Jurnal EurekaMatika, Vol. 9, No. 1, 2021
Journal homepage: https://ejournal.upi.edu/index.php/JEM
e-ISSN: 2528-4231, p-ISSN: 2776-480X

Kriptografi Audio Menggunakan Transposisi dan *Affine Cipher* yang Dikembangkan dengan Algoritma *Blum Blum Shub*

Muhammad Fakhri Naufal*, Rini Marwati dan Ririn Sispiyati

Departemen Pendidikan Matematika
Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam
Universitas Pendidikan Indonesia
*Surel: fakhrin23@gmail.com

ABSTRAK. Di era informasi, teknologi berkembang dengan pesat dan kemudahan dalam bertukar informasi menjadi sangat mudah, namun dengan perkembangan tersebut timbul suatu masalah yaitu keamanan informasi tersebut terutama untuk suatu informasi rahasia. Diperlukan sebuah mekanisme untuk mengamankan informasi dalam bentuk file audio, yaitu dengan kriptografi. Teknik kriptografi audio seperti transposisi membuat data audio teracak sehingga suara yang dihasilkan file audio tersebut tidak dapat dipahami. Namun, untuk meningkatkan keamanan nilai data audio pada file audio diperlukan teknik enkripsi substitusi, salah satunya yaitu *Affine Cipher*. Dengan melakukan pengembangan pada *Affine Cipher* menggunakan pembangkit bilangan acak semu *Blum Blum Shub,* dapat memberikan peningkatan keamanan yang cukup signifikan dibandingkan dengan teknik kriptografi klasik ini. Hasil yang diperoleh dengan mengenkripsi file audio WAV menggunakan *Python* dapat mengamankan file audio sehingga menghasilkan suara acak dan file audio yang terenkripsi dapat didekripsi untuk mendapatkan informasi asli.

Kata Kunci: Kriptografi, Kriptografi Audio, File Audio, Transposisi, *Affine Cipher, Blum Blum Shub*

e-ISSN: 2528-4231, p-ISSN: 2776-480X

Audio Cryptography using Transposition and Developed Affine Cipher with Blum Blum Shub Algorithm

ABSTRACT. *In this information era, technology is developing rapidly and the ease of exchanging information becomes very easy, but with these developments arises a problem that is the security of such information especially for a secret information. It is required a mechanism to secure the information in audio file form, that is cryptography. Audio cryptographic techniques such as transposition make the audio data scrambled so that the sound generated by the audio file is incomprehensible. However, to increase the security of the value of audio data in audio files, it is necessary to encrypt substitution techniques, one of which is Affine Cipher. By developing Affine Cipher using a pseudo random number generator Blum Blum Shub, it can provide a significant improvement security on this classic cryptographic technique. The results obtained by encrypting WAV audio files using Python can secure the audio file so that it generates scrambled sounds and the encrypted audio file can be decrypted to get the original information.*

Keywords: *Cryptography, Audio Cryptography, Audio Files, Transposition, Affine Cipher, Blum Blum Shub*

1. PENDAHULUAN

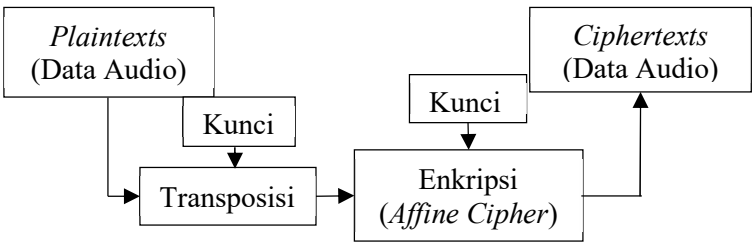
Di era informasi ini, komunikasi memainkan peran yang sangat penting dalam membantu pengembangan teknologi baru. Dalam buku yang ditulis oleh Mitra (Mitra, 2018) disebutkan bahwa, dalam pengembangan teknologi yang pesat ini, bentuk dari perangkat komputer yaitu *processors* dan penyimpanan (*hard disk)* semakin kecil, namun dengan munculnya teknologi baru tersebut tentu kemampuan menghitung dan kapasitas penyimpanan meningkat. Hal ini memungkinkan informasi yang dikomunikasikan jauh lebih mudah, lebih cepat, dan lebih praktis baik dalam jaringan kabel maupun nirkabel. Seiring dengan perkembangan teknologi tersebut, timbul suatu masalah yaitu keamanan dan kerahasiaan dari pertukaran informasi.

