BAB III
METODE PENELITIAN

3.1 Identifikasi Masalah

Implementasi steganografi *Least Significant Bit* dan kriptografi visual *Secret Sharing* untuk keamanan data *QR Code* pada penelitian ini bertujuan untuk meningkatkan salah satu aspek keamanan pesan yaitu kerahasiaan. Penggunaan *QR Code* sebagai media penyimpanan data semakin luas dalam berbagai aplikasi digital, namun data di dalamnya rentan terhadap akses tidak sah jika tidak dilindungi dengan mekanisme keamanan yang memadai. Dengan metode steganografi, seperti *Least Significant Bit*, dapat menyembunyikan data biner ke dalam media gambar, tetapi teknik ini memiliki kelemahan, karena data yang tersembunyi dapat diekstrak dengan mudah jika media tersebut terdeteksi. Oleh karena itu, diperlukan mekanisme tambahan yang mampu melindungi data tersembunyi agar tidak dapat diakses, salah satunya adalah kriptografi visual *Secret Sharing*, yang membagi *stego-image* menjadi beberapa *share* untuk meningkatkan tingkat keamanan. Namun, penggabungan antara steganografi *Least Significant Bit* dan kriptografi visual *Secret Sharing* untuk pengamanan data pada *QR Code* masih jarang diterapkan dalam penelitian. Oleh karena itu, penelitian ini bertujuan untuk menjawab tantangan tersebut dengan merancang sistem pengamanan yang menggabungkan kedua metode ini, sehingga dapat memastikan data *QR Code* tetap aman.

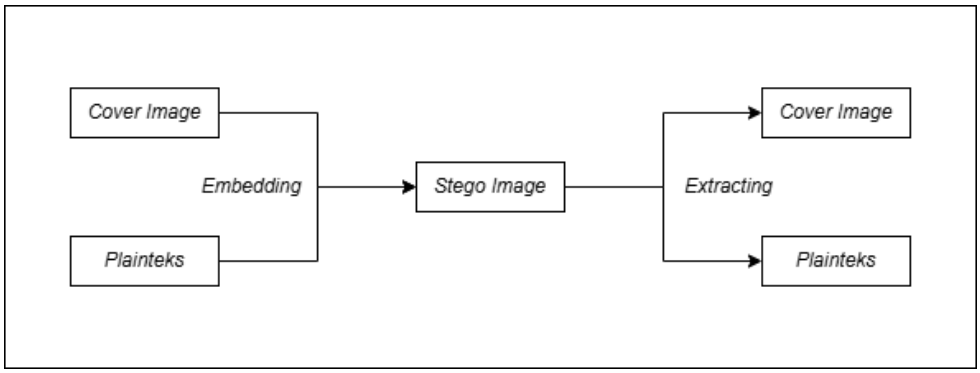
3.2 Model Dasar

Model dasar yang digunakan dalam penelitian ini adalah steganografi *Least Significant Bit* dan kriptografi visual *Secret Sharing*.

3.2.1 *Least Significant Bit*

Dalam penelitian ini, metode *Least Significant Bit* digunakan untuk menyisipkan pesan ke dalam *cover image*. Proses penyisipan memerlukan dua masukan utama, yaitu *cover image* dan plainteks. Plainteks berupa data biner yang diekstrak dari *QR Code* dan berfungsi sebagai elemen penting dalam proses *embedding*. Selain itu, plainteks yang sama juga diperlukan dalam tahap ekstraksi untuk memastikan data yang disembunyikan dapat diambil kembali dengan akurat.

