Aplikasi Algoritma RSA dalam Enkripsi dan Dekripsi Gambar

Ikmal Alfaozi - 13520125
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13520125@std.stei.itb.ac.id

***Abstract*—Pada era sekarang, keamanan pengiriman data melalui internet menjadi sangat penting untuk mencegah orang-orang yang tidak bertanggung jawab mengakses data tersebut. Enkripsi gambar merupakan salah satu teknik enkripsi yang bertujuan untuk menyembunyikan informasi yang ada pada gambar berdasarkan beberapa nilai kunci tertentu. Algoritma RSA merupakan algoritma kriptografi asimetri, yaitu kunci yang digunakan untuk mengenkripsi berbeda dengan kunci yang digunakan untuk mendekripsi. Algoritma RSA ini bisa digunakan untuk enkripsi gambar RGB. Hasil percobaan menunjukkan gambar berhasil dienkripsi dan didekripsi menggunakan algoritma ini. Gambar yang sudah dienkripsi dengan algoritma RSA menghasilkan gambar yang sangat berbeda dengan gambar aslinya.**

***Keywords*—RSA Algortithm, Image Encryption, Cryptography, Key Generation.**

I. PENDAHULUAN

Pada zaman sekarang, internet sangat membantu kita dalam pengiriman data dari satu tempat ke tempat lain. Namun, di samping kemudahan tersebut, data yang kita kirim melalui internet juga rentan terhadap penyadapan oleh pihak yang tidak bertanggung jawab. Salah satu cara untuk melindungi data tersebut adalah dengan mengenkripsinya supaya tidak bisa dibaca oleh pihak lain. Enkripsi merupakan salah satu teknik yang digunakan untuk mengamankan informasi data. Semantara itu, enkripsi gambar adalah teknik enkripsi yang mengubah informasi/format pada gambar ke dalam bentuk informasi/format gambar lain yang sulit dimengerti atau dibaca pihak lain.

Berbeda dengan teks, gambar memiliki karakteristik tersendiri yang membuatnya tidak mudah dienkripsi. Gambar terdiri dari kumpulan piksel yang setiap pikselnya terdiri dari warna-warna tertentu. Pada gambar RGB, setiap piksel pada gambar memuat tiga warna utama yaitu merah, hijau, dan biru. Masing-masing warna tersebut nantinya diproses menjadi sebuah matriks yang kemudian dimanipulasi isinya sehingga menjadi *cipher image* (gambar yang sudah dienkripsi). Untuk mengembalikan c*ipher image* tersebut ke gambar awal dibutuhkan *key* (kunci) yang didapat dari proses enkripsi awal.

RSA merupakan algoritma kriptografi asimetri,yaitu kunci yang digunakan untuk enkripsi berbeda dengan kunci yang digunakan untuk dekripsi. Algoritma RSA ini dikembangkan pada tahun 1976 oleh Ron Rivest, Adi Shamir, dan Leonard Adleman yang juga dikenal sebagai kriptografi kunci-public (*public-key cryptograohy*). RSA biasanya digunakan untuk transmisi data yang aman. Teknik ini mengenkripsi informasi yang akan dikirimkan menjadi bentuk yang tidak dapat dipahami sehingga hanya orang yang mengetahui kunci dekripsinya yang dapat memulihkan informasi data tersebut. Pada makalah ini, algoritma RSA dimodifikasi sedemikian rupa sehingga dapat digunakan untuk enkripsi dan dekripsi gambar RGB.

II. TEORI DASAR

A. Bilangan Bulat

Bilangan bulat adalah bilangan yang tidak mempunyai pecahan desimal, misalnya 8, 21, 8765, -34, 0, dan sebagainya. Kebalikan dari bilangan bulat adalah bilangan riil yang mempunyai titik desimal, seperti 8.0, 34.25, 0.02, dan sebagainya.

B. Sifat Pembagian pada Bilangan Bulat

Misalkan a dan b adalah dua buah bilangan bulat dengan syarat $a \neq 0$ Kita menyatakan a habis membagi b jika terdapat bilangan bulat k sedemikian sehingga $b = ak$ atau dalam notasi ditulis sebagai $a \mid b$ jika $b = ac, c \in Z$ dan $a \neq 0$.

