BAB III
METODOLOGI PENELITIAN

3.1 Identifikasi Masalah

Penelitian ini menggabungkan teknik kriptografi *hybrid* AES dan ECC dengan steganografi *Least Significant Bit* dengan mengimplementasikan AES dan ECC pada pesan teks hingga diperoleh cipherteks lalu cipherteks tersebut akan disisipkan ke dalam *cover image* sehingga diperoleh hasil akhir yaitu *stego-image*. Program yang dibangun terdiri dari 6 fungsi utama, yaitu enkripsi dan dekripsi menggunakan algoritma AES, pembangkitan kunci ECC, enkripsi dan dekripsi menggunakan algoritma ECC, *embedding* dan ekstraksi menggunakan *Least Significant Bit*.

Proses pembangkitan kunci ECC dilakukan oleh pengirim dan penerima pesan, sedangkan proses enkripsi hanya dilakukan oleh pengirim dan proses dekripsi hanya dilakukan oleh penerima. Pada proses enkripsi algoritma AES diperlukan *input* plainteks dan kunci rahasia lalu diperoleh *output* berupa cipherteks. Kemudian proses pembangkitan kunci ECC diperlukan *input* untuk mencari titik-titik pada persamaan kurva eliptik (a, b, p) lalu diperoleh *output* berupa titik-titik yang berada pada kurva eliptik tersebut. Selanjutnya pada proses enkripsi algoritma ECC diperlukan *input* kunci rahasia AES dan titik absis, lalu diperoleh *output* berupa cipherteks. Pada proses steganografi dibutuhkan *input* file gambar, cipherteks hasil enkripsi kriptografi AES dan kriptografi ECC, dan kunci *Least Significant Bit* lalu diperoleh *output stego-image* yang sudah disisipkan pesan.

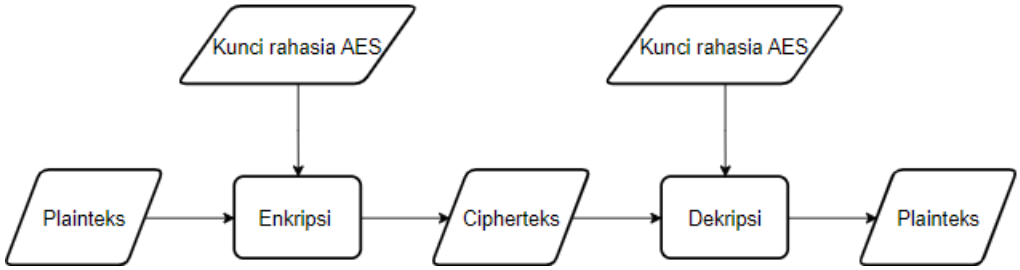
Pada proses ekstraksi dibutuhkan *input stego-image* yang sudah disisipkan pesan dan kunci *Least Significant bit* lalu diperoleh *output* berupa cipherteks yang sudah disisipkan pada *stego-image*. Kemudian pada proses dekripsi kriptografi ECC dibutuhkan input kunci ECC dan cipherteks dari kunci rahasia AES dan menghasilkan *output* berupa kunci rahasia AES. Selanjutnya pada proses dekripsi kriptografi AES dibutuhkan *input* cipherteks dan kunci rahasia AES.

3.2 Model Dasar

Model dasar yang digunakan dalam penelitian ini adalah algoritma kriptografi AES, algoritma kriptografi ECC, dan steganografi *Least Significant Bit*.

3.2.1 Advanced Enryption Standard

Skema kriptografi AES berdasarkan yang dipaparkan dalam BAB II bagian 2.8 akan ditunjukkan dalam Gambar 3.1.



