BAB III
METODE PENELITIAN

3.1 Identifikasi Masalah

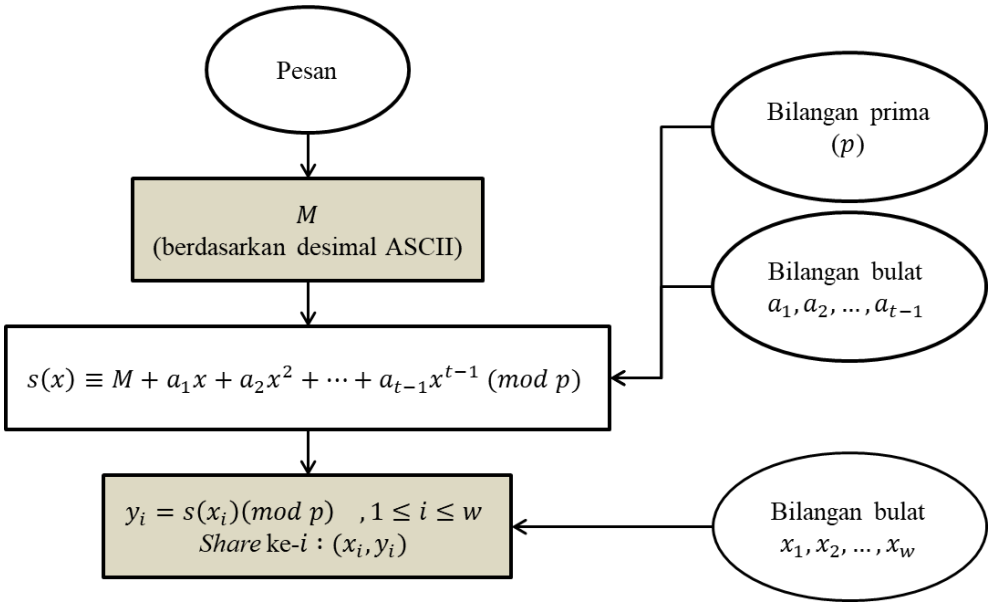
Penggabungan kriptografi *Shamir Secret Sharing* dan steganografi *Least Significant Bit* dengan *QR Code* pada penelitian ini bertujuan untuk meningkatkan salah satu aspek keamanan pesan yaitu kerahasiaan. Pesan rahasia dalam penelitian ini yaitu pesan teks (maksimal 8 karakter). Pesan teks akan melalui proses *sharing* menggunakan Skema (t, w) dengan $t \geq 2$ dan $t \leq w \leq 10$ sehingga menghasilkan *share* (x, y) sebanyak w. Kemudian diterapkan teknik steganografi untuk menyembunyikan setiap *share* (x, y) pada citra RGB dengan format png (*cover image*) menggunakan metode LSB. Penyisipan bit *share* (x, y) ke dalam *pixel cover image* dilakukan secara acak berdasarkan bilangan acak yang dihasilkan dari PRNG. Proses LSB menghasilkan *w* buah *share stego*. Partisipan akan menerima *QR Code* yang berisi *share stego*. Pemberian pesan rahasia dengan cara tersebut dapat menyamarkan sekaligus menyembunyikan keberadaan pesan dari pihak yang tidak berwenang.

3.2 Model Dasar

Model dasar yang digunakan dalam penelitian ini adalah Skema (t, w) dan LSB dengan *cover object* berupa citra RGB.

3.2.1 Skema *Shamir Secret Sharing*

Berikut merupakan model dasar skema SSS atau Skema (t, w) seperti pada penjelasan bagian 2.2.4.c.

