Implementasi *Fragile Watermarking* dan Steganografi *Least Significant Bit* pada File Citra

Muhammad Akyas David Al Aleey - 13520011
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): akyasdavid007@gmail.com

***Abstrak*—Perkembangan teknologi digital memudahkan persebaran informasi melalui berbagai media. Hal ini sejalan dengan meningkatnya kejahatan dalam pemalsuan atau upaya dalam merusak suatu citra digital yang didistribusikan. Oleh karena itu, dibutuhkan pendekatan dalam meingkatkan keamanan dan keaslian data digital. Pada makalah ini, akan dilakukan percobaan *fragile watermarking* dan steganografi LSB pada citra. Penggunaan *fragile* watermarking dimaksudkan untuk melindungi integritas citra dengan menyisipkan *watermark* yang sensitif terhadap perubahan. Sementara steganografi LSB digunakan untuk menyembunyikan pesan rahasia dalam citra tanpa mengubah tampilan visualnya secara signifikan. Dengan demikian, diharapkan implementasi tersebut dapat mendeteksi perubahan yang tidak sah pada citra dan menjaga keaslian citra sekaligus menyembunyikan pesan.**

***Kata Kunci—fragile watermarking,* steganografi, LSB**

I. PENDAHULUAN

Fenomena digitalisasi yang berkembang begitu pesat tentu menuntut seseorang untuk memaksimalkan teknologi digital dalam aktivitasnya sehari-hari. Penggunaan dan pertukaran informasi melalui media digital dalam format yang beraneka ragam menjadi sangat umum di sekitar kita, salah satunya yakni citra digital. Kemudahan distribusi citra digital di berbagai media tidak dapat dipungkiri akan membawa dampak positif maupun negatif. Di satu sisi pengguna maupun pemilik media citra dapat mengakses maupun menyebarkan citra digital ke berbagai media. Namun, di sisi lain, kemudahan tersebut dapat menimbulkan resiko kejahatan berupa pemalsuan data pada file citra sehingga pertukaran informasi menjadi tidak aman. Selain itu, citra digital juga akan sangat mudah untuk diakuisisi oleh pihak lain jika pemilik aslinya tidak mempunyai hak cipta untuk melindungi aset yang ia miliki tersebut. Oleh karena itu, keamanan dan keaslian informasi digital menjadi sangat penting. Sehingga, dibutuhkan suatu metode yang dapat melindungi dan menjaga keaslian suatu citra digital. Salah satu metode yang digunakan untuk melindungi integritas dan keaslian informasi digital tersebut adalah dengan teknik *watermarking* dan steganografi.

Watermarking adalah proses menyisipkan suatu informasi rahasia atau tanda pengenal ke dalam suatu file digital secara permanen tanpa memengaruhi konten aslinya. Sementara steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam suatu media tanpa meninggalkan jejak yang mencurigakan bagi orang lain. Salah satu metode steganografi yang umum digunakan adalah *Least Significant Bit* (LSB), di mana pesan disisipkan ke dalam bit paling tidak signifikan (LSB) dari piksel citra tanpa mengubah secara signifikan tampilan visual citra.

Dalam konteks ini, file citra sering digunakan sebagai media untuk menyimpan informasi penting atau rahasia, seperti hak cipta, data rahasia, atau tanda pengenal digital. Namun, file citra rentan terhadap serangan dan manipulasi oleh pihak yang tidak berwenang.

Dalam metode *watermarking*, keberadaan watermark yang disisipkan harus dapat terdeteksi dengan baik, bahkan setelah melalui transformasi atau serangan seperti kompresi, pemangkasan, atau filtrasi. Di sisi lain, dalam steganografi, pesan yang disembunyikan harus sulit dideteksi oleh pihak yang tidak berwenang, tetapi tetap dapat dipulihkan dengan akurasi tinggi oleh penerima yang sah.

Oleh karena itu, dengan adanya implementasi *fragile watermarking* dan steganografi LSB pada file citra, diharapkan dapat memenuhi kebutuhan akan perlindungan keamanan, keaslian, dan kerahasiaan citra digital.

