BAB III
METODE PENELITIAN

3.1 Identifikasi Masalah

Masalah keamanan dan kerahasiaan suatu data khususnya data pribadi menjadi salah satu aspek penting dalam perkembangan teknologi komunikasi. Kriptografi dan steganografi diperlukan untuk menjaga keamanan dan kerahasiaan data tersebut. RSA yang ditingkatkan dan *Spread Spectrum* merupakan algoritma yang dapat digunakan untuk mengamankan data tersebut.

Algoritma RSA yang ditingkatkan mengenkripsi setiap karakter pada sebuah *plaintext* dengan tipe data teks menggunakan *public key* berupa pasangan kunci (*e, n*) yang telah dibangkitkan sehingga menghasilkan blok-blok yang merepresentasikan nilai hasil enkripsi dari setiap *plaintext* tersebut. Setiap blok-blok tersebut disatukan dengan menggunakan karakter spasi sebagai pemisah antar blok sehingga menghasilkan sebuah *ciphertext* berupa gabungan dari karakter spasi dengan blok-blok *cipher* hasil enkripsi. Algoritma RSA yang ditingkatkan mendekripsi *ciphertext* dengan memisahkan *ciphertext* tersebut menjadi blok-blok *cipher* dan mendekripsi blok *cipher* tersebut menggunakan *private key* berupa pasangan kunci (*d, n*) yang telah dibangkitkan sehingga *ciphertext* tersebut dapat dikembalikan menjadi *plaintext*.

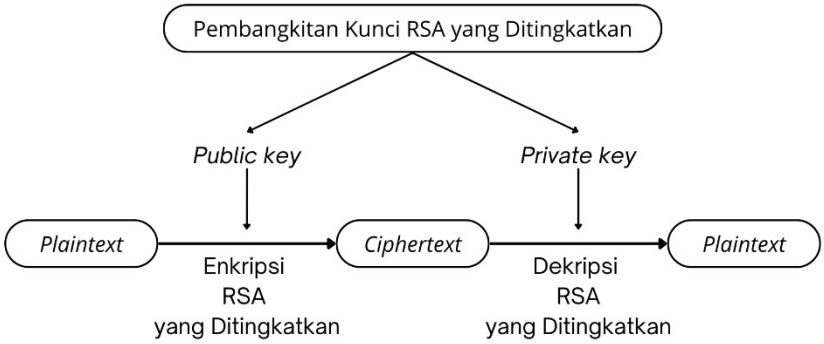
Algoritma *Spread Spectrum* melakukan *embedding* sebuah *embedded message* dengan tipe data teks pada *cover-image* dengan tipe citra PNG menggunakan *stego-key* dengan tipe data teks sehingga menghasilkan *stego-image*. *Ciphertext* yang dihasilkan oleh proses enkripsi algoritma RSA yang ditingkatkan dapat dijadikan *embedded message* pada proses *embedding* algoritma *Spread Spectrum*. Algoritma *Spread Spectrum* melakukan proses *extracting stego-image* menggunakan *stego-key* sehingga proses *extracting* tersebut dapat mengembalikan *embedded message* yang telah disembunyikan.

3.2 Model Dasar

Algoritma kriptografi RSA yang ditingkatkan dan algoritma steganografi *Spread Spectrum* merupakan model dasar yang digunakan dalam penelitian ini.