Teorema Euclidean: Misalkan m dan n adalah dua buah bilangan bulat dengan syarat $n > 0$. Jika m dibagi dengan n maka terdapat dua buah bilangan bulat unik q (*quotient*) dan r (*remainder*), sedemikian sehingga $m = nq + r$ dengan $0 \leq r < n$. Bilangan n disebut pembagi (*divisor*), m disebut yang dibagi (*dividend*), q disebut hasil bagi (*quotient*), dan r disebut sisa (*remainder*). Notasi yang digunakan untuk mengekspresikan hasil bagi dan sisa adalah dengan menggunakan operator mod dan div seperti berikut:

$$q = m \operatorname{div} n, \qquad r = m \bmod r.$$

C. Pembagi Bersama Terbesar (PBB)

Pembagi bersama terbesar (PBB – *greatest common divisor* atau *gcd*) dari a dan b adalah bilangan bulat terbesar x sehingga x habis membagi a dan x habis membagi b. Misalnya $\text{PBB}(6,9) = 3$ karena 3 merupakan bilangan terbesar yang habis membagi 6 dan 3 juga habis membagi 9.

D. Relatif Prima

Dua buah bilangan bulat a dan b dikatakan relatif prima jika PBB$(a, b) = 1$.

E. Aritmetika Modulo

Aritmetika modulo (*modular arithmetic*) merupakan salah satu bidang matematika aritmetika yang berhubungan dengan sisa pembagian. Operator dari aritmetika modulo adalah mod yang memberikan sisa pembagian. Misalkan a adalah bilangan bulat dan m adalah bilangan bulat > 0, operasi a mod m (dibaca "a modulo m") memberikan sisa dari a dibagi dengan m. Dengan kata lain, a mod $m = r$ sedemikian sehingga $a = mq + r$, dengan $0 \leq r < m$. Bilangan m disebut modulus atau modulo, dan hasil aritmetika modulo m terletak di dalam himpunan $\{0, 1, 2, \ldots, m-1\}$.

F. Kongruen

Dua buah bilangan bulat a dan b jika dibagi bilangan positif m mungkin akan menghasilkan sisa yang sama. Oleh karena itu, kita dapat mengatakan a dan b kongruen dalam modulo m, atau ditulis $a \equiv b \bmod m$ atau $b \equiv a \bmod m$. Jika dua buah bilangan bulat a dan b tidak kongruen dalam modulus m ditulis sebegai $a \not\equiv b \bmod m$. Misalkan a dan b adalah bilangan bulat dan m adalah bilangan > 0, maka $a \equiv b \bmod m$ jika m habis membagi $a - b$. Kekongruenan $a \equiv b \bmod m$ dapat dituliskan dalam hubungan $a = b + km$ dengan k adalah bilangan bulat.

G. Sifat-Sifat Perhitungan Arithmetika Modulo

Misalkan m adalah bilangan bulat positif.

1. Jika $a \equiv b \pmod m$ dan c adalah sembarang bilangan bulat maka
   (i) $(a + c) \equiv (b + c) \pmod m$
   (ii) $ac \equiv bc \pmod m$
   (iii) $a^p \equiv b^p \pmod m$, p bilangan bulat tak-negatif
2. Jika $a \equiv b \pmod m$ dan $c \equiv d \pmod m$, maka
   (i) $(a + c) \equiv (b + d) \pmod m$
   (ii) $ac \equiv bd \pmod m$

H. Balikan Modulo (Modulo Invers)

Jika a dan m relatif prima dan $m > 1$, kita dapat menemukan inversi dari a modulo m. Inversi dari a modulo m adalah bilangan bulat x sedemikian sehingga $ax \equiv 1$ mod m.

I. Kekongruenan Lanjar

Kekongruenan lanjar adalah kongruen yang berbentuk $ax \equiv b \bmod m$ dengan m adalah bilangan bulat positif, a dan b sembarang bilangan bulat, dan x adalah peubah. Bentuk $ax \equiv b \bmod m$ dapat ditulis sebagai $ax \equiv b + km$ atau $x = \frac{b+km}{a}$ dengan k adalah bilangan bulat. Dengan mencoba nilai-nilai $k = 0, 1, 2, \ldots$ dan $k = -1, -2, -3, \ldots$ ke dalam persaman $x = \frac{(b+km)}{a}$, kita dapat menemukan bilangan bulat x.