Keamanan adalah hal yang penting untuk dipertimbangkan, terlebih jika informasi tersebut sangat rahasia. Oleh karena itu, sebuah mekanisme diperlukan untuk mengamankan informasi yang dikirim (Mitra, 2018). Proses dari mengubah informasi asli menjadi bentuk tersamar sebagai enkripsi. Proses mengkonversi informasi yang tersamar menjadi informasi asli dikenal sebagai dekripsi. Ilmu dari enkripsi dan dekripsi dikenal sebagai kriptografi. Kriptografi ini dapat menjamin bahwa informasi yang dikirimkan hanya diketahui oleh pengirim dan penerima saja. Kriptografi itu sendiri dapat mengenkripsi dan mendekripsi file berbentuk pesan teks, gambar, video dan juga audio.

Mansi dan Chawla mengemukakan bahwa "Selama bertahun-tahun beberapa teknik enkripsi telah diterapkan, tetapi sebagian besar teknik hanya mengenkripsi data teks, sangat sedikit teknik yang diusulkan untuk data multimedia seperti data audio" (Mansi & Chawla, 2015). Data audio merupakan suatu kumpulan data yang memiliki nilai tertentu. Mengenkripsi data audio berarti mengamankan nilai-nilai di dalamnya sehingga membuat data audio terenkripsi menghasilkan suara yang tidak dapat dimengerti oleh pihak ketiga. Hal ini memungkinkan pengirim mengirimkan data audio pada jaringan apapun dengan aman tanpa seseorang mengetahui data audio asli kecuali penerima yang telah mengetahui kunci yang telah disepakati oleh keduanya.

Salah satu cara untuk mengamankan data audio yaitu dengan melakukan transposisi. Transposisi bertujuan untuk mengacak data audio pada file audio sehingga suara yang dihasilkan tersamarkan. Salah satu cara transposisi yaitu menggunakan pembangkit bilangan acak semu, sehingga posisi dari data audio benar-benar teracak. Algoritma *Blum Blum Shub* merupakan pembangkit bilangan acak semu yang cukup mudah untuk dibangkitkan melalui persamaannya namun bilangan acak yang dihasilkan

tidak mudah untuk diprediksi (Stinson & Rosen, 2006). Tetapi dalam penelitian yang telah dilakukan (Sinha et al., 2015) dijelaskan bahwa, audio yang dienkripsi hanya menggunakan transposisi algoritma *Blum Blum Shub* perlu diberikan peningkatan ketahanan dari file audio, salah satunya dengan cara dilanjutkan dengan menggunakan enkripsi substitusi pada file audio tersebut. Teknik enkripsi substitusi dari kriptografi klasik *Affine Cipher* dapat diterapkan dalam kriptografi audio untuk meningkatkan keamanan file audio di mana data audio tersebut dienkripsi dengan suatu persamaan sehingga akan memberikan *noise* atau gangguan pada file audio. Kriptografi klasik memiliki tingkat keamanan yang kurang baik jika dibandingkan dengan kriptografi modern karena algoritma kriptografi modern terbilang relatif kompleks dibandingkan kriptografi klasik. Oleh karena itu diperlukan suatu pengembangan pada algoritma *Affine Cipher* ini, salah satunya yaitu menggunakan barisan bilangan acak. Barisan bilangan acak memungkinkan suatu nilai yang sama akan menghasilkan nilai yang berbeda setelah melalui proses enkripsi. Algoritma pembangkit barisan bilangan acak yang digunakan adalah Algoritma *Blum Blum Shub.* "Proses penyandian yang diusulkan tersebut dapat menyelesaikan cacat penyandian substitusi dengan memastikan bahwa nilai yang dihasilkan berbeda, dan tidak ada pola yang dapat dikenali dari karakter yang identik saat dienkripsi" (Arroyo & Delima, 2020)\. Proses enkripsi file audio dapat dilihat pada gambar berikut.



