

Bab V

Kesimpulan dan Saran

5.1. Pengantar

Pada bab ini berisi kesimpulan dan saran yang diambil dari hasil analisis, perancangan, implementasi, dan pengujian aplikasi SSAWW.

5.2. Kesimpulan

Berdasarkan analisis dan implementasi sistem, maka dapat ditarik kesimpulan sebagai berikut :

1. Aplikasi SSAWW yang bertujuan untuk memberikan *watermarking* (tanda air) dengan menggunakan teknik *Spread Spectrum* berhasil didesain dan diimplementasikan. Aplikasi SSAWW dikembangkan menggunakan perangkat lunak Microsoft Visual Basic 6.0.
2. Kinerja aplikasi SSAWW cukup baik, terbukti dengan hubungan antara besar ukuran file berbanding lurus dengan lama waktu komputasi.
3. File ter-*watermark* tahan terhadap *attack* yang tidak merubah jumlah sampel tetapi tidak tahan terhadap *attack* yang merubah jumlah sampel.

5.3. Saran

Dunia *watermarking* masih sangat terbuka untuk dikembangkan, aplikasi SSAWW salah satu contoh aplikasi yang menerapkan *watermarking* pada audio wave. Aplikasi ini bisa berguna bagi dunia industri, seperti : industri rekaman yang menggunakan aplikasi ini untuk memberikan label hak cipta, industri periklanan radio yang menggunakan aplikasi ini untuk menghitung iklan, dll. Dilihat dari kegunaannya aplikasi ini masih mungkin untuk dikembangkan, berikut ini adalah saran-saran untuk mengembangkan aplikasi ini agar menjadi lebih baik, yaitu :

1. Memperbaiki metode *pseudo random* agar data teracak lebih bagus.
2. File *host* yang digunakan tidak hanya file wave saja tetapi bisa juga file Mp3, Aac, Wma, dll.
3. File *hidden* yang digunakan tidak hanya file wave dan *bitmap* saja tetapi bisa juga file yang sudah terkena kompresi, misal : mp3, jpeg, dll.
4. Menambahkan metode enkripsi seperti MD5, RSA, RC5 untuk mengenkripsi file yang akan *dihidden*.
5. Menggunakan teknik *watermarking* yang lebih baik supaya *noise* pada file *ter-watermark* bisa lebih berkurang/tidak terdengar dan hasilnya tahan terhadap *attack*.

Daftar Pustaka

- Bender W., Gruhl D., Morimoto N., Lu A., 1996, *Techniques for Data Hiding*.
- Bender W., Paiz F. J., Butera W., Pogreb S., Gruhl D., Hwang R., 2000, *Applications for data hiding*.
- Cummins. Jonathan, Diskin Patrick, Lau Samuel, Parlett Robert, Ryan Mark, 2004. *Steganography And Digital Watermarking*.
- Cvejic Nedeljko, Seppanen Tapio, 2003, *Spread spectrum audio watermarking using frequency hopping and attack characterization*.
- Kirovski Darko, Malvar Henrique S, 2002, *Spread-Spectrum Watermarking of Audio Signals*.
- Supangkat Suhono H., Kuspriyanto, Juanda, 2000, *Watermarking sebagai Teknik Penyembunyian Label Hak Cipta pada Data Digital*.
- Trape Wade, Washington Lawrence C., 2002, *Pseudo random generator*.
- Digital Watermarking Teknologi Pelindung HAKI Multimedia, www.elektroindonesia.com/elektro/ut34.htm.
- Wave Files - The Sonic Spot, www.sonicspot.com/guide/wavefiles.html.
- WAVE File Format, www.borg.com/~jglatt/tech/wave.html.

LAMPIRAN

SKPL

Spesifikasi Kebutuhan Perangkat Lunak

Watermarking Pada Audio Wave Dengan Menggunakan Teknik Spread Spectrum (SSAWW)

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Weiyy Seto Aji Wibowo / 3058

Program Studi Teknik Informatika – Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		<i>SKPL-SSAWW</i>		<i>1/15</i>
		Revisi		<i>Tgl : 10-10-2006</i>

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Program Studi Teknik Informatika	SKPL-SSAWW	2 / 15
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan	6
1.1	Tujuan.....	6
1.2	Lingkup Masalah.....	6
1.3	Definisi, Akronim dan Singkatan	6
1.4	Referensi.....	7
1.5	Deskripsi umum (Overview)	7
2	Deskripsi Kebutuhan.....	8
2.1	Perspektif Produk.....	8
2.2	Fungsi Produk.....	8
2.3	Karakteristik Pengguna	9
2.4	Batasan-batasan.....	10
3	Kebutuhan Khusus.....	10
3.1	Kebutuhan Antarmuka Eksternal.....	10
3.2	Kebutuhan Antarmuka Pengguna	10
3.3	Kebutuhan Antarmuka Perangkat Keras	10
3.4	Kebutuhan Antarmuka Perangkat Lunak	11
4	Kebutuhan Fungsionalitas.....	11
4.1	Aliran Informasi.....	11
4.1.1	DFD Level 0.....	11
4.1.1.1	Entitas.....	11
4.1.1.2	Proses.....	11
4.1.1.3	Topologi.....	12
4.1.2	DFD Level 1.....	12
4.1.2.1	Entitas.....	12
4.1.2.2	Proses.....	12
4.1.2.3	Topologi.....	12
4.1.3	DFD Level 2 Proses <i>Encode</i>	13
4.1.3.1	Entitas.....	13
4.1.3.2	Proses.....	13
4.1.3.3	Topologi.....	14
4.1.4	DFD Level 2 Proses <i>Decode</i>	14
4.1.4.1	Entitas.....	14
4.1.4.2	Proses.....	14
4.1.4.3	Topologi.....	15

Daftar Gambar

Gambar 1. DFD Level 0.	12
Gambar 2. DFD Level 1.	12
Gambar 3. DFD Level 2 Proses Encode.....	14
Gambar 4. DFD Level 2 Proses Decode.....	15

1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SSAWW yang akan dikembangkan. Dokumen ini digunakan untuk mendefinisikan kebutuhan perangkat lunak, yang meliputi antarmuka eksternal, dan atribut, serta mendefinisikan fungsi perangkat lunak, juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat lunak SSAWW dikembangkan dengan tujuan untuk menyisipkan *watermarking* (tanda air) yang berupa bukti kepemilikan pada file audio dengan format *wave* (*.wav). *Watermarking* yang disisipkan bisa berupa file citra (*.bmp) atau file audio (*.wav). Teknik yang digunakan adalah teknik *Spread Spectrum*, dengan teknik ini data akan disebar didalam kawasan frekuensi secara acak.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-SSAWW-XX	Kode yang merepresentasikan kebutuhan pada SSAWW .
DFD	Data Flow Diagram merupakan teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan saat data bergerak dari input menjadi output.
SSAWW	Perangkat lunak yang digunakan untuk menyisipkan data/file kedalam audio yang berformat wav. Teknik yang digunakan dalam proses penyisipan adalah Spread Spectrum.

Program Studi Teknik Informatika	SKPL-SSAWW	6 / 15
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. *Sistem Informasi: Teori, Metodologi dan Tool*, Paulus Mudjihartono, Penerbitan Universitas Atma Jaya, Yogyakarta.
2. *Diktat Mata Kuliah: Basis Data*, RJB. Wahyu Agung., ST., MT., Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta.

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL tersebut terbagi atas 3 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak SSAWW yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SSAWW tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SSAWW yang akan dikembangkan.

2 Deskripsi Kebutuhan

2.1 Perspektif Produk

SSAWW adalah perangkat lunak yang berfungsi untuk memproteksi file audio dengan format *Wave* (*.wav) dengan memberikan *watermarking*. Prinsip dari perangkat lunak ini adalah memberikan sedikit informasi yang berupa bukti kepemilikan pada audio tanpa mempengaruhi kualitas suaranya, sehingga orang lain tidak akan curiga bahwa file audio tersebut sudah disisipi *watermarking*.

Pengguna berinteraksi dengan perangkat lunak ini melalui antarmuka yang disusun menggunakan Microsoft Visual Basic 6.0 yang dapat berjalan pada sistem operasi windows 9x, ME, 2000, dan XP. Secara garis besar proses pada perangkat lunak ini dibagi menjadi dua, yaitu : proses *Encode* dan proses *Decode*.

Proses *Encode* diawali dengan memasukkan file *host* audio yang berformat wav. Kemudian pengguna diminta untuk memasukkan file yang akan disisipkan (*hidden*), file *hidden* bisa berbentuk citra (*.bmp) atau audio (*.wav). Setelah itu pengguna diminta untuk memasukkan *password* dan konfirmasinya. Jika *password* memenuhi syarat maka sistem akan memprosesnya dengan menyisipkan file *hidden* kedalam file *host*.

Proses *Decode* diawali dengan memasukkan file audio yang telah diberi *watermark*. Kemudian pengguna diminta untuk memasukkan *password* yang sama seperti pada proses *Encode* dan sistem akan memprosesnya dengan mengekstrak file *hidden* yang ada. Jika file *hidden* ada maka sistem akan menampilkan file *hidden* tersebut.

2.2 Fungsi Produk

Fungsi produk yang ada pada perangkat lunak SSAWW adalah sebagai berikut :

1. Fungsi *Encode* (SKPL-SSAWW-01) adalah fungsi yang digunakan untuk menyisipkan file *hidden* kedalam file *host* audio.
 - 1.1. Fungsi *Open File Host* (SKPL-SSAWW-01-01) adalah fungsi yang digunakan untuk membuka file *host*.

