BAB III

METODOLOGI PENELITIAN

3.1 Identifikasi Masalah

Keamanan serta kerahasiaan komunikasi sangat penting dalam perkembangan teknologi dewasa ini. Pengirim harus berhati-hati jika mengirimkan pesan melalui pihak ketiga, terutama jika pesan tersebut bersifat rahasia. Misalnya pada kasus pengamanan PIN ATM seorang pengusaha dengan menggunakan pembagian PIN, sehingga PIN dapat dipecah menjadi 4 bagian, sehingga potongan dari PIN tersebut dapat dibagikan. PIN ATM terdiri dari 6 digit angka dengan digit pertama tidak 0 (nol). Pengamanan PIN ATM ini dapat dilakukan dengan cara kriptografi, teknik kriptografi yang dipakai untuk membagi PIN menjadi 4 bagian adalah dengan menggunakan skema pembagian rahasia. Namun, jika hanya menggunakan kriptografi saja, pihak lain (peretas) dapat menemukan dengan mudah potongan informasi tersebut. Maka diperlukan pengamanan lebih lanjut yaitu dengan menggunakan steganografi. Steganografi yang digunakan pada penelitian ini adalah steganografi audio dengan metode *least significant bit* (*LSB*). PIN yang sudah dibagi menjadi 4 tersebut disisipkan ke file audio sehingga sang pengusaha hanya perlu membagikan *4 audio-share* tersebut kepada orang-orang yang sudah diberikan kepercayaan menyimpan *audio-share*. Untuk menkontruksi PIN ATM kembali, diperlukan minimal 3 *audio-share* lalu dilakukan ekstraksi, yang dalam penelitian ini ditetapkan 3 *audio-share*.

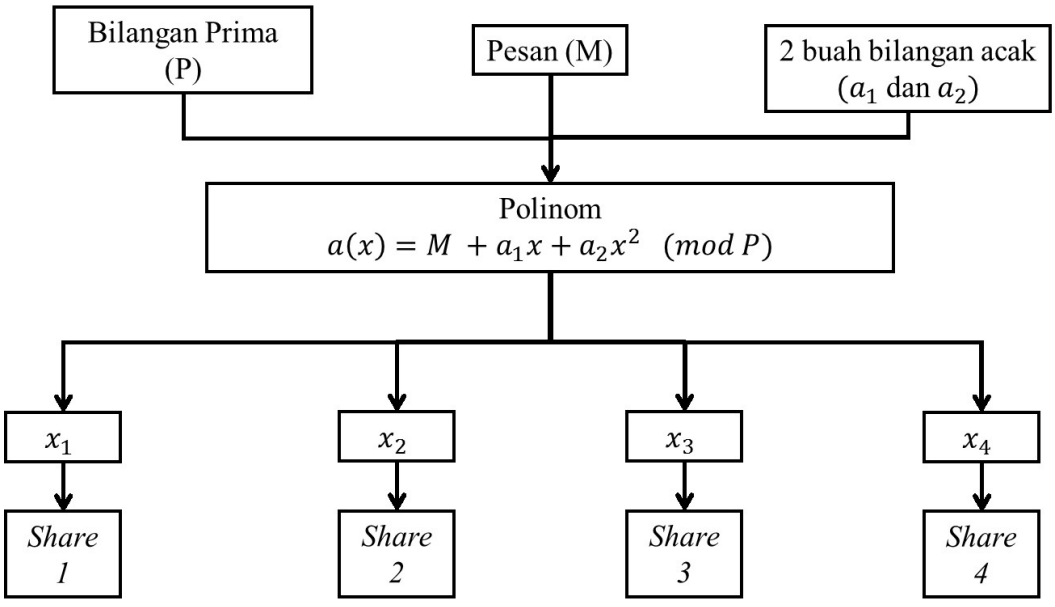
3.2 Model Dasar

Teknik Kriptografi yang digunakan dalam penelitian ini adalah skema pembagian data rahasia (*secret sharing scheme*), dan teknik steganografi yang digunakan dalam penelitian ini adalah teknik *Least Significant Bit* (LSB) dengan media penyematan yaitu audio.

