# PENERAPAN ALGORITMA RSA PADA CITRA DIGITAL

## SKRIPSI

**Diajukan oleh:**
**AMELDA AUNIYAH**
**1911102441019**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR**
**JULI 2024**

**PENERAPAN ALGORITMA RSA PADA CITRA DIGITAL**

# SKRIPSI

Diajukan Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana
Fakultas Sains dan Teknologi Universitas Muhammadiyah Kalimantan Timur

**Diajukan oleh:**
**Amelda Auniyah**
**1911102441019**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR**
**JULI 2024**

# LEMBAR PERSETUJUAN

# PENERAPAN ALGORTIMA RSA PADA CITRA DIGITAL

## SKRIPSI

**Diajukan oleh:**
**AMELDA AUNIYAH**
**1911102441019**

**Disetujui untuk diujikan**
**Pada tanggal 27 juni 2024**

**Pembimbing**

**Sayekti Harits Suryawan, S.kom., M.Kom.**
**NIDN: 1119048901**

**Mengetahui,**
**Koordinator Skripsi**

**Abdul Rahim, S.Kom., M.Cs**
**NIDN: 0009047901**

# LEMBAR PENGESAHAN

# PENERAPAN ALGORTIMA RSA PADA CITRA DIGITAL

## SKRIPSI

**Diajukan Oleh:**
**Amelda Auniyah**
**1911102441019**

**Diseminarkan dan Diujikan**
**Pada tanggal 16 juli 2024**

| Penguji I | Penguji II |
|---|---|
| Abdul Rahim, S.Kom., M.Cs | Sayekti Harits Suryawan, S.Kom, M.Kom |
| NIDN. 0009047901 | NIDN. 1119048901 |

**Mengetahui,**
**Ketua**
**Program Studi Teknik Informatika**

**Arbansyah, S.kom., M.TI**
**NIDN. 1118019203**

## PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertanda tangan di bawah ini:

| | |
|---|---|
| Nama | : Amelda Auniyah |
| NIM | : 1911102441019 |
| Program Studi | : S1 Teknik Informatika |
| Judul Penelitian | : Penerapan Algortima RSA Pada Citra Digital |

Menyatakan bahwa **skripsi** yang saya tulis ini benar-benar hasil karya saya sendiri, dan bukan merupakan hasil plagiasi/falsifikasi/fabrikasi baik sebagian atau seluruhnya.

Atas pernyataan ini, saya siap menanggung risiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam tugas skripsi ini, atau klaim dari pihak lain terhadap keaslian karya saya ini.

Samarinda, 05 Juli 2024

Yang membuat pernyataan


Amelda Auniyah

1911102441019

# ABSTRAK

Keamanan data digital, termasuk gambar, menjadi semakin penting di era informasi saat ini. Penelitian ini bertujuan untuk mengamankan gambar digital melalui pengacakan dan enkripsi menggunakan algoritma RSA. Penelitian ini bertujuan untuk mengamankan gambar digital melalui pengacakan dan enkripsi menggunakan algoritma RSA. Teknik pengacakan meningkatkan keacakan nilai piksel gambar, sementara enkripsi RSA memastikan keamanan data dengan mengubah gambar menjadi bentuk yang tidak dapat dikenali tanpa kunci privat. Hasil menunjukkan peningkatan signifikan dalam entropi gambar setelah pengacakan dan enkripsi, yang kembali normal setelah dekripsi dan pengembalian dari pengacakan. Ini membuktikan bahwa kombinasi pengacakan dan enkripsi RSA efektif dalam mengamankan gambar digital. Saran untuk pengembangan lebih lanjut mencakup penggunaan ukuran kunci yang lebih besar, teknik pengacakan yang lebih kompleks, dan integrasi dengan sistem keamanan lainnya.

Kata Kunci: Keamanan gambar digital, pengacakan, enkripsi RSA, dekripsi, entropi.

## *ABSTRACT*

*The security of digital data, including images, is becoming increasingly important in today's information era. This research aims to secure digital images through randomization and encryption using the RSA algorithm. This research aims to secure digital images through randomization and encryption using the RSA algorithm. Randomization techniques increase the randomness of image pixel values, while RSA encryption ensures data security by converting images into a form that cannot be recognized without the private key. Results show a significant increase in image entropy after randomization and encryption, which returns to normal after decryption and return from randomization. This proves that the combination of randomization and RSA encryption is effective in securing digital images. Suggestions for further development include the use of larger key sizes, more complex randomization techniques, and integration with other security systems.*



*Keywords: Digital image security, randomization, RSA encryption, decryption, entropy.*

# PRAKATA

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Assalammu'alaikum Warahmatullahi Wabarakatuh.

Alhamdulillaahi Robbil'alamin, Segala puji dan syukur saya yang setinggi-tinnginya mengucapkan kepada Allah SWT, karena telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga saya dapat menyelesaikan proposal skripsi ini dengan judul "Penerapan Algoritma RSA pada Citra Digital". Allahumma sholli'ala Muhammad wa'ala ali sayyidina Muhammad, Tidak lupa pula saya hanturkan ucapakan untuk junjungan penulis rasul allah yaitu suri tauladan saya yaitu Nabi Muhammad SAW. Selama saya menyelesaikan skripsi ini tidak lepas dari banyak bantuan, bimbingan, dan saran dari banyak pihak baik secara tidak langsung maupun langsung. Dalam kesempatan ini penulis ingin menyampaikan banyak berterima kasih yang sebesar-besarnya kepada:

1. Dr. Muhammad Musiyam, M.T. Selaku Rektor Universitas Muhammadiyah Kalimantan Timur
2. Bapak Arbansyah, S.Kom., M.TI. Terima kasih selaku kepala Prodi Teknik Informatika.
3. Bapak Sayekti Harits Suryawan, M. Kom selaku dosen pembimbing yang telah banyak membantu penulis selama menyusun proposal skripsi ini, termasuk memberikan bimbingan, ide, saran, dan kritinya dengan sabar sehingga proposal skripsi ini dapat terselesaikan.
4. Bapak Rudiman, S.Kom., M.SC. Selaku dosen pembimbing akademik.
5. Teruntuk orang yang berjasa dan teristimewa dalam hidup saya, kedua orang tua saya yaitu Bapak Aliansyah dan Ibu Juraiyah orang yang hebat yang selalu menjadi penyemangat saya sebagai sandaran terkuat dari kerasnya dunia. Yang tidak pernah henti-hentinya memberikan kasih sayang dengan penuh cinta dan selalu memberikan semangat dan motivasi, Terima kasih selalu berjuang untuk kehidupan saya, dan selalu memberikan doa hingga saya bisa berada dititik ini. Sehat selalu kepada adikku Dira Fahrezi dan keluarga di sana terima kasih telah menjadi alasan penulis untuk menyelesaikan proposal skripsi ini. Karna merekalah menjadi semangat datang untuk pulang kerumah demi menempuh Pendidikan di bangku perkuliahan.
6. Teman-teman terdekat penulis terima kasih selalu ada dalam susah senangnya penulis, telah memberikan semangat dan dukungan dan arahannya walaupun sesama mengerjakan proposal skripsi. Terima kasih untuk kalian untuk perjuangannya bersama-sama.Teruntuk diri sendiri terimakasih telah kuat memalui proses ini yang bisa dibilang tidak mudah dan serta telah bertanggung jawab untuk menyelesaikan apa yang telah di mulai dari awal.

Skripsi ini masih jauh dari kata sempurna, saya sebagai penulis menyadari bahwa dalam penulisan laporan skripsi ini masih banyak kesalahan dan kekurangan. Saran dan kritik yang membangun semangat saya sangat diharapkan untuk menuju kesempurnaan laporan ini. Semoga laporan saya ini dapat memberikan manfaat bagi pembaca. Aamiin.

Wassalamu'alaikum wa rahmatullahi wa barakatuh.

Samarinda 05 Juli 2024

Amelda Auniyah

# DAFTAR ISI

# DAFTAR TABEL

**Tabel** **Halaman**

# DAFTAR GAMBAR

**Gambar** **Halaman**

# DAFTAR LAMPIRAN

**Lampiran** **Halaman**

# BAB I

# PENDAHULUAN

## 1.1 Latar belakang

Keamanan merupakan salah satu faktor penting dalam penyimpanan dan pengiriman gambar. Keamanan dapat diartikan sekelompok langkah, prosedur, dan strategi yang digunakan untuk menghentikan dan mengamati akses ilegal, pemecahan masalah, pengungkapan, gangguan dan penyesuaian sumber yang didapat (Saputro, Hidayati & H. Ujianto, 2020). Keamanan pertukaran gambar banyak dilakukan menggunakan media internet, salah satunya adalah menggunakan berbagai macam aplikasi yaitu WhatsApp, Instagram, Telegram, dan yang lain. Maka pesan gambarlebih cepat tersampaikan bahkan hanya dalam hitungan detik serta tidak memakan banyak waktu (Deskiva, 2018).

