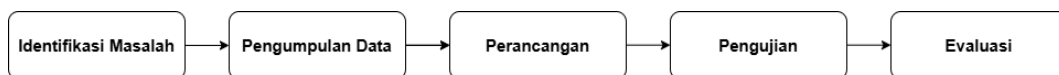


BAB III METODOLOGI PENELITIAN

3.1. Metodologi Penelitian

Metodologi penelitian ini menggunakan pendekatan eksperimental dengan tahapan sistematis yang disajikan dalam gambar 3.1, meliputi identifikasi masalah, pengumpulan data, perancangan, pengujian dan evaluasi. Fokus utama penelitian ini adalah optimasi penyisipan data dalam steganografi dengan kombinasi *AVL Tree* dan *Pixel Indicator Technique* (PIT) untuk meningkatkan kualitas *stego-image*.



Gambar 3.1 Tahapan Penelitian

3.1.1. Identifikasi Masalah

Pada tahap pertama, identifikasi masalah dilakukan melalui studi literatur yang mencakup penelitian-penelitian sebelumnya serta berbagai referensi yang relevan.

3.1.2. Pengumpulan Data

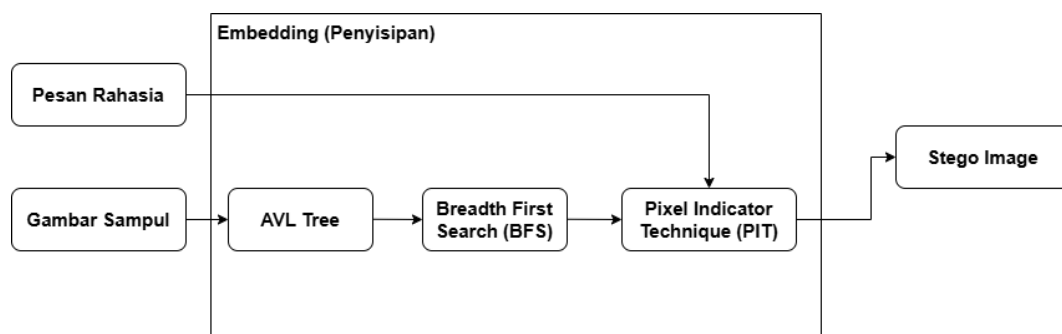
Pengumpulan data dalam penelitian ini dilakukan dengan menggunakan dataset gambar digital berwarna 24-bit dalam format BMP, yang disesuaikan dengan penelitian terdahulu yang berjudul “*High-Secured Image LSB Steganography Using AVL-Tree with Random RGB Channel Substitution*”, agar hasil perbandingan menjadi valid. Gambar-gambar tersebut dipilih untuk menguji efektivitas metode penyisipan data yang dikembangkan. Data teks yang akan disisipkan diperoleh dari internet, dengan ukuran 1, 2, 4, 8, 16 dan 32 KB. Proses

pengumpulan data mencakup pemilihan dataset gambar sebagai *cover image*, serta penyesuaian ukuran data teks.

3.1.3. Perancangan

Perancangan metode ini bertujuan untuk mengembangkan sistem penyisipan dan ekstraksi data dalam steganografi gambar menggunakan kombinasi *AVL Tree* dan modifikasi *Pixel Indicator Technique* (PIT). Berikut adalah tahapan dalam proses penyisipan dan ekstraksi data:

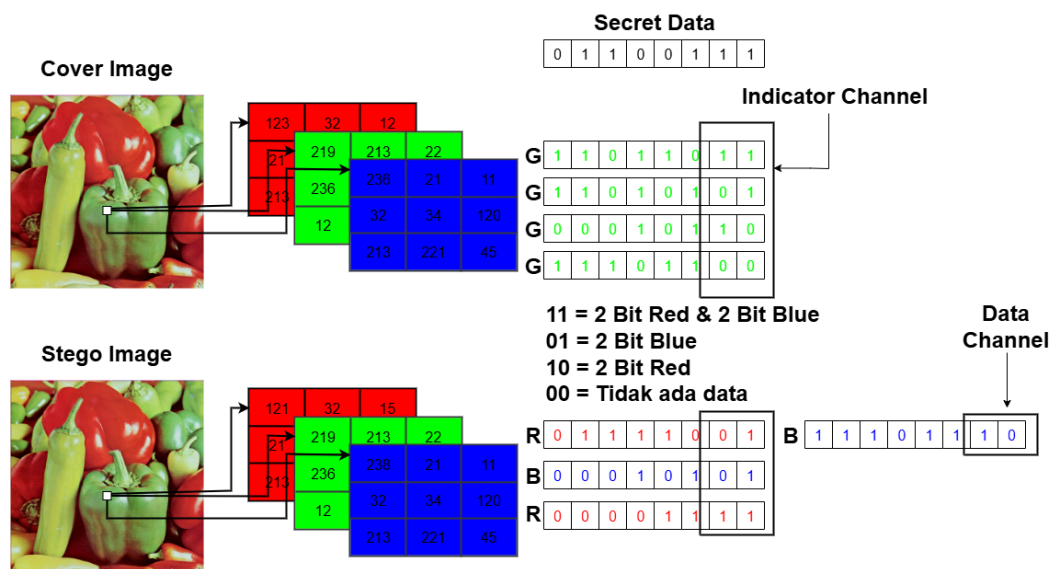
a. Penyisipan (*Embedding*)



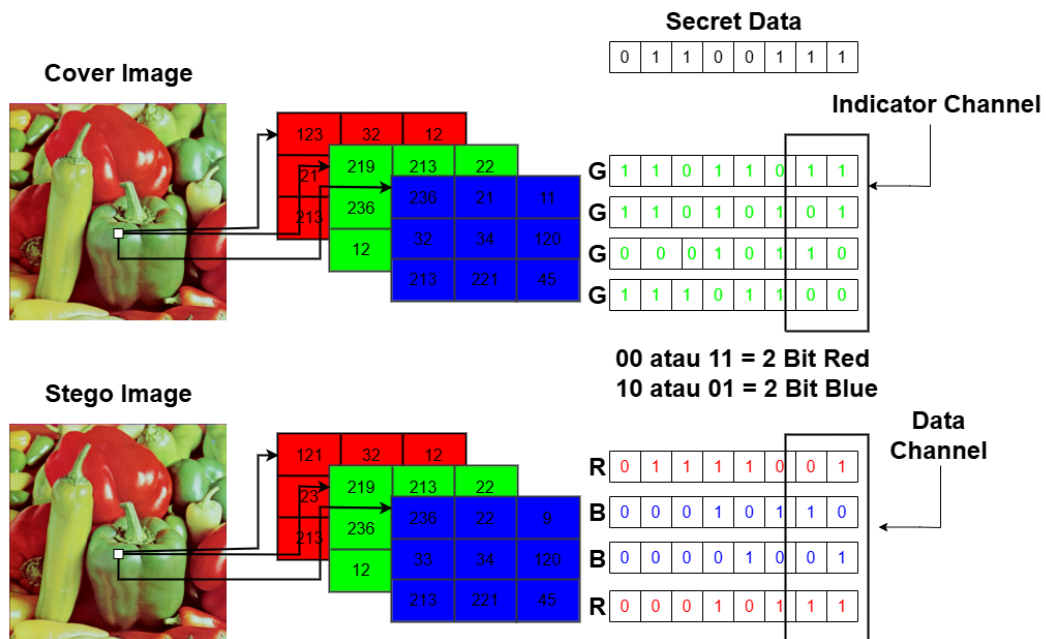
Gambar 3.2 Alur Penyisipan

Gambar 3.2 menunjukkan alur penyisipan yang dimulai dengan menerima dua input utama, yaitu pesan rahasia dan *cover image*. Pesan rahasia dikonversi ke dalam format biner agar dapat disisipkan ke dalam *cover image* secara sistematis. Sementara itu, *cover image* dikonversi menjadi array piksel dalam format RGB. Selanjutnya, pembangunan *AVL Tree* dilakukan dengan memasukkan setiap piksel berdasarkan nilai bit dari saluran hijau (G). *AVL Tree* digunakan untuk mengacak posisi piksel yang akan digunakan dalam penyisipan pesan rahasia, sehingga bit pesan tidak disisipkan secara berurutan. Setelah membangun *AVL Tree* berdasarkan

