JMT (Jurnal Matematika dan Terapan) **5** (1) (2023), 1-11
https://doi.org/10.21009/jmt.5.1.1
e-ISSN: 2615-6792

Implementasi Kriptografi *Secret Sharing Scheme* dan Steganografi Audio *Least Significant Bit* (LSB)

Alvira Firjan Humaira[1,a)], Rini Marwati[1,b)], dan Kartika Yulianti[1,c)]

[1]*Departemen Pendidikan Matematika, Universitas Pendidikan Indonesia, Jl. Dr. Setia Budhi No. 229, Bandung 40154, Indonesia*

[a)]alvirafirjanhumaira@gmail.com, [b)]rinimarwati@upi.edu, [c)]kartika.yulianti@upi.edu

Abstract

As long as communication technology continues to develop, information security becomes very important because crime in cyberspace is increasingly common. To improve information security, in this study a combination of secret sharing scheme (t, w) cryptography with the least significant bit (LSB) audio steganography was constructed. Secret sharing scheme cryptography is a modern cryptography that is able to prevent centralized information situations from occurring because it does not require a key for encryption. In addition, this method also makes it difficult for hackers to reconstruct messages because it is difficult to collect minimum shares. The implementation of this merger resulted in a prototype program using the Python 3.10 programming language with schema (3,4). The cover steganography media used is audio, and messages that can be processed by the program are a six-digit PIN number with a non-zero first digit. The results obtained from the encryption and embedding program are 4 pieces of audio-share which sound the same as the original audio, so that the existence of information in the audio is difficult to know. The result of the decryption and extracting program is a PIN that can be reconstructed.

Keywords: cryptography, secret sharing scheme, audio steganography, least significant bit.

Abstrak

Selama komunikasi teknologi terus berkembang, keamanan informasi menjadi sangat penting karena kejahatan di dunia maya semakin marak terjadi. Untuk meningkatkan keamanan informasi, pada penelitian ini dikonstruksi penggabungan kriptografi *secret sharing scheme* (*t, w*) dengan steganografi audio *least significant bit* (LSB). Kriptografi *secret sharing scheme* merupakan kriptografi *modern* yang mampu mencegah terjadinya situasi informasi terpusat karena tidak memerlukan kunci untuk enkripsi. Selain itu, metode ini juga menyulitkan peretas dalam merekonstruksi pesan karena sulitnya mengumpulkan minimal *share*. Implementasi dari penggabungan tersebut, dihasilkan suatu *prototype* program menggunakan bahasa pemrograman Python 3.10 dengan skema (3,4). Media *cover* steganografi yang digunakan adalah audio, dan pesan yang dapat diolah program adalah PIN angka enam digit dengan digit pertama tidak nol. Hasil yang diperoleh dari program enkripsi dan *embedding* adalah 4 buah *audio-share* yang terdengar sama dengan audio yang asli, sehingga keberadaan informasi di dalam audio sulit diketahui. Hasil pada program dekripsi dan *extracting* adalah PIN yang dapat dikonstruksi kembali.

Kata-kata kunci: Kriptografi, *secret sharing scheme*, steganografi audio, *least significant bit.*

PENDAHULUAN

Selama komunikasi teknologi terus berkembang, keamanan informasi yang dikirimkan melalui internet tidaklah aman, karena kejahatan di dunia maya semakin marak terjadi. Para ahli komunikasi data memusatkan perhatiannya pada ancaman peretas yang dapat mengakses informasi rahasia [1].

Diterima: 10 Februari 2023, Direvisi: 28 Februari 2023, Disetujui: 28 Februari 2023

Ketika informasi rahasia akan dikirimkan kepada pihak kedua melalui internet diperlukan suatu pengamanan. Teknik yang banyak digunakan untuk mengamankan informasi adalah dengan teknik kriptografi dan steganografi [2].

Kriptografi merupakan salah satu cara menjaga kerahasiaan data dengan cara mengubah informasi ke dalam bentuk yang tidak dapat dipahami lagi maknanya [3]. Tetapi terkadang memanipulasi informasi, menyebabkan peretas bisa saja curiga akan adanya informasi rahasia. Tidak seperti kriptografi, steganografi berfokus pada menyembunyikan keberadaan informasi dibandingkan konten informasi. Steganografi merupakan seni untuk menyembunyikan informasi ke dalam informasi lain (media *cover*). Ketika teknik kriptografi dan steganografi digabungkan, akan memberikan efek tingkat keamanan yang tinggi [4].

Pada teknik kriptografi yang membutuhkan kunci untuk enkripsi dan dekripsi, membuat sebuah informasi rahasia menjadi terpusat [2]. Hal ini dikarenakan jika terjadi kerusakan pada sistem penyimpanan kunci, informasi tidak dapat didapatkan kembali, atau pengungkapan data akan menjadi mudah ketika peretas berhasil mendapatkan pemilik atau penyimpan kunci [4]. Masalah informasi terpusat tersebut dapat diatasi oleh teknik kriptografi *secret sharing scheme*, di mana informasi akan dibagi menjadi beberapa bagian (*share*) dan dibagikan, sehingga peretas akan kesulitan mengumpulkan minimal *share* untuk merekonstruksi informasi semula.