J. Bilangan Prima

Suatu bilangan bulat p ($p > 1$) disebut bilangan prima jika bilangan p tersebut hanya habis dibagi 1 dan dirinya sendiri. Contoh bilangan prima adalah 2, 3, 5, 7, dan seterusnya. Bilangan bulat positif yang lebih dari sama dengan 2 selain bilangan prima itu sendiri disebut bilangan komposit. Misalnya 10 adalah bilangan komposit karena 10 dapat dibagi oleh 2 dan 5, selain 1 dan 10 itu sendiri. Bilangan prima ini sangat berguna dalam perhitungan aritmetika modulo dengan perpangkatan yang besar. Berikut adalah teorema yang memanfaatkan bilangan prima dan konsep relatif prima.

Teorema Fermat: Jika p adalah bilangan prima dan a adalah bilangan bulat yang tidak habis dibagi dengan p, yaitu PBB$(a,p) = 1$, maka $a^{p-1} = 1 \pmod p$.

K. Kriptografi

Kriptografi adalah ilmu yang mempelajari cara untuk menjaga suatu informasi tetap aman dengan cara menyandikan informasi tersebut sehingga sulit dibaca atau dimengerti oleh pihak yang tidak berhak. Pesan yang dirahasiakan dinamakan plainteks, sedangkan pesan hasil penyandian disebut cipherteks. Untuk mengembalikan pesan yang sudah disandikan ke pesan semula, seseorang harus mengetahui metode penyandian dan kunci penyandian pesan tersebut sehingga hanya orang yang berhak yang bisa mengembalikannya ke keadaan awal. Proses menyandikan plainteks menjadi cipherteks disebut enkripsi, sedangkan proses membalikkan cipherteks menjadi plainteks disebut dekripsi. Berikut adalah proses enkripsi dan dekripsi:

Terdapat dua aplikasi dari kriptografi, yaitu pengiriman data melalui saluran komunikasi dan penyimpanan data di dalam *disk storage*. Sebelum dikirimkan lewat saluran komunikasi, data dienkripsi terlebih dahulu menjadi cipherteks. Kemudian, cipherteks tersebut dikembalikan ke plainteks di tempat penerima. Sementara itu untuk penyimpanan data di dalam *disk storage*, data disimpan dalam bentuk cipherteks sehingga hanya orang yang berhak yang dapat membalikkan cipherteks tersebut ke dalam plainteks.

Misalnya cipherteks dilambangkan dengan C dan plainteks dilambangkan dengan P. Notasi matematis yang biasa digunakan untuk operasi enkripsi dan dekripsi adalah sebagai berikut:

Fungsi enkripsi E memetakan P ke C,

$$E(P) = C$$

Fungsi dekripsi D memetakan C ke P,

$$D(C) = P$$

Algoritma kriptografi merupakan langkah-langkah yang digunakan untuk proses enkripsi dan dekripsi. Semakin banyaknya waktu yang dibutuhkan untuk memacahkan data cipherteks menjadi plainteks, algoritma kriptografi tersebut dapat dikatakan semakin kuat.

Algoritma kriptografi disebut algoritma *restricted* jika kekutan kriptografi tersebut ditentukan dengan menjaga kerahasiaan algoritmanya. Namun, kriptografi yang sekarang atau kriptografi modern tidak lagi mendasarkan kekuatannya pada algoritma sehingga algoritmanya tidak dirahasiakan dan dapat diketahui oleh semua orang. Pada kriptografi modern,

kekuatan kriptografinya terletak pada kunci, yaitu berupa karakter atau bilangan bulat tertentu. Kunci ini bisa kita analogikan seperti sebuah *pasword* atau PIN pada kartu ATM sehingga hanya orang yang berhaklah yang dapat membukanya. Berikut adalah proses enkripsi dan dekripsi yang menggunakan kunci:

Sistem kriptografi adalah gabungan dari algoritma kriptografi, kunci, plainteks, dan cipherteks. Jika kunci yang digunakan untuk enkripsi dan dekripsi sama ($Key1 = Key2$), sistem kriptografinya disebut sebaga sistem kriptografi simetri (*symmetric-key cryptosystem*) dan algoritma kriptografinya disebut algoritma simetri. Contoh dari algoritma simetri adalah DES (*Data Encryption Standar*). Sedangkan jika kunci untuk enkripsi dan dekripsinya berbeda ($Key1 \neq Key2$), sistem kriptografinya disebut sistem kriptografi asimetri (*asymmetric cryptosystem*) dan algoritma kriptografinya disebut algoritma asimetri. Contoh algoritma asimetri adalah RSA (Rivest-Shamir-Adleman).

Kriptografi simetri sering juga disebut sebagai kriptogafi kunci-pribadi (*private-key cryptography*) karena baik kunci enkripsi maupun dekripsi keduanya harus dirahasiakan. Kekurangan dari sistem ini adalah baik pengirim maupun penerima pesan harus memiliki kunci yang sama sehingga pengirim pesan harus mencari cara untuk memberitahukan kunci kepada penerima agar kunci tersebut tidak bisa diketahui oleh pihak lain.

Kriptografi asimetri terkadang diebut sebagai kriptografi kunci-public (*public-key cryptography*). Algaoritma asimetri mempunyai dua buah kunci, yaitu kunci publik (*pubilc*-key) dan kunci pribadi (*private-key*). Kunci public digunakan untuk enkripsi dan tidak dirahasiakan, sedangkan kunci pribadi digunakan untuk dekripsi dan dirahasiakan. Pengirim pesan hanya mengetahui kunci publik dari si penerima pesan yang digunakan ketika mengenkripsi pesan yang akan dikirim. Penerima pesan memiliki kunci pribadi sehingga hanya dia lah yang dapat mendekrispi pesan dari pengirim pesan.

L. RSA (Rivest-Shamir-Adleman)

Algoritma RSA diperkenalkan poleh tiga peneliti MIT (Massacussets Institute of Technology), yaitu Ronald Rivest, Adi Shamir, dan Leornad Adleman pada tahun 1976. Algoritma RSA termasuk ke dalam algoritma asimetri karena kunci enkripsinya berbeda dengan kunci untuk dekripsi. Algoritma RSA menggunakan sepasang kunci, yaitu kunci publik e untuk mengenkripsi pesan dan kunci privat p untuk mendekripsi pesan. Kunci publik tidak dirahasiakan, sedangkan kunci privat dirahasiakan dan hanya diketahui oleh pemilik kunci.

Berikut adalah langkah-langkah pembangkitan pasangan kunci di dalam RSA:

1. Pilih dua buah bilangan prima sembarang, misalkan a dan b (dirahasiakan).
2. Hitung $n = a \times b$, nilai n tidak perlu dirahasiakan.
3. Hitung $m = (a-1)(b-1)$. Setelah dihitung, a dan b dapat dihapus untuk mencegah diketahui oleh pihak lain.
4. Pilih sebuah bilangan bulat untuk kunci publik, misalkan e, yang relatif prima terhadap m.
5. Hitung kunci dekripsi, misalkan d, dengan kekongruaenan $ed \equiv 1 \pmod{m}$.

Langkah enkripsi:

1. Nyatakan pesan mejadi blok-blok plainteks: $p_1, p_2, p_3, \ldots$ sedemikian rupa sehingga nilai $p_i \leq n-1$.
2. Hitung blok cipherteks c_i untuk blok plainteks p_i dengan persamaan $c_i = p_i^e \bmod n$, yang dalam hal ini e adalah kunci publik.

Langkah dekripsi:

1. Proses dekripsi dilakukan dengan menggunakan persamaan $p_i = c_i^d \bmod n$, yang dalam hal ini d adalah kunci pribadi.