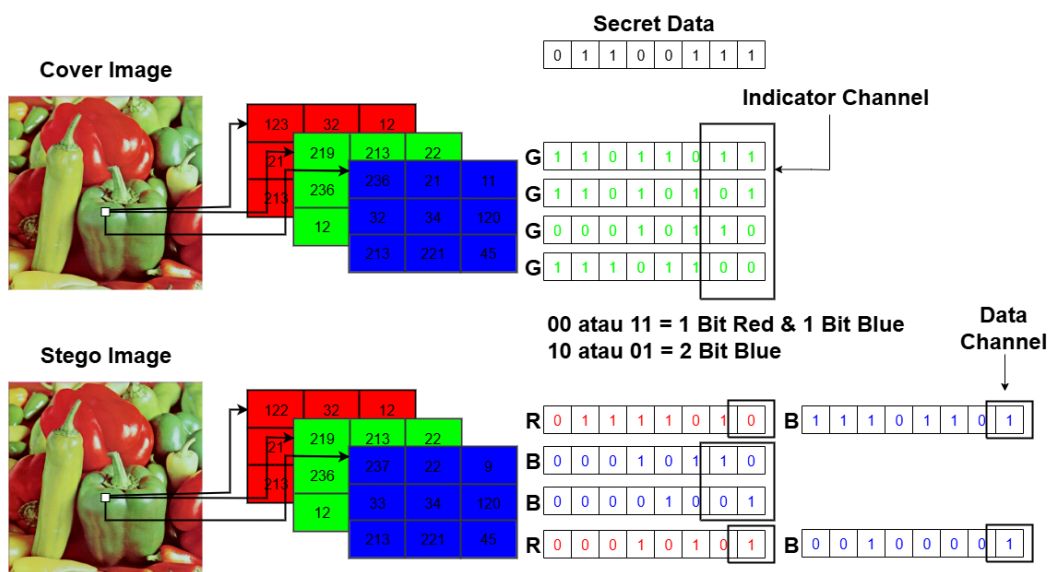
nilai bit channel hijau (G), traversal *Breadth First Search* (BFS) digunakan untuk menyusun urutan atau lokasi piksel yang akan digunakan dalam proses penyisipan data. Pada tahap penyisipan data, pesan yang telah dikonversi menjadi bit-bit biner disisipkan ke dalam saluran merah atau biru berdasarkan nilai bit saluran hijau dengan menggunakan metode *Pixel Indicator Technique* (PIT) yang telah dimodifikasi, seperti yang ditunjukkan pada Gambar 3.3(c).



(a) PIT (Gutub et al., 2008)



(b) PIT (Njoum et al., 2024)



(c) PIT (Penelitian Sekarang)

Gambar 3.3 *Pixel Indicator Technique* (PIT)

Gambar 3.3 terdiri dari tiga bagian yang menunjukkan perbandingan antara metode PIT dari penelitian sebelumnya dan modifikasi PIT yang dilakukan pada penelitian ini. Gambar 3.3(a) merupakan PIT standar yang digunakan dalam

penelitian (Gutub et al., 2008). Proses penyisipan data dilakukan berdasarkan nilai bit *least significant bit* (LSB) dari saluran hijau (G). Jika 2 bit terakhir dari saluran hijau (G) memiliki nilai bit 00, maka tidak ada bit pesan yang disisipkan. Jika 2 bit terakhir dari saluran hijau (G) memiliki nilai bit 01, maka 2 bit pesan akan disisipkan ke saluran biru (B). Jika 2 bit terakhir dari saluran hijau (G) memiliki nilai bit 10, maka 2 bit pesan akan disisipkan ke saluran merah (R). Sementara itu, Jika 2 bit terakhir dari saluran hijau (G) memiliki nilai bit 11, maka 2 bit pesan akan disisipkan ke saluran merah (R) dan biru (B).