Gambar 3.1 Skema Kriptografi AES

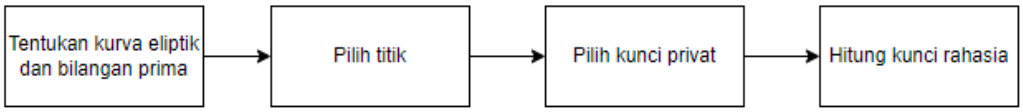
Pada algoritma kriptografi AES, pengirim melakukan proses enkripsi AES dengan menggunakan kunci dan plainteks yang diinginkan, kemudian cipherteks hasil enkripsi bersama dengan kunci rahasia baru dikirimkan kepada penerima untuk proses dekripsi. Seperti yang telah dijelaskan sebelumnya, proses enkripsi dalam AES melibatkan transformasi data secara berulang menggunakan serangkaian putaran (*rounds*) yang melibatkan substitusi, pergeseran, dan pencampuran bit yang membuatnya sangat sulit untuk dipecahkan tanpa menggunakan kunci yang tepat.

Selain itu, AES juga menerapkan operasi bitwise, yaitu operasi dalam representasi biner dari data. Operasi bitwise memungkinkan untuk dilakukan secara paralel dengan kecepatan yang tinggi menggunakan perangkat komputer. Namun, salah satu kekhawatiran yang muncul adalah ketika kunci rahasia AES harus ditransmisikan dari satu pihak ke pihak lain secara aman. Jika kunci tersebut jatuh ke tangan yang salah atau disadap oleh pihak yang tidak berwenang, maka keamanan data yang dienkripsi menggunakan kunci tersebut dapat terancam.

3.2.2 Elliptic Curve Cryptography

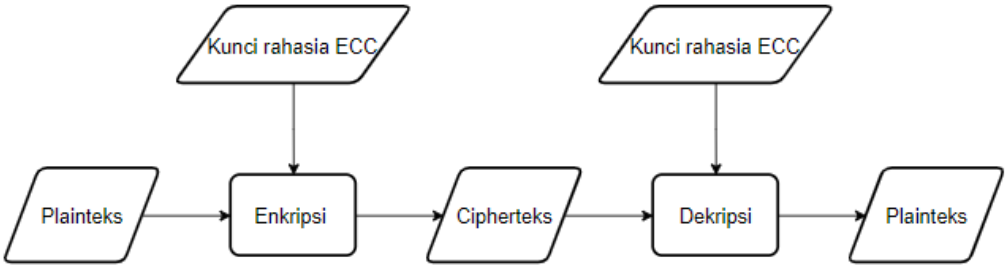
Elliptic Curved Cryptography (ECC) adalah kurva matematik yang beroperasi dalam lapangan (*field*). Oleh karena itu operasi penjumlahan dua buah

titik di dalam kurva eliptik selalu menghasilkan titik yang terletak di kurva eliptik tersebut. Skema pembangkitan kunci ECC ditunjukkan pada Gambar 3.2.



Gambar 3.2 Skema Pembangkitan Kunci ECC

Langkah pertama yang dilakukan adalah menentukan kurva eliptik dan lapangan berhingga (*galois field)* yang akan dipakai untuk proses enkripsi sehingga diperoleh himpunan penyelesaiannya. Selanjutnya yaitu menentukan kunci privat yang akan dipakai untuk proses enkripsi. Selanjutnya yaitu memilih titik awal kurva dari salah satu titik di himpunan penyelesaian yang telah diperoleh. Setelah memiliki kurva eliptik, kunci privat, dan titik awal kurva, selanjutnya menghitung kunci publik dengan cara mengalikan kunci privat dengan titik awal kurva. Kunci publik ini yang nantinya akan dipakai pada proses enkripsi dan dekripsi. Skema kriptografi ECC ditunjukkan pada Gambar 3.3.