Program Studi Teknik Informatika	SKPL-SSAWW	8 / 15
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- 1.2. Fungsi *Open File Hidden* (SKPL-SSAWW-01-02) adalah fungsi yang digunakan untuk membuka file *hidden*.
- 1.3. Fungsi *Input Password* (SKPL-SSAWW-01-03) adalah fungsi yang digunakan untuk mengecek kualitas *password* yang dimasukkan oleh user.
- 1.4. Fungsi *Watermark Embedding* (SKPL-SSAWW-01-04) adalah fungsi yang digunakan untuk memproses penyisipan data *hidden* kedalam kawasan frekuensi dari file *host* secara acak.
2. Fungsi *Decode* (SKPL-SSAWW-02) adalah fungsi yang digunakan untuk mengekstrak kembali file *hidden* yang disisipkan kedalam file *host*.
 - 2.1. Fungsi *Open File Watermark* (SKPL-SSAWW-02-01) adalah fungsi yang digunakan untuk membuka file ter-*watermark*.
 - 2.2. Fungsi *Input Password* (SKPL-SSAWW-02-02) adalah fungsi yang digunakan untuk memasukkan *password* pada proses *Decode*.
 - 2.3. Fungsi *Watermark Extracting* (SKPL-SSAWW-02-03) adalah fungsi yang digunakan untuk membaca kembali data *hidden* yang disisipkan kedalam kawasan frekuensi.
3. Fungsi *Help* (SKPL-SSAWW-03) adalah fungsi yang digunakan untuk menampilkan informasi deskripsi/keterangan dari perangkat lunak SSAWW dan memberikan petunjuk-petunjuk penggunaan perangkat lunak ini.

2.3 Karakteristik Pengguna

Pengguna dari perangkat lunak ini adalah pengguna yang ingin memproteksi file audio *wave* dengan menyisipkan bukti kepemilikan kedalam file tersebut. Pengguna perangkat lunak SSAWW mempunyai karakteristik, sebagai berikut :

1. Memahami pengoperasian komputer.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SSAWW ini adalah :

1. Keterbatasan Perangkat Keras

Kecepatan dalam memproses *Encode* dan *Decode* tergantung dari perangkat keras yang digunakan.

2. Keterbatasan Ukuran File

Ukuran file *host* dan file *hidden* mempengaruhi proses *Encode* dan *Decode*. Semakin besar ukuran file *host* dan file *hidden* maka proses akan semakin lama demikian sebaliknya semakin kecil ukuran file *host* dan file *hidden* maka proses akan semakin cepat.

3 Kebutuhan Khusus

3.1 Kebutuhan Antarmuka Eksternal

Kebutuhan antarmuka eksternal pada perangkat lunak SSAWW mencakup kebutuhan antarmuka pengguna, antarmuka perangkat keras (*hardware*), dan antarmuka perangkat lunak (*software*).

3.2 Kebutuhan Antarmuka Pengguna

Pengguna berinteraksi dengan perangkat lunak SSAWW menggunakan GUI (*Graphical User Interface*). Sebagai sarana masukan menggunakan *mouse* dan *keyboard* dan perangkat keluarannya menggunakan monitor untuk *graphical display* dan *speaker* untuk suaranya.

3.3 Kebutuhan Antarmuka Perangkat Keras

Perangkat lunak SSAWW menggunakan perangkat keras sebagai berikut :

1. *CPU*, digunakan sebagai tempat untuk mengolah proses yang dilakukan pada perangkat lunak SSAWW dan menyimpan hasilnya.

2. *Keyboard*, digunakan untuk memasukkan input data berupa karakter atau text yang harus diinputkan oleh user.
3. *Mouse*, digunakan untuk melakukan input yang berupa *event click, drag and drop* dan *on focus*.
4. *Monitor*, digunakan untuk menampilkan antarmuka perangkat lunak.
5. *Speaker*, digunakan sebagai piranti output suara.

3.4 Kebutuhan Antarmuka Perangkat Lunak

Perangkat lunak yang dibutuhkan oleh perangkat lunak SSAWW adalah sebagai berikut :

1. Sistem operasi : Microsoft Windows 9x, 2000, ME atau XP.
2. Aplikasi pengembang : Microsoft Visual Basic 6.0.

4 Kebutuhan Fungsionalitas

4.1 Aliran Informasi

4.1.1 DFD Level 0

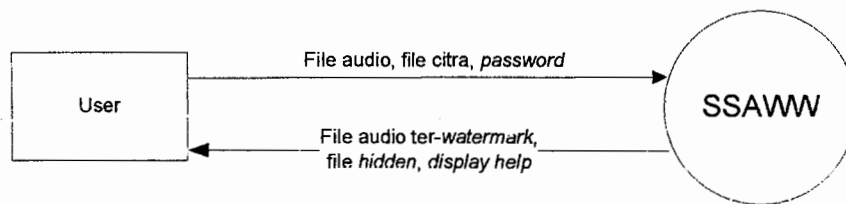
4.1.1.1 Entitas

1. *User*

4.1.1.2 Proses

Aplikasi SSAWW adalah aplikasi audio *watermarking* yang menggunakan teknik *spread spectrum*. Aplikasi ini digunakan untuk menyisipkan bukti kepemilikan yang berupa file audio (*.wav) atau citra (*.bmp) kedalam file audio (*.wav), atau membaca kembali file yang disisipkan sebelumnya. Secara garis besar aplikasi ini dibagi menjadi tiga bagian, yaitu : *Encode*, *Decode* dan *Help*.

4.1.1.3 Topologi



Gambar 1. DFD Level 0

4.1.2 DFD Level 1

4.1.2.1 Entitas

1. *User*

4.1.2.2 Proses

1. *Encode*

Pada proses ini *user* akan menginputkan file audio (*.wav) yang akan digunakan sebagai *host*, file audio (*.wav) atau citra (*.bmp) yang akan digunakan sebagai file *hidden*, dan *password*. Proses ini akan menyisipkan file *hidden* kedalam file *host*. Hasil keluaran dari proses ini adalah file audio *ter-watermark*.

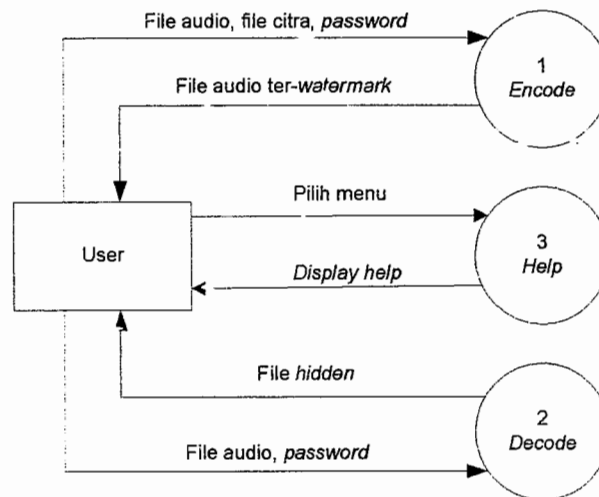
2. *Decode*

Pada proses ini *user* akan menginputkan file audio (*.wav) *ter-watermark* dan *password*. Proses ini akan membaca kembali file yang telah disisipkan sebelumnya didalam file *host*. Hasil keluaran dari proses ini adalah file *hidden*.

3. *Help*

Adalah proses untuk menampilkan informasi deskripsi dari aplikasi SSAWW dan memberikan pilihan bantuan kepada *user* yang mengalami kesulitan saat menjalankan aplikasi SSAWW.

4.1.2.3 Topologi



Gambar 2. DFD Level 1

4.1.3 DFD Level 2 Proses *Encode*

4.1.3.1 Entitas

1. *User*

4.1.3.2 Proses

1. *Open File Host*

Proses ini digunakan untuk membuka file audio yang akan digunakan sebagai *host*. File yang bisa dimasukkan hanya file audio dengan format *wave* (*.wav).

2. *Open File Hidden*

Proses ini digunakan untuk membuka file yang akan digunakan sebagai file *hidden*. File yang bisa dimasukkan hanya file dengan format *wave* (*.wav) atau *bitmap* (*.bmp).

3. *Input Password*

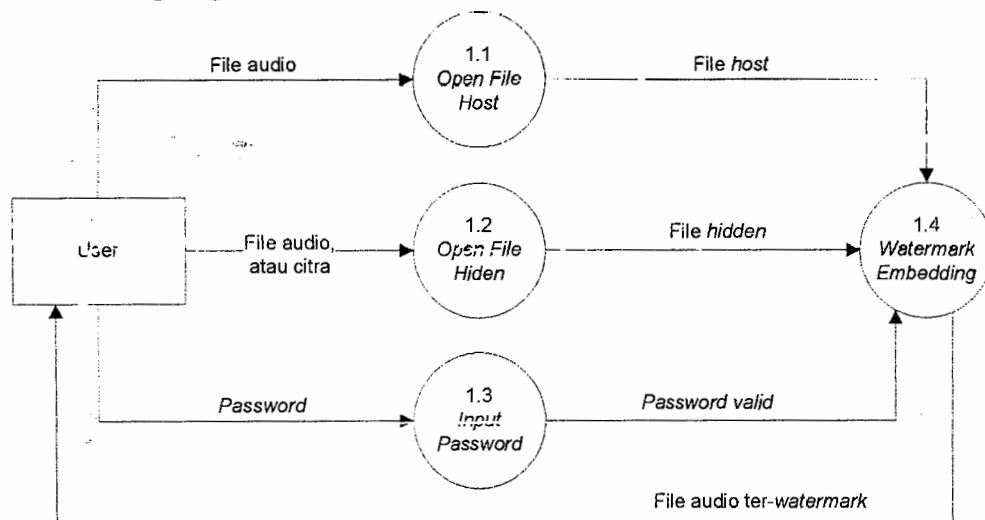
Proses ini digunakan untuk mengecek kualitas *password* yang dimasukkan oleh *user*. Password memenuhi syarat apabila berjumlah lima karakter atau lebih dan tidak ada perulangan (misal : aaaaa, abcabc, ababab, dll). Apabila

kualitas *password* memenuhi syarat maka *password* menjadi *valid* dan bisa digunakan dalam proses selanjutnya, yaitu proses *Watermark Embedding*.

4. *Watermark Embedding*

Inti dari proses *Encode* ada pada proses ini, pada proses ini file *hidden* akan disisipkan kedalam file *host*. Pada proses penyisipannya file *hidden* diubah dahulu kedalam bit-bit, kemudian bit-bit data *hidden* tersebut disisipkan kedalam kawasan frekuensi dari file *host* secara acak. Setiap satu blok kawasan frekuensi hanya bisa disisipi dua bit saja, yaitu diawal blok dan ditengah blok. Jika proses ini sukses dilakukan maka akan menghasilkan file audio ter-*watermark*.

4.1.3.3 Topologi



Gambar 3. DFD Level 2 Proses *Encode*

4.1.4 DFD Level 2 Proses *Decode*

4.1.4.1 Entitas

1. *User*

4.1.4.2 Proses

1. *Open File Watermark*

Proses ini digunakan untuk membuka file audio ter-*watermark*. File yang bisa dimasukkan hanya file audio dengan format *wave* (*.wav).