Informasi asli memang akan tersamar jika menggunakan kriptografi *secret sharing scheme*, tetapi peretas masih bisa mencari keberadaan informasi yang sudah dipecah tersebut. Dengan demikian, diperlukan teknik steganografi untuk menyembunyikan *share*. Ada berbagai media penampung (cover) yang dipakai untuk steganografi. Pada penelitian ini dipakai *cover* sebuah audio dikarenakan pertukaran audio di internet merupakan hal yang biasa [2]. Metode yang dipakai adalah *least significant bit* (LSB) dikarenakan memiliki keunggulan di mana tidak ada perbedaan yang signifikan antara audio asli dengan audio terembed meski dilakukan dengan cara yang sederhana dan mudah dilakukan [2][5].

METODE

Teknik kriptografi yang dipakai adalah *secret sharing scheme* (*t*, *w*), di mana metode ini membagi sebuah pesan rahasia (*secret*) menjadi sejumlah bagian (*share*), kemudian sebanyak *w* partisipan akan menerima *share*. *Secret* akan bisa dikonstruksi kembali jika terdapat minimal *t* partisipan yang mengumpulkan share dan dihitung menggunakan interpolasi Lagrange. Jika kurang dari *t*, maka *secret* tidak akan bisa dikonstruksi kembali [3].

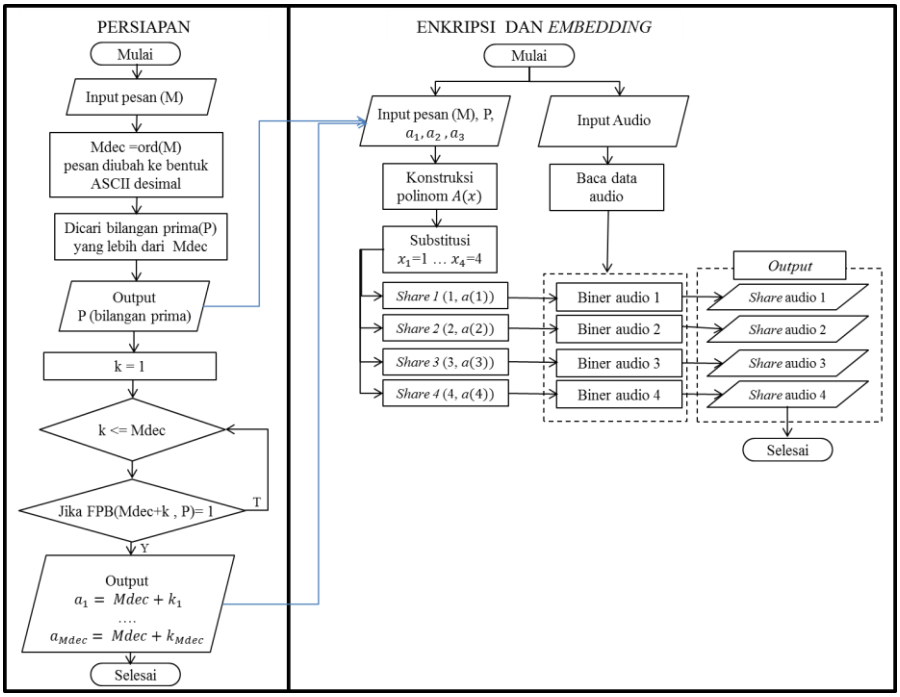
Salah satu teknik steganografi yang terjaga keamanannya adalah *least significant bit*, hal itu karena teknik ini menyembunyikan biner pesan ke digit LSB media *cover* sehingga peretas akan kesulitan mencari di mana keberadaan pesan karena media *cover* terembed informasi tidak akan jauh berbeda dengan media *cover* asli dan tidak bisa langsung dibedakan jika hanya dengan indra manusia. Pada penelitian ini, *cover* yang digunakan adalah audio. Karena steganografi audio merupakan teknik yang baik untuk mengamankan informasi rahasia yang akan dikirimkan melalui internet [6].

Pada penelitian ini, dimisalkan kasus pengamanan PIN ATM menggunakan pembagian PIN, sehingga PIN dapat dipecah menjadi 4 bagian dan dibagikan. PIN terdiri dari enam digit angka dengan digit pertama tidak 0 (nol). Untuk merekonstruksi PIN kembali, diperlukan minimal 3 *audio-share* yang dalam penelitian ini ditetapkan 3 *audio*-share, sehingga skema yang dipakai disebut skema (3, 4).

Model Dasar

Sebelum melakukan tahap enkripsi, terlebih dulu dilakukan tahap persiapan, pesan (*M*) akan diinputkan dan pilih bilangan prima (*P*) yang lebih besar dari *M*, kemudian *P* yang diperoleh akan menjadi masukan untuk mendapatkan *list* bilangan yang relatif prima dengan *P*. Selanjutnya masukkan pesan (*M*), bilangan prima (*P*), pilih juga bilangan acak sebanyak $t - 1 = 3 - 1 = 2$ yang relatif prima dengan *P* untuk menjadi masukan pada program Enkripsi dan *Embeeding*. Selanjutnya nyatakan ke dalam polinomial $a(x) = M + a_1x + a_2x^2 \ (mod\ P)$. Untuk 4 partisipan, pilih 4 buah bilangan bulat berbeda, misal x_1, x_2, x_3, dan x_4 dalam modulus *P*. Setiap partisipan akan memperoleh *share* $(x_i,\ a(x_i))$. Setelah *share* diperoleh, selanjutnya masing-masing digit angka pada setiap *share* diubah ke dalam bentuk biner ASCII, dan setiap biner *share* dibuat memiliki panjang yang sama dengan

banyak data audio yang akan disisipkan, dengan menambahkan angka 1 sebanyak data audio dikurangi panjang *share* biner. Kemudian dilakukan penyisipan *share* ke file audio dengan metode LSB, sehingga diperoleh 4 *audio* share yang akan tersimpan dalam satu folder. *Flowchart* dari tahapan enkripsi tersebut terdapat pada Gambar 1.