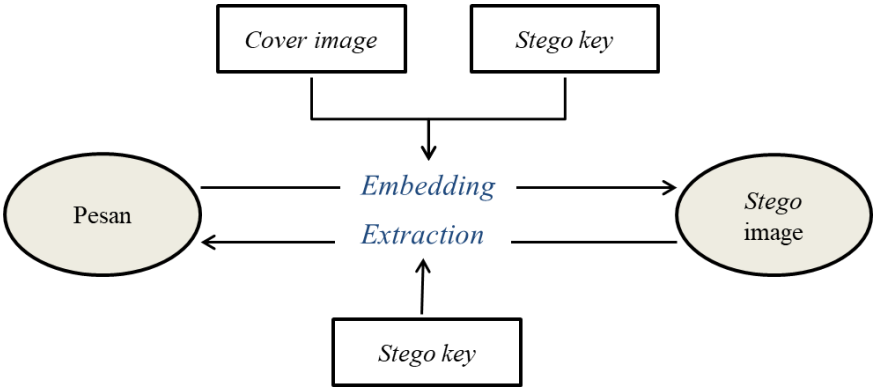


Gambar 3.1 Skema (t, w)

Pada Skema (t, w) pesan teks akan diubah menjadi *integer* M dengan cara mengkonversi setiap karakter pada pesan ke dalam desimal ASCII. M menjadi plainteks pada proses *sharing* disertai masukan bilangan prima dan bilangan bulat a_i seperti pada Gambar 3.1 untuk membentuk $s(x)$. *Share* diperoleh dari subtitusi masukan bilangan bulat x_i ke dalam $s(x_i)$.

3.2.2 Skema Steganografi *Least Significant Bit*

Berikut merupakan model dasar steganografi dengan metode *LSB* seperti pada penjelasan bagian 2.3.2:



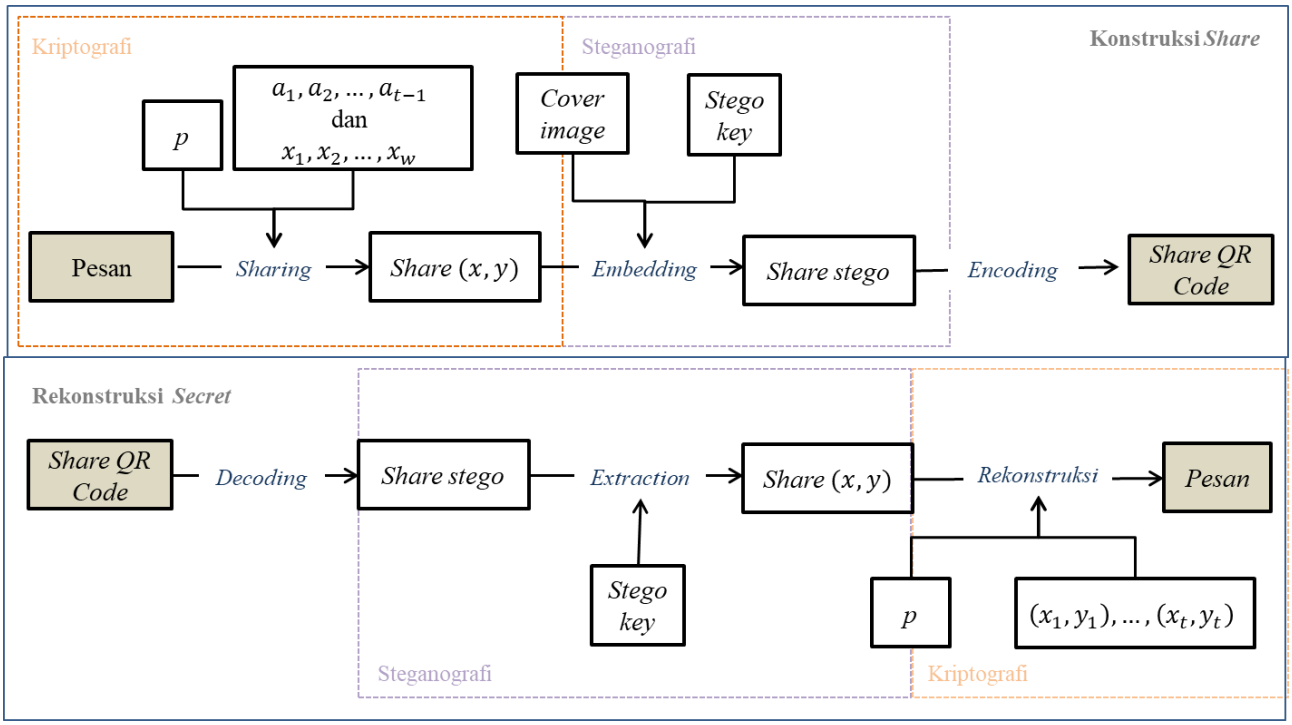
Gambar 3.2 Skema Steganografi LSB

Dalam metode LSB pada penelitian ini, proses *embedding* pesan membutuhkan masukan *cover image* dan *stego key*. *Stego key* berupa bilangan acak yang akan dibangkitkan dengan algoritma LCG sebagai metode PRNG. *Stego key* yang sama juga diperlukan untuk proses *extraction*.