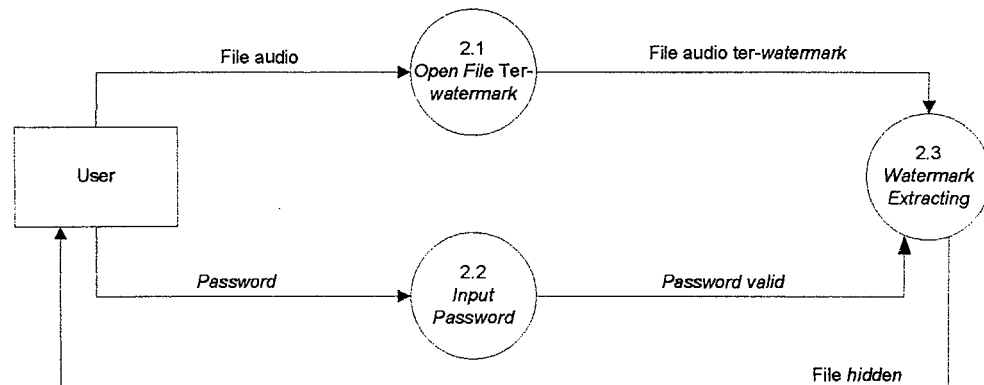
2. *Input Password*

Proses ini digunakan untuk memasukkan *password* yang akan digunakan pada proses *Decode*. Supaya data *hidden* bisa dibaca kembali maka *password* yang dimasukkan harus sama dengan *password* pada saat proses *Encode*. Password memenuhi syarat apabila berjumlah lima karakter atau lebih dan tidak ada perulangan (misal : aaaaa, abcabc, ababab, dll). Apabila *password* memenuhi syarat maka *password* menjadi *valid* dan bisa digunakan dalam proses selanjutnya, yaitu proses *Watermark Extracting*.

3. *Watermark Extracting*

Proses ini merupakan inti dari proses *Decode*, pada proses ini sistem akan menyaring bit-bit data *hidden* yang disisipkan didalam kawasan frekuensi. Supaya data *hidden* bisa dibaca maka bit-bit tersebut akan diubah dahulu kedalam byte. Jika proses ini sukses dilakukan maka akan menghasilkan file *hidden*.

4.1.4.3 Topologi



Gambar 4. DFD Level 2 Proses *Decode*

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

**Watermarking Pada Audio Wave
Dengan Menggunakan Teknik Spread Spectrum
(SSAWW)**

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Welly Seto Aji Wibowo / 3058

Program Studi Teknik Informatika – Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

2006

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>DPPL – SSAWW</i>		1/27
		Revisi		Tgl : 9-11-2006

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan	7
1.1	Tujuan	7
1.2	Lingkup Masalah	7
1.3	Definisi, Akronim dan Singkatan	7
1.4	Referensi	8
1.5	Deskripsi umum	8
2	Rancangan Arsitektur	9
2.1	Deskripsi Modul Main Menu	9
2.2	Deskripsi Modul Encode	9
2.3	Deskripsi Modul Decode	9
2.4	Deskripsi Modul Help	10
3	Deskripsi Antarmuka dan Prosedural	10
3.1	Antarmuka Main Menu	10
3.1.1	Deskripsi Tombol <i>Encode</i>	11
3.1.2	Deskripsi Tombol <i>Decode</i>	11
3.1.3	Deskripsi Tombol <i>Help</i>	11
3.1.4	Deskripsi Tombol <i>Exit</i>	11
3.2	Antarmuka Encode Proses Open File Host	11
3.2.1	Deskripsi Tombol <i>Open</i>	12
3.2.2	Deskripsi Tombol <i>Play</i>	12
3.2.3	Deskripsi Tombol <i>Back</i>	12
3.2.4	Deskripsi Tombol <i>Next</i>	13
3.2.5	Deskripsi Tombol <i>Cancel</i>	13
3.3	Antarmuka Encode Proses Open File Hidden	13
3.3.1	Deskripsi Tombol <i>Open</i>	14
3.3.2	Deskripsi Tombol <i>Run</i>	14
3.3.3	Deskripsi Tombol <i>Back</i>	15
3.3.4	Deskripsi Tombol <i>Next</i>	15
3.3.5	Deskripsi Tombol <i>Cancel</i>	15
3.4	Antarmuka Encode Proses Input Password	15
3.4.1	Deskripsi Tombol <i>Back</i>	16
3.4.2	Deskripsi Tombol <i>Next</i>	16
3.4.3	Deskripsi Tombol <i>Cancel</i>	17
3.5	Antarmuka Encode Proses Watermark Embedding	17
3.5.1	Deskripsi Event Load	17
3.5.2	Deskripsi Tombol <i>Stop</i>	18
3.6	Antarmuka Encode Result	19
3.6.1	Deskripsi Tombol <i>Play(1)</i>	20
3.6.2	Deskripsi Tombol <i>Play(2)</i>	20
3.6.3	Deskripsi Tombol <i>Exit</i>	20
3.7	Antarmuka Decode Proses Open File Watermark	20
3.7.1	Deskripsi Tombol <i>Open</i>	21
3.7.2	Deskripsi Tombol <i>Play</i>	21
3.7.3	Deskripsi Tombol <i>Back</i>	21
3.7.4	Deskripsi Tombol <i>Next</i>	22
3.7.5	Deskripsi Tombol <i>Cancel</i>	22
3.8	Antarmuka Decode Proses Input Password	22
3.8.1	Deskripsi Tombol <i>Back</i>	23
3.8.2	Deskripsi Tombol <i>Next</i>	23
3.8.3	Deskripsi Tombol <i>Cancel</i>	23
3.9	Antarmuka Encode Proses Watermark Extracting	23
3.9.1	Deskripsi Event Load	24
3.9.2	Deskripsi Tombol <i>Stop</i>	25
3.10	Antarmuka Decode Result	25

3.10.1	Deskripsi Tombol <i>Play</i>	26
3.10.2	Deskripsi Tombol <i>Exit</i>	26
3.11	Antarmuka Help.....	26

Daftar Gambar

Gambar 1. Rancangan Arsitektur.....	9
Gambar 2. Antarmuka Main Menu	10
Gambar 3. Antarmuka Encode Proses Open File Host.....	12
Gambar 4. Antarmuka Encode Proses Open File Hidden.....	14
Gambar 5. Antarmuka Encode Proses Input Password	16
Gambar 6. Antarmuka Encode Proses Watermark Embedding.	17
Gambar 7. Antarmuka Encode Result	19
Gambar 8. Antarmuka Decode Proses Open File Watermark.....	20
Gambar 9. Antarmuka Decode Proses Input Password	22
Gambar 10. antarmuka Decode Proses Watermark Extracting.	24
Gambar 11. Antarmuka Decode Result	26
Gambar 12. Antarmuka Help	27

1 Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

1.2 Lingkup Masalah

Perangkat lunak SSAWW dikembangkan dengan tujuan untuk menyisipkan *watermark* (tanda air) yang berupa bukti kepemilikan pada file audio dengan format *Wave* (*.wav). *Watermark* yang disisipkan bisa berupa file citra (*.bmp) atau file audio (*.wav). Teknik yang digunakan adalah teknik *Spread Spectrum*; dengan teknik ini data akan disebar didalam domain frekuensi secara *acak*.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
SSAW	Perangkat lunak yang digunakan untuk menyisipkan data/file kedalam audio yang berformat wav. Teknik yang digunakan dalam proses penyisipan adalah Spread Spectrum.

1.4 Referensi

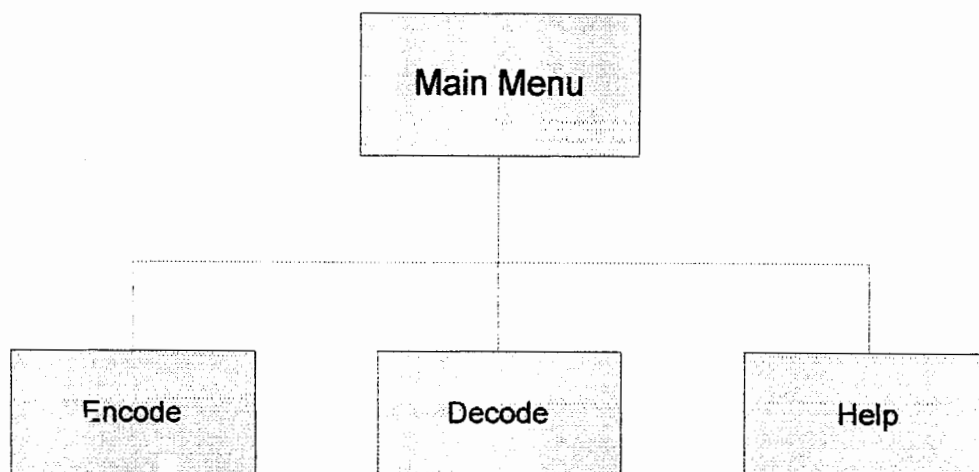
Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. *Sistem Informasi: Teori, Metodologi dan Tool*, Paulus Mudjihartono, Penerbitan Universitas Atma Jaya, Yogyakarta.
2. *Diktat Mata Kuliah: Basis Data*, RJB. Wahyu Agung., ST., MT., Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta.

1.5 Deskripsi umum

Dokumen ini terbagi dalam tiga bagian utama, yaitu : bagian pertama yang berisi penjelasan tentang dokumen DPPL yang mencakup tujuan pembuatan dokumen, lingkup masalah yang diselesaikan oleh perangkat lunak yang dikembangkan, definisi, referensi dan deskripsi umum. Bagian kedua berisi deskripsi arsitektural, sedangkan bagian ketiga berisikan deskripsi antarmuka, yaitu tampilan dari antarmuka-antarmuka yang akan dikembangkan dan deskripsi perancangan rinci, yaitu button-button yang digunakan disebuah antarmuka, juga feature-feature yang ada.

2 Rancangan Arsitektur



Gambar 1. Rancangan Arsitektur

2.1 Deskripsi Modul *Main Menu*

Modul ini adalah modul yang pertama kali di-load pada perangkat lunak SSAWW. Pada modul ini user bisa memilih proses yang akan dilakukan, yaitu : proses *Encode*, proses *Decode*, dan *Help*.