GAMBAR 1. *Flowchart* dari Program Persiapan dan Program Enkripsi *Embedding*

Pada tahap dekripsi dan *extracting*, partisipan memasukkan *share*-audio sebanyak 3, selanjutnya program akan membaca data audio dan akan diubah kedalam bentuk ASCII biner. Kemudian dilakukan ekstraksi digit terakhir audio dan diubah hasil ekstraksi tersebut kedalam bentuk ASCII desimal yang artinya *share* sudah didapatkan. *Share* yang sudah diperoleh tersebut dihitung menggunakan interpolasi Lagrange dan disubstitusikan x = 0 untuk mendapatkan pesan (M). *Flowchart* dari proses dekripsi dan *extracting* ini dapat dilihat pada gambar 2.

Konstruksi Program Aplikasi

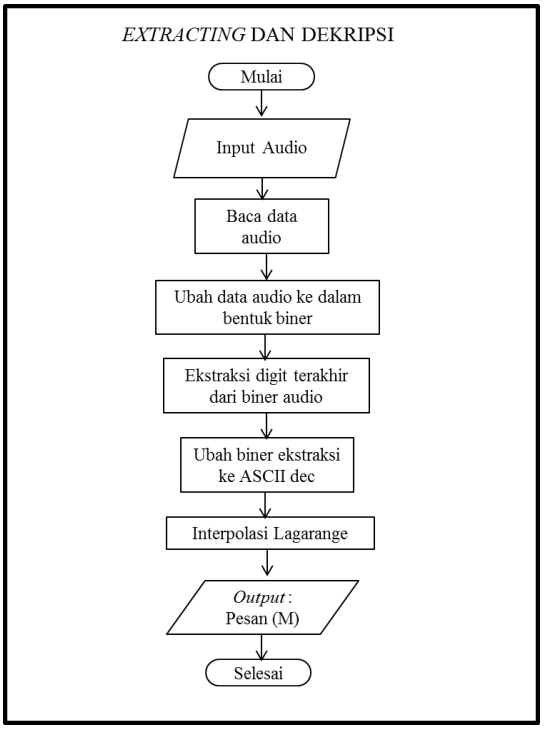
Program aplikasi terdiri dari 3 buah program, yang terdiri dari program aplikasi, program enkripsi dan *embedding*, dan program dekripsi dan *extracting*. Program persiapan bertujuan untuk memudahkan *user* memilih bilangan prima (*P*) yang lebih besar dari pesan (*M*) dan memilih bilangan acak yang relatif prima dengan *P*. Program enkripsi dan *embedding* berfungsi untuk proses enkripsi pesan menjadi *share* dan *embedding* pecahan pesan (*share*) ke dalam file audio, program dekripsi dan *extracting* bertujuan untuk mengekstraksi audio dan mengembalikan kembali pesan (*M*) semula.

Rancangan masukan (*input*) serta luaran (*output*) program dapat dilihat pada tabel 1. *Input* dari program persiapan adalah pesan yang akan disisipkan, dengan *output list* bilangan prima yang lebih besar dari pesan. Setelah diperoleh bilangan prima, kemudian *input*-kan P (yang dipilih *user*) untuk memperoleh *list* bilangan yang relatif prima dengan P. *Input* dari program enkripsi dan *embedding* adalah pesan (*M*); *P*; a_1, a_2; dan 4 file audio, dengan *output* adalah 4 buah *audio-share*. *Input* dari program dekripsi dan *extracting* adalah 3 *audio-share* dan *P*, dengan *output* berupa pesan (*M*) semula.

TABEL 1. Rancangan Input dan Output Program

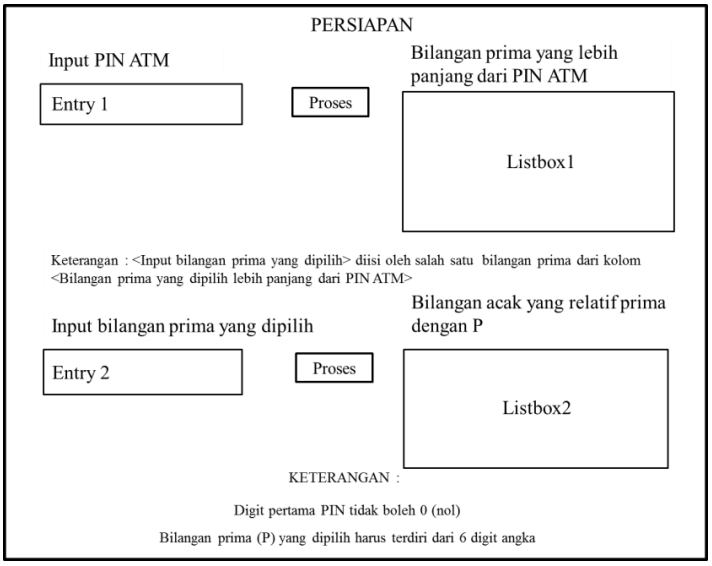
	Persiapan	**Enkripsi dan *Embedding***	**Dekripsi dan *Extracting***
Input	PIN : integer	PIN: integer P: integer a_1, a_2: integer	P: integer 3 *share-audio*: audio .*wav

