Jurnal Nasional Informatika dan Teknologi Jaringan
E-ISSN : 2540-7600
P-ISSN : 2540-7597

APLIKASI KEAMANAN FILE AUDIO WAV (WAVEFORM) DENGAN TERAPAN ALGORITMA RSA

Raja Nasrul Fuad[1], Haikal Nando Winata[2]
Institut Teknologi Medan Fakultas Teknologi Industri, Prodi Teknik Informatika
fuad@itm.ac.id[1], ekalnata@itm.ac.id[2]

Abstrak— The WAV file format that is widely used rough on various kinds of multimedia and gaming platforms. Ease of access and technological development with a variety of media to facilitate the exchange of information to various places. The data are important and need to be kept confidential secret for a wide range of security threats so that data can be intercepted and acknowledged by third parties during the shipping process. Of these problems led to the idea to create an application data security functions can secure the data using the RSA algorithm. The programming language is C # with Visual Studio software, the processed data is a sample each byte in WAV file, the header will be the same as that originally WAV files can be played even if the information has been withheld. RSA algorithm can be implemented into a programming language that WAV files can be processed and secured the data.

Keywords— Data Security, Cryptography, RSA algorithm, WAV Files.

I. PENDAHULUAN

Citra Format wave (*.WAV) merupakan salah satu format file suara yang banyak dipakai dalam sistem operasi windows untuk keperluan game dan multimedia. Wave merupakan format kasar (raw format) dimana suara langsung direkam dan dikuantisasi menjadi digital. Kemudahan pembuatan dan pengolahan Format dasar dari file ini secara default tidak mendukung kompresi dan dikenal dengan nama PCM (Pulse Code Modulation). Pengiriman data audio terutama data audio wav dengan perkembangan teknologi informasi yang meningkat pesat seperti mudahnya internet diakses dengan berbagai media seperti pada handphone, ipad, notebook, dan sebagainya sehinngga memudahkan pertukaran informasi ke berbagai tempat. Data tersebut perlu dijaga keasliannya dan keutuhannya jika ingin dikirim karena data tersebut dapat disadap dan diketahui oleh pihak ketiga selama proses pengiriman. Untuk mengirimkan data yang bersifat penting maupun rahasia perlu dilakukan pengamanan agar tidak diketahui oleh pihak ketiga.

Algoritma RSA (Rivest Shamir Adleman) merupakan salah satu algoritma public key yang populer dipakai dan bahkan hingga saat ini Algoritma RSA masih dianggap aman karena semakin panjang kunci yang digunakan, semakin sulit untuk dipecahkan karena sulitnya memecahkan pemfaktoran bilangan prima dari suatu bilangan yang sangat besar.

Berdasarkan uraian diatas maka penulis tertarik untuk membuat tugas akhir dengan judul "Perancangan Aplikasi Keamanan File Audio WAV (Wafeform) Menggunakan Algoritma Kriptografi RSA (Rivest Shamir Adleman)".

II. METODOLOGI PENELITIAN

A. Keamanan Data

Data dapat diartikan sebagai kenyataan yang digambarkan oleh nilai, bilangan-bilangan, untaian, karakter atau symbol-symbol yang membawa arti tertentu. Informasi sendiri dapat didefinisikan sebagai hasil dari pengolahan data dalam bentuk yang lebih berguna bagi penerimanya, yang digunakan sebagai alat bantu dalam pengambilan.

Keamanan adalah keadaan bebas dari bahaya. Istilah ini dapat digunakan dengan hubungan kepada kejahatan, dan segala bentuk kecelakaan. Keamanan merupakan topik yang luas termasuk keamanan nasional terhadap seorang teroris, keamanan komputer terhadap hacker, keamanan rumah terhadap maling dan penyusup lainnya, keamanan financial terhadap kehancuran ekonomi dan banyak situasi berhubungan lainnya. Host Kompuer yang terhubung ke network, mempunyai ancaman keamanan lebih besar daripada host yang tidak berhubungan kemana-mana. Dengan mengendalikan network security resiko tersebut dapat dikurangi. [1]

B. Algoritma RSA

RSA merupakan algoritma kriptografi asimetris. Ditemukan pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Leonard Adleman. Nama RSA sendiri diambil dari inisial nama depan ketiga penemunya tersebut. [2]

Terdapat 3 algoritma pada sistem kriptografi RSA, yaitu algoritma pembangkitan kunci, algoritma enkripsi, dan algoritma dekripsi:

Algoritma RSA memiliki besaran – besaran seperti berikut : [3]

p dan q adalah bilangan prima (rahasia)
n = p x q (tidak rahasia)
ϕ(n)=(p-1) x (q-1) (rahasia)
e (kunci enkripsi) (tidak rahasia)
d (kunci dekripsi) (rahasia)
m (plainteks) (rahasia)
c (chiperteks) (tidak rahasia)

yang besar sehingga sangat sulit untuk difaktorisasi.

