Penyembunyian Peta Pulau Raftel dalam Peta Pulau Skypea dalam Anime One Piece dengan Steganografi dan Algoritma Kunci Publik RSA 1024-bit

Fadhil Muhammad Rafi' – 13518079 (*Author*)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): fadhilrafi90@gmail.com

***Abstract*—Makalah ini mengajukan ide tentang penyembunyian peta Pulau Raftel yang mengandung harta karun Raja Bajak Laut di dalam Peta Pulau Skypea pada Anime One Piece. Penyembunyian ini dilakukan dengan menggunakan Steganografi Least Significant Bit. Untuk meningkatkan keamanan peta Pulau Raftel terlebih dahulu dienkripsi dengan menggunakan RSA. Hal ini bertujuan agar peta Pulau Raftel tetap digambar untuk kepentingan pihak-pihak tertentu yang dapat dipercayai oleh Raja Bajak Laut dengan konsekuensi orang-orang tersebut tidak bisa menjadi Raja Bajak Laut jika mengetahui kunci RSA serta metode ekstraksi dari peta tersebut.**

***Keywords*—Raftel One Piece, Steganografi, Least Significant Bit, RSA**

I. PENDAHULUAN

Pulau Raftel pada anime One Piece adalah pulau yang dicari-cari oleh semua orang terutama bajak laut karena di dalamnya tersimpan harta karun yang ditemukan oleh Raja Bajak Laut pada dua puluh tahun lalu. Namun, Raja Bajak Laut tidak membuat peta untuk menuju Pulau Raftel karena harta tersebut tidak boleh ditemukan oleh sembarang orang. Hal ini membuat kru-kru Raja Bajak Laut itu tidak dapat kembali ke pulau tersebut untuk mengambil barang mereka yang tertinggal.

Masalah tersebut memunculkan ide bahwa peta Pulau Raftel tetap dapat digambar mengingat sudah ada banyak algoritma kriptografi dan steganografi yang sudah diketahui. Jika peta menuju Pulau Raftel tetap digambar kemudian peta tersebut disembunyikan pada suatu peta pulau yang lain, jalan menuju Pulau Raftel akan tetap menjadi rahasia bagi publik, namun pihak-pihak yang memiliki kepentingan untuk menuju pulau tersebut.

Raja Bajak Laut dapat melakukan penyembunyian peta Pulau Raftel pada peta Pulau Skypea−pulau lain pada anime One Piece. Raja Bajak Laut dapat menggunakan metode Steganografi Least Significant Bit untuk menyembunyikan peta tersebut sehingga publik tidak akan mengetahui bahwa peta Pulau Skypea yang ada selama ini mengandung peta untuk menuju ke Pulau Raftel dimana harta karun Raja Bajak Laut berada.

Untuk menambahkan lapisan keamanan, sebelum peta Pulau Raftel disembunyikan di dalam peta Pulau Skypea menggunakan Steganografi, gambar peta Pulau Raftel dalam bentuk biner dapat dienkripsi terlebih dahulu dengan menggunakan algoritma RSA dengan kunci rahasia yang dibangkitkan oleh Raja Bajak Laut itu sendiri. Hasil biner dari enkripsi ini selanjutnya akan diselipkan pada pada berkas biner peta Pulau Skypea.

Rumusan dari enkripsi dan penyembunyian peta Pulau Raftel ini dapat diberikan oleh Raja Bajak Laut kepada orang-orang yang dipilihnya terutama kru-kru bajak lautnya dengan konsekuensi orang-orang tersebut tidak dapat menjadi Raja Bajak Laut selanjutnya. Orang-orang ini nantinya tidak dibolehkan menyebarkan rumusan enkripsi dan penyembunyian ini ke orang lain hingga Raja Bajak Laut selanjutnya ditemukan.

