ENKRIPSI DAN DEKRIPSI AUDIO FORMAT AMR DENGAN ALGORITMA KRIPTOGRAFI LOKI97

A. A. Ngurah Pradnya Adhika[1], I Ketut Gede Suhartana[2], S. Kom., M. Kom
I Made Widiartha[3], S. Si., M. Kom
Jurusan Ilmu Komputer FMIPA Universitas Udayana
Email: junk_wach@rocketmail.com

ABSTRAK

Algoritma kriptografi LOKI97 merupakan algoritma block cipher pada kriptografi yang digunakan untuk mengamankan data dengan cara menyandikan data. Algoritma LOKI97 termasuk dalam kandidat Advanced Encryption Standard (AES) yang diajukan kepada National Institute of Standards and Technology (NIST). Proses yang dilakukan dalam penyandian datanya, yaitu proses enkripsi dan dekripsi. Algoritma LOKI97 melakukan proses enkripsi dan dekripsi data 128 bit. Kunci yang digunakan bisa sepanjang 128, 192, atau 256 bit. Untuk memudahkan penggunaan algoritma LOKI97 maka dibuat suatu aplikasi algoritma LOKI97 dengan bahasa pemrograman Delphi 7.0 yang dapat mengenkripsi dan mendekripsi file audio yang berformat AMR. Berdasarkan pengujian yang dilakukan dengan RMS (Root Mean Square) dapat disimpulkan bahwa algoritma kriptografi LOKI97 dapat digunakan untuk melakukan enkripsi pada audio format AMR sehingga kerahasiaan data menjadi aman. Disamping itu aplikasi ini juga mampu mengembalikan hasil enkripsi audio format AMR.
Kata kunci : LOKI97, AMR, RMS, enkripsi, dekripsi.

ABSTRACT

LOKI97 cryptographic algorithm is a block cipher algorithm used in cryptography to secure data by encrypting the data. LOKI97 algorithm included in the candidate's Advanced Encryption Standard (AES) submitted to the National Institute of Standards and Technology (NIST). Performed in the data encoding process is encryption and decryption process. LOKI97 algorithms perform encryption and decryption of 128-bit data. The ke can be initialised using throughout 128, 192, or 256-bit. To facilitate the use of algorithms LOKI97 then made a LOKI97 algorithm application with Delphi 7.0 programming language which able to encrypt and decrypt audio files with AMR format. Based on testing performed by RMS (Root Mean Square) can be concluded that LOKI97 cryptographic algorithms can be used to encrypt the AMR audio format so that the confidentiality of the data to be safe. Besides, the application is also able to restore the encryption AMR audio format.
Keywords : LOKI97, AMR, RMS, encryption, decryption.

I. PENDAHULUAN

1.1 Latar Belakang Masalah

Sesuai dengan perkembangan zaman penggunaan multimedia sudah sangant populer di kalangan masyarakat. Terdapat berbagai jenis multimedia yang dikembangkan seperti audio, video, dan gambar bergerak. Audio memiliki berbagai jenis format seperti WAV, MP3, AMR, dan lai-lain. Audio format AMR (*Adaptive Multi Rate*) sering digunakan sebagai format untuk hasil perekaman suara. Salah satu tujuan perekaman suara yaitu untuk menyimpan hasil wawancara dari seorang narasumber. Untuk menjaga hasil rekaman yang didapat dari wawancara agar tidak terjadi kebocoran dari pihak lain diperlukan sistem pengamanan data audio hasil rekaman tersebut. Salah satu cara untuk

mengamankan data adalah dengan merubah data tersebut dalam bentuk data yang lain yang tidak dapat dimengerti oleh pihak lain, yaitu dengan cara penyandian. Dalam kriptografi terdapat beberapa algoritma yang dapat menyandikan data. Salah satu algotitma kriptografi yang termasuk dalam kandidat *Advanced Encryption Standard* (AES) yang diajukan kepada *National Institute of Standards and Technology* (NIST) adalah algoritma LOKI97. LOKI97 merupakan sebuah algoritma blok kode yang menggunakan kunci rahasia yang mengoperasikan data sebesar 128 bit dengan kunci sebesar 128, 192, atau 256 bit. LOKI97 merupakan evolusi dari LOKI89 dan LOKI91 dengan memperkuat penjadwalan kunci dengan memperbesar ukuran kunci yang digunakan. Dalam paper ini dibahas tentang enkripsi dan dekripsi data file audio dengan algoritma LOKI97, dengan pengujian tingkat keamanan menggunakan *Root Mean Square.*

