

Perlindungan Hak Cipta Pada Data Audio Menggunakan Teknik *Watermarking Phase Coding*

Martharany Rumondang

Departemen Teknik Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132

E-mail : if12062@students.if.itb.ac.id, martharany@yahoo.com

Abstrak

Media digital telah menggantikan peran media analog dalam berbagai aplikasi. Hal ini disebabkan karena kelebihan yang dimiliki media digital. Namun, terdapat satu kelemahan dari penggunaan media digital, yaitu mengenai perlindungan hak cipta dari suatu media digital.

Digital watermarking dikembangkan sebagai salah satu jawaban untuk melindungi hak cipta suatu data digital. Dalam Tugas Akhir ini, diimplementasikan *watermarking* pada berkas MP3 dengan menggunakan teknik *watermarking phase coding*, yang menyembunyikan data dengan cara menukarkan fase asli segmen inisial dari sinyal suara dengan fase absolut dari sinyal *watermark*.

Perangkat lunak dibangun menggunakan bahasa pemrograman Java 1.5 dan IDE Netbeans 5.0. Secara umum, perangkat lunak yang dibangun cukup baik karena berkas MP3 yang dihasilkan setelah proses penyisipan *watermark* masih memiliki kualitas suara yang relatif baik, tahan terhadap operasi pencuplikan ulang, pengubahan format berkas MP3, serta pemberian derau maupun pemotongan berkas di bagian tengah dan akhir berkas.

Kata kunci : fase relatif, fase absolut, *watermark*, *phase coding*, MPEG/Audio Layer III (MP3).

1. Pendahuluan

Saat ini media digital, seperti video, audio, dan gambar, telah menggantikan peran media analog dalam berbagai aplikasi. Keberhasilan dari penerapan media digital ini disebabkan karena beberapa kelebihan yang dimiliki media digital, seperti transmisi yang bebas derau, penyimpanan yang padat, penyalinan yang sempurna, dan kemudahan untuk melakukan pengeditan. Di samping semua kelebihan yang dimiliki media digital seperti yang telah dipaparkan sebelumnya, terdapat kelemahan dari penggunaan media digital. Masalah terbesar adalah mengenai hak intelektual (hak cipta) dan kebenaran konten dari suatu media digital. Selama ini penggandaan atas produk digital tersebut dilakukan secara bebas dan leluasa. citra digital, mata tidak bisa membedakan apakah citra tersebut disisipi *watermark* atau

Pemegang hak cipta atas produk digital tersebut dirugikan karena tidak mendapat royalti dari usaha penggandaan tersebut. Untuk alasan ini, pencipta dan pendistribusi dari suatu data digital mencari solusi terhadap masalah tersebut yaitu perlindungan hak cipta dari data multimedia.

Digital *watermarking* dikembangkan sebagai salah satu jawaban untuk menentukan keabsahan pencipta atau pendistribusi suatu data digital dan integritas suatu data digital. Teknik *watermarking* bekerja dengan menyisipkan sedikit informasi yang menunjukkan kepemilikan, tujuan, atau data lain, pada media digital tanpa mempengaruhi kualitasnya. Jadi pada

tidak. Demikian pula jika diterapkan pada audio atau musik, telinga tidak bisa mendengar sisipan informasi tadi. Sehingga pada teknologi ini dikenal suatu persyaratan bahwa *watermark* haruslah *imperceptible* atau tidak terdeteksi oleh indera penglihatan (*human visual system / HVS*) atau indera pendengaran (*human auditory system / HAS*).