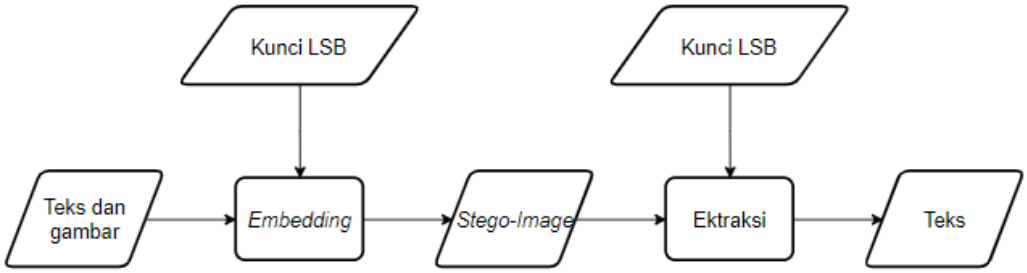


Gambar 3.3 Skema Kriptografi ECC

3.2.3 Least Significant Bit

Least Significant Bit adalah metode steganografi dimana bit terakhir pada suatu media diganti dengan pesan yang sudah diubah bentuknya ke dalam biner. Pada tahap *embedding*, masing-masing digit angka pada setiap pesan yang sudah diubah ke dalam bentuk biner akan disisipkan ke file gambar (*cover-image*) dengan menggunakan LSB sehingga akan diperoleh *stego-image*. Pemilihan lokasi *pixel* yang akan disisipkan pesan dipilih secara acak dengan metode *pseudorandom number generator* (PRNG) menggunakan *seed*. Kemudian pada tahap ekstraksi dilakukan dengan cara membangkitkan PRNG menggunakan seed yang sama untuk mengetahui lokasi *pixel*. Selanjutnya bit-bit tersebut disusun dan dikelompokkan

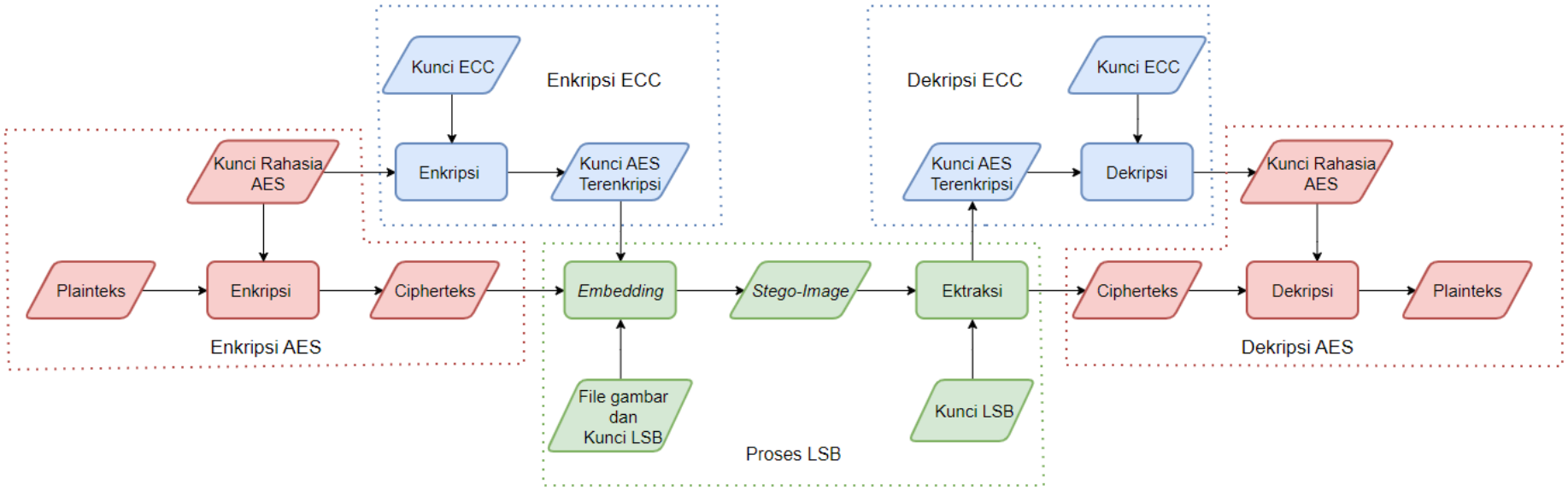
setiap 8 digit. Bit-bit yang sudah dikelompokkan diubah ke dalam huruf sehingga diperoleh pesan berupa teks. Skema dari steganografi LSB ditunjukkan pada Gambar 3.4.