3.2.1 Model RSA yang Ditingkatkan

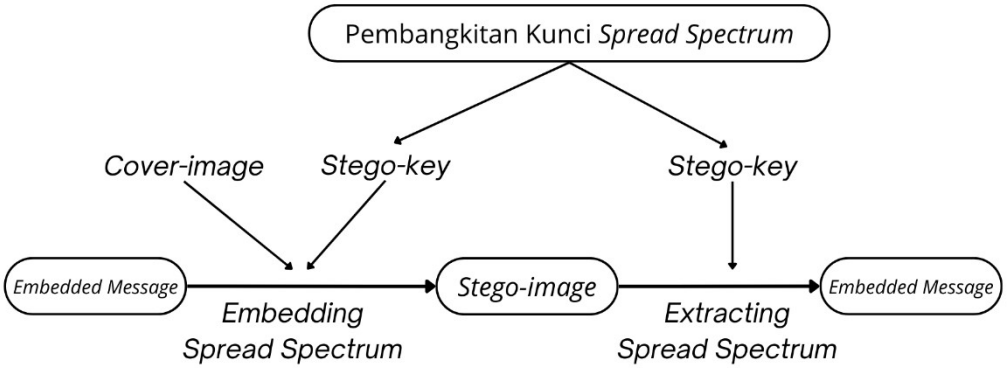
Algoritma RSA merupakan salah satu algoritma kriptografi asimetris yang digunakan untuk mengamankan data. Peningkatan pada algoritma RSA dapat dilakukan, salah satunya dengan cara menambahkan satu bilangan prima pada proses pembangkitan kunci. Algoritma RSA menggunakan perkalian antara bilangan prima pada proses pembangkitan kunci. Pembangkitan kunci tersebut menghasilkan *private-key* yang digunakan pada proses dekripsi dan *public-key* yang digunakan pada proses enkripsi.



Gambar 3.1 Skema Algoritma RSA yang Ditingkatkan

3.2.2 Model *Spread Spectrum*

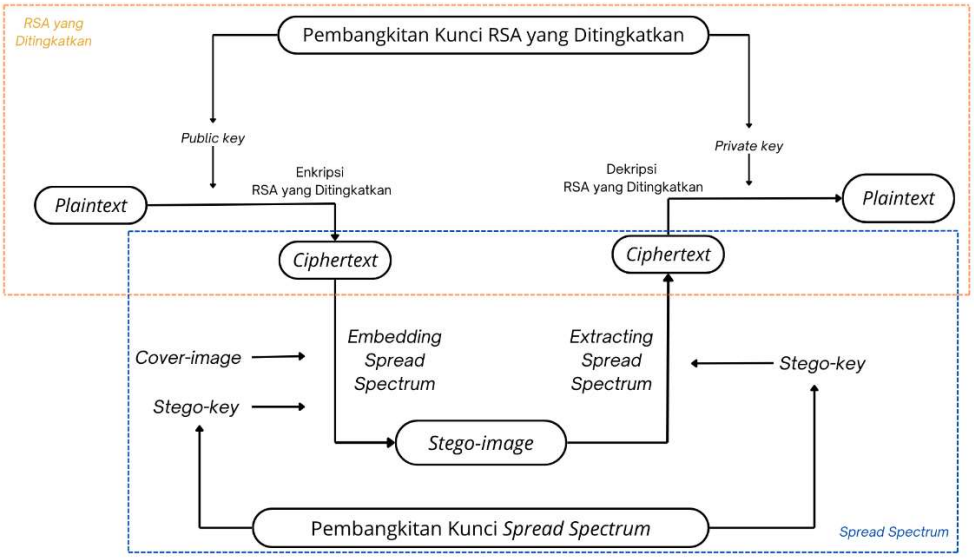
Algoritma *Spread Spectrum* merupakan salah satu algoritma steganografi yang digunakan untuk menyembunyikan data. Algoritma *Spread Spectrum* menyebarkan data pada suatu objek melalui perhitungan modulasi. Salah satu objek yang digunakan sebagai objek penyisipan adalah citra digital. Data dengan tipe data teks merupakan salah satu data yang dapat disembunyikan pada citra digital. Data disebar dan dilakukan perhitungan modulasi dengan *pseudonoise*, yaitu bilangan acak yang telah dibangkitkan dengan metode LCG. *Stego-key* atau kunci steganografi dan tiga bilangan bulat diperlukan untuk membangkitkan *pseudonoise* yang akan digunakan pada proses *embedding* dan *extracting* citra digital.