3.2.1 Skema Pembagian Data Rahasia (3, 4)

Skema pembagian rahasia dilakukan untuk membagi pesan kedalam beberapa *share* sehingga peretas tidak akan tahu isi pesan tersebut jika *share* tidak memenuhi dengan n yang ditetapkan *dealer* dan penyimpanan pesan tidak harus terpusat serta tidak memerlukan kunci ketika akan mekonstruksi pesan. Dalam proses ini penulis menggunakan skema (3,4) agar memudahkan dalam membuat tempat masukan (*input*) audio ketika pesan akan dikonstruksi kembali.

Pada tahap ini, akan di-*input* pesan (*M*) , lalu pilih bilangan prima (*P*) yang harus lebih besar dari *M* dan juga lebih besar dari partisipan, yang dalam penelitian ini banyaknya partisipan adalah 4. Nilai *P* tidak perlu dirahasiakan. Pilih bilangan acak $t-1$ buah bilangan bulat dalam modulus *P*, sebut bilangan-bilangan bulat tersebut adalah a_1, dan a_2 , lalu dinyatakan ke dalam polinomial $a(x) = M + a_1x + a_2x^2 \ (mod\ P)$ sedemikian sehingga $(0) = M \ (mod\ P)$. Polinom $a(x)$ harus dirahasiakan. Untuk 4 partisipan, pilih 4 buah bilangan bulat berbeda, misal x_1, x_2, x_3, dan x_4 dalam modulus *P*. Setiap orang memperoleh *share* (x_i, y_i) yang dalam hal ini $y_i = a(x_i)\ (mod\ P)$.

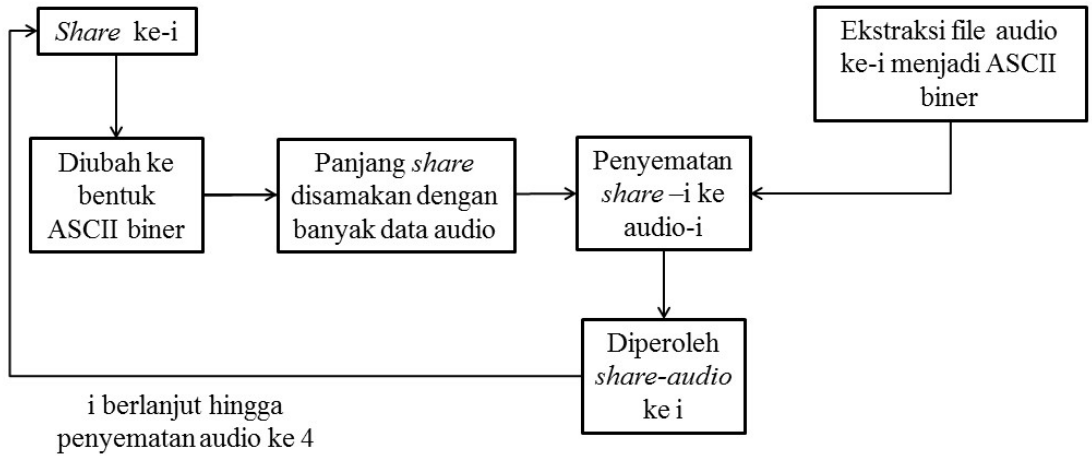


Gambar 3.1 Skema Pembagian Data Rahasia (3, 4)

3.2.2 Skema Steganografi Audio

Steganografi merupakan salah satu cara untuk meningkatkan keamanan pesan dengan menyembunyikan pesan ke suatu media. Metode *Least Significant Bit* (LSB) merupakan salah satu metode steganografi dengan cara mengganti bit terakhir dengan pesan yang sudah diubah ke dalam bentuk biner. Media yang dipakai dalam penelitian ini adalah file audio dengan format .*wav.