Ada kalanya gambar digunakan dalam berbagai banyak bidang seperti keamanan, medis, ilmu, teknik, seni, hiburan, iklan, pendidikan serta pelatihan. Dengan bertambahnya penggunaan teknik digital bagi transmisi dan penyimpanan gambar, masalah mendasar untuk melindungi kerahasiaan, kebutuhan dan keaslian gambar memang perlu diperhatikan(Christian, Sitorus & Nirmala, 2023). Hal ini dikarenakan kerahasiaan suatu informasi sangatlah penting dan bersifat pribadi. Gambar dapat didefinisikan sebagai hasil buatan dari manusia, hewan, tanaman, dan lain-lain yang dihasil media gambar tersebut. Terdapat aktifitas manusia khususnya pertukaran gambar(Yudanto & Suartana, 2022). Citra memiliki elemen terkecil yaitu piksel yang merupakan komponen gambar terkecil, memiliki nilai numerik intensitas piksel yang berkisaran warna-warna tertentu(Hendrawaty, TB & Munawir, 2022). Pada gambar RGB, setiap piksel pada gambar memuat tiga warna utama yaitu merah, hijau, dan biru. Masing-masing warna tersebut nantina diproses menjadi sebuah matriks yang kemudian dimanupulasi isinya sehingga menjadi *chipper image* (gambar yang sudah dienkripsi)(Fakhrizal *et al.*, 2023) .

Salah satu cara menjaga kerahasiaan data berupa citra adalah dengan cara menyembunyikan bentuk dari citra. Menyembunyikan bentuk citra dapat dilakukan dengan menggunakan Algoritma kriptografi. Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu enkripsi, dekripsi, dan kunci. Enkripsi adalah mengubah pesan atau data menjadi kode-kode yang tidak dimengerti. Keamanan dari kriptografi didapat dengan merahasiakan kunci yang dimiliki dari orang lain, tanpa harus merahasiakan algoritma itu sendiri(Sutejo, 2021).

Algoritma RSA adalah salah satu kriptografi asimetri yang sangat populer dipakai dan bahkan masih banyak hingga saat ini, algoritma RSA ini merupakan jenis kriptografi yang menggunakan dua kunci yang berbeda: kunci public (*public key*) dan kunci pribadi (*private key*). Dengan demikian, maka terdapat satu kunci, yakni kunci publik, yang dapat dikirimkan melalui saluran yang bebas, tanpa adanya suatu kemanan tertentu(Safarina & Shamir, 2017). Keamanan pada algoritma RSA ini bisa dilihat dari hasil factor-faktor prima dari bilangan besar. Hasil pemfaktoran itulah yang digunakan untuk memeperoleh kunci privat. Hal ini bertolak belakang dengan kriptografi simetri yang hanya menggunakan satu jenis kunci dan kunci tersebut harus terus terjaga keamanan serta kerahasiaannya. Dalam kriptografi asimetri, dua kunci tersebut diatur sedemikian sehingga memiliki hubungan dalam suatu persamaan aritmatika modulo(Trisnawati *et al.*, 2023).

Beberapa penelitian terdahulu terkait melakukan Kriptografi RSA, diantaranya, (Nazir, Arnellis & Dewi, 2019), melakukan penelitian tentang “Penerapan Algoritma RSA untuk File Citra Menggunakan Visual Basic” bertujuan agar file citra menjadi rahasia dan menjadi terjaga secara aman (Harbani & Fahreza, 2019), melakukan penelitian tentang “Aplikasi Keamanan Data Gambar Menggunakan

Algoritma RSA (Rivest Shamir Adleman) Berbasis Desktop" bertujuan agar mengamankan kerahasiaan file gambar dengan menggunakan kunci, dan menerapkan algoritma RSA dalam aplikasi pengamanan file gambar. (Azhar & Yuliany, 2019), melakukan penelitian tentang "Implementasi Algoritma RSA Untuk Enkripsi dan Dekripsi File" bertujuan agar meningkatkan dalaam keamanan data atau dokumen. (Khamsyar & Basri, 2022), melakukan penelitian tentang "Aplikasi Enkripsi Menggunakan Metode (Rivest Shamir Adleman) RSA", bertujuan untuk mengamankan gambar menggunakan aplikasi enkripsi metode RSA agar bersifat aman. (Baharsyah, Bandung & Bandung, 2023), melakukan penelitian tentang "Implementasi Algoritma RSA dalam Enkripsi dan Dekripsi File Teks", bertujuan agar file teks yang dikirimkan tidak terjadi kebobolan data.

Pada penelitian sebelumnya terdapat penelitian (Alfaozi, 2021), melakukan penelitian tentang "Aplikasi Algoritma RSA dalam Enkripsi dan Dekripsi Gambar". Pada penelitian ini hanya melihat keamanan aplikasinya, akan tetapi pada pola gambar aslinya masih terlihat, enkripsi yang dilakukan pada setiap pixel masih menggunakan metode dan kunci yang sama sehingga gambar aslinya masih terlihat. Kekurangan pada program tersebut yang di buat dibutuhin waktu untuk menjalankan karena program sangat lambat karena menggunakan operasi yang tidak tepat. Berdasarkan penelitian sebelumnya maka perbedaanya adalah bertujuan untuk lebih meningkatkan kualitas citra, menerapkan Algoritma RSA dalam mengenkripsi gambar, maka penulis mencoba untuk melihat algoritma RSA pada file citra digital dengan menganalisa algoritma RSA terhadap proses pengiriman. Dapat meningkatkan tingkat keamanan gambar yang telah dikirimkan, melalui proses enkripsi dan dekripsi yang menggunakan kunci dan password agar informasi yang terdapat pada gambar tetap terjaga kerahasiannya. Dan lebih mengutaman kemanan setiap kunci dan memakai kunci yang berbeda.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka penulis dapat merumuskan permasalahan yaitu (i) Bagaimana mengimplementasikan enkripsi gambar menggunakan algoritma RSA secara baik dan benar untuk meningkatkan keamanan pada gambar? (ii) Bagaimana pengujian penerapan algoritma RSA pada citra digital?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini yaitu (i) Meningkatkan keamanan gambar dan melindungi gambar dari akses orang yang tidak bertanggung jawab. Dan menerapkan algoritma RSA dalam mengamankan gambar dalam proses pengiriman dan membangun sistem pengamanan gambar agar gambar dapat ditingkatkan secara aman dengan melakukan enkripsi dan disimpan pada media lain berupa citra digital. (ii) Mengetahui keakuratan algoritma RSA melalui pengujian dengan mengenkripsi gambar

## 1.4 Manfaat Penelitian

Manfaat dari penelitian ini yaitu (i) Menambah pengetahuan mengenai penerapan kriptografi, khususnya pada algoritma RSA pada citra digital. Selama pengiriman, keaslian citra digital tetap terjaga dan resiko akses yang tidak sah dapat diminimalkan. (ii) Keamanan informasi dalam citra digital menjadi lebih terjamin, dengan menggunakan enkripsi RSA, citra dapat terlindung dari ancaman pemalsuan.

# BAB II

# METODE PENELITIAN

## 2.1 Obyek penelitian

Obyek penelitian ini yang digunakan dalam penelitian ini adalah penerapan algoritma RSA pada citra digital. Penerapan ini mencangkup proses enkripsi dan dekripsi pada citra digital menggunakan algoritma RSA. Citra digital yang digunakan ini dapat berupa berbagai jenis gambar dalam format seperti JPEG, PNG, atau BMP. Karena penelitian ini bertujuan untuk menguji aktifitas dan keamanan penerapan algoritma RSA, pada citra digital serta untuk mengevaluasi kualitas dari citra yang dihasilkan setelah di enkripsi dan dekripsi.

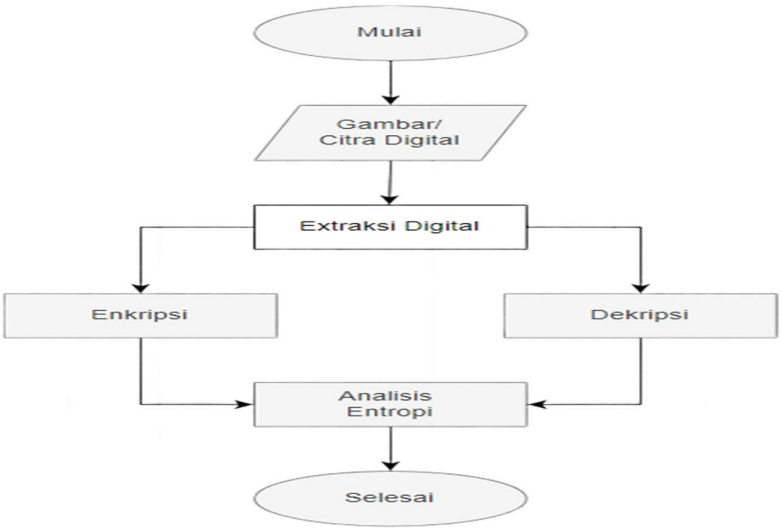
Penelitian ini dilakukan agar tau pentingnya keamanan gambar dalam konteks citra digital sering kali terdapat yang tidak baik. Dengan menerapkan algoritma RSA ini maka citra digital dapat diamankan tanpa memikir ada resiko yang berbahaya, sehingga dapat memberikan kenyamanan yang baik dalam menjaga kerahasiaan pada citra itu sendiri.

## 2.2 Alat dan Bahan

Untuk membantu proses penelitian dibutuhkan peralatan yang mendukung, Adapun peralatan yang digunakan pada penelitian ini yaitu perangkat keras (Hardware) yang di pakai adalah Laptop Lenovo Ideapad Slim 1, Processor AMD Ryzen 3 7000, dan RAM 8 GB. Perangkat lunak (Software) yang dipakai OS Windows 11, dan Python 3 atau google colab.