Gambar 3.2 Skema Algoritma *Spread Spectrum*

3.3 Pengembangan Model Dasar

Pengembangan model pada penelitian ini adalah dengan menggabungkan algoritma pada model dasar, yaitu algoritma kriptografi RSA yang ditingkatkan dan algoritma steganografi *Spread Spectrum* sehingga data yang dikirimkan akan lebih terjamin keamanan dan kerahasiaannya. Pengirim pesan mengenkripsi data dalam bentuk teks yang akan dikirimkan menggunakan algoritma RSA yang ditingkatkan dengan *public-key* yang telah dibangkitkan oleh penerima pesan sehingga menghasilkan suatu *ciphertext* yang akan disembunyikan pada sebuah citra digital. Pengirim pesan melakukan proses *embedding ciphertext* pada citra digital menggunakan algoritma *Spread Spectrum* sehingga menghasilkan *stego-image*. Penerima pesan melakukan proses *extracting* pada citra digital (*stego-image*) yang diterima dari pengirim pesan menggunakan algoritma *Spread Spectrum* dengan kunci steganografi yang dikirimkan oleh pengirim pesan. Penerima pesan memperoleh teks yang disisipkan (*embedded message*) berupa *ciphertext* dan melakukan proses dekripsi *ciphertext* tersebut menggunakan algoritma RSA yang ditingkatkan dengan *private-key* yang telah dibangkitkan oleh penerima pesan.



Gambar 3.3 Skema Pengembangan Model

3.4 Konstruksi Program Aplikasi

Program aplikasi pada penelitian ini menggunakan bahasa pemrograman Python dengan rincian konstruksi sebagai berikut.

3.4.1 *Input* dan *Output*

Program aplikasi pada penelitian ini menggunakan bahasa pemrograman Python dengan *input* dan *output* dari aplikasi adalah sebagai berikut.

1. *Input* dan *output* Pembangkitan Kunci

Tabel 3.1 Input dan Output Pembangkitan Kunci RSA yang Ditingkatkan dan *Spread Spectrum*

Keterangan	RSA yang Ditingkatkan	*Spread Spectrum*
Input	• Nilai p_1 • Nilai p_2 • Nilai p_3 • Nilai e	• Nilai a • Nilai c • Nilai m
Output	• Nilai $\phi(n)$ • Nilai n • *Public key* (e, n) • *Private key* (d, n)	-

2. *Input* dan *output* Enkripsi dan Dekripsi RSA yang ditingkatkan

Tabel 3.2 Input dan Output Enkripsi dan Dekripsi RSA yang Ditingkatkan

Keterangan	Enkripsi	Dekripsi
Input	• *Plaintext* • *Public key*	• *Ciphertext* • *Private key*
Output	*Ciphertext*	*Plaintext*

3. Input dan output *embedding* dan *extracting Spread Spectrum*

Tabel 3.3 Input dan Output *Embedding* dan *Extracting Spread Spectrum*

Keterangan	*Embedding*	*Extracting*
Input	• *Ciphertext* • *Cover-image* • *Stego-key*	• *Stego-image* • *Stego-key*
Output	*Stego-image*	*Ciphertext*

4. Input dan output pengujian citra digital

Tabel 3.4 Input dan Output Pengujian Citra Digital

Keterangan	Uji PSNR
Input	• *Cover-image* • *Stego-image*
Output	*PSNR value*

3.4.2 Algoritma Deskriptif

Program aplikasi yang akan dikembangkan menggunakan kombinasi antara algoritma kriptografi RSA yang ditingkatkan dan algoritma steganografi *Spread Spectrum*, serta pengujian citra menggunakan pengujian PSNR. Algoritma deskriptif program aplikasi berdasarkan masing-masing algoritma yang digunakan pada penelitian ini adalah sebagai berikut.

A. Algoritma Kriptografi RSA yang Ditingkatkan

Algoritma RSA memiliki tahapan pembangkitan kunci, enkripsi, serta dekripsi dalam mengamankan suatu *plaintext*. Algoritma deskriptif pada setiap tahapan tersebut adalah sebagai berikut.

a. Pembangkitan kunci
    1. Masukkan tiga bilangan prima $p_1, p_2,$ dan p_3.
    2. Tekan tombol proses yang terdapat di bawah *input* nilai p_1 pada program aplikasi.
    3. Diperoleh nilai n dan $\phi(n)$.
    4. Masukkan nilai e yang relatif prima dengan $\phi(n)$.
    5. Tekan tombol proses yang terdapat di bawah *input* nilai e pada program aplikasi.
    6. Diperoleh nilai d, *public key* (e, n), dan *private key* (d, n).

b. Enkripsi
    1. Masukkan *plaintext* dan *public key* (e, n).
    2. Tekan tombol proses yang terdapat pada program aplikasi.
    3. Diperoleh *ciphertext* yang merupakan gabungan blok-blok *cipher* hasil enkripsi.

c. Dekripsi
    1. Masukkan *ciphertext* dan *private key* (d, n).
    2. Tekan tombol proses yang terdapat pada program aplikasi.
    3. Diperoleh *plaintext*.