II. LANDASAN TEORI

A. Steganografi

Steganografi berasal dari bahasa Yunani, terdiri dari *steganos* yang berarti tersembunyi dan *graphien* yang berarti tulisan sehingga secara bahasa steganografi berarti 'tulisan tersembunyi. Secara harfiah, steganografi adalah suatu teknik penyembunyian data rahasia ke dalam media data lain sehingga keberadaan data rahasia tidak diketahui atau disadari oleh orang lain. Dalam merancang suatu sistem steganografi, ada beberapa faktor yang harus diperhatikan seperti faktor *imperceptibility*, *fidelity*, *capacity*, *robustness*, *recovery*, dan *undetecability*. Suatu sistem steganografi akan dianggap memiliki kinerja yang sangat baik jika dapat memenuhi semua faktor tersebut dengan tingkat atau level yang tinggi. Akan tetapi dalam implementasinya hal tersebut akan sulit diwujudkan karena beberapa faktor tersebut memiliki sifat berkompetisi (*trade-off factor*) satu sama lain. Saat salah satu faktor ditingkatkan maka faktor yang lain akan mengalami penurunan, sehingga suatu sistem steganografi akan memiliki kelebihan dan kekurangannya masing-masing.

Terdapat beberapa istilah dalam dunia steganografi, antara lain, *embedded message* atau *secret message*—pesan yang

disembunyikan dalam steganografi, bisa berupa teks, gambar, audio, atau video, *cover object*—media digital yang digunakan untuk menyembunyikan *embedded message*, bisa berupa teks, gambar, audio, atau video, *stego object* atau *stego data*—*cover object* yang didalamnya sudah diselipkan *embedded message*, *stego key*—kunci yang digunakan untuk menyisipan *embedded message* ke dalam *cover object* dan mengekstraksi *embedded message* dari *stego object.*

Beberapa metode steganografi sudah banyak dipublikasikan dalam beberapa tahun terakhir ini mulai dari yang paling sederhana sampai yang sangat rumit dengan menggunakan gabungan berbagai persamaan matematika. Metode yang paling sederhana adalah teknik penyisipan Least Significant Bit (LSB) yang bekerja pada ranah spasial (Johnson and Jajodia,1998). Teknik dalam ranah frekuensi memanfaatkan DCT yang diusulkan oleh Barni et al (1998) yang menggunakan teknik seperti dalam algoritma JPEG (Wallace, 1991). Kemudian teknik yang diusulkan oleh Zhao et al (2004) yang menggunakan ranah wavelet dan memanfaatkan sistem chaos yang disebut "logistic map". Dengan menggabungkan dan memodifikasi beberapa teknik yang telah dikembangkan saat ini, diharapkan didapatkan suatu sistem steganografi yang memiliki kapasitas penyimpanan yang besar dan ketahanan yang tinggi.

B. Least Significant Bit

Least Significant Bit merupakan suatu metode dalam steganografi yang paling populer. Metode ini memanfaatkan kelemahan indra visual manusia dalam mengamati sedikit perubahan warna yang terjadi dalam *byte-byte* yang terdapat pada gambar. Setiap gambar *grayscale*, terdiri atas piksel-piksel yang mana 1x1 piksel dapat direpresentasikan sebagai 1 *byte* atau 8 bit pesan. Least Significant Bit mengubah nilai *n bytes* awal pada suatu gambar—*n* adalah jumlah pesan yang ingin disisipkan—satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Dengan kata lain, metode ini mengubah bit terakhir dalam *byte* menjadi satu bit pesan yang ingin disisipkan. Perubahan yang kecil ini tidak berpengaruh terhadap persepsi indra penglihatan manusia. Sedangkan untuk gambar RGB, 1x1 piksel direpresentasikan sebagai 3 *bytes (R, G,* dan *B)* sehingga setiap pikselnya dapat menyimpan 3 bits pesan. Misalkan terdapat 1x1 piksel gambar RGB yang akan menjadi *cover object* dengan representasi bit sebagai berikut.