2.2 Deskripsi Modul *Encode*

Modul ini berisi proses-proses untuk menyisipkan file *hidden* ke dalam file *host*. Didalam modul ini terdiri dari dua antarmuka, yaitu : antarmuka *Encode*, dan antarmuka *Encode Result*. Antarmuka *Encode* memiliki empat proses, yaitu : *Open File Host*, *Open File Hidden*, *Input Password*, dan *Watermark Embedding*.

2.3 Deskripsi Modul *Decode*

Modul ini berisi proses-proses untuk mengekstrak file *hidden* yang ada dalam file *host*. Didalam modul ini terdiri dari dua antarmuka, yaitu : antarmuka *Decode*, dan antarmuka *Decode Result*. Antarmuka *Decode* memiliki tiga proses, yaitu : *Open File Watermark*, *Input Password*, dan *Watermark Extracting*.

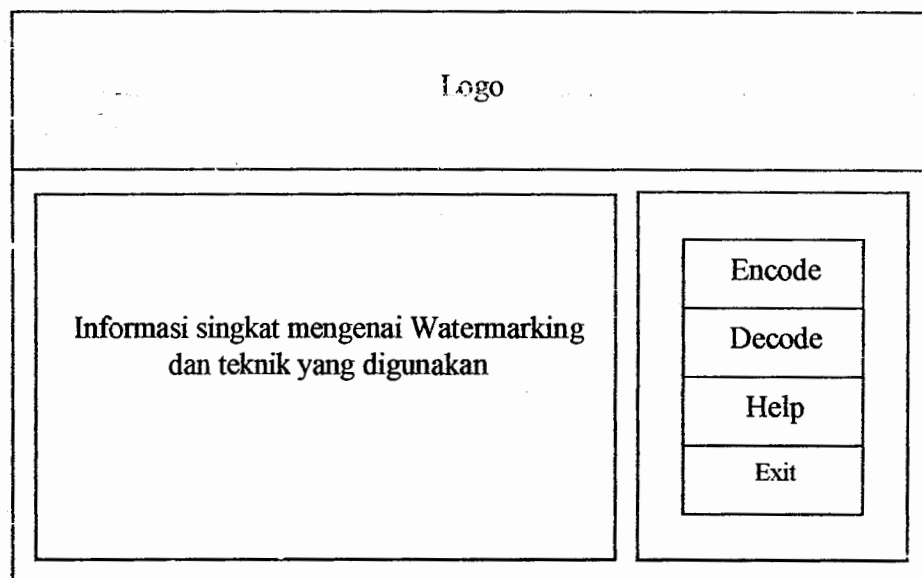
2.4 Deskripsi Modul *Help*

Modul ini digunakan untuk menampilkan informasi deskripsi/keterangan dari perangkat lunak SSAWW dan memberikan petunjuk-petunjuk penggunaan perangkat lunak ini.

3 Deskripsi Antarmuka dan Prosedural

3.1 Antarmuka *Main Menu*

Antarmuka ini merupakan antarmuka menu utama yang akan ditampilkan ketika pertama kali program dijalankan. Melalui tombol-tombol yang terdapat pada antarmuka ini user dapat memilih proses yang akan dilakukan, yaitu proses *Encode* untuk menyisipkan data berupa file *wave* atau *bitmap* ke dalam file audio *host* atau proses *Decode* untuk mengambil kembali data yang disisipkan kedalam file audio *host*. Terdapat fasilitas *Help* yang berisikan informasi deskripsi/keterangan mengenai perangkat lunak SSAWW dan petunjuk-petunjuk penggunaan perangkat lunak SSAWW.



Gambar 2. Antarmuka *Main Menu*

3.1.1 Deskripsi Tombol *Encode*

Jika menekan tombol ini user akan memulai proses *Encode*, proses *Encode* dimulai dengan memasukkan file *Host* pada antarmuka *Encode* proses *Open File Host*.

Secara prosedural :

- Open(*Encode*)
- ShowFrame(*Open File Host*)
- Close(*Main Menu*)

3.1.2 Deskripsi Tombol *Decode*

Jika menekan tombol ini user akan memulai proses *Decode*, proses *Decode* dimulai dengan memasukkan file ter-*watermark* pada antarmuka *Decode* proses *Open File Watermark*.

Secara prosedural :

- Open(*Decode*)
- ShowFrame(*Open File Watermark*)
- Close(*Main Menu*)

3.1.3 Deskripsi Tombol *Help*

Merupakan tombol untuk memanggil antarmuka *Help*.

Secara prosedural :

- Open(*Help*)
- Close(*Main Menu*)

3.1.4 Deskripsi Tombol *Exit*

Tombol ini digunakan untuk menutup antarmuka *Main Menu* dan keluar dari sistem.

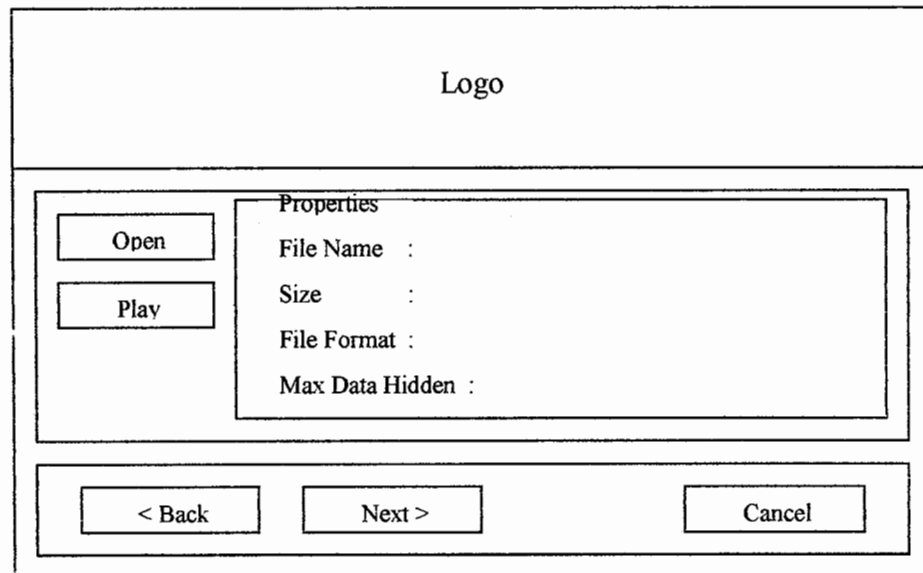
Secara prosedural :

- Exit

3.2 Antarmuka *Encode* Proses *Open File Host*

Pada antarmuka ini user akan memasukkan file *wave* yang akan digunakan sebagai *Host*. Kemudian akan dibaca *header* dan nilai-nilai yang terdapat pada file *wave* tersebut. Prosedur yang digunakan untuk membuka file *wave* dan membaca *header* dan nilai di dalamnya diimplementasikan dengan nama *LoadFileHost* yang disimpan dalam modul *Function.bas*. Selain itu akan dicari ukuran maksimal dari file *hidden* yang dapat dimasukkan dengan menggunakan *function getMaxDataHidden*.

Program Studi Teknik Informatika	DPPL - SSAWW	11 / 27
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		



Secara prosedural :
Open(Main Menu)
Close(Encode)

3.2.4 Deskripsi Tombol *Next*

Jika user telah menentukan file *host* maka akan menuju ke langkah selanjutnya, yaitu proses *Open File Hidden*.

Secara prosedural :
If FileHost ($\neq$) null then {jika user sudah memilih file Host}
 showFrame(Open File Hidden)
 hideFrame(Open File Host)
Else {jika user belum memilih file Host}
 Output ('User belum memilih file Host')
End if

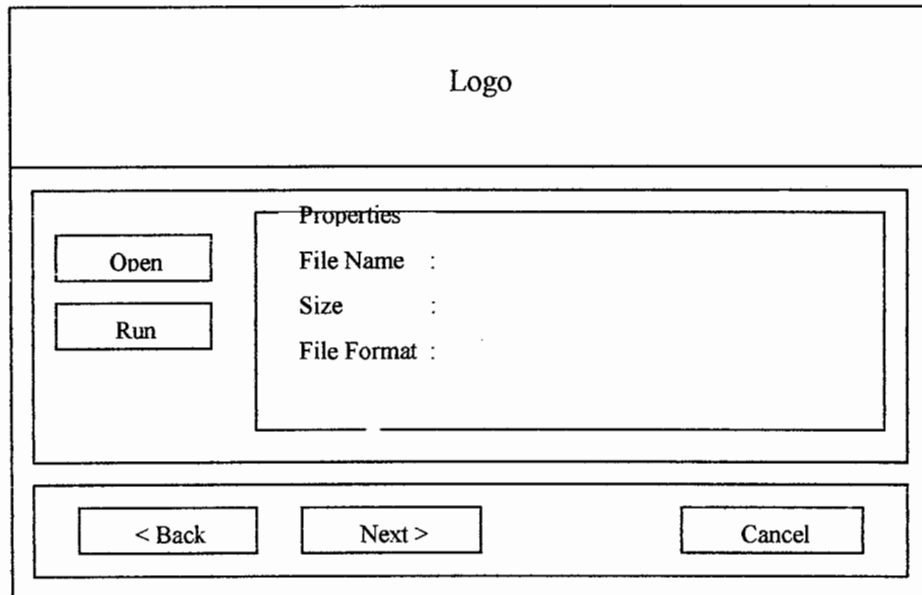
3.2.5 Deskripsi Tombol *Cancel*

Membatalkan proses *Encode* dan kembali ke antarmuka *Main Menu*.

Secara prosedural :
Open(Main Menu)
Close(Encode)

3.3 Antarmuka *Encode Proses Open File Hidden*

Antarmuka ini digunakan untuk memilih file yang akan disisipkan ke dalam file *wave*. File yang disisipkan dapat berupa citra (*.bmp) atau audio (*.wav). Kemudian akan dibaca byte-byte yang terdapat pada file *hidden* tersebut dengan menggunakan function *LoadFileHidden*. Karena byte-byte file *hidden* akan disisipkan kedalam kawasan frekuensi maka byte-byte tersebut diubah dahulu kedalam bit-bit/*binary string*.



