# BAB I

## PENDAHULUAN

### A. Latar Belakang

Informasi merupakan kebutuhan setiap orang dalam melakukan pekerjaan sehari-hari. Teknologi informasi ini berkembang pesat dan mempengaruhi hampir seluruh aspek kehidupan manusia. Terutama di dalam pengamanan distribusi dokumen pada suatu jaringan. Salah satunya adalah citra digital. Citra digital telah digunakan secara luas dalam berbagai macam proses sehingga perlindungan citra digital dari pihak yang tidak memiliki hak akses menjadi sangat penting. Hampir semua informasi ini dikumpulkan dan disimpan dalam komputer kemudian dikirimkan melalui jaringan, misalnya internet. Bila informasi penting ini jatuh ke tangan orang yang salah, maka akan menyebabkan hal yang tidak diinginkan, seperti manipulasi gambar dengan bentuk yang negatif yang dapat merugikan pemilik gambar. Hal inilah yang menyebabkan perlindungan citra digital menjadi sangat penting. Salah satu metode yang digunakan untuk menjaga keamanan data tersebut adalah kriptografi.

Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Secara garis besar,

proses enkripsi adalah proses pengacakan “naskah asli”(*plaintext*) menjadi “naskah acak” (*ciphertext*) yang sulit untuk dibaca oleh seseorang yang tidak mempunyai kunci dekripsi.

Kriptografi memiliki dua jenis algoritma jika ditinjau dari kunci yang digunakan, yaitu algiritma simetris dan asimetris. Algoritma simetris merupakan algoritma yang menggunakan kunci yang sama pada proses enkripsi dan dekripsinya. Sedangkan Algoritma asimetris merupakan algoritma yang memiliki kunci yang berbeda untuk proses enkripsi dan dekripsinya. Salah satu contoh algoritma simetris adalah algoritma RC4 (*Riverst Code* 4), dan salah satu contoh algoritma asimetris adalah algoritma elgamal.

Berdasarkan latar belakang yang telah penulis uraikan maka penulis melakukan penelitian yang berjudul “Enkripsi Gambar Menggunakan Algoritma RC4 dan Elgamal”.

## B. Rumusan Masalah

Berdasarkan dari uraian latar belakang diatas, maka rumusan masalah yang akan dibahas adalah bagaimana cara enkripsi dan dekripsi dalam mengamankan citra digital menggunakan algoritma RC4 dan Elgamal ?

## C. Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah dengan harapan penelitian terfokus dengan batasan-batasan yang dibuat. Adapun batasan dalam penelitian ini sebagai berikut :

1. Sistem yang dirancang hanya untuk mengenkripsi dan mendekripsi gambar.
2. Sistem yang dirancang menggunakan aplikasi MATLAB.

3. Algoritma yang digunakan adalah RC4 dan Elgamal.
4. Gambar yang dienkripsi adalah gambar 2 Dimensi.
5. Gambar yang dienrkipsi adalah logo perusahaan.
6. Menggunakan citra maksimal berukuran 300x300 piksel.
7. Gambar yang digunakan adalah gambar yang bereksistensi bmp.

## D. Tujuan Penelitian

Sesuai dengan permasalahan yang telah diuraikan diatas, maka tujuan dari penelitian ini adalah untuk menerapkan algoritma RC4 dan Elgamal pada proses enkripsi dan dekripsi dalam mengamankan citra digital.

## E. Manfaat Penelitian

Manfaat dari penelitian ini adalah untuk dapat memberikan solusi bagi masalah pengenkripsian dan pendekripsian dalam mengamankan citra digital.

## F. Sistematika Penulisan

Sistematika yang digunakan dalam penyusunan proposal ini adalah sebagai berikut :

**BAB I PENDAHULUAN**

Bagian ini akan menggambarkan dasar masalah, perincian masalah, batasan masalah, tujuan penelitian, manfaat penelitian, strategi pemilihan informasi dan sistematika penyusunan.

**BAB II TINJAUAN PUSTAKA**

Dalam bab ini berisikan beberapa teori-teori yang mendukung dalam pembahasan penyusunan tugas akhir ini serta bahasa pemrograman yang digunakan sehingga memudahkan penulis dalam menyelesaikan masalah.

**BAB III METODE PENELITIAN**

Pada bab ini menjelaskan mengenai tahapan - tahapan yang dilalui dalam penyelesaian tugas akhir ini, yaitu tempat penelitian dan waktu penelitian, metode pengumpulan data, alat dan bahan penelitian, tahap penelitian, metode pengujian serta gambaran desain sistem yang akan dirancang atau dibuat.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini menjelaskan implementasi sistem dan pengujian sistem dalam menjalankan aplikasi yang digunakan dan hasilnya serta pembahasan pengujiannya.

