ADA Journal of Information System Research
Volume 1, No 3, June 2024 Page: 121−128
ISSN 3025-9568 (media online)

Implementasi Algoritma Lucifer untuk Autentikasi Sound Chat

Ade Sani Afrianda

[2] Program Studi Teknik Informatika, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma, Medan, Indonesia
Email: adsani7252@gmail.com

Abstrak−*Sound chat* merupakan salah satu cara yang digunakan untuk mengirim sebuah pesan melalui aplikasi *chatting*, dimana *sound chat* berisi pesan dalam bentuk suara.Terdapat permasalahan pada proses pengiriman *sound chat*, dimana pada proses pengiriman dapat terjadi krusakan atau manipulasi isi *sound chat* yang dapat dilakukan oleh pihak-pihak yang tidak berwenang terhadap *sound chat* tersebut. Dalam hal ini diperlukan sebuah keamanan untuk menjaga keaslian *sound chat* agar terhindar dari kerusakan atau manipulasi yang dapat dilakukanoleh pihak-pihak yang tidak berwenang. Teknik kriptografi dapat digunakan untuk menjaga keaslian *sound chat* dengan proses enkripsi serta dekripsi sehingga dapat memanimalisir kerusakan atau manipulasi *sound chat* yang dapat dilakukan oleh pihak-pihak yang tidak berwenang. Algoritma *lucifer* adalah algoritma yang dapat melakukan teknik kriptografi, dimana algoritma *lucifer* melakukan enkripsi pada data yang telah terbagi dalam *block-block*. Penelitian ini mengggunakan algoritma *lucifer* untuk proses enkripsi dan dekripsi. Sebuah *soud chat* akan di enkripsi menggunkan algoritma *lucifer* sehingga menghasilkan *ciphersound*. Proses ini dapat meningkatkan keamanan dan keaslian *sound chat* pada proses pengiriman serta tidak akan menimbulkan kecurigaan dari pihak lain yang tidak bertanggung jawab untuk mengetahui keaslian *sound chat* tersebut.

Kata Kunci: *Kriptografi; Lucifer; Sound Chat*

Abstract−Voice chat is one of the ways used to send a message through a chat application, where voice chat contains messages in the form of voice. There are problems in the process of sending voice chats, where in the delivery process there can be damage or manipulation of sound that can be done by parties who are not involved in the voice chat. In this case, security is needed to maintain the authenticity of the chat voice to avoid damage or what can be done by non-existent parties. Cryptographic techniques can be used to maintain the authenticity of voice chats with encryption and decryption processes so that they can minimize damage or damage to voice chats that can be done by parties who do not. The lucifer algorithm is an algorithm that can perform cryptographic techniques, where the lucifer algorithm encrypts data that has been divided into blocks. This study uses the lucifer algorithm for the encryption and decryption process. A voice chat will be encrypted using the lucifer algorithm to produce ciphersound. This process can increase the security and authenticity of the chat voice in the sending process and will not arouse suspicion from other parties who are not responsible for knowing the authenticity of the chat voice.

Keywords: *Cryptography; Lucifer; Sound Chat*

1. PENDAHULUAN

Aplikasi chatting yang berfungsi sebagai media berkomunikasi juga digunakan dalam kegiatan komunikasi. Kemajuan sistem informasi di bidang ilmu komputer dan telekomunikasi sangatlah berkembang dan maju dengan pesat. Teknologi komputer sangat dibutuhkan untuk pekerjaan perkantoran ataupun personal menjadi lebih cepat, pertukaran data-data dan pesan semakin mudah dilakukan. Dalam hal ini diperlukan adanya keamanan data untuk mengamankan data dari berbagai ancaman yang mungkin akan timbul. Ada beberapa cara melakukan pengamanan data ataupun pesan, di antaranya adalah dengan menggunakan teknik penyamaran data yang disebut dengan kriptografi.

Horst Feistel menemukan cara menyimpan data dengan melakukan enkripsi pada data yang telah terbagi dalam block-block. Beliau saat itu menemukan apa yang disebut Lucifer algorithm yang merupakan block cipher pertama yang menjadi dasar dari block cipher-block cipher selanjutnya. Lucifer juga memiliki banyak variant yang salah satunya menjadi DES. Salah satunya yaitu adalah dikembangkannya Lucifer dengan 16-round Feistel network yang juga bekerja di 128 bit block dan menggunakan 128-bit key. Versi ini namun masih bisa dipecahkan dengan 236 plaintext dan 236 kompleksitas [1].