Gambar 1. Proses Enkripsi File Audio

Gambar 1 menjelaskan tentang proses enkripsi pada file audio menggunakan transposisi dan enkripsi substitusi. Berdasarkan Gambar 1, data audio asli yang telah diperoleh dari file audio asli dienkripsi menggunakan transposisi setelah itu dilanjutkan menggunakan enkripsi *Affine Cipher* dengan masing-masing kuncinya sehingga diperoleh data audio terenkripsi. Kemudian dibentuk file audio dari data audio terenkripsi sehingga menghasilkan file audio yang suaranya tersamarkan, file audio tersebut dapat dikirim melalui jaringan tidak aman (publik) karena informasi yang terdapat pada file audio tidak dapat diperoleh tanpa menggunakan kunci.

2. METODOLOGI

Terdapat tahapan-tahapan yang dilakukan dalam penelitian ini, diantaranya:

1. Studi literatur mengenai proses enkripsi pada file audio, *Affine Cipher,* dan algoritma *blum blum shub*.

   Selama bertahun-tahun kriptografi telah diterapkan namun sebagian besar hanya diterapkan pada file teks saja, Sebagian kecilnya diterapkan pada file multimedia lainnya seperti file audio. Dengan mengamankan file audio menggunakan kriptografi, membuat suara yang dihasilkan setelah proses enkripsi menjadi tidak dapat dikenali oleh penyadap. File audio asli dapat diperoleh dengan melakukan proses dekripsi pada file audio yang telah dienkripsi menggunakan kunci yang telah disepakati. Salah satu cara mengamankan file audio yaitu dengan melakukan transposisi pada data audio menggunakan pembangkit bilangan acak semu *Blum Blum Shub*, tetapi untuk meningkatkan keamanan diperlukan teknik enkripsi substitusi pada data audio tersebut. Salah satu algoritma kriptografi yaitu *Affine Cipher* dapat diterapkan pada kriptografi audio ini untuk melakukan proses enkripsi substitusi. Namun *Affine Cipher* merupakan kriptografi klasik yang keamanannya kurang baik, maka diperlukan pengembangan pada *Affine Cipher* untuk meningkatkan keamanan serta kerahasiaan file audio. Pembangkit barisan bilangan acak dapat digunakan untuk mengembangkan *Affine Cipher,* salah satunya yaitu algoritma pembangkit bilangan acak semu *Blum Blum Shub*
2. Merancang proses transposisi data audio menggunakan algoritma *blum blum shub.*

   Transposisi yang dilakukan yaitu dengan menukar nilai-nilai data audio sehingga urutannya teracak, maka dari itu dibutuhkan barisan bilangan acak sebanyak *n* dengan *n* yaitu banyaknya data audio. Misalkan bilangan acak yang dihasilkan oleh algoritma *Blum Blum Shub* yaitu $X_1, X_2, X_3, \ldots\ldots, X_n$ yang memiliki nilai berbeda satu sama lain dalam modulus *M*. Penulis menggunakan metode ranking untuk mendapatkan nilai yang memenuhi $0 < X_i \leq n$, $i = 1,2,3,\ldots,n$ dari bilangan acak yang diperoleh. Misalkan R_i merupakan urutan nilai X_i dari yang terkecil hingga yang terbesar, maka diperoleh bilangan acak $R_1, R_2, R_3, \ldots\ldots, R_n$ yang memenuhi $0 < R_i \leq n$, $i = 1,2,3,\ldots,n$ dan R_i merupakan nilai yang berbeda satu sama lainnya. Setelah itu, barulah proses transposisi dapat dilakukan dengan mengurutkan data audio sesuai bilangan acak tersebut.
3. Model dasar yang digunakan adalah kriptografi klasik *Affine Cipher.*

Teknik enkripsi diperlukan untuk mengamankan nilai-nilai data audio yang sebenarnya namun selain itu enkripsi pada file audio juga akan memberikan *noise* atau gangguan suara sehingga akan menyamarkan suara aslinya agar terdengar tidak jelas. Bermacam-macam kriptografi dapat diterapkan pada file audio, disini penulis menggunakan kriptografi klasik *Affine Cipher* sebagai model dasar.

Rumus yang digunakan adalah:

$$A_{j,d}(x) = (jx + d) \bmod m$$

Dimana setiap data audio x dienkripsi dengan menggunakan kunci j dan d serta menggunakan modulus m. Jika dalam alfabet nilai m yaitu 26, maka dalam audio memiliki dua nilai tergantung *bit* yang digunakan pada file audio.