Bangkitkan bilangan prima besar p dan q.

$n = p\,x\,q$ .....(1.0)

$\phi(n) = (p-1)\,x\,(q-1)$ .....(1.1)

$e \overset{R}{\leftarrow} \mathbb{Z}_{\phi(n)}\ dengan\ \gcd(e, \phi(n)) = 1$ .....(1.2)

$d = e^{-1}\ pada\ \mathbb{Z}_{\phi(n)}$ .....(1.3)

$K_{publik} = (e,n)$

$K_{privat} = (d,n)$

Gbr. 1 Pembangkit Kunci RSA

Direkomendasikan besar p dan q adalah 512 bit sehingga n berukuran 1024 bit. Karena p dan q adalah bilangan prima, ϕ(n)=(p-1) x (q-1). Kemudian pilih sebuah integer e dipilih secara acak dari Z_(ϕ(n)) yang memenuhi gcd(e, ϕ(n)) sehingga e merupakan generator pada Z_(ϕ(n)). Selanjutnya algoritma pembangkit kunci RSA menghitung d invers perkalian e pada Z_(ϕ(n)). Pada akhirnya algoritma pembangkit kunci RSA menerapkan (e, n) sebagai kunci publik dan d sebagai kunci privat atau tetap di rahasiakan.

C. Enkripsi

Setelah kunci publik Kpublik dibangkitkan oleh pendekripsi (Bob) maka sembarang orang dapat menggunakan kunci publik Bob untuk mengirim teks sandi ke Bob. Algoritma enkripsi RSA menggunakan fungsi eksponensial dalam modular n seperti yang diberikan oleh gambar 2 (Rifki Sadikin,2012： 251)

Input : $K_{publik} = (e,n), P \in \mathbb{Z}_n$

Output : $C \in \mathbb{Z}_n$

$C = P^e \bmod n$ .....(2.0)

Gbr. 2 Enkripsi RSA

D. Dekripsi

Jika Bob mendapatkan teks audio yang dienkripsi dengan kunci publik Bob maka Bob dapat menggunakan kunci privatnya untuk mengembalikan teks asli. Sama seperti enkripsi, algoritma dekripsi RSA merupakan eksponensial modular n dengan menggunakan kunci privat seperti yang diberikan oleh Algoritma 2.3.

Algoritma 2.3 Dekripsi RSA

Input : $K_{privat} = d, K_{publik} = (e,n),\ C \in \mathbb{Z}_n$

Output : $P \in \mathbb{Z}_n$

$P = C^d \bmod n$ .....(3.0)

Gbr. 3 Dekripsi RSA

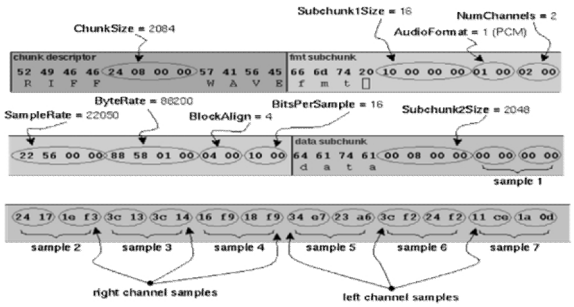
E. Data Audio

Audio (Suara) adalah fenomena fisik yang dihasilkan oleh getaran suatu benda yang berupa sinyal analog dengan amplitude yang berubah secara kontiniu terhadap satuan waktu yang disebut frekuensi. [4].