10000010 01111011 01110101

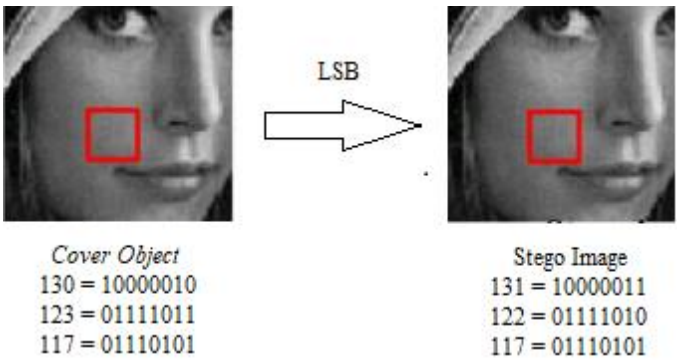
Akan disisipkan suatu pesan dengan representasi bit sebagai berikut.

101

Dengan menggunakan metode Least Significant Bit akan dihasilkan suatu *stego image* sebagai berikut.

10000011 01111010 01110101

Dapat dilihat bahwa pergeseran warna yang hanya sebesar satu atau bahkan tidak terjadi perubahan warna ini jika digabungkan dengan piksel-piksel yang lain tidak akan dapat diperhatikan dengan jelas oleh mata manusia. Dengan pemilihan *cover object* yang baik, mata manusia bahkan tidak dapat memperhatikan sedikitpun perbedaan yang terdapat antara *cover object* dan *stego image*.



Gambar 1. Penyembunyian pesan dengan LSB

Untuk ekstraksi *embedded message* pada *stego image* sebenarnya dapat dilakukan dengan mudah. Misalkan terdapat suatu *stego image* sebagai berikut.

00110011 10100010 11100010 10101011 00100110

10010110 11001001 11111001 10001000 10100011

Ekstraksi pesan dapat dilakukan dengan mengambil bit terakhir pada setiap *byte stego image*. Dengan demikian, *embedded message* yang terdapat pada *stego image* tersebut adalah sebagai berikut.

1001001101

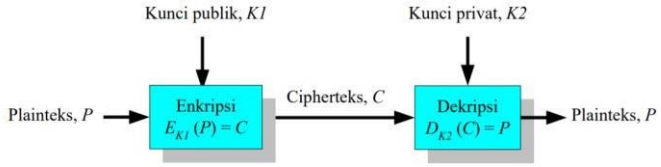
Apabila seseorang paham dengan Least Significant Bit curiga dengan *stego image* yang dilihatnya, dia dapat mencoba melakukan ekstraksi pesan dari setiap *bytes* dalam *stego image* dan mendapatkan *embedded message* yang disembunyikan di dalamnya dengan mudah. Oleh karena itu, penyembunyi pesan harus teliti dalam memilih gambar yang akan digunakan sebagai *cover object* untuk menghindari kecurigaan orang lain yang akan melihat *stego image* yang dibuatnya.

Embedded message yang dapat disisipkan kedalam *cover object* paling besar memiliki jumlah *bytes* sebanyak 1/8 kali jumlah *bytes* dari *cover object*. Hal ini disebabkan oleh setiap 1 bit *embedded message* yang disisipkan ke setiap 1 bytes *cover object*.

C. Kriptografi Kunci Nirsimetri

Kriptografi kunci nirsimetri adalah suatu algoritma kriptografi dimana kunci yang digunakan untuk melakukan enkripsi suatu pesan berbeda dengan kunci yang digunakan untuk melakukan dekripsi dari sandi hasil enkripsi pesan tersebut. Ide dari kriptografi kunci nirsimetri ini muncul pada tahun 1976 oleh Whitfield Diffie dan Martin E. Hellman, keduanya merupakan ilmuwan asal Stanford University. Kriptografi ini pada implementasinya membutuhkan kunci

publik dan kunci privat dimana kunci publik umumnya digunakan untuk mengenkripsi pesan sedangkan kunci privat digunakan untuk mendekripsi pesan. Kriptografi kunci nirsimetri ini disebut juga dengan kriptografi kunci publik. Hal ini disebabkan kunci untuk enkripsi diumumkan kepada publik (tidak rahasia), hanya kunci private yang bersifat rahasia dan hanya diketahui oleh pihak yang melakukan enkripsi terhadap pesan tersebut.