II. TEORI DASAR

A. *Image Watermarking*

Image Watermarking merupakan teknik dalam menyisipkan suatu informasi rahasia atau tanda pengenal yang mengacu pada pemilik gambar ke dalam suatu file citra digital tanpa mengubah secara signifikan konten visualnya. untuk tujuan melindungi integritas, kepemilikan, *copyright*, ataupun menjaga keaslian konten serta memastikan bahwa citra tersebut tidak dapat dimanipulasi atau digunakan tanpa izin yang sah. Informasi yang disisipkan ini sering disebut sebagai *watermark*. *Watermark* ini dapat berupa teks, logo, gambar, atau data lainnya yang memiliki tanda pengenal atau pesan rahasia. Proses penyisipan *watermark* atau embedding dapat dilakukan dengan menggunakan beberapa metode umum seperti LSB (*Lease Significant Bit)*, substitusi, transformasi wavelet, atau transformasi domain lainnya. Proses penyisipan tersebut diusahakan untuk tidak merusak konten visual citra secara signifikan dan tetap menjaga kualitas citra.

Image watermark dapat diklasifikasikan menjadi dua jenis, yaitu:

1. *Visible watermarking*

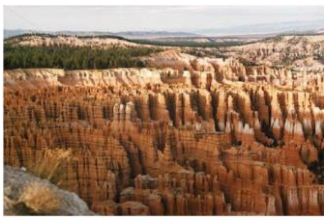
   Visible watermarking adalah metode *watermarking* yang menghasilkan tanda pengenal atau pesan yang terlihat secara jelas pada citra digital. Pada metode ini, pengguna dapat dengan mudah melihat *watermark* yang disisipkan pada citra.

Gambar 1. Contoh *Visible watermarking*

Metode ini umumnya digunakan untuk tujuan branding perlindungan hak cipta, atau sebagai bentuk tanda digital yang dikenali oleh semua orang yang melihat citra tersebut. Meskipun *watermark* yang disisipkan terlihat jelas, metode ini memiliki kelemahan yaitu dapat dengan mudah dihapus atau dimanipulasi oleh pihak yang tidak berwenang.

2. *Invisible watermarking*

   Invisible watermarking adalah menyisipkan tanda pengenal atau pesan ke dalam citra digital sehingga *watermark* tersebut tidak terlihat secara kasat mata. Dalam penerapannya, *watermark* disisipkan ke dalam citra sedemikian sehingga tidak mengganggu keonten visual secara signifikan atau terlihat oleh mata secara langsung.



Gambar 2. Contoh *Invisible watermarking*

Metode ini sering dimanfaatkan dalam aplikasi seperti perlindungan hak cipta, penelusuran plagiarisme, atau pemantauan dan keaslian konten digital. Berbeda dengan *visible watermarking*, metode ini memberikan tingkat keamanan yang lebih tinggi.

Berdasarkan tujuannya, *invisible watermarking* dapat diklasifikasikan menjadi dua jenis, yaitu:

1. *Fragile watermarking*

   Fragile watermarking bertujuan untuk menjaga integritas atau orisinalitas citra digital serta *tamper-proofing.* Dalam metode ini, *watermark* yang disisipkan sangat sensitif terhadap perubahan pada citra. *Watermark* tersebut menjadi rusak atau pecah jika dilakukan modifikasi, penghapusan, atau penyisipan data yang tidak sah pada citra.

Gambar 3. Contoh *Fragile watermarking*

2. *Robust watermarking*

   Robust watermarking bertujuan untuk menyisipkan label kepemilikan atau *copyright* sehingga dapat memberikan keamanan dan keandalan dalam melindungi atau mengidentifikasi citra digital. Pada metode ini, *watermarking* dirancang untuk tetap dapat diekstraksi dengan akurat bahkan setelah adanya manipulasi seperti kompresi, *cropping*, *editing*, atau serangan terhadap citra digital lainnya. Pada *robust watermarking*, umumnya menginginkan tercapainya ketiga karakteristik berikut:

   a) *Imperceptible* (Tidak Terlihat)

   Watermark yang disisipkan seharusnya dapat menjamin bahwa citra tetap memiliki tampilan yang baik dan tidak mengalami perbedaan yang signifkan setelah proses *watermarking*.

   b) *Robustness* (Ketahanan)

   Karakteristik ini mengacu pada ketahanan *watermarking* terhadap berbagai macam serangan atau manipulasi yang mungkin terjadi pada citra digital.

   c) *Secure* (Keamanan)

   Watermarking yang *secure* dirancang untuk mencegah atau mendeteksi usaha-usaha terhadap penghapusan atau manipulasi *watermark* pada citra digital.