Gambar 4. Antarmuka Encode Proses Cpen File Hidden

3.3.1 Deskripsi Tombol *Open*

Merupakan tombol untuk menampilkan *Dialog Box Open* untuk membuka file dengan format *.wav atau *.bmp yang akan digunakan sebagai file *hidden*. Dan mengubah file *hidden* tersebut menjadi *binary string*.

Secara prosedural :

Input(FileHidden)

If MaxDataHidden < ukuran(filehidden)

Output('Ukuran file hidden terlalu besar, silahkan ulangi lagi memasukkan file hidden')

else

{memanggil function LoadFileHidden untuk membaca header Dan nilai dari File Hidden, nilainya disimpan dalam array DataHidden()}

DataHidden() = LoadFileHidden(FileHidden)

{Memanggil function BytetoBit untuk mengubah array DataHidden() dari byte ke bit dan disimpan dalam array bitDataHidden}

BitDataHidden()=BytetoBit(DataHidden())

End if

3.3.2 Deskripsi Tombol *Run*

Untuk membuka file *hidden* yang telah dipanggil sebelumnya.

Secara prosedural :

If FileHidden (≠) null then {jika user sudah memasukkan file hidden}

Run (FileHidden)

Else

{jika user belum memasukkan file hidden}

Output ('User belum memasukkan file hidden')

End if

3.3.3 Deskripsi Tombol *Back*

Menuju ke langkah sebelumnya, yaitu kembali ke proses *Open File Host*.

Secara prosedural :

```
ShowFrame(Open File Host)
HideFrame(Open File Hidden)
```

3.3.4 Deskripsi Tombol *Next*

Jika user telah memasukkan file *hidden* maka akan menuju ke langkah selanjutnya, yaitu proses *Input Password*.

Secara prosedural :

```
If FileHidden (≠) null then      {jika user sudah memasukkan file hidden}
  ShowFrame(Input Password)
  HideFrame(Open File Hidden)
Else                             {jika user belum memasukkan file hidden}
  Output ("User belum memasukkan file hidden")
End if
```

3.3.5 Deskripsi Tombol *Cancel*

Membatalkan proses *Encode* dan kembali ke antarmuka *Main Menu*.

Secara prosedural :

```
Open(Main Menu)
Close(Encode)
```

3.4 Antarmuka *Encode Proses Input Password*

Antarmuka ini digunakan untuk menginputkan *Password* yang akan menjadi kunci pada proses penyisipan. Pada antarmuka ini akan dicek kualitas *password* yang dimasukkan oleh *user*. *Password* memenuhi syarat apabila berjumlah lima karakter atau lebih dan tidak ada perulangan (misal : aaaaa, abcabc, ababab, dll). Apabila kualitas *password* memenuhi syarat maka proses *Watermark Embedding* bisa dilakukan.

Gambar 5. Antarmuka Encode Proses Input Password

3.4.1 Deskripsi Tombol *Back*

Menuju ke langkah sebelumnya, yaitu kembali ke proses *Open File Hidden*.

Secara prosedural :

```
ShowFrame(Open File Hidden)
HideFrame(Input Password)
```

3.4.2 Deskripsi Tombol *Next*

Jika user telah memasukkan *password*, maka akan dicek kualitas *password* tersebut. Jika memenuhi syarat maka akan menuju ke langkah berikutnya, yaitu : proses *Watermark Embedding*.

Secara prosedural :

```
Input(Password)
Input(Verify)
```

```
{memanggil function KeyQuality untuk Mencari kualitas Password dan nilainya
Disimpan dalam variabel KualitasPassword}
KualitasPassword=KeyQuality(Password)
```

```
{jika kualitas password kurang dari 20, maka password tidak memenuhi syarat}
If KualitasPassword<20 then
    Output('Password tidak memenuhi syarat, silahkan ulangi pemasukkan password')
Elseif Password (≠) Verify then
    Output('Password dan verify tidak sama, silahkan ulangi pemasukkan password dan
    verify')
Else
    ShowFrame(Watermark Embedding)
    HideFrame(Input Password)
End if
```

3.4.3 Deskripsi Tombol *Cancel*

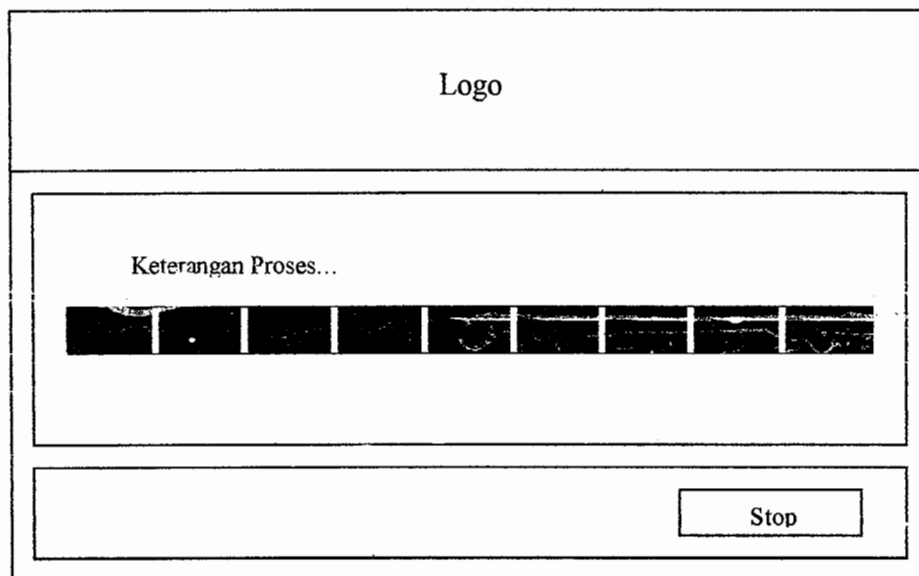
Membatalkan proses *Encode* dan kembali ke antarmuka *Main Menu*.

Secara prosedural :

Open(Main Menu)
Close(Encode)

3.5 Antarmuka *Encode* Proses *Watermark Embedding*

Antarmuka ini digunakan untuk menyisipkan file *hidden* kedalam file *host*. Pada saat pertama kali antarmuka ini di-load sistem langsung melakukan proses *Watermark Embedding*. Disini user hanya bisa mengamati proses yang berlangsung melalui label keterangan proses dan *progress bar* proses. Jika proses *Watermark Embedding* sukses dilakukan maka menghasilkan file audio ter-*watermark*.



Gambar 6. Antarmuka *Encode* Proses *Watermark Embedding*

3.5.1 Deskripsi Event Load

Saat antarmuka ini di-load maka proses *Watermark Embedding* dilakukan. Jika proses *Watermark Embedding* sukses dilakukan maka akan menghasilkan file audio ter-*watermark* dan akan tampil antarmuka *Encode Result*.

Secara prosedural :

Input(DataAsli())
Input(bitDataHidden())
Input(FileOutput)

Program Studi Teknik Informatika	DPPL - SSAWW	17 / 27
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

{Mengacak indeks supaya posisi penyisipan dilakukan secaraacak, dan hasilnya disimpan didalam array index}
 RandomIndex(Index())

Do

{perulangan for dibawah ini digunakan untuk mengambil 8 sample data asli pada Posisi index(i)+1 sebagai masukan pada proses FFT}

for j=1 to j=8 do

DomainFFT(j)=DataAsli(index(i)+j)

{memanggil prosedur FFT dan hasilnya disimpan diarray frekuensi}

Frekuensi()=FFT(DomainFFT())

for x=1 to x=2 do

bitF() = BytetoBitF(Frekuensi(y)) *{mengubah dari byte ke bit}*

bitF(6)=bitDataHidden(i) *{idealnya penyisipan data diposisi bit ke 6}*

{setelah proses penyisipan data diubah kembali dari bit ke byte}

Frekuensi(y) = BittoByteF(bitF())

{menuju ke frekuensi berikutnya, frekuensi yang disisipkan adalah frekuensi yang ke 1 dan ke 5}

y=(8/2)+1

{memanggil prosedur IFFT untuk mengubah Kembali data dari kawasan frekuensi ke kawasan waktu}

DataTerwatermark()=IFFT(Frekuensi())

{memasukkan data terwatermark kedalam data asli}

for k=1 to k=8 do

DataAsli(index(i)+k)=dataTerwatermark(k)

{menampilkan pergerakan progressbar dari proses yang sedang berlangsung}

ProgressBar(i * 100 / panjang(BitDataHidden))

i=i+1

{looping dilakukan selama i lebih kecil dari Jumlah bit Data Hidden}

While i>panjang(bitDataHidden(i))

{membuat file baru yang merupakan hasil keluaran dari proses Encode, nilainya diambil dari array DataAsli}

Open FileOutput For Binary Write As #FileHandler

Put #FileHandler,,DataAsli()

Close #FileHandler

Open(Encode Result) *{memanggil antarmuka Encode Result}*

Close(Encode) *{menutup antarmuka Encode}*

3.5.2 Deskripsi Tombol Stop

Tombol ini digunakan untuk membatalkan proses *Watermark Embedding* dan kembali ke antarmuka *Encode* proses *Open File Host*.

Program Studi Teknik Informatika	DPPL - SSAWW	18 / 27
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Secara prosedural :

```
Stop(Watermark Embedding)      {menghentikan proses Watermark Embedding}  
Output('User membatalkan proses Watermark Embedding')  
ShowFrame(Open File Host)  
HideFrame(Watermark Embedding)
```

3.6 Antarmuka *Encode Result*

Antarmuka ini menampilkan informasi mengenai file *wave* sebelum dilakukan proses penyisipan dan setelah dilakukan penyisipan. Terdapat dua tombol Play yang digunakan untuk membandingkan suara file audio asli dengan suara file audio ter-*watermark*. Dan terdapat dua gambar *spectrum* yang digunakan untuk membandingkan file asli dengan file ter-*watermark* jika dilihat dari kawasan waktu.

The interface consists of a main window with a title bar. Inside, there are two vertically stacked panels. The top panel is labeled 'File Input' and contains a 'Play (1)' button, a text field for 'File Name', and labels for 'Size' and 'File Format'. Below this is a 'Spectrum' label and a large rectangular area labeled 'Gambar Spectrum'. The bottom panel is labeled 'File Output' and contains a 'Play (2)' button, a text field for 'File Name', and labels for 'Size' and 'File Format'. Below this is another 'Spectrum' label and a large rectangular area labeled 'Gambar Spectrum'. At the bottom right of the window is an 'Exit' button.