III. RSA DALAM ENKRIPSI GAMBAR

Algoritma RSA biasanya digunakan dalam enkripsi teks. Dalam makalah ini, algoritma RSA akan digunakan untuk mengenkripsi gambar RGB. Sebelum proses enkripsi, gambar dibagi menjadi tiga buah matriks, yaitu matriks untuk warna merah (*red*), matriks untuk warna hijau (*green*), dan matriks untuk warna biru (*blue*). Selanjutnya, masing-masing matriks tersebut dienkripsi menjadi *cipher image* dengan menggunakan algoritma RSA yang sudah disesuaikan untuk enkripsi gambar. Hal yang perlu diperhatikan dalam proses enkripsi dan dekripsi gambar adalah rentang dari nilai warna, yaitu dari 0 sampai 255. Nilai warna sendiri sebenarnya menggambarkan seberapa terang warna tersebut. Berikut adalah ilustrasi proses pada enkripsi dan dekripsi gambar:

Prosedur untuk enkripsi gambar:

1. Baca masukkan gambar dan bentuk menjadi sebuah matriks RGB.
2. Bagi matriks RGB tersebut ke dalam tiga buah matriks R (untuk komponen warna merah), G (untuk komponen warna hijau), dan B (untuk komponen warna biru).
3. Pilih dua bilangan prima p dan q, dengan $p \neq q$.
4. Hitung nilai $n = pq$.
5. Hitung nilai $m = (p-1)(q-1)$.
6. Pilih bilangan bulat positif e, untuk kunci publik, sedemikian sehingga $1 < e < m$ dan $\text{PBB}(e, m) = 1$, e dan m relative prima.
7. Tentukan nilai kunci dekripsi d sedemikian sehingga $ed \equiv 1 \pmod{\text{m}}$.
8. Hitung masing-masing elemen matriks R, G, dan B sehingga diperoleh matriks hasil enkripsi dengan

fungsi $c = p^e \bmod n$, dengan p adalah elemen masing-masing matriks, e adalah kunci publik, dan c adalah elemen matriks hasil enkripsi.

9. Hitung masing-masing elemen matriks pada langkah 8 dengan formula $c' = c \bmod 256$ dan simpan $k = c \operatorname{div} 256$ ke dalam sebuah matriks baru, misalkan $MKey$, sesuai dengan posisi elemen c'. Matriks $MKey$ merupakan matriks kunci yang nantinya digunakan untuk proses dekripsi.
10. Gabungkan kembali matriks hasil enkripsi R, G, dan B lalu konversi matriks tersebut ke dalam bentuk gambar.

Prosedur untuk dekripsi gambar:

1. Tambahkan matriks R, G, dan B dengan matriks $MKey$, matriks yang diperoleh dari hasil enkripsi.
2. Hitung masing-masing elemen matriks R, G, dan B hasil penjumlahan tadi dengan fungsi $p = c^d \bmod n$, dengan c adalah elemen masing-masing matriks, d adalah kunci dekripsi, dan p adalah elemen matriks hasil dekripsi.

IV. IMPLEMENTASI

Pada makalah ini, penulis menggunakan bahasa python untuk implementasi algoritma enkripsi dan dekripsi gambar dengan RSA. Alasan penulis menggunakan bahasa python karena bahasa python mudah untuk dimengerti dan mempunyai banyak *library* yang memudahkan dalam pembuatan algoritma enkripsi dan dekripsi ini. Penulis memanfaatkan *library numpy* untuk pemrosesan matriks, *PIL* untuk pemrosesan gambar, *random* untuk memperoleh bilangan secara acak, dan *sympy* untuk pemeriksaan bilangan prima dan memperoleh PBB dari dua bilangan positif. Berikut adalah penjelasan fungsi dan prosedur serta algoritma utama yang diimplementasikan dalam bahasa python.

A. *image_to_rgb*

Pada *source code* berikut, gambar dikonversi ke dalam bentuk matriks RGB. Tujuannya adalah agar gambar tersebut dapat diproses lebih lanjut dalam tahap enkripsi dan dekripsi.

```
def image_to_rgb(filePath):
    with Image.open(filePath) as img:
        return np.array(img.convert("RGB"))
```

Gambar 1. Fungsi untuk konversi gambar ke matriks RGB

B. *inspectColor*

Gambar yang sudah dikonversi ke dalam matriks RGB selanjutnya dibagi ke dalam matriks komponen-komponennya, yaitu matriks R untuk komponen warna merah, matriks G untuk komponen warna hijau, dan matriks B untuk komponen warna biru.

```
def inspectColor(img_array):
    return np.int64(np.array(img_array[:,:,0])), np.int64(np.array
    (img_array[:,:,1])), np.int64(np.array(img_array[:,:,2]))
```