Metode audio *watermarking* yang sering dikaji dapat dibagi menjadi :

1. Domain waktu

Metode ini bekerja dengan cara mengubah data audio dalam *domain* waktu yang akan disisipkan *watermark*. Contohnya dengan mengubah LSB (*Least Significant Bit*) dari data tersebut. Secara umum metode ini rentan terhadap proses kompresi, transmisi dan *encoding*. Beberapa teknik algoritma yang termasuk dalam metode ini adalah:

a. *Compressed-domain watermarking*

Pada teknik ini hanya representasi data yang terkompresi yang diberi *watermark*. Saat data di *uncompressed* maka *watermark* tidak lagi tersedia.

b. *Bit dithering*

Watermark disisipkan pada tiap LSB, baik pada representasi data terkompresi atau tidak. Teknik ini membuat derau pada sinyal.

c. *Amplitude modulation*

Cara ini membuat setiap puncak sinyal dimodifikasi agar jatuh ke dalam pita-pita amplitudo yang telah ditentukan

d. *Echo hiding*

Dalam metode ini salinan-salinan terputus-putus dari sinyal dicampur dengan sinyal asli dengan rentang waktu yang cukup kecil. Rentang waktu ini cukup kecil sehingga amplitudo salinannya cukup kecil sehingga tidak terdengar.

2. Domain frekuensi

Metode ini bekerja dengan cara mengubah *spectral content* dalam *domain* frekuensi dari sinyal. Misalnya dengan cara membuang komponen frekuensi tertentu atau menambahkan

data sebagai derau dengan amplitudo rendah sehingga tidak terdengar. Beberapa teknik yang bekerja dengan metode ini:

a. *Phase coding*

Bekerja berdasarkan karakteristik sistem pendengaran manusia (*Human Auditory System*) yang mengabaikan suara yang lebih lemah jika dua suara itu datang bersamaan. Secara garis besar data *watermark* dibuat menjadi derau dengan amplitudo yang lebih lemah dibandingkan amplitudo data audio lalu digabungkan

b. *Frequency band modification*

Informasi *watermark* ditambahkan dengan cara membuang atau menyisipkan ke dalam pita-pita (*band*) *spectral* tertentu.

c. *Spread spectrum*

Dalam metode ini, sinyal yang membawa data *watermark* dimodulasikan ke dalam derau pita lebar (*wideband noise*) setelah sebelumnya di multiplikasi dengan suatu *pseudorandom sequence*.

Pada Tugas Akhir ini digunakan teknik *watermarking Phase coding* yang memanfaatkan kelemahan sistem pendengaran manusia yang secara umum tidak dapat mendengar suara pada amplitudo yang lebih lemah. Teknik ini cukup *robust* sebagai mekanisme perlindungan hak cipta pada data audio.

2. *Phase coding*

Phase coding bekerja berdasarkan karakteristik sistem pendengaran manusia yang mengabaikan suara yang lebih lemah jika dua suara itu datang bersamaan. Ide dasar dibalik teknik *phase coding* adalah menyembunyikan data dengan cara menukarkan fase asli segmen inisial dari sinyal suara dengan fase absolut dari sinyal *watermark* dengan tetap menjaga fase relatif

antara segmen sinyal menggunakan beda fase segmen dari sinyal asli. Ketika beda fase antara sinyal asli dan sinyal yang dimodifikasi besarnya kecil, maka perbedaan suara yang dihasilkan tidak terdeteksi oleh pendengaran manusia.

Phase coding termasuk dalam kelompok teknik audio *watermarking* berbasis domain frekuensi yang bekerja dengan cara mengubah *spectral content* dalam domain *phasecoding*.

2.1. Parameter

Terdapat beberapa parameter yang berpengaruh dalam teknik *watermarking phase coding*. Parameter tersebut antara lain fase relatif dan amplitudo sinyal suara, fase absolut data *watermark*, serta beda fase relatif antara segmen sinyal suara.

Phase coding didasarkan pada kenyataan bahwa pendengaran manusia lebih peka terhadap beda fase relatif dalam sinyal audio dibandingkan dengan fase absolut yang ditambahkan. Jika hubungan fase antar setiap komponen frekuensi diubah secara dramatis, akan terjadi dispersi fase yang tampak dengan jelas. Akan tetapi, selama modifikasi fase cukup kecil (tergantung pada pengamat) dan dengan menjaga beda fase relatif antara segmen-segmen sinyal suara, maka modifikasi fase yang dilakukan tidak akan terdengar.