Pada tahap sebelumnya telah didapatkan 4 buah *share*. Masing-masing digit angka pada setiap share akan diubah ke dalam bentuk ASCII biner, setelah itu setiap *share* dibuat memiliki panjang yang sama dengan audio yang akan disisipkan, dengan menambahkan angka 1 sebanyak banyak data audio dikurangi panjang share biner. Kemudian untuk setiap *share* akan disisipkan ke file audio dengan metode LSB. Sehingga akan diperoleh 4 *audio-share* yang akan tersimpan di satu folder.



Gambar 3.2 Skema Steganografi Audio

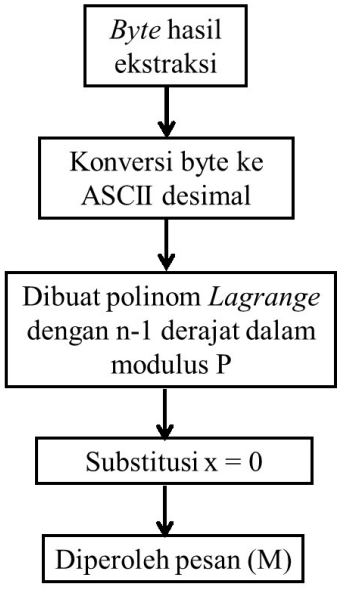
3.2.3 Skema Ekstraksi Steganografi LSB

Untuk mendapatkan pesan (*M*) kembali dibutuhkan *share* sebanyak *t* yang sudah ditetapkan saat proses enkripsi. Ekstraksi steganografi dengan metode LSB pada *Audio-share* dimulai dengan pembacaan data audio, lalu dikonversi ke bentuk biner ASCII. Kemudian diambil angka terakhir dari setiap *byte* dan dikelompokkan setiap 8 digit. Hal itu berlangsung hingga ekstraksi audio ke-3.

Gambar 3.3 Skema Ekstraksi Steganografi Metode LSB

3.2.4 Skema Konstruksi Data Rahasia

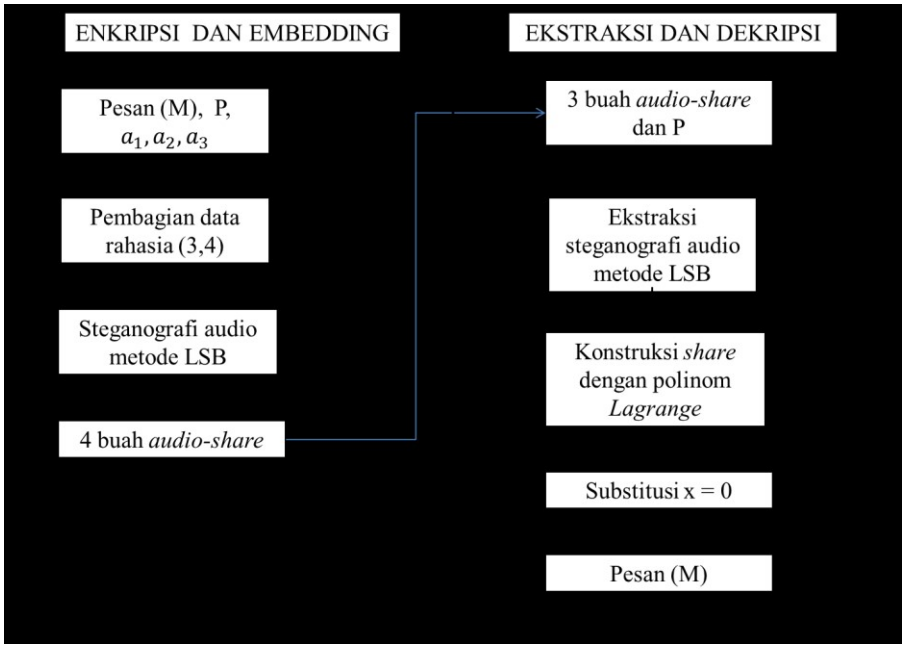
Hasil ekstraksi LSB kemudian dikonversi ke dalam bentuk ASCII desimal dan dibuat polinom *Lagrange* dengan $t - 1 = 3 - 1 = 2$ derajat dalam modulo P. Substitusi $x = 0$ untuk mendapatkan pesan (M) kembali.