Gambar 3.4 Skema Steganografi LSB

3.3 Pengembangan Model Dasar

Implementasi penggabungan algoritma kriptografi AES, algoritma kriptografi ECC, dan algoritma steganografi LSB bertujuan untuk meningkatkan keamanan pada pesan rahasia. Dari permasalahan transmisi kunci yang ditemukan saat implementasi algoritma AES, maka algoritma kriptografi ECC akan diterapkan agar pertukaran kunci AES lebih aman. Penggabungan kedua algoritma kriptografi ini disebut juga dengan kriptografi *hybrid*. Kemudian algoritma steganografi bertujuan untuk menyembunyikan pesan yang telah di enkripsi guna mengurangi kecurigaan dari pihak ke-3. Skema implementasi pengembangan model dasar ini akan ditunjukkan dalam Gambar 3.5.



Gambar 3.5 Skema Pengembangan Model Dasar

Dalam implementasi pengembangan modelnya, pengirim dan penerima membangkitkan kunci dengan algoritma ECC, hingga diperoleh kunci ECC. Kemudian, pengirim mengenkripsi pesan menggunakan algoritma AES sehingga menjadi cipherteks. Setelah itu kunci rahasia AES akan dienkripsi menggunakan algoritma ECC. Selanjutnya, pengirim memilih file gambar yang akan disisipkan teks dan memasukkan kunci LSBnya.

Selanjutnya dilakukan proses *embedding* dari kunci AES yang telah dienkripsi dan cipherteks yang diperoleh ke dalam gambar hingga diperoleh *stego-image*. Pada proses dekripsi, penerima menyiapkan *stego-image* dan kunci LSB untuk melakukan proses ekstraksi hingga diperoleh teks yang telah disisipkan ke dalam gambar. Selanjutnya, pihak penerima mendekripsi kunci AES terenkripsi dengan menggunakan kunci ECC yang dimiliki. Kemudian setelah terdekripsi, kunci AES tersebut digunakan untuk mendekripsi cipherteks.

3.4 Konstruksi Program Aplikasi

Pembuatan program aplikasi dari hasil penggabungan algoritma kriptografi AES, algoritma kriptografi ECC, dan algoritma steganografi LSB ini akan dikonstruksi menggunakan *Graphical User Interface* (GUI) dalam bahasa pemrograman *Phyton*. Program ini akan memiliki beberapa pilihan menu seperti pembangkitan kunci ECC, enkripsi, dekripsi, *embedding*, dan ekstraksi.

3.4.1 Input dan Output

Input untuk progam aplikasi ini berupa teks dan gambar. Teks akan dienkripsi sehingga diperoleh *cipherteks* lalu cipherteks tersebut akan diembedding ke dalam gambar sehingga diperoleh hasil akhir yaitu *stego-image*.

3.4.2 Algoritma Deskriptif

Berikut adalah algoritma deskriptif program aplikasi yang akan dikonstruksi:

Enkripsi	Dekripsi
1. Siapkan teks yang akan diamankan dan gambar yang akan disisipkan pesan. 2. Pengirim dan penerima melakukan proses pembangkitan kunci ECC	1. Siapkan *stego-image* yang ingin diekstraksi 2. Penerima melakukan proses ekstraksi gambar menjadi pesan dengan menggunakan kunci LSB yang telah dibangkitkan hingga

Enkripsi	Dekripsi
dengan menggunakan algoritma pembangkitan kunci ECC. 3. Pengirim dan penerima melakukan proses pembangkitan kunci LSB dengan menggunakan *pseudorandom number generator* dan metode LCG. 4. Pengirim melakukan proses enkripsi pesan dengan menggunakan kunci AES hinga diperoleh cipherteks 5. Pengirim melakukan proses enkripsi untuk kunci rahasia AES dengan menggunakan kunci ECC yang telah dibangkitkan hingga diperoleh cipherteks dari kunci rahasia AES. 6. Pengirim melakukan proses embedding pesan ke dalam gambar yang telah disiapkan dengan menggunakan kunci LSB hingga diperoleh hasil akhir yaitu *stego-image*.	diperoleh teks yang disembunyikan pada *stego-image*. 3. Penerima melakukan proses dekripsi kunci rahasia AES dengan menggunakan kunci ECC yang telah dibangkitkan hingga diperoleh plainteks dari kunci rahasia AES. 4. Penerima melakukan proses dekripsi pesan dengan menggunakan kunci rahasia AES yang telah didapatkan hingga diperoleh plainteks.