**BAB V PENUTUP**

Bagian ini menggambarkan tujuan dan gagasan.

# BAB II

# TINJAUAN PUSTAKA

## A. Tinjauan Penelitian Terdahulu

1. "*Analisa Algoritma Kriptografi RC4 Pada Enkripsi Citra Digital* ". Pada penelitian ini menggunakan algoritma RC4 untuk mengenkripsi dan mendekripsi citra digital. Penelitian ini berfokus pada kecepatan yang dibutuhkan oleh algoritma RC4 dalam mengenkripsi dan mendekripsi suatu citra digital. Kelebihan dari penelitian ini adalah mampu mengenkripsi dan mendekripsi tiga jenis eksistensi gambar. Selain itu pada penelitian ini juga dapat mengenkripsi dan mendekripsi gambar dengan ukuran 500x500 piksel. Adapun kekurangan dari penelitian ini adalah hanya berfokus pada perhitungan waktu yang dibutuhkan dalam mengenkripsi dan mendekripsi suatu citra digital. Kekurangan lain dalam penelitian adalah hanya menggunakan satu algoritma saja.
2. " *Analisis Kemampuan Algoritma Elgamal Untuk Kriptografi Citra* ". Pada penelitian ini menggunakan algoritma elgamal untuk mengenkripsi dan mendekripsi citra digital. Kelebihan dari penelitian ini adalah menggunakan metode enkripsi dengan *blocksize*, sehingga semakin sedikit pembagian *blocksize* yang digunakan maka waktu enkripsi dan dekripsi semakin cepat. Adapun kekurangan dari penelitian ini adalah ukuran dari hasil enkripsi

citra digital memiliki ukuran dua kali lebih besar dari pada citra aslinya, sehingga memakan memory yang cukup besar.

3. "*Aplikasi Enkripsi Citra Dan Teks Menggunakan Algoritma Diffie-Hellman dan Elgamal* ". Pada penelitian ini menggabungkan dua algoritma sekaligus dalam mengamankan *file* citra dan teks. Kelebihan dari penelitian ini adalah menggunakan dua algoritma didalam melakukan enkripsi dan dekripsi, sehingga tingkat kesulitan semakin tinggi. Adapun kekurangan dari penelitian ini adalah menggunakan dua algoritma yang hampir mirip, yaitu sama-sama menggunakan metode tukar kunci, sehingga kemungkinan bocornya kunci yang digunakan semakin besar.

Berdasarkan penelitian terdahulu yang diuraikan diatas, maka peneliti bermaksud membuat suatu sistem yang dapat mengenkripsi dan mendekripsi suatu citra digital dengan menggunakan dua algoritma, yaitu RC4 dan Elgamal.

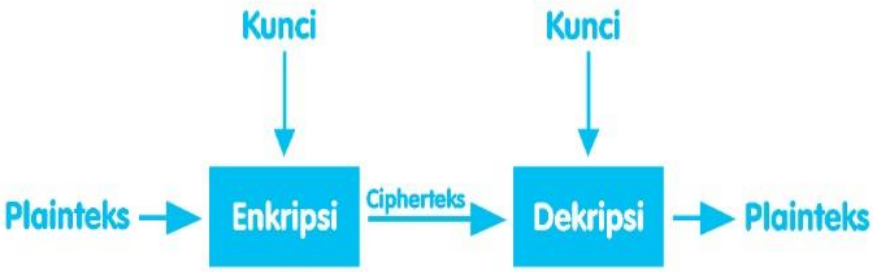
## B. Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu *kripto* dan *graphia*. *Kripto* artinya menyembunyikan, sedangkan *graphia* artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi. Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi.

## 1. Terminologi

Ketika melakukan pertukaran informasi, sebuah pesan asli dikirim oleh pengirim (*sender*) ke penerima (*recipient*) dengan harapan pesan asli tersebut sampai ke tangan penerima dengan aman dan kerahasiaannya terjaga.

Pesan asli disebut dengan plainteks (*plaintext*) atau teks-jelas (*cleartext*). Agar pesan tidak dapat diketahui artinya oleh pihak lain, maka pesan perlu disandikan ke bentuk lain. Bentuk pesan yang sudah tersandi disebut cipherteks (*ciphertext*) atau kriptogram (*cryptogram*). Cipherteks harus dapat diubah kembali ke bentuk plainteks agar si penerima dapat membaca pesan tersebut. Proses mengubah plainteks ke dalam bentuk cipherteks disebut dengan enkripsi (*encryption*). Sedangkan proses mengembalikan cipherteks ke dalam bentuk plainteks disebut dengan dekripsi (*decryption*). Skema kriptografi dapat dilihat pada gambar 2.1.




