BAB III
METODE PENELITIAN

Pada penelitian ini dilakukan studi literatur, pengembangan model, kemudian diimplementasikan ke dalam sebuah program aplikasi menggunakan GUI Python. Berikut merupakan langkah-langkah yang dilakukan dalam menyelesaikan penelitian:

3.1 Identifikasi Masalah

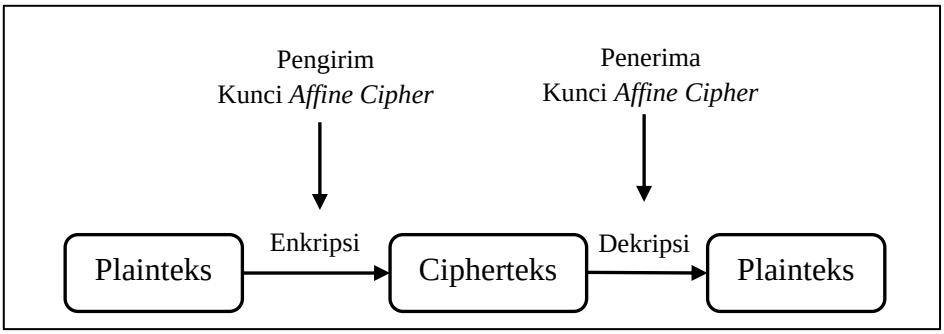
Pesan teks merupakan salah satu informasi yang perlu diamankan, terutama jika pesan tersebut bersifat rahasia. Pengamanan pesan dapat dilakukan dengan menyamarkan isi pesan tersebut. Kriptografi merupakan teknik yang digunakan untuk menyamarkan isi pesan sehingga tidak bisa dilihat oleh pihak yang tidak berwenang. Kemudian pesan dapat ditingkatkan keamanannya dengan penyisipan pada suatu media yaitu gambar. Dengan begitu orang lain tidak akan curiga jika sebenarnya terdapat pesan dibalik gambar tersebut. Steganografi merupakan metode yang digunakan untuk menyembunyikan keberadaan pesan. Penyamaran pesan dilakukan dengan menerapkan algoritma *Affine Cipher* sedangkan untuk penyisipan pesan dilakukan dengan menggunakan *Least Significant Bit-2*.

3.2 Model Dasar

Model dasar yang digunakan pada penelitian ini adalah algoritma kriptografi *Affine Cipher* dan steganografi *Least Significant Bit-2* (LSB-2).

3.2.1 Affine Cipher

Affine Cipher ini bersifat monoalfabetik, di mana setiap huruf dipetakan menjadi nilai numerik kemudian dienkripsi menggunakan fungsi matematika dan dikembalikan menjadi huruf. *Affine Cipher* merupakan algoritma kriptografi simetris yang berarti pada proses enkripsi dan dekripsi menggunakan kunci yang sama. Proses enkripsi dilakukan dengan menggunakan rumus $e_K(x) = (ax + b) mod\ n$ dan proses dekripsi dilakukan menggunakan rumus $d_K(y) = a^{-1}(y - b) mod\ n$.



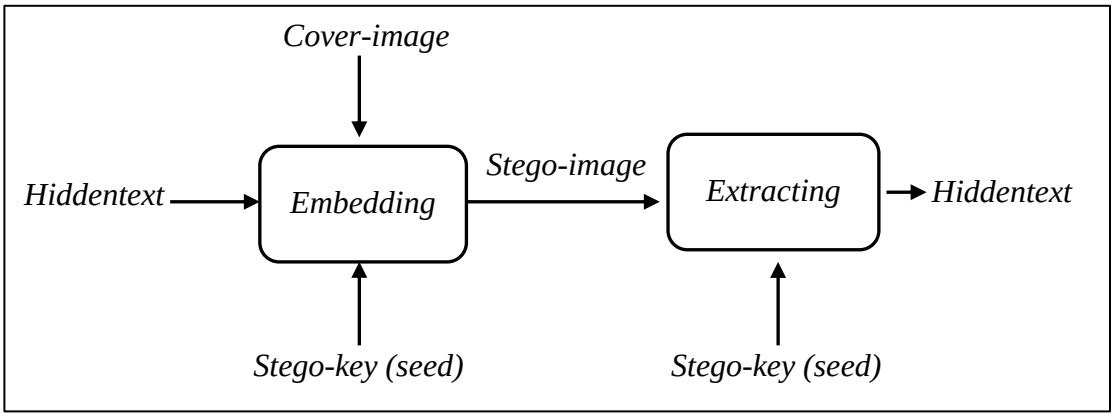
Gambar 3.1 Skema Kriptografi *Affine Cipher*

