BAB III
METODOLOGI PENELITIAN

Metode yang akan digunakan dalam penulisan penelitian ini adalah studi literatur, pengembangan model dan pengujian model ke dalam program *python*. Langkah-langkah penelitian akan diuraikan sebagai berikut:

3.1 Identifikasi Masalah

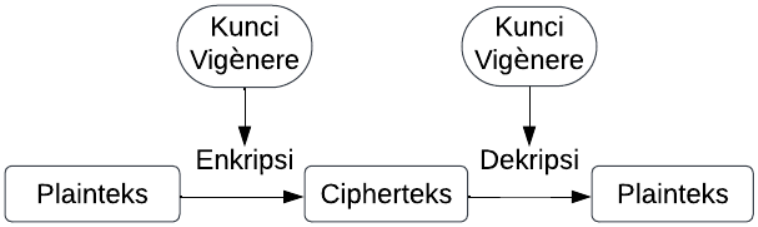
Penelitian ini menggabungkan teknik kriptografi Vigenère dan steganografi LSB dengan algoritma *Blum Blum Shub* untuk meningkatkan keamanan komunikasi data. Implementasi dilakukan dengan mengenkripsi pesan menggunakan algoritma Vigenère untuk menghasilkan cipherteks. Selanjutnya, cipherteks tersebut disisipkan ke dalam *cover-object* menggunakan steganografi LSB dengan algoritma *Blum Blum Shub*.

Program ini terdiri dari 4 fungsi utama, yaitu Enkripsi Vigenère, Dekripsi Vigenère, *embedding* dan *extraction*. Teknik kriptografi Vigenère digunakan untuk mengenkripsi pesan dengan kunci rahasia. Hasil enkripsi ini kemudian disisipkan ke dalam gambar menggunakan steganografi LSB dengan algoritma *Blum Blum Shub*. Penerima pesan dapat menggunakan kunci algoritma *Blum Blum Shub* untuk mengekstraksi cipherteks dari gambar terenkripsi. Selanjutnya, cipherteks tersebut didekripsi menggunakan kunci Vigenère yang tepat untuk mendapatkan kembali pesan asli.

3.2 Model Dasar

Model dasar yang akan digunakan pada penelitian adalah dengan mengenkripsi pesan menggunakan kriptografi Vigènere dan untuk menyisipkan sebuah pesan ke media penyisipan yaitu gambar menggunakan algoritma LSB. Vigènere adalah sebuah algoritma klasik yang digunakan untuk melakukan enkripsi dan dekripsi teks. Algoritma ini menggunakan pendekatan simetris, di mana hanya satu algoritma yang digunakan untuk kedua proses tersebut. Vigènere *cipher* sendiri merupakan algoritma kriptografi klasik dan termasuk *polyalphabetic cipher*, yaitu memungkinkan lebih dari satu cipherteks pada setiap huruf ketika di enkripsi. Tabel Vigenère square digunakan untuk enkripsi dan dekripsi.

Salah satu cara untuk mengembangkan Vigènere *cipher* adalah dengan menggunakan urutan tabel ASCII dari urutan 32 hingga 126. Rentang ini mencakup karakter-karakter yang umum digunakan dalam teks, seperti huruf besar dan kecil, angka, dan karakter khusus. Dengan menggunakan rentang ini, setiap karakter dalam teks dapat dienkripsi atau didekripsi dengan menggunakan subtitusi yang sesuai dalam rentang tersebut.

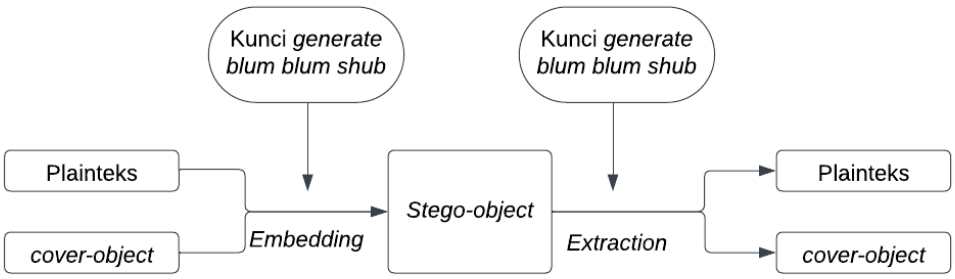


Gambar 3.1 Skema Vigènere

Skema pada Gambar 3.1 dapat dijelaskan sebagai berikut:

1. Pengirim menentukan plainteks yang akan dikirimkan kepada penerima. Kunci Vigènere yang akan digunakan dibangkitkan oleh penerima.
2. Pengirim melakukan enkripsi terhadap plainteks menggunakan kunci Vigènere dan menghasilkan cipherteks.
3. Pengirim mengirimkan cipherteks kepada penerima.
4. Penerima menggunakan kunci Vigènere yang sama untuk melakukan dekripsi terhadap cipherteks yang diterima dari pengirim
5. Setelah proses dekripsi, penerima memperoleh kembali plainteks yang asli.

Algoritma LSB (*Least Significant Bit*) adalah sebuah metode steganografi yang digunakan untuk menyembunyikan informasi rahasia atau pesan tersembunyi dalam citra atau media digital lainnya. Metode ini bekerja dengan memanfaatkan *bit* terakhir (*Least Significant Bit*) dari setiap piksel dalam citra sebagai tempat penyimpanan pesan tersembunyi. Algoritma LSB menggunakan kunci yang dihasilkan oleh generator bilangan acak *Blum Blum Shub*.



Gambar 3.2 Skema Algoritma LSB

Skema pada Gambar 3.2 dapat dijelaskan sebagai berikut:

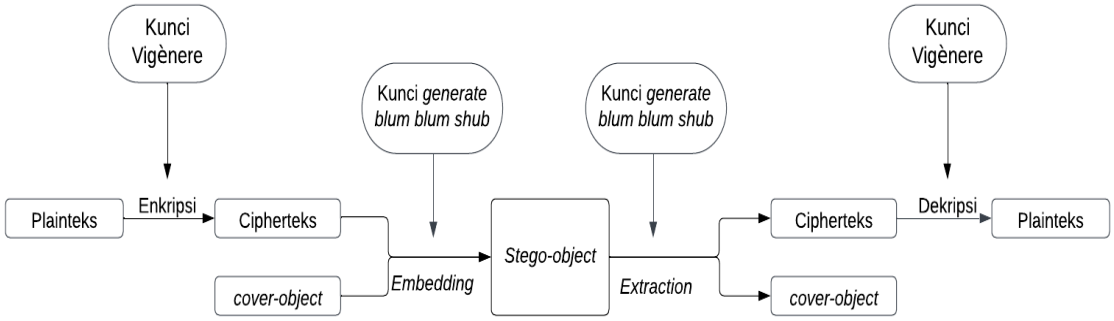
1. Pengirim menentukan plainteks yang akan disembunyikan dalam *cover-object*.
2. Pengirim juga menggunakan algoritma *Blum Blum Shub* untuk menghasilkan urutan bilangan acak yang akan digunakan dalam proses penyisipan.
3. Pengirim melakukan proses *embedding* dengan memanfaatkan algoritma *Blum Blum Shub*. Proses ini akan menghasilkan sebuah *stego-object*, yaitu gambar yang telah menyembunyikan pesan rahasia.
4. Penerima menerima *stego-object*
5. Penerima menggunakan algoritma *Blum Blum Shub* dengan parameter yang sama yang digunakan oleh pengirim untuk menghasilkan urutan bilangan acak yang sama.
6. Penerima mengekstrak *bit* pesan dari piksel gambar menggunakan algoritma *Blum Blum Shub*.
7. Setelah proses ekstraksi, penerima memperoleh kembali plainteks

3.3 Pengembangan Model Dasar

Dalam pengembangan model, dua teknik kriptografi, yaitu algoritma Vigènere *Cipher* dan steganografi *Least Significant Bit* (LSB), akan digabungkan dengan memanfaatkan *Blum Blum Shub* sebagai *pseudo random number generator* (PRNG). Pada tahap pengirim pesan, langkah pertama adalah mengenkripsi pesan awal atau plainteks menggunakan algoritma Vigènere *Cipher*. Setiap karakter pada plainteks akan diubah menjadi kode ASCII, sedangkan kunci yang dipilih akan

diulang secara berulang hingga memiliki panjang yang sama dengan plainteks. Setelah itu, setiap karakter plainteks akan digeser sejumlah nilai dari karakter kunci yang sesuai menggunakan rumus $C_i = ((P_i - 32) + (K_i - 32) \ mod \ 95) + 32$ Hasil dari proses ini akan menjadi cipherteks yang akan dikirimkan.