Berdasarkan masalah yang terdapat diatas, maka diperlukan peningkatan keamanan agar tidak mudah dipecahkan oleh cryptanalysis. Adapun cara untuk mendeteksi keaslian sound chat adalah dengan cara mengkombinasikan ilmu kriptografi dengan Algoritma Lucifer. maka dapat meminimalisir adanya serangan dari pihak yang tidak bertanggung jawab dengan memanfaatkan kelamahan dari Algoritma Lucifer. Sehingga diharapkan tidak akan menimbulkan kecurigaan dari pihak lain yang tidak bertanggung jawab untuk mengetahui keaslian sound chat tersebut.

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Dekripsi menggunakan kunci dekripsi mendapatkan kembali data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang mengandalkan kerahasiaan algoritma

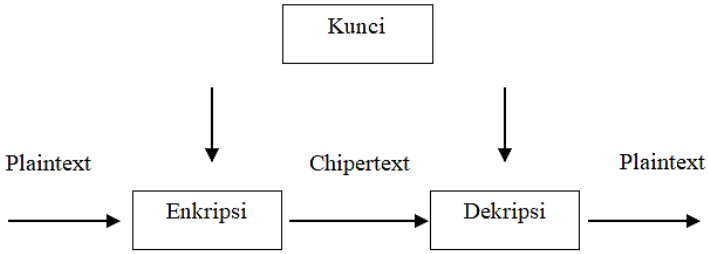
ADA Journal of Information System Research
Volume 1, No 3, June 2024 Page: 121−128
ISSN 3025-9568 (media online)

dianggap sesuatu yang tidak baik. Rahasia terletak dibeberapa parameter yang digunakan, jadi kunci ditentukan oleh parameter. Parameter yang menentukan kunci dekripsi itulah yang harus dirahasiakan (parameter menjadi ekuivalen dengan kunci). Enkripsi adalah proses pengacakan naskah asli (*plaintext*) menjadi naskah acak (*ciphertext*) yang sulit untuk dibaca oleh seseorang yang tidak mempunyai kunci dekripsi. Sedangkan dekripsi adalah kebalikan dari enkripsi [4].

Algoritma kriptografi adalah suatu fungsi matematis yang digunakan untuk melakukan enkripsi dan dekripsi. Ada dua macam algoritma kriptografi, yaitu algoritma simetri (*symmetric algorithms*) dan algoritma asimetri (*asymmetric algorithms*).

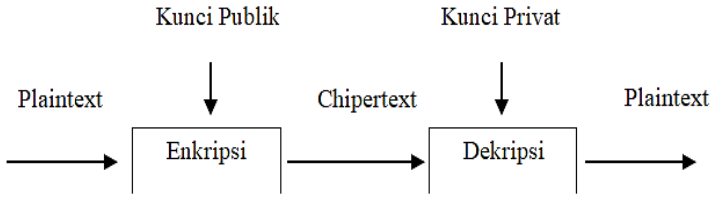
1. Kriptografi Kunci Simetri
   Kriptografi simetri disebut juga sebagai kriptografi konvensional. Kriptografi simetri adalah kriptografi yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Kriptografi simetri sering disebut sebagai algoritma kunci rahasia, algoritma kunci tunggal, atau algoritma satu kunci dan mengharuskan pengirim dan penerima menyetujui suatu kunci sebelum mereka dapat berkomunikasi dengan aman [4].

   Berdasarkan uraian di atas, maka dapat disimpulkan bahwa kriptografi kunci simetri merupakan algoritma kunci tunggal atau algoritma satu kunci yang digunakan oleh dua orang atau dua golongan ketika ingin bertukar informasi, yang dalam hal ini disebut sebagai pengirim dan penerima untuk melakukan proses enkripsi dan dekripsi sehingga mereka dapat bertukar informasi atau berkomunikasi dengan aman. Berikut adalah gambar yang mengilustrasikan kinerja dari proses enkripsi dan dekripsi kunci simetri



2. Kriptografi Kunci Asimetri
   Kriptografi kunci asimetri yang sering disebut juga kriptografi kunci publik adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma asimetri ini disebut kunci publik karena kunci untuk enkripsi dapat dibuat publik yang berarti semua prang boleh mengetahuinya. Sembarang orang dapat menggunakan kunci enkripsi tersebut untuk mengenkrip pesan, namun hanya orang tertentu yaitu calon penerima pesan dan sekaligus pemilik kunci dekripsi yang merupakan pasangan kunci publik, yang dapat melakukan dekripsi terhadap pesan tersebut. Dalam sistem ini, kunci enkripsi disebut kunci publik, sementara kunci dekripsi disebut dengan kunci privat [4].

