

## MAKALAH SEMINAR TUGAS AKHIR

### PERANCANGAN PROGRAM APLIKASI KONVERSI DAN KOMPRESI SUARA KE FORMAT MP3

Hapsoro Haryo Baskoro\*, Maman Somantri, MT\*\*, Imam Santoso, MT\*\*

Pada saat ini format mp3 sangat populer di bidang multimedia dalam hal penyimpanan data suara, sebelum ada format ini untuk menyimpan data suara dibutuhkan ruang media penyimpanan yang cukup besar. Pada saat menentukan tema tugas akhir ini, untuk mengubah suara analog menjadi data suara berformat mp3 dibutuhkan dua perangkat lunak yang berbeda.

Untuk itu dalam tugas akhir ini dibahas tentang perancangan program aplikasi yang bertujuan untuk membuat data suara berformat mp3 yang cukup sederhana dan efisien. Program aplikasi ini hanya membunyikan satu tombol untuk menjalankan perintah sehingga cukup mudah untuk menggunakannya.

Hasil akhir dari tugas akhir ini adalah suatu program aplikasi dengan ukuran data yang tidak terlalu besar sehingga mudah untuk dipindah-pindahkan dan dipasang ke komputer yang berbeda-beda. Program aplikasi ini menghasilkan format mp3 dengan konfigurasi suara minimal 8 bit, Mono, dan 8.000 Hz.

#### I. PENDAHULUAN

##### 1.1 Latar Belakang

Pada saat sekarang ini, komputer multimedia telah menjadi hal yang amat populer dan diminati oleh banyak orang, terutama dalam bidang *audio processing*. Hal ini berupa pengkonversian suara analog menjadi digital, atau dengan kata lain suara yang didengar oleh manusia dapat disimpan ke dalam media digital seperti hard disk dan media penyimpanan yang paling populer pada saat ini adalah Compact Disk atau CD.

Sebelum teknologi ini ada, perekaman suara dilakukan pada media pita kaset. Tetapi *lifetime* dari pita kaset ini rendah sehingga suara yang dihasilkan semakin lama semakin buruk. Oleh karena itu munculah teknologi CD Audio, tapi teknologi ini juga masih memiliki kelemahan. Kelemahan yang terbesar adalah format DAT yang ada di dalamnya memiliki kapasitas yang besar sehingga sangat memboroskan media penyimpanannya. Kelemahan yang sama juga dialami oleh format suara digital yang lain, yaitu WAV. Oleh karena itu diciptakan suatu teknologi kompresi suara digital, salah satu format kompresi yang terkenal saat ini adalah MP3.

##### 1.2 Tujuan dan Manfaat.

Dengan tugas akhir ini akan dibuat suatu program aplikasi yang dapat merekam suara

analog menjadi data suara digital dengan format mp3 yang bersifat sederhana, mudah, dan fleksibel. Sederhana dalam artian jumlah kapasitas file program aplikasi yang tidak terlalu besar sehingga mudah disimpan dalam sebuah disket, mudah dalam artian mudah dalam pengoperasiannya yaitu dengan hanya menyentuh satu tombol saja, serta fleksibel dalam artian mudah dipasangkan di komputer mana saja dengan sistem operasi windows tanpa harus melakukan proses instalisasi.

##### 1.3 BATASAN MASALAH.

Pada tugas akhir ini akan dihadirkan beberapa masalah tetapi dengan lingkup yang jelas, yaitu :

- Program aplikasi yang dibuat menggunakan kompilator Borland Delphi 6.0
- Program aplikasi yang dibuat diproyeksikan untuk dijalankan pada komputer pribadi dengan spesifikasi minimal, yaitu prosesor Pentium MMX, sistem operasi Windows 98, sound card ( kartu suara ) 16 bit, VGA card 4 MB, dan monitor 14".
- Program aplikasi dibuat untuk mengkonversi suara analog menjadi suara digital dengan tambahan berupa tampilan waktu untuk mengetahui lamanya proses kerja program aplikasi.

## II. LANDASAN TEORI

### 2.1 TEKNOLOGI MP3

Walaupun media penyimpanan sekarang ini sudah relatif sangat besar dan sangat murah jika dibandingkan 10 tahun yang lalu, ukuran data yang besar ini tetap mengganggu. Media penyimpanan CD dapat menyimpan data sebanyak 650 MB ( 1 MB = 2 pangkat 20 = 1.048.576 byte ), DVD *single layer* dapat menyimpan data 4,7 GB, DVD *double layer* dapat menyimpan data sebanyak 8,5 GB, apalagi Hard Disk yang ada pada saat ini sudah mencapai kapasitas di atas 100 GB.

Masalah besarnya ukuran data suara tetap harus dipecahkan walaupun media penyimpanan yang tersedia sudah begitu besar. Meskipun demikian, masih banyak kendala lain yang menyebabkan ukuran data ini cukup mengganggu, antara lain proses pembacaan dari media penyimpanan yang masih belum maksimal sehingga dibutuhkan waktu yang cukup lama untuk membaca data suara dari media penyimpanan, proses transfer dengan kapasitas data suara yang besar sangat tidak ekonomis jika digunakan dalam jaringan komputer karena data sebesar itu akan sangat mengganggu lalu lintas jaringan.