1.2 Perumusan Masalah

Berdasarkan uraian latar belakang, masalah yang dapat diidentifikasi penulis adalah:

a. Bagaimana cara mengenkripsi dan mendekripsi suatu data file audio berformat AMR dengan menggunakan algoritna LOKI97.
b. Bagaimana mengetahui tingkat keamanan dari algoritma LOKI97.

II. Tinjauan Pustaka

2.1 Operator Logika

Operator biner identik dengan bit pada komputer, yang melibatkan angka 0 dan angka 1. Operator yang digunakan pada algoritma LOKI97 adalah XOR. Operator XOR digunakan untuk dua inputan. Jika kedua inputan nilainya sama, maka nilai outputnya 0, dan jika kedua inputan nilainya berbeda, maka nilai outpunya 1.

Tabel 2.1 Operator XOR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

2.2 Dasar Matematika

2.2.1 Relasi Fungsi

Definisi 2.1 Suatu relasi f dari A ke B dikatakan suatu fungsi apabila setiap $x \in A$ dipasangkan atau dipetakan pada tepat satu unsur di B.

(*Bartle, 1994*).

Definisi 2.2 $f : A \rightarrow B$ disebut fungsi *inspektif* atau satu-satu apabila

$$x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$$

atau apabila

$$f(x_1) = f(x_2) \Rightarrow x_1 = x_2$$

(*Bartle, 1994*).

Proses enkripsi dan proses dekripsi dapat dinyatakan dalam notasi matematika sebagai berikut:

$$\mathrm{E_K(P) = C} \quad \text{dan} \quad \mathrm{D_K(C) = P}$$

dan keseluruhan dapat dinyatakan sebagai:

$$\mathrm{D_K\big(E_K(P)\big) = P}$$

Relasi antara h0069mpunan P (palinteks) dengan himpunan C

(cipherteks) harus merupakan fungsi korespondensi satu-satu (*one to one relation*). Maksudnya, dalam proses dekripsi hanya ada satu elemen C yang menyatakan satu elemen P.

III. METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian dalam perancangan dan pengimplementasian aplikasi ini adalah menggunakan metode SDLC (*System Development Life Cycle*) dengan tahapan meliputi perencanaan, analisis, perancangan, implementasi, pengujian, dan pemeliharaan.

3.2 Proses *Padding*

Proses *padding* adalah suatu proses penambahan byte-byte *dummy* pada byte-byte sisa yang masih kosong pada blok plainteks, disimpan pada posisi paling terakhir.

3.3 Alat Perancangan Sistem

Alat perancangan sistem merupakan suatu alat bantu untuk memudahkan penjelasan cara kerja algoritma dan aliran data yang akan disandikan.

IV. IMPLEMENTASI DAN PEMBAHASAN

4.1 Tahap Analisis

4.1.1 Algoritma LOKI97

Algoritma LOKI97 melakukan proses enkripsi dan dekripsi data 128 bit. Kunci yang digunakan bisa sepanjang 128, 192, atau 256 bit. LOKI97 merupakan algoritma blok cipher, sehingga data keluaran akan menghasilkan panjang yang sama dengan data masukan, yaitu 128 bit pula. (Ariyus, 2008).