3.2.2 *Least Significant Bit-2*

Steganografi merupakan salah satu cara untuk meningkatkan keamanan pesan dengan menyembunyikan pesan ke suatu media. *Least Significant Bit* (LSB) merupakan salah satu algoritma steganografi dengan cara mengganti bit terakhir dengan pesan yang sudah diubah ke dalam bentuk biner. *Least Significant Bit-2* (LSB-2) merupakan modifikasi dari *Least Significant Bit* (LSB), di mana bit yang diganti merupakan bit ke-8 dikurangi dengan 2 (nilai LSB-2) atau sama dengan bit ke-6. Media yang dipakai dalam penelitian ini adalah file gambar dengan format .*png.

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-2 sehingga akan diperoleh *stego-image*. Tetapi untuk pemilihan lokasi *pixel* dipilih secara acak yang dapat dilakukan dengan metode *pseudorandom number generator* (PRNG) menggunakan *seed*.

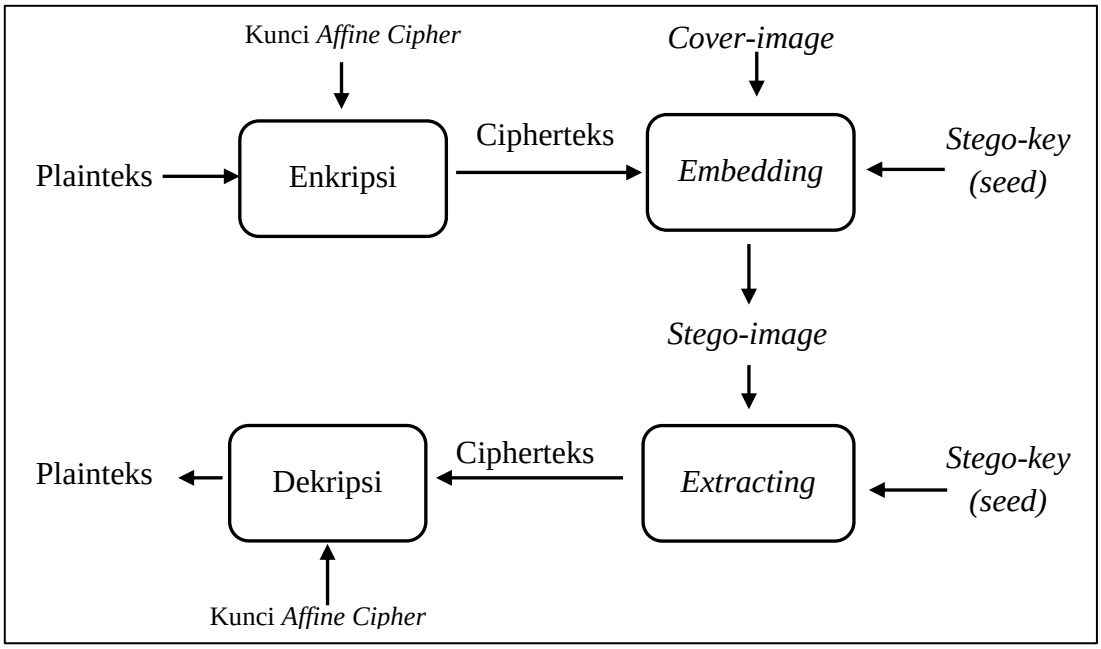
Tahap *extracting* dilakukan dengan cara membangkitkan PRNG menggunakan *seed* yang sama untuk mengetahui lokasi *pixe*l. Dengan mengambil setiap bit ke-6 dari setiap lokasi *pixel* pada *stego-image*. Kemudian bit-bit tersebut disusun dan dikelompokkan setiap 8 digit. Bit-bit yang sudah dikelompokkan diubah ke dalam huruf sehingga dipeoleh pesan berupa teks.