Gambar 2. Fungsi untuk memisahkan matriks RGB menjadi matriks komponen-komponennya

C. *save*

Prosedure save digunakan untuk menyimpan gambar hasil enkripsi dan dekripsi.

```
def save(img,filepath):
    pil_img = Image.fromarray(img.astype(np.uint8), mode = "RGB")
    pil_img.save(filepath)
    return
```

Gambar 3. Prosedur untuk menyimpan gambar hasil enkripsi dan dekripsi

D. *generatePrime*

Fungsi generatePrime digunakan untuk menghasilkan dua buah bilangan prima, misalkan p dan q. Bilangan prima yang dihasilkan berada di antara 128 dan 1024 agar tingkat keamanan dari algoritmanya semakin tinggi.

```
def generatePrime():
    p = random.randrange(128,1024)
    while (not(isprime(p))):
        p = random.randrange(128,1024)

    q = random.randrange(128,1024)
    while (p == q or not(isprime(q))):
        q = random.randrange(128,1024)

    return p,q
```

Gambar 3. Fungsi untuk menghasilkan dua buah bilangan prima

E. *mod*

Fungsi mod digunakan untuk operasi modulo agar tidak terjadi overflow ketika suatu bilangan dipangkatkan dengan angka yang sangat besar.

```
def mod(m,k,n):
    if (k == 1):
        return m % n
    elif (k % 2) == 1:
        m1 = mod(m**2 % n,k//2,n)
        return (m1*m % n)
    else:
        return mod(m**2 % n, k//2, n)
```

Gambar 4. Fungsi untuk operasi modulo

F. *encryption*

Fungsi encryption digunakan dalam proses enkripsi matriks. Fungsi encryption ini menerima tiga buah parameter, yaitu m untuk matriks, e untuk kunci enkripsi, dan sebuah bilangan positif n (hasil kali dua buah bilangan prima p dan q). Setelah proses enkripsi selesai, fungsi ini akan mengembalikan sebuah matriks hasil enkripsi dan sebuah matriks kunci yang nantinya digunakan dalam proses dekripsi gambar.

```
def encryption(m,e,n):
    m_chip = np.copy(m)
    m_key = np.empty_like(m)
    for i in range(len(m)):
        for j in range(len(m[0])):
            m_chip[i][j] = mod(m_chip[i][j],e,n)
            m_key[i][j] = m_chip[i][j] // 256
            m_chip[i][j] %= 256
    return m_chip, m_key
```

Gambar 5. Fungsi untuk proses enkripsi matriks

G. decryption

Fungsi decryption digunakan dalam proses dekripsi matriks yang sebelumnya sudah dienkripsi. Fungsi decryption ini menerima empat buah parameter, yaitu m_chip untuk matriks hasil enkripsi, m_key untuk matriks kunci, d untuk kunci dekripsi, dan sebuah bilangan bulat positif n (hasil kali dua buah bilangan prima p dan q). Setelah proses dekripsi selesai, fungsi ini akan mengembalikan sebuah matriks hasil dekripsi.

```
def decription(m_chip,m_key,d,n):
    m_plain = np.copy(m_chip)
    for i in range(len(m_chip)):
        for j in range(len(m_chip[0])):
            m_plain[i][j] = mod(m_plain[i][j] + 256*m_key[i][j], d, n)
    return m_plain
```