Gambar 2. Alur kriptografi kunci nirsimetri

Ide kriptografi kunci nirsimetri ini dapat dianalogikan seperti mengirim surat menggunakan kotak yang dapat dikunci dengan suatu gembok. Misalkan A dan B akan saling mengirim surat dengan kriptografi kunci nirsimetri sebagai pengaman. A akan mengirim kotak surat dengan gembok dalam keadaan gembok terbuka kepada. Bob memasukkan surat ke dalam kotak surat, lalu mengunci kotak tersebut dengan gembok milik A. Dalam hal ini, surat terdapat di dalam kotak surat dan kotak digembok dengan gembok A. Kemudian, B mengirim kembali surat tersebut kepada A. Selanjutnya, A menerima kotak surat tersebut dan membukanya dengan kunci gembok milik A.

Keuntungan yang diperoleh dari kriptografi kunci nirsimetri, yaitu:

1. Tidak diperlukan pengiriman kunci privat.
2. Jumlah kunci dapat ditekan karena setiap orang yang perlu memiliki sepasang kunci publik dan privat saja.

D. Algoritma RSA

RSA merupakan algoritma kunci-publik yang paling terkenal dan paling banyak aplikasinya dalam enkripsi data sehari-hari. Kata RSA ini diambil dari nama dari tiga peneliti dari MIT yang mengembangkannya, yaitu Ronald Rivest, Adi Shamir, dan Len Adleman pada tahun 1976. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan bulat yang besar menjadi faktor-faktor prima.

Komponen-komponen penting dalam algoritma RSA, antara lain:

1. *p* dan *q* berupa bilangan prima (rahasia)
2. $n = p.q$ (bits dari RSA) (tidak rahasia)
3. $\Phi(n) = (p-1)(q-1)$ (rahasia)
4. *e*, dimana $PBB(e, \Phi(n)) = 1$ (tidak rahasia)
5. *d*, dimana $d \equiv e^{-1} \pmod{\Phi(n)}$ (rahasia)
6. *m* (plainteks) (rahasia)
7. *c* (cipherteks) (tidak rahasia)

Dari tujuh komponen di atas, kunci publik (*pub*) dan kunci privat (*pri*) dari algoritma RSA adalah pasangan dari:

$$pub = (e, n) \quad (1)$$

$$pri = (d, n) \quad (2)$$

Langkah-langkah untuk melakukan enkripsi pesan dengan menggunakan algoritma RSA, yaitu:

1. Nyatakan pesan menjadi blok-blok plainteks: m_1, m_2, m_3, m_4, … (dimana: $0 \le m_i < n - 1$).
2. Untuk setiap blok plainteks, hitunglah blok cipherteks c_i untuk blok plainteks m_i menggunakan kunci publik *e* dengan persamaan berikut.

$$c_i = m_i^{\,e} \bmod n \quad (3)$$

3. Gabungkan blok-blok cipherteks tersebut secara terurutsehingga menjadi satu cipherteks utuh.

Langkah-langkah untuk melakukan dekripsi pesan dengan menggunakan algoritma RSA, yaitu:

1. Nyatakan cipherteks menjadi blok-blok cipherteks: c_1, c_2, c_3, c_4, … (dimana: $0 \le c_i < n - 1$).
2. Untuk setiap blok cipherteks, hitunglah blok plainteks m_i untuk blok cipherteks c_i menggunakan kunci privat *d* dengan persamaan berikut ini.

$$m_i = c_i^{\,d} \bmod n \quad (4)$$

3. Gabungkan blok-blok plainteks tersebut secara terurut sehingga menjadi satu kesatuan plainteks yang utuh.

III. IMPLEMENTASI

A. Implementasi Pembangkitan Kunci RSA

Pembangkitkan kunci publik pada algoritma RSA dapat langsung dilakukan dengan mengambil pasangan *e* dan *n* sehingga diperoleh kunci publik seperti pada II.D nomor (1)