## 2.3 Prosedur penelitian

Agar penelitian dapat dilaksanakan secara terstruktur, diperlukan panduan atau acuan berupa prosedur penelitian. Diagram prosedur penelitian dapat dilihat dalam Gambar 2.1.




**Gambar 2.1** Prosedur Penelitian.

Berikut adalah penjelasan langkah-langkah pada gambar diatas:

### 2.3.1 Gambar/Citra Digital

Tahap awal ini adalah pengambilan gambar yang akan di enkripsi dan didekripsi. Sebelum melakukan proses tahap selanjutnya, citra maka yang dilakukan adalah gambar dibagi menjadi tiga buah matriks, yaitu matriks warna merah (red), matriks warna hijau (green), dan matriks warna biru (blue). Gambar inj dapat berbentuk gambar digital dalam format seperti JPEG atau PNG. Pengambilan gambar ini dapat dilakukan dengan menggunakan kamera atau memanfaatkan data yang sudah tersedia di media internet. Selanjutnya, tahap dimana melakukan gambar yang diperoleh dengan melakukan proses enkripsi dan dekripsi ke dalam media ekstraksi gambar dan menerapkan kombinasi algoritma RSA. Dibawah ini kode uploud gambar yang akan ditunjukkan pada gambar 2.2.

**Gambar 2. 2** Kode Uploud Gambar

```
# Fuction to upload image
def upload_and_process_image():
    upload = FileUpload(accept='image/*', multiple=False)
    container = VBox([upload])

    def on_upload_change(change):
        for filename, file_info in upload.value.items():
            img = Image.open(BytesIO(file_info['content']))
            display(img)
```

```
Upload an image to scramble and encrypt using RSA:
        Upload (0)
```

Berikut ini adalah gambar/citra yang akan digunakan dalam enkripsi ditunjukkan pada gambar 2.3.

**Gambar 2. 3** Gambar/Citra Digital

### 2.3.2 Data extraction

Tahap ini melakukan pada gambar yang merujuk kepada proses pengambilan nilai-nilai numerik merepresensasikan warna atau intensitas dari setiap piksel dalm sebuah gambar. Pada gambar digital pixel adalah unit-unit terkecil dari informasi warna yang secara kolektif membentuk gambar tersebut. Pada proses ekstraksi data piksel ini melibatkan akases terhadap nilai-nilai, biasanya direpresentasikan dalam bentuk larik numerik, dan dapat mencangkup berbagai aspek seperti (i) Ukuran gambar pada data extraction melibatkan pengambilan informasi mengenai ukuran gambar, yang dinyatakan dalam jumlah piksel. Ukuran gambar mempengaruhi jumlah total data yang akan dienkripsi, yang penting untuk perencanaan penyimpanan dan pengiriman data terenkripsi. (ii) Ruang warna: Identifikasi ruang warna yang digunakan dalam gambar RGB. (iii) Mengubah data pixel ke format yang cocok untuk analisis lebih lanjut, seperti menggunakan larik numerik untuk algoritma pembelajaran mesin atau teknik pemrosesan gambar, Tahap ini dilakukan untuk memastikan bahwa gambar yang akan dienkripsi telah dipersiapkan dengan baik dan sesuai dengan kebutuhan sebelum dilakukan langkah-langkah keamanan lebih lanjut. Berikut ini adalah informasi coding data extraction yang diperoleh dapat dilihat pada tabel 2.4.

```
# Fungsi untuk ekstraksi data gambar
def extract_image_data(image):
    image_data = {
        "Ukuran": image.size,
        "Mode Warna": image.mode,
        "Format": image.format,
    }
    return image_data
```

**Gambar 2. 4** *Coding* Extraction Data

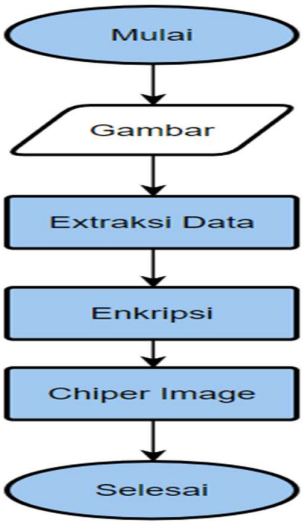
Berikut ini adalah informasi data extraction yang diperoleh dapat dilihat pada tabel 2.1.

**Tabel 2.1** Informasi Data Extraction

| Tahapan | Langkah | Keterangan |
|---|---|---|
| 1 | Definisikan fungsi | Fungsi *extract_image_data* didefinisikan dengan satu parameter yaitu image |
| 2 | Ambil ukuran gambar | *image.size* memberikan ukuran (width, height) dari gambar. |
| 3 | Ambil mode warna gambar | *image.mode* memberikan informasi tentang mode warna gambar (misalnya "RGB", "L", dll.). |
| 4 | Ambil format gambar | *image.format* memberikan informasi tentang format gambar (misalnya "JPEG", "PNG", dll.). |
| 5 | Simpan data dalam dictionary | Data yang diekstrak (ukuran, mode warna, dan format) disimpan dalam dictionary bernama *image_data.* |
| 6 | Kembalikan data gambar | Fungsi mengembalikan dictionary *image_data* yang berisi data gambar. |

### 2.3.3 Enkripsi

Dari tahap ini yaitu alur dari proses enkripsi ditujukkan dimna gambar yang telah disiapkan akan dienkripsi menggunakan algoritma yang telah ditentukan yaitu algoritma RSA. Proses ini memerlukan kunci rahasia untuk gambar yang akan dinekripsi. Dibawah ini alur yang akan dienkripsi ditunjukkan pada gambar 2.4.




**Gambar 2.5** Alur Enkripsi Algoritma RSA

```
# Fungsi untuk enkripsi pesan menggunakan RSA
def rsa_encrypt(message, public_key):
    e, n = public_key
    cipher = pow(message, e, n)
    return cipher
```

**Gambar 2.6** Coding Enkripsi.

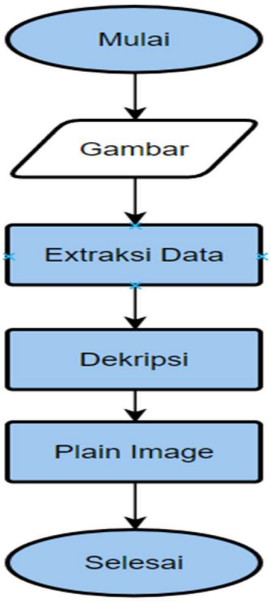
Berikut ini adalah table 2.2 yang berisi tentang input dan penjelasan yang digunakan dalam kode coding diatas tersebut:

**Tabel 2. 2** Keterangan Coding Enkripsi

| Fungsi | Langkah | Keterangan |
|---|---|---|
| Ekstraksi Kunci Publik | *e, n = public_key* | Mengambil komponen `e` (eksponen publik) dan `n` (modulus) dari pasangan kunci publik yang diberikan. |
| Enkripsi Pesan | *cipher = pow (message, e, n)* | Mengenkripsi pesan dengan menggunakan operasi modular eksponen. Fungsi `pow` menghitung *`message^e`* |
| Mengembalikan Cipher | *return cipher* | Mengembalikan nilai cipher yang merupakan hasil enkripsi pesan dengan kunci publik. |

### 2.3.4 Dekripsi

Dari proses sebelumnya di enkripsi, langkah selanjutya adalah proses dekripsi. dalam proses ini, gambar yang telah terenkripsi akan didekripsi Kembali menjadi bentuk aslinya mengunakan kunci rahasia yang sama yang digunakan untuk enkripsi. Proses dekripsi ini hasilnya akan mendaptakan file citra yang semula sebelum dienkripsi.




**Gambar 2. 7** Diagram alir dekripsi

Tahap pertama ini adalah proses memuat gambar yang sudah dienkripsi ke dalam folder di computer dan laptop. Setelah gambar di proses, langkah selanjutnya mengekstraksi gambar yang telah dienkripsi dari gambar tersebut. Gambar yang telah diekstraksi kemudian akan didekripsi menggunakan kunci privat. Proses ini akan mengembalikan gambar ke bentuk sebelum dienkripsi. Setelah gambar di

dekripsi, langkah terakhir adalah untuk mengembalikan gambar aslinya, harus melakukan proses menggabungkan kembali gambar yang telah didekripsi ke dalam format gambar yang sudah sesuai, seperti RGB, JPEG, PNG, dan gambar yang lainnya.

```
# Fungsi untuk dekripsi pesan menggunakan RSA
def rsa_decrypt(cipher, private_key):
    d, n = private_key
    message = pow(cipher, d, n)
    return message
```

**Gambar 2. 8** Coding Dekripsi

Berikut ini adalah table 2.3 yang berisi tentang input dan penjelasan yang digunakan dalam kode coding diatas tersebut:

**Tabel 2. 3** Keterangan coding dekripsi

| Fungsi | Langkah | Keterangan |
|---|---|---|
| Ekstraksi Kunci Privat | *d, n = private_key* | Mengambil komponen `d` (eksponen privat) dan `n` (modulus) dari pasangan kunci privat yang diberikan. |
| Dekripsi Pesan | *message = pow (cipher, d, n)* | Mendekripsi pesan dengan menggunakan operasi modular eksponen. Fungsi `pow` menghitung `cipher^d % n`. |
| Mengembalikan Pesan | *return message* | Mengembalikan nilai pesan yang merupakan hasil dekripsi cipher dengan kunci privat. |