4.1.2 Proses Enkripsi

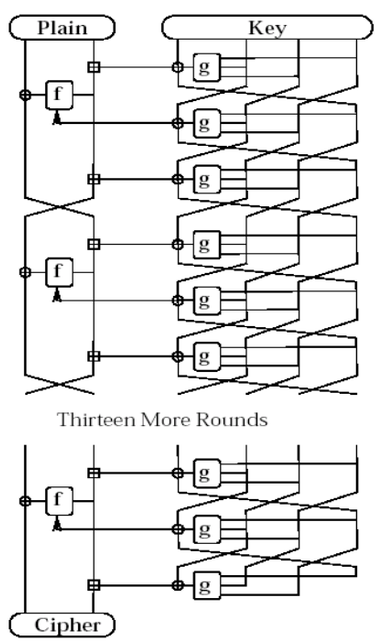
Langkah-langkah:

1. Plainteks dibagi menjadi blok-blok yang masing-masing *n* bit, dalam hal ini kita menggunakan 128 bit.
2. Masing-masing di pecah menjadi dua buah blok sama panjang (L & R).
3. Setelah plainteks dibagi menjadi dua blok sama panjang kemudian dilakukan putaran sebanyak 16 kali dengan menggunakan Jaringan Feistel.
4. Pada setiap putaran dilakukan proses operasi XOR sebagai berikut :

   R i = L i-1 XOR f(R i-1 + SK 3i-2, SK 3i-1)

   L i = R i-1 + SK 3i-2 + SK 3i
5. Hasil putaran terakhir akan mendapatkan chiperteks yang berupa gabungan dari L dan R.

Untuk lebih jelasnya proses enkripsi satu putaran dapat di lihat pada gambar 4.1



Gambar 4.1 Proses Enkripsi Yang Terjadi Pada 16 Putaran

(Sumber: Lawrie Brown, Josef Pieprzyk 1998)

Algoritma LOKI97 menggunakan jadwal pembangkitan subkunci berdasarkan jaringan Feistel yang tidak seimbang yang mengoperasikan 4 buah blok sebesar 64 bit. Dikarenakan ukuran kunci yang digunakan bisa memiliki 3 buah ukuran yang berbeda, maka inisialisasi kunci untuk masing-masing ukuran pun berbeda. Misalkan untuk kunci sepanjang 256 bit, inisialisasinya adalah

[K40|K30|K20|K10] = [Ka|Kb|Kc|Kd].

Untuk kunci dengan panjang 192 bit, maka inisialisasinya adalah dengan cara:

[K40|K30|K20|K10] = [Ka|Kb|Kc|f(Ka,Kb)]

Sedangkan untuk kunci sepanjang 128 bit, inisialisasi pembangkit subkuncinya adalah sebagai berikut :

[K40|K30|K20|K10] = [Ka|Kb|f(Kb,Ka)|f(Ka,Kb)]

Berikut ini adalah proses untuk mendapatkan subkunci Ski dengan melakukan putaran sebanyak 48 kali.

SKi = K1 i = K4 i-1 XOR gi(K1 i-1, K3 i-1,
K2 i-1)
K4 i = K3 i-1
K3 i = K2 i-1
K2 i = K1 i-1

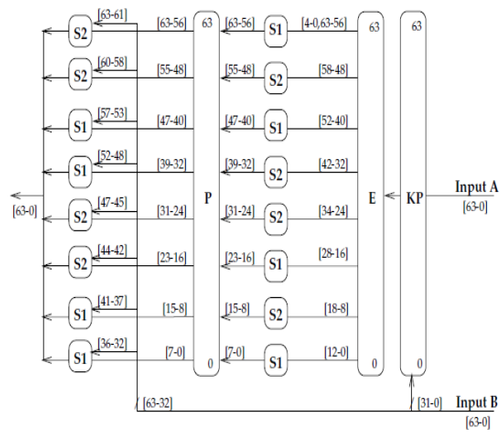
dengan I mulai dari 1 sampai dengan 48 dan :

gi(K1, K3, K2i)= f(K1+K3+ (Delta * i), K2)

Delta = $\lfloor$(sqrt(5)-1) * 263$\rfloor$ = $9E3779B97F4A7C15_{16}$

Tiga proses dari penjadwalan kunci sangat dibutuhkan untuk menghasilkan tiga sub kunci untuk setiap iterasi dari penghitungan data. Jadi jumlah keseluruhannya 48 iterasi yang dibutuhkan untuk penjadwalan kunci.