B. Steganografi

Steganografi merupakan suatu ilmu dan seni menyembunyikan pesan rahasia dengan suatu cara sedemikian rupa sehingga tidak ada seorang pun dapat menyadari atau mencurigai keberadaan pesan rahasia tersebut. Keberadaan steganografi ini sudah ada sejak abad kelima sebelum masehi, dan seiring perkembangan zaman, implementasi steganografi pun juga semakin bervariasi pada berbagai media, seperti halnya steganografi pada media digital.

Steganografi digital sendiri adalah cabang dari steganografi yang menyembunyikan informasi atau pesan rahasia dalam format digital, seperti file gambar, audio, video, dokumen teks, atau file digital lainnya. Dokumen-dokumen digital yang

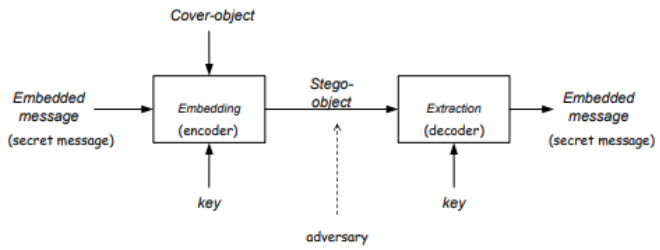
digunakan sebagai media dalam penyembunyian pesan tersebut dinamakan *carrier file*. Proses penyisipan pesan dilakukan dengan memanfaatkan karakteristik atau celah yang ada dalam *carrier file* sedemikian sehingga tidak mengganggu kualitas atau tampilan visual media tersebut. Steganografi tidak hanya menyembunyikan isi pesan, tetapi juga keberadaan pesan itu sendiri. Dengan kata lain, pihak yang berwenang tidak akan mengetahui adanya pesan tersembunyi dalam *carrier file*, sedangkan penerima yang berwenang dapat mengungkap isi pesan tersebut.

Terdapat beberapa istilah yang berkaitan dengan steganografi, diantaranya yaitu:

1. *Embedded message* atau *secret message* – pesan berupa teks, gambar, audio, maupun video yang disembunyikan.
2. *Cover-object* – pesan berupa teks, gambar, audio, maupun video yang digunakan untuk menyembunyikan *embedde message.*
3. *Stego-object* – pesan yang dihasilkan dari pengolahan antara *cover-object* dengan *embedded message*.
4. *Stego-key* – kunci yang digunakan untuk menyisipkan pesan dan mengekstraksi pesan dari *stego-object*.

Gambar 4. Contoh steganografi

Berikut ilustrasi alur proses steganografi yang umum dilakukan.



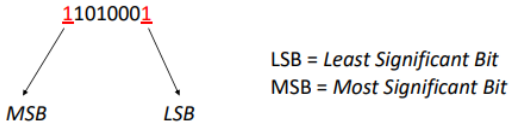
Gambar 5. Diagram alur proses steganografi

Steganografi dikatakan bagus jika dapat mencapai beberapa kriteria berikut:

1. *Imperceptible* – Keberadaan pesan rahasia tidak dapat dipersepsi secara visual maupun audial.
2. *Fidelity* – Kualitas *cover-object* tidak jauh berubah akibat penyisipan pesan rahasia, dimana citra digital masih terlihat dengan baik setelah penyisipan.
3. *Recovery* – Pesan yang disembunyikan harus dapat diekstraksi kembali.
4. *Capacity* – Ukuran pesan yang disembunyikan sedapat mungkin besar.

C. *Least Significant Bit*

Metode *Least Significant Bit* (LSB) merupakan salah satu metode yang umum digunakan pada steganografi dalam hal menyisipkan pesan tersembunyi ke dalam media digital, seperti gambar atau audio. Metode ini memanfaatkan bit paling tidak signifikan dari piksel untuk menyimpan bit-bit pesan tersembunyi. Misalkan dalam setiap *byte* bit-bitnya tersusun dari kiri ke kanan dalam urutan yang paling signifikan (*mmost significant bit* atau MSB) hingga kurang signifikan (*least significant bit* atau LSB), Pada citra digal sendiri terdiri dari sejumlah piksel, dan setiap piksel ini dapat direpresentasikan ke dalam bit-bit dengan panjang tertentu. Susunan bit pada setiap *byte* adalah sebagai berikut.