**Gambar 2.1** Skema Proses Enkripsi dan Dekripsi Menggunakan Kunci

## 2. Komponen kriptografi

Pada dasarnya komponen kriptografi terdiri dari beberapa komponen, yaitu:

### a. Enkripsi

Merupakan cara pengamanan data yang dikirimkan sehingga terjaga kerahasiaannya. Pesan asli disebut *plaintext* (teks biasa), yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *cipher* atau kode.

### b. Dekripsi

Merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunakan untuk enkripsi.

### c. Kunci

Merupakan kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, yaitu kunci rahasia *(private key*) dan kunci umum (*public key*).

### d. *Ciphertext*

Merupakan suatu pesan yang telah melalui proses enkripsi. Pesan yang ada pada teks-kode ini tidak bisa dibaca karena berupa karakter-karakter yang tidak mempunyai makna atau arti.

### e. *Plaintext*

Sering disebut dengan *cleartext*. Teks-asli atau teks-biasa ini merupakan pesan yang ditulis atau diketik yang memiliki makna. Teks-asli inilah yang diproses menggunakan algoritma kriptografi untuk menjadi *ciphertext* (teks-kode).

**f. Pesan**

Dapat berupa data atau informasi yang dikirim (melalui kurir, saluran komunikasi data, dsb) atau yang disimpan di dalam media perekaman (kertas, dsb).

**g. *Cryptanalysis***

Bisa diartikan sebagai analisis kode atau suatu ilmu untuk mendapatkan teks-asli tanpa harus mengetahui kunci yang sah secara wajar.

**3. Tujuan kriptografi**

Terdapat 4 tujuan dasar kriptografi sebagai berikut :

**a. Kerahasiaan (*confidentiality*)**

Adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak. Di dalam kriptografi, layanan ini direalisasikan dengan menyandikan pesan menjadi *ciphertext*.

**b. Integritas data (*data integrity*)**

Adalah layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman.

**c. Otentikasi (*authentication*)**

Adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*) maupun mengidentifikasi kebenaran sumber pesan ( *date origin authentication*).

**d. Nirpenyangkalan (*non-repudiation*)**

Adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

**4. Kriptografi kunci simetri dan asimetri**

**a. Algoritma simetri**

Algoritma simetris merupakan algoritma yang menggunakan kunci yang sama pada proses enkripsi dan dekripsinya. Algoritma simetris juga biasa disebut algoritma konvensional. Sistem kriptografi dengan algoritma simetris ini mengasumsikan pengirim dan penerima pesan sudah berbagi kunci yang sama sebelum bertukar pesan. Keamanan algoritma simetris ini terletak pada kerahasiaan kuncinya. Beberapa sistem kriptografi yang menggunakan algoritma simetris adalah *One Time Pad*, RC4, *Data Encryption Standard* (DES), *Advanced Encryption Standard* (AES).

**b. Algoritma asimetri**

Algoritma asimetris merupakan algoritma yang memiliki kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma ini biasa juga disebut dengan algoritma kunci publik karena kunci untuk proses enkripsi bersifat publik, artinya siapapun bisa menggunakan kunci tersebut untuk melakukan enkripsi. Sementara itu, kunci untuk mendekripsikan pesan bersifat privat sehingga hanya si penerima pesan yang dapat mendekripsikan pesan karena hanya ia yang mengetahui kunci privatnya sendiri. Contoh sistem kriptografi yang menggunakan algoritma kunci publik adalah RSA (*Rivest Shamir Adleman*), ElGamal dan LUC.

**5. Kriptografi kunci publik**

Konsep kriptografi kunci publik sederhana dan elegan. Pada kriptografi kunci publik, setiap pengguna memiliki sepasang kunci, satu kunci digunakan untuk enkripsi dan satu kunci digunakan untuk dekripsi. Kunci yang digunakan untuk enkripsi diumumkan kepada publik oleh karena itu tidak rahasia dan dinamakan dengan *key publik.* Sedangkan untuk kunci dekripsi bersifat rahasia karena yang hanya mengetahuinya hanyalah pengirim dan penerima pesan, sehingga dinamakan *private key*.