Gambar 7. Antarmuka *Encode Result*

3.6.1 Deskripsi Tombol *Play*(1)

Untuk memainkan file *Wave Host*.

Secara prosedural :

Input(FileHost)
Play(FileHost)

3.6.2 Deskripsi Tombol *Play*(2)

Untuk memainkan file *Wave ter-watermark*.

Secara prosedural :

Input(FileWatermark)
Play(FileWatermark)

3.6.3 Deskripsi Tombol *Exit*

Tombol ini digunakan untuk menutup antarmuka *Encode Result* dan mengakhiri proses *Encode*. Dan akan kembali ke antarmuka *Main Menu*.

Secara prosedural :

Open(Main Menu)
Close(Encode Result)

3.7 Antarmuka *Decode Proses Open File Watermark*

Pada antarmuka ini user akan memasukkan file *wave ter-watermark*. Kemudian akan dibaca *header* dan nilai-nilai yang terdapat pada file *wave* tersebut. Prosedur yang digunakan untuk membuka file *wave* dan membaca *header* dan nilai di dalamnya diimplementasikan dengan nama *LoadFileWatermark* yang disimpan dalam modul *Function.bas*.

Gambar 8. Antarmuka Decode Proses Open File Watermark

3.7.1 Deskripsi Tombol *Open*

Merupakan tombol untuk menampilkan *Dialog Box Open* untuk membuka file ter-watermark dengan format *.wav. Dan mengambil header dan nilai dari file wave tersebut dengan menggunakan prosedur *LoadFileWatermark*.

Secara prosedural:

Input(FileWatermark)

{memanggil prosedur *LoadFileWatermark* untuk membaca header dan nilai dari *FileWatermark*}

LoadFileWatermark (FileWatermark)

3.7.2 Deskripsi Tombol *Play*

Untuk memainkan file audio ter-watermark yang telah dipanggil sebelumnya

Secara prosedural :

If FileWatermark (≠) null then {jika user sudah memilih file ter-watermark}

Play(FileWatermark)

Else {jika user belum memilih file ter-watermark}

Output ('User belum memilih file ter-watermark')

End if

3.7.3 Deskripsi Tombol *Back*

Menuju ke langkah sebelumnya, yaitu kembali ke antarmuka *Menu Utama*.

Secara prosedural :

Open(Main Menu)

Close(Decode)

3.7.4 Deskripsi Tombol *Next*

Jika user telah menentukan file *watermark* maka akan menuju ke langkah selanjutnya, yaitu memanggil proses *Input Password*.

Secara prosedural :

```
If FileWatermark (≠) null then      {jika user sudah memilih file ter-watermark;}
    ShowFrame(Input Password)
    HideFrame(Open File Watermark)
Else                                  {jika user belum memilih file ter-watermark}
    Output ("User belum memilih file ter-watermark")
End if
```

3.7.5 Deskripsi Tombol *Cancel*

Membatalkan proses *Decode* dan kembali ke antarmuka *Main Menu*.

Secara prosedural :

```
Open(Main Menu)
Close(D'ecode)
```

3.8 Antarmuka *Decode Proses Input Password*

Antarmuka ini digunakan untuk memasukkan *password* yang akan digunakan pada proses *Decode*. Supaya data *hidden* bisa dibaca kembali maka *password* yang dimasukkan harus sama dengan *password* pada saat proses *Encode*. *Password* memenuhi syarat apabila berjumlah lima karakter atau lebih dan tidak ada perulangan (misal : aaaaa, abcabc, ababab, dll). Apabila *password* memenuhi syarat maka proses *Watermark Extracting* bisa dilakukan.

The image shows a graphical user interface window titled 'Decode Proses Input Password'. At the top center is a box labeled 'Logo'. Below it is a large rectangular frame containing a 'Password' label and a corresponding text input field. At the bottom of the window, there are three buttons: '< Back' on the left, 'Next >' in the center, and 'Cancel' on the right.

Gambar 9. Antarmuka *Decode Proses Input Password*

3.8.1 Deskripsi Tombol *Back*

Menuju ke langkah sebelumnya, yaitu kembali ke proses *Open File Watermark*.

Secara prosedural :

```
ShowFrame(Open File Watermark)
HideFrame(Input Password)
```

3.8.2 Deskripsi Tombol *Next*

Jika user telah memasukkan *password*, maka akan dicek jumlah karakter dari *password* tersebut. Jika berjumlah lima karakter atau lebih maka bisa menuju ke langkah berikutnya, yaitu : proses *Watermark Extracting*.

Secara prosedural :

```
Input(Password)

{jika karakter password kurang dari 5, maka password tidak memenuhi syarat}
If panjang(Password) < 5 then
    Output('Password tidak memenuhi syarat, silahkan ulangi pemasukkan password')
Else
    ShowFrame(Watermark Extracting)
    HideFrame(Input Password)
End if
```

3.8.3 Deskripsi Tombol *Cancel*

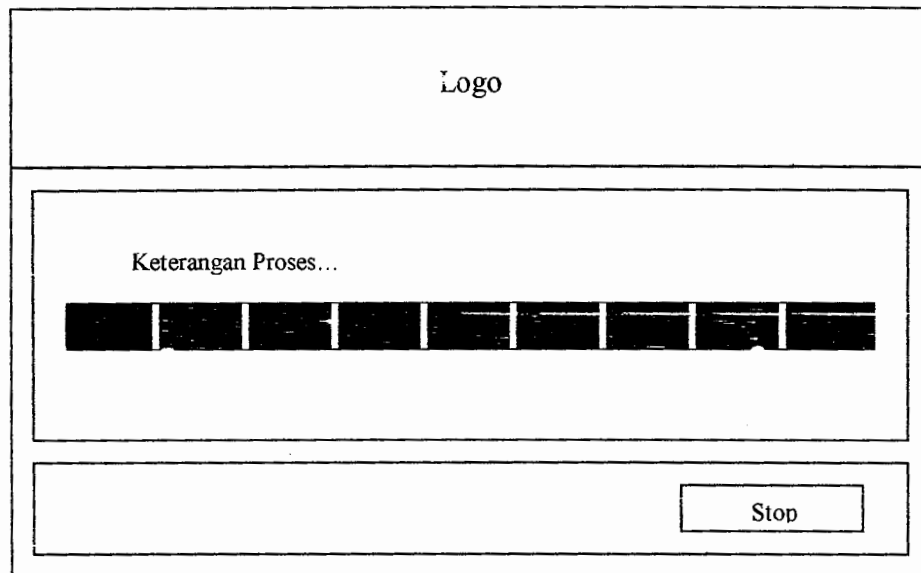
Membatalkan proses *Decode* dan kembali ke antarmuka *Main Menu*.

Secara prosedural :

```
Open(Main Menu)
Close(Decode)
```

3.9 Antarmuka *Encode Proses Watermark Extracting*

Antarmuka ini digunakan untuk membaca kembali file *hidden* yang disisipkan kedalam file *host*. Pada saat pertama kali antarmuka ini di-load sistem langsung melakukan proses *Watermark Extracting*. Disini user hanya bisa mengamati proses yang berlangsung melalui label keterangan proses dan *progress bar* proses. Jika proses *Watermark Extracting* sukses dilakukan maka file *hidden* akan ditampilkan pada antarmuka *Decode Result*.



Gambar 10. Antarmuka Decode Proses Watermark Extracting

3.9.1 Deskripsi Event Load

Saat antarmuka ini di-load maka proses *Watermark Extracting* dilakukan. Jika proses *Watermark Extracting* sukses dilakukan maka akan menghasilkan file *hidden* dan akan tampil antarmuka *Decode Result*.

Secara prosedural :

Input(DataWatermark())

{Mengacak indeks supaya posisi penyisipan Dilakukan secara acak, dan hasilnya disimpan didalam array index}

RandomIndex(Index())

Do

{perulangan fordibawah ini digunakan untuk mengambil 8 sample data Watermark pada posisi index(i)+1 sebagai masukan pada proses FFT}

For j=1 to j=8 do

DomainFFT(j)=DataWatermark(index(i)+j)

{memanggil prosedur FFT dan hasilnya disimpan diarray frekuensi}

Frekuensi()=FFT(DomainFFT())

For x=1 to x=2 do

bitF() = BytetoBitF(Frekuensi(y)) {mengubah dari byte ke bit}

bitDataHidden(i) = bitF(6) {pembacaan bit diposisi bit ke 6}

{setelah proses penyisipan data diubah kembali dari bit ke byte}

Frekuensi(y) = BittoByteF(bitF())

{menuju ke frekuensi berikutnya, frekuensi yang diproses adalah frekuensi yang ke 1 dan ke 5}

y=(8/2)+1

```

    {menampilkan pergerakan progressbar dari proses yang sedang berlangsung}
    ProgressBar(i * 100 / panjang(BitDataHidden))

    i=i+1
    {looping dilakukan selama i lebih kecil dari Jumlah bit Data Watermark}
    While i>panjang(DataWatermark(i))

    {memanggil prosedur byteToBit untuk mengubah data dari bit ke byte}
    DataHidden()=bitToByte(bitDataHidden())

    {membuat file baru yang merupakan hasil keluaran dari proses decode dengan nilai dari
    array DataHidden}
    Open FileWatermark for binary write as #FileHandler
    put #FileHandler,, DataHidden()
    close #FileHandler

    Open(Decode Result)          {memanggil antarmuka Decode Result}
    Close(Decode)                {menutup antarmuka Decode}

```

3.9.2 Deskripsi Tombol Stop

Tombol ini digunakan untuk membatalkan proses *Watermark Extracting* dan kembali ke antarmuka *Decode* proses *Open File Watermark*.