Gambar 3.4 Skema Konstruksi Data Rahasia

3.2.5 Skema Penggabungan Pembagian Data Rahasia dan Steganografi Audio Metode LSB

Skema penggabungan kriptografi pembagian data rahasia dan steganografi audio menggunakan metode LSB sebagai berikut.



Gambar 3.5 Skema penggabungan kriptografi pembagian data rahasia dan steganografi audio menggunakan metode LSB

3.3 Konstruksi Program Aplikasi

Adapun prosedur konstruksi program aplikasi dalam penelitian ini yaitu:

3.3.1 Perancangan Program Aplikasi

Program aplikasi ini terdiri dari 3 buah program, yaitu program persiapan, program enkripsi dan *embedding*, juga program dekripsi dan *extracting*. Program persiapan bertujuan untuk memudahkan pengguna memilih bilangan prima (P) yang lebih besar dari panjang pesan (M) dan memilih bilangan acak yang relatif prima dengan P, program enkripsi dan *embedding* untuk proses enkripsi pesan dan *embedding* pesan ke dalam file audio, program *extracting* dan dekripsi untuk proses *extracting* serta merekonstruksi kembali pesan.

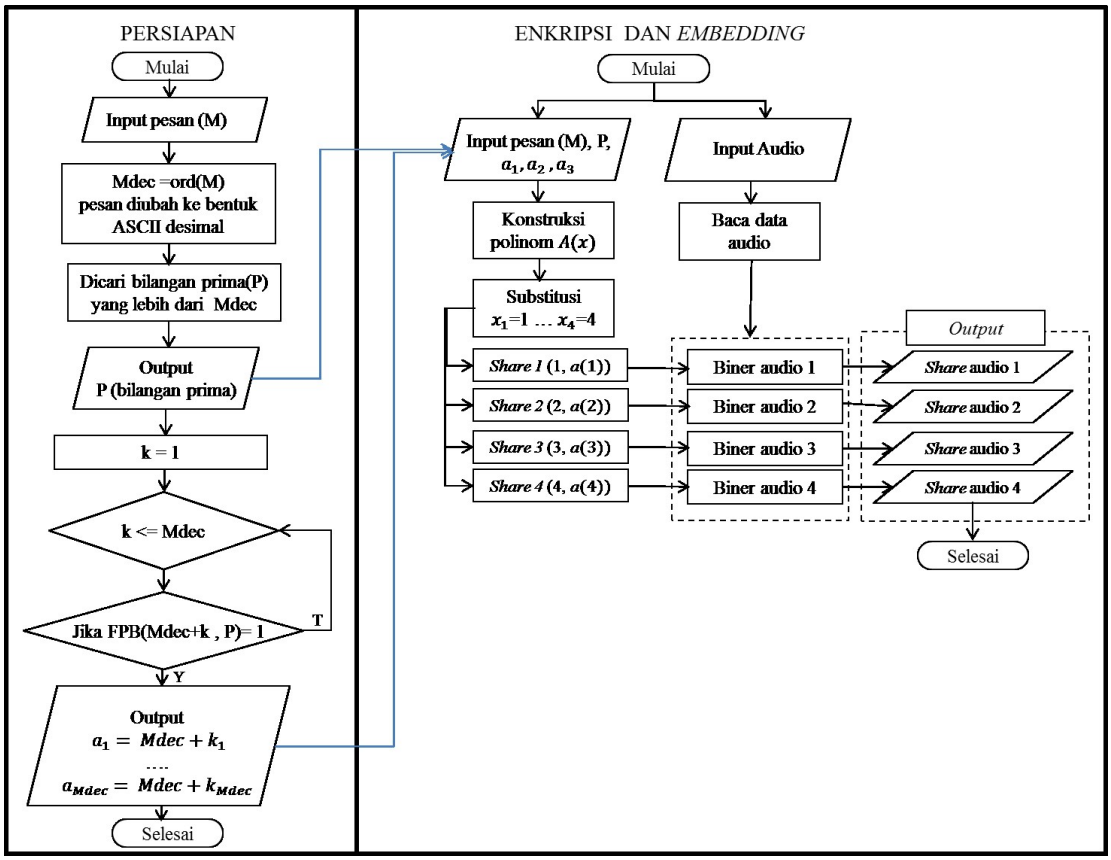
Pada tahap ini, dilakukan perancangan masukan (*input*) apa saja yang diperlukan beserta luaran (*output*) dari setiap program. Input dari program persiapan adalah pesan yang akan disisipkan, dengan *output list* bilangan prima yang lebih besar dari pesan. Setelah diperoleh bilangan prima *P*, lalu *input*-kan *P* (yang dipilih oleh *user*) untuk mendapatkan bilangan acak yang relatif prima dengan *P*. *Input* dari program enkripsi dan *embedding* adalah pesan, *P*; a_1, a_2; dan 4 file audio, dengan *output* adalah 4 *share-audio*. *Input* dari program dekripsi dan *extracting* adalah 3 *share audio*, dengan *output* adalah pesan (*M*) yang sudah dikonstruksi.