**6. Teori bilangan**

Teori bilangan adalah cabang dari ilmu matematika murni yang mempelajari sifat-sifat bilangan bulat dan mengandung berbagi masalah terbuka yang dapat dengan mudah dimengerti oleh orang yang bukan ahli matematika sekalipun. Bilangan bulat adalah bilangan yang tidak mempunyai pecahan desimal, misalnya 8, 21, 8765, -34 dan 0. Teori bilangan kerap digunakan untuk mendasari pemahaman kriptografi, khususnya sistem kriptografi kunci publik. Pesan-pesan yang disandikan dapat direpresentasikan oleh bilang-bilangan yang dalam pemrosesannya menggunakan operasi matematika untuk mentransformasikan nilai *input* yang berupa bilanga bulat menjadi nilai *output* yang berupa bilangan bulat pula. Teori bilangan ini memiliki beberapa jenis bilangan yang sering digunakan pada algoritma-algoritma yang digunakan untuk mengamankan pesan, diantaranya adalah teori bilangan aritmetika modular atau dikenal dengan modular yaitu sisa hasil pembagian.

**a. Aritmetika modular**

Aritmetika modular merupakan operasi matematika yang kerap diimplementasikan pada metode kriptografi. Aritmetika modular adalah suatu perhitungan matematika yang berhunungan dengan hasil sisa pembagian. Dalam aritmetikam modular, pada bilangan bulat positif *n* dan bilangan bulat *m*, jika dilakukan pembagian *n* dan *m* dengan syarat $n > 0$, akan dihasilkan bilangan *q* dan sisa pembagian *r*. Operator yang digunakan pada aritmetika modular adalah mod sehingga notasi matematika dari aritmetika modular dapat dinyatakan sebagai *m* mod. Misalkan *a* adalah bilangan bulat dan *m* adalah bilangan bulat > 0. Operasi *a mod m* dibaca "*a modulo m*" memberikan sisa jika *a* dibagi dengan *m*.

## C. Citra Digital

Citra (*image*) adalah kombinasi antara titik, garis, bidang dan warna untuk menciptakan imitasi dari suatu objek. Citra bisa berwujud gambar dua dimensi, seperti lukisan, foto, dan berwujud tiga dimensi, seperti patung. Citra sebagai salah satu komponen multimedia memegang peranan sangat penting sebagai bentuk informasi visual. Sebuah gambar dapat memberikan informasi yang lebih banyak daripada informasi tersebut disajikan dalam bentuk kata-kata (tekstual).

Sebuah citra diubah ke dalam bentuk digital agar dapat disimpan ke dalam komputer atau media penyimpanan lain. Untuk mengubah citra ke dalam bentuk digital digunakan beberapa alat, seperti kamera digital, *scanner* dan *handycam*. Ada tiga jenis citra yang umum digunakan dalam pemrosesan citra. Ketiga jenis citra tersebut yaitu citra berwarna, citra berskala keabuan, dan citra biner.

Ada beberapa property yang dimiliki oleh citra digital :

1. Resolusi

Resolusi merupakan ukuran gambar dalam satuan *pixel*. Seperti yang dijelaskan sebelumnya *pixel* merupakan titik-titik warna pada gambar. Resolusi menentukan jumlah *pixel* dalam citra. Semakin besar resolusi, maka semakin besar dan detil dari kualitas citra tersebut.

2. Kumpulan warna / *pixel*

Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (*pixel* atau *picture element*) . Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi. Bagian terkecil dari citra digital adalah *pixel*. *Pixel* mewakili warna pada gambar. Salah satu representasi dari warna *pixel* yaitu RGB.

a. *Red, green, blue* (RGB)

Citra berwarna, atau biasa dinamakan citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna menggunakan delapan bit (nilainya berkisar antara 0 sampai dengan 255). Dengan demikian, kemungkinan warna yang dapat disajikan mencapai 255 x 255 x 255 atau 16.581.375 warna. Kebanyakan *computer* sekarang ini menggunakan representasi ini untuk mengedit atau melihat sebuah citra digital. Pembentukan warna berdasarkan 3 warna dasar, yaitu merah, hijau, dan biru.

Citra yang direpresentasikan dalam model wama RGB terdiri dari tiga komponen citra, masing-masing untuk setiap warna primer (R,G,B). Ketika ditampilkan di monitor RGB, tiga kombinasi citra ini berada di layar *fosfor* untuk menghasilkan warna citra komposit. Jumlah bit yang digunakan untuk

merepresentasikan setiap piksel dalam *space* RGB disebut *pixel depth*. Perhatikan bahwa citra RGB di mana Setiap citra *red*, *green*, dan *blue* adalah citra 8-bit.