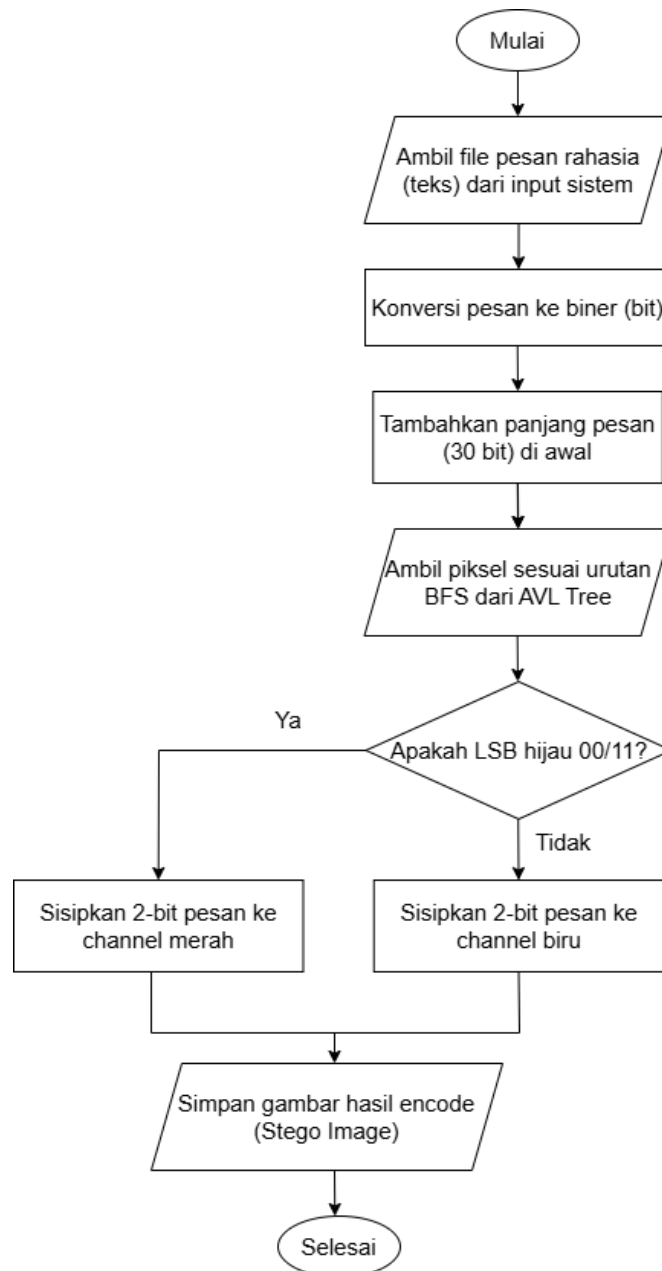
Gambar 3.3(b) menggambarkan PIT yang digunakan dalam penelitian (Njourn et al., 2024). Proses penyisipan data dilakukan berdasarkan nilai bit *least significant bit* (LSB) dari saluran hijau (G). Jika 2 bit terakhir dari saluran hijau (G) memiliki nilai bit 00 atau 11, maka 2 bit pesan akan disisipkan ke saluran merah (R). Namun, jika 2 bit terakhir dari saluran hijau (G) memiliki nilai bit 10 atau 01, maka 2 bit pesan akan disisipkan ke saluran biru (B).

Gambar 3.3(c) merupakan modifikasi PIT yang digunakan dalam penelitian ini. Proses penyisipan data dilakukan berdasarkan nilai bit *least significant bit* (LSB) dari saluran hijau (G). Jika 2 bit terakhir dari saluran hijau (G) memiliki nilai bit 00 atau 11, maka masing-masing 1 bit pesan akan disisipkan ke saluran merah (R) dan biru (B). Namun, jika 2 bit terakhir dari saluran hijau (G) memiliki nilai bit 10 atau 01, maka 2 bit pesan akan disisipkan hanya ke saluran biru (B). Setelah seluruh bit pesan berhasil disisipkan ke dalam *cover image*, tahap akhir adalah penyimpanan *stego-image*, di mana gambar yang telah mengandung pesan rahasia disimpan kembali dalam format BMP.