B. Algoritma Steganografi *Spread Spectrum*

Algoritma *Spread Spectrum* memiliki tahapan pembangkitan kunci, *embedding*, serta *extracting* dalam menyembunyikan suatu *embedded message*. Proses pembangkitan kunci algoritma *Spread Spectrum* pada program aplikasi yang akan dikembangkan merupakan proses untuk menyimpan nilai $a, c,$ $serta$ m yang akan digunakan pada proses pembangkitan *pseudonoise* menggunakan metode LCG. Setelah memperoleh *stego-image* hasil dari proses *embedding*, *cover-image* dapat dibandingkan dengan *stego-image* menggunakan pengujian nilai PSNR. Algoritma deskriptif pada setiap tahapan tersebut adalah sebagai berikut.

a. Pembangkitan kunci
    1. Masukkan nilai $a, c,$ dan m.
    2. Tekan tombol bangkitkan kunci yang terdapat pada program aplikasi.

b. *Embedding*
    1. Pilih *cover-image* yang akan digunakan untuk menyembunyikan *embedded message.*
    2. Masukkan *embedded message* dan *stego-key.*
    3. Tekan tombol proses yang terdapat pada program aplikasi.
    4. Diperoleh *stego-image*.

c. *Extracting*
    1. Pilih *stego-image.*
    2. Masukkan *stego-key.*
    3. Tekan tombol proses yang terdapat pada program aplikasi.
    4. Diperoleh *embedded message*.

C. Pengujian Citra Digital

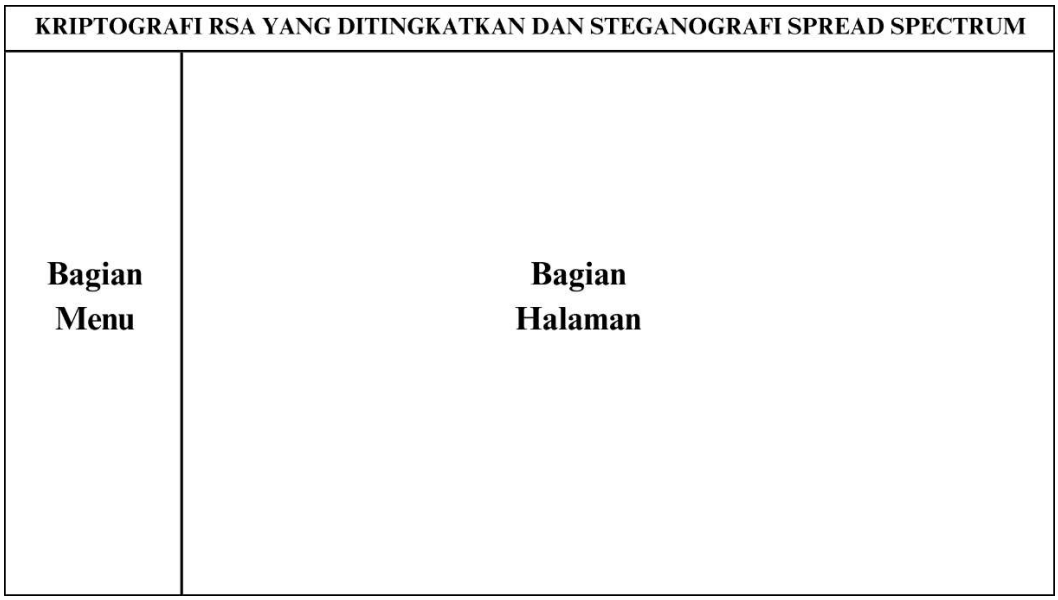
Pengujian citra digital dilakukan untuk mengetahui kualitas sebuah citra. PSNR merupakan salah satu metode yang dilakukan untuk mengetahui kualitas sebuah citra digital. Pengujian PSNR dilakukan dengan membandingkan citra asli dan citra yang telah diubah. Semakin tinggi nilai PSNR suatu pengujian citra, maka kualitas citra tersebut akan semakin tinggi. Suatu citra memiliki kualitas yang baik jika nilai PSNR hasil pengujian citra tersebut lebih besar dari 30dB. Algoritma deskriptif pada pengujian citra digital tersebut adalah sebagai berikut.