### 2.3.5 Analisis entropi

Proses ini salah satu metode yang digunakan untuk memvalidasi tingkat pengecakan kriptografi. Entropi mengkur ketidak pastian atau keacakan informasi. Dalam Konteks kroptografi, entropi keluaran algoritma kriptografi yang menunjukkan seberapa sulitnya algoritma tersebut untuk diprediksi. (i) Entropi ini dapat diukur dalam bit per symbol. Semakin tinggi entropi kelaran algoritma, maka semakin banyak bit per symbol yang dibutuhkan untuk diprediksi dan karenya lebih diacak. (ii) Rumusan Entropi bisa di lihat bahwa entropi (H) dari sebuah variable acak X yang dapat mengambil n nilai yang berbeda dengan probabilitas masing-masing P(x_i) dapat dihitung menggunakan rumus berikut:

$$H(x) = -\sum_{i=1}^{n} \mathrm{P}(xi) log_2 \mathrm{P}(xi) \quad (1)$$

Keterangan:

| | |
|---|---|
| H*(x)* | : Entropi dari variabel acak *x* |
| P*(xi)* | : Probabilitas dari kejadian *xi* |
| $Log_2$ | : Logaritma basis 2 |
| n | : Jumlah total kemungkinan kejadian |
| ∑ | :Simbol sigma (jumlah), yang menunjukkan bahwa kita menjumlahkan seluruh nilai yang dihitung di dalamnya. |

(iii) Penerapan dalam validasi disebut juga dalam validasi tingkat pengacakan kriptografi, entropi keluaran algoritma dihitung dan dibandingkan dengan entropi maksimum yang mungkin. Entropi maksimum adalah $\log_2(n)$, di mana n adalah jumlah nilai yang mungkin diambil oleh keluaran. (iv) Interpretasi hasil ini di bagi menjadi 2 bagian yaitu Entropi tinggi (dekat dengan entropi maksimum): Menunjukkan bahwa algoritma cukup acak dan sulit untuk diprediksi. Dan yang kedua Entropi rendah menunjukkan bahwa keluaran algoritma tidak cukup acak dan mungkin dapat diprediksi oleh penyerangan.

Jadi tahap ini bahwa entropi adalah sebagai alat yang berharga untuk memvalidasi tingkat pengacakan kriptografi pada citra digital itu sendiri. Dengan memahami konsep entropi ini dan cara menghitungnya, maka dapat menilai keacakan keluaran algoritma kriptografi dan memastkan keamanan sistem kriptografi pada saat ini.

```
# Fungsi untuk menghitung entropi dari distribusi nilai piksel dalam gambar
def calculate_entropy(image):
    pixels = np.array(image)
    histogram, _ = np.histogram(pixels, bins=256, range=(0, 256))
    histogram = histogram.astype(float) / pixels.size
    histogram = histogram[histogram != 0]
    entropy = -np.sum(histogram * np.log2(histogram))
    return entropy
```

**Gambar 2. 9** Coding Entropi

**Tabel 2. 4** Penjelasan Coding Entropi

| Tahapan | Langkah | Keterangan |
|---|---|---|
| Konversi Citra ke Array Numpy | *pixels = np.array(image)* | Mengonversi citra menjadi array numpy yang berisi nilai-nilai piksel citra tersebut. |
| Menghitung Histogram | *histogram, _ = np.histogram(pixels, bins=256, range=(0, 256))* | Menghitung histogram dari nilai-nilai piksel dengan 256 bins, rentang dari 0 sampai 256. Histogram ini menunjukkan frekuensi kemunculan setiap nilai piksel dalam citra. |
| Normalisasi Histogram | *histogram = histogram.astype(float) / pixels.size* | Mengubah histogram menjadi tipe data float dan menormalisasinya dengan membagi setiap nilai histogram dengan jumlah total piksel dalam citra. Ini mengubah histogram menjadi distribusi probabilitas. |
| Menghilangkan Nilai Nol dari Histogram | *histogram = histogram [histogram ! = 0]* | Menghapus nilai-nilai nol dari histogram untuk menghindari kesalahan dalam perhitungan logaritma, karena logaritma dari nol tidak terdefinisi. |
| Menghitung Entropi | *entropy = -np.sum(histogram * np.log2(histogram))* | Menghitung entropi menggunakan rumus entropi Shannon. Rumus ini mengalikan setiap nilai dalam histogram dengan logaritma basis 2 dari nilai tersebut, lalu menjumlahkan hasilnya dan mengubah tanda menjadi negatif. |
| Mengembalikan Nilai Entropi | *return entropy* | Mengembalikan nilai entropi yang telah dihitung, yang menggambarkan tingkat ketidakpastian atau kerandoman dalam distribusi nilai piksel citra. |

# BAB III

# HASIL DAN PEMBAHASAN

## 3.1 Hasil Penelitian

Penelitian ini berjudul penerapan algoritma RSA pada citra digital. Berdasarkan hasil penelitian jurnal sebelumnya penulis melakukan penelitian tentang penerapan algoritma RSA pada citra digital. Fokus utama adalah pada evaluasi terlebih dahulu untuk mengetahui efektivitas algoritma yang digunakan. Penerapan pada algoritma RSA untuk enkripsi pada gambar serta mentransformasi entropi yang digunakan untuk evaluasi kualitas keamanan gambar. Pengujian ini dilakukan untuk melihat entropi pada gambar yang melalui beberapa tahap proses seperti, pengacakan (s*crambling*), enkripsi, dekripsi, dan pengembalian gambar (*unscrambling*). Hasil pengujian ini dilakukan dengan menggunggah gambar, kemudian gambar tersebut diproses melalui beberapa beberpa langkah: pengacakan, enkripsi, dekripsi, dan pengembalian. Pada setiap langkah, hasil entropi gambar dari setelah dihitung untuk mengukur perubahan tingkat acak.

### 3.1.1 Gambar Asli

Pada tahap ini gambar 3.1 merupakan gambar asli ini sebelum melakukan proses enkripisi. Gambar asli ini berfungsi sebagai input awal dalam proses enkripsi dan dekripsi agar mengetahui nilai piksel dalam gambar asli nantinya.

**Gambar 3. 1** Gambar asli.

### 3.1.2 Ekstraksi Data

Pada ekstraksi data ini merupakan langkah sebelum melakukan dalam proses enkrispi dan dekripsi pada citra digital. Proses ini melibatkan nilai-nilai numerik yang mewakili ukuran, warna, dan format file dari setiap pixel dalam gamba. Hasil ekstraksi data dapat dilihat pada tabel 3.1.

**Tabel 3.1** Hasil Ekstraksi Gambar Asli

| Ukuran (Dimensi) | Tipe Warna | Format |
|---|---|---|
| 225 x 225 | RGB | JPEG |

### 3.1.3 Enkripsi

Gambar yang telah di *scrambling* dienkripsi menggunakan algoritma RSA dengan fungsi *encrypt_image()*. Kunci publik digunakan untuk mengenkripsi data gambar, menghasilkan data terenkripsi yang tidak dapat dikenali. Berikut ini adalah hasil gambar yang telah di *enkripsi* bisa dilihat pada gambar 3.2.

**Gambar 3. 2** Gambar Asli Sebelum Proses Enkripsi


**Gambar 3. 3** Hasil Gambar Enkripsi.

Hasil gambar yang telah di enkripsi ditampilkan dalam bentuk noise acak menunjukkan bahwa gambar asli telah diacak.

### 3.1.4 Dekripsi

Setelah proses yang telah di enkripsi kemudian didekripsi menggunakan kunci privat dengan fungsi *decrypt_image()*. Proses ini mengembalikan gambar ke bentuk scrambled sebelum enkripsi. Pada dekripsi ini menggunakan algoritma RSA yang sama seperti yang digunakan pada proses enkripsi. Setelah proses dekripsi selesai, gambar yang telah didekripsi ditampilkan dan dibandingkan dengan gambar asli sebelum proses enkripsi.


**Gambar 3. 4** Gambar Enkripsi Yang Akan Dekripsi

**Gambar 3. 5** Hasil gambar Dekripsi.

## 3.2 Algoritma RSA Termodifikasi (*Scrambling* dan *Unscrambling*)

Modifikasi algoritma RSA dengan teknik scramble dan unscramble bertujuan untuk meningkatkan lagi keamanan data visual seperti gambar yang dienkripsi. Proses dimulai dengan *scrambling* gambar, di mana distribusi piksel pada gambar diacak untuk menghilangkan pola atau struktur visual yang ada. Dengan ini dilakukan menambahkan offset acak pada nilai komponen warna piksel, sehingga membuat gambar lebih sulit dikenali dan meningkatkan keamanan. Setelah gambar diacak, langkah berikutnya adalah enkripsi RSA, di mana nilai-nilai piksel yang telah diacak dienkripsi menggunakan kunci publik RSA. Proses enkripsi RSA mengubah setiap nilai piksel menjadi cipher menggunakan eksponen enkripsi dan modulus.