   Berdasarkan uraian di atas, maka dapat disimpulkan bahwa kriptografi kunci asimetri merupakan algoritma yang menggunakan kunci yang berbeda untuk melakukan proses enkripsi dan dekripsi, kunci yang berbeda tersebut ialah kunci public dan kunci privat. Kunci public digunakan pada saat ingin melakukan proses enkripsi, sedangkan kunci privat digunakan untuk proses dekripsi. Berikut adalah gambar yang mengilustrasikan kinerja dari proses enkripsi dan dekripsi kunci asimetri.



Gambar 2. Kriptografi Kunci Asimetri

2.2 Algoritma Lucifer

Algoritma Lucifer merupakan algoritma *block cipher* pertama yang menggunakan *feistel cipher*. Algoritma ini ditemukan oleh Horst Feistel, dan menjadi cikal bakal algoritma DES. Dalam melakukan pengenkripsian, Lucifer mengenkripsi 128 bits dari *plainteks* dan menggunakan kunci sebesar 128 bits juga. Fungsi-f digunakan oleh lucifer di dalam proses enkripsi di dalam jaringan feistel yang digunakan [5].

a. Langkah-langkah proses enkripsi algoritma lucifer :
   1. *Plaintext* dibagi menjadi dua bagian yaitu *Left bit* (L0) dan *Right bit* (R0).
   2. Lakukan perhitungan pertama dengan rumus :

$$R_{i+1} = L_i \oplus F(R_i , K_i) \quad (1)$$

   Untuk mecari nilai R dan hasilnya akan digunakan untuk R_1 di putaran kedua.

ADA Journal of Information System Research
Volume 1, No 3, June 2024 Page: 121−128
ISSN 3025-9568 (media online)

3. Untuk mencari nilai L menggunakan rumus :

$$L_{i+1} = R_i \quad (2)$$

4. Setelah setiap putaran kecuali yang terakhir, bagian kanan dan kiri *block* ditukar.

b. Langkah-langkah proses dekripsi algoritma lucifer :

1. *Ciphertext* dibagi menjadi dua bagian yaitu *Right bit* (Rn) dan *Left bit* (Ln).
2. Lakukan perhitungan pertama dengan rumus :

$$L_{i-1} = R_i \oplus F(L_i , K_i) \quad (3)$$

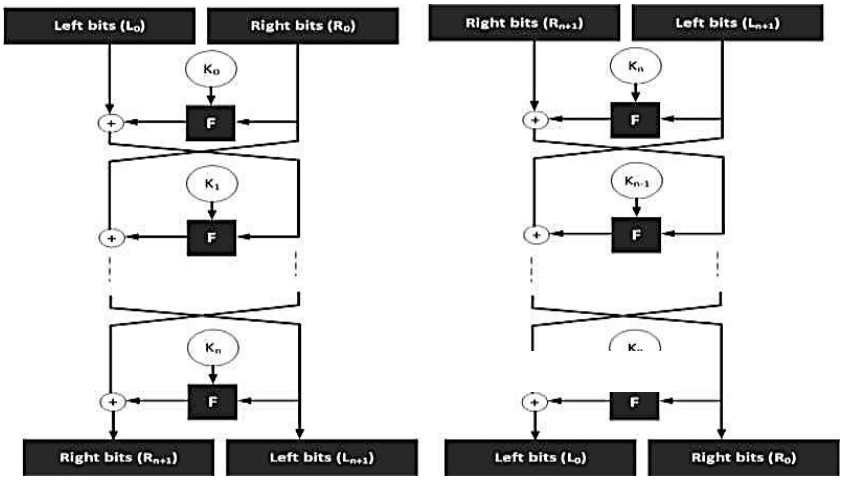
Untuk mecari nilai L dan hasilnya akan digunakan untuk L_i di putaran kedua.

3. Untuk mencari nilai R menggunakan rumus :

$$R_{i-1} = L_i \quad (4)$$

4. Setelah setiap putaran kecuali yang terakhir, bagian kanan dan kiri *block* ditukar.

Proses di atas masing-masing diulang sebanyak 16 ronde. Dari data di atas, tampak bahwa algoritma lucifer, yang merupakan algoritma pertama yang memanfaatkan *feistel cipher* telah menggunakan fungsi permutasi dan iterasi dalam pengimplementasiannya. Fungsi-fungsi tersebut dibuat dengan tujuan untuk menyulitkan kriptoanalis dalam memecahkan cipherteks yang dihasilkan [6]. Berdasarkan uraian di atas, maka dapat disimpulkan bahwa Algoritma Lucifer merupakan algoritma *block cipher* pertama yang menggunakan *Feistel network*, yang proses kerjanya menggunakan *bit-rotation*, *substitution*, dan *permutation.* Skema dari penggunaan jaringan feistel pada lucifer dapat dilihat pada gambar di bawah ini.