Amplitudo sinyal suara digunakan bersama dengan nilai fase sinyal suara untuk mengubah kembali sinyal suara dari domain frekuensi menjadi domain waktu sehingga dapat didengarkan.

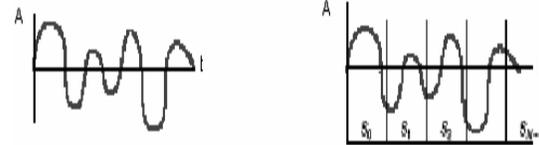
2.2. Enkoding

Langkah-langkah untuk melakukan penyisipan *watermark* dengan teknik *phase coding* adalah sebagai berikut :

1. Bagi urutan suara menjadi N segmen, $s[i]$, $0 \leq i \leq L-1$, dimana setiap segmen memiliki panjang yang sama yaitu sebesar L . (Gambar 1)

frekuensi dari sinyal. *Phase coding* merupakan metode paling efektif dari segi perbandingan *noise signal-to-perceived*.

Dalam pembahasan berikutnya akan dijelaskan mengenai parameter apa saja yang mempengaruhi *phase coding*, bagaimana cara menyisipkan *watermark* ke dalam data audio (enkoding), dan cara pengekstraksian *watermark* dari data audio yang telah disisipi (dekoding) dengan teknik



Gambar 1 Sinyal asli dan sinyal yang dibagi menjadi beberapa segmen

2. Hitung DFT (dalam hal ini menggunakan FFT) pada masing-masing segmen. Hasil dari perhitungan ini adalah berupa $X(k)$ untuk masing-masing segmen dimana $0 \leq k \leq L-1$.
3. Hitung nilai fase ϕ dan amplitudo A untuk tiap-tiap segmen dimana a_k adalah bagian *real* dari nilai FFT dan b_k adalah bagian *imaginer*-nya. Hasil yang diperoleh untuk tiap segmennya akan tampak seperti pada Gambar 2.

$$|A| = \sqrt{a_k^2 + b_k^2} \quad (2.1)$$

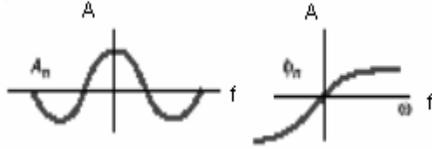
$$\phi = \tan^{-1}\left(\frac{b_k}{a_k}\right) \quad (2.2)$$

4. Setelah itu hitung beda fase antara fase segmen yang berdekatan.

$$\Delta\phi_{N-1}(\omega_k) = \phi_{N-1}(\omega_k) - \phi_N(\omega_k) \quad (2.3)$$

5. Fase absolut dari sinyal data *watermark* ditambahkan ke dalam beda fase yang dihasilkan. Sinyal *watermark* dengan panjang Lw , $w[j]$, $0 \leq j \leq Lw-1$, disajikan sebagai $\Phi_{data} = \pi/2$ atau $\Phi_{data} = -\pi/2$ yang merepresentasikan bit 1 atau

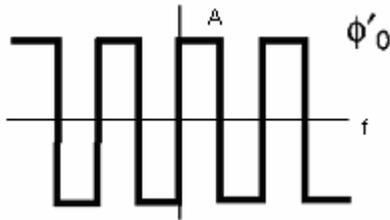
0. Hasil yang diperoleh akan tampak seperti Gambar 3.



Gambar 2 Magnitudo dan plot fase dari FFT

6. Substitusikan fase segmen awal Φ'_0 dengan fase sinyal *watermark* Φ'_{data} (Gambar 4)

$$\Phi'_0 = \Phi'_{data} \quad (2.4)$$

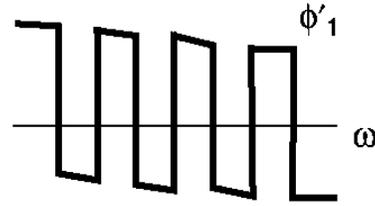


Gambar 3 Fase absolut dari sinyal data *watermark*

7. Buat matriks fase untuk $N > 0$ dengan menggunakan beda fase untuk menjaga relativitas fase antara segmen suara. Hal ini dilakukan untuk menjaga kesinambungan sinyal setelah proses modifikasi fase segmen awal.