Karena itu dibutuhkan satu cara untuk mereduksi ukuran data suara digital. Teknik yang digunakan adalah dengan teknik kompresi data. Kompresi yang digunakan haruslah memenuhi spesifikasi sebagai berikut :

- Tetap menjaga kualitas suara asli.
- Memiliki kekuatan (*robustness*) terhadap kesalahan-kesalahan bit secara random pada seluruh kanal yang dikompresi.
- Memiliki faktor kompleksitas yang tidak terlalu tinggi.
- *Encoding* dan *decoding* tidak membutuhkan daya *processor* yang terlalu besar.
- *Delay* waktu yang disebabkan untuk proses *dekoding* masih dapat ditolerir.

Untuk itulah dihasilkan suatu teknik kompresi yang bernama MP3. Saat ini bagi pengguna MP3 ada sedikit salah paham mengenai MP3. Banyak yang menyangka bahwa MP3 adalah singkatan dari MPEG-3. Tetapi pada kenyataannya MP3 bukanlah singkatan dari MPEG-3, melainkan singkatan dari MPEG Audio Layer 3. MPEG-3 dan *Layer 3* adalah hal yang berbeda. *Layer 3* adalah bagian dari MPEG yang berupa *coding audio* yang sangat dahsyat.

*Layer 3* didefinisikan sebagai bagian suara dari standar MPEG-1 dan MPEG-2.

MPEG merupakan singkatan dari Moving Picture Expert Group yang merupakan kumpulan para pakar dalam bidang *coding* untuk citra bergerak (video) dan suara (audio). MPEG bergerak di bawah bimbingan ISO (International Standard Organization) dan IEC (International Elektro-Technical Commission).

MPEG melakukan kompresi suara dengan menggunakan metode *perceptual coding*. Ada tiga jenis *coding* yang ada, yaitu *Layer 1*, *Layer 2*, dan *Layer 3*. Makin tinggi nilai *Layer*, makin tinggi kompleksitas perhitungan, *delay* pada *codec* makin besar, dan kinerjanya makin tinggi. Setiap *Layer*-pun memiliki kompatibilitas ke bawah (*backward compatibility*). Maksudnya *decoder Layer N* akan mampu juga untuk melakukan *decoding* terhadap data *Layer 1*, *Layer 2*, dan *Layer 3*. Sedangkan *decoder Layer 2* hanya mampu untuk melakukan *decoding* terhadap data *Layer 1* dan *Layer 2*.

Setiap *Layer* memiliki standar masing-masing. Standar tersebut meliputi format *bitstream* dan *decoder*-nya. Untuk memungkinkan perbaikan di masa mendatang, standar tersebut tidak meliputi *encoder*. Walaupun demikian setiap *Layer* memiliki banyak persamaannya, diantaranya :

- Semua *Layer* menggunakan struktur dasar yang sama. Menggunakan teknik yang sama, yaitu dengan memanfaatkan *perceptual subband coding*.
- Semua *Layer* menggunakan analisis *filterbank* (*polyphase* dengan 32 *subband*). *Layer 3* menambahkan transformasi MDCT untuk meningkatkan resolusi frekuensi.
- Semua *Layer* menggunakan informasi *header* yang sama dalam *bitstream*-nya untuk mendukung struktur hirarki dari standar yang telah ditetapkan.
- Semua *Layer* memiliki sensitifitas terhadap kesalahan bit (*bit error*) yang mirip. Ketiga *Layer* tersebut menggunakan struktur *bitstream* yang lebih sensitif terhadap *bit error*, yaitu *header*, *bit allocation*, *scale factor*, dan *side information*. Sedangkan yang kurang sensitif terhadap *bit error*, yaitu data komponen spektrum.
- Semua *Layer* menggunakan *sampling rate* 32 KHz, 44,1 KHz atau 48 KHz.
- Semua *Layer* diijinkan bekerja dengan *bit rate* yang mirip :  
*Layer 1* : 32 kbps – 448 kbps.  
*Layer 2* : 32 kbps – 384 kbps.

Layer 3 : 32 kbps – 320 kbps.

Setiap *Layer* menawarkan kemampuan kompresi yang berbeda-beda. Lengkapnya dapat kita lihat pada tabel di bawah ini. Faktor pembandingan kompresi yang digunakan untuk MP3 adalah suara dengan format stereo, 16 bit per sample, frekuensi sampling 48 KHz. Dengan format seperti ini bit rate tanpa kompresi adalah 1.536.000 bit per detik. Dengan menggunakan *Layer 3*, data suara dengan format seperti ini dapat dihasilkan hanya dengan menggunakan bit rate 128.000 bit per detik, sehingga faktor kompresi yang dihasilkan adalah 12 : 1. Suatu keuntungan yang sangat besar.