Setelah gambar dienkripsi dan didekripsi dengan RSA, dilakukanlah pada nilai-nilai piksel yang terenkripsi, menggunakan kunci privat RSA untuk mengembalikan nilai piksel ke bentuk aslinya. Terakhir, gambar yang telah didekripsi menjalani proses *unscrambling* untuk mengembalikan distribusi piksel ke posisi semula. *Unscrambling* ini memastikan bahwa gambar yang didekripsi dapat ditampilkan dalam format yang dapat dikenali seperti sebelum enkripsi dilakukan. Dengan cara ini, modifikasi RSA yang melibatkan *scrambling* dan *unscrambling* tidak hanya meningkatkan keamanan dengan mengacak data visual sebelum enkripsi, tetapi juga memastikan integritas dan keaslian gambar setelah proses enkripsi dan dekripsi.

### 3.2.1 Gambar Asli Termodifikasi

Pada tahap ini, dilakukan analisis terhadap gambar asli yang di unggah sebagai dataset, sama seperti pada algoritma RSA asli. Gambar ini berfungsi sebagai input awal dalam proses enkripsi dan dekripsi. Nilai piksel dalam gambar asli akan menjadi dasar untuk penerapan teknik *scrambling*, enkripsi RSA, dekripsi, dan *unscrambling*. Gambar asli disimpan dalam format yang memungkinkan analisis lebih lanjut dan mempermudah pemantauan perubahan selama proses.

**Gambar 3. 6** Gambar Asli Sebelum Proses Enkripsi Termodifikasi

### 3.2.2 Ekstraksi Data Termodifikasi

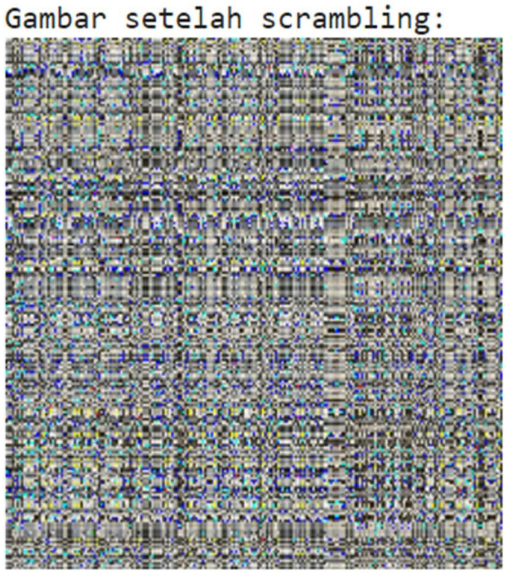
Pada proses ekstraksi data merupakan tahap awal dalam pengelolaan gambar dimana informasi gambar asli diambil dan dianalisis. Informasi ini melingkupi ukuran, warna, dan format file. Hasil ekstraksi data dapat dilihat pada Tabel 3.2.

**Tabel 3. 2** Hasil Ekstraksi Data Gambar Asli termodifikasi

| Ukuran (Dimensi) | Tipe Warna | Format |
|---|---|---|
| 225 x 225 | RGB | JPEG |

### 3.2.3 *Scrambling* dengan *Caesar Cihper*

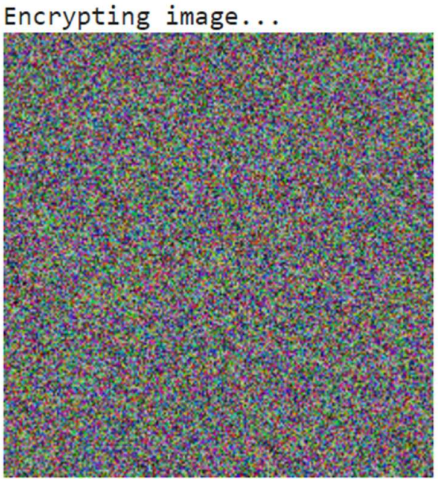
Tujuan dari pengacakan pada gambar yaitu untuk mengubah struktur spasial dari pixel dalam gambar, namun secara visual, gambar yang teracak tetap mempertahankan komposisi warna yang mirip dengan gambar asli. Gambar asli kemudian diacak menggunakan fungsi *scramble_image()*. Proses ini melibatkan dua langkah utama yaitu *Caesar cipher* mengubah nilai pixel gambar berdasarkan pergeseran yang ditentukan lalu dengan adanya permutasi menukar baris dan kolom gambar secara acak. Setiap piksel gambar digeser nilainya berdasarkan posisi piksel tersebut. Fungsi *caesar_cipher* menambahkan nilai shift yang dihitung dari koordinat piksel (x dan y) ke nilai piksel RGB (*Red, Green, Blue*). Proses ini menciptakan kerandoman tambahan pada gambar sehingga gambar menjadi lebih sulit dikenali.




**Gambar 3.7** Hasil Gambar Setelah Discrambling.

### 3.2.4 Enkripsi Termodifikasi

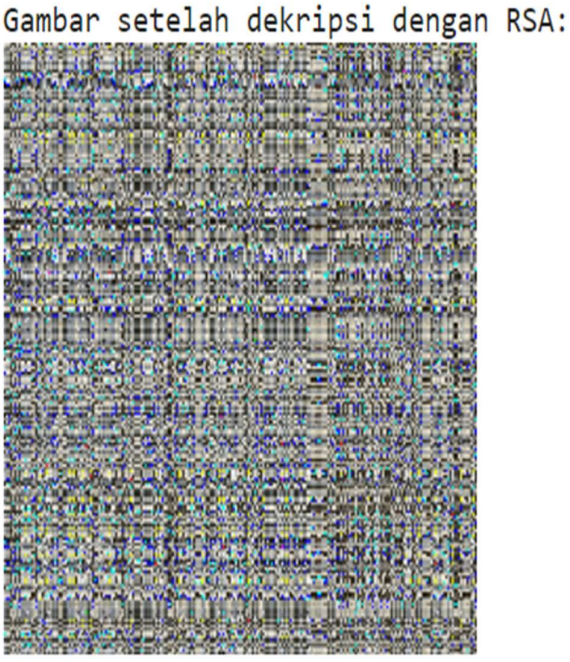
Pada gambar yang sudah diacak kemudian dienkripsi menggunakan Algoritma RSA dengan kunvi publik. Enkripsi ini mengubah gambar menjadi bentuk yang tidak dapat dikenal, dan meningkatkan keamanan dengan menjadikan gambar sulit untuk difahami atau dimengerti.




**Gambar 3.8** Setelah dilakukan Enkripsi

### 3.2.5 Dekripsi Termodifikasi

Setelah proses yang telah di enkripsi kemudian dekripsi menggunakan kunci privat RSA yang sesuai, ini menunjukkan bahwa proses dekripsi mengembalikan gambar keadaan yang sangat mirip dengan keadaan setelah enkripsi.

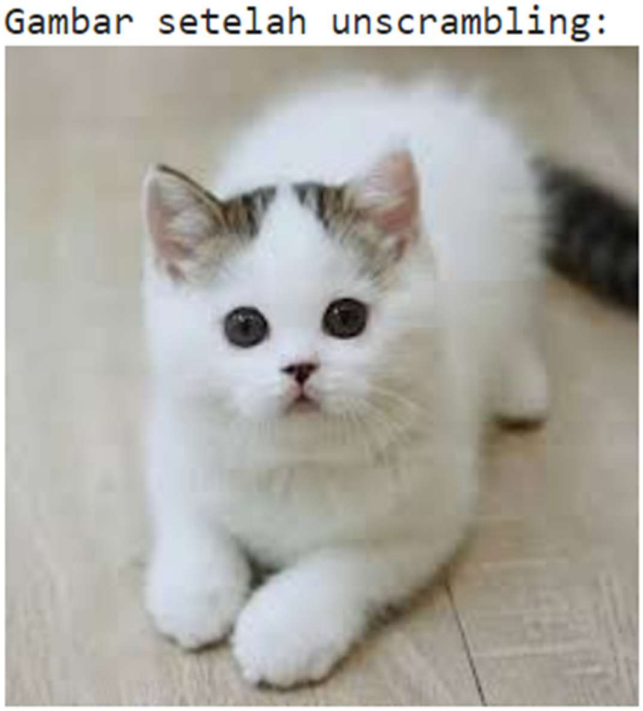



\

**Gambar 3.9** Hasil Gambar Setelah Dekripsi.

### 3.2.6 *Unscrambling*

Langkah terakhir dalam melakukan proses unscrambling yaitu mengembalikan gambar yang telah didekripsi ke bentuk gambar aslinya dengan menggunakan algoritma *unscrambling*. Pada proses ini mengembalikan efek dari *scrambling* yang dilakukan sebelumnya sehingga piksel pada gambar kembali possisi semula.




**Gambar 3.10** Gambar Setelah *Unscrambling*

### 3.2.7 Analisis Entropi

Analisis entropi digunakan untuk mengukur tingkat keacakan dalam sebuah gambar yang telah melalui proses enkripsi dan deksripsi. Nilai entropi yang tinggi menunjukkan tingkat keacakan pada gambar, yang berarti data menjadi lebih sulit untuk diprediksi da lebih aman. Berikut adalah hasil pengujian entropi secara menyeluruh, yang diperoleh dapat dilihat pada tabel 3.1.