Gambar 6. Contoh susunan bit LSB dan MSB

Berdasarkan susunan tersebut, dengan mengubah nilai bit LSB, tentunya tidak akan mengubah persepsi dan kualitas citra secara signifikan. Hal ini dikarenakan, perubahan bit LSB hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Seperti contoh:

- *byte* 11010001 = 209
- *byte* 11010000 = 208

Misalkan 11010001 menyatakan warna merah, maka dengan mengubah LSB dari *byte* tersebut, hasilnya juga menyatakan warna merah dengan perubahan yang begitu sedikit. Perubahan tentu tidak dapat dibedakan oleh mata manusia.

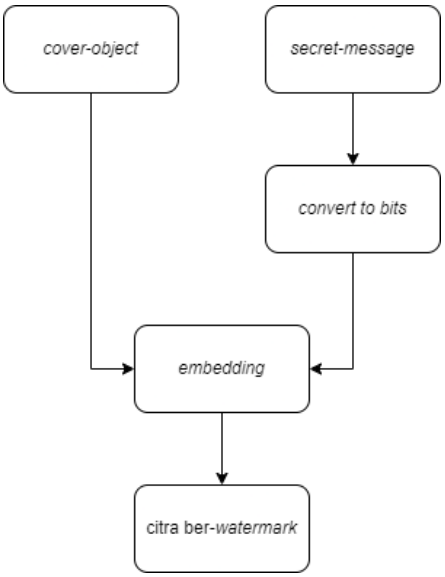
Gambar 7. Contoh perubahan bit LSB pada citra

III. Rancangan Solusi dan Implementasi

Rancangan implementasi dari penerapan *fragile watermarking* dan steganografi menggunakan metode *least significant bit* terdiri dari dua bagian, yaitu proses penyisipan pesan rahasia sebagai *watermark* dan proses ekstraksi.

a. Penyisipan pesan rahasia atau teks sebagai *watermark*

Berikut merupakan alur proses penyisipan *watermark* pada file citra digital.



Gambar 8. Diagram proses penyisipan *watermark* dan pesan rahasia

Proses *embedding* pesan atau *watermark* ke dalam citra dilakukan dengan menerima input pesan kemudian mengonversinya kedalam bits-bits yang bersesuaian. Hasil konversi tersebut harus sesuai dengan ukuran gambar. Setelah didapatkan *watermark* berupa pesan rahasia tersebut, tahap penyisipan dimulai dengan mengiterasi setiap piksel pada citra, kemudian menyisipkan bit-bit pesan tersebut kedalam *Least Significant Bit* (LSB) pada setiap warna (RGB).

b. Ekstraksi

Proses ekstraksi *watermark* cukup sederhana, yakni melakukan iterasi terhadap setiap piksel pada citra ber-*watermark*, kemudian memisahkan setiap bit pada *Least Significant Bit* (LSB) dan menggabungkannya serta mengonversinya kembali menjadi teks atau pesan rahasia seperti semula.

IV. Hasil dan Analisis

Berdasarkan rancangan solusi yang telah dijabarkan di atas, tahap implementasi terbagi menjadi dua bagian utama, yakni pembuatan fungsi *embedding* dan ekstraksi. Kode program di tulis dalam bahasa python dengan mekanisme interaksi menggunakan CLI atau *command-line interface*.

Berikut merupakan *source code* untuk masing-masing fungsi yang diimplementasikan.