3.3 Pengembangan Model

Pada penelitian ini akan dilakukan pengembangan model dasar berupa penggabungan kriptografi SSS dan steganografi LSB dengan *QR Code* kemudian diimplementasikan dalam bentuk program aplikasi menggunakan Python. Pesan asli dari model ini yaitu pesan teks, kemudian hasil konversi setiap karakternya dalam bentuk desimal berdasarkan ASCII sebagai plainteks (dalam hal ini *integer* M). Plainteks akan di-*sharing* menggunakan Skema (t, w) untuk menghasilkan *share*(x, y) sebanyak w. Setiap *share* akan dilanjutkan dengan proses *embedding* menggunakan metode LSB dengan *cover image* berupa citra RGB yang sama sehingga menghasilkan *share stego* sebanyak w. Kemudian *share stego* akan diubah menjadi *QR Code*, sehingga setiap partisipan menerima *share* dalam bentuk *QR Code*.

Untuk mengembalikan pesan asli, sebanyak t partisipan mengumpulkan *QR Code*. Kemudian *QR Code* dikembalikan menjadi *share stego*. Selanjutnya dilakukan *extraction* sehingga diperoleh *share* (x, y). *Share* (x, y) direkonstruksi menjadi plainteks menggunakan metode interpolasi Lagrange. Plainteks akan diterjemahkan kembali menjad pesan asli yang dapat dipahami partisipan. Skema pengembangan model yang telah dipaparkan tersebut dapat dilihat pada Gambar 3.3 berikut.



Gambar 3.3 Skema Pengembangan Model

3.4 Konstruksi Program Aplikasi

Program aplikasi penggabungan kriptografi SSS dan steganografi LSB dengan *QR Code* akan dibuat menggunakan Python. Ada pun input dan *output* dari rancangan program adalah sebagai berikut.

Tabel 3.1 Input dan *Ouput* Rancangan Program

Keterangan	Konstruksi *Share*	Rekonstruksi *Secret*
Input	• Pesan teks • Citra RGB (*.png) • Nilai t, w, dan p • Nilai $x_1, \dots, x_w$ • *Stego key*	• *QR Code* • Nilai t dan p • *Stego key*
Output	*QR Code*	Pesan teks

3.4.1 Algoritma Deskriptif

Algoritma deskriptif untuk konstruksi *share* dan rekonstruksi *secret* berdasarkan pengembangan model akan dijelaskan sebagai berikut.

a) Konstruksi *Share*

Pada proses konstruksi *share, Dealer* berperan sebagai *user*. Algoritma proses konstruksi *share* diuraikan sebagai berikut.

1) Masukan pesan teks masimal 8 karakter berupa karakter ASCII dari 32 sampai 126.
2) Masukan w, dan t, dengan $2 \leq t \leq w \leq 10$.
3) Program menampilkan *integer* M.
4) Masukan bilangan prima p secara acak, $p \geq M$.
5) Program menampilkan bilangan bulat acak dalam modulo p sebanyak $t - 1$.
6) Masukan *w* bilangan bulat acak x berbeda dalam modulo p.
7) Program menampilkan share (x, y).
8) Masukan sebuah citra RGB dengan format png dan *stego key*.
9) Program menyimpan *w* buah s*hare stego*.
10) Program menampilkan *QR Code* dari *share stego*.