1. Pilih *cover-image* dan *stego-image.*
2. Tekan tombol proses yang terdapat pada program aplikasi.
3. Diperoleh nilai hasil pengujian PSNR.

3.4.3 Rancangan Tampilan Program Aplikasi

Tampilan program aplikasi pada penelitian ini menggunakan GUI (*Graphical User Interface*) dalam bahasa pemrograman Python. Rancangan tampilan pada program aplikasi ini memiliki dua bagian utama yang akan ditampilkan, yaitu bagian halaman (*page*) yang digunakan untuk menampilkan halaman pembangkitan kunci, enkripsi, *embedding*, *extracting*, atau dekripsi dan

bagian menu yang berfungsi untuk mengganti setiap halaman yang akan ditampilkan. Rancangan tampilan utama pada program aplikasi yang akan dikonstruksi dapat dilihat pada Gambar 3.4.



Gambar 3.4 Rancangan Tampilan Program Utama

Terdapat rancangan tampilan pada bagian menu yang dapat dilihat pada Gambar 3.5.

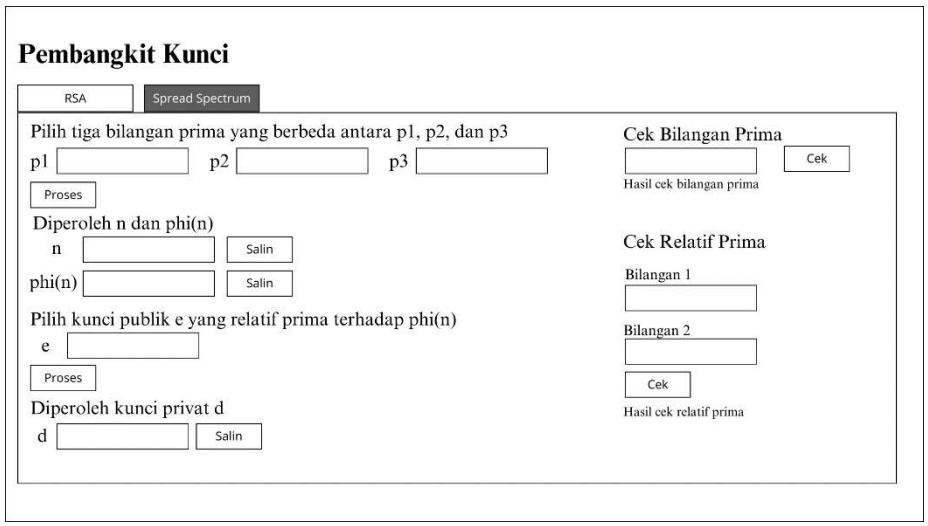
Gambar 3.5 Rancangan Tampilan Bagian Menu

Terdapat lima halaman pada program aplikasi ini, rancangan tampilan pada setiap halaman tersebut adalah sebagai berikut:

1. Halaman Pembangkitan Kunci

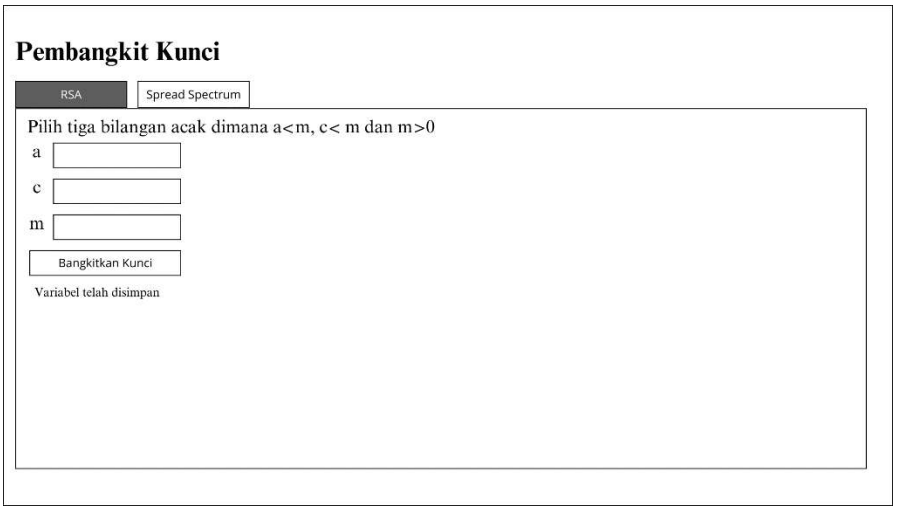
   Terdapat bagian tambahan pada halaman pembangkitan kunci, yaitu bagian untuk pembangkitan kunci RSA dan *Spread Spectrum* dengan rancangan tampilan konstruksi yang dapat dilihat pada Gambar 3.6 dan Gambar 3.7.

   a. RSA



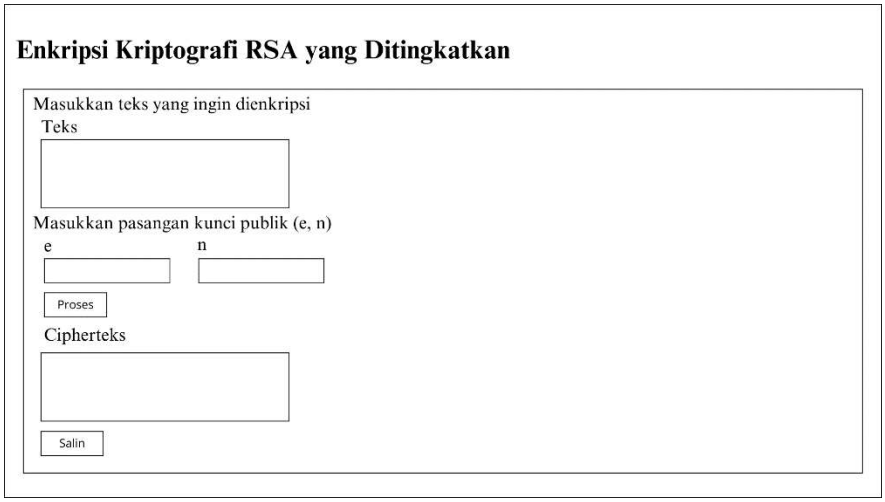
Gambar 3.6 Rancangan Tampilan Halaman Pembangkit Kunci RSA

   b. *Spread Spectrum*



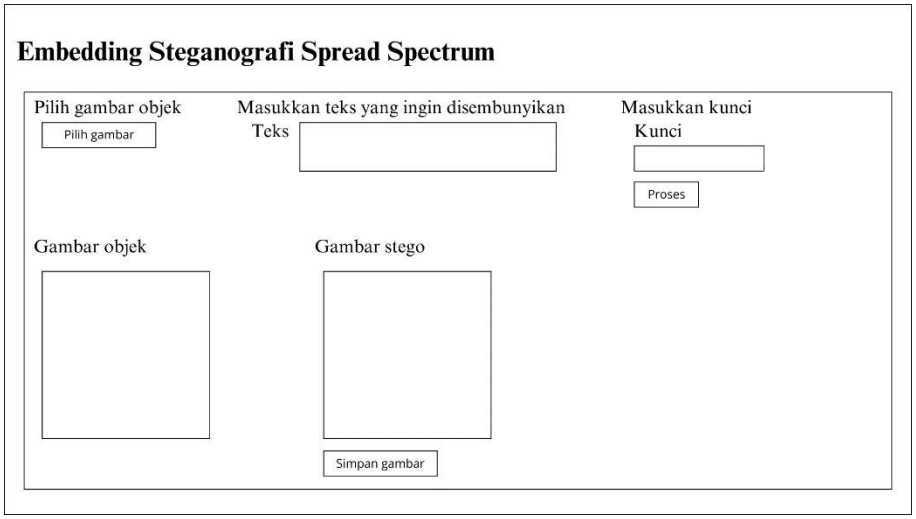
Gambar 3.7 Rancangan Tampilan Halaman Pembangkit Kunci *Spread Spectrum*