<i>Layer</i>	Bit Rate	Faktor Kompresi
I	384 kbps	4
II	192 kbps	8
III	128 kbps	12

**Tabel 2.1.** Faktor kompresi MPEG audio pada setiap *Layer*.

## 2.2. TEKNIK KOMPRESI MPEG AUDIO

Sebenarnya teknik kompresi MP3 adalah teknik kompresi MPEG-1 Audio atau MPEG-2 Audio. Teknik kompresi MP3 hanya menggunakan MPEG-2 Audio. Teknik kompresi MP3 hanya menggunakan *Layer 3*, dari 3 *Layer* yang ada. Teknik kompresi MPEG Audio *Layer 1* dikenal dengan nama MP1, sedangkan teknik kompresi MPEG Audio *Layer 2* dikenal dengan nama MP2.

Teknik kompresi yang dilakukan MPEG Audio merupakan suatu teknik *Lossy Compression*. Maksudnya, informasi yang terkandung dalam data yang terkompresi akan dihilangkan sebagian data aslinya agar faktor kompresi menjadi lebih tinggi. *Lossy Compression* digunakan untuk melakukan kompresi terhadap data yang tidak terlalu penting dan karena pertimbangan ilmiah (melalui analisis matematis dan biologis) dapat dihilangkan tanpa mempengaruhi persepsi penggunaannya. Contoh *Lossy Compression* adalah format JPG pada gambar, MPEG pada video dan MP3 pada audio. Kebalikan dari *lossy compression* adalah *loseless compression*. Pada *Loseless Compression*, data yang terkompresi mengandung seluruh informasi data aslinya tanpa ada informasi yang dihilangkan. Teknik kompresi ini digunakan untuk melakukan kompresi terhadap data yang sangat penting dan tidak boleh rusak atau hilang

satu bit-pun. Contoh *loseless compression* adalah format file ZIP, RAR, LHA, ARJ, dan CAB.

Walaupun pada dasarnya kompresi MPEG Audio merupakan *lossy compression*, MPEG menjamin bahwa secara persepsi kompresi MPEG Audio adalah *loseless compression*. Maksudnya walaupun MP3 menghilangkan sebagian data, kemungkinan besar kita sebagai pendengar tidak akan menyadari hal tersebut. Sebagai buktinya, dua ilmuwan yaitu Greewin dan Rayden telah menguji suatu suara dengan format 16 bit, stereo, frekuensi sampling 48 KHz. Beberapa pendengar suara yang sudah pakar diundang untuk mengujinya. Suara tersebut dikompresi dengan format MP3 dengan tetap menyimpan format aslinya. Ketika kedua suara tersebut diperdengarkan, para pendengar yang sudah pakar tersebut tidak berhasil membedakan perbedaan antara format MP3 dengan format aslinya. Berbagai analisis terhadap anatomi manusia dan perhitungan matematis berada di balik kemampuan besar ini.

Kemampuan MP3 dalam mengkompresi file secara besar-besaran pada hakikatnya memanfaatkan sifat dari panca indra manusia. Telinga manusia mempunyai kelemahan yang ternyata dapat dimanfaatkan untuk melakukan kompresi. Kelemahan ini dikenal dengan nama *Auditory Masking*. *Masking* ini adalah kelemahan secara persepsi dari sistem pendengaran manusia yang terjadi kapanpun ketika adanya suatu sinyal kuat pada frekuensi tertentu. Dengan adanya sinyal yang kuat ini akan menyebabkan frekuensi-frekuensi di sekitar sinyal tersebut menjadi kurang terdengar oleh telinga manusia. Fenomena ini telah diselidiki oleh banyak ahli dengan menggunakan eksperimen-eksperimen *psychoacoustic*.

Ada satu analogi yang dapat diperhatikan. Suatu malam ada seorang pengendara mobil melihat pejalan kaki yang menyeberang jalan, tetapi sesaat kemudian lampu sorot dari sebuah mobil yang berlawanan arah mengenai mata pengendara. Maka pada saat itu pengendara tidak dapat melihat pejalan kaki yang menyeberang jalan tersebut, yang dapat dilihat hanyalah cahaya lampu sorot tersebut. Hal ini disebabkan karena cahaya dari lampu sorot menyebabkan terjadinya *masking*, menutupi kemampuan mata menerima cahaya yang lebih lemah.

Sifat lain dari sistem pendengaran manusia yang dimanfaatkan oleh MP3 adalah sifat non-linieritas. Sensitivitas telinga manusia untuk berbagai frekuensi tidak tersebar secara

linier. Setelah diadakan penyelidikan, ternyata untuk frekuensi rendah telinga kita mempunyai sensitivitas yang tinggi. Makin tinggi frekuensi, makin rendah sensitivitas telinga.