b) Rekonstruksi *Secret*

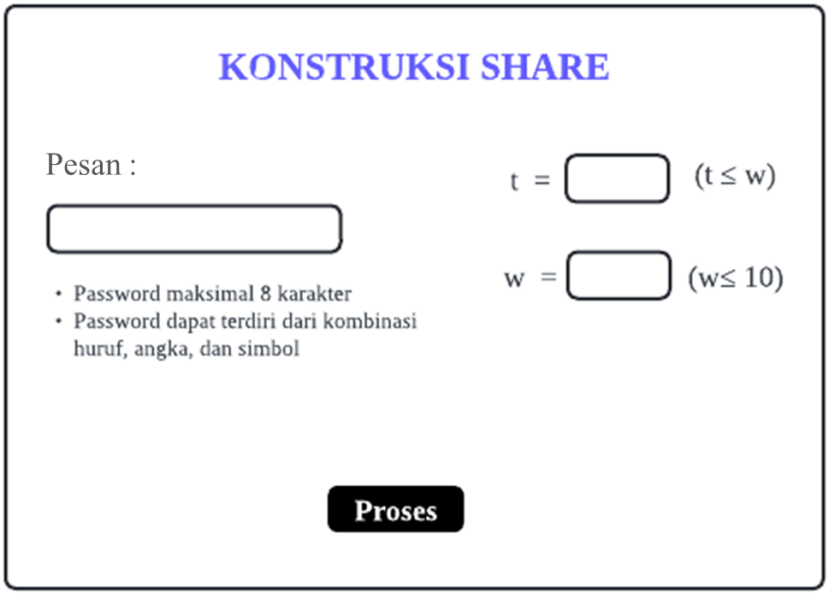
Pada proses rekonstruksi *secret*, partisipan berperan sebagai *user*. Algoritma rekonstruksi *secret* diuraikan sebagai berikut.

1) Masukan t buah *QR Code*, *stego key*, dan nilai p.
2) Program menyimpan t buah *stego share*.
3) Program mengekstraksi *share* (x, y) dari *share stego*.
4) Proses interpolasi Lagrange dari t buah titik (x, y).
5) Program menampilkan pesan.

3.4.2 Desain Tampilan

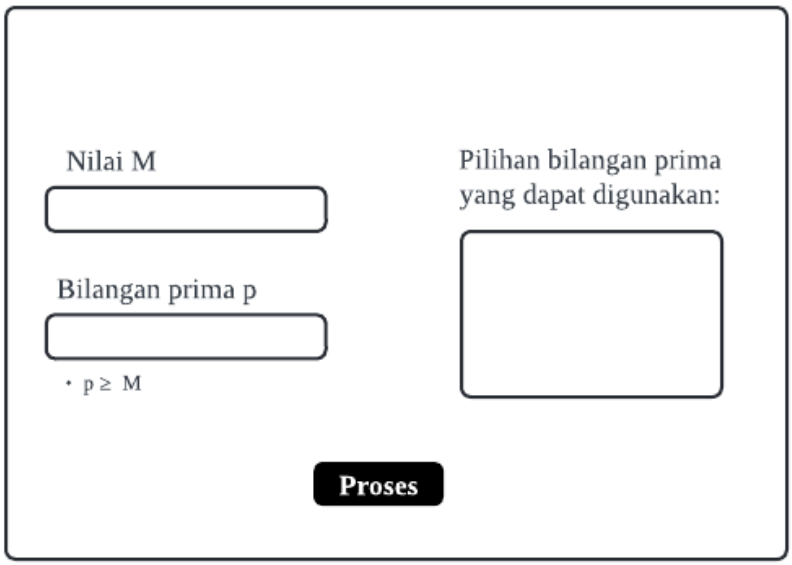
Desain tampilan utama program aplikasi akan mempunyai dua *button*, yaitu *button* **Konstruski *Share*** dan *button* **Rekonstruksi *Secret*.** Berikut desain tampilan pada program aplikasi yang akan dibuat:

Gambar 3.4 Desain Tampilan Utama Program



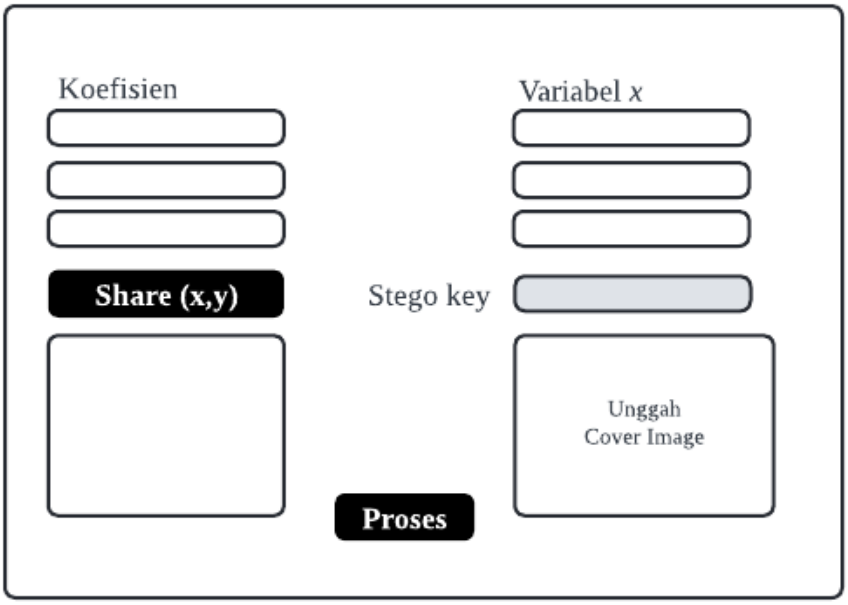
Gambar 3.5 Desain Tampilan Konstruksi *Share* (1)

Jika *user* klik **Konstruksi *Share*** program akan beralih ke tampilan inisiasi seperti pada Gambar 3.5. Pada tampilan ini terdapat *entry box* untuk *input* pesan, nilai w dan t. Kemudian terdapat *button* **Proses** yang mengarah ke tampilan berikutnya.

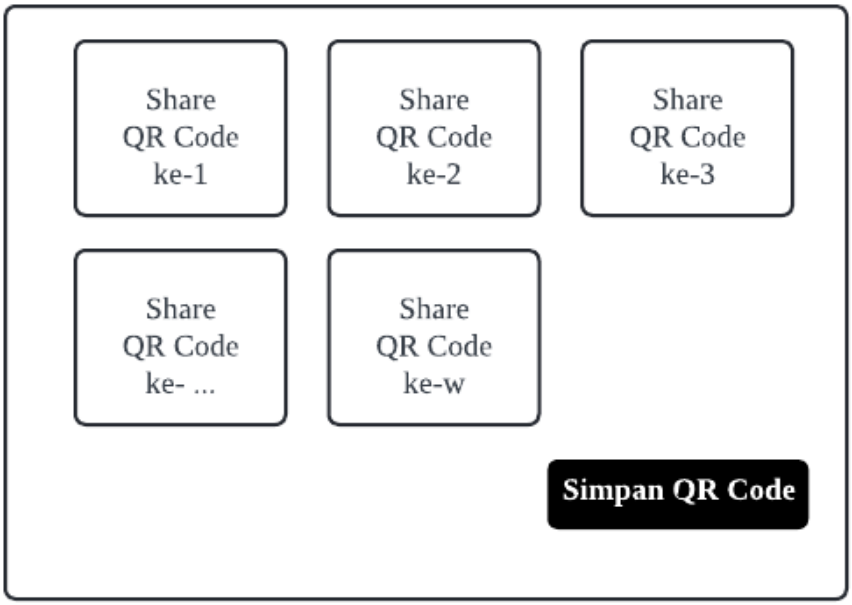


Gambar 3. 6 Desain Tampilan Konstruksi *Share* (2)