1. *Embedding*

```
def embed_watermark(original_image_path, watermark_text, output_image_path):
  """Embeds a text watermark using LSB steganography"""
  # Open the original image
  image = Image.open(original_image_path)
  width, height = image.size

  # Convert the watermark text to binary
  watermark_bits = convert_text_to_bits(watermark_text)

  # Check if the watermark fits within the image
  max_watermark_length = width * height * 3  # Each pixel has 3 color channels (RGB)
  if len(watermark_bits) > max_watermark_length:
      raise ValueError("Watermark too large for the image")

  # Embed the watermark in the image
  pixel_index = 0
  for y in range(height):
      for x in range(width):
          # Get the pixel at the current coordinates
          pixel = list(image.getpixel((x, y)))

          # Embed the watermark bits into the least significant bits of each color channel
          for channel in range(3):  # 3 channels: R, G, B
              if pixel_index < len(watermark_bits):
                  pixel[channel] = (pixel[channel] & 0xFE) | watermark_bits[pixel_index]
                  pixel_index += 1

          # Update the pixel in the image
          image.putpixel((x, y), tuple(pixel))

  # Save the watermarked image
  image.save(output_image_path)
```

2. Ekstraksi

```
def extract_watermark(watermarked_image_path, watermark_length):
  """Extracts the compressed image watermark from a watermarked image"""
  image = Image.open(watermarked_image_path)
  width, height = image.size
```

```
  watermark_bits = []
  for y in range(height):
    for x in range(width):
        pixel = image.getpixel((x, y))  # Get
the color values at the current pixel

        if isinstance(pixel, int):  # Grayscale
image
            channel = pixel
            channel_bit = channel & 0x01  #
Extract the LSB of the channel

watermark_bits.append(channel_bit)
                else:  # RGB image
                    for channel in pixel:
                        channel_bit = channel &
0x01  # Extract the LSB of the channel

watermark_bits.append(channel_bit)


  watermark_text = ""
  char_bits = ""
  for i in range(0, len(watermark_bits), 8):
    char_bits = "".join(str(bit) for bit in
watermark_bits[i:i+8])
    char_value = chr(int(char_bits, 2))
    watermark_text += char_value

  return watermark_text[:watermark_length]
```

Kedua *source code* tersebut merupakan fungsi utama pada implementasi kali ini. Untuk melakukan penyisipan pesan pada citra, pertama-tama program memanggil fungsi untuk mengonversi pesan tersebut menjadi pesan berbentuk bit. Kemudian, untuk setiap bit-bit hasil konversi tersebut diproses dan disisipkan kedalam setiap piksel pada citra original. Sementara itu, untuk melakukan ekstraksi pesan dari citra berisikan *watermark*, tahapan-tahapan yang ada pada proses penyisipan tersebut dilakukan dengan urutan yang sama dalam urutan terbalik hingga didapatkan pesan rahasia.

Untuk pengujian pertama, akan dilakukan penyisipan pesan pada sebuah citra kemudian melakukan ekstraksi terhadap hasil dari penyisipan tersebut. Berikut hasil pengujian untuk cobaan pertama.

```
Watermark operations:
1. Embed watermark text into image.
2. Extract watermark text from image.
.....................................................................
Input the operation to be performed: 1
Input original image path: cover_image.png
Input watermark text: This image belong to David
Watermark embedded successfully!
.............................
Embedding result saved in: watermarked_image.png
```

Gambar 9. Hasil *embedding watermark* ke dalam citra

Pada kasus di atas, pesan yang disisipkan pada gambar yakni “This image belong to David”. Setelah dilakukan penyisipan, didapat hasil berupa gambar ber-*watermark* berikut.

Gambar 10. Perbandingan gambar asli (kiri), dengan gambar yang telah disisipkan *watermark* (kanan)

Dapat dilihat bahwa kedua gambar tersebut tidak dapat dibedakan dengan mata telanjang. Penyisipan pesan berhasil dilakukan tanpa mengubah tampilan dan kualitas citra aslinya. Dan setelah diekstraksi, didapatkan hasil sebagai berikut.

```
Watermark operations:
1. Embed watermark text into image.
2. Extract watermark text from image.
.....................................................................
Input the operation to be performed: 2
Input image path to be extracted: watermarked_image.png
Extracted watermark: This image belong to David
```

Gambar 11. Hasil ekstraksi *watermark* pada gambar

Pesan yang disisipkan sebelumnya berhasil diekstraksi kembali dengan isi yang masih sama.