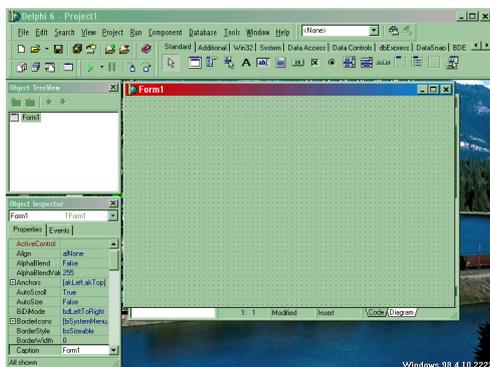
### III. PERANCANGAN SISTEM

#### 3.1. ANALISA KEBUTUHAN.

Untuk membuat program aplikasi dibutuhkan beberapa persyaratan tertentu, antara lain :

1. Kebutuhan perangkat keras.
  - Minimal bisa berjalan dengan prosesor Pentium MMX.
  - Memiliki peralatan yang mendukung pengolahan suara, yaitu untuk memasukkan suara (mikrofon) dan mengeluarkan suara (soundcard dan speaker).
2. Kebutuhan perangkat lunak.
  - Minimal bisa digunakan dengan sistem operasi Windows 98.
  - Bahasa pemrograman Borland Delphi 6.0.

Delphi sebenarnya berasal dari bahasa pemrograman yang cukup terkenal, yaitu Pascal. Pada saat itu sistem operasi masih berbasis teks, yaitu DOS. Pada tahun 1995 telah dikembangkan sistem operasi berbasis visual atau GUI ( Graphical User Interface ), yaitu Microsoft Windows 95. Karena pemrograman Windows dengan Pascal masih dirasa cukup sulit, maka diciptakanlah pemrograman berbasis visual, yaitu Delphi.



Gambar 3.3. Borland Delphi 6.0

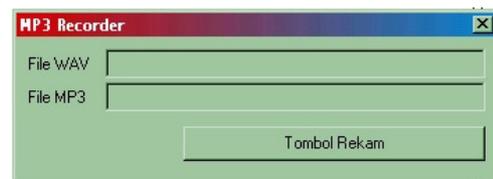
Setelah menentukan kebutuhan, kemudian dianalisa komponen apa saja yang akan dibutuhkan dalam pembuatan program aplikasi ini. Pada dasarnya program aplikasi dibagi atas tiga bagian, ketiga bagian tersebut memiliki komponen tertentu, antara lain :

1. Mengkonversi suara analog menjadi File WAV.
  - TButton  
Digunakan untuk memulai kerja program aplikasi.
2. Mengkompresi file WAV ke file MP3.
  - TACMConvertor  
Komponen ini bukan komponen standar yang dimiliki oleh delphi, dan harus diimpor dari luar.
3. Menyimpan data suara.
  - TSaveDialog  
Digunakan untuk menyimpan dan memberi nama data suara digital.
  - TEdit  
Digunakan untuk menampilkan nama file data setelah disimpan.

#### 3.2. DESAIN SISTEM.

##### 3.2.1. Desain Tampilan.

Rancangan awal dari tampilan program aplikasi ini adalah sebagai berikut :



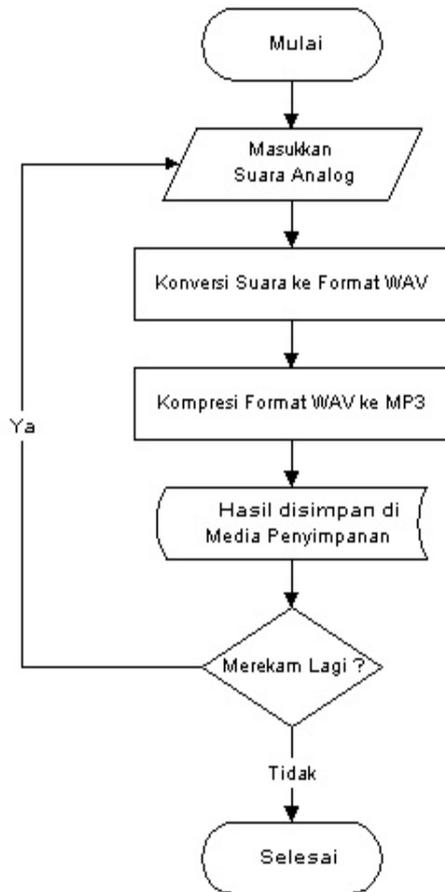
Gambar 3.1. Desain awal program aplikasi.

Pada tampilan di atas hanya komponen visual yang terlihat, seperti TForm, TButton, dan TEdit. Sedangkan komponen yang lain tidak terlihat karena bersifat non visual.

##### 3.2.2. Proses Kerja Program Aplikasi.

Sebelum merancang dan membuat program aplikasi terlebih dahulu memahami cara kerja program aplikasi ini.

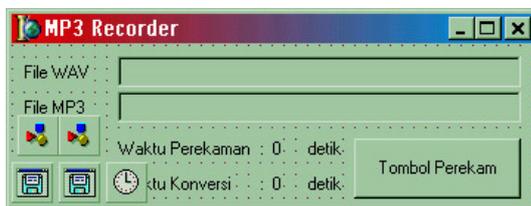
Secara ringkas proses kerja dari program aplikasi ini dapat digambarkan menjadi diagram alur dibawah ini:



Gambar 3.2. Diagram Alur.

### 3.3. MEMBUAT PROGRAM APLIKASI

#### 3.3.1. Merancang Tampilan Program Aplikasi



Gambar 3.3. Rancangan Program Aplikasi.

Secara berturut-turut dipasang beberapa komponen untuk membentuk rancangan tampilan program aplikasi, antara lain :

1. 1 buah komponen TButton.
2. 2 buah komponen TSaveDialog.
3. 2 buah komponen TEdit.
4. 2 buah komponen TACMConvertor.