**Tabel 3 3** Hasil Perhitungan Entropi Gambar

| Proses | Hasil Entropy(bits/pixel) | Hasil Entropi Termodifikasi |
|---|---|---|
| Gambar Asli | 7.09 | 7.09 |
| Gambar Setelah *Scrambling* | - | 8.00 |
| Data Terenkripsi | 7.76 | 7.95 |
| Gambar Setelah Dekripsi | 7.69 | 7.72 |
| Gambar Setelah *Unscrambling* | - | 7.21 |

Secara keseluruhan, hasil yang menunjukkan bahwa operasi Gambar asli memiliki entropi yang memberikan baseline untuk kerandoman gambar. Gambar setelah scrambling menunjukkan peningkatan

entropi yang mengindikasikan kerandoman lebih tinggi setelah aplikasi Caesar cipher dan permutasi. Data terenkripsi dengan RSA menunjukkan entropi yang lebih tinggi, menandakan kerandoman yang signifikan dalam data terenkripsi. Gambar setelah dekripsi Menunjukkan bahwa meskipun data telah kembali ke bentuk yang dapat diinterpretasi, tingkat keragamannya masih tinggi, mendekati gambar asli tetapi dengan sedikit perbedaan dan *unscrambling* menunjukkan bahwa proses pembalikan pengacakan berhasil mengembalikan sebagian besar informasi ke bentuk semula, meskipun ada sedikit penurunan keragaman informasi dibandingkan dengan gambar asli.

## 3.3 Pembahasan

Penelitian ini menggunakan penerapan algoritma RSA ***(****Rivest-Shamir-Adleman)* dalam pengamanan citra digital. Algoritma RSA adalah sebuah algoritma kriptografi yang menggunakan metode enkripsi dan dekripsi asimetris. Asimetris berarti bahwa ada dua kunci yang berbeda digunakan untuk enkripsi dan dekripsi data: kunci publik dan kunci privat.

a. ***Scrambling* dengan *Ceasar Cipher*:** *Scrambling* ini dilakukan dengan menggeser nilai piksel berdasarkan pola baris dan kolom, mirip dengan metode Caesar cipher pada teks. Dalam metode ini, setiap piksel gambar diacak dengan menambahkan nilai acak berdasarkan posisi baris dan kolomnya, menghasilkan pengacakan yang deterministik namun kompleks. Setelah citra diacak, proses enkripsi RSA dilakukan pada citra yang telah diacak. Pengacakan ini menambah lapisan tambahan keamanan dengan mengaburkan pola gambar asli sebelum enkripsi dilakukan.

b. **Enkripsi dan Dekripsi:** Algoritma RSA terbukti efektif dalam mengamankan citra digital melalui proses enkripsi dan dekripsi. Hasil penelitian menunjukkan bahwa citra yang telah terenkripsi tidak dapat dikenali tanpa kunci dekripsi yang benar, yang membuktikan bahwa algoritma ini berhasil dalam menjaga kerahasiaan citra digital. Pada langkah ini, kunci publik dan privat RSA dihasilkan menggunakan RSA.generate(2048) yang menghasilkan kunci dengan panjang 2048 bit. Pada tahap dekripsi, citra berhasil dikembalikan ke bentuk aslinya, yang menunjukkan bahwa proses dekripsi menggunakan kunci privat yang benar dapat mengembalikan citra yang terenkripsi ke bentuk semula tanpa kehilangan kualitas.

***c. Unscrambling*:** Unscrambling ini menggunakan dengan membalik proses scrambling yang telah dilakukan sebelumnya. Permutasi dibalik menggunakan urutan permutasi terbalik dan nilai-nilai piksel diubah kembali menggunakan fungsi reverse_caesar_cipher. Proses unscrambling ini memastikan gambar kembali ke susunan baris dan kolom semula dan nilai piksel yang sesuai dengan gambar asli sebelum di-scramble.

# BAB IV

# SIMPULAN DAN SARAN

## 4.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang dilakukan enkripsi dan dekripsi gambar menggunakan algoritma RSA serta teknik pengacakan dengan *Ceasar cipher* dan permutasi, dapat diambil kesimpulan sebagai berikut: (i) Teknik *scrambling* yang melibatkan *Caesar cipher* dan permutasi berhasil meningkatkan kerandoman gambar, seperti yang ditunjukkan oleh peningkatan entropi setelah proses scrambling. *Caesar cipher* menggeser nilai-nilai piksel berdasarkan posisi piksel, sementara permutasi menukar baris dan kolom gambar secara acak, menghasilkan gambar yang sulit dikenali tanpa kunci yang tepat. (ii) Algoritma RSA berhasil mengenkripsi gambar yang telah di-*scrambling*, menghasilkan data terenkripsi dengan entropi yang sangat tinggi, mencerminkan kerandoman maksimum. Proses dekripsi dengan kunci privat RSA berhasil mengembalikan gambar ke bentuk scrambled sebelum enkripsi, yang kemudian dapat di *unscramble* kembali ke bentuk aslinya dengan benar. (iii) Proses enkripsi dan scrambling yang diterapkan memberikan tingkat keamanan yang baik untuk data gambar, menjadikannya sulit dikenali dan diakses tanpa kunci yang tepat. Penggunaan entropi sebagai ukuran kerandoman efektif dalam menganalisis perubahan yang terjadi pada gambar selama proses enkripsi dan dekripsi.

## 4.2 Saran

Berdasarkan hasil analisis ini, beberapa saran dapat diberikan untuk pengembangan lebih lanjut: (i) Mengingat algoritma RSA memiliki keterbatasan dalam ukuran data yang dapat dienkripsi sekaligus, disarankan untuk mengeksplorasi metode enkripsi yang lebih efisien untuk gambar yang berukuran besar, seperti kombinasi RSA dan AES (*Advanced Encryption Standard).* (ii) Peningkatan keamanan kunci menggunakan kunci yang lebih panjang dan kompleks dalam algoritma RSA dapat meningkatkan keamanan lebih lanjut terhadap serangan kriptografi. Hal ini dapat dilakukan dengan memperluas rentang nilai untuk p dan q dalam generasi kunci RSA. (iii) Integrasi teknik ini dengan sistem keamanan yang lebih luas, seperti sistem autentikasi dan otorisasi yang kuat, akan memberikan perlindungan tambahan terhadap akses tidak sah. (iv) Penelitian lebih lanjut dapat dilakukan untuk mengoptimalkan performa enkripsi dan dekripsi, terutama untuk gambar dengan resolusi tinggi atau dalam skala besar. (v) Uji cobalah pada berbagai jenis gambar dan format yang berbeda akan membantu dalam memahami lebih baik keandalan dan fleksibilitas metode ini dalam berbagai skenario. Dengan saran-saran ini, diharapkan teknik pengacakan dan enkripsi gambar dapat lebih disempurnakan dan diimplementasikan secara efektif dalam praktik keamanan digital. (vi) Studi khasus lebih lanjut dengan menggunakan gambar yang lebih besar dan beragam, untuk mengevaluasi kinerja dan keamanan dari pendekatan yang diusulkan. Untuk peneliti kedepannya bisa lebih teliti lagi untuk menreapkan algoritma RSA dalam enkripsi dan dekripsi pada cittra digital.

## DAFTAR RUJUKAN


Alfaozi, I. (2021) 'Aplikasi Algoritma RSA dalam Enkripsi dan Dekripsi Gambar', *Makalah IF2120 Matematika Diskrit* [Preprint].

Azhar, J.K. & Yuliany, S. (2019) 'Implementasi Algoritma RSA ( Rivest , Shamir dan', (December).

Baharsyah, M.M.I., Bandung, I.T. & Bandung, J.G. (2023) 'Implementasi Algoritma RSA dalam Enkripsi dan Dekripsi File Teks'.

Christian, C., Sitorus, S.H. & Nirmala, I. (2023) 'Coding : Jurnal Komputer dan Aplikasi IMPLEMENTASI ALGORITMA RSA DAN ONE TIME PASSWORD (OTP) UNTUK PENGAMANAN DATA PENGGUNA DAN PROSES TRANSAKSI PADA WEBSITE E-COMMERCE [1] Calvin Christian, [2] Sampe Hotlan Sitorus, [3] Irma Nirmala', *Coding : Jurnal Komputer dan Aplikasi*, 11(1), pp. 62–72.

Deskiva, Z.Z. (2018) 'Implementasi Kriptografi Modern Dengan Metode Rsa Pada Data Citra Digital', *Publikasi Ilmiah Teknologi Informasi Neumann*, 3(1), pp. 44–49.

Fakhrizal, F. *et al.* (2023) 'Implementasi Security System Menggunakan Kriptografi Algoritma Simetris Untuk Pengamanan Video', *Bigint Journal of Computing*, 1(1), pp. 9–18. Available at: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=FP0rNUcAAAAJ&pagesize=100&citation_for_view=FP0rNUcAAAAJ:TQgYirikUcIC.

Harbani, A. & Fahreza, M.A. (2019) 'Aplikasi Keamanan Data Gambar Menggunakan Algoritma RSA (Rivest Shamir Adleman) Berbasis Desktop', *Teknois : Jurnal Ilmiah Teknologi Informasi dan Sains*, 9(1), pp. 1–9. Available at: https://doi.org/10.36350/jbs.v9i1.1.