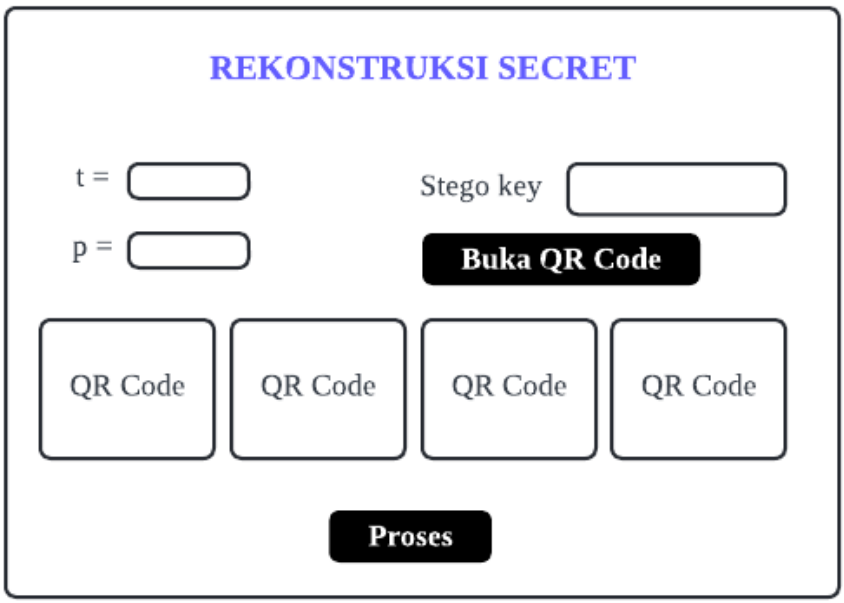
Pada tampilan selanjutnya (Gambar 3.6) terdapat *text box* yang menampilkan nilai *M* dan *entry box* untuk *input* nilai p. *Button* **Proses** akan mengubah tampilan menjadi seperti pada Gambar 3.7, di mana terdapat *text box* untuk *input* koefisien a_j dan *variabel* x_i. *Button* $\boldsymbol{Share}(\boldsymbol{x},\boldsymbol{y})$ untuk menampilkan *share* (x, y) pada *text box* di bawahnya. Pada tampilan ini *user* mengunggah *cover image* dan *input stego key*. *Button* **Proses** untuk menuju tampilan berikutnya (Gambar 3.8) di mana program menampilkan hasil akhir proses konstruksi *share* berupa *QR Code*. *QR Code* dapat di simpan dengan menekan *button* **Simpan QR Code**.



Gambar 3.7 Desain Tampilan Konstruksi *Share* (3)

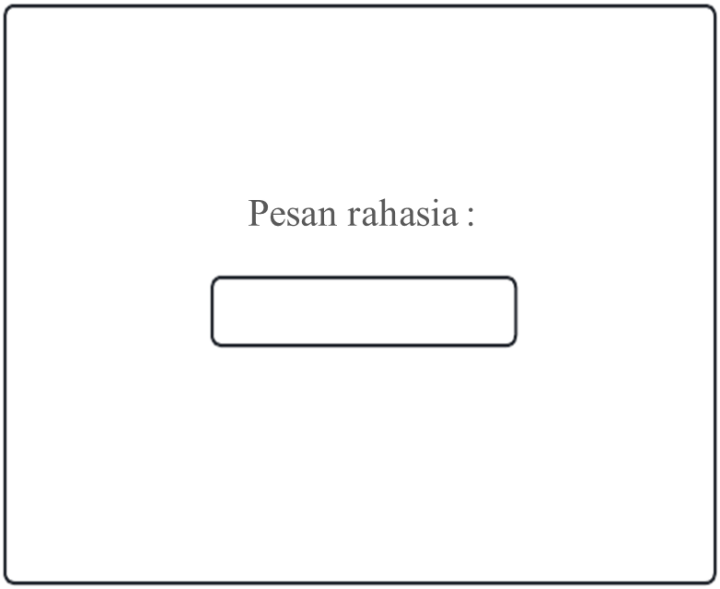


Gambar 3.8 Desain Tampilan Konstruksi *Share* (4)



Gambar 3.9 Desain Tampilan Rekonstruksi Secret (1)

Jika *user* menekan **Rekonstruksi Secret** maka program akan menampilkan Gambar 3.9 sebagai tampilan inisiasi proses rekonstruksi *secret*. Pada tampilan tersebut terdapat *entry box input* t, p, dan *stego key*. *Button* **Buka QR Code** untuk mengunggah share *QR Code* dari *file folder* ke program. *Button* **Proses** untuk menuju tampilan berikutnya yaitu pada Gambar 3.10, program menampilkan *text box* berisi pesan rahasia.