Tabel 3.1

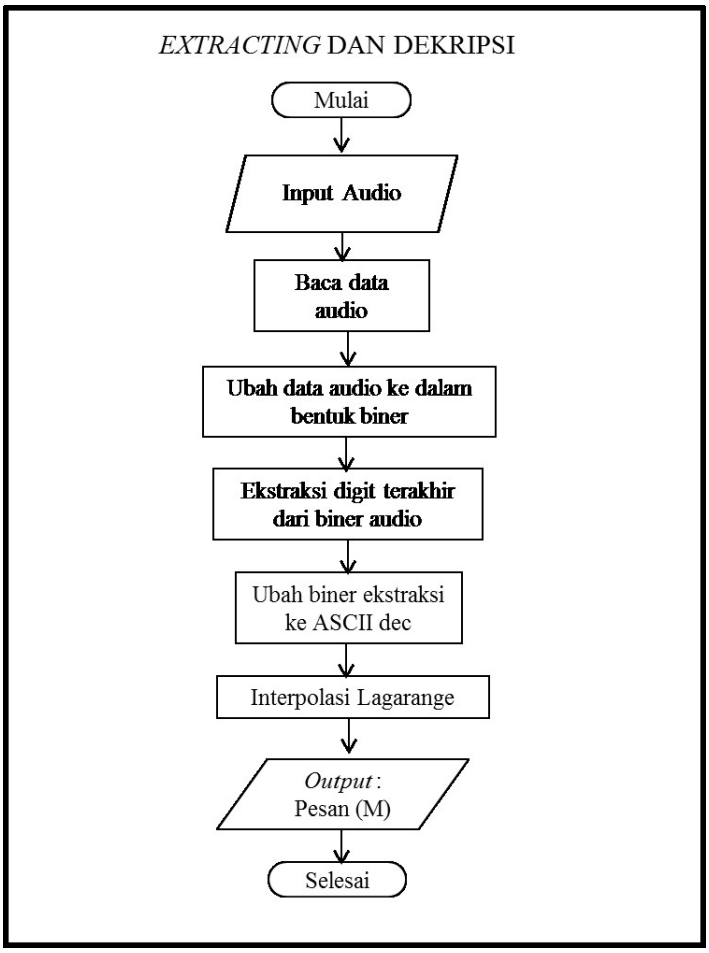
Rancangan Input dan Output Program

	Persiapan	Enkripsi dan *Embedding*	Dekripsi dan *Extracting*
Input	PIN: integer P: integer (setelah P diperoleh)	PIN: integer P: integer $\boldsymbol{a_1, a_2}$: integer 4 file audio: audio .*wave	P: integer 3 *share-audio:* audio .*wav
Output	Bilangan prima yang lebih besar dari PIN: integer Bilangan acak yang relatif prima dengan P: integer	4 *share-audio*: audio .*wav	PIN: integer

Berikut diagram alir (*flowchart*) dari rancangan program.