	Persiapan	**Enkripsi dan *Embedding***	**Dekripsi dan *Extracting***
	P: integer (setelah P diperoleh)	4 file audio : audio .*wav	
Output	*List* bilangan prima yang lebih besar dari PIN: list (integer) *List* bilangan acak yang relatif prima dengan P: list (integer)	4 *share-audio*: audio .*wav	PIN: integer

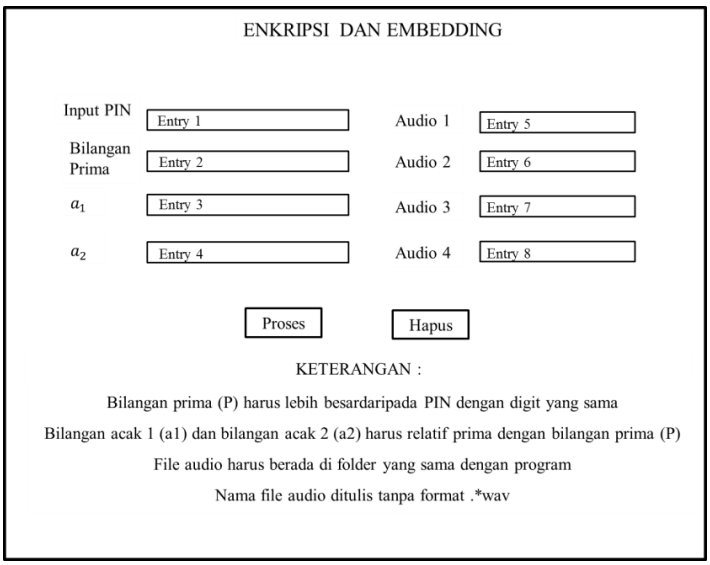


GAMBAR 2. *Flowchart* Program Dekripsi *Extracting*

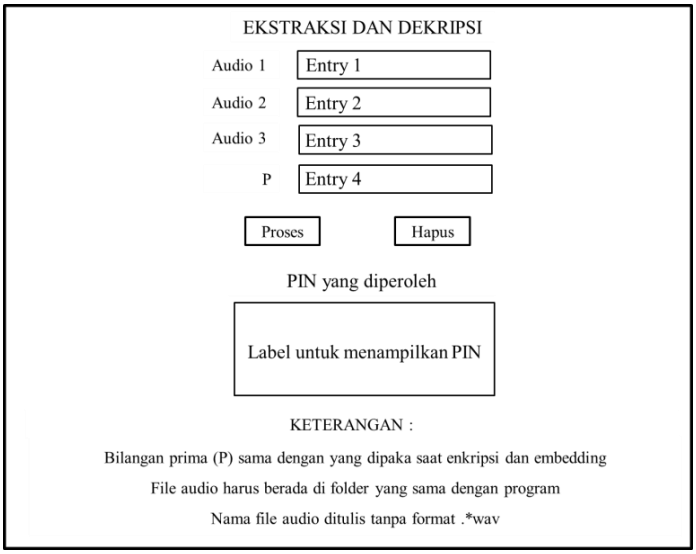
Rancangan tampilan program terdapat pada Gambar 3, Gambar 4, dan Gambar 5.



GAMBAR 3. Rancangan Tampilan Program Persiapan



GAMBAR 4. Rancangan Tampilan Program Enkripsi dan *Embedding*



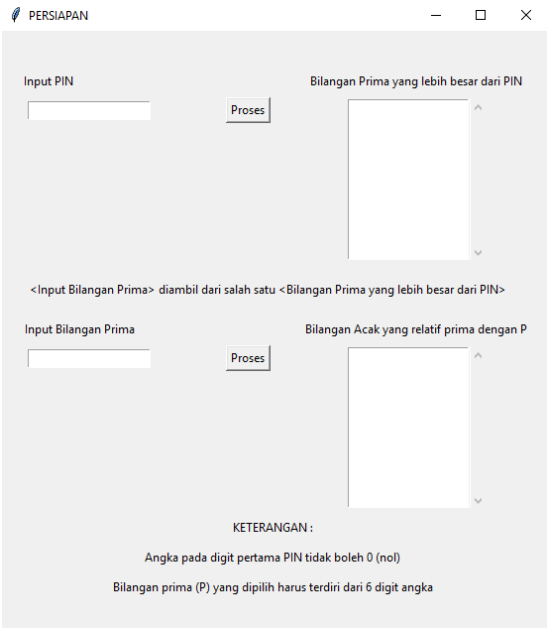
GAMBAR 5. Rancangan Tampilan Program Dekripsi dan *Extracting*

HASIL DAN DISKUSI

Program aplikasi dibuat untuk memudahkan *user* dalam melakukan enkripsi dan *embedding* juga untuk mengembalikan pesan dengan *extracting* dan dekripsi *secret sharing scheme* (PIN 6 digit) dengan skema (3, 4) dan steganografi audio LSB, dilengkapi program persiapan yang akan memudahkan mencari bilangan prima yang lebih besar dari PIN dan *list* bilangan yang relative prima dengan bilangan prima (P) yang dipilih. Program dibuat menggunakan bahasa pemrograman Python 3.10 dengan IDE (*Integrated Development Environment*) PyCharm *Community Edition* 2021.3.3.