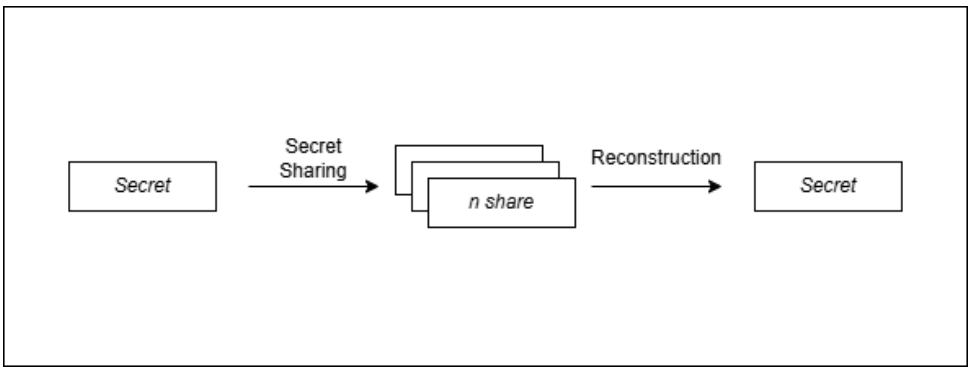
Gambar 3.1 berikut merupakan model dasar steganografi dengan metode *Least Significant Bit* seperti pada penjelasan bagian 2.3.2:



Gambar 3.1 Skema Steganografi *Least Significant Bit*

3.2.2 Visual *Secret Sharing*

Visual *Secret Sharing* adalah salah satu metode kriptografi untuk merahasiakan citra dengan cara membagi citra sebanyak n *shares*. *Shares* yang diperoleh merupakan gambar abstrak yang apabila seluruh *shares* digabungkan, citra pesan dapat diperoleh kembali, sebagaimana ditunjukkan pada Gambar 3.2.