5. 1 buah komponen TTimer.
6. Dan beberapa komponen TLabel.

#### 3.3.2. Penulisan Instruksi Program Aplikasi.

Secara garis besar program aplikasi dibagi menjadi 2 bagian yaitu :

##### 1. Bagian Perekaman

Proses Perekaman dimulai setelah menekan Tombol Perekam dan untuk menghentikannya adalah dengan menekan tombol yang sama. Untuk membedakan antara memulai dan menghentikan proses ini, Program menggunakan operasi invers terhadap variabel "Rekam". Variabel Rekam diisi dengan nilai *true* atau *false*. Jika bernilai *true*, program mulai merekam suara, jika bernilai *false*, program menghentikan perekaman. Contoh instruksi program (procedure BtnRecordClick) ada di bawah ini.

```

Rekam := not Rekam;
If Rekam then
Begin
  BtnRecord.Caption := 'Tombol Stop';
  .....
  {Bagian Perekaman}
  .....
end
else
Begin
  BtnRecord.Caption := 'Tombol Perekam';
  .....
  {Bagian penghentian Perekaman dan penyimpanan file}
  .....
end;
  
```

Rekam adalah variabel boolean yang terlebih dahulu diisi dengan nilai "False" di dalam procedure FormCreate. BtnRecord adalah variabel dari komponen TButton.

Pada bagian perekaman dan penghentian perekaman akan digunakan fungsi MCI, karena fungsi perekaman yang terdapat pada komponen TMediaPlayer memiliki masalah dalam merekam suara sehingga tidak bisa dipakai dalam pembuatan program aplikasi ini. Sebagai penggantinya digunakanlah MCI.

Pada pembuatan program aplikasi ini, fungsi MCI yang dipakai adalah mciSendString karena penggunaannya relatif sederhana. Fungsi mciSendString dideklarasikan pada MCI sebagai berikut :

```

MCIERROR mciSendString(
    LPCTSTR lpszCommand, LPTSTR
    lpszReturnString, UINT cchReturn,
    HANDLE hwndCallback);

```

Parameter `lpszCommand` adalah perintah yang diberikan pada piranti multimedia. Parameter `lpszReturnString` dan `cchReturn` masing-masing menyatakan string dan panjang string yang digunakan untuk menampung hasil kembalian dari perintah `lpszCommand`. Parameter `hwndCallback` adalah alamat jendela yang menerima kejadian dari perintah `lpszCommand`. Kegunaan parameter ini mirip dengan penanganan kejadian pada Object Inspector.

Fungsi `mciSendString` mengembalikan 0 jika berhasil dan kode kesalahan jika gagal. Ada banyak perintah multimedia yang bisa diberikan pada `mciSendString`. Berikut ini adalah beberapa perintah yang akan digunakan :

- **OPEN** `lpszDeviceID lpszOpenFlags lpszFlags`
- **SET** `lpszDeviceID lpszSetting lpszFlags`
- **RECORD** `lpszDeviceID lpszRecordFlags lpszFlags`
- **STOP** `lpszDeviceID lpszStopFlags lpszFlags`
- **SAVE** `lpszDeviceID lpszSaveFlags lpszFlags`
- **CLOSE** `lpszDeviceID lpszFlags`

Perintah-perintah di atas memang terlihat sedikit rumit, tetapi pada dasarnya cukup sederhana. Parameter `lpszDeviceID` adalah piranti multimedia yang digunakan, misalnya "waveaudio", "sequencer", "cdaudio", dan sebagainya. Berikutnya, parameter `lpszFlags` dapat bernilai 'wait', 'notify', atau gabungan keduanya. Apabila diberikan nilai 'wait' maka fungsi `mciSendString` dijalankan sampai selesai, jika tidak, fungsi ini langsung kembali. Apabila diberikan nilai 'notify' berarti parameter `hwndCallback` berisi handle jendela yang akan menerima kejadian dari piranti multimedia. Contoh instruksi program pada bagian perekaman ada di bawah ini

```

MciSendString('OPEN NEW TYPE Waveaudio
ALIAS recordsound', nil, 0, 0);
MciSendString('SET recordsound TIME
FORMAT MS BITPERSAMPLE 8 '
+ 'CHANNELS 1 SAMPLEPERSEC 8000 '
+ 'BYTESPERSEC 8000', nil, 0, 0);

```

```

MciSendString('RECORD recordsound',
nil,0,0);

```

Pada bagian penghentian perekaman dan penyimpanan file, instruksi programnya adalah sebagai berikut.

```

MciSendString('STOP recordsound', nil, 0, 0);
MciSendString(PChar('SAVE recordsound "" +
SaveWAV.FileName + ""'), nil, 0, 0);
MciSendString('CLOSE recordsound', nil, 0, 0);

```

Setelah proses perekaman ini selesai, program akan mengaktifkan komponen `SaveDialog1 (SaveWav)` untuk menyimpan dan memberi nama file yang kemudian nama file tersebut akan dicantumkan pada komponen `Edit1 (FileWav)`.