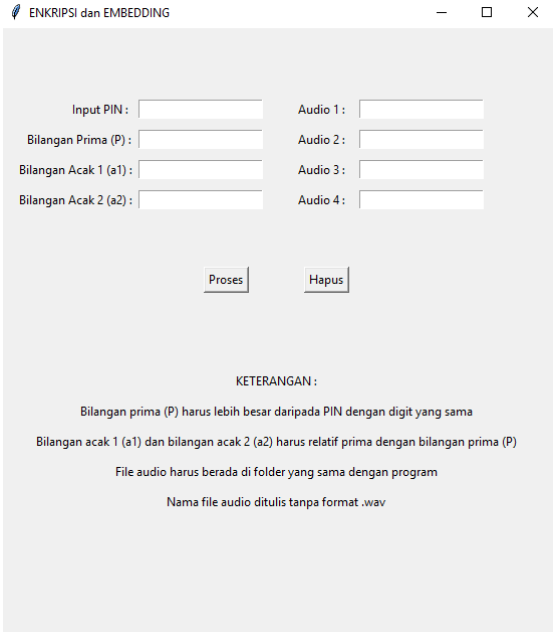
Tampilan Program Aplikasi dan Penggunaannya

Aplikasi persiapan, *user* meng-*input*-kan PIN ke *entry* paling kiri atas, lalu tekan proses sehingga akan tampil pada *listbox* 1 *list* bilangan prima yang lebih besar dari PIN. Selanjutnya *user* memilh salah satu bilangan prima dan meng-*input*-kan ke *entry* bilangan prima yang berada di sebelah kiri bawah, lalu jika tombol proses ditekan pada *listbox* 2 akan ditampilkan *list* bilangan yang relatif prima dengan *P*. Tampilan aplikasi persiapan dapat dilihat pada Gambar 6.



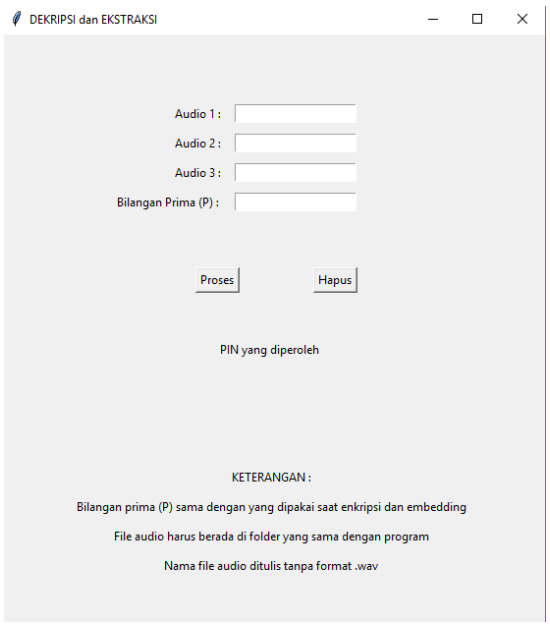
GAMBAR 6. Tampilan Program Persiapan

Pada program aplikasi enkripsi dan *embedding user* meng-*input*-kan PIN, bilangan prima (*P*), dua bilangan acak yang relatif prima dengan *P*, dan 4 buah audio dengan format .*wav. Terdapat tombol proses yang berfungsi untuk memulai proses enkripsi dan *embedding*. Selanjutnya akan dihasilkan 4 buah *audio-share* yang akan tersimpan di folder yang sama dengan program. Tampilan program enkripsi dan *embedding* dapat dilihat pada Gambar 7.



GAMBAR 7. Tampilan Program Enkripsi dan *Embedding*

Ketika partisipan membutuhkan kembali informasi PIN, partisipan dapat menggunakan aplikasi dekripsi dan *extracting*. Masukan pada program ini adalah 3 buah *audio-share* dan bilangan prima *P*. Jika *user* menekan tombol proses maka proses *extracting* dan dekripsi akan dimulai kemudian hasil konstruksi PIN akan tampil pada label. Tampilan dari program dekripsi dan *extracting* dapat dilihat pada Gambar 8.