Proses deskripsi sama dengan enkripsi yaitu menggunkan sub kunci untuk membalikannya yaitu dengan cara membalikan sub kunci tersebut yaitu SK_{3i-2} dan SK_{3i}



Gambar 4.2 Fungsi F(A,B) dalam satu iterasi (Sumber: Lawrie Brown, Josef Pieprzyk 1998)

Keterangan :

- **Permutasi Kunci (A,B)**

Permutasi Kunci yang sederhana yang membagi dua masukan A sebesar 64 bit menjadi 32 bit dan menggunakan 32 bit yang paling kanan dari masukan B untuk menentukan untuk menukar pasangan bit yang cocok yaitu bit 1 atau bit 0.

- **Perluasan**

Fungsi perluasan yaitu dengan menggunakan memilih bagian berada diluar batas yaitu 13 bit (S1) atau 11 bit (S2) jadi setidaknya ada beberapa bit yang mempengaruhi dua buah S-Box secara serempak. E menghasilkan keluaran 96 bit dari masukan 64 bit sebagai berikut:

[4-0,63-56|58-48|52-40|42-32|34-24|28-16|18-8|12-0]

- **S-Box**

  Merupakan suatu table substitusi yang digunakan pada kebanyakan algoritma block cipher. S-box pertama kali digunakan pada Lucifer, kemudian DES dan setelah itu banyak algoritma menggunakan kunci Yaitu Sa( ) = [S1,S2,S1,S2,S2,S1,S2,S1] dan Sb( ) = [S2,S2,S1,S1,S2,S2,S1,S1].

  Pada Sa( ) masukkannya adalah data ditambah kunci dari keluaran E.
Sementara itu pada Sb( ) bit paling atas adalah murni kunci bit (dari bagian paling bawh, bagian paling kanan 32 bit dari B).

- **Permutasi (P)**

  Permutasi P menyebarkan keluaran S-Box melewati 64 bit secara keseluruhan dengan menggunakan pola regular latin-square. Yaitu :
[56,48,40,32,24,16,08,00,57,49,41,33,25,17,
09,01,58,50,42,34,26,18,10,02,59,51,43,35,
27,19,11,03,60,52,44,36,28,20,12,04,61,53,
45,37,29,21,13,05,62,54,46,38,30,22,14,06,
63,55,47,39,31,23,15,07]

4.1.3 Proses Dekripsi

Proses dekripsi pada algoritma LOKI97 merupakan kebalikan dari proses enkripsi. Pada proses *whitening*, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan.

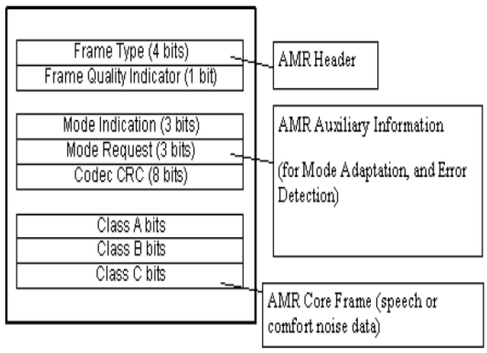
L i-1 = R i XOR f(L i – SK 3i, SK 3i-1)
R i-1 = L i – SK 3i – SK 3i-2

Sub kunci yang digunakan pada proses *whiteneing* setelah iterasi terakhir diterapkan sebelum iteasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses *whitening* sebelum iterasi pertama digunakan pada *whitening* setelah iterasi terkahir. Akibatnya, untuk melakukan dekripsi, hal yang harus dilakukan semata-mata hanyalah menerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik.