## 2. Bagian Konversi

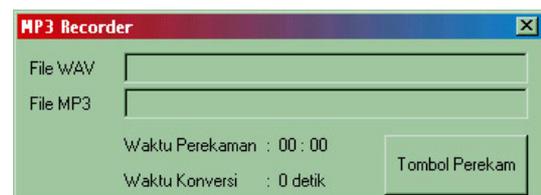
Pada bagian ini berfungsi untuk mengkonversi file WAV ke MP3, dan juga membutuhkan komponen tambahan dari luar yaitu `TACMConvertor` dan unit tambahan 'Wavs'. Program pada bagian ini cukup panjang dan kompleks.

Bagian ini diaktifkan setelah proses perekaman selesai, yaitu dengan cara mengaktifkan komponen `SaveDialog2 (SaveMP3)` setelah komponen `Edit1 (FileWav)` terisi nama file wav. Setelah mengisikan nama file MP3 pada `SaveMP3`, kemudian nama file tersebut akan dicantumkan pada komponen `Edit2 (FileMP3)`. Setelah komponen `FileMP3` terisi nama file MP3, proses kompresi mulai berjalan.

## IV. PENGUJIAN DAN ANALISA

### 4.1 PENGUJIAN PROGRAM APLIKASI.

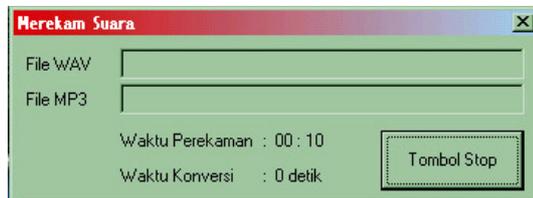
Pertama-tama nama program aplikasi diklik terlebih dahulu, kemudian tampilan program aplikasi akan muncul di layar monitor, seperti di bawah ini.



**Gambar 4.1.** Tampilan program aplikasi.

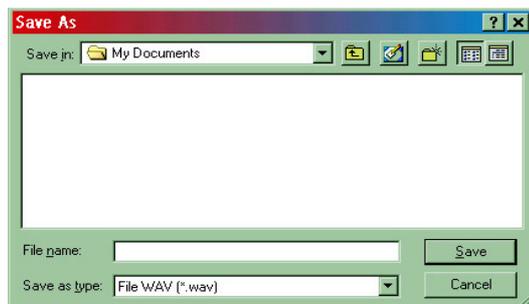
Setelah sumber suara yang akan direkam berbunyi, tombol perekam pada program aplikasi

tersebut segera ditekan kemudian proses perekaman segera berjalan dan tampilan program aplikasinya menjadi seperti berikut.



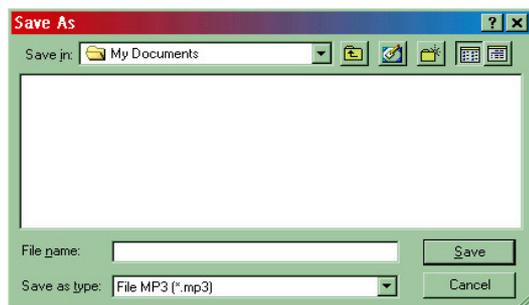
**Gambar 4.2.** Tampilan proses perekaman.

Setelah perekaman dianggap cukup maka tombol stop segera ditekan, kemudian secara otomatis muncul tampilan baru untuk memberi nama dan menyimpan file WAV, seperti di bawah ini.



**Gambar 4.3.** Tampilan untuk memberi nama file WAV.

Setelah tombol “Save” ditekan, kemudian file Wav akan secara otomatis akan disimpan di media penyimpanan. Setelah selesai menyimpan, muncul lagi tampilan yang sama untuk memberi nama file MP3, seperti dibawah ini.



**Gambar 4.4.** Tampilan untuk memberi nama file MP3.

Setelah tombol “Save” ditekan, kemudian berjalanlah proses kompresi file WAV ke MP3 dan file MP3 tersebut secara otomatis langsung tersimpan dan proses kerja program aplikasi

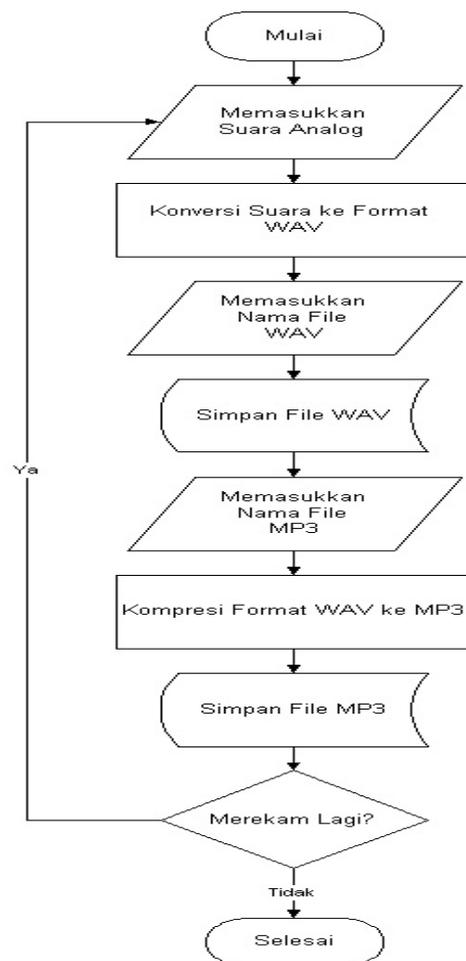
secara keseluruhan telah berhenti, seperti tampilan di bawah ini.