Gambar 3. Skema Enkripsi Dekripsi Lucifer

2.3 Key Byte Access Schedule

Key Byte Access Schedule adalah aturan penjadwalan pengaksesan kunci pada algoritma Lucifer, dimana panjang maksimal *Message Byte* adalah 8, dan C-I-D *Round* mempunyai kapasitas maksimal 16, serta *Key Byte* mempunyai kapasitas maksimal 15 yang terhitung mulai dari 0, berikut gambaran *Key Byte Access Schedule* :

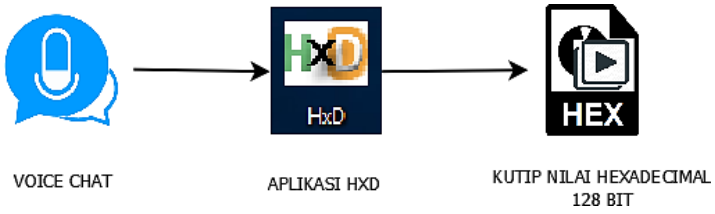
		MESSAGE BYTE							
		0	1	2	3	4	5	6	7
C-I-D ROUND	1	0	1	2	3	4	5	6	7
	2	7	8	9	10	11	12	13	14
	3	14	15	0	1	2	3	4	5
	4	5	6	7	8	9	10	11	12
	5	12	13	14	15	0	1	2	3
	6	3	4	5	6	7	8	9	10
	7	10	11	12	13	14	15	0	1
	8	1	2	3	4	5	6	7	8
	9	8	9	10	11	12	13	14	15
	10	15	0	1	2	3	4	5	6
	11	6	7	8	9	10	11	12	13
	12	13	14	15	0	1	2	3	4
	13	4	5	6	7	8	9	10	11
	14	11	12	13	14	15	0	1	2
	15	2	3	4	5	6	7	8	9
	16	9	10	11	12	13	14	15	0

Gambar 4. *Key Byte Access Schedule*[6]

ADA Journal of Information System Research
Volume 1, No 3, June 2024 Page: 121−128
ISSN 3025-9568 (media online)

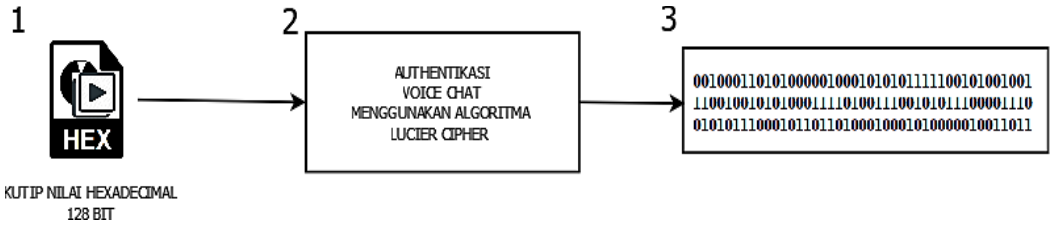
3. HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan mengenai pengumpulan data dan informasi yaitu bagaimana teknik penerapan algoritma lucifer dalam mengamankan *voice chat* pada aplikasi yang dirancang dan dibangun. Proses enkripsi algoritma Lucifer dilakukan dengan memberikan nilai input sebesar 128 bit, yang kemudian dilakukan proses pembentukan kunci dengan panjang bernilai sama dengan nilai input yaitu 128 bit. Objek pada penelitian ini yang akan dienkripsi adalah *voice chat* sementara ini didapatkan dari aplikasi whatsapp. Selanjutnya pada objek voice chat dilakukan ekstrasi nilai dengan bantuan aplikasi HxD. Berikut skema proses ekstrasi nilai hexadecimal dari *file voice chat* menggunaka aplikasi HxD.



Gambar 5. Skema Ekstrasi Nilai Voice Chat

Setelah didapat nilai hexadecimal selanjutnya proses penerapan algoritma luciver, dengan skema proses sebagai berikut.