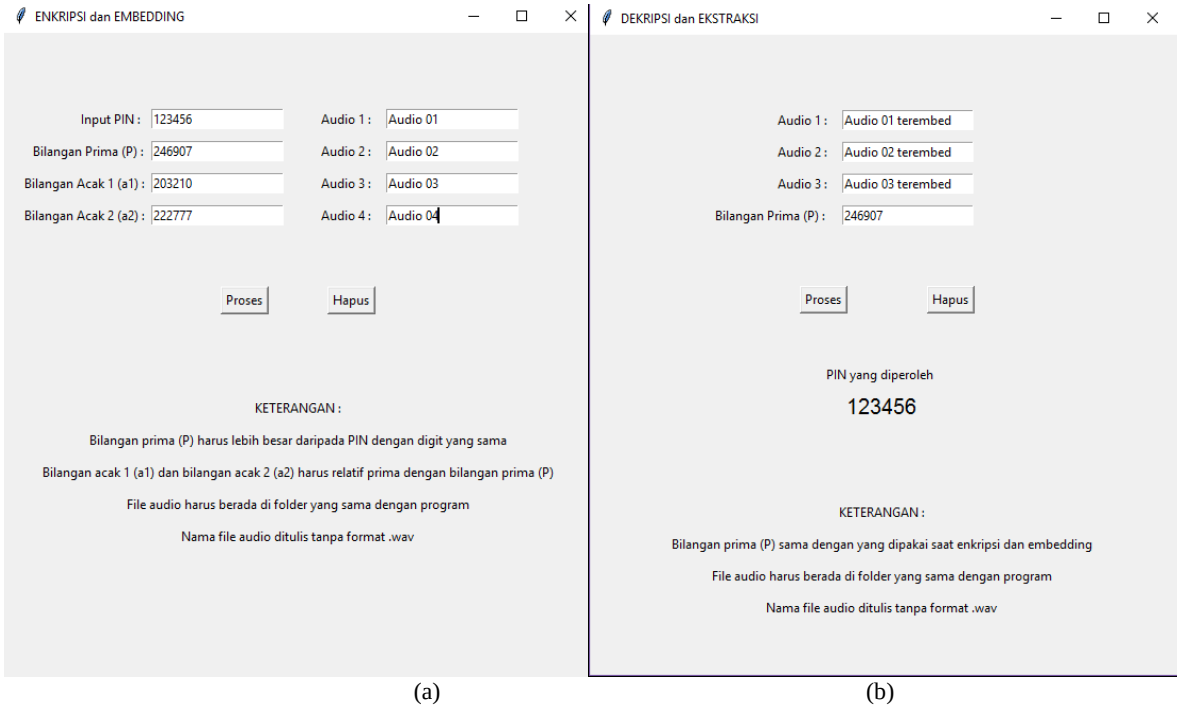


GAMBAR 8. Tampilan Program Dekripsi dan *Extracting*

Validasi

Validasi program dilakukan dengan membandingkan PIN yang menjadi masukan pada program enkripsi dan *embedding* sama dengan PIN hasil konstruksi pada program dekripsi dan *extracting*, validasi selanjutnya adalah membandingkan hasil program dengan hasil perhitungan manual.

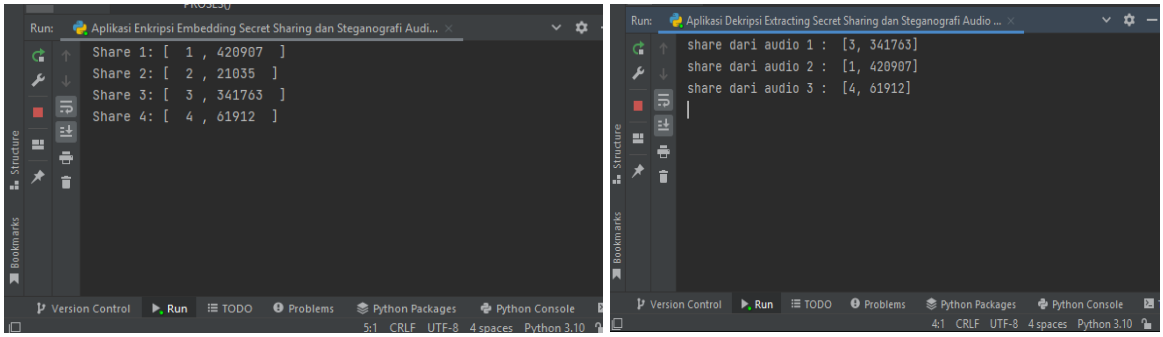
Validasi pertama dilakukan dengan menginputkan PIN = 123456, P = 246907, a_1= 203210, a_2= 222777, dan 4 file audio dengan nama file audio 01, audio 02, audio 03, dan audio 04. Hasil yang didapatkan adalah PIN hasil konstruksi program dekripsi dan *extracting* sama dengan PIN yang menjadi masukan pada program enkripsi dan *embedding*. Validasi pertama dapat dilihat pada gambar 9.



(a) (b)

GAMBAR 9. (a) Tampilan Masukan pada Program Enkripsi *Embedding*; (b) Tampilan PIN hasil konstruksi pada Program Dekripsi dan *Extracting*

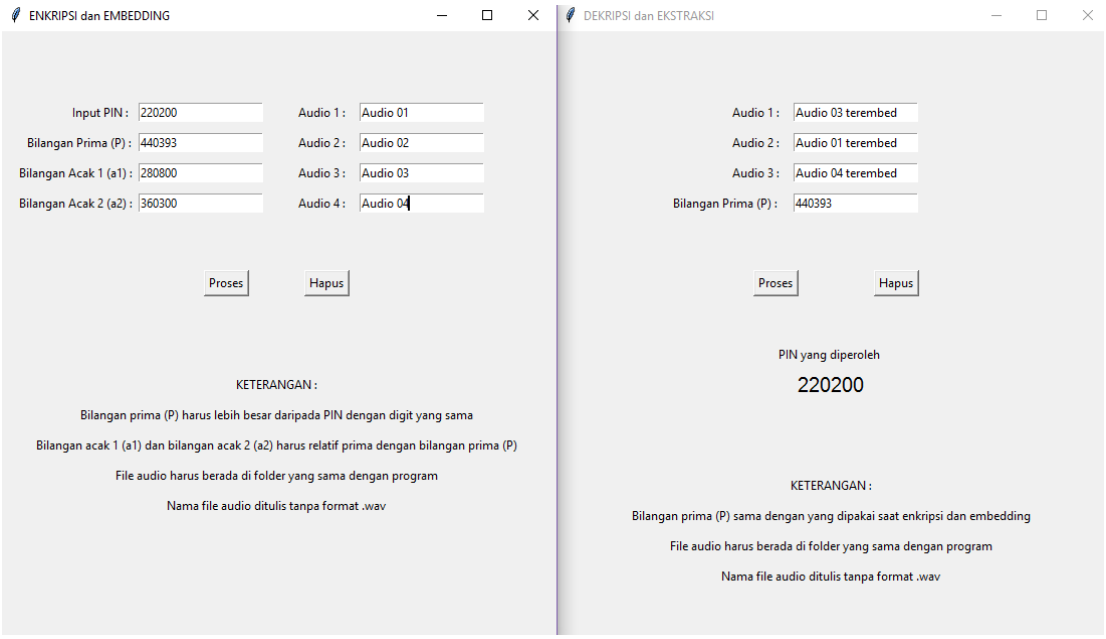
Pada percobaan lain, penulis mencoba menggunakan bilangan *input* yang berbeda dengan sebelumnya. Penulis menggunakan *PIN* = 220200, P = 440393, a_1= 280800, a_2= 360300. Dilihat berdasarkan pasangan terurut yang didapatkan, program enkripsi *embedding* dan program dekripsi *extracting* menunjukkan pasangan terurut seperti pada Gambar 10.