4. Modifikasi terhadap *Affine Cipher* terletak pada variabel d yaitu kunci pergeseran yang diubah menjadi bilangan acak yang dibangkitkan dari algoritma *blum blum shub.*

   Algoritma *Blum Blum Shub* dapat meningkatkan keamanan algoritma *Affine Cipher* dengan memanfaatkan barisan bilangan acak yang tidak dapat diprediksinya. Dengan rumus $X_{n+1} = X_n^2 \bmod M$, barisan bilangan acak yang dapat dibentuk sesuai banyaknya data audio pada file audio yang akan dienkripsi. Jika pada awalnya rumus *Affine Cipher* yaitu $A_{j,d}(x) = (jx + d) \bmod m$ dengan nilai j dan d konstan yang telah ditentukan sebelum proses enkripsi dilakukan, maka pada pengembangan *Affine Cipher* ini dilakukan pergantian nilai d atau nilai pergeseran dengan barisan bilangan acak yang telah dibentuk dengan algoritma *Blum Blum Shub*.

   Rumus yang digunakan adalah:

$$A_{j,d}(x_i) = (jx_i + X_i)$$

5. File audio yang telah dienkripsi dilakukan validasi atau kecocokan suara dan data audio dengan file audio yang telah didekripsi.

3. HASIL DAN PEMBAHASAN

Proses enkripsi file audio pada penelitian ini diuraikan dalam langkah-langkah sebagai berikut:

1. File audio dibaca menjadi data audio dalam *bit*. Pada file audio 16 *bit*, data audio x_i berada dalam rentang nilai [-32768, 32767].
2. Bilangan acak dibentuk menggunakan algoritma *blum blum shub* melalui persamaan:

   $$X_{i+1} = X_i^2 \bmod M$$

   dengan X_0 adalah nilai awal sebagai pembangkit yang relatif prima dengan $M = pq$, yaitu perkalian dua bilangan prima besar yang memenuhi $p \equiv q \equiv$

3 mod 4. Syarat ini harus dipenuhi untuk menjamin fungsi *f* injektif (Stinson & Rosen, 2006) yang didefinisikan sebagai berikut:

$$f: \quad A \rightarrow B$$
$$[X] \rightarrow [X^2 \ mod \ M]$$

Untuk $X \in \mathbb{Z}_M$

3. Kemudian, dibentuk bilangan acak baru R_i yang memiliki nilai peringkat X_i dari terkecil ke terbesar sehingga bilangan acak baru adalah bilangan bulat memenuhi $0 < R_i \leq n$ dengan n banyaknya data audio. Setelah itu bentuk data audio teracak $x_i^{'} = x_{Ri}$.
4. Data audio teracak $x_i^{'}$ diberikan pergeseran sebesar +32768 sehingga $x_i^{'} \in \mathbb{Z}_m$ dengan $m = 65536$ yaitu nilai modulus sesuai *bit* yang digunakan. Setelah itu, data audio dienkripsi menggunakan persamaan *Affine Cipher* yang telah dikembangkan menggunakan algoritma *blum blum shub* sebagai berikut:

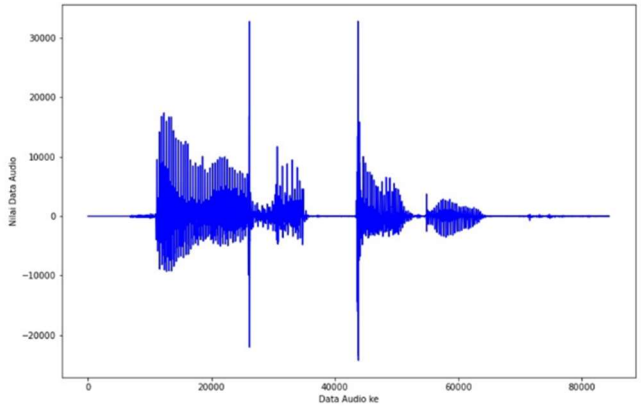
$$c_i = j\, x_i^{'} + X_i \pmod{m}$$ (Arroyo & Delima, 2020)

dengan *c* adalah data audio terenkripsi dan *j* adalah kunci *Affine Cipher* yang relatif prima dengan *m*. Setelah data audio dienkripsi, diberikan kembali pergeseran sebanyak -32768 untuk dibentuk menjadi file audio sehingga $c_i \in [-32768, 32767]$.