2. Halaman Enkripsi



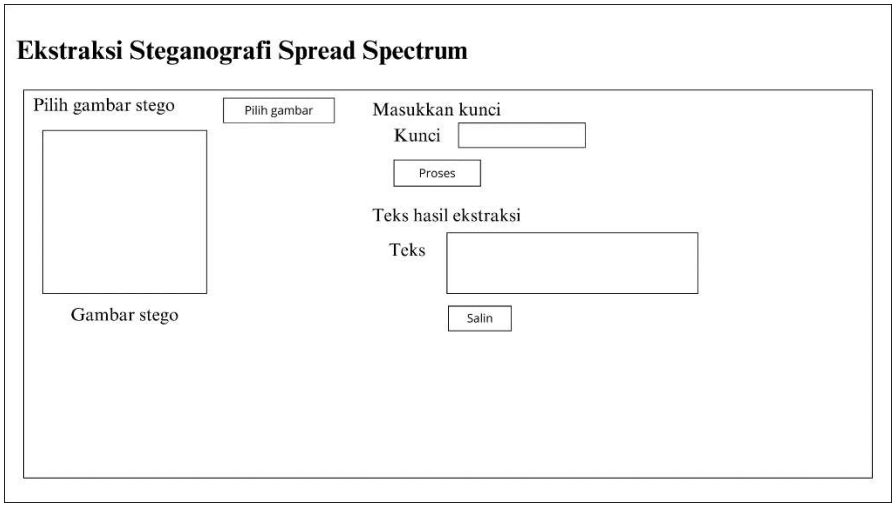
Gambar 3.8 Rancangan Tampilan Halaman Enkripsi

3. Halaman *Embedding*



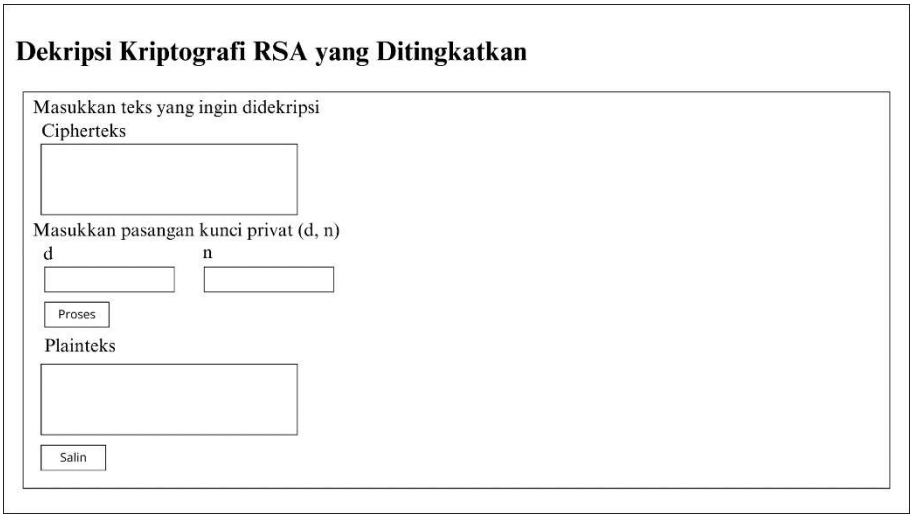
Gambar 3.9 Rancangan Tampilan Halaman *Embedding*