Selanjutnya, metode steganografi LSB akan digunakan untuk menyembunyikan cipherteks ke dalam gambar. Proses ini melibatkan pemilihan lokasi piksel secara acak menggunakan *pseudo random number generator* (PRNG) berbasis *Blum Blum Shub*. Kunci yang sama akan digunakan oleh pengirim dan penerima pesan untuk memastikan pemilihan lokasi piksel yang sama. *Bit-bit* cipherteks akan disisipkan ke dalam *bit* LSB dari lokasi piksel yang telah ditentukan secara acak. Pada tahap penerima pesan, penerima akan membangkitkan PRNG *Blum Blum Shub* dengan menggunakan Kunci yang sama dengan pengirim pesan. Hal ini memungkinkan penerima untuk mengetahui lokasi piksel yang digunakan untuk menyimpan *bit-bit* cipherteks. Penerima akan membaca *bit* LSB dari setiap lokasi piksel tersebut dan menyusunnya kembali menjadi cipherteks. Selanjutnya, cipherteks akan didekripsi menggunakan algoritma Vigènere *Cipher* dengan menggunakan kunci yang sama dengan pengirim untuk mengembalikan plainteks asli.



Gambar 3.3 Skema Model Pengembangan

3.4 Konstruksi Program aplikasi

Konstruksi program aplikasi menggunakan bahasa pemrograman *Python*

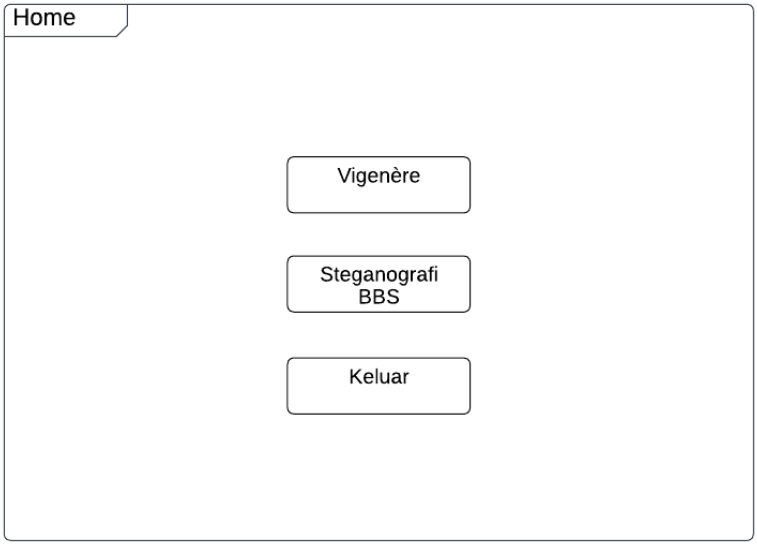
dengan rincian sebagai berikut:

3.4.1. *Input* dan *Output*

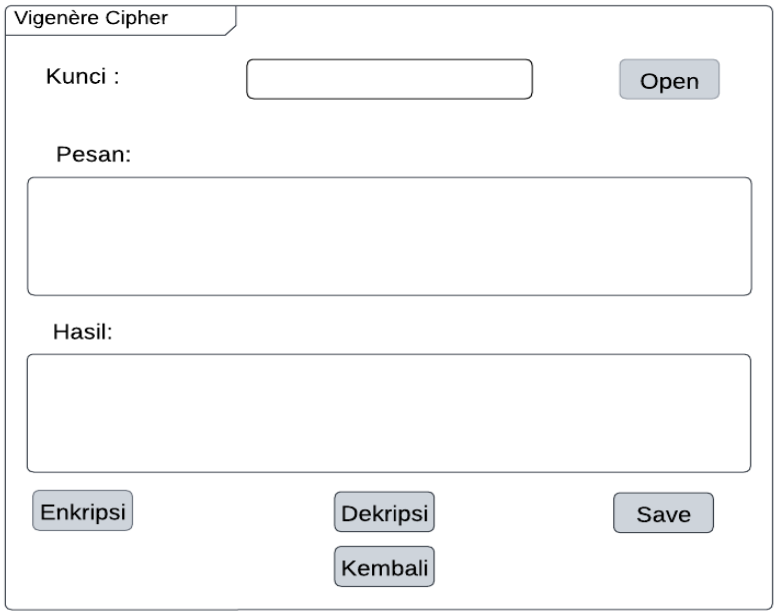
Untuk proses enkripsi pada program ini menggunakan algoritma Vigenère *Cipher* dengan input berupa *file* .txt dan kunci Vigenère *Cipher*. Setelah proses enkripsi, cipherteks akan disembunyikan dalam *cover-object* menggunakan teknik steganografi *Least Significant Bit* dengan bantuan algoritma *Blum Blum Shub* yang memerlukan *seed*, *p*, dan *q* sebagai parameter. Hasil akhir dari program adalah sebuah *stego-image* yang mengandung pesan terenkripsi yang telah tersembunyi secara rahasia dalam *cover-object*. Untuk melakukan ekstraksi, cipherteks yang tersembunyi dalam *stego-object* akan diekstrak menggunakan teknik steganografi LSB dengan algoritma *Blum Blum Shub* dan parameter *seed*, *p*, dan *q* yang sesuai. Setelah mendapatkan cipherteks, dilakukan dekripsi menggunakan Vigenère *Cipher* dengan menggunakan kunci yang sama seperti pada proses enkripsi. Hasil dekripsi ini adalah plainteks yang merupakan pesan awal sebelum dilakukan enkripsi dan penyembunyian dalam gambar.

3.4.2. Rancangan Tampilan Program Aplikasi

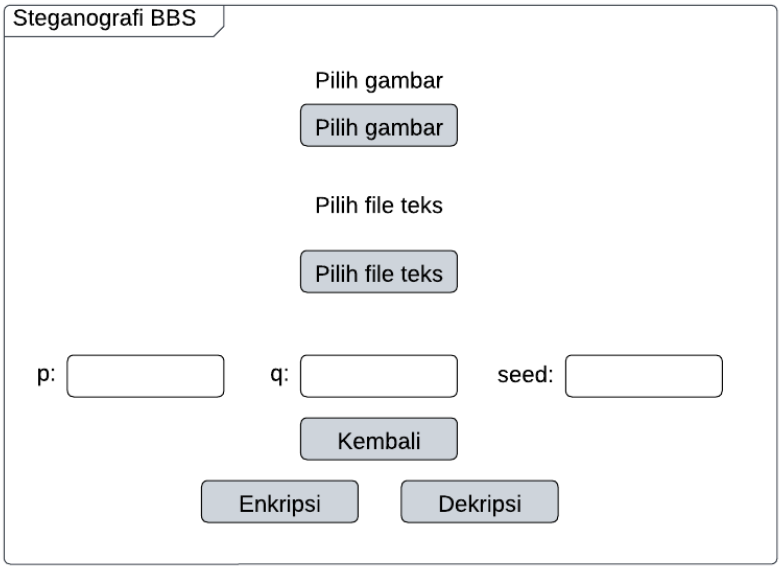
Rancangan tampilan utama program aplikasinya akan mempunyai 3 *Button*, yaitu *Button* Vigenère, *Button* Steganografi *Blum Blum Shub*, dan *Button* Keluar. Berikut merupakan rancangan tampilan program aplikasi yang akan dibuat:



Gambar 3. 4 Rancangan Tampilan Utama



Gambar 3. 5 Rancangan Tampilan Vigènere *Cipher*



Gambar 3. 6 Rancangan Tampilan Steganografi *Blum Blum Shub*

3.4.3. Algoritma Implementasi Steganografi Kombinasi *Least Significant Bit* dan *Blum Blum Shub* Dengan Kriptografi Vigenère *Cipher* Untuk Penyisipan Pesan Rahasia Dalam Gambar

Algoritma untuk melakukan steganografi kombinasi *least significant bit* dan *Blum Blum Shub* dengan kriptografi Vigenère *Cipher* untuk penyisipan pesan rahsia dalam gambar di penelitian ini diuraikan sebagai berikut:

Algoritma Enkripsi Vigenère

a. Pengguna memasukkan *file* teks (.txt) yang akan dienkripsi.
b. Pengguna memasukkan nilai kunci Vigenère.
c. Pengguna menentukan lokasi *file* hasil enkripsi akan disimpan.
d. Proses enkripsi Vigenère dilakukan pada isi teks dalam *file* menggunakan nilai kunci Vigenère yang telah diberikan oleh pengguna Dengan menggunakan formula sebagai berikut :

$$C_i = ((P_i - 32) + (K_i - 32) \ mod \ 95) + 32$$

e. Teks yang telah dienkripsi disimpan pada lokasi yang telah ditentukan oleh pengguna dalam format *file* teks (.txt).