Dalam kondisi setiap warna piksel RGB (maka, *triplet* dari Nilai (R,G,B)) mempunyai kedalaman 24-bit (3 lapis citra dengan jumlah bit per lapis). Citra *full-color* sering digunakan untuk menyatakan citra berwarna RGB 24-bit. Total jumlah warna dalam citra 24-bit adalah $(2^8)^3$ = 16,777,216.

Adapun dalam setiap gambar RGB, jika dipecah menjadi 3 bagian, masing-masing yaitu *red*, *green*, dan *blue*, maka nilai setiap lapisan tersebut yaitu :

R : 8-bit = 0 – 255

G : 8-bit = 0 – 255

B : 8-bit = 0 – 255

Oleh karena itu setiap warna yang ada pada masing-masing lapisan maksimal hanya memiliki nilai 255.

Didalam pengolahan citra digital, dikenal juga istilah operasi piksel atau kadang juga disebut operasi piksel ke piksel. Operasi piksel adalah operasi pengolahan citra yang memetakan hubungan setiap piksel yang bergantung pada piksel itu sendiri.

Untuk pemilihan *pixel* dalam proses enkripsi dan dekripsi ada beberapa metode yang dapat dilakukan :

1. *One-pixel* rgb

Pada metode ini, pemilihan dilakukan dengan memilih satu buah *pixel*. Jadi setiap proses melibatkan hanya satu *pixel* rgb.

2. *One-segment* rgb

Pada metode ini, pemilihan dilakukan hanya pada satu segment warna, misal merah. Proses enkripsi dan dekripsi dilakukan persegment pixel (satu pixel 3 kali proses untuk menciptakan *pixel chiper-citra*).

3. *N-pixel* rgb

Pada metode ini, pemilihan dilakukan ke beberapa *pixel* sekaligus. Jadi pada setiap proses melibatkan beberapa *pixel* rgb sekaligus.

*4. N-segment* rgb

Pemilihan dilakukan pada beberapa *pixel* sekaligus. Namun , pemrosesan dilakukan persegmen dari *pixel-pixel* tersebut. Jadi sederatan *pixel* tersebut diproses 3 kali sesuai dengan segmen pada rgb. Contoh : dipilih 2 *pixel* dengan nilai 0xffab12 dan 0xab13de. Maka pemrosesan 3 kali dengan nilai masing-masing ffab untuk nilai merah, ab13 hijau dan 12de untuk biru.

Beberapa format yang umum digunakan dalam pengolahan citra digital antara lain :

1. Bitmap (BMP)

Bitmap adalah representasi dari citra grafis yang terdiri dari susunan titik (*pixel*) yang tersimpan di memori komputer. Nilai setiap titik diawali oleh satu bit data (untuk gambar hitam putih) atau lebih (untuk gambar berwarna). Kerapatan titik-titik tersebut dinamakan resolusi, yang menunjukkan seberapa tajam gambar ini ditampilkan, ditunjukkan dengan jumlah baris dan kolom. Format bitmap ini cocok digunakan untuk menyimpan citra digital yang memiliki banyak variasi dalam bentuknya maupun warnanya, seperti foto, lukisan, dan *frame video*. Format

*file* yang menggunakan format bitmap ini antara lain adalah BMP, DIB, PCX, GIF, dan JPG. Format yang menjadi standar dalam sistem operasi *Microsoft Windows* adalah format bitmap BMP atau DIB.

Karakteristik lain dari bitmap yang juga penting adalah jumlah warna yang dapat disimpan dalam bitmap tersebut. Ini ditentukan oleh banyaknya bit yang digunakan untuk menyimpan setiap titik dari bitmap yang menggunakan satuan bpp (bit per *pixel*). Dalam *Windows* dikenal bitmap dengan 1, 4, 8, 16, dan 24 bit per *pixel*. Jumlah warna maksimum yang dapat disimpan dalam suatu bitmap adalah sebanyak 2n, dimana n adalah banyaknya bit yang digunakan untuk menyimpan satu titik dari bitmap.

Saat ini format BMP memang kalah populer dibandingkan dengan format JPG atau GIF. Hal ini karena berkas BMP pada umumnya tidak dimanfaatkan sehingga ukuran berkasnya relatif lebih besar daripada berkas JPG maupun GIF. Hal ini juga yang menyebabkan format BMP sudah jarang digunakan. Meskipun format BMP tidak mangkus dari segi ukuran berkas, namun format BMP mempunyai kelebihan dari segi kualitas gambar. Citra dalam format BMP lebih bagus daripada citra dalam format yang lainnya, karena citra dalam format BMP umumnya tidak dimampatkan sehingga tidak ada informasi yang hilang. Terjemahan bebas bitmap adalah pemetaan bit. Artinya, nilai intensitas *pixel* di dalam citra dipetakan disejumlah bit tertentu. Peta bit yang umum adalah 8, artinya setiap *pixel* panjangnya 8 bit. Delapan bit ini merepresentasikan nilai intensitas *pixel*. Dengan demikian ada sebanyak 28 = 256 derajat keabuan, mulai dari 0-255.

