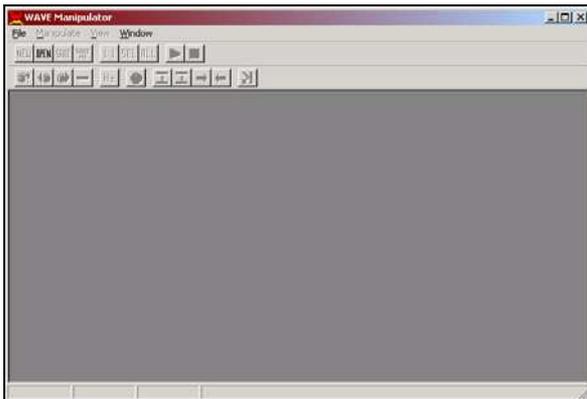
Untuk memasukkan sample ke dalam buffer, perlu diingat bahwa data audio pada file WAV menggunakan metode Little Endian. Kalau pada waktu membuka file WAV dilakukan penggabungan byte menjadi sample, maka untuk menyimpan file WAV perlu dilakukan pemisahan dari sample ke byte. Hal ini dilakukan dengan mengubah nilai sample yang dalam desimal menjadi hexadesimal. Perlu diingat sehubungan dengan metode Little Endian, maka nilai hexadesimal perlu dibalik, namun pembalikan dilakukan setiap 1 byte, contoh hexa 0F38 dibalik menjadi 380F, bukan 83F0.

Untuk lebih memudahkan maka dapat dilakukan dengan pembagian dengan angka 256. Sebagai

contoh untuk nilai sample 4000, maka lakukan pembagian dengan 256, akan menghasilkan nilai 15 dengan sisa 160. Angka 160 ini diambil sebagai byte pertama dari sample. Lalu 15 dibagi lagi dengan 256, akan menghasilkan 0 dengan sisa 15. Angka 15 ini diambil sebagai byte kedua.

IMPLEMENTASI PROGRAM

Program WAVE Manipulator ini dirancang dengan menggunakan sistem MDI Application. Yang dimaksud adalah *Multi Document Interface*, dimana dengan sistem ini maka user dapat bekerja dengan lebih dari satu dokumen dan dalam program ini user akan bisa membuka lebih dari satu file WAV secara bersamaan. Tampilan dari program ini menggunakan tampilan sebagaimana program-program MDI lainnya, yaitu dilengkapi dengan menu utama (Main Menu), toolbar, serta statusbar. Pertama kali program dijalankan maka tidak ada file yang dibuka, sehingga bagian tengah program akan kosong. Seperti gambar 11.



Gambar 11. Tampilan awal program WAVE Manipulator

Sedangkan susunan menu yang ada pada program ini adalah sebagai berikut:

1. File
 - Terdiri atas submenu open, save, saveas dan close. Yang berguna untuk membuka file, menyimpan file dan menutup program.
2. Manipulate
 - Terdiri dari submenu
 1. undo untuk membatalkan manipulasi yang telah dilakukan,
 2. volume untuk memanipulasi volume. Sub menu ini terdiri dari 4 submenu lagi yaitu fade in, fade out, silence dan volume
 3. channel untuk memanipulasi channel. Sub menu ini terdiri dari 4 submenu lagi yaitu change number of channel, exchange, pan left -> right, dan pan right -> left.

4. Change Sample rate untuk memanipulasi sample rate dari file wav
 5. Time warp untuk memperpanjang dan memperpendek file
 6. Echo untuk menambah file echo
3. View
 - Menu ini berguna untuk menampilkan child window yang tertutup oleh window yang lain.
 4. Window
 - Terdiri atas 2 submenu untuk mengatur child windows yang ada yaitu Cascade atau Tile.

Selain menu yang disebutkan masih ada tambahan 2 toolbar, seperti yang bisa dilihat pada gambar 12 untuk toolbar standar. Dan gambar 13 untuk toolbar manipulasi.