**Gambar 4.5.** Tampilan akhir program aplikasi.

Proses kerja program aplikasi bisa dilakukan lagi dengan menekan “Tombol Perekaman”, maka prosesnya akan sama seperti di atas. Untuk mematikan program aplikasi ini cukup menekan tombol “x” yang ada di pojok kanan atas.

Dari hasil pengujian di atas, proses kerja program aplikasi secara ringkas dapat digambarkan menjadi diagram alur di bawah ini :



**Gambar 4.6.** Diagram Alur.

#### 4.2. ANALISA DATA SUARA HASIL KERJA PROGRAM APLIKASI.

Dengan konfigurasi suara 8 bit, Mono, dan 8.000 Hz, telah diambil 10 sampel data suara yang diambil secara acak, yang terdiri atas 5 data dengan bunyi (bunyi) dan 5 data tanpa bunyi (sunyi). Data tersebut adalah sebagai berikut :

No.	Nama File	Waktu Konversi (Detik)
1	Bunyi1	18,23
2	Bunyi2	22,17
3	Bunyi3	23,82
4	Bunyi4	34,94
5	Bunyi5	42,12
6	Sunyi1	15,94
7	Sunyi2	19,48
8	Sunyi3	22,10
9	Sunyi4	33,23
10	Sunyi5	43,63

**Tabel 4.1.** Data Waktu Konversi

Untuk mengukur durasi file WAV digunakan perangkat lunak Sound Recorder (program bawaan Windows), sedangkan untuk mengukur durasi file MP3 digunakan perangkat lunak WinAmp v2.78c.

No.	Nama File WAV	Ukuran (Byte)	Durasi (Detik)	Rasio Ukuran per Durasi (Byte/ Detik)
1	Bunyi1	1.339.544	167,43	8.000,62
2	Bunyi2	1.548.044	193,50	8.000,23
3	Bunyi3	1.920.544	240,06	8.000,27
4	Bunyi4	2.772.044	346,50	8.000,13
5	Bunyi5	3.620.044	452,50	8.000,10
6	Sunyi1	1.337.044	167,12	8.000,50
7	Sunyi2	1.547.044	193,37	8.000,43
8	Sunyi3	1.916.544	239,56	8.000,27
9	Sunyi4	2.752.044	344,00	8.000,13
10	Sunyi5	3.591.044	448,87	8.000,19
<b>Rata-rata</b>				<b>8.000,29</b>

**Tabel 4.2.** Rasio antara ukuran dengan durasi file WAV

No.	Nama File MP3	Ukuran (Byte)	Durasi (Detik)	Rasio Ukuran per Durasi (Byte/ Detik)
1	Bunyi1	167.201	167	1.001,20
2	Bunyi2	193.264	193	1.001,37
3	Bunyi3	239.847	239	1.003,54
4	Bunyi4	346.262	346	1.000,76
5	Bunyi5	452.249	452	1.000,55
6	Sunyi1	166.906	166	1.005,46
7	Sunyi2	193.112	193	1.000,58
8	Sunyi3	239.342	239	1.001,43
9	Sunyi4	343.744	343	1.002,17
10	Sunyi5	448.649	448	1.001,45
<b>Rata-rata</b>				<b>1.001,92</b>

**Tabel 4.3.** Rasio antara ukuran dengan durasi file MP3

No.	Nama File	Ukuran File WAV (Byte)	Ukuran File MP3 (Byte)	Rasio Kompresi
1	Bunyi1	1.339.544	167.201	8,01
2	Bunyi2	1.548.044	193.264	8,01
3	Bunyi3	1.920.544	239.847	8,01
4	Bunyi4	2.772.044	346.262	8,01
5	Bunyi5	3.620.044	452.249	8,00
6	Sunyi1	1.337.044	166.906	8,01
7	Sunyi2	1.547.044	193.112	8,01
8	Sunyi3	1.916.544	239.342	8,01
9	Sunyi4	2.752.044	343.744	8,01
10	Sunyi5	3.591.044	448.649	8,00
<b>Rata-rata</b>				<b>8,01</b>

**Tabel 4.4.** Rasio Kompresi File MP3

Dengan konfigurasi suara 8 bit, Mono, dan 8.000 Hz diperoleh analisa, antara lain :

- Perbandingan antara ukuran dan durasi file WAV adalah 8.000 Bytes/detik.
- Perbandingan antara ukuran dan durasi file MP3 adalah 1.002 Bytes/detik.
- Perbandingan ukuran data antara file MP3 dengan file WAV adalah 1 : 8,01.
- Semakin lama waktu untuk merekam maka semakin lama juga waktu untuk mengkonversi file WAV ke MP3.
- Besar kecilnya ukuran volume suara analog, sama sekali tidak mempengaruhi ukuran data suara digital.