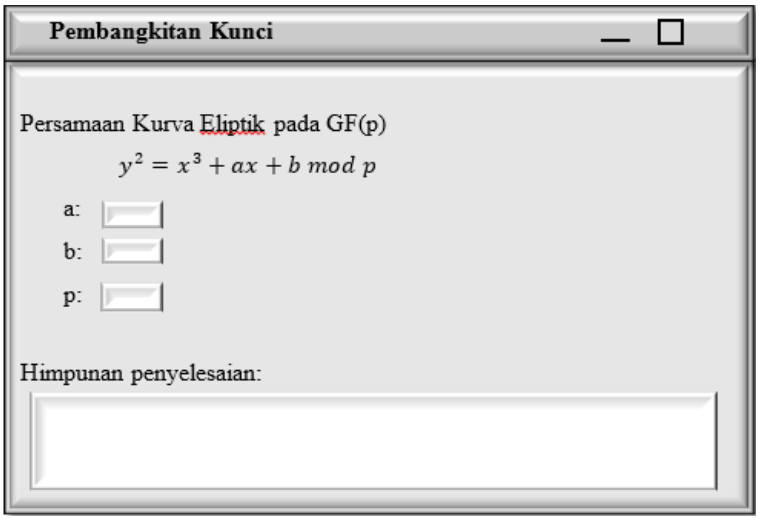
3.4.3 Rancangan Tampilan Program Aplikasi

Rancangan tampilan program aplikasinya akan memiliki menu utama yang berisi 6 tab, yaitu tab pembangkitan kunci ECC, operasi titik, enkripsi, dekripsi, *embedding*, dan ekstraksi. Rancangan tampilan program aplikasi yang akan dibuat ada pada gambar-gambar dibawah ini.