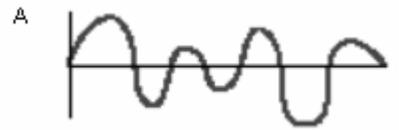
$$\begin{bmatrix} (\phi'_1(\omega_k) = \phi'_0(\omega_k) + \Delta\phi_1(\omega_k)) \\ \dots \\ (\phi'_n(\omega_k) = \phi'_{n-1}(\omega_k) + \Delta\phi_n(\omega_k)) \\ \dots \\ (\phi'_N(\omega_k) = \phi'_{N-1}(\omega_k) + \Delta\phi_N(\omega_k)) \end{bmatrix}$$

8. Gabungkan segmen-segmen yang telah dimodifikasi fasenya tersebut menjadi satu.



Gambar 4 Menambahkan fase absolut *watermark* ke beda fase sinyal asli

9. Hitung kembali nilai fase yang baru dan nilai amplitudo yang sudah dihitung sebelumnya untuk melakukan *inverse* DFT terhadap masing-masing segmen untuk mengembalikan sinyal ke domain waktu (Gambar 5).



Gambar 5 Sinyal *watermarking* yang dihasilkan dengan teknik *phase coding*

2.3. Dekoding

Dekoding dilakukan dengan melakukan sinkronisasi terhadap proses encoding. Panjang segmen dan panjang *watermark* harus diketahui. Proses dekoding memerlukan sinyal suara asli untuk melakukan pendeteksian. Langkah-langkah pendeteksian *watermark* yang dilakukan adalah sebagai berikut :

1. Ambil n bagian pertama dari sinyal suara dimana n adalah panjang segmen encoding yang diketahui, $s[0] \dots s[n-1]$. Perhitungan hanya dilakukan terhadap n elemen pertama sinyal karena data *watermark* disisipkan hanya di segmen awal sinyal suara asli.
2. Lakukan FFT terhadap $n-1$ sinyal tersebut, kemudian cari nilai fasenya Φ sesuai persamaan 2.2.
3. Konversi nilai fase yang didapatkan, $\pi/2$ menjadi bit 1 dan $-\pi/2$ menjadi bit 0 sebanyak panjang bit *watermark*. Nilai fase dibandingkan dengan nilai fase sinyal suara asli.

4. Didapatkan data bit-bit *watermark* sesuai dengan hasil konversi. Bit-bit *watermark* dibandingkan dengan bit-bit *watermark* asli untuk mengetahui kebenarannya.

3. Deskripsi Umum Perangkat Lunak

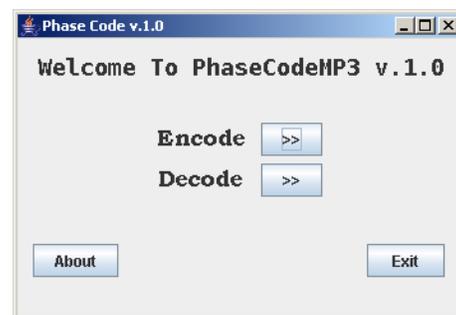
Perangkat lunak ini diberi nama PhaseCodeMP3. Perangkat lunak PhaseCodeMP3 memiliki dua fungsionalitas utama yaitu fungsi penyisipan *watermark* (enkoding) dan fungsi pengekstraksian *watermark* (dekoding). Dalam proses enkoding dan dekoding diperlukan masukan berupa berkas MP3 dan file *watermark* berformat teks. Enkoding dilakukan dengan menyisipkan bit *watermark* ke dalam sinyal berkas MP3 dengan modifikasi fase, sedangkan dekoding dilakukan dengan mengekstraksi fase yang didapat untuk selanjutnya dikonversi menjadi bentuk bit.