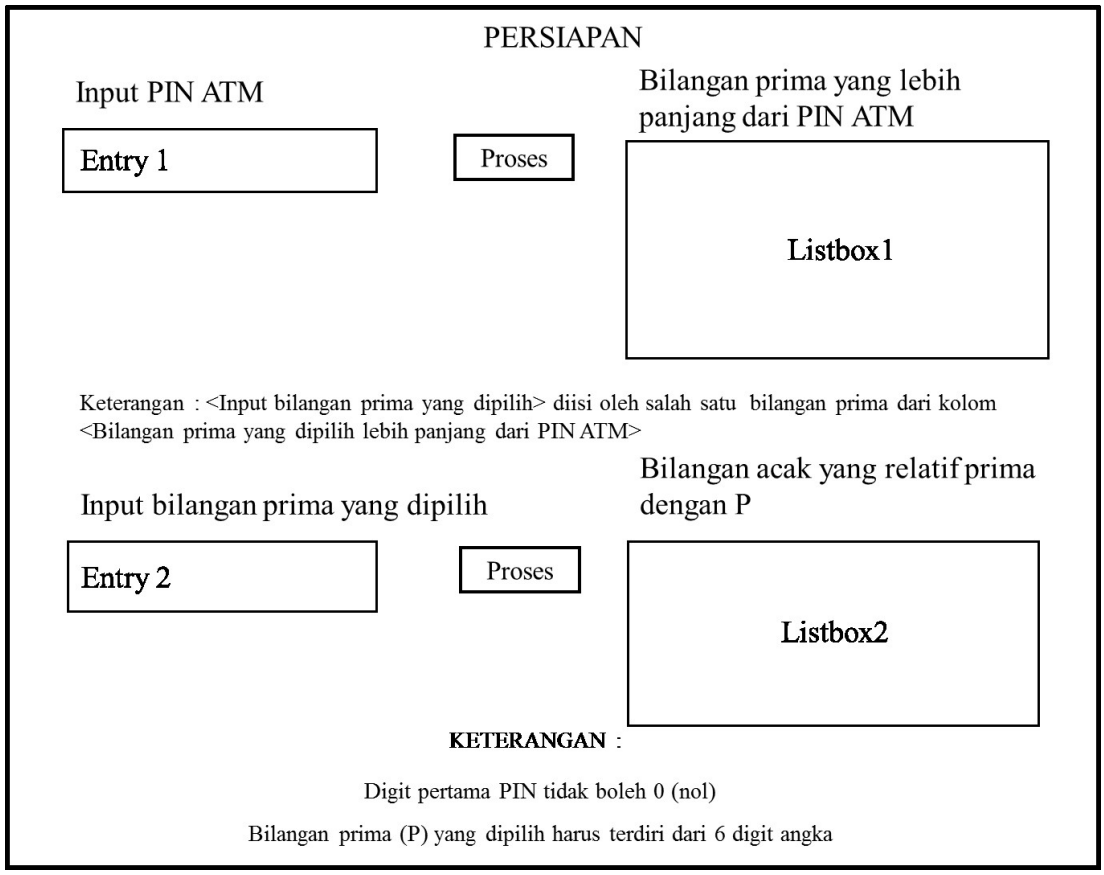
Gambar 3.6 *Flowchart* Program Persiapan dan Program Enkripsi dan *Embedding*