Secara prosedural :

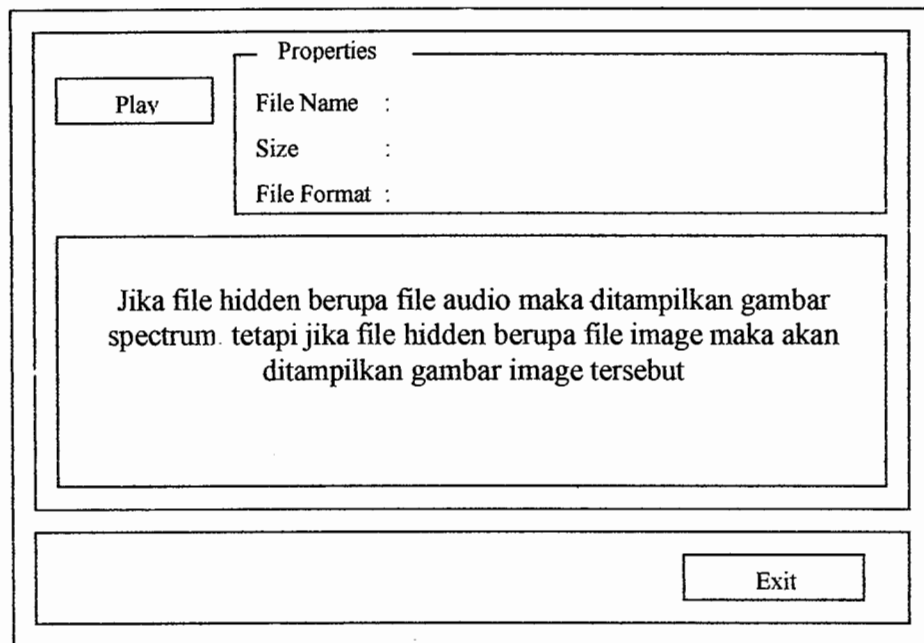
```

    Stop(Watermark Extracting)    {menghentikan proses Watermark Extracting}
    Output("User membatalkan proses Watermark Extracting")
    ShowFrame(Open File Watermark)
    HideFrame(Watermark Extracting)

```

3.10 Antarmuka Decode Result

Jika proses *Watermark Extracting* sukses dilakukan maka akan ditampilkan antarmuka ini, antarmuka ini digunakan untuk menampilkan file *hidden* yang merupakan hasil dari proses *Decode*.



Gambar 11. Antarmuka *Decode Result*

3.10.1 Deskripsi Tombol *Play*

Tombol ini hanya akan aktif jika file *hidden* berupa file audio *wave*. Tombol ini digunakan untuk memainkan file *hidden*.

Secara prosedural :

Play (FileWatermark)

3.10.2 Deskripsi Tombol *Exit*

Tombol ini digunakan untuk menutup antarmuka *Decode Result* dan mengakhiri proses *Decode*. Dan akan kembali ke antarmuka *Main Menu*.

Secara prosedural :

Open(Main Menu)

Close(Decode Result)

3.11 Antarmuka *Help*

Antarmuka ini digunakan untuk menampilkan petunjuk-petunjuk penggunaan program ini. Pada bagian sebelah kiri user dapat memilih topik-topik yang ada dan sebelah kanan adalah keterangan mengenai topik tersebut.

Index / Topik	Keterangan mengenai topik tersebut.
------------------	-------------------------------------

Gambar 12. Antarmuka *Help*

PDHUPL

**PERENCANAAN, DESKRIPSI, DAN HASIL
UJI PERANGKAT LUNAK**

**Watermarking Pada Audio Wave Dengan
Menggunakan Teknik Spread Spectrum
(SSAWW)**

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Welly Seto Aji Wibowo / 3058

Program Studi Teknik Informatika – Fakultas Teknologi Industri

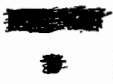
Universitas Atma Jaya Yogyakarta

2006

	Program Studi Teknik Informatika FTI - UAJY	Nomor Dokumen		Halaman
		<i>PDHUPL-SSAWW</i>		<i>1/14</i>
		Revisi	-	<i>Tgl: 10-12-2006</i>

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan	6
1.1	Tujuan Pembuatan Dokumen	6
1.2	Deskripsi Umum Sistem	6
1.3	Deskripsi Dokumen (Ikhtisar)	6
1.4	Definisi dan Singkatan	7
1.5	Dokumen Referensi	7
2	Lingkungan Pengujian Perangkat Lunak	8
2.1	Perangkat Lunak Pengujian	8
2.2	Perangkat Keras Pengujian	8
2.3	Sumber Daya Manusia	8
2.4	Prosedur Umum Pengujian	8
2.4.1	Pengenalan dan Latihan	8
2.4.2	Persiapan Perangkat Keras	9
2.4.3	Persiapan Perangkat Lunak	9
2.4.4	Pelaksanaan	9
2.4.5	Pelaporan Hasil	9
3	Identifikasi dan Rencana Pengujian	9
4	Deskripsi dan Hasil Uji	10
4.1	Identifikasi Kelas Pengujian Antarmuka Encode	10
4.1.1	Identifikasi Butir Pengujian <i>Open File Host</i> – ENC-01-01	10
4.1.2	Identifikasi Butir Pengujian <i>Open File Hidden</i> – ENC-01-02	11
4.1.3	Identifikasi Butir Pengujian <i>Input Password</i> – ENC-01-03	11
4.1.4	Identifikasi Butir Pengujian <i>Watermark Embedding</i> – ENC-01-04	11
4.2	Identifikasi Kelas Pengujian Antarmuka Decode	11
4.2.1	Identifikasi Butir Pengujian <i>Open File Watermark</i> – DEC-02-01	11
4.2.2	Identifikasi Butir Pengujian <i>Input Password</i> – DEC-02-02	11
4.2.3	Identifikasi Butir Pengujian <i>Watermark Extracting</i> – DEC-02-03	12
4.3	Identifikasi Butir Pengujian Help – HLP-03	12

Daftar Tabel

Tabel 1. Definisi	7
Tabel 2. Identifikasi Pengujian	9
Tabel 3. Deskripsi dan Hasil Pengujian	13

1 Pendahuluan

1.1 Tujuan Pembuatan Dokumen

Dokumen PDHUPL ini adalah dokumen yang berisi perencanaan, deskripsi dan hasil pengujian perangkat lunak SSAWW (*Spread Spectrum Audio Wave Watermarking*). Dokumen PDHUPL ini dibuat untuk jurusan Teknik Informatika Universitas Atmajaya Yogyakarta. Selanjutnya dokumen ini dipergunakan sebagai bahan panduan untuk melakukan pengujian terhadap sistem SSAWW. PDHUPL ini juga akan digunakan untuk menguji keseluruhan sistem SSAWW.

1.2 Deskripsi Umum Sistem

SSAWW adalah aplikasi yang digunakan untuk memberikan *watermarking* pada file audio *wave* dengan menggunakan teknik spread spectrum. Sistem ini secara garis besar terdiri dari tiga komponen besar, yaitu :

- a. modul yang menangani proses *Encode*.
- b. modul yang menangani proses *Decode*.
- c. modul *help*.

1.3 Deskripsi Dokumen (Ikhtisar)

Dokumen PDHUPL ini mempunyai sistematika penulisan sebagai berikut:

Bagian 1. Pendahuluan

1.1. Tujuan Pembuatan Dokumen

1.2. Deskripsi Umum Sistem

1.3. Deskripsi Dokumen atau Ikhtisar

1.4. Definisi dan Singkatan

1.5. Dokumen Referensi

Bagian 2. Lingkungan Pengujian Perangkat Lunak

2.1. Perangkat Lunak Pengujian

Program Studi Teknik Informatika	PDHUPL - SSAWW	6 / 14
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2.2. Perangkat Keras Pengujian

2.3. Sumber Daya Manusia

2.4. Prosedur Umum Pengujian

2.4.1. Pengenalan dan Latihan

2.4.2. Persiapan Perangkat Keras

2.4.3. Persiapan Perangkat Lunak

2.4.4. Pelaksanaan

2.4.5. Pelaporan Hasil

Bagian 3. Identifikasi dan Rencana Pengujian

Bagian 4. Deskripsi dan Hasil uji

4.1. Identifikasi Kelas Pengujian

4.1.1. Identifikasi Butir Pengujian

1.4 Definisi dan Singkatan

Tabel 1. Definisi

Kata Kunci dan Frase	Definisi
<i>Encode</i>	Proses untuk menyisipkan <i>watermarking</i> pada audio <i>wave</i> .
<i>Decode</i>	Proses untuk membaca kembali <i>watermarking</i> yang telah disisipkan sebelumnya pada audio <i>wave</i> .
<i>File host</i>	File yang akan ditumpangi/diberi <i>watermarking</i> .
<i>File hidden</i>	File yang akan disisipkan kedalam file <i>host</i> .
<i>Audio wave</i>	File audio dengan format <i>wave</i>
PDHUPL-SSAWW	Dokumen yang berisi tentang perencanaan, deskripsi dan hasil uji perangkat lunak dari sistem SSAWW.

1.5 Dokumen Referensi

1. Wibowo, Welly Seto Aji, 2006, *Spesifikasi Kebutuhan Perangkat Lunak SSAWW*, Universitas Atma Jaya Yogyakarta.

Program Studi Teknik Informatika	PDHUPL - SSAWW	7 / 14
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Wibowo, Welly Seto Aji, 2006, *Deskripsi Perancangan Perangkat Lunak SSAWW*, Universitas Atma Jaya Yogyakarta.

2 Lingkungan Pengujian Perangkat Lunak

2.1 Perangkat Lunak Pengujian

Perangkat lunak Pengujian berupa:

1. Windows XP SP 2 dari Microsoft sebagai sistem operasi.

2.2 Perangkat Keras Pengujian

1. Komputer PC, dengan spesifikasi minimal Pentium II 400 Mhz, 64 MB RAM.

2.3 Sumber Daya Manusia

Sumber daya pengujian ini berupa:

1. Tester → terdiri dari 30 orang dengan spesifikasi minimal lulusan SMU atau sederajat, dan bisa mengoperasikan komputer.

2.4 Prosedur Umum Pengujian

2.4.1 Pengenalan dan Latihan

Pengenalan dan Latihan perangkat lunak SSAWW diharapkan tidak memerlukan waktu lama. SSAWW diharapkan dapat dipelajari langsung dari antarmuka *help* (bantuan), tanpa melalui pelatihan khusus. Pengguna SSAWW adalah pengguna yang ingin memproteksi file *wave* audio dengan menyisipkan bukti kepemilikan kedalam file *wave* tersebut dan telah familiar dengan penggunaan program berbasis GUI.