Sedangkan untuk pembangkitan kunci privat, terlebih dahulu harus dicari nilai *d* yang merupakan modulo invers e terhadap nilai *Φ*. Implementasi modulo invers pada bahasa pemrograman python adalah sebagai berikut.

```
def mod_inv(a,m):
  m0 = m
  x, y = 0, 1

  if (m == 1):
      return 0

  while (a > 1):
```

```
        q = a // m
        t = m
        m = a % m
        a = t

        t = y


        y = x - q * y
        x = t

    if (x < 0):
        x = x + m0

    return x
```

Dengan algoritma di atas, dapat diperoleh pasangan kunci privat untuk algoritma RSA adalah sebagai berikut.

$$pri = (mod\_inv(e, \Phi), n) \quad (5)$$

B. Encoding Peta dalam Bits

Penyisipan peta Pulau Raftel ke dalam peta Pulau Skypea, tidak dapat dilakukan secara langsung, melainkan harus dilakukan *encoding* untuk merubah bentuk berkas agar dapat dapat disisipkan antara peta yang satu dengan peta yang lainnya. Perlu dicatat pula, peta yang menjadi *cover object* yang mana merupakan peta Pulau Skypea harus memiliki jumlah bytes delapan kali lebih besar dibandingkan *embedded image* yang mana adalah peta Pulau Raftel.

Dalam melakukan encoding gambar peta, terlebih dahulu peta diubah ke dalam format base64 kemudian kemudian string base64 tersebut dilakukan encoding kembali untuk diubah menjadi bentuk biner. Berikut merupakan implementasi python untuk encoding suatu berkas gambar ke dalam bentuk string base64 dan encoding suatu string base64 menjadi biner yang secara berturut-turut ditunjukkan oleh fungsi img_to_base64 dan base64_to_bin secara berturut-turut.

```
import base64

def img_to_base64(img):
    enc_base64 = base64.b64encode(img.read())


    return enc_base64
```

```
def base64_to_bin(base64):
  codes = [None for i in range(len(base64))]
  i, j = 0, 0

  while (i < len(base64)):
    code = ord(base64[i])
    i += 1

    if (
         code >= int('0xD800', 16) and
         code <= int('0xDBFF', 16) and
         i < len(base64)
    ):
      next_code = ord(base64[i])
      i += 1

      if (
           (next_code & int('0xFC00', 16)) ==
          int('0xDC00', 16)
         ):
        codes[j] = (
            (code & int('0x3FF', 16)) << 10) +
            (next_code & int('0x3FF', 16)) +
            int('0x10000', 16)
        j += 1
      else:
        codes[j] = code
        j += 1
        i -= 1
    else:
      codes[j] = code
      j += 1

  codes = codes[:j]
  bytes = [None for i in range(len(codes) * 4)]
  k = 0

  for code in codes:
    if (code <= int('0x7F', 16)):
      bytes[k] = code
      k += 1
    elif (code <= int('0x7FF', 16)):
      bytes[k] = b'11000000' | (code >> 6)
      bytes[k] = b'10000000' |
        (code & int('0x3F', 16))
      k += 2
    elif (code <= int('0xFFFF', 16)):
      bytes[k] = b'11100000' | (code >> 12)
      bytes[k] = b'10000000' |
        ((code & int('0xFFF', 16)) >> 6)
      bytes[k] = b'10000000' |
        (code & int('0x3F', 16))
      k += 3
    else:
      bytes[k] = b'11110000' |
        (code >> 18)
      bytes[k] = b'10000000' |
        ((code & int('0x3FFFF', 16)) >> 12)
      bytes[k] = b'10000000' |
        ((code & int('0xFFF', 16)) >> 6)
      bytes[k] = b'10000000' |
        (code & int('0x3F', 16))
      k += 4

  return bytes[:k]
```