Algoritma Dekripsi Vigenère

a. Penerima pesan memasukkan *file* teks (.txt) hasil enkripsi dan nilai kunci Vigenère yang sama dengan pengirim.
b. Pengguna memasukkan nilai kunci Vigenère yang digunakan saat proses enkripsi.
c. Proses dekripsi Vigenère dilakukan pada isi teks dalam *file* menggunakan nilai kunci Vigenère yang telah diberikan oleh penerima dengan menggunakan formula sebagai berikut :

$$P_i = ((C_i - 32) - (K_i - 32) \ mod \ 95) + 32$$

d. Hasil dekripsi berupa plainteks yang dapat dibaca dan dimengerti oleh penerima pesan.

Embedding Steganografi

a. Pengguna memilih gambar *cover-object* yang akan digunakan sebagai gambar stego-image untuk menyisipkan pesan tersembunyi.

b. Pengguna memilih *file* teks .txt yang berisi pesan yang akan disisipkan dalam gambar.
c. Pengguna memasukkan nilai *p, q,* dan *seed* yang diperlukan untuk algoritma *Blum Blum Shub*.
d. Program melakukan pengecekan apakah nilai *p* dan *q* adalah bilangan prima besar yang memenuhi kondisi $p \equiv q \equiv 3 \bmod 4$.
e. Program melakukan pengecekan apakah nilai *seed* dan *n* (*pq*) adalah bilangan relatif prima.
f. Program membaca data *cover-object* dan pesan teks yang akan disisipkan.
g. Program mengonversi pesan teks menjadi representasi biner dan menambahkan *bit* padding "00000000".
h. Program melakukan iterasi algoritma *Blum Blum Shub* untuk mendapatkan urutan indeks piksel dalam gambar menggunakan formula sebagai berikut:

$$X_{n+1} = X_n^2 \bmod n.$$

i. Program menyisipkan *bit* pesan ke dalam *bit* LSB (*Least Significant Bit*) pada komponen warna RGB pada setiap piksel gambar yang sesuai dengan urutan indeks piksel.
j. Setelah selesai penyisipan pesan, gambar hasil *embedding* (*stego-image*) disimpan dengan format "_encrypted.png" dan informasi data piksel yang diubah disimpan dalam *file* "encrypted_data.csv".
k. Program menampilkan pesan sukses bahwa proses penyisipan telah selesai dan *stego-image* tersimpan.

Extraction Steganografi

a. Pengguna memilih gambar *stego-object* yang telah disisipkan pesan tersembunyi.
b. Pengguna memasukkan nilai *p, q,* dan *seed* yang digunakan saat proses penyisipan.
c. Program melakukan pengecekan apakah nilai *p* dan *q* adalah bilangan prima besar yang memenuhi kondisi $p \equiv q \equiv 3 \bmod 4$.
d. Program melakukan pengecekan apakah nilai *seed* dan *n* (*pq*) adalah bilangan relatif prima.

e. Program membaca data piksel pada gambar *stego-object*.
f. Program melakukan iterasi algoritma *Blum Blum Shub* untuk mendapatkan urutan indeks piksel dalam gambar.
g. Program melakukan ekstraksi *bit* dari *bit* LSB (*Least Significant Bit*) pada komponen warna (RGB) pada setiap piksel gambar sesuai dengan urutan indeks piksel hingga menemukan *bit* padding "00000000".
h. Setelah *bit* pesan berhasil diekstraksi, program mengonversi *bit* pesan ke dalam bentuk teks dan menyimpan pesan tersembunyi dalam *file* "decrypted_message.txt".
i. Program menampilkan pesan sukses bahwa proses ekstraksi telah selesai dan pesan tersembunyi berhasil ditemukan.