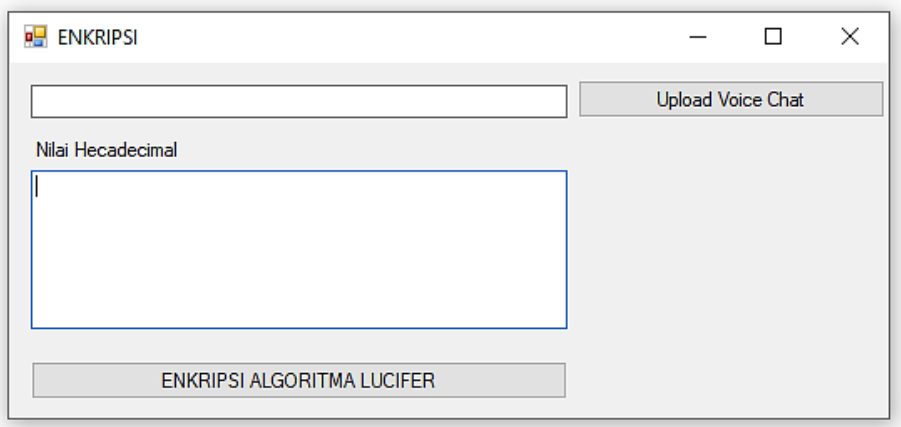
Gambar 6. Skema Enkripsi *Voice Chat* dengan Algoritma Luciver

Berdasarkan gambar diatas maka dilakukan proses enkripsi, berikut penjelasannya:

1. Nilai Hexadecimal hasil proses ekstrasi menggunakan aplikasi HxD
2. Penerapan Algoritma Luciver
   a. Memasukan *plainvoice* sebagai nilai input.
   b. Melakukan proses pembentukan kunci dengan panjang kunci 128 bit.
   c. Proses enkripsi *plainvoice* dengan kunci.
3. Nilai Biner yang selanjutnya di konversi ke bilangan hexadecimal.

3.1 Pengujian

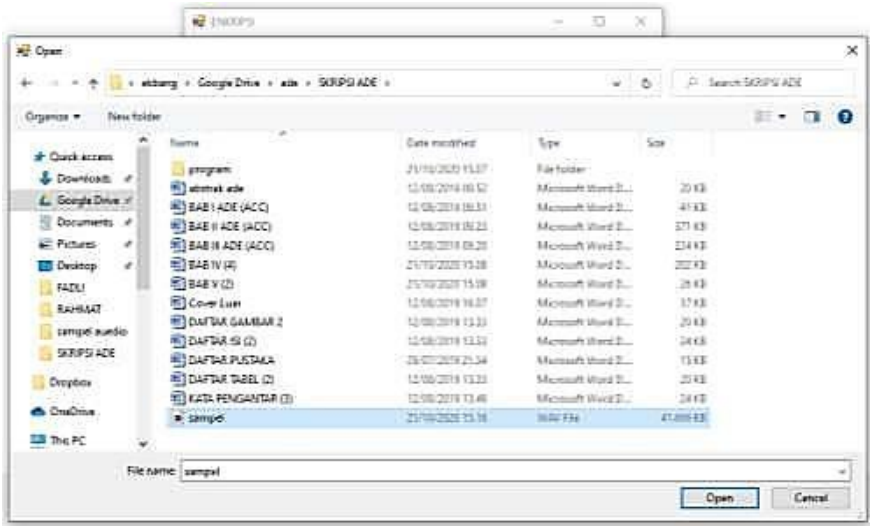
Informasi yang ada pada *file voice chat* tidak dapat diakses oleh orang tidak berkepentingan. Berikut tampilan form pengamanan *file voice chat*,



Gambar 7. Tampilan Form Enkripsi Pengamanan *File voice chat*

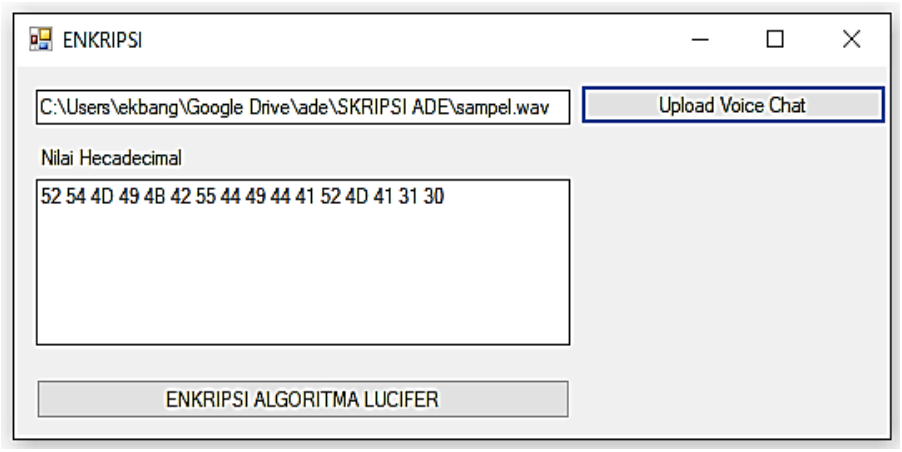
Langkah selanjutnya adalah melakukan proses akses terhadap *file voice* chat yang telah disimpan dalam direktori penyimpanan untuk diproses, dengan mengklik tombol "*Upload Voice Chat*" atau lebih jelas dapat dilihat pada gambar berikut ini.