File audio yang digunakan berupa pesan singkat yang direkam menggunakan aplikasi *audacity* dengan dekripsi sebagai berikut:

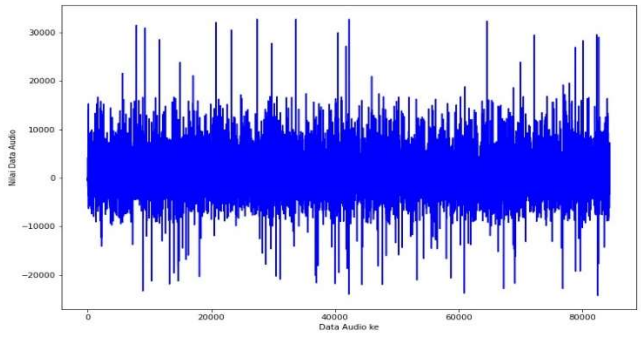
Tabel 1. Dekripsi File Audio

Durasi	± 0:02
Banyaknya Data Audio	84480
Bit	16 *bit*
Format	*Wav*
Sample Rate	44100



Gambar 2. Plot Data Audio

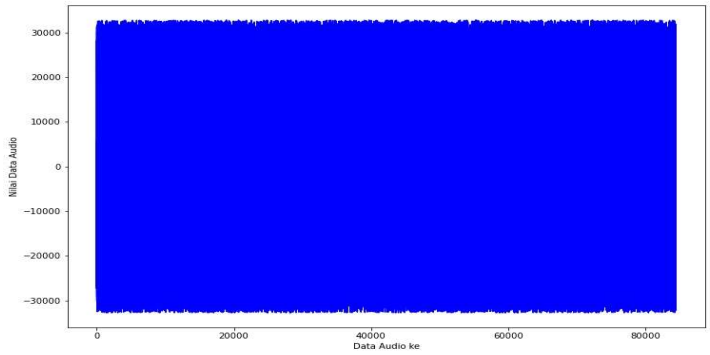
Gambar 2 merupakan plot data audio asli yang terdapat pada file audio asli seperti pada Tabel 1. Dilakukan enkripsi transposisi pada data audio menggunakan algoritma *blum blum shub* dengan menggunakan kunci: $X_0 = 62225$, $p = 1000003$, $q = 400003$, sehingga diperoleh nilai $M = pq = 400004200009$. Setelah proses transposisi dilakukan, diperoleh plot data audio yang ditampilkan pada Gambar 3.



Gambar 3. Plot Data Audio Teracak

Gambar 3 merupakan plot data audio yang telah ditransposisi menggunakan algoritma *blum blum shub.* Berdasarkan Gambar 3, terlihat bahwa data audio sudah teracak dan menghasilkan suara yang tersamarkan, namun untuk meningkatkan keaamanan file audio yaitu nilai data audionya, dilakukan enkripsi substitusi menggunakan *Affine Cipher* yang dikembangkan dengan algoritma *blum blum shub*. Dengan menggunakan bilangan acak yang telah dibangkitkan sebelumnya

menggunakan algoritma *blum blum shub*, digunakan kunci *j = 32911* sehingga diperoleh plot data audio pada Gambar 4.



Gambar 4. Plot Data Audio Terenkripsi

Gambar 4 merupakan plot data audio yang telah ditransposisi dan dienkripsi menggunakan modifikasi *Affine Cipher.* Kemudian, data audio terenkripsi pada Gambar 4 tersebut dibentuk kembali menjadi file audio dan menghasilkan suara yang tersamarkan. File audio dapat dikirim melalui jaringan tidak aman karena peretas tidak dapat memahami informasi pada file audio.

Peningkatan keaamanan dari *Affine Cipher* menjadi *Affine Cipher* yang dikembangkan dengan algoritma *blum blum shub* dapat dilihat dari berbagai aspek, diantaranya:

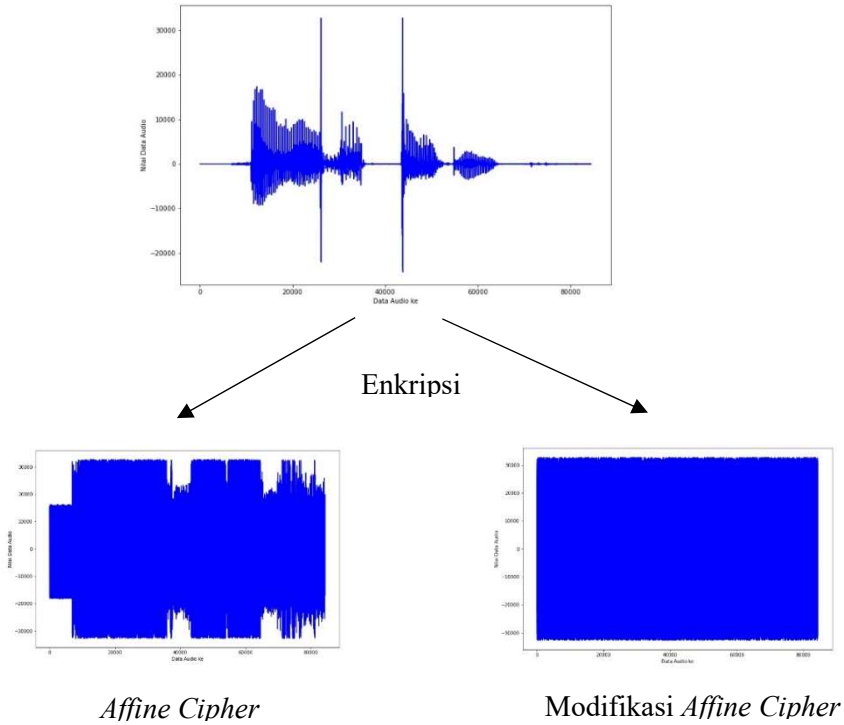
1. Banyaknya kunci yang mungkin digunakan

Affine cipher pada file audio menggunakan pasangan kunci *j,d* dengan banyaknya kunci *j* yang relatif prima dengan *m* sebanyak 32.767 dan banyaknya kunci *d* yang mungkin sebanyak 65.535 sehingga banyaknya pasangan kunci yang mungkin adalah 32.767 x 65.535 = 2.147.385.345. Namun, pada modifikasi *Affine Cipher*, kunci konstan *d* diubah menjadi bilangan acak sehingga banyaknya pasangan kunci yang mungkin adalah 32.767 x 65.535 x *n* = 2.147.385.345 dengan *n* banyaknya data audio. Banyaknya pasangan yang mungkin digunakan meningkat sebanyak *n* kali.

2. Plot data audio

Peningkatan keamanan tidak selalu dapat dilihat dari bentuk plot data audio, namun cara ini dapat melihat apakah data audio benar-benar terenkripsi. Dilakukan enkripsi *Affine Cipher* dan modifikasi *Affine Cipher* pada data audio asli sehingga diperoleh perbandingan plot data yang terdapat pada Gambar 5.

Gambar 5 merupakan perbandingan dari data audio yang dienkripsi menggunakan *Affine Cipher* dengan data audio yang dienkripsi menggunakan modifikasi *Affine Cipher.* Pada Gambar 5 tersebut, dapat terlihat pola plot data audio terenkripsi menggunakan *Affine Cipher* masih memiliki pola yang identik dari plot data audio asli. Namun, hasil yang diperoleh bergantung dari pengambilan nilai kunci.



Gambar 5. Perbandingan Enkripsi Menggunakan Affine Cipher dan Modifikasi Affine Cipher

3. Koefisien korelasi *pearson*

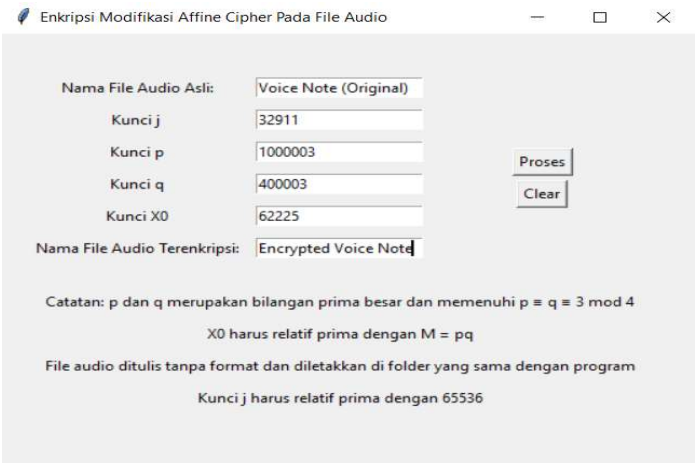
Koefisien korelasi *pearson* merupakan salah satu parameter yang penting digunakan untuk mengevaluasi kesamaan antara data audio asli dan data audio terenkripsi (George et al., 2015), (Moreno-Alvarado et al., 2020). Evaluasi statistik ini untuk menguji kualitas dari algoritma enkripsi. Perhitungan korelasi koefisien menentukan tingkat dari korelasi antara dua file dan koefisien korelasi selalu dalam rentang [-1,1] (Kordov, 2019), (Akgül , 2015).