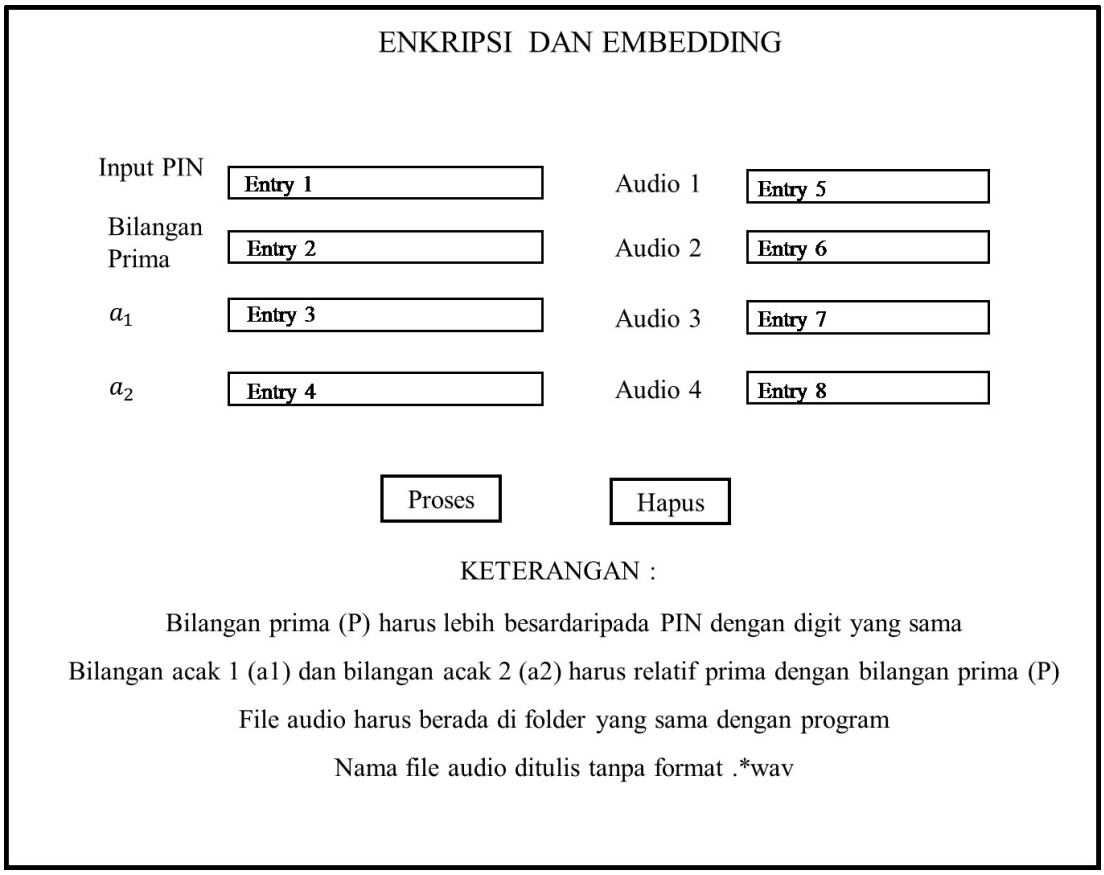
Gambar 3.7 *Flowchart* Program Dekripsi dan *Extracting*