Gambar 3.2 Skema Steganografi *Least Significant Bit-2* (LSB-2)

3.3 Pengembangan Model

Pada penelitian ini akan menggabungkan kriptografi *Affine Cipher* dan steganografi *Least Significant Bit-2* (LSB-2) dengan tujuan untuk meningkatkan keamanan pesan. Langkah pertama yang dilakukan adalah plainteks (pesan asli) dienkripsi menggunakan algoritma *Affine Cipher* menjadi cipherteks (pesan acak) dengan kunci *Affine Cipher*. Kemudian dilakukan proses *embedding*, di mana cipherteks yang didapat akan disembunyikan pada gambar berformat .*png menggunakan metode LSB-2 sehingga menghasilkan *stego-image*. Namun untuk pemilihan lokasi *pixel* dipilih secara acak dengan *Pseudorandom Number Generator* (PRNG) menggunakan *seed*. Untuk proses *extracting*, langkah pertama membangkitkan PRNG menggunakan *seed* yang sama untuk mengetahui lokasi *pixe*l. Dengan mengambil setiap bit ke-6 dari setiap lokasi *pixel* pada *stego-image*. Bit-bit tersebut disusun dan dikelompokkan setiap 8 digit. Bit-bit yang sudah dikelompokkan diubah ke dalam huruf sehingga diperoleh pesan berupa cipherteks. Kemudian cipherteks yang diperoleh didekripsi menggunakan kunci *Affine Cipher* sehingga dapat menghasilkan pesan asli (plainteks) yang dikirim oleh pengirim pesan.



Gambar 3.3 Skema Penggabungan Kriptografi *Affine Cipher* dan Steganografi *Least Significant Bit-2* (LSB-2)

3.4 Kontruksi Program Aplikasi

Program penggabungan kriptografi *Affine Cipher* dan steganografi *Least Significant Bit-2* (LSB-2) ini akan dibuat menggunakan GUI Python. Program tersebut bertujuan untuk memudahkan proses enkripsi, dekripsi, *embedding* dan *extracting*.

3.4.1 *Input* dan *Output*

Pada tahap ini, dilakukan perancangan masukan (*input*) apa saja yang diperlukan beserta luaran (*output*) dari setiap program. *Input* dari program enkripsi dan *embedding* adalah kunci *Affine Cipher*, plainteks (pesan asli), gambar (*cover-image*) dan *stego-key* (*seed*) dengan *output* adalah *stego-image*. *Input* dari program dekripsi dan *extracting* adalah *stego-image*, *stego-key* (*seed*) dan kunci *Affine Cipher* dengan *output* adalah plainteks.

Tabel 3.1 Rancangan *Input* dan *Output* Program

Keterangan	Enkripsi dan *Embedding*	Dekripsi dan *Extracting*
Input	– Kunci *Affine Cipher* – Plainteks – *Cover Image* – *Stego-key* (*seed*)	– Kunci *Affine Cipher* – *Stego Image* – *Stego-key* (*seed*)
Output	– *Stego Image*	– Plainteks

3.4.2 Algoritma Penggabungan

Algoritma untuk melakukan proses penyisipan pesan rahasia pada gambar menggunakan *Affine Cipher* dan *Least Significant Bit-2* (LSB-2) pada aplikasi di penelitian ini diuraikan sebagai berikut:

a. Enkripsi dan *Embedding*

Berikut langkah-langkah enkripsi dan *embedding*:

1) Pengirim menentukan plainteks dan kunci *Affine Cipher*.
2) Pengirim melakukan enkripsi dengan memasukkan plainteks dan kunci yang sudah ditetapkan kemudian didapat cipherteks.
3) Cipherteks yang diperoleh akan disisipkan pada gambar.
4) Untuk melakukan proses *embedding*, pengirim menentukan *stego-key* (*seed*) dan gambar (*cover-image*) sebagai media penyembunyian pesan.
5) Gambar yang dihasilkan adalah *stego-image*.
6) Pengirim kemudian mengirimkan *stego-image*, kunci *Affine Cipher* dan *seed* pada penerima.

b. Dekripsi dan *Extracting*

Setelah menerima *stego-image*, kunci *Affine Cipher* dan *seed*, penerima kemudian melakukan *extracting* dan dekripsi untuk mengembalikan plainteks. Berikut langkah-langkah *extracting* dan dekripsi:

1) Penerima melakukan proses *extracting* dengan memasukkan *stego-image* dan *seed* yang didapat dari pengirim. Kemudian didapat cipherteks.
2) Cipherteks yang diperoleh akan didekripsi menggunakan kunci *Affine Cipher* yang didapat dari pengirim.
3) Kemudian dari proses dekripsi tersebut didapat plainteks.