4.1.4 AMR (Adaptive Multi Rate)

AMR (Adaptive Multi Rate) merupakan file audio terkompresi, file ini berekstensi AMR, dan berukuran sangat kecil. file ini dirahasiakan dari sebuah encoder yang menyesuaikan dengan kemampuan *processing* dan *streaming* MCU/CPU, sehingga file yang dihasilkan memiliki bit-rate yang sesuai dengan MCU/CPU/Processor yang digunakan, file ini biasanya lebih banyak ditanamkan pada perangkat mobile (*Embeded File*), dan bisa ditransformasikan dalam bentuk file MP3.

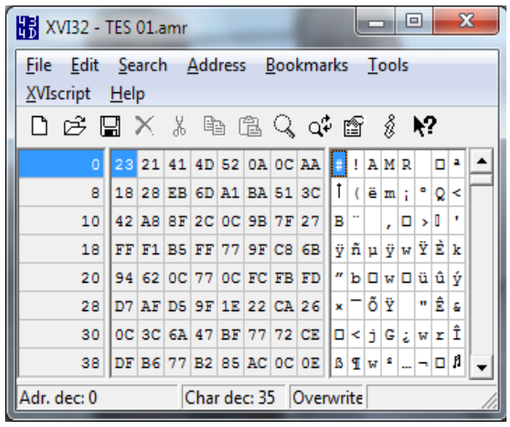
Sistem pengkodean AMR dirancang untuk beroperasi pada teknologi seluler digital GSM untuk model kanal *full rate* (22.8 kb/s) dan model kanal *half rate* (11.4 kb/s) dan untuk menjaga kualitas yang tinggi terhadap gangguan yang bervariasi dan kondisi kanal. Tidak seperti sistem pengkodean di GSM sebelumnya, yang beroperasi pada laju yang tetap dan level proteksi yang konstan, maka AMR mampu beradaptasi dengan kondisi trafik dan kanal radio.



Gambar 4.3 Struktur File AMR
(Sumber: ARIB, 1999)

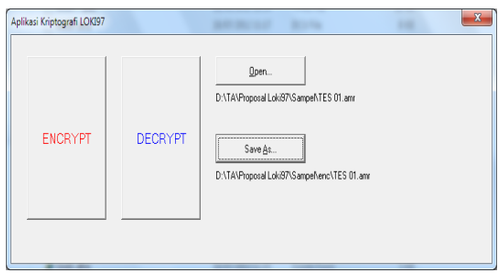
4.2 Hasil Program

Contoh file audio yang akan dienkripsi dan didekripsi berikut ini diambil dari file yang berekstensi .amr yang berukuran 3 KB (Kilo Byte). File tersebut jika dilihat dalam bentuk heksadesimal akan tampak seperti gambar di bawah ini:



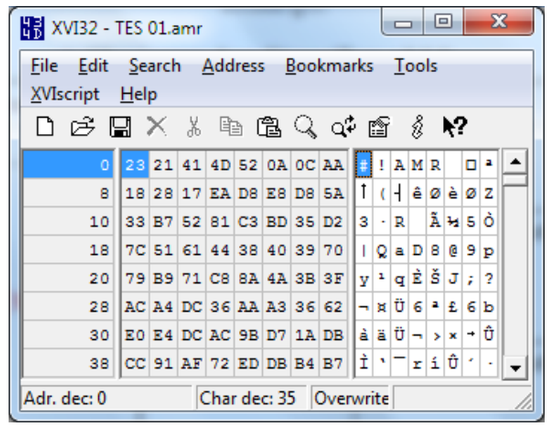
Gambar 4.4 File audio sebelum dienkripsi (plainteks)

Aplikasi yang akan ditampilkan adalah sebagai berikut :



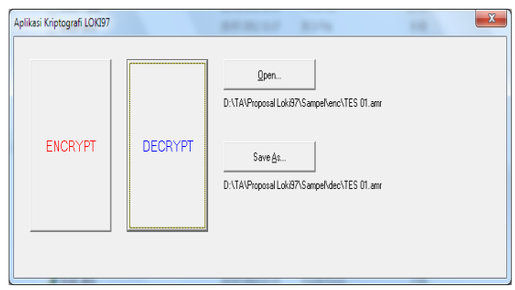
Gambar 4.5 Tampilan aplikasi untuk menyimpan hasil enkripsi