Gambar 3.2 Skema Kriptografi Visual *Secret Sharing*

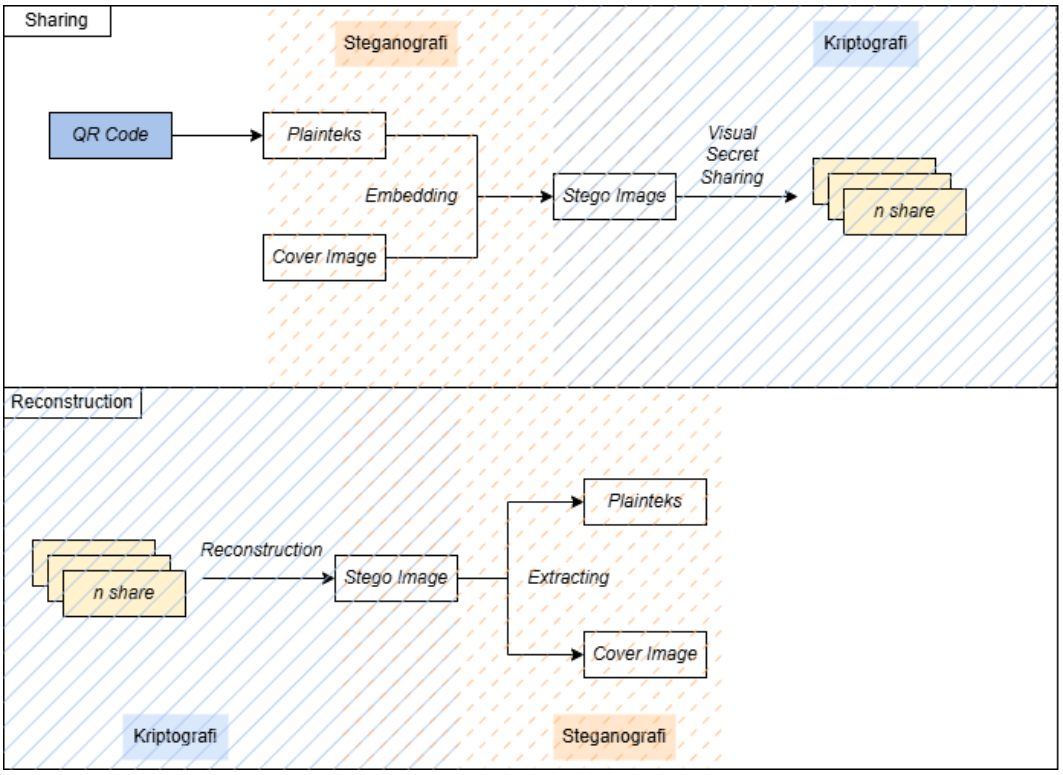
3.3 Pengembangan Model

Pada penelitian ini, akan dilakukan pengembangan model dasar berupa penggabungan metode steganografi *Least Significant Bit* dan kriptografi Visual *Secret Sharing* untuk melindungi data biner dari *QR Code*. Model ini akan diimplementasikan dalam bentuk program aplikasi menggunakan bahasa pemrograman *Python*.

Data awal yang digunakan dalam model ini berupa data *QR Code* yang diekstrak menjadi data biner. Data biner ini akan berfungsi sebagai pesan rahasia yang akan disisipkan ke dalam *cover image* berupa citra *grayscale* menggunakan metode steganografi *Least Significant Bit*. Proses *embedding* menghasilkan *stego-*

image yang menyimpan data biner secara tersembunyi tanpa mengubah tampilan visual dari *cover image* secara signifikan. Selanjutnya, *stego-image* akan diproses menggunakan metode kriptografi Visual *Secret Sharing* untuk dibagi menjadi *n share*. Setiap *share* berupa citra yang secara individu tidak bermakna, sehingga data tidak dapat diakses tanpa mengumpulkan semua *share*. Setiap *share* akan didistribusikan kepada partisipan untuk meningkatkan keamanan data yang tersembunyi.

Untuk mengembalikan data *QR Code* yang asli, partisipan harus mengumpulkan *k share*. *Share-share* tersebut akan digabungkan kembali untuk merekonstruksi *stego-image*. Setelah *stego-image* terbentuk, proses ekstraksi dilakukan menggunakan metode steganografi *Least Significant Bit* untuk memperoleh kembali data biner. Data biner ini kemudian dikonversi ulang menjadi *QR Code* yang sama dengan data awal. Proses lengkap dari pengembangan model dapat dilihat pada diagram di Gambar 3.3 berikut.



Gambar 3.3 Skema Pengembangan Model

3.4 Konstruksi Program Aplikasi

3.4.1 *Input* dan *Output*

Pada program aplikasi ini, *input* pertama adalah *QR Code* dalam bentuk data biner yang akan di-*embed* ke dalam *cover image grayscale* menggunakan metode *Least Significant Bit*, menghasilkan *stego image*. Selanjutnya, *stego image* ini akan dibagi menjadi beberapa bagian (*share*) menggunakan metode Visual *Secret Sharing*. Untuk *output* proses ini, program menghasilkan beberapa *share* yang dapat dibagikan kepada partisipan.

Dalam proses *reconstruction*, pengguna menginputkan beberapa *share* yang telah diterima, kemudian menggunakan metode Visual *Secret Sharing* untuk menggabungkannya kembali menjadi *stego image*. Setelah itu, dilakukan *extracting* pada stego image untuk memperoleh kembali data biner *QR Code*, yang kemudian dikembalikan menjadi pesan atau data dari *QR Code* semula. Tabel 3.1 menampilkan detail *input* dan *output* dari program aplikasi ini.

Tabel 3.1 *Input* dan *Output* Program Aplikasi

	Input	*Output*
Sharing	*QR Code*	Data Biner
	Data Biner + *Cover Image*	*Stego Image*
	Stego Image	*n share*
Reconstruction	*k share*	Data Biner
	Data Biner	Pesan atau Data *QR Code*

3.4.2 Algoritma Deskriptif

Algoritma deskriptif untuk *sharing* dan *reconstruction* berdasarkan pengembangan model akan dijelaskan sebagai berikut.

a. Sharing

Langkah-langkah untuk melakukan *sharing* akan dijelaskan sebagai berikut:

1) Masukan *QR Code* berupa file gambar dengan format PNG atau JPG.
2) Masukan *cover image* dalam format *grayscale* yang akan digunakan untuk menyisipkan data *QR Code*.
3) Program akan mengekstrak data *QR Code* menjadi data biner.