2. *Joint Photograpic Experts* (JPEG)

*Joint Photograpic Experts* (JPEG , dibaca jay-peg) di rancang untuk kompresi beberapa *full-color* atau *gray-scale* dari suatu gambar yang asli, seperti pemandangan asli di dunia ini. JPEG bekerja dengan baik pada *continous tone images* seperi *photographs* tetapi tidak terlalu bagus pada ketajaman gambar dan seni pewarnaan seperti penulisan, kartun yang sederhana atau gambar yang mengunakan banyak garis.

## D. Matlab

Matlab merupakan kependekan dari *matrix laboratory* dikarenakan setiap data pada matlab menggunakan dasar matriks. Matlab adalah bahasa pemrograman tinggi, tertutup, dan *case sensitive* dalam lingkungan komputasi numerik yang dikembangkan oleh *Mathworks*. Salah satu kelebihannya yang paling populer adalah kemampuan membuat grafik dengan *visualisasi* terbaik. Matlab mempunyai banyak *tools* yang dapat membantu berbagai disiplin ilmu. Ini merupakan salah satu penyebab industri menggunakan matlab. Selain itu matlab mempunyai banyak *library* yang sangat membantu untuk menyelesaikan permasalahan matematika seperti membuat simulasi fungsi, pemodelan matematika dan perancangan GUI.

Matlab merupakan suatu sistem interaktif yang memiliki elemen data dalam suatu *array* sehingga tidak lagi kita dipusingkan dengan masalah dimensi. Hal ini memungkinkan kita untuk memecahkan banyak masalah teknis yang terkait dengan komputasi, kususnya yang berhubungan dengan *matrix* dan formulasi *vektor,* yang mana masalah tersebut merupakan momok apabila kita harus menyelesaikannya dengan menggunakan bahasa level rendah seperti *Pascall*, C dan *Basic*.

Dalam lingkungan perguruan tinggi teknik, Matlab merupakan perangkat standar untuk memperkenalkan dan mengembangkan penyajian materi matematika, rekayasa dan keilmuan. Di industri, matlab merupakan perangkat pilihan untuk penelitian dengan produktifitas yang tinggi, pengembangan dan analisanya. Fitur-fitur matlab sudah banyak dikembangkan, dan lebih kita kenal dengan nama *toolbox*. Sangat penting bagi seorang pengguna Matlab, *toolbox* mana yang mandukung untuk *learn* dan *apply technologi* yang sedang dipelajarinya. *Toolbox-toolbox* ini merupakan kumpulan dari fungsi-fungsi matlab (*M-files*) yang telah dikembangkan ke suatu lingkungan kerja MATLAB untuk memecahkan masalah dalam kelas *particular*. Area-area yang sudah bisa dipecahkan dengan *toolbox* saat ini meliputi pengolahan sinyal, *system* kontrol, *neural networks*, *fuzzy logic*, *wavelets*, dan lain-lain.

Selain yang disebutkan diatas, matlab juga banyak digunakan pada pengolahan citra digital. Dalam matlab juga menyediakan banyak fitur mengenai citra digital, diantaranya untuk melihat nilai-nilai pada *pixel* maka dapat dilihat melalui *pixel region*, untuk mengubah gambar dari berwarna ke warna abu-abu dapat menggunakan fungsi-fungsi tertentu, dan masih banyak lagi fitur-fitur yang dapat digunakan dalam pengolahan citra digital.

Lingkungan kerja yang umum digunakan pada matlab :

1. *Command window*

*Command Window* adalah jendela yang digunakan untuk menuliskan atau mengeksekusi semua perintah matlab, seperti menuliskan perintah perhitungan sederhana, memanggil fungsi, mencari informasi (*help*) tentang sebuah fungsi dengan aturan penulisannya, demo program, dan sebagainya.

2. *Work space*

Jendela ruang kerja berisi informasi penggunaan variabel di dalam memori matlab.

3. *Command hsitory*

Jendelah *history* berisi informasi tentang perintah yang pernah dituliskan sebelumnya.