- Ukuran File WAV dan MP3 dipengaruhi oleh lamanya waktu perekaman.
- Lamanya waktu kompresi dipengaruhi oleh jenis perangkat keras yang dipakai, seperti prosesor dan kapasitas memori RAM.

#### 4.3. ANALISA PERBANDINGAN.

Untuk membandingkannya dengan perangkat lunak yang telah beredar secara umum, dipilihlah "Soundo'LE" ( program bawaan Sounccard ) untuk menghasilkan File data suara WAV dan "Xing MP3 Encoder" untuk mengubah menjadi File data suara MP3. Karena setingan konfigurasi suara terendah dari perangkat lunak ini adalah 8 bit, Mono, dan 16.000 Hz, sehingga hanya bisa dilakukan analisis perbandingan terhadap Rasio kompresi. Dari beberapa kali pengambilan sampel hasil dari software ini, diperoleh data sebagai berikut.

No.	Nama File	Ukuran File WAV (Byte)	Ukuran File MP3 (Byte)	Rasio Kompresi
1.	Tes1	2.728.046	341.064	8,00
2.	Tes2	3.840.046	480.024	8,00
3.	Tes3	5.440.046	680.040	8,00
4.	Tes4	7.680.046	960.048	8,00
5.	Tes5	9.600.046	1.200.024	8,00
6.	Tes6	11.554.816	1.444.032	8,00
<b>Rata-rata</b>				<b>8,00</b>

**Tabel 4.5.** Rasio Kompresi Xing MP3 Encoder

Dari data di atas diperoleh hasil bahwa rasio kompresi rata-rata "Xing MP3 Encoder" adalah 1 : 8,00. Sedangkan rasio kompresi program aplikasi adalah 1 : 8,01. Maka dari hasil analisa perbandingan ini bisa disimpulkan bahwa program aplikasi ini bisa menghasilkan file data suara MP3 yang tidak kalah dari program yang lain.

#### V. KESIMPULAN DAN SARAN

##### 5.1. KESIMPULAN.

Dari keseluruhan pembahasan bab-bab pada uraian sebelumnya maka dapat disimpulkan dari proyek tugas akhir ini, sebagai berikut :

1. Program aplikasi ini menghasilkan file data suara digital dengan format WAV dan MP3 dengan karakteristik sebagai berikut :
  - Rasio ukuran terhadap durasi file WAV adalah 8.000 Bytes/detik.
  - Rasio ukuran terhadap durasi file MP3 adalah 1.002 Bytes/detik.

- Rasio kompresi file MP3 terhadap file WAV adalah 1: 8,01.

2. Program aplikasi ini menghasilkan format mp3 dengan konfigurasi 8 bit, Mono, dan 8.000 Hz.

##### 5.2. SARAN-SARAN.

Program aplikasi untuk mengkonversi suara analog menjadi digital dengan format MP3 ini masih dimungkinkan untuk pengembangan lebih lanjut sejalan dengan perkembangan teknologi. Berikut ini saran-saran yang dapat penulis berikan :

1. Kualitas suara hasil dari program aplikasi bisa dibuat lebih bagus dengan cara :
  - Kanal dibuat variabel antara mono dan stereo.
  - Bit sampling dibuat variabel antara 8 hingga 16 bit.
  - Frekuensi sampling dibuat variabel dengan jangkauan antara 8.000 Hz hingga 48.000 Hz. Semakin besar semakin bagus.
2. Untuk mempersingkat waktu dalam pembuatan file MP3 disarankan untuk langsung mengkonversi suara analog ke digital berformat MP3 tanpa harus membuat terlebih dahulu file WAV.
3. Program aplikasi ini bisa dimodifikasi menjadi kegunaan tertentu, seperti membuat buku harian lisan.

#### DAFTAR PUSTAKA

1. Abdul Kadir, Pemrograman Dasar Turbo Pascal, Andi Offset, 1991.
2. Jogiyanto H.M., Turbo Pascal Versi 5.0, Jilid 1 dan 2, Andi Offset, 1991.
3. Antony Pranata, Pemrograman Borland Delphi 6 Edisi 4, Andi Offset, 2003.
4. Anonim, MPEG Player dengan Delphi, Mikrodata Edisi Januari 1999.
5. Anonim, Sekelumit tentang Format MP3 dan MP3 Player, Komputek Edisi 205, Maret 2001.
6. Ian Sommerville, Software Engineering, Edisi 6, Jilid 1, Erlangga, 2003.
7. ---,MP3. <http://en.wikipedia.org/wiki/mp3>, Februari 2007.



**Hapsoro Haryo Baskoro [L2F305214]**  
Mahasiswa Jurusan Teknik Elektro  
Bidang Konsentrasi Teknik Komputer  
dan Informatika.  
Universitas Diponegoro Semarang.  
Email : hapsoro\_hb@yahoo.com

Menyetujui

Dosen Pembimbing

Pembimbing I

Pembimbing II

Maman Somantri, ST, MT    Iman Santoso, ST, MT