C. Implementasi Enkripsi dan Dekripsi Peta

Untuk peta Pulau Raftel yang menjadi *embedded image*, sebelum dilakukan pengubahan ke dalam bentuk biner, peta Pulau Raftel dalam bentuk base64 dienkripsi terlebih dahulu menggunakan algoritma RSA. Enkripsi base64 peta akan dilakukan untuk setiap 100 bytes, hal ini dilakukan untuk

mempersulit didekripsinya berkas peta jika terdapat pihak yang berhasil mengambil *embedded image* yang disisipkan dalam *cover object.* Untuk implementasi ini, hanya dibatasi untuk RSA 1024-bit sehingga nilai *p* dan *q* yang akan digunakan haruslah sebesar 512 bit. Pada penggunaan RSA 1024-bit, untuk setiap 100 karakter pasti akan dipetakan menjadi 104 *bytes.* Implementasi algoritma enkripsi dan dekripsi RSA adalah sebagai berikut.

```
# power_mod(a,b,c) = (a^b) mod c
def encrypt_RSA(plaintext, pub_key):
  enc = []
  for char in plaintext:
     enc.append(power_mod(ord(char), pub_key[0], pub_key[1]))

  return ``.join(enc)

def encrypt_base64(base64):
  length = len(base64) // 100
  enc = ``
  for i in range(length):
     bytes = base64[100*i : 100*(i+1)]
     enc += encrypt_RSA(bytes)

  return enc
```

```
def decrypt_RSA(ciphertext, pri_key):
  dec = ''
  for code in ciphertext:
      dec  +=  chr(power_mod(code,  pri_key[0], pri_key[1]))

  return dec

def encrypt_base64(cipher):
  length = len(cipher) // 104
  dec = ``
  for i in range(length):
     bytes = base64[104*i : 104*(i+1)]
     dec += decrypt_RSA(bytes)

  return dec
```

D. Implementasi Least Significant Bit

Sebagaimana dibahas pada bagian II.B, implementasi Least Significant mengubah bit terakhir dari setiap bytes dari *cover object* bersesuaian dengan bit *embedded message.* Implementasi dari Least Significant Bit menggunakan python adalah sebagai berikut.

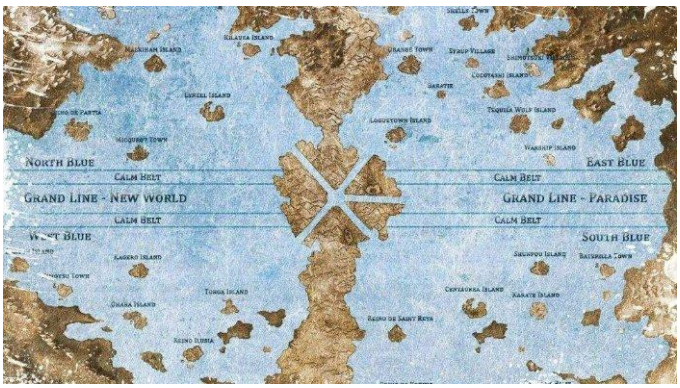
```
def LSB(cover_obj, emb_img):
   i = 0
   stego_img = []
   for bytes in cover_obj:
      new_bytes=set_last_bit(bytes, emb_img[i])
      stego_image.append(new_bytes)

  return stego_image
```

IV. PENGUJIAN

A. Enkripsi Peta Pulau Raftel

Enkripsi peta Pulau Raftel dilakukan sebagai lapisan kedua Pengaman agar ketika nilai biner dari peta Pulau Raftel dapat diambil, pihak yang berupaya melakukan steganalisis tidak mampu langsung mendapatkan gambar peta Pulau Skypea. Berikut merupakan pengandaian gambar peta Pulau Raftel—peta Pulau Raftel yang sebenarnya tidak ada.