Gambar 3.4 Flowchart Proses Pembentukan AVL Tree dan BFS

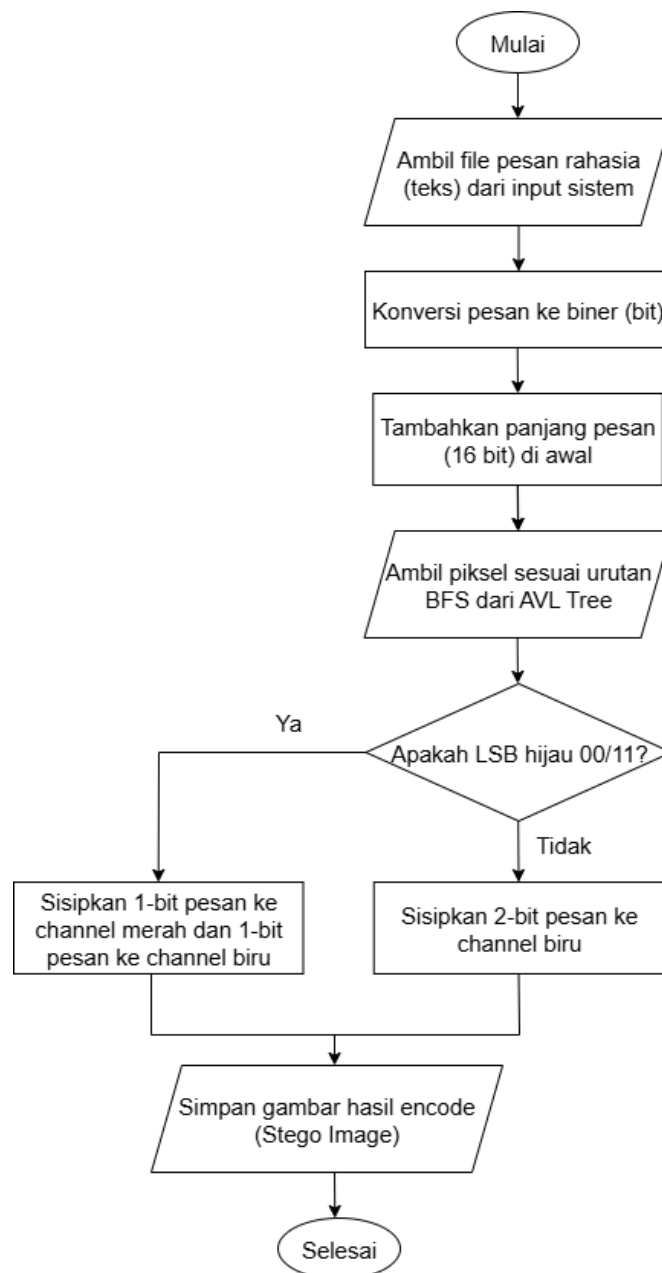
Gambar 3.4 menggambarkan tahapan pembentukan AVL Tree dan proses Breadth First Search (BFS) untuk menyusun urutan piksel yang digunakan dalam proses penyisipan dan ekstraksi data. Seluruh piksel dari cover image diekstrak dan dimasukkan ke dalam struktur data AVL Tree berdasarkan nilai channel hijau (G) yang dianggap paling sensitif terhadap perubahan visual. Setiap node AVL Tree menyimpan nilai R, G, B serta indeks posisi piksel. Setelah struktur AVL Tree selesai dibangun, dilakukan proses traversal menggunakan algoritma BFS untuk menghasilkan urutan piksel yang akan digunakan pada proses penyisipan (*embedding*) dan ekstraksi (*extracting*).