Perangkat lunak PhaseCodeMP3 memiliki kelas-kelas sebagai berikut :

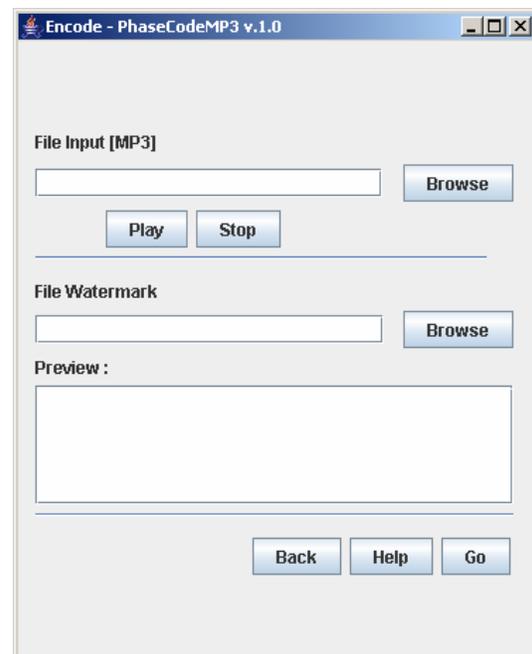
1. *Encode*
Kelas ini berfungsi untuk melakukan proses penyisipan *watermark* ke dalam berkas media MP3 yang telah dipilih pengguna
2. *Decode*
Kelas ini berfungsi untuk melakukan proses ekstraksi *watermark* dari berkas media penampung
3. MP3
Kelas ini berfungsi untuk membaca dan menyimpan berkas MP3
4. *Myplayer*
Kelas ini berfungsi untuk memutar berkas MP3
5. GUI (*EncodeGUI* dan *DecodeGUI*)
Kelas ini berfungsi untuk mempermudah interaksi pengguna dengan perangkat lunak PhaseCodeMP3 ini.
6. FFT
Kelas ini berfungsi untuk melakukan penghitungan FFT maupun penghitungan *invers* FFT dalam pemrosesan sinyal berkas MP3 yang akan dimanipulasi.

4. Implementasi Aplikasi

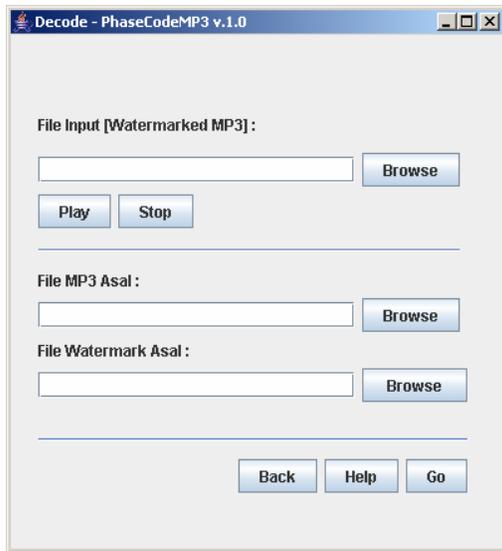
Lingkungan yang digunakan untuk membangun perangkat lunak PhaseCodeMP3 adalah lingkungan berbasis Windows. Bahasa pemrograman yang digunakan untuk membangun PhaseCodeMP3 adalah Java versi 1.5. Sedangkan kompilator sekaligus IDE yang digunakan untuk memudahkan pengembangan perangkat lunak adalah NetBeans 5.0. Gambar 6-8 memperlihatkan antarmuka dari PhaseCodeMP3.