3.4.3 *Library Coding*

Terdapat beberapa *library* Python yang digunakan untuk pembuatan program aplikasi, diantanya:

a. *Math*

Math adalah *library* Python yang digunakan untuk perhitungan ilmiah dan matematika yang kompleks. *Library* ini memiliki fungsi *math* seperti mengevaluasi operasi matematika biasa, operasi modulo, operasi trigonometri, operasi logaritma, dan lain-lain.

b. *NumPy*

NumPy (Numerical Python) adalah *library* Python yang fokus pada *scientific computing*. *NumPy* menyediakan fungsi yang siap pakai untuk memudahkan melakukan perhitungan *scientic* seperti matriks, aljabar, statistik dan lain-lain.

c. *Open CV*

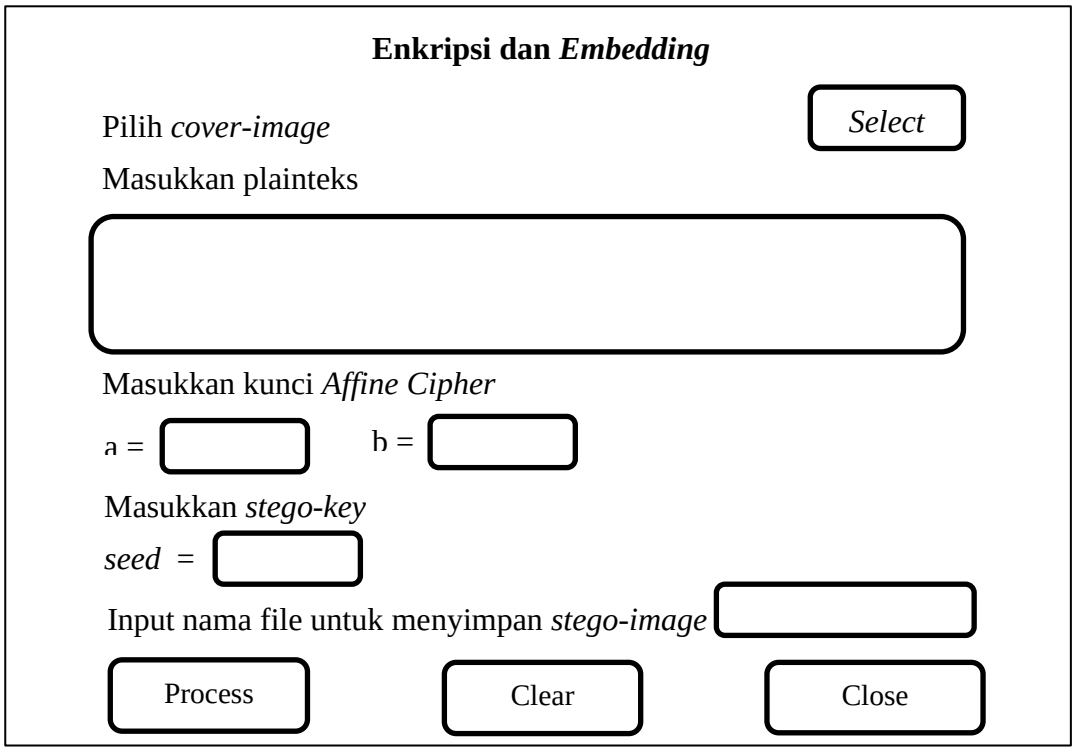
OpenCV (*Open Source Computer Vision Library*) adalah sebuah *library* Python yang digunakan untuk mengolah gambar. Pengolahan gambar untuk memperbaiki kualitas gambar atau untuk identifikasi gambar.