Gambar 3.5 Flowchart Penyisipan Data (Njoun et al., 2024)

Gambar 3.5 menunjukkan alur proses penyisipan data rahasia ke dalam *cover image* menggunakan metode AVL Tree dan *Pixel Indicator Technique* (PIT) berdasarkan penelitian (Njoun et al., 2024). Proses dimulai dengan pengambilan file pesan rahasia dari input sistem, kemudian dikonversi ke bentuk biner (bit) dan

ditambahkan header berupa panjang pesan (30 bit) di bagian awal untuk mempermudah proses ekstraksi. Urutan piksel untuk penyisipan data ditentukan berdasarkan hasil traversal algoritma *Breadth-First Search* (BFS) pada struktur AVL Tree yang ditunjukkan pada Gambar 3.4. Berdasarkan nilai 2 Least Significant Bit (LSB) dari channel hijau, proses menentukan mekanisme penyisipan data: jika LSB hijau adalah 00 atau 11, maka 2 bit pesan disisipkan ke saluran merah; jika tidak, maka 2 bit pesan disisipkan ke saluran biru. Proses ini diulang hingga seluruh pesan tersisipkan, dan hasil akhirnya adalah stego-image yang disimpan dalam format bitmap.



Gambar 3.6 Flowchart Penyisipan Data (Penelitian Sekarang)

Gambar 3.6 menunjukkan alur proses penyisipan data rahasia ke dalam *cover image* menggunakan metode AVL Tree dan Modifikasi *Pixel Indicator Technique* (PIT) yang digunakan dalam penelitian ini. Proses dimulai dengan pengambilan file pesan rahasia dari input sistem, kemudian dikonversi ke bentuk biner (bit) dan ditambahkan header berupa panjang pesan (16 bit) di bagian awal untuk

mempermudah proses ekstraksi. Urutan piksel untuk penyisipan data ditentukan berdasarkan hasil traversal algoritma *Breadth-First Search* (BFS) pada struktur AVL Tree yang ditunjukkan pada Gambar 3.4. Berdasarkan nilai 2 *Least Significant Bit* (LSB) dari channel hijau, proses menentukan mekanisme penyisipan data: jika LSB hijau adalah 00 atau 11, maka masing-masing 1 bit pesan disisipkan ke saluran merah dan biru; jika tidak, maka 2 bit pesan disisipkan ke saluran biru. Proses ini diulang hingga seluruh pesan tersisipkan, dan hasil akhirnya adalah stego-image yang disimpan dalam format bitmap.

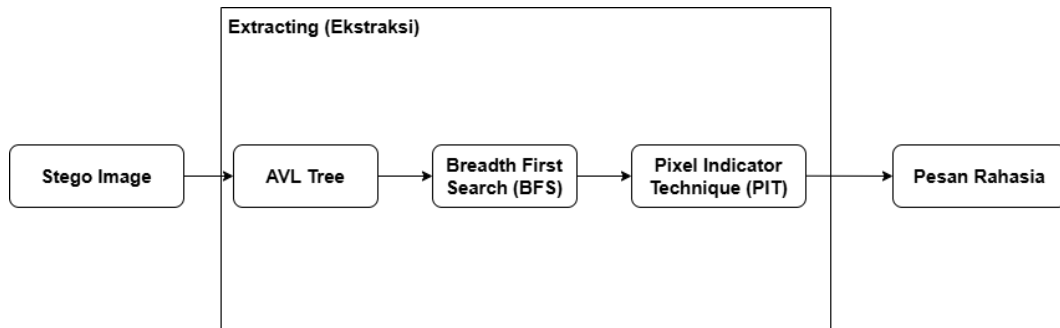
Tabel 3.1 Perbandingan Modifikasi PIT

Parameter	PIT (Gutub et al., 2008)	PIT (Njourn et al., 2024)	PIT (Penelitian Sekarang)
Kapasitas per pixel	-	2 Bit	2 Bit
00	Tidak ada Data	2 Bit Red	1 Bit Red dan 1 Bit Blue
01	2 Bit Blue	2 Bit Blue	2 Bit Blue
10	2 Bit Red	2 Bit Blue	2 Bit Blue
11	2 Bit Red dan 2 Bit Blue	2 Bit Red	1 Bit Red dan 1 Bit Blue
MSE	-	0.0124	Kemungkinan akan turun
PSNR	-	67.21	Kemungkinan akan naik