Gambar 6. Fungsi untuk proses dekripsi matriks

H. RSA

Fungsi RSA merupakan fungsi utama yang menjalankan proses enkripsi dan dekripsi gambar. Fungsi ini menerima sebuah parameter berupa alamat dari gambar yang akan dienkripsi dan mengembalikan dua buah matriks, yaitu matriks hasil enkripsi, dan matriks hasil dekripsi.

```
def RSA(file_path):
# Proses enkripsi
    # Langkah 1
    img = image_to_rgb(file_path)
    # Langkah 2
    r, g, b= inspectColor(img)
    # Langkah 3
    p, q = generatePrime()
    # Langkah 4 dan 5
    n = p*q
    m = (p-1)*(q-1)
    # Langkah 6
    e = random.randrange(1,m)
    while (gcd(e,m) != 1):
        e = random.randrange(1,m)
    # Langkah 7
    k = 1
    while ((1+m*k) % e != 0):
        k += 1
    d = int((1+m*k)/e)
    # Langkah 8 dan 9
    r, MKey_r = encryption(r,e,n)
    g, MKey_g = encryption(g,e,n)
    b, MKey_b = encryption(b,e,n)
    # Langkah 10
    img_chip = np.dstack([r,g,b])

# Proses dekripsi
    r = decription(r,MKey_r,d,n)
    g = decription(g,MKey_g,d,n)
    b = decription(b,MKey_b,d,n)
    img_plain = np.dstack([r,g,b])

    return img_chip, img_plain
```

Gambar 7. Fungsi untuk algoritma RSA

I. Algoritma Utama

Algoritma utama merupakan algoritma untuk menjalankan semua fungsi yang sudah diimplementasikan sebelumnya. Algoritma ini akan meminta user untuk memasukkan sebuah alamat gambar yang akan dienkripsi. Setelah proses enkripsi dan dekripi selesai, program akan menyimpan gambar hasil enkripsi ke dalam file yang bernama chiper_image.jpg dan gambar hasil dekripsi ke dalam file yang bernama plain_image.jpg.

```
# ALGORITMA UTAMA
filepath = input("Masukkan path relatif gambar: ")
img_chip, img_plain = RSA(filepath)
save(img_chip,"chiper_image.jpg")
save(img_plain,"plain_image.jpg")
```

Gambar 8. *Source code* program utama

Keseluruhan algoritma di atas dapat dilihat pada tautan berikut https://github.com/ikmalalfaozi/RSA-for-Image.

V. HASIL

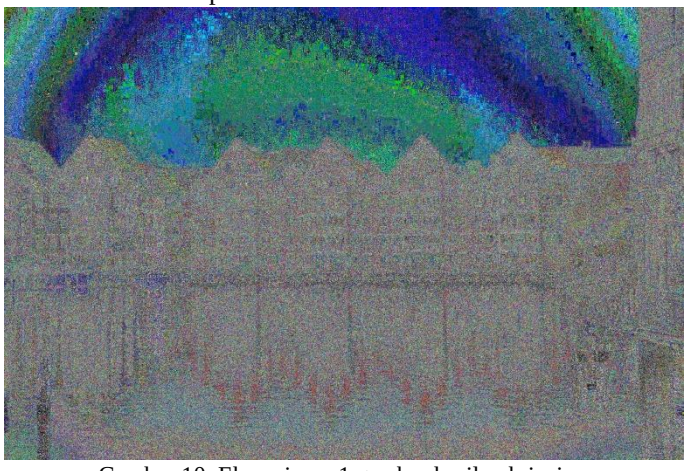
Berikut adalah hasil dari enkripsi dan dekripsi gambar menggunakan algoritma RSA.

Eksperimen 1:

Gambar asli:

Gambar 9. Eksperimen 1 gambar asli

Gambar hasil enkripsi:



Gambar 10. Eksperimen 1 gambar hasil enkripsi

Gambar hasil dekripsi:

Gambar 11. Eksperimen 1 gambar hasil dekripsi

Gambar hasil dekripsi:

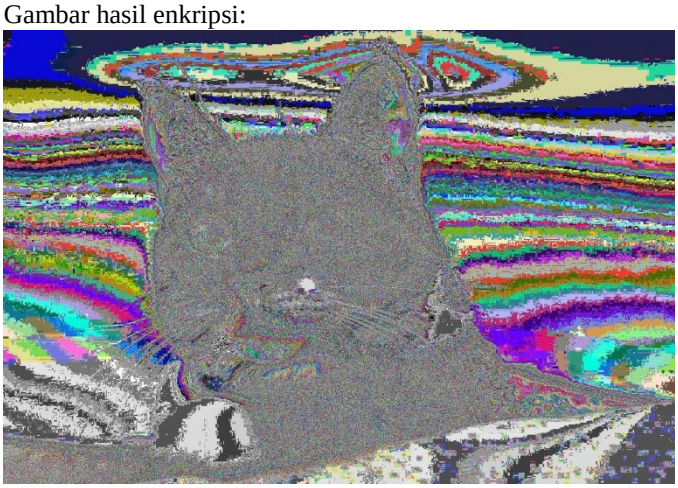
Gambar 14. Eksperimen 2 gambar hasil dekripsi