F. File WAV

WAV adalah format audio standar Microsoft dan IBM untuk personal computer (PC), biasanya menggunakan coding PCM (Pulse Code Modulation), WAV adalah data tidak terkompres sehinngga seluruh sample audio disimpan semuanya di harddisk. Software yang dapat menciptakan WAV dari analog sound misalnya adalah Windows Sound Recorder. File audio ini jarang sekali digunakan di internet karena ukurannya yang relative besar dengan batasan maksimal untuk file WAV adalah 2GB. [5]

Bebagai macam format dan struktur file audio. Misalnya file Wav memiliki struktur seperti Gambar 2.1. [5]



Gbr. 4 Struktur File WAVE dalam Bentuk Hexa

Pada struktur file WAV di atas terdiri dari:

1. Chunk Descriptor yang terdiri dari data: 52 49 46 46 28 08 00 00 57 41 56 45.
2. Fmt subChunk yang terdiri data subChunkIsize, audioFormat, numChannel, sampleRate, byteRate dan BlockAlign yaitu:
   66 6D 74 20 10 00 00 00 01 00 02 00 22 56 00 00 ## 50 01 00 04 00 10 00
3. Data subChunk yang terdiri dari subChunk2Size serta sample-sample yaitu:
   64 61 74 61 00 0##1 00 00 00 00#2 24 17 1E F3#3 3C 13 3C 14 #4 16 F9 18 F9 34 E7 23 A6 3C F2 24 F2 24 F2 11 CE 1A 0D

III. METODE PENELITIAN

A. Analisis Sistem Berjalan

Pada tahap ini akan diuraikan aspek matematis algoritma RSA dengan melakukan pengujian secara manual, dengan membentuk kunci dan melakukan enkripsi dan dekripsi terhadap file audio WAV. Setelah itu di implementasikan pada C#.

B. Proses Pembuatan Kunci

Penulis akan paparkan studi kasus untuk melakukan operasi pembuatan kunci RSA. Dengan memilih bilangan prima P dan Q secara acak sehingga menghasilkan kunci (e, n) dan (d, n). Langkah-langkah proses pembuatan kunci adalah sebagai berikut :

Langkah pertama dalam pembangkitan kunci adalah memilih secara acak dua bulangan prima. Dua bilangan prima yang penulis pilih adalah P = 17 dan Q = 19.

Langkah kedua mencari nilai n, dengan rumus n = p x q sehingga didapatlah :

n = 17 x 19 = 323

Langkah ketiga adalah mencari nilai ϕ(n), dengan persamaan

ϕ(n)=(p-1) x (q-1). Jadi didapatkan hasil :

ϕ(n)=(17-1) x (19-1)

ϕ(n)=16 x 18=288

Langkah keempat adalah menentukan nilai e, dimana nilai e harus relatif prima terhadap ϕ(n) atau gcd (e,ϕ(n))=1. Diantara nilai 1 - ϕ(n) penulis memilih nilai e=59. Dengan penjelasan sebagai berikut :

gcd (59,288)

TABEL I
PERHITUNGAN GREATEST COMMON DEVISOR PADA KUNCI E

A	B	Q = A/B	R = A Mod B
59	288	0	59
288	59	4	52
59	52	1	7
52	7	7	3
7	3	2	1
3	1	3	0
1	0		

Jadi gcd(59, 288) = 1

Langkah kelima adalah menghitung d, dengan persamaan d=e^(-1) pada Z_ϕ(n) . Dari hasil perhitungan 1 - ϕ(n) ditentukanlah nilai d=83. Dengan menggunakan algoritma Extended Euclid sebagai berikut :

Inisialisasi

$A \leftarrow a;\ B \leftarrow b;$
$S_1 \leftarrow 1;\ S_2 \leftarrow 0;$
$T_1 \leftarrow 0;\ T_2 \leftarrow 1;$

Perhitungan

$Q \leftarrow A / B;$
$R \leftarrow A - Q * B;$
$A \leftarrow B, B \leftarrow R;$
$S \leftarrow S_1 - Q * S_2;$
$S_1 \leftarrow S_2, S_3 \leftarrow S;$
$T \leftarrow T_1 - Q * T_2;$
$T_1 \leftarrow T_2;\ T_2 \leftarrow T;$