ADA Journal of Information System Research
Volume 1, No 3, June 2024 Page: 121−128
ISSN 3025-9568 (media online)



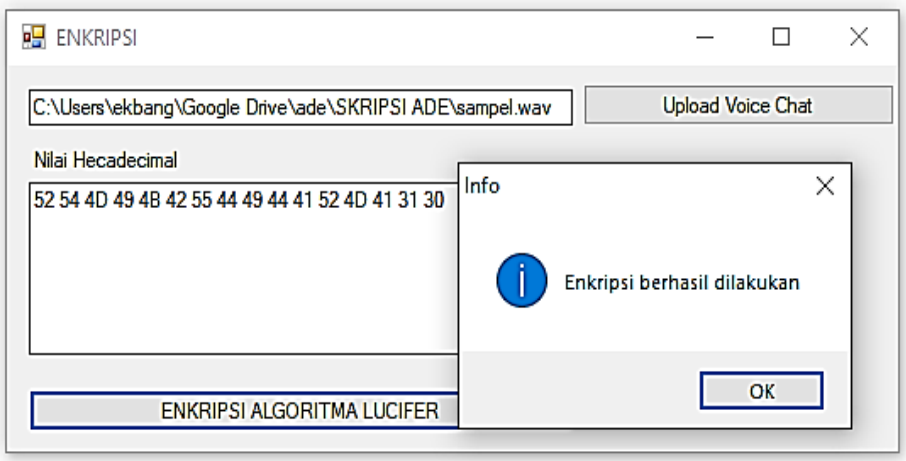
Gambar 8. Proses *Upload Voice Chat*

Setelah proses *Upload Voice Chat* dilakukan maka tampil spesifikasi *file voice chat*, seperti pada tampilan gambar berikut ini.



Gambar 9. Spesifikasi *File* voice *chat*

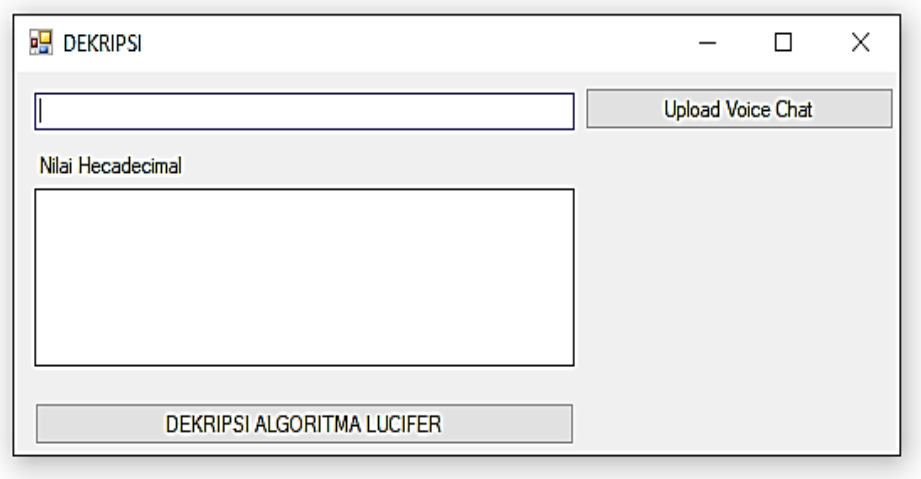
Selanjutnya klik tombol "ENKRIPSI ALGORITMA LUCIFER" untuk melakukan proses pengamanan terhadap *file voice chat* sehingga menghasilkan *voice chat* yang terenkripsi. Pada penelitian ini ditampilkan dalam bentuk nilai hexadecimal, serta akan muncu pertanyaan untuk proses penyimpanan *file voice chat*. Lebih jelas dapat dilihat pada gambar berikut ini