Tabel 2. *Nilai Korelasi Masing-Masing Teknik Enkripsi*

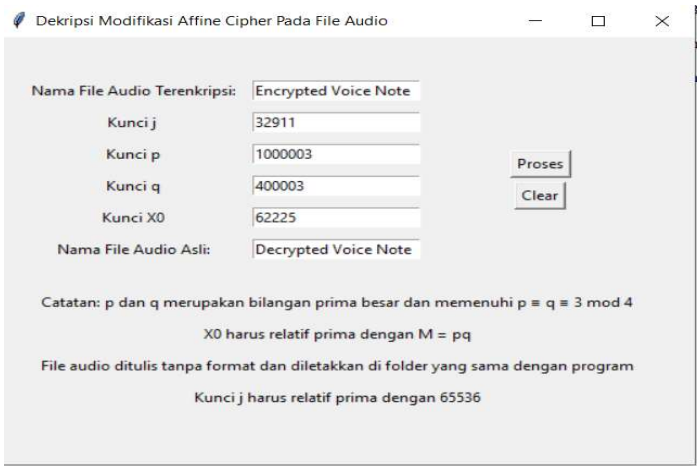
Teknik Enkripsi	Nilai Korelasi *Pearson r*
Affine Cipher	0,00353209
Modifikasi *Affine Cipher*	-0,00325918

Tabel 2 merupakan nilai korelasi dari teknik enkripsi menggunakan *Affine Cipher* dan modifikasi *Affine Cipher.* Berdasarkan Tabel 2, nilai korelasi *pearson* untuk hasil enkripsi data audio menggunakan modifikasi *Affine Cipher* lebih mendekati 0 dari pada *Affine Cipher*. Oleh karena itu, data audio terenkripsi oleh modifikasi *Affine Cipher* memiliki hubungan yang lebih kecil dengan data audio asli dibandingkan *Affine Cipher*. Dengan nilai korelasi yang mendekati 0, akan sulit untuk menentukan data audio asli menggunakan data audio terenkripsi yang dikirim melalui jaringan tidak aman. Namun, hasil ini dapat berubah sesuai kunci yang digunakan di masing-masing teknik enkripsi.

Dengan menggunakan program aplikasi yang telah dibuat menggunakan *Python*, dilakukan validasi untuk melihat kecocokan data audio dan suara yang dihasilkan dari file audio.

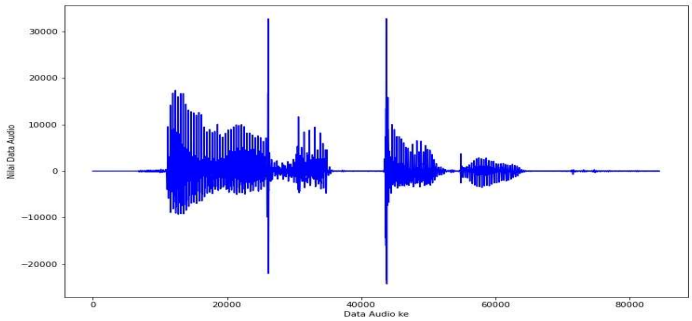


Gambar 6. Tampilan Program Aplikasi Enkripsi



Gambar 7. Tampilan Program Aplikasi Dekripsi

Diperoleh plot data audio sebagai berikut:



Gambar 8. Plot Data Audio Terdekripsi

Gambar 8 merupakan plot data audio terdekripsi yang diperoleh dari file audio terenkripsi. Berdasarkan Gambar 8 tersebut, dibentuk file audio untuk diperoleh informasi asli dari pengirim. File audio yang memiliki data audio terdekripsi ini menghasilkan suara atau informasi yang sama dengan suara atau informasi asli sehingga tujuan dari mengamankan informasi pada file audio terpenuhi. Transposisi *Blum Blum Shub* dan modifikasi *Affine Cipher* telah terbukti dapat mengamankan file audio dengan enkripsi serta file audio dapat didekripsi dan kembali pada informasi asli baik dari suara yang dihasilkan maupun plot data audio.