*4. Graphical user interface* (GUI)

*Graphical user interface* adalah sebuah aplikasi *display* dari matlab yang mengandung tugas, perintah, atau komponen program yang mempermudah *user* (pengguna) dalam menjalankan sebuah program dalam matlab.

*Graphical user interface* merupakan fitur interaktif yang melibatkan dua *file*, yaitu *figure file* dan *m-file*. *Figure file* memuat tampilan yang berisi *menu*, *button, radio button*, *drop-down list*, dan lain-lain. Adapun *m-file* diigunakan untuk memuat *script* untuk menampilkan dan mengontrol GUI itu sendiri.

Variabel dan *operator* yang digunakan dalam matlab :

Seperi bahasa pemrogrman lainnya, matlab pun memiliki variabel. Tetapi dalam penulisannya, variabel di dalam matlab tidak perlu dideklarasikan karena matlab mampu mengenali tipe data variabel dan isi variabel tersebut. Aturan penulisan variabel pada matlab bersifat *case sensitive*, diawali dengan huruf dan selanjutnya boleh menggunakan gabungan huruf-angka atau tanda garis bawah. matlab mampu mengenali sampai 31 karakter pertama nama variabel dan mengabaikan selanjutnya.

Semua tipe data di matlab memiliki bentuk yang sama, yaitu *array*. *Array* minimal berukuran 0 X 0 dan dapat bertambah menjadi *array* n X m dimensi dengan

sembarang ukuran. matlab mempunyai beberapa tipe data besar (atau *class*), yaitu *logical, char, numeric, cell, structure, java classes,* dan *function* logika

Operator pada MATLAB diklasifikasikan menjadi tiga bagian, yaitu operator aritmetika, operator relasional, dan operator logika.

1. Operator aritmetika

Operator Aritmetika digunakan untuk mengerjakan komputasi numerik. Operator Aritmetika yang dipake pada aplikasi MATLAB dapat dilihat pada tabel 2.1.

**Tabel 2.1** Operator Aritmetika

| OPERATOR | KETERANGAN |
|---|---|
| + | Penjumlahan |
| - | Pengurangan |
| * | Perkalian (aturan matriks) |
| .* | Perkalian masing-masing elemen yang bersesuaian (aturan *array*) |
| / | Pembagian kanan (matriks) |
| ./ | Pembagian kanan (*array*) |
| \ | Pembagian kiri (matriks) |
| .\ | Pembagian kiri (*array*) |
| ^ | Perpangkatan (matriks) |
| .^ | Perpangkatan (*array*) |
| : | Langkah |

2. Operator Rasional

Operator relasional digunakan untuk membandingkan dua operator secara kuantitatif seperti terlihat pada tabel 2.2.

**Tabel 2.2** Operator rasional

| OPERATOR | KETERANGAN |
|---|---|
| = = | Sama dengan |
| ~= | Tidak sama dengan |
| < | Kurang dari |
| > | Lebih dari |
| <= | Kurang dari sama dengan |

| OPERATOR | KETERANGAN |
|---|---|
| >= | Lebih dari sama dengan |

3. Operator logika

Operator logika yang digunakan pada aplikasi MATLAB dapat dilihat pada tabel 2.3.

**Tabel 2.3** Operator Logika

| OPERATOR | KETERANGAN |
|---|---|
| & | Akan menghasilkan nilai 1 jika kedua elemen yang bersesuaian memiliki nilia *true* dan 0 untuk yang lainnya |
| \| | Akan bernilai 1 jika salah satu elemennya *true* |
| ~ | Komplen dari elemen yang di-*input*-kan |
| *xor* | Akan bernilai 1 jika salah satu dari kedua elemen memiliki nilai berbeda. Dan bernilai 0 jika sama |

Adapun kelas integer pada matlab dapat dilihat pada tabel 2.4.

**Tabel 2.4** Operator Kelas Integer

| ***CLASS*** | ***RANGE OF VALUES*** | ***FUNCTION*** |
|---|---|---|
| *Signed 8-bit integer* | $-2^7$ to $2^7-1$ | Int8 |
| *Signed 16-bit integer* | $-2^{15}$ to $2^{15}-1$ | Int16 |
| *Signed 32-bit integer* | $-2^{31}$ to $2^{31}-1$ | Int32 |
| *Signed 64-bit integer* | $-2^{63}$ to $2^{63}-1$ | Int64 |
| *Unsigned 8-bit integer* | 0 to $2^8-1$ | Uint8 |
| *Unsigned 16-bit integer* | 0 to $2^{16}-1$ | Uint16 |
| *Unsigned 32-bit integer* | 0 to $2^{32}-1$ | Uint32 |
| *Unsigned 64-bit integer* | 0 to $2^{64}-1$ | Uint64 |

Adapun format gambar yang ada pada matlab dapat dilihat pada tabel 2.5.