Gambar 10. Tampilan Form Enkripsi Berhasil

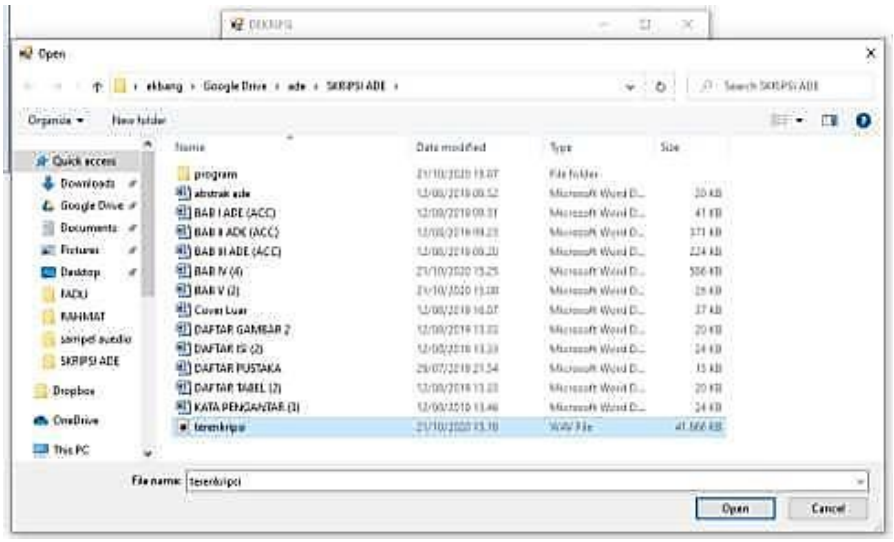
ADA Journal of Information System Research
Volume 1, No 3, June 2024 Page: 121−128
ISSN 3025-9568 (media online)

Selanjutnya akan dilakukan proses dekripsi algoritma *lucifer* pada *file voice chat* yangtelah di enkripsi. Adapun tampilan form dekripsi dapat dilihat pada gambar dibawah ini :



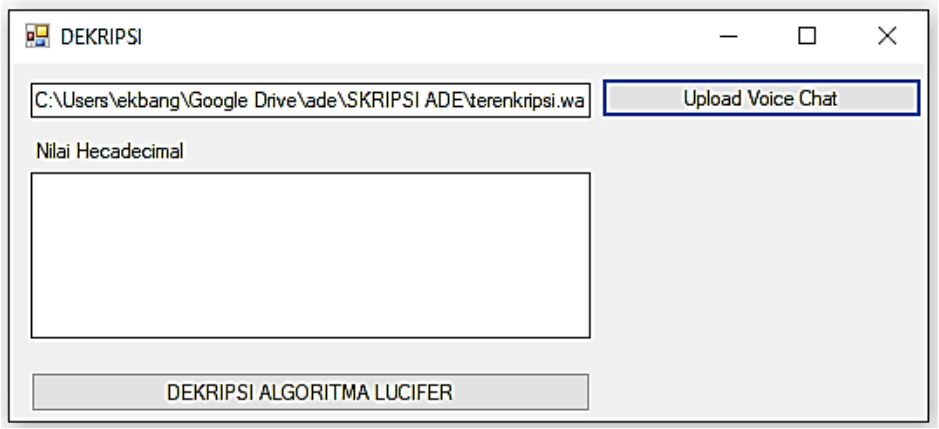
Gambar 11. Tampilan Form Dekripsi Pengamanan *File voice chat*

Langkah selanjutnya adalah melakukan proses akses terhadap *file voice chat* yang telah disimpan dalam direktori penyimpanan untuk diproses, dengan mengklik tombol "*Upload Voice Chat*" atau lebih jelas dapat dilihat pada gambar berikut ini.



Gambar 12. Proses *Upload Voice* Chat yang didekripsi

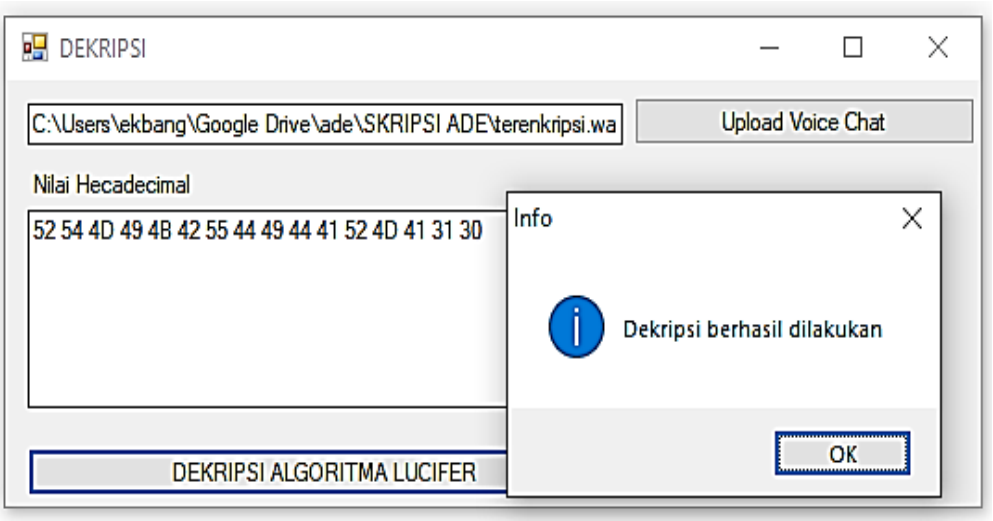
Setelah proses *Upload Voice Chat* dilakukan maka tampil spesifikasi *file voice chat*, seperti pada tampilan gambar berikut ini :