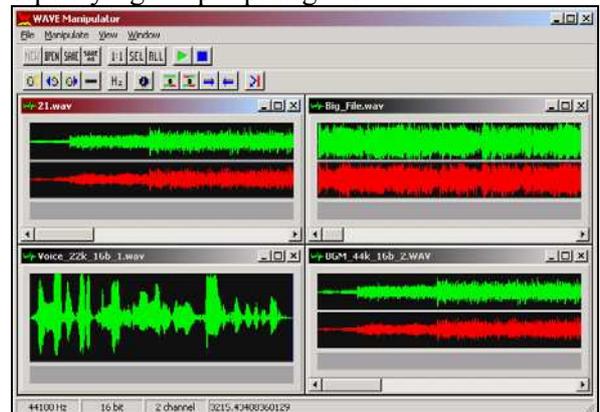
Gambar 12. Toolbar standard

Gambar 13. Toolbar Manipulasi

Sama seperti halnya dengan menu utama, banyak tombol pada toolbar ini yang tidak menyala jika tidak ada file yang sedang dibuka. Untuk tombol New pada toolbar ini masih belum diimplementasikan, jadi tombol ini tidak akan pernah menyala.

PENGUJIAN PROGRAM

Berikut ini adalah pengujian program terhadap fitur-fitur yang dipunyainya, dimulai dari fitur MDI seperti yang nampak pada gambar 14.



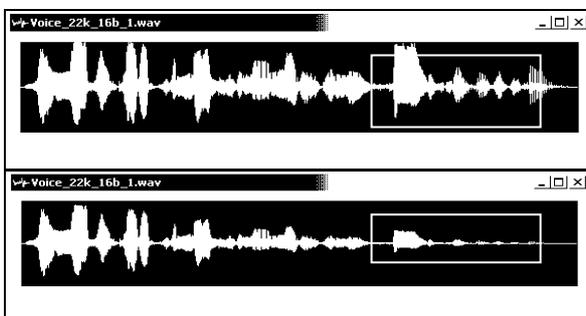
Gambar 14. Pengujian fitur MDI

Gambar 15 bawah merupakan hasil pengujian program untuk fade in terhadap satu file wav di gambar 15 atas.

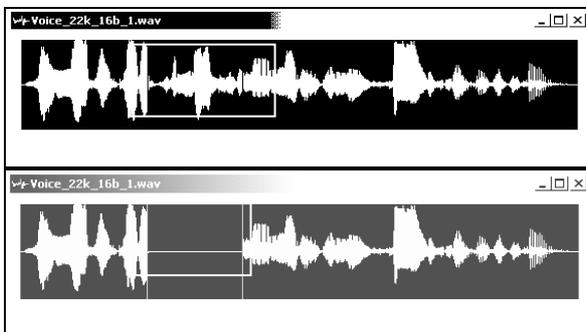


Gambar 15. Contoh efek fade in pada file wav

Sedangkan untuk efek fade out dan silent terdapat gambar 16 bawah dan 17 bawah terhadap sumber dari gambar 16 atas dan gambar 17 atas.



Gambar 16. Efek fade out



Gambar 17. Efek silent

KESIMPULAN

Hasil pembahasan di atas dapat disimpulkan seperti berikut yaitu:

- Untuk membaca atau menulis file WAV maka akan lebih baik jika buffer yang digunakan berukuran besar, karena akan meningkatkan kecepatan atau performance.
- Bila mengubah WAV mono menjadi stereo lalu diubah menjadi mono lagi, maka akan diperoleh file WAV yang tidak bersuara sama sekali karena algoritma yang digunakan bertolak belakang.
- Komponen TImage dari Borland Delphi bisa digunakan untuk membuat gambar, tetapi kom-

ponen ini memiliki performance yang sangat mengecewakan sehingga jika akan menggambaran seluruh sample dalam file WAV bisa memakan waktu yang sangat lama.

DAFTAR PUSTAKA

- Wave Files-The Sonic Spot, Available: <<http://www.sonicspot.com/guide/wavefiles.html>>
- WAVE File Format, Available: <<http://www.borg.com/~jglatt/tech/wave.htm>>
- Microsoft WAVE soundfile format, Available: <<http://www.stanford.edu/CCRMA/Courses/422/projects/WaveFormat/>>
- JWCS Digital Audio Programming, Jonathan Worthington, Available: <<http://thunder.Prohosting.com/~jwcs/dap/index.html>>
- Bit Wars, Mitch Gallagher, Available: <<http://archive.keyboardonline.com/features/bitwars/bitwars.shtml>>