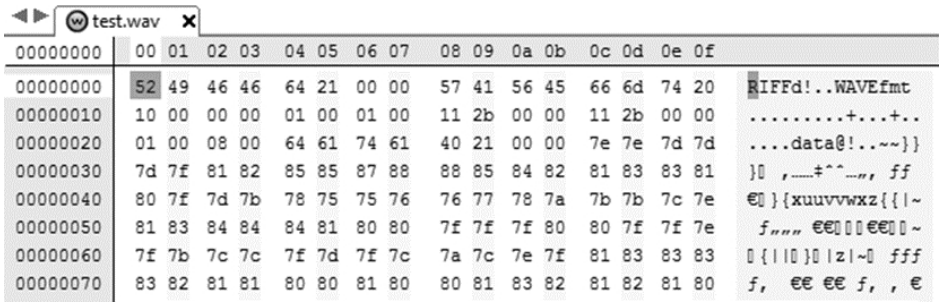
TABEL II
PERHITUNGAN KUNCI D DENGAN ALGORITMA EXTENDED EUCLID

k	A	B	Q	R	S_1
init	83	288			1
1	83	288	0	83	1
2	288	83	3	39	0
3	83	39	2	5	1
4	39	5	7	4	-3
5	5	4	1	1	7
6	4	1	4	0	-52
7	**1**	0			**59**
S_2	**S**	**T_1**	**T_2**	**T**	
0		0	1		
0	1	0	1	0	
1	-3	1	0	1	
-3	7	0	1	-2	
7	-52	1	-2	15	
-52	59	-2	15	-17	
59	-288	15	-17	83	
-288		**-17**	83		

Pada iterasi terakhir nilai gcd(83, 288), s dan t ditemukan, yaitu nilai A = 1, S1 = 59, dan T1 = -17. Jika mengacu pada persamaan s x a+t x b=gcd 〖(a,b)〗 dapat diuji bahwa 59 x 83+ -17 x 288=1.

C. Plainteks

Sebelum melakukan enkripsi dan dekripsi terlebih dahulu dipilih file WAV yang akan di masukkan ke dalam proses pengamanan informasi. Penulis akan mengambil contoh sebuah file WAV seperti gambar 4.



Gbr. 5 Contoh File WAV "test.wav" degan bilangan HEX

Dari gambar 4 dapat dilihat header dan data penyusun file WAV, di dalam sistem yang akan dirancang header file WAV akan di biarkan saja seperti awal adanya. Kemudian data yang ada di struktur file WAV ini lah yang akan di enkripsikan. Dari gambar diatas maka di dapatkan :

TABEL III
PLAINTEKS PADA STRUKTUR FILE WAV

Tipe	Value (HEX)
ChunkID	52 49 46 46
ChunkSize	64 21 00 00
Format	57 41 56 45
SubChunk1ID	66 6D 74 20
SubChunk1Size	10 00 00 00
AudioFormat	01 00
NumChannels	01 00
SampleRate	11 2B 00 00
ByteRate	11 2B 00 00
BlockAlign	01 00
BitsPerSample	08 00
SubChunk2ID	64 61 74 61
SubChunk2Size	40 21 00 00
Data	7E 7E 7D 7D 7D 7F 81 82 (…)

D. Proses Enkripsi RSA

Dari proses pembangkitan kunci dan plainteks yang dipaparkan di atas, kunci yang akan digunakan dalam melakukan enkripsi adalah n = 323 dan e = 59. Jika m1 sampai m8 adalah blok plainteks, maka cipherteksnya adalah c1
sampai c8 dengan menggunakan rumus C= P^e mod n maka di dapatkan :

C_1 = ⟦m_1⟧ ^e mod n
.
.
C_8= ⟦m_8⟧ ^e mod n

C1 = 12659 mod 323 = 65
C2 = 12659 mod 323 = 65
(C3 … C7)
C8 = 13059 mod 323 = 80

Hasil dari proses enkripsi tersebut adalah : 65 65 311 311 311 223 173 80.

E. Cipherteks

Setelah proses enkripsi selesai maka akan mendapatkan cipherteks dimana cipherteks yang akan di masukkan kedalam file WAV ter-enkripsi adalah cipherteks yang diberikan pemisah (splitter) yaitu karakter spasi (" ") bernilai byte = 20 dan data cipherteks yang akan di kembalikan adalah bentuk string ke byte. Contoh:

TABEL IV
INTERPRETASI STRING KE BYTE

STRING		BYTE
65 65 311 311 311 223 173 80	→	36 35 20 36 35 20 33 31 31 20 33 31 31 20 33 31 31 20 32 32 33 20 31 37 33 20 38 30

Karena tipe data byte maksimal sampai dengan nilai 255 maka hasil enkripsi setiap satu karaker akan langsung diubah menjadi data byte dapat kita lihat dari tabel 3.7 angka 6 di ubah menjadi byte 36 dan angka 5 diubah menjadi byte 35 dan begitu juga nilai selanjutnya.