Gambar 3. Pengandaian peta Pulau Raftel

Gambar 4. Peta Pulau Skypea

Terlebih dahulu akan dibangkitkan pasangan kunci publik dan kunci privat dengan menggunakan algoritma RSA. Dengan memilih nilai *p* dan *q* sebesar 512 bit—agar *n* yang dihasilkan sebesar 1024 bit—secara berturut-turut bernilai sebagai berikut.

```
// nilai p
10449063873256633832537583607799732666830833461 4
619683235822684691889114777018590629011853749403
```

```
74207341620762128103426782085092480950430126381851494218593
```

```
// nilai q
955840999003608048752058539559665234004148317436500935344076871608303040545496634169874722640662621420940394604960838808790279757226932720277464702223703 9
```

Dengan pemilihan nilai *e* sebesar 65537 serta dengan diikuti nilai *p* dan *q* di atas, dihasilkan kunci publik dan kunci privat secara berturut-turut sebagai berikut.

```
// kunci publik
(65537,998764365126613099773800618819858321756412477012477514492261778482803434773874441353358776417054117981492682731206119202668317758960215463272695337660889314050066343731432059162569812365590941559897334725385464350249941627031973464104696725537384635852383398999854372697543149907786402730636896980818127066127)
```

```
// kunci privat
(198679997988245647218381046852228404424040610080039513013353801071197701931275629327515505462431074984317271887137674297753461193457791628345229277672919682993313421041482393869733638483839322904922344864510478567049560492746007192273087178465532407153672980461721004017416052812257001170840429944253591639693,998764365126613099773800618819858321756412477012775144922617784828034347738744413533587764170541179814926827312061192026683177589602154632726953376608893140500663437314320591625698123655909415598973347253854643502499416270319734641046967255373846358523833989998543726975431499077864027306368969808181270661 27)
```

Dengan menggunakan fungsi img_to_base64 diikuti dengan fungsi decrypt_base64, diperoleh potongan base64 peta Pulau Raftel yang telah dienkripsi adalah sebagai berikut.

```
57dbf457d6cb69f3dd901cd47b09cb6ae3db69e3cb79e3cc69e3ce57d4c36ae6cb69f7ce6ae3db6de3c37de7cc69f3d06af3df6c04c39c0bcc9c25d26b198474e5c390004f79dadd6e0acf78e7dc6df6d09119e26e1adb89e8dc6df4d16807e36c2aba8ceac46108cfa105f372e5d38decc c9c27d2a0d9b95b19cc6ce3db7de8bf
.
.
.
sSvdiV7sSL1WLEqcUXVYKGwKStSjThEr3YlaiJi7FaTCNMGjEr3Yle7FQlKFWrT/ANAP/9k=
```

B. Encoding Peta Pulau Skypea dan Encrypted Peta Pulau Raftel

Terlebih dahulu, sudah dipastikan bahwa, gambar peta Pulau Skypea memiliki jumlah bytes lebih dari delapan kali jumlah bytes pada gambar peta Pulau Raftel. Selanjutnya, dengan menggunakan fungsi img_to_base64 dan base64_to_bin kepada peta Pulau Skypea dan dengan menggunakan fungsi base64_to_bin kepada berkas base64 peta Pulau Raftel yang telah dienkripsi diperoleh berkas biner peta Pulau Skypea dan peta Pulau Raftel yang telah dienkripsi sebagai berikut.

```
// Berkas biner peta Pulau Skypea
01001100  01111010  01101100  01110001  01001100
01111010  01010010  01000010  01010001  01010110
01000110  01010100  01100001  00110001  01110000
01001011  01010101  01101101  01100100  01000010
01010001  01101011  01000110  01010010  01010001
01010101  01000110  01000010  01010101  01010101
01000110  01000011  01010001  01010101  01000110
01000101  01001100  01111010  01001010  00110011
01010001  01101011  01010010  01000010  01010001
01010110  01010110  01000101  01010001  01101011
01000110  01010010  01010010  01010101  01000110
00110011  01010110  01010101  01010110  01000011
01010001  01010110  01000110  01000111  01010001
01101100  01000110  01010110  01010010  00110000
01001010  00110011  01100100  00110000  01101100
01000011  01100100  00110010  01001110  01001001
01010001  01101110  01100011  00110100  01010100
01000101  01001110  00110011  01100001  00110000
.
.
.
01010101  00110101  01000010  01010101  01000011
00111000  00110101  01100001  01111010  00110000
```

```
// Berkas biner peta Pulau Raftel yang telah
// dienkripsi
00110101  00110111  01100100  01100010  01100110
00110100  00110101  00110111  01100100  00110110
01100011  01100010  00110110  00111001  01100110
00110011  01100100  01100100  00111001  00110000
00110001  01100011  01100100  00110100  00110111
01100010  00110000  00111001  01100011  01100010
00110110  01100001  01100101  00110011  01100100
01100010  00110110  00111001  01100101  00110011
01100011  01100010  00110111  00111001  01100101
00110011  01100011  01100011  00110110  00001010
00111001  01100101  00110011  01100011  01100101
00110101  00110111  01100100  00110100  01100011
.
.
.
00110111  01000110  01010001  01101100  01001011
01000110  01010111  01110010  01010100  00101111
01000001  01001110  01000001  01010000  00101111
```