d. *Tkinter*

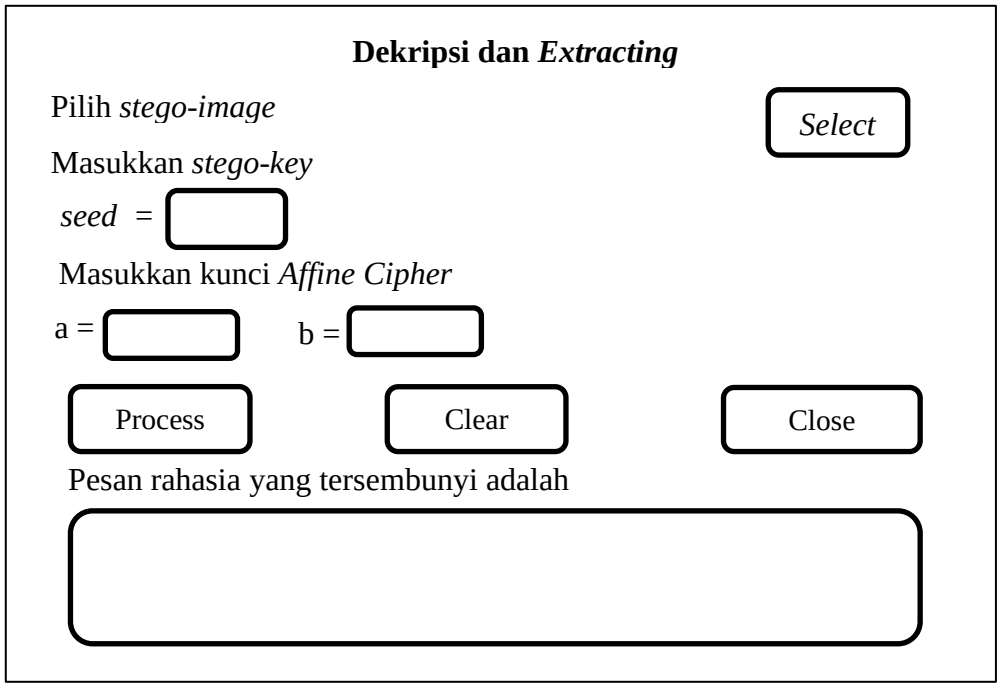
Tkinter adalah *graphic user interface* (GUI) standar Python digunakan untuk membuat tampilan aplikasi dengan komponen-komponen yang ada pada modul *tkinter* seperti *button*, *textbox*, *label*, *frame*, *window* yang mana sangat mendukung dalam penciptaan aplikasi GUI .

3.4.4 Rancangan Tampilan

Program aplikasi komputer dibuat untuk mempermudah enkripsi dan *embedding*, *extracting* dan dekripsi, serta validasi pada gabungan kriptografi dan steganografi. Program dibuat dengan bahasa pemograman Python. Rancangan program aplikasi komputer dapat dilihat pada Gambar 3.4.



Gambar 3.4 Rancangan Tampilan Program Aplikasi Enkripsi dan *Embedding*



Gambar 3.5 Rancangan Tampilan Program Aplikasi Dekripsi dan *Extracting*

3.5 Tahap Validasi

Pada tahap ini dilakukan validasi terhadap program aplikasi yang dirancang. Program aplikasi tervalidasi jika plainteks dapat diperoleh kembali pada proses *extracting* dan dekripsi serta melakukan pengujian *stego-image* yang diperoleh menggunakan *Peak Signal to Noise Ratio* (PSNR) untuk melihat kualitas *stego-image*. Untuk menghitung PSNR, perlu dilakukan perhitungan *Mean Squared Error* (MSE). MSE merupakan rata-rata dari selisih kuadrat eror antara *pixel cover-image* dan *stego-image*.

3.6 Penarikan Kesimpulan

Setelah program aplikasi tervalidasi, maka algoritma penggabungan ini dapat digunakan dalam penyandian pesan dan program yang telah dibuat dapat digunakan sebagai implementasi. Dengan penggabungan algoritma *Affine Cipher* dan *Least Significant Bit-2* (LSB-2) diharapkan dapat meningkatkan keamanan pesan sehingga penyadap kesulitan dalam memecahkan pesan rahasia.