2.4.2 Persiapan Perangkat Keras

Persiapan perangkat keras berupa :

1. Komputer yang sudah ter-*install* aplikasi SSAWW.
2. Keyboard.
3. Mouse.
4. Speaker

2.4.3 Persiapan Perangkat Lunak

Persiapan perangkat lunak berupa :

1. Instalasi aplikasi SSAWW.

2.4.4 Pelaksanaan

Pelaksanaan pengujian akan dilaksanakan dalam dua tahap, yaitu pengujian unit (modul-modul kecil) dan pengujian sistem secara keseluruhan.

2.4.5 Pelaporan Hasil

Hasil pengujian akan diserahkan kepada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta.

3 Identifikasi dan Rencana Pengujian

Tabel 2. Identifikasi Pengujian

Kelas Uji	Butir Uji	Identifikasi		Tingkat Pengujian	Jenis Pengujian	Jadwal
		SKPL	PDHUPL			
Pengujian antar muka Encode	Pengujian Open File Host	SKPL-SSAWW-01-01	ENC-01-01	Pengujian Unit	Black Box	5/12/2006
	Pengujian Open File Hidden	SKPL-SSAWW-01-02	ENC-01-02	Pengujian Unit	Black Box	5/12/2006
	Pengujian	SKPL-SSAWW-	ENC-01-03	Pengujian Unit	Black Box	5/12/2006

Program Studi Teknik Informatika

PDHUPL - SSAWW

9 / 14

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

	<i>Input Password</i>	01-03				
	Pengujian <i>Watermark Embedding</i>	SKPL-SSAWW-01-04	ENC-01-04	Pengujian Unit	Black Box	5/12/2006
Pengujian antar muka <i>Decode</i>	Pengujian <i>Open File Watermark</i>	SKPL-SSAWW-02-01	DEC-02-01	Pengujian Unit	Black Box	6/12/2006
	Pengujian <i>Input Password</i>	SKPL-SSAWW-02-02	DEC-02-02	Pengujian Unit	Black Box	6/12/2006
	Pengujian <i>Watermark Extracting</i>	SKPL-SSAWW-02-03	DEC-02-03	Pengujian Unit	Black Box	6/12/2006
Pengujian antarmuka <i>Help</i>	Pengujian <i>Help</i>	SKPL-SSAWW-03	HLP-03	Pengujian Sistem	White Box	6/12/2006

4 Deskripsi dan Hasil Uji

4.1 Identifikasi Kelas Pengujian Antarmuka *Encode*

Kelas pengujian antarmuka *Encode* adalah kelas pengujian yang meliputi pengujian-pengujian untuk melakukan proses *Encode* yang melibatkan fungsi antarmuka dengan user sebagai penggunanya.

4.1.1 Identifikasi Butir Pengujian *Open File Host* – ENC-01-01

Butir pengujian ini menguji memasukkan file audio yang akan digunakan sebagai *host*. User akan memilih file *host* melalui antarmuka Common Dialog Show Open yang akan menuntun user menuju ke direktori file *host* tersebut.

4.1.2 Identifikasi Butir Pengujian *Open File Hidden* – ENC-01-02

Butir pengujian ini menguji memasukkan file yang akan digunakan sebagai file *hidden*. User akan memilih file *hidden* melalui antarmuka Common Dialog Show Open yang akan menuntun user menuju ke direktori file *hidden* tersebut.

4.1.3 Identifikasi Butir Pengujian *Input Password* – ENC-01-03

Butir pengujian ini menguji kualitas *password* yang dimasukkan oleh user. Jika kualitas *password* memenuhi syarat maka pengujian berikutnya bisa dilakukan.

4.1.4 Identifikasi Butir Pengujian *Watermark Embedding* – ENC-01-04

Butir pengujian ini menguji menyisipkan data *hidden* kedalam kawasan frekuensi pada file *host* secara acak.

4.2 Identifikasi Kelas Pengujian Antarmuka *Decode*

Kelas pengujian antarmuka *Decode* adalah kelas pengujian yang meliputi pengujian-pengujian untuk melakukan proses *Decode* yang melibatkan fungsi antarmuka dengan user sebagai penggunanya.

4.2.1 Identifikasi Butir Pengujian *Open File Watermark* – DEC-02-01

Butir pengujian ini menguji memasukkan file audio ter-*watermark*. User akan memilih file ter-*watermark* melalui antarmuka Common Dialog Show Open yang akan menuntun user menuju ke direktori file ter-*watermark* tersebut.

4.2.2 Identifikasi Butir Pengujian *Input Password* – DEC-02-02

Butir pengujian ini menguji *password* yang dimasukkan oleh user. Jika *password* memenuhi syarat maka pengujian berikutnya bisa dilakukan.

Program Studi Teknik Informatika	PDHUPL - SSAWW	11 / 14
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4.2.3 Identifikasi Butir Pengujian *Watermark Extracting* – DEC-02-03

Butir pengujian ini menguji membaca kembali data *hidden* yang disisipkan kedalam kawasan frekuensi pada file *host*.

4.3 Identifikasi Butir Pengujian *Help* – HLP-03

Butir pengujian ini menguji antarmuka *Help* yang akan memberi bantuan kepada *user* dalam menggunakan aplikasi SSAWW ini.

Tabel 3. Deskripsi dan Hasil Pengujian

Identifikasi	Deskripsi	Prosedur Pengujian	Masukkan	Isi/urutan yg diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
ENC-01-01	Pengujian membuka file <i>host</i>	- Pilih menu "Open" pada antarmuka <i>Encode proses Open File Host</i> - Pilih file <i>host</i>	- Tekan tombol "Open" - Direktori file <i>host</i> berada	- Antarmuka <i>Common Dialog Show Open</i> - Keterangan : nama file, ukuran file, format file, dan ukuran maksimal file hidden - Menu "Play" dan menu "Next" aktif	- Keluar antarmuka <i>Common Dialog Show Open</i> - Keluar keterangan : nama file, ukuran file, format file, dan ukuran maksimal file hidden - Menu "Play" dan menu "Next" menjadi aktif	- Antarmuka <i>Common Dialog Show Open</i> - Keterangan : nama file, ukuran file, format file, dan ukuran maksimal file hidden - Menu "Play" dan menu "Next" aktif	Handal
ENC-01-02	Pengujian membuka file <i>hidden</i>	- Pilih menu "Open" pada antarmuka <i>Encode proses Open File Hidden</i> - Pilih file <i>hidden</i>	- Tekan tombol "Open" - Direktori file <i>hidden</i> berada	- Antarmuka <i>Common Dialog Show Open</i> - Keterangan : nama file, ukuran file, dan format file - Menu "Run" dan menu "Next" aktif	- Keluar antarmuka <i>Common Dialog Show Open</i> - Keluar keterangan : nama file, ukuran file, dan format file - Menu "Run" dan menu "Next" menjadi aktif	- Antarmuka <i>Common Dialog Show Open</i> - Keterangan : nama file, ukuran file, dan format file - Menu "Run" dan menu "Next" aktif	Handal
ENC-01-03	Pengujian memasukkan <i>password</i> dan mengecek kualitas <i>password</i> yang dimasukkan tersebut	- Masukkan <i>password</i> minimal 5 karakter maksimal 10 karakter tanpa ada pengulangan	- Masukkan <i>password</i> 'abcde' pada <i>password</i> textbox - Masukkan <i>password</i> 'abcde' pada <i>password</i> textbox <i>verify</i>	- <i>Progress Bar</i> kualitas <i>password</i> bergerak - Menu "Next" aktif	- <i>Progress Bar</i> kualitas <i>password</i> menjadi bergerak - Menu "Next" menjadi aktif	- <i>Progress Bar</i> kualitas <i>password</i> bergerak - Menu "Next" aktif	Handal
ENC-01-04	Pengujian menyisipkan data <i>hidden</i> kedalam file <i>host</i>	- Pilih menu "Next" pada antarmuka <i>Encode proses Input Password</i>	- Tekan tombol "Next"	- <i>Progress Bar</i> proses <i>Watermark Embedding</i> bergerak - File ter- <i>watermark</i> - Antarmuka <i>Encode Result</i>	- <i>Progress Bar</i> proses <i>Watermark Embedding</i> menjadi bergerak - Menghasilkan file ter- <i>watermark</i> - Keluar antarmuka <i>Encode</i>	- <i>Progress Bar</i> proses <i>Watermark Embedding</i> bergerak - File ter- <i>watermark</i> - Antarmuka <i>Encode Result</i>	Handal

DEC-02-01	Pengujian membuka file ter-watermark	- Pilih menu "Open" pada antarmuka Decode proses Open File Watermark - Pilih file ter-watermark	- Tekan tombol "Open" - Direktori file ter-watermark berada	- Antarmuka Common Dialog Show Open - Keterangan : nama file, ukuran file, dan format file - Menu "Play" dan menu "Next" aktif	- Keluar antarmuka Common Dialog Show Open - Keluar keterangan : nama file, ukuran file, dan format file - Menu "Play" dan menu "Next" menjadi aktif - Menu "Next" menjadi aktif	- Antarmuka Common Dialog Show Open - Keterangan : nama file, ukuran file, dan format file - Menu "Play" dan menu "Next" aktif - Menu "Next" aktif	Handal
DEC-02-02	Pengujian memasukkan password	- Masukkan password minimal 5 karakter maksimal 10 karakter tanpa ada pengulangan	- Masukkan password yang sama pada proses Encode, yaitu 'abcde' pada textbox password	- Menu "Next" aktif	- Menu "Next" menjadi aktif	- Menu "Next" aktif	Handal
DEC-02-03	Pengujian membaca kembali data hidden yang disisipkan kedalam file host	- Pilih menu "Next" pada antarmuka Decode proses Input Password	- Tekan tombol "Next"	- Progress Bar proses Watermark Extracting bergerak - File hidden - Antarmuka Decode Result	- Progress Bar proses Watermark Extracting menjadi bergerak - Tampil file hidden - Keluar antarmuka Decode Result	- Progress Bar proses Watermark Extracting bergerak - File hidden - Antarmuka Decode Result	Handal
HELP-03	Pengujian antarmuka Help	- Pilih menu Help - Klik "Petunjuk Pemakaian Proses Encode"	- Klik mouse	- Petunjuk pemakaian proses Encode	- Halaman petunjuk Pemakaian Proses Encode terbuka	- Petunjuk pemakaian proses Encode	Handal