C. Penyisipan Berkas Biner Peta Pulau Raftel Terenkripsi ke dalam Peta Pulau Skypea

Dengan menggunakan fungsi LSB, diperoleh berkas biner *stego image* beserta *stego image* dalam bentuk gambar sebagai berikut.

```
01001100  01111010  01101101  01110001  01001100
01111011  01010010  01000011  01010000  01010110
01000111  01010101  01100000  00110001  01110001
01001011  01010100  01101101  01100101  01000010

.

.

.

01010100  00110100  01000010  01010100  01000011
00111000  00110101  01100001  01111011  00110001
```

Gambar 5. Stego image peta Pulau Raftel yang disembunyikan di dalam peta Pulau Skypea

V. KESIMPULAN

Penyembunyian peta Pulau Raftel pada peta Pulau Skypea sangat baik diimplementasikan dengan menggunakan steganografi dengan tambahan enkripsi RSA 1024-bit untuk berkas peta Pulau Raftel. Dengan adanya tambahan enkripsi RSA 1024-bit akan sangat sulit menemukan gambar peta Pulau Raftel yang asli walaupun *embedded image* sudah diekstrak dari *stego image* berupa peta Pulau Skypea. Oleh karena itu, peta Pulau Raftel dapat digambar oleh Raja Bajak Laut tanpa mengkhawatirkan kerahasiaan dari peta tersebut terhadap publik.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur kepada rahmat Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga makalah berjudul “Penyembunyian Peta Pulau Raftel dalam Peta Pulau Skypea dalam Anime One Piece dengan Steganografi dan Algoritma Kunci Publik RSA 1024-bit” dapat diselesaikan dengan baik. Penulis mengucapkan terima kasih kepada selaku dosen pengajar IF4020 Kriptografi, Dr. Rinaldi Munir, S.T, M.T. yang telah memberikan bimbingan dan ilmu terkait materi Kriptografi ini, khususnya pada algoritma kriptografi kunci publik dan steganografi beserta dengan pengaplikasiannya. Penulis juga mengucapkan terima kasih kepada teman-teman yang telah memberikan dukungan kepada penulis untuk menuliskan makalah ini. Tak lupa penulis juga mengucapkan terima kasih kepada para penulis sumber referensi yang telah memberikan pengetahuan yang dibutuhkan penulis untuk menyelesaikan makalah ini.

REFERENSI

[1] Munir, Rinaldi. 2021. Slide Kuliah Kriptografi (Bandung: Institut Teknologi Bandung)

[2] Abid, Yahya. 2019. Steganography Techniques for Digital Images (Palaplye: Springer)

[3] R.L. Rivest, A. Shamir, and L. Adleman. 1976. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems (Massasuchet: Massasuchet Institute of Technology)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Desember 2021

Fadhil Muhammad Rafi’ 13518079