Gambar 13. Spesifikasi *File voice chat* yang didekripsi

ADA Journal of Information System Research
Volume 1, No 3, June 2024 Page: 121−128
ISSN 3025-9568 (media online)

Selanjutnya klik tombol "DEKRIPSI ALGORITMA LUCIFER" untuk melakukan proses pengembalian terhadap *file voice chat* sehingga menghasilkan *voice chat* semula.



Gambar 14. Tampilan Form Dekripsi Berhasil

3.2 Hasil Pengujian

Proses pengujian dalam penelitian ini dilakukan terhadap beberapa *file voice chat* yang berbeda, dengan menggunakan aplikasi yang telah dibangun dengan menggunakan bahasa pemograman Visual Microsoft VB. Net 2008. Berikut tabel hasil pengujian yang telah dilakukan:

Tabel 1. Hasil Pengujian

No.	Spesifikasi File Voice Chat Asli	Ciphervoice	Keterangan
1	NamaFile: sampelUkuran: 485 Kb Durasi: 01:00Jenis: *wav	2B27282E2A21252 82927272C2B2624 27	Berhasil
2.	NamaFile: sampel1 Ukuran: 693 Kb Durasi: 00:57Jenis: *. mp3	0C00D49F80EFA7 85DDCAD01A0FFFF700	Berhasil
3.	NamaFile: sampel2 Ukuran: 820 Kb Durasi: 01:20Jenis: *.mp3	DD0AC7F9C00E1AAF5CD4AE0DA C0ABB2E0	Berhasil

4. KESIMPULAN

Dari hasil penelitian yang dilakukan terhadap pengamanan sound chat pada perancangan aplikasi chatting menggunakan algoritma Lucifer, maka terdapat beberapa kesimpulan berdasarkan uraian yang telah tercantum pada bab-bab sebelumnya. Adapun kesimpulan dari hasil penelitian ini yaitu didalam mengautentifikasi sound chat menggunakan teknik enkripsi akan mengacak nilai hexadesimal sound chat sehingga memanimalisir terjadinya kerusakan pada keaslian sound chat. Algoritma Lucifer dapat melakukan proses enkripsi terhadap voice chat dengan merubah nilai-nilai yang terdapat di dalam sound chat menjadi nilai Hexadecimal sehingga sound chat tidak mudah untuk di pahami. Aplikasi yang dirancang menggunakan Microsoft Visual Basic 2008 berdasarkan tahap-tahap proses enkripsi dari algoritma Lucifer sehingga dapat mempermudah proses pengamanan sound chat.

REFERENCES

[1] M. Mizan, "Studi Perbandingan Algoritma Simetri Lucifer dan Blowfish,"Institut Teknologi Bandung, Bandung, 2010.
[2] Pengertian Menurut Para Ahli, (2016,jun.1). Pengertian Implementasi [online].Available:https://www.pengertianmenurutparaahli.net/pengertian- implementasi/
[3] Wikipedia, (2019, Jul 27). Algoritma [online]. Available: https://id.wikipedia.org/wiki/Algoritme

ADA Journal of Information System Research
Volume 1, No 3, June 2024 Page: 121−128
ISSN 3025-9568 (media online)

[4] S. Kromodimoeljo, Teori dan Aplikasi Kriptografi, : SPK IT Consulting, 2009.
[5] B. Aroya and E. Biham, "Differential Cryptanalysis of Lucifer," *Journal of Cryptology*, vol. 9, no. 1, pp. 21-34, 1996.
[6] B. Buchanan, "Feistel Ciphers," Medium Corporation, 16 September 2018. [Online]. Available: https://medium.com/asecuritysite-when-bob-met-alice/feistel-ciphers-18b400f9e1a9. [Accessed 28 April 2019].
[7] S. Dharwiyanti and R. S. Wahono, "Pengantar Unified Modeling Language (UML)," IlmuKomputer.Com, 2003.
[8] E. Shygan, "Daftar Simbol UML," 24 May 2013. [Online]. Available: https://id.scribd.com/document/143412967/Daftar-Simbol-Uml. [Accessed 08 May 2019].
[9] E. Sutanta, Pengantar Teknologi Informasi, Yogyakarta: Graha Ilmu. 2005
[10] Hendrayudi, Dasar-Dasar Pemograman Microsoft Visual Basic 2008, Bandung:SatuNusa, 2011.