Gambar 6 Layar Saji Utama



Gambar 7 Layar Saji Enkoding



Gambar 8 Layar Saji Dekoding

5. Pengujian

Pengujian perangkat lunak PhaseCodeMP3 dilakukan dengan menggunakan perangkat keras yang memiliki spesifikasi sebagai berikut :

1. CPU : Intel Pentium IV 3 GHz
2. HD : 80GB
3. Memori : 512 MB DDRAM

Rancangan pengujian yang akan dilakukan meliputi pengujian terhadap tiga hal, yaitu :

1. Kebenaran perangkat lunak, yang diukur dari kesesuaian *watermark* yang disisipkan dengan *watermark* hasil ekstraksi. Besarnya kesesuaian dihitung dalam persentase dengan membandingkan jumlah bit *watermark* yang berhasil diekstraksi dengan jumlah bit *watermark* yang asli.
2. Kinerja perangkat lunak yang diukur dari kecepatan penyisipan maupun pengeksktraksian *watermark*. Pengujian ini dilakukan untuk mengetahui seberapa besar kinerja perangkat lunak yang dibangun
3. Ketahanan data terhadap pemrosesan berkas MP3. Pengujian dilakukan untuk mengetahui sejauh mana kesesuaian bit *watermark* hasil ekstraksi dengan bit

watermark asli, jika berkas MP3 dimanipulasi.

Tabel 1-8 memperlihatkan hasil pengujian yang dilakukan.

No.	Nama Berkas	Tipe	Ukuran
1	Chris Brown - Run It.mp3	Stereo	156 KB
2	Hips don't lie.mp3	Stereo	451 KB
3	Scooter-Maria(I Like it Loud).mp3	Stereo	412 KB
4	Sonar ping.mp3	Stereo	18 KB
5	SSEYO MIXA1 ex1.mp3	Mono	57 KB
6	Paris Hilton - Stars Are Blind.mp3	Mono	207 KB
7	Diam's - Jeune Demoiselle.mp3	Mono	403 KB
8	Hilary Duff Wake Up.mp3	Mono	412 KB

Tabel 1 Berkas MP3 yang diuji

No.	Nama Berkas	Ukuran
1	Copyright.txt	18 bytes
2	ID3.txt	118 bytes

Tabel 2 Berkas *watermark* yang diuji

No.	Proses	Hasil
1	Pemutaran berkas MP3 sebelum disisipi <i>watermark</i>	Berhasil
2	Penyisipan sebuah <i>watermark</i> (.txt) ke dalam sebuah berkas MP3	Berhasil
3	Pemutaran berkas MP3 yang telah disisipi <i>watermark</i>	Berhasil
4	Ekstraksi <i>watermark</i> dari berkas MP3 yang telah disisipi <i>watermark</i>	Berhasil

Tabel 3 Hasil pengujian kebenaran perangkat lunak

No.	Sampling rate asal	Sampling rate baru	Hasil Ekstraksi
1.1	44.1 KHz	48 KHz	Baik
1.2	44.1 KHz	48 KHz	Baik
2.1	44.1 KHz	48 KHz	Baik
2.2	44.1 KHz	48 KHz	Baik
3.1	44.1 KHz	48 KHz	Baik
3.2	44.1 KHz	48 KHz	Baik
4.1	44.1 KHz	48 KHz	Baik
4.2	44.1 KHz	48 KHz	Baik

No.	Sampling rate asal	Sampling rate baru	Hasil Ekstraksi
5.1	44.1 KHz	48 KHz	Baik
5.2	44.1 KHz	48 KHz	Baik
6.1	44.1 KHz	48 KHz	Baik
6.2	44.1 KHz	48 KHz	Baik
7.1	44.1 KHz	48 KHz	Baik
7.2	44.1 KHz	48 KHz	Baik
8.1	44.1 KHz	48 KHz	Baik
8.2	44.1 KHz	48 KHz	Baik

Tabel 4 Hasil pengujian ketahanan - resampling

No.	Hasil ekstraksi terhadap pemotongan bagian berkas MP3	
	Sepertiga bagian tengah	Bagian akhir
1.1	Baik	Baik
1.2	Baik	Baik
2.1	Baik	Baik
2.2	Baik	Baik
3.1	Baik	Baik
3.2	Baik	Baik
4.1	Baik	Baik
4.2	Baik	Baik
5.1	Baik	Baik
5.2	Baik	Baik
6.1	Baik	Baik
6.2	Baik	Baik
7.1	Baik	Baik
7.2	Baik	Baik
8.1	Baik	Baik
8.2	Baik	Baik