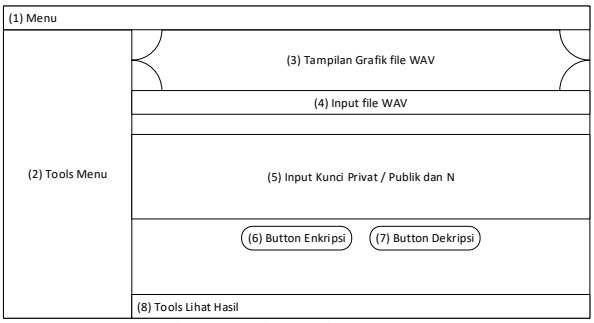
F. Proses Dekripsi RSA

Setelah pesan ter-enkripsi sampai kepada sipenerima maka file WAV akan di dekripsi menggunakan kunci privat yang telah disiapkan pada langkah pembuatan kunci sebelumnya. (d = 83, n = 323)
Cipherteks = 65 65 311 311 311 223 173 80
m_1 = ⟦c_1⟧ ^e mod n
.
.
m_8= ⟦c_8⟧ ^e mod n

m1 = 6583 mod 323 = 126
m2 = 6583 mod 323 = 126
(m3 … m7)
m8 = 8083 mod 323 = 130

Maka plainteks yang akan di dapatkan adalah :
7E 7E 7D 7D 7D 7F 81 82

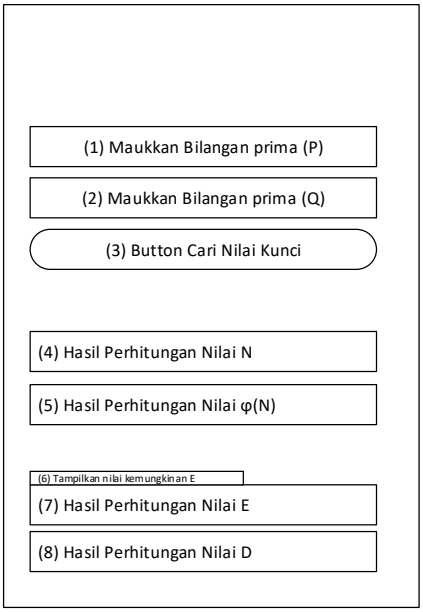
G. Rancangan Interface Halaman Utama Enkripsi dan Dekripsi



Gbr. 6 Interface Halaman Utama

Keterangan:
1. Bagian menu akan menampilkan About untuk pergi ke form tentang pembuat aplikasi.
2. Tools menu berisikan LinkLabel untuk pergi ke form Pembangkit Kunci, GCD Calculator, Extended Euclid Calculator, WAV Voice Recorder dan File Email Sender.
3. Untuk menampilakan grafik frekuensi dari file audio WAV digunakan tools “chart”.
4. File WAV yang akan di enkripsi ataupun dekripsi akan di pilih.
5. Melalui bagian ini untuk dimasukkan kedalam proses pengamanan informasi.
6. Kunci RSA terdiri dari kunci publik (e, n) dan kunci privat (d, n) di dalam aplikasi untuk memasukkan bagian kunci digunakan pada bagian 5.
7. Tombol untuk memulai proses pengamanan informasi dengan cara enkripsi.
8. Tombol untuk memulai proses pengamanan informasi dengan cara dekripsi.
9. Jika proses pengamanan informasi enkripsi ataupun dekripsi sudah selesai maka hasil output dapat dilihat menggunakan tombol buka folder, lihat grafik, atupun bisa di play secara langsung.