(a) (b)

GAMBAR 10. (a) Hasil *Share* yang Diperoleh Program Enkripsi *Embedding*; (b) Hasil *Share* yang Diperoleh Program Dekripsi dan *Extracting*

Begitu juga dengan hasil perolehan PIN pada program dekripsi *extracting* dengan program enkripsi *embedding* yang menampilkan PIN yang sama. Hal tersebut terdapat pada Gambar 11.



(a) (b)

GAMBAR 11. (a) Tampilan Masukan Baru (percobaan 2) pada Program Enkripsi *Embedding*; (b) Tampilan PIN (percobaan 2) hasil konstruksi pada Program Dekripsi dan *Extracting*

Validasi kedua adalah membandingkan hasil perhitungan program dengan hasil perhitungan manual. Berikut adalah perbandingan antara program dengan perhitungan manual.

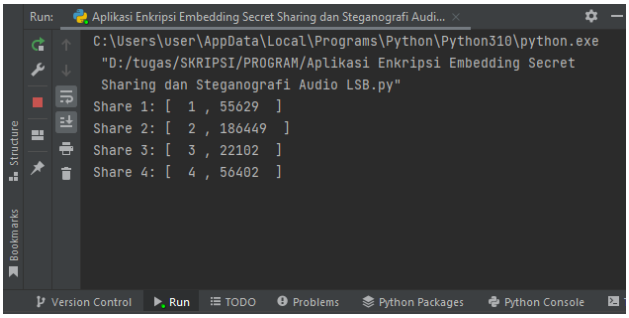
Perbandingan Hasil Perolehan Share

Perbandingan hasil perolehan *share* manual dengan perolehan *share* dari program dapat dilihat pada Gambar 12.

Share yang diperoleh manual

x	$a(x)$
1	55629
2	186449
3	22102
4	56402

Share yang diperoleh program



GAMBAR 12. Tampilan *Share* yang diperoleh secara manual dan *Share* yang Diperoleh Program

Dapat dilihat pada Gambar 12 bahwa *share* yang dihitung manual memiliki nilai pasangan terurut yang sama dengan *share* yang diperoleh program.

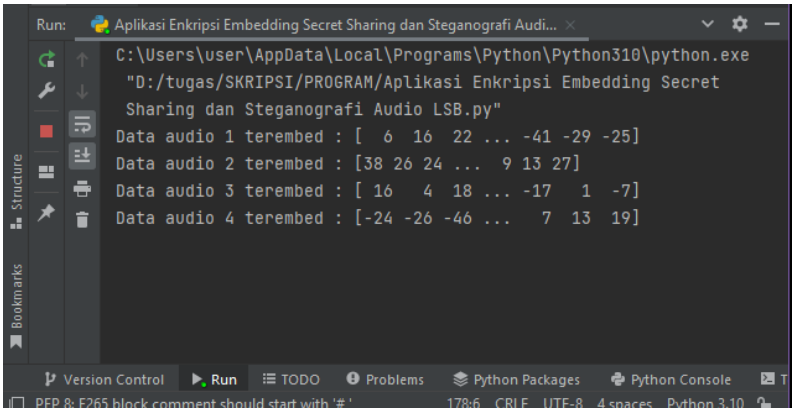
Perbandingan Data Terembed

Berikut data terembed yang penulis dapatkan dengan menyamakan *byte* dengan ASCII desimal.

TABEL 1. **DATA AUDIO TEREMBED**

Data audio 1 terembed	[6 16 22 … -41 -29 -25]
Data audio 2 terembed	[38 26 24 … 9 13 27]
Data audio 3 terembed	[16 4 18 … -17 1 -7]
Data audio 4 terembed	[-24 -26 -46 … 7 13 19

Berikut adalah data terembed yang diperoleh program, dapat dilihat pada Gambar 13.



GAMBAR 13. Tampilan *Share* yang diperoleh secara manual dan *Share* yang Diperoleh Program

Perbandingan Hasil Konstruksi PIN Menggunakan Interpolasi Lagrange

Konstruksi PIN menggunakan interpolasi Lagrange untuk mendapatkan *PIN* = 123456, menggunakan audio 01 terembed untuk mendapatkan *share* 1 yaitu [1, 55629], audio 02 terembed untuk mendapatkan *share* 02 yaitu [2, 186449], dan audio 03 terembed untuk mendapatkan *share* 3

yaitu [3, 22102], Selanjutnya dilakukan perhitungan interpolasi Lagrange secara manual, dengan membentuk polinom Lagrange berderajat t – 1 = 3 – 1 = 2 sebagai berikut.