Hendrawaty, H., TB, D.R.Y. & Munawir, M. (2022) 'Analisis Hasil Enkripsi Dan Dekripsi Citra Rgb 24 Bit Menggunakan Algoritma Elgamal Berdasarkan Ukuran, Dan Warna Citra Asli', *Journal of Informatics ...*, 8(1), pp. 12–16. Available at: http://jurnal.uui.ac.id/index.php/jics/article/view/2041%0Ahttp://jurnal.uui.ac.id/index.php/jics/article/download/2041/1113.

Khamsyar, A. & Basri, M. (2022) 'Aplikasi Enkripsi Gambar Menggunakan Metode (Rivest Shamir Adleman) Rsa', *Jurnal Sintaks Logika*, 2(3), pp. 39–45. Available at: https://doi.org/10.31850/jsilog.v2i3.1850.

Nazir, Y., Arnellis & Dewi, M.P. (2019) 'Penerapan Algoritma Rivest Shamir Adleman (RSA) untuk File Citra Menggunakan Visual Basic', *Jurnal UNPjoMath*, 2(4), pp. 61–66.

Safarina, N. & Shamir, A. (2017) 'Penerapan Algoritma Rsa Dan Des Pada Pada Pengamanan File Teks', *Pelita Informatika Budi Darma*, XVI(1), pp. 55–60.

Saputro, T.H., Hidayati, N.H. & H. Ujianto, E.I. (2020) 'Survei Tentang Algoritma Kriptografi Asimetris', *Jurnal Informatika Polinema*, 6(2), pp. 67–72. Available at: https://doi.org/10.33795/jip.v6i2.345.

Sutejo, S. (2021) 'Implementasi Algoritma Kriptografi Rsa (Rivest Shamir Adleman) Untuk Keamanan Data Rekam Medis Pasien', *INTECOMS: Journal of Information Technology and Computer Science*, 4(1), pp. 104–114. Available at: https://doi.org/10.31539/intecoms.v4i1.2437.

Trisnawati, T.T. *et al.* (2023) 'Penerapan Algoritma Rivest-Shamir-Adleman (RSA) pada Enkripsi Uniform Resource Locator (URL) Website untuk Keamanan Data', *Euler : Jurnal Ilmiah Matematika, Sains dan Teknologi*, 11(2), pp. 205–215. Available at: https://doi.org/10.37905/euler.v11i2.21169.

Yudanto, Y.S. & Suartana, I.M. (2022) 'Analisis Kekuatan Enkripsi Data Pada Citra Digital Menggunakan Metode Rubiks Cube', *Journal of Informatics and Computer Science (JINACS)*, 3(04), pp. 557–563. Available at: https://doi.org/10.26740/jinacs.v3n04.p557-563.

# LAMPIRAN

## Lampiran 1 Surat Izin Penelitian

UMKT
Program Studi
Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Muhammadiyah Kalimantan Timur
Berkarakter | Berwawasan | Berkemajuan
Telp. 0541-748511 Fax.0541-766832
Website http://informatika.umkt.ac.id
email: informatika@umkt.ac.id

بسم الله الرحمن الرحيم

Nomor : 056-009/KET/FST.1/A/2024
Lampiran : -
Perihal : **Keterangan Melakukan Penelitian**

*Assalamu'alaikum Warrahmatullahi Wabarrakatuh*

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Amin.

Dengan surat ini, kami menerangkan bahwa mahasiswa berikut:

| No | Nama | NIM |
|---|---|---|
| 1 | Amelda Auniyah | 1911102441019 |
| 2 | Winda Amelia Pratiwi | 1911102441105 |
| 3 | Ima Sulistia Nopi Wulandari | 1911102441173 |
| 4 | Nadhiya Safira Arrahmah | 1911102441112 |

Melakukan penelitian dengan citra digital menggunakan algoritma kriptografi kunci publik.

Demikian hal ini disampaikan, atas kerjasamanya kami ucapkan terima kasih.

*Wassalamu'alaikum Warrahmatullahi Wabarrakatuh*

Samarinda, 28 Dzulhijjah 1445 H
5 Juli 2024 M

Ketua Program Studi S1 Teknik Informatika

**Arbansyah, S.Kom., M.TI**
**NIDN. 1118019203**

**Lampiran 2 Citra Digital**

**Lampiran 3 Kartu Kendali Bimbingan**

**KARTU KENDALI BIMBINGAN LAPORAN SKRIPSI**

Nama Mahasiswa : Amelda Auniyah

NIM : 1911102441019

Nama Dosen Pembimbing : Sayekti Harits Suryawan, S.kom., M.Kom.

Judul Penelitian : PENERAPAN ALGORITMA *RSA* PADA CITRA DIGITAL

| No | Tanggal | Uraian Pembimbingan | Paraf Dosen |
|---|---|---|---|
| 1 | 7/03/2024 | Membahas alur-alur penulisan pada bab1 dan metode yang digunakan | |
| 2 | 24/04/2024 | Membahas untuk latar belakang, rumusan masalah, tujuan penelitian, dan manfaat penelitian. | |
| 3 | 05/05/2024 | Memperbaiki di latar belakang bagian penelitian sebelumnya (Alfaozi, 2021 | |
| 4 | 07/05/2024 | Membahas penjelasan pada isi dari bab 2 metode penelitian | |
| 5 | 08/05/2024 | Perbaikan diagram pada bab 2 dan menjelaskan isi diagram tersebut | |
| 6 | 10/05/2024 | Perbaikan proposal pada bab 2 pada penjelasan diagram | |
| 7 | 13/05/2024 | Revisi penjelasan dan pembahasan untuk memproses coding | |
| 8 | 15/05/2024 | Acc proposal pada bab1 dan bab 2 untuk riview desk | |
| 9 | 05/06/2024 | Menambahkan entropi pada bab 2 dan revisian proposal setelah selesai review desk | |
| 10 | 26/06/2024 | Pembahasan coding untuk bab 3 dan penyusunan | |
| 11 | 28/06/2024 | Memperbaiki coding untuk penerapan algoritma | |

Dosen Pembim bing

(Sayekti Harits Suryawan, S.kom., M.Kom)
NIDN:1119048901

Ketua Program Studi

(Arbansyah, S.Kom., M.TI)
NIDN.1119019203

## Lampiran 4 Seluruh Coding Python

```python
# Fungsi Extended Euclidean Algorithm
def extended_gcd(a, b):
    if a == 0:
        return b, 0, 1
    g, x1, y1 = extended_gcd(b % a, a)
    x = y1 - (b // a) * x1
    y = x1
    return g, x, y

# Fungsi untuk menghasilkan kunci publik dan privat RSA
def generate_rsa_keys():
    p = sympy.randprime(11, 20)
    q = sympy.randprime(11, 20)
    while p == q:
        q = sympy.randprime(11, 20)

    n = p * q
    phi = (p - 1) * (q - 1)
```

```python
    e = random.randrange(2, phi)
    g = sympy.gcd(e, phi)
    while g != 1:
        e = random.randrange(2, phi)
        g = sympy.gcd(e, phi)

    d = mod_inverse(e, phi)

    public_key = (e, n)
    private_key = (d, n)

    return public_key, private_key

# Fungsi untuk enkripsi pesan menggunakan RSA
def rsa_encrypt(message, public_key):
    e, n = public_key
    cipher = pow(message, e, n)
    return cipher
```

```python
# Fungsi untuk dekripsi pesan menggunakan RSA
def rsa_decrypt(cipher, private_key):
    d, n = private_key
    message = pow(cipher, d, n)
    return message

# Fungsi untuk menghitung entropi dari distribusi nilai piksel dalam gambar
def calculate_entropy(image):
    pixels = np.array(image)
    histogram, _ = np.histogram(pixels, bins=256, range=(0, 256))
    histogram = histogram.astype(float) / pixels.size
    histogram = histogram[histogram != 0]
    entropy = -np.sum(histogram * np.log2(histogram))
    return entropy
```

```python
# Fungsi untuk mengacak gambar
def scramble_image(image):
    random.seed(1234)  # Seed untuk hasil yang deterministik
    pixels = image.load()
    width, height = image.size
    scramble_map = {}

    for x in range(width):
        for y in range(height):
            r, g, b = pixels[x, y]
            r_scrambled = (r + random.randint(0, 255)) % 256
            g_scrambled = (g + random.randint(0, 255)) % 256
            b_scrambled = (b + random.randint(0, 255)) % 256
            pixels[x, y] = (r_scrambled, g_scrambled, b_scrambled)
            scramble_map[(x, y)] = (r_scrambled - r, g_scrambled - g, b_scrambled - b)

    return image, scramble_map
```

```python
# Fungsi untuk mengembalikan pengacakan gambar
def unscramble_image(image, scramble_map):
    pixels = image.load()
    width, height = image.size

    for x in range(width):
        for y in range(height):
            r, g, b = pixels[x, y]
            r_diff, g_diff, b_diff = scramble_map[(x, y)]
            r_unscrambled = (r - r_diff) % 256
            g_unscrambled = (g - g_diff) % 256
            b_unscrambled = (b - b_diff) % 256
            pixels[x, y] = (r_unscrambled, g_unscrambled, b_unscrambled)

    return image
```

```python
# Function for image data extraction
def extract_image_data(image):
    image_data = {
        "Ukuran": image.size,
        "Mode Warna": image.mode,
        "Format": image.format,
    }
    return image_data

# Fuction to upload image
def upload_and_process_image():
    upload = FileUpload(accept='image/*', multiple=False)
    container = VBox([upload])

    def on_upload_change(change):
        for filename, file_info in upload.value.items():
            img = Image.open(BytesIO(file_info['content']))
            display(img)

            # Ekstraksi data gambar
            image_data = extract_image_data(img)
            print("Data Gambar:", image_data)
```