H. Rancangan Interface Halaman Pembuatan Kunci



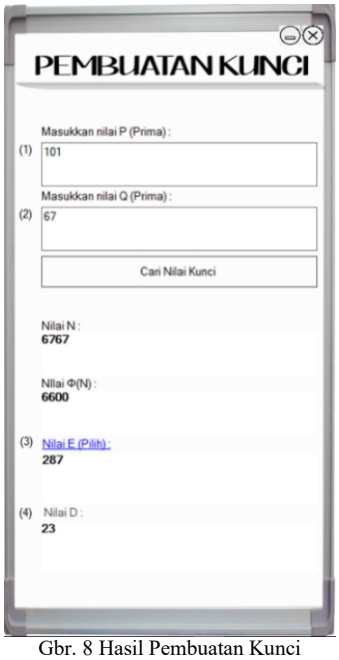
Gbr. 7 Rancangan Interface Halaman Pembuatan Kunci

Keterangan :
1. Textbox untuk memasukkan nilai P bilangan prima.
2. Textbox untuk memasukkan nilai Q bilangan prima.
3. Button untuk mencaari nilai kunci sesuai dengan bilangan prima P dan Q.
4. Textbox hasil perhitungan nilai N.
5. Textbox hasil perhitungan nilai ϕ(N).
6. LinkedLabel untuk memunculkan form daftar nilai kemugkinan E.
7. Textbox hasil perhitungan nilai E.
8. Textbox hasil perhitungan nilai D.

IV. HASIL DAN PEMBAHASAN

A. Pembuatan Kunci RSA

Selebum memulai melakukan peroses pengamanan data maka beberapa persiapan harus dilakukan salah satunya adalah kunci RSA. Kunci publik dan kunci privat yang hanya boleh di ketahui oleh pendekripsi bisa di buat melalui form ini dengan mengisi bilangan prima P dan Q.

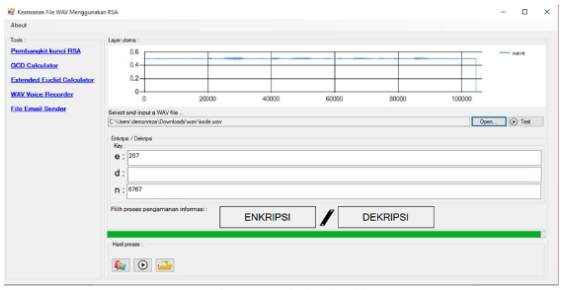


Gbr. 8 Hasil Pembuatan Kunci

Pada gambar 8 dapat dilihat bahwa hasil pembangkit kunci dari bilangan prima P = 101 dan Q = 67 adalah N = 6767, ϕ(N) = 6600, E = 287 dan D = 23. Kemudian hasil yang di dapat digunakan ke halaman utama dengan cara mencatat hasil yang telah di dapatkan.

B. Enkripsi File WAV

Setelah kunci publik dan kunci privat di buat maka proses selanjutnya adalah melakukan enkripsi data dengan menggunakan kunci publik. Maka di form utama di input-kan seperti gambar berikut ini :



Gbr. 9 Enkripsi File WAV

Langkah pertama yang harus dilakukan adalah memilih (input) file WAV yang akan di enkripsi, disini penulis mengambil contoh file kode.wav sengan cara mengklick tombol "Open" dan menentukan di mana letak file WAV disimpan, kemudian penulis memasukkan kunci publik (e, n) di textbox yang ber-label "e" dan "n" sesuai dengan kunci yang telah didapatkan pada bahasan sebelumnya yaitu (287, 6767).

Langkah Kedua adalah melakukan proses pengamanan informasi enkripsi data WAV dengan cara mengklick tombol "ENKRIPSI". Maka progressbar (bar proses) akan berjalan dari 0 hingga 100% (terlihat pada gambar bar berwarna hijau) menyatakan bahwa proses enkripsi berkas WAV sedang berjalan hingga akhirnya muncul pesan "Complete! byte error = 0" dan bar menjadi 100%.

Langkah Ketiga adalah hasil proses setelah proses enkripsi berhasil untuk melihat hasilnya ada 3 pilihan yaitu

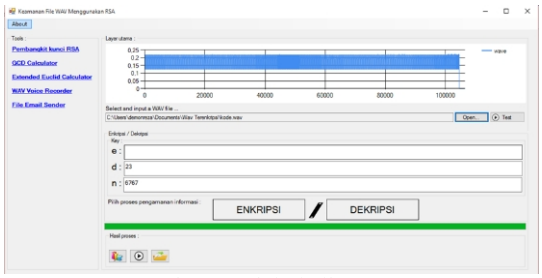
1. Open File Folder
2. Play File
3. Show Graph

Ketiga pilihan tersebut memiliki fungsi masing-masing yang pertama Show Graph berfungsi menampilkan gambar frekuensi suara hasil enkripsi berkas WAV. Kedua Play File berfungsi memainkan berkas suara hasil enkripsi menggunakan Sound Player standar yang telah di setting dalam sistem operasi. Open File Folder berfungsi untuk membuka folder penyimpanan file WAV yang telah terenkripsi.