4. Halaman *Extracting*



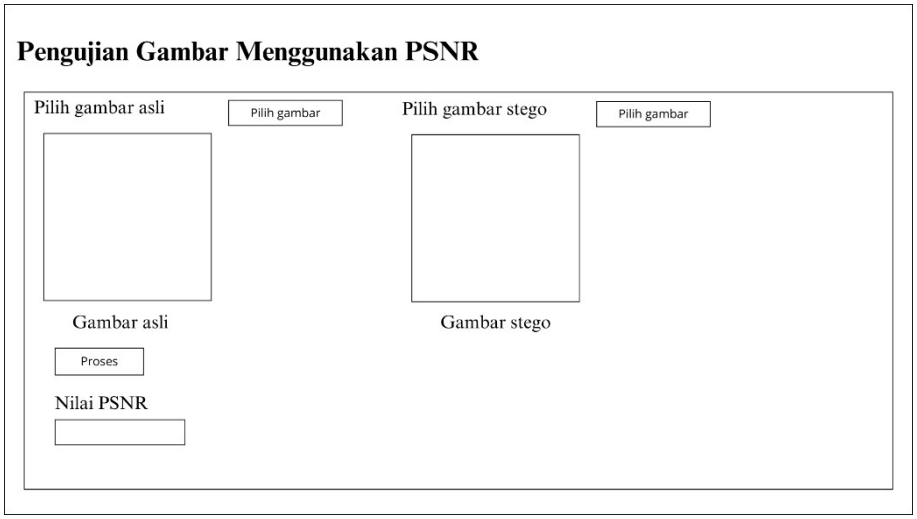
Gambar 3.10 Rancangan Tampilan Halaman *Extracting*

5. Halaman Dekripsi



Gambar 3.11 Rancangan Tampilan Halaman Dekripsi

6. Halaman Pengujian Gambar



Gambar 3.12 Rancangan Tampilan Halaman Penguji Gambar

3.4.4 *Library* dan *Module* Program Aplikasi

Program aplikasi pada penelitian ini menggunakan beberapa *library* dan *module* yang terdapat pada bahasa pemrograman Python. *Library* yang digunakan program aplikasi pada penelitian ini adalah sebagai berikut:

1. Tkinter, merupakan *library* pada Python yang digunakan untuk menampilkan GUI pada program aplikasi.
2. Pillow (PIL), merupakan *library* pada Python yang digunakan untuk mengolah citra digital, seperti menampilkan atau memanipulasi citra digital.

Module yang digunakan program aplikasi pada penelitian ini adalah sebagai berikut:

1. *Messagebox*, merupakan *module* pada *library* Tkinter yang digunakan untuk menampilkan kotak dialog yang berisikan informasi
2. *Filedialog*, merupakan *module* pada *library* Tkinter yang digunakan untuk memilih atau menyimpan suatu *file* seperti citra digital.
3. *Image*, merupakan *module* pada *library* PIL yang digunakan untuk mengolah citra digital secara sederhana.
4. *ImageTk*, merupakan *module* pada *library* PIL yang digunakan untuk mengolah citra digital yang lebih rumit.

5. *Math*, merupakan *module* yang sudah terinstal bersama dengan Python yang digunakan untuk perhitungan ilmiah dan matematika.

3.5 Validasi

Pada tahap ini dilakukan validasi terhadap program aplikasi yang dirancang. Validasi dilakukan untuk mengetahui jika program yang dirancang dapat menghasilkan *ciphertext* dan menyisipkannya pada citra digital serta mengembalikan *ciphertext* yang diperoleh dari *extracting* citra digital tersebut menjadi *plaintext* berdasarkan algoritma RSA yang ditingkatkan dan algoritma *Spread Spectrum*. *Stego-image* yang dihasilkan dari proses *embedding* akan dilakukan pengujian kualitas citra menggunakan PSNR dengan membandingkan *stego-image* dengan *cover-image*. Validasi tersebut dilakukan dengan memperlihatkan kesamaan antara *plaintext, stego-image,* serta *ciphertext* yang dihasilkan oleh program aplikasi sesuai dengan proses perhitungan manual.

3.6 Penarikan Kesimpulan

Pada tahap ini akan dilakukan penarikan kesimpulan berdasarkan pengembangan model dan hasil penelitian yang dilakukan, serta memberikan saran-saran untuk peneliti selanjutnya untuk mendapatkan hasil penelitian yang lebih baik. Program aplikasi tervalidasi jika algoritma yang digunakan dapat mengamankan, menyembunyikan, dan mengembalikan pesan rahasia yang akan dikirimkan.