Setelah dilakukan proses enkripsi, maka file audio tetap tersimpan dalam format .amr. Namun suara yang terdengar jika diputar menjadi tidak jelas atau tidak dapat dimengerti. File audio tersebut jika dilihat kembali dalam bentuk heksadesimal adalah sebagai berikut:



Gambar 4.6 Bentuk heksadesimal file audio setelah dienkripsi (cipherteks)

Cipherteks tersebut akan didekripsikan kembali. Aplikasi yang akan ditampilkan adalah sebagai berikut:



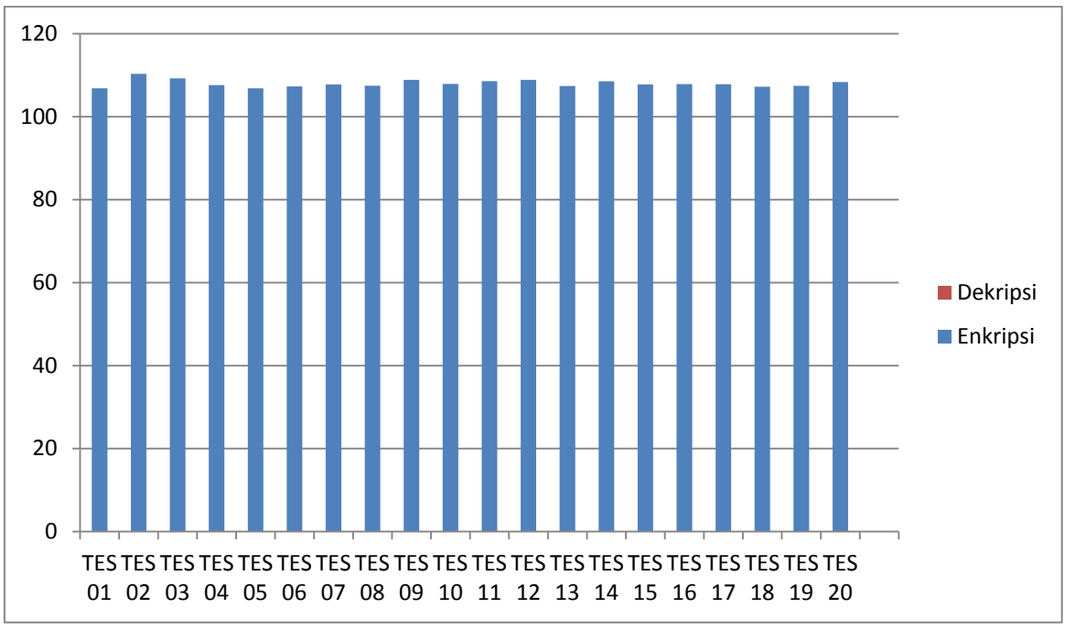
Gambar 4.7 Tampilan aplikasi untuk menyimpan hasil dekripsi

Berikut akan ditampilkan hasil uji algoritma LOKI97 sebanyak 20 sampel file audio berformat .amr dengan menggunakan RMS. Pengujian dilakukan dengan membandingkan nilai RMS antara file audio sebelum dienkripsi dengan file audio setelah dienkripsi. Kemudian membandingkan nilai RMS antara file audio sebelum

dienkripsi dengan file audio setelah didekripsi.