Gambar 3.10 Desain Tampilan Rekonstruksi *Secret* (2)

3.4.3 *Library*

Pada pembuatan program aplikasi akan digunakan beberapa *library* dari Python, di antaranya yaitu:

1) *Tkinter*

Tkinter adalah *graphic user interface* (GUI) standar Python yang digunakan untuk membuat tampilan visual program aplikasi. Komponen-komponen yang tersedia pada *library* di antaranya yaitu *label*, *button*, *text box*, *entry box*, *list box*, *window*, dan lainnya.

2) *Math*

Math merupakan *library* yang menyediakan fungsi-fungsi matematika dasar untuk digunakan pada operasi matematika sederhana. *Library* ini dapat menghitung operasi matematika biasa, operasi modulo, operasi logaritma, akar kuadrat, pangkat, eksponen, dan lain-lain.

3) *Random*

Random adalah *library* untuk menghasilkan bilangan acak. Pada pembuatan program, *library* ini berfungsi memberikan bilangan acak dengan *range* tertentu sebagai koefisien.

4) *NumPy*

NumPy (*Numerical Python*) adalah *library* yang menyediakan fungsi siap pakai untuk memudahkan melakukan perhitungan saintifik seperti matriks, aljabar, dan lain-lain. *NumPy* dapat mengolah data dalam bentuk *array*.

5) *Sympy*

Sympy adalah *library* untuk *symbolic mathematic*. *Sympy* memudahkan *user* untuk menyelesaikan permasalahan matematika dengan metode yang sederhana.

6) *qrcode*

qrcode adalah *library* untuk membuat *QR Code*. *Library* ini dapat tidak membaca *QR Code.*

7) *Pyzbar*

Pyzbar adalah *library* untuk membaca *barcode* dan *QR Code.*

8) PIL

PIL (*Python Imaging Library)* adalah *library* mengolah citra digital. PIL menyediakan fitur memproses citra digital dari *file folder* di komputer, membuat citra baru, mengubah ukuran dan mode citra digital, dan lainnya.

9) *Request*

Request adalah *library* untuk memudah interaksi dengan web API (*Application Programming Interface*). Pada program ini *request* digunakan untuk berinteraksi dengan ImgBB untuk mengunggah dan mengunduh *stego image* secara *online*.

10) *Imgbbpy*

Imgbbpy adalah *library* untuk mengunggah citra ke imgbb.com. ImgBB merupakan layanan *hosting* dan berbagi gambar secara gratis. ImgBB mengizinkan pengguna mengunggah gambar dengan ukuran maksimal 32 MB per gambar dan mendapatkan tautan langsung dari gambar tersebut. Untuk dapat mengunggah citra langsung dari program Python dibutuhkan *Imgbb API Key* yang diperoleh dari https://api.imgbb.com.

3.5 Proses Validasi

Pada tahap ini dilakukan validasi terhadap program aplikasi yang dibuat. Validasi dilakukan dengan memberikan contoh kasus pada program aplikasi. Perhitungan dari program pada proses konstruksi share dengan Skema (t,w) akan dibandingkan dengan proses perhitungan manual dengan bantuan Microsoft Excel. Selain itu, *stego image* yang dihasilkan program akan diuji kualitasnya menggunakan parameter PSNR. Program aplikasi tervalidasi jika *QR Code* dapat dikembalikan menjadi pesan asli.

3.6 Pengambilan Kesimpulan

Pada tahap akhir, dilakukan pengambilan kesimpulan dari hasil penelitian yang telah dilakukan. Setelah program aplikasi tervalidasi, maka algoritma penggabungan kriptografi dan steganografi ini dapat digunakan dalam membagikan informasi rahasia dengan ketentuan yang sudah dijelaskan, sehingga program aplikasi yang telah dibuat dapat digunakan oleh *user* yang membutuhkan. Penggabungan

kriptografi SSS dan steganografi LSB dengan *QR Code* diharapkan dapat meningkatkan keamanan pembagian informasi rahasia sehingga penyadap kesulitan memecahkan informasi tersebut tersebut.