4) Program akan melakukan *embedding* data biner *QR Code* ke dalam *cover image* menggunakan metode *Least Significant Bit*.
5) Program menyimpan hasil *embedding* berupa *stego image* dalam format PNG.
6) Program akan membagi *stego image* menjadi beberapa *share* dengan metode Visual *Secret Sharing* dan menyimpan hasilnya dalam format PNG.
7) Program menampilkan file *share image* yang dapat didistribusikan ke partisipan.

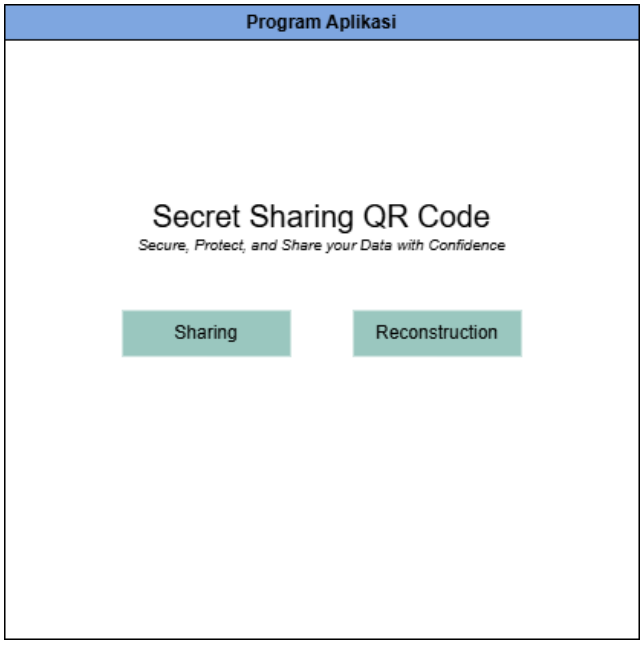
b. Reconstruction

Langkah-langkah untuk melakukan *reconstruction* akan dijelaskan sebagai berikut:

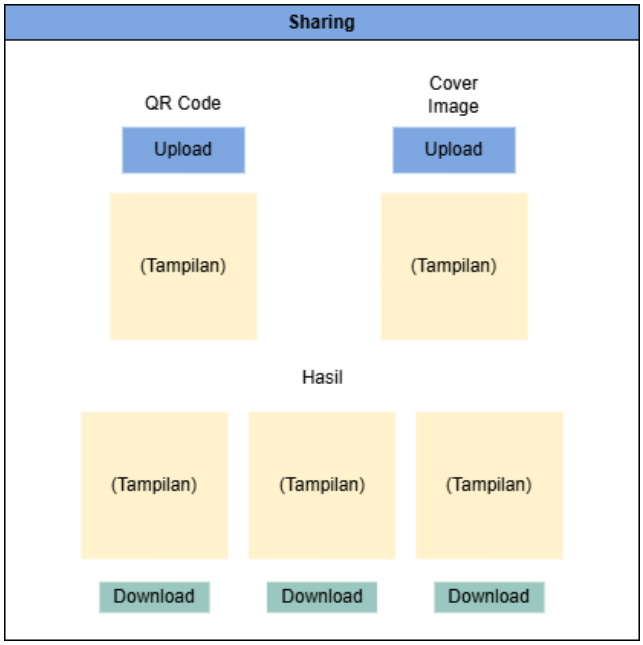
1) Masukan beberapa *share image* yang telah diterima ke dalam program.
2) Program akan merekonstruksi *stego image* dari *share* menggunakan metode Visual *Secret Sharing*.
3) Program melakukan proses *extracting* pada *stego image* untuk mendapatkan data biner dari *QR Code*.
4) Program mengembalikan data biner menjadi data *QR Code* asli dan menampilkan hasilnya.