Tabel 4.1 Hasil uji algoritma LOKI97 sebanyak 20 sampel file audio berformat .amr dengan menggunakan RMS

No.	Contoh Audio	Nilai RMS	
		Enkripsi	Dekripsi
1	TES 01.amr (2.29 KB)	106.827504986676	0
2	TES 02.amr (2.57 KB)	110.323817130157	0
3	TES 03.amr (3.20 KB)	109.239994897946	0
4	TES 04.amr (3.76 KB)	107.636463627752	0
5	TES 05.amr (4.01 KB)	106.830827413184	0
6	TES 06.amr (4.23 KB)	107.309270992778	0
7	TES 07.amr (6.24 KB)	107.794984388871	0
8	TES 08.amr (6.67 KB)	107.469527775551	0
9	TES 09.amr (7.26 KB)	108.890891414857	0
10	TES 10.amr (8.31 KB)	107.915859313549	0
11	TES 11.amr (9.83 KB)	108.550594395536	0
12	TES 12.amr (11.93 KB)	108.870746531426	0
13	TES 13.amr (10.46 KB)	107.392000457354	0
14	TES 14.amr (12.40 KB)	108.528937021529	0
15	TES 15.amr (12.93 KB)	107.794427564051	0
16	TES 16.amr (14.24 KB)	107.855476993238	0
17	TES 17.amr (14.57 KB)	107.824715937577	0
18	TES 18.amr (16.34 KB)	107.245871462113	0
19	TES 19.amr (18.12 KB)	107.438060374936	0
20	TES 20.amr (19.19 KB)	108.363781360794	0



Gambar 4.6 Grafik hasil uji algoritma LOKI97 sebanyak 20 sampel file audio berformat .amr dengan menggunakan RMS

V. KESIMPULAN

Hasil *chipertext* pada proses enkripsi file AMR menggunakan algoritma kriptografi LOKI97 adalah relatif aman dimana dari hasil penelitian dengan menggunakan file-file yang diujicobakan menghasilkan nilai RMS di atas 100 dengan rata-rata 108,005187701993 . Hasil uji RMS terhadap file audio sebelum dienkripsi dengan file audio yang telah mengalami proses dekripsi bernilai 0. Artinya tidak ada perbedaan antara file audio sebelum dienkripsi dengan file audio yang telah mengalami proses dekripsi.

Kerahasiaan audio dapat terjaga sehingga tidak dapat dimengerti oleh orang yang tidak berhak. Disamping itu aplikasi ini juga mampu mengembalikan hasil audio yang tidak dapat dimengerti tersebut menjadi dapat dimengerti kembali.

DAFTAR PUSTAKA

[1] Andri, Yuli. 2009. "*Implementasi Algoritma Kriptografi DES, RSA, dan Algoritma Kompresi LZW pada Berkas Digital*". http://repository.usu.ac.id, (Diakses 28 Oktober 2011)

[2] ARIB. 1999. "*Mandatory Speech Codec speech processing functions ; AMR speech Codec Frame Structure".* http://www.arib.or.jp, (Diakses 31 Januari 2012)

[3] Ariyus, Dony. 2008. *Pengantar Ilmu Kriptografii (Teori, Analisis, dan Implementasi).* Yogyakarta: Andi Offset.

[4] Avon, Budiyono. 2004. *Enkripsi Data Kunci Simetris Dengan Algoritma Kriptografi Loki97.* Institut Teknologi Bandung. www.pdfio.com, (Diakses 1 Januari 2012)

[5] Bartle, Robert G. 1994. *Introduction to Real Analysis Second Edition.* Singapore: John Wiley.

[6] Brown, Lawrie dan Pieprzyk Josef. 1998. *"Introducing the new LOKI97 Block Cipher"* http://citeseerx.ist.psu.edu, (Diakses 1 November 2011)

[7] Bahri, Kusnassriyanto Saiful dan Wawan Sjachriyanto, 2008. "*Teknik Pemrograman DELPHI*", Bandung: INFORMATIKA.

[8] Rahayu, Sapty. 2005 . *Cryptografi (Suplemen Bahan Ajar Mata Kuliah Proteksi dan Teknik Keamanan Sistem Informasi).* http://bebas.vlsm.org, (Diakses 31 Oktober 2011)

[9] Ristanto, Bambang. 2006. "*Pengaruh Feeding Terhadap Tingkat Kekasaran Permukaan Terhadap Proses Pentekrapan Rata Dengan Spesimen Baja Karbon*". http://koleksi.pustakaskripsi.com, (Diakses 31 Oktober 2011)