Eksperimen 2:

Gambar asli:

Gambar 12. Eksperimen 2 gambar asli

Gambar hasil enkripsi:



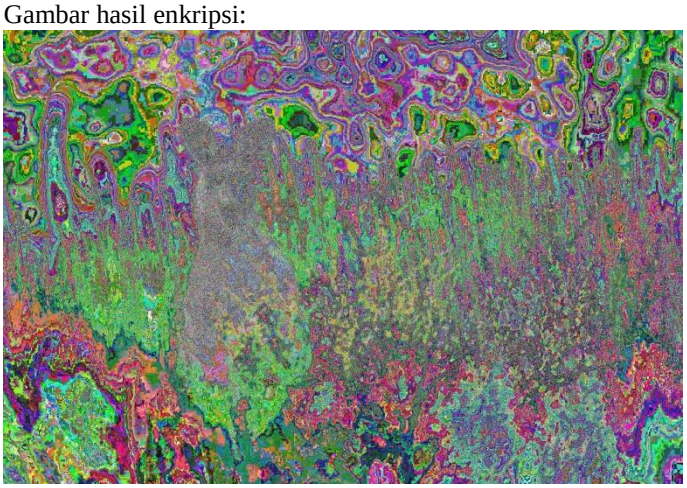
Gambar 13. Eksperimen 2 gambar hasil enkripsi

Eksperimen 3:

Gambar asli:

Gambar 15. Eksperimen 3 gambar asli

Gambar hasil enkripsi:



Gambar 16. Eksperimen 3 gambar hasil enkripsi

Gambar hasil dekripsi:

Gambar 17. Eksperimen 3 gambar hasil dekripsi

VI. KESIMPULAN

Gambar hasil enkripsi dengan algoritma RSA sangat berbeda dengan gambar aslinya. Namun, untuk gambar yang sederhana seperti gambar pada eksperimen kedua, pola gambar aslinya masih terlihat. Selain itu, kekurangan dari program yang penulis buat adalah waktu yang dibutuhkan untuk menjalankan program sangat lambat. Hal itu disebabkan oleh operasi modulo yang sangat banyak.

VII. UCAPAN TERIMA KASIH

Pertama-tama penulis panjatkan puji syukur kehadirat Allah SWT karena atas rahmat-Nya penulis dapat menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada keluarga yang selalu mendukung penulis selama proses penulisan. Akhir kata, penulis mengucapkan terima kasih kepada semua dosen pengampu mata kuliah Matematika Diskrit, terutama Ibu Dr. Nur Ulfa Maulidevi, S.T., M.Sc selaku dosen K03 atas ilmu yang telah diberikan kepada penulis sehingga penulis dapat mennyelesaikan makalah ini.

REFERENSI

[1] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Teori-Bilangan-2020-Bagian1.pdf diakses pada 5 Desember 2021 pukul 09.00.

[2] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Teori-Bilangan-2020-Bagian2.pdf diakses pada 5 Desember 2021 pukul 14.00.

[3] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Teori-Bilangan-2020-Bagian3.pdf diakses pada 5 Desember 2021 pukul 19.00.

[4] Munir, Rinaldi. 2010. *Matematika Diskrit (edisi 3).* Bandung: Informatika Bandung.

[5] Anandakumar, S. (2015). Image cryptography using RSA algorithm in network security. *International Journal of Computer Science & Engineering Technology*, 5(9), 326-330.

[6] Chepuri, S. (2017). An rgb image encryption using rsa algorithm. *International Journal of Current Trends in Engineering & Research (IJCTER)*, *3*(3), 1-7.

[7] Bhagat, P., Singh, M., Vishwavidhalaya, K., & Kalan. (2017). Image Encryption/Decryption Using RSA Algorithm. *International Journal of Computer Science and Mobile Applications*, 5(5), 1-14.

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.

Cilacap, 12 Desember 2021

Ikmal Alfaozi 13520125