a. Menu utama

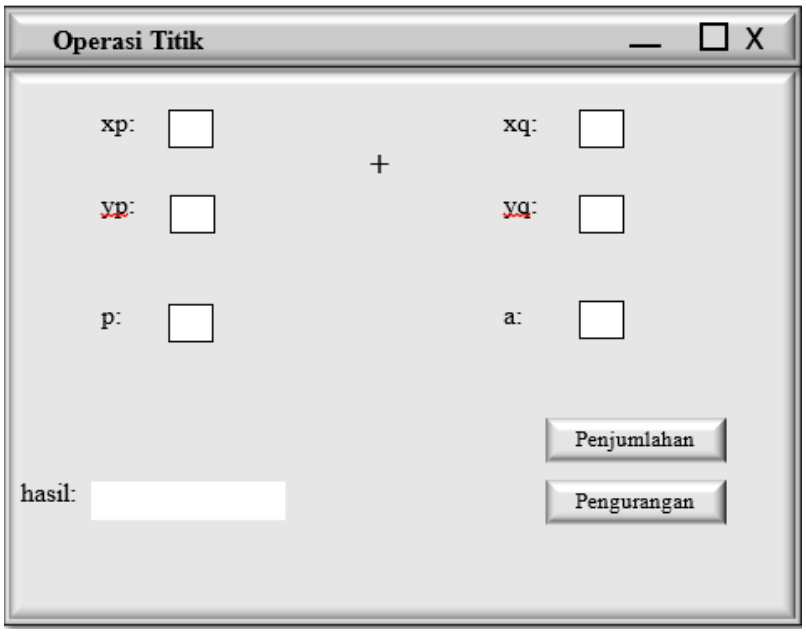
Gambar 3.6 Menu Utama

b. Pembangkitan kunci



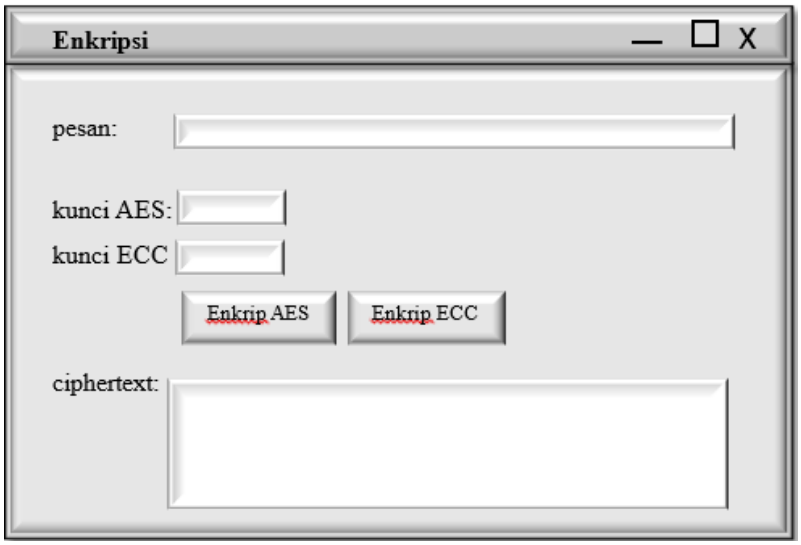
Gambar 3.7 Proses Pembangkitan Kunci

c. Operasi titik



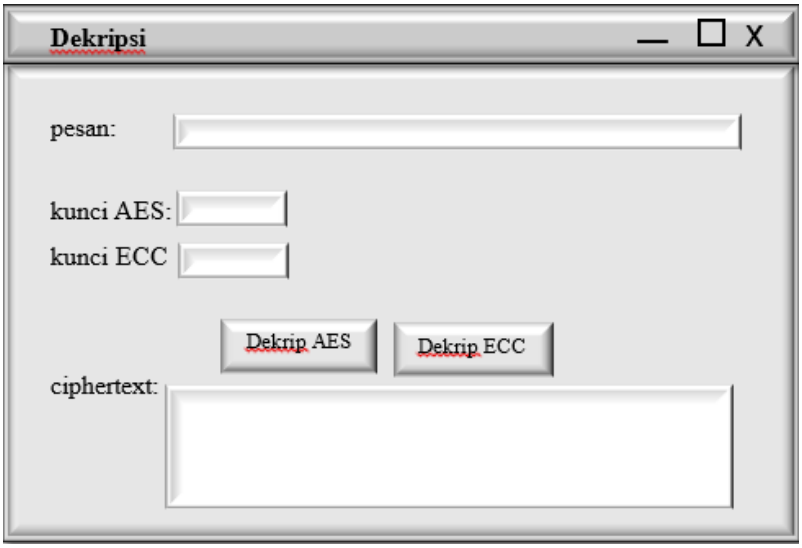
Gambar 3.8 Proses Operasi Titik

d. Enkripsi



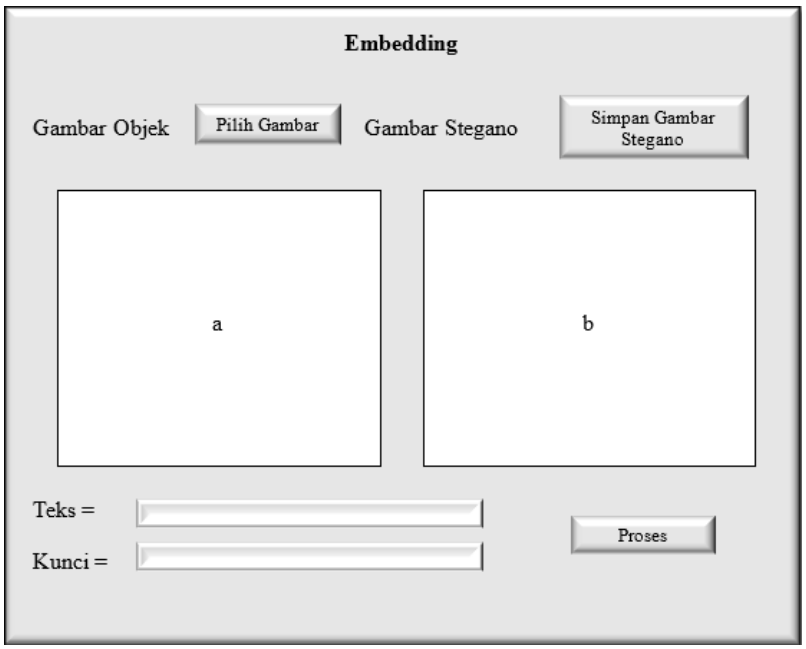
Gambar 3.9 Proses Eknripsi

e. Dekripsi



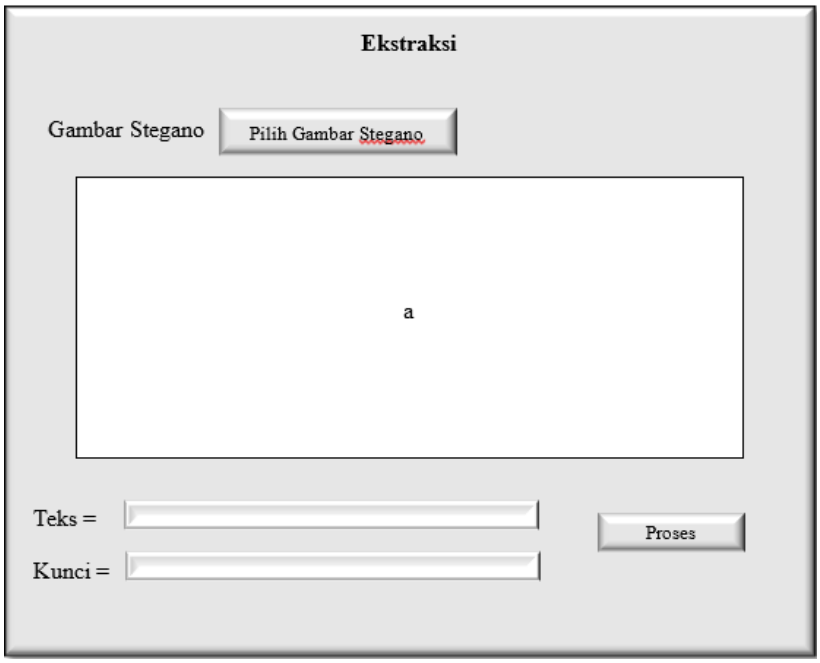
Gambar 3.10 Proses Dekripsi

f. *Embedding*



Gambar 3.11 Proses Embedding

g. Ekstraksi



Gambar 3.12 Proses Ekstraksi

3.4.4 Library Program

Dalam penggunaannya, *library Python* yang akan digunakan untuk menunjang pembuatan aplikasi adalah sebagai berikut:

1. *Tkinter*

   Tkinter adalah antarmuka grafis dari TCL (*Tool Command Language*), yang memfasilitasi pembuatan grafis program. *Tkinter* adalah *graphic user interface* (GUI) standar *Python* yang digunakan untuk membuat tampilan aplikasi dengan komponen modul *Tkinter* seperti tombol, kotak teks, label, dan lainnya.

2. *Pycryptodome*

   Pycryptodome adalah suatu *library Python* yang menyediakan berbagai fungsi kriptografi seperti AES, ECC, dan lainnya.

3. *Pillow*

   Pillow merupakan *library Python* yang fungsinya adalah untuk memanipulasi file gambar. *Pillow* diciptakan oleh Fredrik Lundh pada tahun 1955, dan pengembangannya dihentikan pada tahun 2011.

3.5 Proses Validasi

Proses validasi dilakukan untuk memastikan program aplikasi yang dirancang berjalan dengan tepat. Proses validasi dilakukan dengan cara melakukan percobaan pada program aplikasi dengan menggunakan beberapa contoh. Program aplikasi akan tervalidasi jika pada seluruh percobaan, cipherteks dan kunci AES yang disisipkan pada *stego-image* dapat dikembalikan menjadi plainteks.

3.6 Pengambilan Kesimpulan

Pada tahap ini akan ditarik kesimpulan dari pengembangan model yang dilakukan melalui hasil dari proses validasi, yaitu terkait penggabungan kriptografi *hybrid* AES & ECC dengan steganografi LSB beserta implementasinya ke dalam program aplikasi dengan menggunakan GUI *Python* yang dilakukan dalam beberapa percobaan.