Tabel 5 Hasil pengujian ketahanan - cropping

No.	Hasil Pengujian terhadap pemberian derau pada bagian berkas MP3		
	Bagian awal	Bagian tengah	Bagian akhir
1.1	Tidak baik	Baik	Baik
1.2	Tidak baik	Baik	Baik
2.1	Tidak baik	Baik	Baik
2.2	Tidak baik	Baik	Baik
3.1	Tidak baik	Baik	Baik
3.2	Tidak baik	Baik	Baik
4.1	Tidak baik	Baik	Baik
4.2	Tidak baik	Baik	Baik
5.1	Tidak baik	Baik	Baik
5.2	Tidak baik	Baik	Baik
6.1	Tidak baik	Baik	Baik
6.2	Tidak baik	Baik	Baik
7.1	Tidak baik	Baik	Baik
7.2	Tidak baik	Baik	Baik
8.1	Tidak baik	Baik	Baik
8.2	Tidak baik	Baik	Baik

Tabel 6 Hasil pengujian ketahanan - pemberian derau

No	Hasil pemutaran berkas MP3	Hasil ekstraksi berkas MP3
1.1	Baik	Baik
1.2	Baik	Baik
2.1	Baik	Baik
2.2	Baik	Baik
3.1	Baik	Baik
3.2	Baik	Baik
4.1	Baik	Baik
4.2	Baik	Baik
5.1	Ada derau	Baik
5.2	Ada derau	Baik
6.1	Baik	Baik
6.2	Baik	Baik
7.1	Baik	Baik
7.2	Baik	Baik
8.1	Baik	Baik
8.2	Baik	Baik

Tabel 7 Hasil pengujian ketahanan - perubahan format

No.	t ₁ (ms)	t ₂ (ms)	x ₁ (dB)	x ₂ (dB)	Kualitas Suara	
					Subjektif	PSNR (dB)
1.1	1828	3375	-12.25	-12.4	Baik	38.35
1.2	1984	3688	-12.25	-12.4	Baik	38.35
2.1	2734	5438	-16.38	-16.455	Baik	46.82
2.2	3500	9641	-16.38	-16.455	Baik	46.82
3.1	2578	4907	-11.315	-11.45	Baik	38.57
3.2	2765	4953	-11.315	-11.45	Baik	38.57
4.1	985	2000	-17.36	-17.79	Baik	32.33
4.2	1046	2172	-17.36	-17.79	Baik	32.33
5.1	1157	1641	-12.47	-12.76	Ada derau	32.87
5.2	1219	2250	-12.47	-12.76	Ada derau	32.87
6.1	1438	3296	-8.79	-8.96	Baik	34.44
6.2	1469	3266	-8.79	-8.96	Baik	34.44
7.1	2063	3609	-10.66	-10.81	Baik	37.15
7.2	2187	3703	-10.66	-10.81	Baik	37.15
8.1	2141	4547	-9.12	-9.31	Baik	33.8
8.2	2203	5438	-9.12	-9.31	Baik	33.8

Tabel 8 Hasil pengujian kinerja perangkat lunak dimana t₁ : waktu encoding, t₂ : waktu dekode, x₁ : kekuatan sinyal awal, x₂ : kekuatan sinyal setelah penyisipan)

6. Kesimpulan dan Saran

6.1. Kesimpulan

Kesimpulan yang dapat diambil setelah mengerjakan Tugas Akhir ini adalah :