4. KESIMPULAN DAN SARAN

Berdasarkan dari pembahasan yang dipaparkan sebelumnya, dapat ditarik kesimpulan sebagai berikut:

1. Algoritma transposisi dengan menggunakan *Blum Blum Shub* pada file audio yaitu dengan membangkitkan bilangan acak terlebih dahulu menggunakan algoritma *Blum Blum Shub.* Setelah itu, dibentuk bilangan acak baru yang berisi nilai peringkat bilangan acak *Blum Blum Shub* dari terkecil sampai terbesar. Kemudian, data audio diacak sesuai dengan bilangan acak baru.
2. Algoritma modifikasi *Affine Cipher* pada file audio memiliki perbedaan pada nilai pergesaran yang dimodifikasi dari *Affine Cipher*. Nilai pergeseran *d* yang bernilai konstan pada algoritma *Affine Cipher,* diubah menjadi bilangan acak yang dihasilkan dari algoritma *Blum Blum Shub* sehingga setiap data audio memiliki nilai pergeseran yang berbeda.
3. Modifikasi *Affine Cipher* menggunakan algoritma *Blum Blum Shub* memiliki keamanan yang jauh lebih tinggi pada file audio dengan banyaknya pasangan kunci yang mungkin pada file audio 8 *bit* sebanyak $127 \times 255 \times n$ dengan n banyaknya data audio dan 32.767 x 65.535 x n pada file audio 16 *bit* lebih besar sebanyak n kali dari banyaknya kemungkinan pasangan kunci *Affine Cipher.* Selain dari itu, diperoleh nilai korelasi *Pearson* r modifikasi *Affine Cipher* lebih kecil dibandingkan *Affine Cipher* asli yang berarti modifikasi *Affine Cipher* memiliki hubungan keterkaitan lebih rendah antara data audio setelah dienkripsi dengan data audio asli.
4. Program aplikasi transposisi dan modifikasi *Affine Cipher* menggunakan *Python* dikonstruksi menggunakan modul *Tkinter* untuk tampilan program aplikasi, *Scipy* untuk mengubah file audio menjadi data audio ataupun sebaliknya, dan *Pandas* untuk mengolah data audio (enkripsi dan dekripsi).

Berdasarkan pembahasan penelitian ini, penulis menyarankan algoritma modifikasi *Affine Cipher* untuk diaplikasikan pada file audio yang telah disisipkan suatu pesan (steganografi audio) karena terbilang cukup baik untuk mengamankan suatu pesan yang telah disisipkan pada file audio.

5. DAFTAR PUSTAKA

Akgül, A., Kaçar, S., & Pehlivan, İ. (2015). An audio data encryption with single and double dimension discrete-time chaotic systems. *Tojsat*, *5*(3), 14-23.

Arroyo, J.C.T. & Delima, A.J.P. (2020). An Improved Affine Cipher using Blum Blum Shub Algorithm. *International Journal of Advanced Trends in Computer Science and Engineering, 9,* 3295 – 3298.

George, S. N., Augustine, N., & Pattathil, D. P. (2015). Audio security through compressive sampling and cellular automata. *Multimedia Tools and Applications*, *74*(23), 10393-10417.

Kordov, K. (2019). A novel audio encryption algorithm with permutation-substitution architecture. *Electronics*, *8*(5), 530.

Mansi, M., & Chawla, M. R. (2015). A review on audio cryptography. *International Journal of Modern Communication Technologies and Research*, *3*(7), 265721.

Mitra, P. (2018). *Introductory Chapter: Recent Advances in Cryptography and Network Security*. Guwahati: Department of Computer Science and Engineering.

Moreno-Alvarado, R., Rivera-Jaramillo, E., Nakano, M., & Perez-Meana, H. (2020). Simultaneous audio encryption and compression using compressive sensing techniques. *Electronics*, *9*(5), 863.

Sinha, N., Bhowmick, A., & Kishore, B. (2015). Encrypted information hiding using audio steganography and audio cryptography. *International Journal of Computer Applications*, *112*(5), 0975 – 8887.

Stinson, D. R. & Rosen, K.H. (Penyunting). (2006). *Criptography: Theory and Practice.* 3rd Ed. Chapman & Hall/CRC: Ontario.