Keterangan

Simbol - = Belum diketahui

b. Ekstraksi (*Extracting*)



Gambar 3.7 Alur Ekstraksi

Gambar 3.7 merupakan alur ekstraksi yang dimulai dengan menerima *stego-image* sebagai input, yaitu gambar yang telah mengandung pesan rahasia. *Stego image* dikonversi menjadi array piksel dalam format RGB agar dapat diakses dan diproses lebih lanjut. Langkah berikutnya adalah rekonstruksi *AVL Tree*, di mana struktur *AVL Tree* dibangun kembali berdasarkan nilai bit saluran hijau dari *stego-image*. Proses ini dilakukan untuk memastikan urutan piksel yang digunakan dalam ekstraksi sesuai dengan urutan yang digunakan saat penyisipan. Setelah *AVL Tree* terbentuk, traversal *Breadth First Search* (BFS) digunakan untuk menentukan kembali urutan piksel yang akan diekstrak agar sesuai dengan penyisipan sebelumnya. Pada tahap ekstraksi data, bit pesan yang telah disisipkan sebelumnya diambil kembali berdasarkan posisi penyisipan yang ditentukan oleh saluran hijau. Proses ini mengikuti aturan yang sama seperti pada penyisipan menggunakan *Pixel Indicator Technique* (PIT) yang telah dimodifikasi. Sistem dapat mengetahui di saluran mana pesan telah disisipkan. Setelah seluruh bit pesan berhasil diekstrak, bit-bit pesan yang telah diambil dikonversi kembali ke format aslinya agar dapat dibaca.

3.1.4. Pengujian

Penelitian ini memiliki kebutuhan perangkat keras dan perangkat lunak untuk mendukung proses pengujian. Perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini disajikan pada Tabel 3.2.

Tabel 3.2 Perangkat Keras dan Perangkat Lunak yang digunakan

Jenis	Spesifikasi
Processor	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
RAM	8 GB
Storage	SSD 259 GB
Sistem Operasi	Windows 11
Bahasa Pemrograman	Python, HTML
Lingkungan Pengembangan	Visual Studio Code

Pengujian dilakukan pada berbagai skenario penyisipan dengan ukuran data antara 1 KB hingga 10 KB. Cover image yang digunakan mengacu pada penelitian sebelumnya berjudul “*High-Secured Image LSB Steganography Using AVL-Tree with Random RGB Channel Substitution*”, yaitu Tiffany.bmp, Baboon.bmp dan Pepper.bmp. Tabel 3.3 menyajikan gambaran lengkap dari skenario pengujian yang digunakan dalam penelitian ini. Persentase perubahan dihitung untuk masing-masing parameter menggunakan rumus (1) dan (2).

$$\text{Persentase MSE} = \frac{\text{Selisih antara MSElama dan MSEbaru}}{\text{MSElama}} \times 100 \quad (1)$$

$$\text{Persentase PSNR} = \frac{\text{Selisih antara PSNRLama dan PSNRbaru}}{\text{PSNRLama}} \times 100 \quad (2)$$

Tabel 3.3 Pengujian

Pesan Rahasia (KB)	<i>Stego-Image</i>					
	Tiffany.bmp		Baboon.bmp		Pepper.bmp	
	MSE	PSNR (dB)	MSE	PSNR (dB)	MSE	PSNR (dB)
1						
2						
4						
8						
16						
32						

3.1.5. Evaluasi

Evaluasi kualitas *stego-image* dilakukan dengan mengukur *Mean Square Error* (MSE) dan *Peak Signal-to-Noise Ratio* (PSNR). MSE digunakan untuk menghitung rata-rata perbedaan kuadrat antara piksel gambar asli dan *stego-image*, di mana nilai MSE yang lebih rendah menunjukkan distorsi yang lebih kecil. PSNR mengukur rasio antara nilai maksimum piksel dan tingkat distorsi akibat penyisipan data, dengan nilai PSNR yang lebih tinggi menunjukkan kualitas *stego-image* yang lebih baik.