Gbr. 10 Outpun Grafik Suara Hasil Enkripsi

C. Dekripsi File WAV

Setelah Alice mengirmkan teks sandi kepada Bob maka langkah selanjutnya adalah melakukan dekripsi teks sandi untuk mendapatkan teks asli. Seperti langkah enkripsi pada bahasan sebelumnya langkah dekripsi dapat dilakukan dengan cara meng-input file WAV yang telah dienkripsi dan memakai kunci privat, untuk jelasnya pada gambar 11.



Gbr. 11 Dekripsi File WAV

Dapat dilihat pada gambar 11 grafik berkas WAV terenkripsi seperti persegi yang teratur frekuensinya menunjukkan getaran dengan amplitude yang berubah secara kontiniu terhadap satuan waktu yang terlihat sama dari awal hingga akhir sehingga menghasilkan suara yang tidak jelas. Ini merupakan langkah pertama dalam melakukan proses dekripsi.

Langkah berikutnya sampai akhir hasil proses sebenarnya hamper sama dengan langkah enkripsi yaitu memasukkan kunci privat (d, n) kedalam textbox ber-label "d" dan "n". Kunci yang dimasukkan adalah kunci privat yang dibuat sebelumnya yaitu (23, 6767).

Langkah ketiga adalah memulai proses pengamanan informasi dekripsi dngan mengklick tombol "DEKRIPSI". Proses akan ditandai dengan progressbar hingga 100% dan muncul pesan "Complete! byte error = 0".

Hasil dari proses dekripsi bisa dilihat dengan 3 pilihan yaitu Show Graph, Play File, Open File Folder.

Output yang dihasilkan bila didengar secara langsung menggunakan Pilihan Play File, Sedangkan untuk melihat output lainnya terlihat pada gambar hasil dekripsi sebagai berkut :

Gambar 12 Output Grafik Frekuensi Dekripsi File WAV

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pengujian dan pembahasan dari bab – bab yang telah dibahas sebelumnya adapun yang dapat penulis simpulkan sebagai berikut :

1. Perangkat lunak (software) dapat melakukan penyandian data audio dengan menerapkan algoritma RSA dan struktur data audio WAV.
2. Ukuran berkas audio WAV menjadi bertambah besar setelah dilakukan enkripsi menggunakan algoritma RSA berdasarkan besar kunci yang digunakan.

B. Saran

Saran – saran pengembangan yang penulis dapat berikan untuk penelitian ini adalah :

1. Pembagian blok dalam menerapkan algortima RSA di dalam perangkat lunak tidak hanya sebatas ukuran byte, melainkan bisa dibuat maksimal sebanyak nilai N pada kunci RSA yang digunakan, agar perangkat lunak dapat mengurangi waktu pengerjaan enkripsi ataupun dekripsi.
2. Perangkat lunak yang dihasilkan dari penelitian ini masih sederhana, sehingga perlu dilakukan pengembangan lebeih lanjut misalnya tidak hanya berkas WAV malainkan berkas MP3, FLAC, WMA, AAC dan melainkan juga dapat digabung untuk berkas teks, gambar dan video.
3. Penggunaan dua atau lebih algoritma akan membuat berkas cipherteks semakin sulit dipecahkan. Untuk itu saran dari penulis agar membangun sistem yang lebih kuat dengan menggunakan dua atau lebih algoritma.

REFERENCE

[1] Sitohang (2013) Perangkat Aplikasi Keamanan Data Text Menggunakan Electronic CodeBook dengan Algoritma DES. V, 2 – 3.
[2] Wibowo (2009) Penerapan Algoritma Kriptografi Asimetris RSA untuk Keamanan Data di Oracle. 5, 6-7.
[3] Dewanto Joko (2013) Pembuatan Aplikasi SMS Kriptografi RSA dengan Android. 10, 274.
[4] Nurasyiah (2013) Perancangan Aplikasi Kompresi File Audio Dengan Algoritma Aritmic Coding. IV, 104 – 106.
[5] Rahendi (2012) Analisis dan Implementasi Kompresi File Audio dengan Menggunakan Algoritma Run Length Encoding (RLE). 1, 3 - 4..