3.3.2 Rancangan Tampilan Program Aplikasi

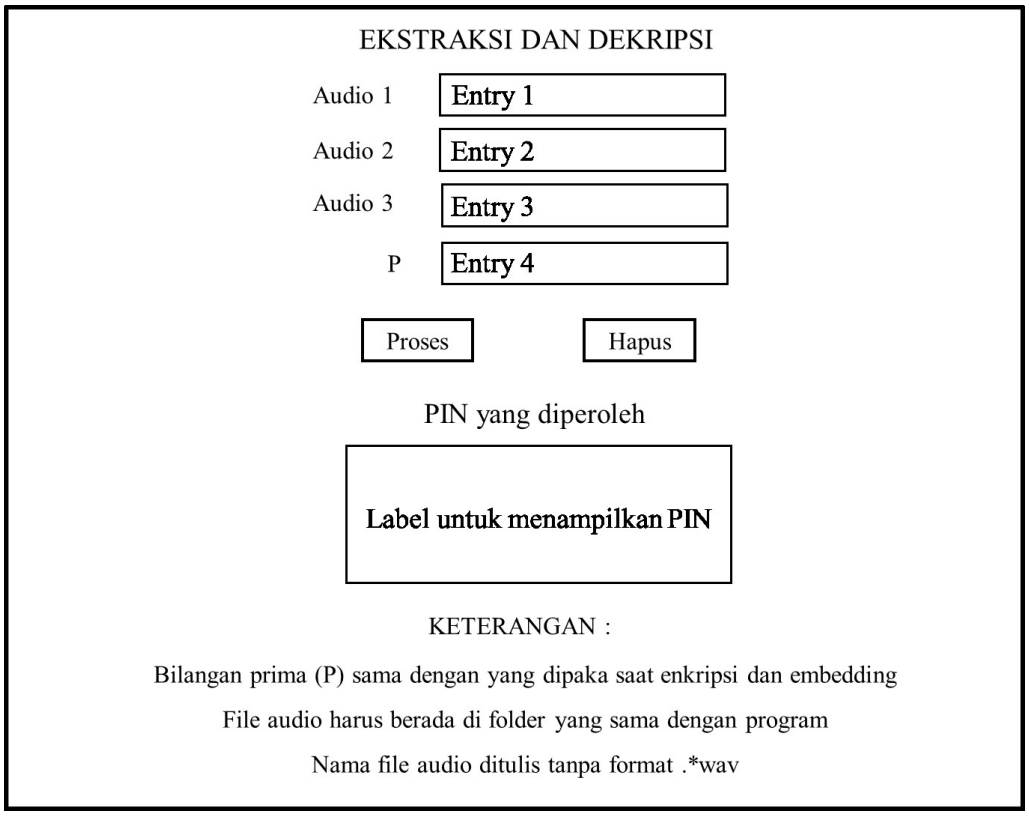
Program aplikasi menggunakan aplikasi *Python* 3.10 dan IDE PyCharm dengan tujuan untuk memudahkan proses enkripsi dan dekripsi serta validasi pada penggabungan skema pembagian data rahasia dengan steganografi audio metode *least significant bit* (LSB). Tampilan program aplikasi dapat dilihat seperti berikut:



Gambar 3.8 Rancangan Tampilan Program Aplikasi Persiapan



Gambar 3.9 Rancangan Tampilan Program Aplikasi Enkripsi dan *Embedding*



Gambar 3.10 Rancangan Tampilan Program Aplikasi Dekripsi dan *Extracting*

3.5 Implementasi dan Validasi

Implementasi dilakukan pada PIN ATM seorang pengusaha yang terdiri dari 6 digit angka dengan digit pertama tidak 0 (nol). PIN tersebut akan diamankan dengan cara PIN dipecah menjadi 4 bagian menggunakan pembagian data rahasia, kemudian agar tidak terdeteksi peretas potongan PIN (*share*) akan disisipkan ke dalam audio berformat *wave*.

Pada tahap ini, dilakukan validasi terhadap program. Tahap validasi dilakukan untuk mengetahui apakah *share* hasil skema pembagian rahasia yang disisipkan (*embedding*) pada audio dapat dikonstruksi kembali menjadi *plaintext* pada proses *extracting* LSB dan dekripsi skema pembagian rahasia, selain itu juga pengecekan program dilakukan dengan membandingkan hasil perhitungan program dan hasil perhitungan manual. Jika pesan hasil konstruksi sama dengan *plaintext* dan hasil

perhitungan program sama dengan hasil perhitungan manual, maka program dinyatakan berhasil.