$$L(x) = a(x_1)\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + a(x_2)\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + a(x_2)\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_1-x_2)} \pmod{246907}$$

$$= 55629\frac{(x-2)(x-3)}{(1-2)(1-3)} + 186449\frac{(x-1)(x-3)}{(2-1)(2-3)} + 22102\frac{(x-1)(x-2)}{(3-1)(3-2)} \pmod{246907}$$

$$= 55629\frac{(x-2)(x-3)}{2} - 186449(x-1)(x-3) + 11051(x-1)(x-2) \pmod{246907}$$

Karena $\frac{55629}{2}$ tidak menghasilkan bilangan bulat, maka $\frac{1}{2}$ dalam modulus 246907 adalah $2^{-1} \pmod{246907} = 123454$, sehingga

$$L(x) = (55629)(123454)(x-2)(x-3) - 189449(x-1)(x-3) + 11051(x-1)(x-2) \pmod{246907}$$

$$= 6867622566(x-2)(x-3) - 189449(x-1)(x-3) + 11051(x-1)(x-2) \pmod{246907}$$

Untuk mendapatkan PIN, polinom Lagrange di atas disubstitusi x = 0, sehingga menjadi

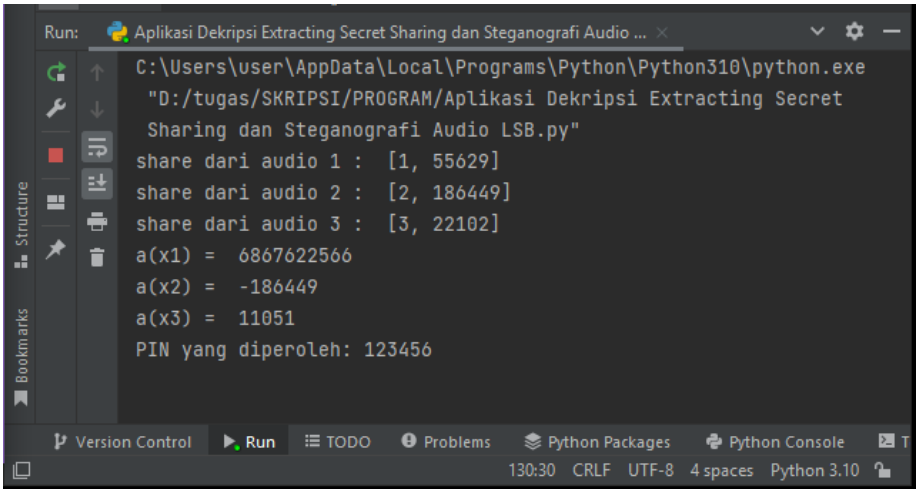
$$L(0) = 6867622566(0-2)(0-3) - 189449(0-1)(0-3) + 11051(0-1)(0-2) \pmod{246907}$$

$$= 41205735396 - 559347 + 22102 \pmod{246907}$$

$$= 41205198151 \pmod{9829}$$

$$= 123456$$

Jadi, PIN = 123456 dapat dikonstruksi kembali, yang dimana ini sama dengan hasil interpolasi yang dilakukan program dekripsi *extracting* yang dapat dilihat pada gambar 14.



GAMBAR 14. Tampilan Hasil Interpolasi Lagrange dari IDE program Dekripsi dan *Extracting*

Dikarenakan validasi dengan menyamakan PIN hasil konstruksi pada program dekripsi dan *embedding* dengan PIN yang menjadi masukan pada program enkripsi *embedding* menghasilkan PIN yang sama, juga pada validasi kedua dimana hasil perhitungan manual dan hasil perhitungan program mendapatkan hasil yang sama, maka program dinyatakan berhasil.

KESIMPULAN

Penggabungan teknik kriptografi *secret sharing scheme* dengan steganografi audio LSB ini mampu meminimalisir terjadinya kriptanalisis karena *share* akan disembunyikan keberadaannya di dalam audio lalu dikirimkan kepada yang berhak menerimanya sehingga peretas akan kesulitan mencari minimal *share-audio* yang dibutuhkan untuk menkontruksi pesan.

Dalam implementasinya, program yang dihasilkan dari penggabungan kriptografi *secret sharing scheme* dengan skema (3, 4) dan steganografi audio LSB menggunakan bahasa pemrograman Python 3.10 dapat mengenkripsi dan *embedding* PIN ke dalam audio, sehingga *audio-share* yang diperoleh tidak terdengar berbeda dengan audio yang asli. Hasil dari program dekripsi dan *extracting* adalah dapat merekonstruksi *share* menjadi PIN semula.

REFERENSI

[1] Chhadwa, H., D'souza, G., Godane, S., & Sharma P 2018, 'Audio Steganography using RSA Algorithm', International Journal of Soft Computing and Engineering, vol. 8, pp. 22-24.

[2] A. F. Humaira, "Penggabungan Kriptografi Skema Pembagian Data Rahasia dan Steganografi Audio *Least Significant Bit* (LSB)" S1 Skripsi, Universitas Pendidikan Indonesia, 2022.

[3] Munir, R. "Kriptografi" (Informatika, Bandung, 2019).

[4] Syawal, M.F., Fikriansyah, D.C. and Agani, N., 2016. Implementasi Teknik Steganografi Menggunakan Algoritma Vigenere Cipher Dan Metode LSB. *Jurnal TiCom*, *4*(3), p.91-99.

[5] Singh, P., 2016. A comparative study of audio steganography techniques. *International Research Journal of Engineering and Technology (IRJET)*, *3*(4), pp.580-585.

[6] R. Chalid, 2009 "Aplikasi Audio Steganografi untuk Melindungi Data Menggunakan Bahasa Pemrograman Java". Fakultas Teknologi Industri, Universitas Gunadarma.