1. Teknik *phase coding* merupakan teknik yang cukup *robust* dalam penyisipan *watermark* ke dalam suatu berkas MP3 karena teknik ini tahan terhadap proses pencuplikan ulang, pemotongan berkas MP3 (selain bagian awal berkas), pemberian derau (selain bagian awal berkas), dan kompresi (pengubahan format berkas).
2. Kunci untuk melakukan proses dekoding dari teknik *phase coding* ini adalah *watermark* itu sendiri
3. Kualitas suara yang dihasilkan oleh berkas MP3 yang telah disisipi *watermark* dengan teknik *phase coding* cukup baik (hampir tidak terdeteksi adanya derau).
4. Ukuran *watermark* yang dapat disisipkan dengan teknik *phase coding* relatif lebih kecil dibandingkan dengan teknik *audio watermarking* lainnya. Hal ini disebabkan karena ukuran *watermark* yang disisipkan tidak boleh lebih dari panjang segmen pembagian suara dan modifikasi fase yang semakin banyak dapat menyebabkan derau di awal sinyal suara.
5. Kelemahan teknik ini adalah jika dilakukan pemotongan atau pemberian derau pada bagian awal berkas MP3 yang disisipi *watermark*, maka *watermark* dapat hilang atau tidak dapat diekstraksi dengan baik.

6.2. Saran

Beberapa saran untuk pengembangan Tugas Akhir ini antara lain :

1. Dalam pengembangan perangkat lunak PhaseCodeMP3, hanya dapat menyisipkan *watermark* pada berkas MP3 dengan ukuran maksimal 500 KB (kurang lebih seukuran berkas *ringtone*). Untuk pengembangan lebih lanjut, sangat disarankan untuk memodifikasi pemakaian memori sehingga perangkat lunak ini dapat melakukan penyisipan pada berbagai ukuran berkas MP3.
2. Dalam pengembangan perangkat lunak PhaseCodeMP3, hanya menyisipkan *watermark* berupa berkas teks (.txt). Untuk pengembangan lebih lanjut, sebaiknya tipe berkas *watermark* yang disisipkan dapat lebih bervariasi.
3. Dalam Tugas Akhir ini, data audio yang didukung hanyalah yang memiliki format MP3. Untuk pengembangan lebih lanjut, perangkat lunak dapat dibuat sehingga mendukung *watermarking* pada format data audio lain, seperti WAV, MIDI, dan lain sebagainya.
4. Perangkat lunak PhaseCodeMP3 masih kurang dalam memfasilitasi perlindungan hak cipta data audio, seperti pembangkitan informasi kepemilikan (jika diperlukan) dan penanganan kewenangan pihak yang menyisipkan serta pihak yang mengekstraksi data audio tersebut. Untuk pengembangan lebih lanjut, dapat dilakukan perumusan skenario untuk memfasilitasi perlindungan hak cipta data audio menggunakan teknik *watermarking phase coding*.

7. Daftar Pustaka

- [BAL06] Baldwin, Richard G. 2006. *Fun with Java, Understanding the Fast Fourier Transform (FFT) Algorithm*.
URL <http://www.developer.com>
Tanggal akses : 31 Agustus 2006
- [HAC00] Hacker, Scot. 2000. *MP3 : The Definitive Guide*. O'Reilly.

- [LEE00] Lee, Edward A., Pravin Varaiya (2000). *Structure and Interpretation of Signals and Systems*. Addison Wesley.
- [MOR96] Morimoto, N. Bender, W. Gruhl, D. Lu, A. (1996). *Techniques For Data Hiding*. IBM Systems Journal Vol. 35, No. 3&4, MIT Media Lab. G321-5608.
- [MUN04] Munir, Rinaldi. 2004. *Kuliah IF5054 Kriptografi : Steganografi dan Watermarking*.
- [EKL00] Eklund, Roberta. *Audio Watermarking Techniques*.
URL <http://www.musemagic.com>
Tanggal akses : 1 Agustus 2006
- [SMI06] Smith, Steven W. 2006. *How The FFT Works*.
URL <http://www.dspguide.com>
Tanggal akses : 2 September 2006
- [TJO05] Tjokronegoro, Harijono A. 2005. *Pengolahan Sinyal*. Bandung : ITB.
- [WIL05] Wiley Interscience (2005). *Digital Signal Processing and Applications with the C6713 and C6416 DSK*. ISBN 0471690074 - 543s - DDU
- [WIN05] Wintarsih, Endah. 2005. *Robust Dan Non-Blind Watermarking Pada Berkas MPEG/AUDIO Layer III (MP3) Dengan Teknik Echo Data Hiding*. Bandung : ITB.