Selain itu, dilakukan pengujian lainnya yaitu mengekstraksi gambar ber-*watermark* yang telah di-*resize* yang semula berukuran 768 x 768 menjadi 512 x 512 piksel. Berikut pesan *watermark* yang didapat.

```
Watermark operations:
1. Embed watermark text into image.
2. Extract watermark text from image.
.....................................................................
Input the operation to be performed: 2
Input image path to be extracted: edited_image.png
Extracted watermark: ▼uY[õ]µÝxùs◄S_ý◄=µñwß_;
```

Gambar 12. Hasil ekstraksi *watermark* pada gambar yang di-*resize*

Pengujian lain juga dapat dilakukan dengan melakukan kompresi terhadap gambar yang sudah disisipkan *watermark*. Berikut hasil yang didapatkan.

```
PS C:\Users\akyas\Documents\Course\Semester 6\Kriptografi\Watermarking program> python robust_watermarking.py
Watermark operations:
1. Embed watermark text into image.
2. Extract watermark text from image.
.....................................................................
Input the operation to be performed: 2
Input image path to be extracted: compress_image.png
Extracted watermark: ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

Gambar 13. Hasil ekstraksi *watermark* pada gambar yang dikompres

Berdasarkan kedua pengujian tersebut, terbukti bahwa manupilasi gambar ber-*watermark* akan menyebabkan *watermark* menjadi rusak dan tidak berarti lagi. *Watermark* tersebut tidak dapat diekstraksi menjadi pesan awal.

KESIMPULAN DAN SARAN

Implementasi *fragile watermarking* dan steganografi *Least Significant Bit (LSB)* pada file citra memiliki tujuan utama yakni meningkatkan keamanan, keaslian, dan kerahasiaan citra digital. Melalui *fragile watermarking*, citra digital tersebut dapat dilindungi dengan menyisipkan *watermark* yang sensitif terhadap adanya manipulasi. Jika hal tersebut terjadi, maka *watermark* tersebut akan rusak atau hilang, sehingga dapat ditarik kesimpulan yakni adanya manipulasi citra yang tidak sah. Sementara itu, peran steganografi dengan menggunakan LSB yakni menyembunyikan pesan rahasia di dalam citra tanpa mengubah secara signifikan tampilan visualnya. Hasil penyisipan *watermark* dengan memanfaatkan steganografi menghasilkan gambar yang secara kasat mata tidak dapat dilihat perbedaannya dengan citra asli. Pesan tersebut disisipkan ke dalam bit paling tidak signifikan dari piksel citra. Hal ini menyebabkan citra tetap terjaga sekaligus dapat sambal menyembunyikan informasi rahasia di dalamnya.

Berdasarkan hasil pengujian terhadap implementasi yang telah dilakukan di atas. Program tersebut terbukti berhasil melakukan *fragile watermarking* dan steganografi *Least Significant Bit (LSB)* pada suatu citra digital. Pesan dapat dimasukkan ke dalam citra dengan memanfaatkan bit-bit dari piksel citra tersebut. Pesan juga berhasil diekstrak sehingga didapatkan pesan dengan isi yang sama. Dengan melakukan manipulasi seperti *resize* dan *compress* pada citra, *watermark* yang ada di dalamnya menjadi rusak dan tidak dapat diekstraksi kembali. Hal ini mungkin berlaku bagi percobaan manipulasi gambar yang lainnya, seperti *crop*, penambahan *brightness*, penambahan *noise*¸ pengubahan warna, dll.

Melalui hasil tersebut, implementasi *fragile watermarking* dan steganografi LSB pada file citra memiliki potensi dalam perlindungan hak cipta, yaitu mengidentifikasi pemilik hak cipta dan pencegahan penyalahgunaan citra. Namun, diperlukan eksplorasi metode dan algoritma yang lebih kompleks untuk memperkuat keamanan implementasi tersebut. Diperlukan juga uji coba dengan skala yang lebih besar untuk memberikan pemahaman dan skalabilitas yang lebih terkait kinerja dari metode ini.

REFERENSI

[1] Munir, Rinaldi, Slide Kuliah IF4020 Kriptografi: Digital Watermarking, 2023. Diakses pada 22 Mei 2023.

[2] Munir, Rinaldi, Slide Kuliah IF4020 Kriptografi: Steganografi (Bagian 1), 2023. Diakses pada 22 Mei 2023.

[3] S. Rashita dan D. Cita, “Implementasi Digital Watermarking Pada Citra Menggunakan Metode Least Significant Bit”, Jurnal Informatika dan Komputer Volume 21 No. 3, Desember 2016.