3.4.3 Desain Tampilan

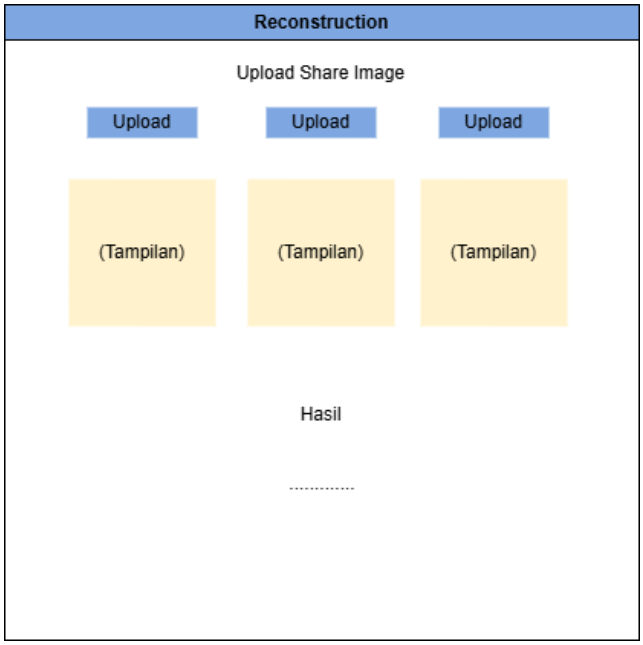
Gambar 3.4, 3.5, dan 3.6 berikut adalah desain tampilan program awal, *sharing*, dan *reconstruction* dalam penelitian ini:



Gambar 3.4 Rancangan Tampilan Program Bagian Awal



Gambar 3.5 Rancangan Tampilan Program Bagian *Sharing*



Gambar 3.6 Rancangan Tampilan Program Bagian *Reconstruction*

3.4.4 *Library Python*

Terdapat beberapa *library* dari *Python* yang digunakan dalam pemrograman yang dilakukan. Berikut *library* yang digunakan dalam penelitian ini:

1) *Tkinter*

   Tkinter adalah *library* bawaan *Python* untuk membangun antarmuka grafis (GUI), seperti *Button*, *Textbox*, *Label*, Frame dan lainnya.

2) *Qrcode*

Qrcode adalah *library* untuk membuat *QR Code* dari teks atau data, menghasilkan gambar *QR Code* dengan mudah.

3) *Pyzbar*

Pyzbar digunakan untuk membaca dan memindai *QR Code* atau *barcode*, mengekstrak data dari gambar dengan cepat.

4) Numpy

Numpy adalah *library* untuk komputasi ilmiah, mendukung operasi numerik cepat pada *array* multidimensi.

5) *Pillow* (PIL)

Pillow adalah *library* manipulasi gambar yang mendukung operasi seperti membaca, menyimpan, dan mengedit gambar.

6) *OpenCV*

OpenCV adalah *library* canggih untuk pemrosesan gambar dan video, seperti deteksi objek dan pengolahan citra tingkat lanjut.

3.5 Proses Validasi

Proses validasi dilakukan untuk memastikan bahwa program aplikasi berjalan sesuai dengan model yang dirancang. Validasi dilakukan dengan menguji dua jenis data *QR Code*, yaitu *QR Code* yang berisi pesan teks dan *QR Code* yang berisi *link* atau tautan URL. Pada setiap kasus, program diuji untuk memastikan bahwa data *QR Code* dapat dikembalikan sepenuhnya setelah melalui proses *embedding*, pembagian *share*, dan dekripsi. Validasi ini menilai apakah data asli tetap terjaga baik secara visual maupun dari segi isi. Program dianggap tervalidasi apabila seluruh jenis data yang diuji dapat direkonstruksi dengan akurasi tinggi.

3.6 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan sebagai tahapan terakhir berdasarkan hasil yang diperoleh dari penelitian ini, serta pemberian saran untuk penelitian selanjutnya agar memperoleh hasil yang lebih baik.