3.4.4 *Library Python*

Library Python merupakan hasil gabungan dari berbagai paket dan modul yang memiliki fungsionalitas serupa, dengan tujuan untuk memudahkan pembuatan aplikasi bagi penggunanya. Berikut adalah beberapa *library Python* yang akan digunakan dalam program :

1. *Tkinter*

   Tkinter adalah *library* yang digunakan untuk membuat antarmuka grafis pada aplikasi *Python*. Dengan *Tkinter*, para pengembang dapat dengan mudah membuat jendela, tombol, kotak teks, dan elemen-elemen lainnya untuk berinteraksi dengan pengguna.

2. *Filedialog*

   Filedialog adalah *library* yang menyediakan dialog untuk memilih *file* dari sistem *file*. *Filedialog* digunakan untuk memberikan kemampuan bagi pengguna untuk memilih gambar dari sistem mereka yang akan digunakan dalam proses steganografi.

3. *Messagebox*

   Messagebox adalah *library* yang digunakan untuk menampilkan kotak dialog pesan kepada pengguna. *Messagebox* digunakan untuk memberikan informasi atau pesan konfirmasi kepada pengguna terkait hasil proses steganografi atau jika terjadi kesalahan dalam proses

tersebut. Dengan *Messagebox*, aplikasi dapat memberikan umpan balik yang lebih informatif kepada pengguna tentang status dan hasil dari operasi steganografi.

4. PIL (*Python Imaging Library*)

   PIL adalah *library* yang menyediakan berbagai fungsi untuk manipulasi gambar dalam *Python*. PIL digunakan untuk membuka gambar yang dipilih oleh pengguna dan untuk menyimpan gambar yang telah diubah setelah proses steganografi. PIL memungkinkan aplikasi untuk melakukan operasi seperti membuka, menyimpan, mengubah ukuran, dan mengubah format gambar.

5. *OS*

   OS adalah *library* yang menyediakan berbagai fungsi untuk berinteraksi dengan sistem operasi. Dalam kode tersebut, *OS* digunakan untuk mengatur atau memanipulasi *path file* gambar yang digunakan dalam proses steganografi. Dengan menggunakan *OS*, aplikasi dapat dengan mudah berinteraksi dengan sistem *file*

6. *Tabulate*

   Tabulate adalah *library* yang digunakan untuk menyajikan data dalam bentuk tabel. *Tabulate* digunakan untuk menampilkan hasil perbandingan atau informasi lain dalam bentuk tabel, yang memudahkan pengguna untuk memahami dan membandingkan data dengan lebih terstruktur dan rapi.

7. *Subprocess*

   Subprocess adalah *library* dalam *Python* yang memungkinkan kita untuk menjalankan perintah *shell* atau perintah dari sistem operasi yang terpisah dari *Python* itu sendiri. Menggunakan *subprocess*, dapat berkomunikasi dengan sistem operasi secara langsung melalui kode *Python*. *Library* ini menyediakan berbagai fungsi dan metode yang memungkinkan kita untuk menjalankan perintah-perintah seperti menjalankan program eksternal, mengambil output dari perintah,

mengalihkan input atau output dari perintah, mengatur variabel lingkungan, dan lainnya.

8. *Math.GCD*

   Math.GCD adalah fungsi dari library math yang digunakan untuk mendapatkan nilai GCD (*Greatest Common Divisor*) atau faktor persekutuan terbesar antara dua angka. Fungsi ini sangat berguna dalam pengembangan algoritma yang memerlukan angka-angka acak dengan sifat tertentu.

3.5 Proses Validasi

Pada tahap ini dilakukan validasi terhadap program aplikasi yang dirancang. Validasi dilakukan dengan memberikan contoh pada program aplikasi yang selanjutnya akan dicocokkan dengan proses perhitungan manual. Program aplikasi tervalidasi jika cipherteks yang sudah disisipkan pada *stego-image* dapat dikembalikan menjadi plainteks dan juga hasil enkripsi, dekripsi, *embedding*, dan ekstraksi pada program aplikasi sama dengan perhitungan manual.

3.6 Pengambilan Kesimpulan

Pada tahap akhir, dilakukan penarikan kesimpulan dari hasil penelitian yang telah dilakukan dan memberikan rekomendasi-rekomendasi untuk peneliti selanjutnya agar mendapatkan hasil penelitian yang lebih baik.