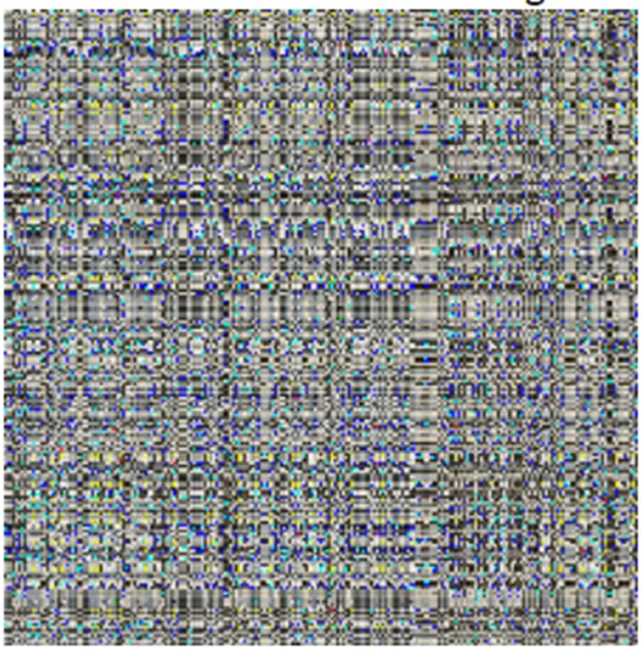
```python
            # Ekstraksi data gambar
            image_data = extract_image_data(img)
            print("Data Gambar:", image_data)

            # Generate RSA keys
            public_key, private_key = generate_rsa_keys()
            print('Public Key:', public_key)
            print('Private Key:', private_key)

            # Hitung entropi gambar asli
            original_entropy = calculate_entropy(img)
            print(f'Original Image Entropy: {original_entropy:.2f}')

            # Scramble image
            print('Scrambling image...')
            scrambled_img, scramble_map = scramble_image(img.copy())
            display(scrambled_img)

            # Hitung entropi gambar setelah diacak
            scrambled_entropy = calculate_entropy(scrambled_img)
            print(f'Scrambled Image Entropy: {scrambled_entropy:.2f}')
```

```python
            # Hitung entropi gambar setelah diacak
            scrambled_entropy = calculate_entropy(scrambled_img)
            print(f'Scrambled Image Entropy: {scrambled_entropy:.2f}')

            # Encrypt scrambled image
            print('Encrypting image...')
            encrypted_img = scrambled_img.copy()
            pixels = encrypted_img.load()
            width, height = encrypted_img.size

            for x in range(width):
                for y in range(height):
                    r, g, b = pixels[x, y]
                    r_encrypted = rsa_encrypt(r, public_key) % 256
                    g_encrypted = rsa_encrypt(g, public_key) % 256
                    b_encrypted = rsa_encrypt(b, public_key) % 256
                    pixels[x, y] = (r_encrypted, g_encrypted, b_encrypted)

            display(encrypted_img)

            # Hitung entropi gambar setelah enkripsi
            encrypted_entropy = calculate_entropy(encrypted_img)
            print(f'Encrypted Image Entropy: {encrypted_entropy:.2f}')
```

```
            # Decrypt encrypted image
            print('Decrypting image...')
            decrypted_img = encrypted_img.copy()
            pixels = decrypted_img.load()

            for x in range(width):
                for y in range(height):
                    r_encrypted, g_encrypted, b_encrypted = pixels[x, y]
                    r_decrypted = rsa_decrypt(r_encrypted, private_key) % 256
                    g_decrypted = rsa_decrypt(g_encrypted, private_key) % 256
                    b_decrypted = rsa_decrypt(b_encrypted, private_key) % 256
                    pixels[x, y] = (r_decrypted, g_decrypted, b_decrypted)

            display(decrypted_img)

            # Hitung entropi gambar setelah dekripsi
            decrypted_entropy = calculate_entropy(decrypted_img)
            print(f'Decrypted Image Entropy: {decrypted_entropy:.2f}')

            # Unscramble decrypted image
            print('Unscrambling image...')
            unscrambled_img = unscramble_image(decrypted_img.copy(), scramble_map)
            display(unscrambled_img)
```

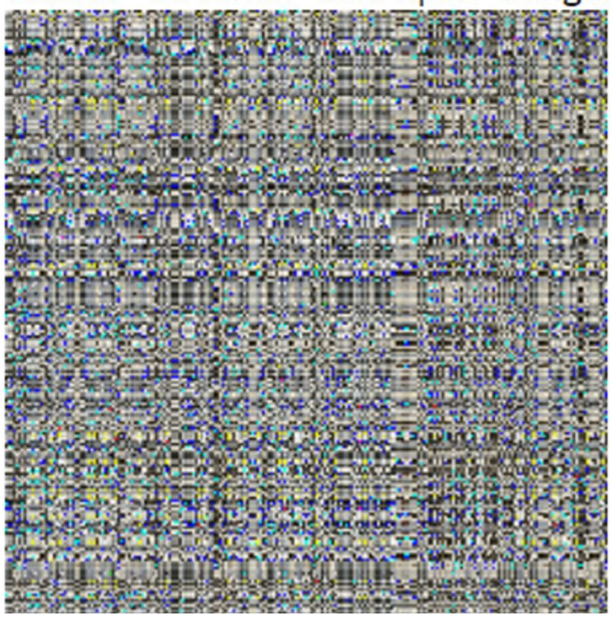
```
            # Hitung entropi gambar setelah unscrambling
            unscrambled_entropy = calculate_entropy(unscrambled_img)
            print(f'Unscrambled Image Entropy: {unscrambled_entropy:.2f}')

            # Pastikan entropi setelah unscrambling sama dengan entropi asli
            if np.isclose(original_entropy, unscrambled_entropy, atol=0.01):
                print('Entropy check passed: The entropies are close enough.')
            else:
                print('Entropy check failed: The entropies differ.')

    upload.observe(on_upload_change, names='value')
    display(container)

print('Upload an image to scramble and encrypt using RSA:')
upload_and_process_image()
```

Gambar setelah scrambling:

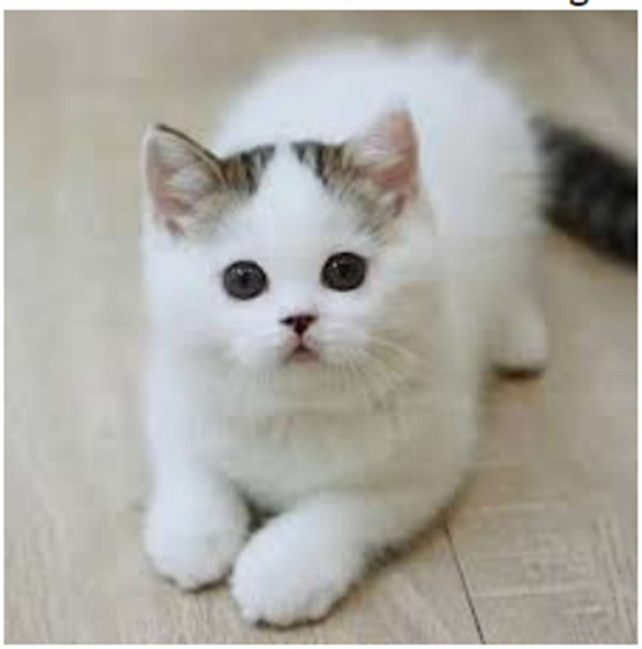


Encrypting image...

Gambar setelah dekripsi dengan RSA:



Gambar setelah unscrambling:



# RIWAYAT HIDUP

Amelda Auniyah, lahir di Jone, pada tanggal 08 April 2001, anak pertama dari 2 saudara, anak dari pasangan Bapak Aliansyah dan Ibu Juraiyah. Pada umur 6 tahun penulis memulai Pendidikan Sekolah Dasar di SDN 009 Tanah Grogot dan memperoleh ijazah pada tahun2013. Kemudian melanjutkan ke SMP Negeri 3 Tanah Grogot dan lulus pada tahun 2016. Kemudian melanjutkan Pendidikan SMA Muhammadiyah Tanah Grogot dan lulus pada tahun 2019. Kemudian melanjutkan Pendidikan di perguruan tinggi dimulai tahun 2019 di Fakultas Sains dan Teknologi dengan Program Studi Teknik Informatika Universitas Muhammadiyah Kalimantan Timur.

Selama dibangku perkuliahan Penulis aktif dalam kegiatan Universitas, program studi dan organisasi Ikatan Muhammadiyah Kalimantan Timur (IMM). Dengan semangat dan diiringi doa dalam menjalani aktivitas akademik di UMKT, Alhamdulillah penulis menyelesaikan bangku perkuliahan pada tahun 2024 dengan merampungkan skripsi yang berjudul "PENERAPAN ALGORITMA *RSA* PADA CITRA DIGITAL"