**Tabel 2.5** Format Gambar

| **FORMAT** | **DEKRIPSI** | ***RECOGNIZED EXTENSION*** |
|---|---|---|
| TIFF | *Tagged image file* format | .tiff |
| JPEG | *Joint photographics expert's group* | .jpeg |
| GIF | *Graphics interchange format* | .gif |
| BMP | *Windows bitmap* | .bmp |
| PNG | *Portable network graphics* | .png |
| XWD | *X-window dump* | .xwd |

Selain yang telah disebutkan diatas, matlab juga menyediakan histogram. Dalam histogram selain untuk menampilkan grafik mengenai data-data, histogram juga dapat digunakan untuk menganalisis citra digital yang biasa disebut histogram citra. Histogram citra merupakan diagram yang menggambarkan frekuensi setiap nilai intensitas yang muncul di seluruh piksel citra. Nilai besar menyatakan bahwa piksel-piksel yang mempunyai intensitas tersebut sangat banyak.

Pada citra berskala keabuan, jumlah aras keabuan (biasa disimbolkan dengan L) sebanyak 256. Nilai aras dimulai dari O hingga 255. Adapun histogram untuk suatu aras dinyatakan dengan *hist* (k+1) dengan k menyatakan nilai aras (O sampai dengan L-1). Jadi, *hist* (k+1) menyatakan jumlah piksel bernilai k. Penggunaan k+1 pada *hist* diperlukan mengingat dalam *octave* dan matlab tidak ada indeks nol atau *hist* (0). Perlu diketahui, (0:255) untuk membentuk nilai dari 0,1,2, dan seterusnya sampai dengan 255. Dengan perkataan lain, (0:255) membentuk larik 1 x 256. Tanda koma atas di belakang (0:255) menyatakan operasi *transpos*. Hasilnya berupa larik berukuran 256 x 1. Perintah bar digunakan untuk membuat diagram batang.

Pada pengolahan citra, histogram mempunyai peran cukup penting. Manfaat yang dapat diperoleh seperti berikut :

1. Berguna untuk mengamati penyebaran intensitas warna dan dapat dipakai untuk pengambilan keputusan, misalnya dalam peningkatan kecerahan atau peregangan kontras serta sebaran warna.
2. Berguna untuk penentuan batas-batas dalam pemisahan objek dari latar belakangnya.
3. Memberikan persentase komposisi warna dan tekstur intensitas untuk kepentingan identifikasi Citra.
4. Dapat digunakan untuk mencari citra yang memiliki kemiripan komposisi warna.

## E. Algoritma RC4

Algoritma kriptografi *Rivest Code* 4 (RC4) merupakan salah satu algoritma kunci simetris dibuat oleh RSA Data *Security Inc* (RSADSI) yang berbentuk *stream chiper*. Algoritma ini ditemukan pada tahun 1987 oleh Ronald Rivest dan menjadi simbol keamanan RSA(merupakan singkatan dari tiga nama penemu: Rivest Shamir Adleman). RC4 menggunakan panjang kunci dari 1 sampai 256 *byte* yang digunakan untuk menginisialisasikan tabel sepanjang 256 *byte*. Tabel ini digunakan untuk generasi yang berikut dari *pseudo random* yang menggunakan XOR dengan plainteks untuk menghasilkan cipherteks. Masing-masing elemen dalam tabel saling ditukarkan minimal sekali.

RC4 merupakan salah satu jenis *stream cipher* sehingga RC4 memproses unit atau input data, pesan atau informasi pada satu saat. Unit atau data pada

umumnya sebuah *byte* atau bahkan kadang-kadang bit (*byte* dalam hal RC4) [BUD98] sehingga dengan cara ini enkripsi atau dekripsi dapat dilaksanakan pada panjang yang variabel. Algoritma ini tidak harus menunggu sejumlah input data, pesan atau informasi tertentu sebelum diproses, atau menambahkan *byte* tambahan untuk mengenkrip.

Algoritma RC4 menggunakan dua buah *S-Box* yaitu *array* sepanjang 256 yang berisi permutasi dari bilangan 0 sampai 255, dan *S-Box* kedua, yang berisi permutasi merupakan fungsi dari kunci dengan panjang yang variabel.

Cara kerja algoritma RC4 yaitu inisialisasi *S-Box* pertama, S[0], S[1],...,S[255], dengan bilangan 0 sampai 255. Pertama isi secara berurutan S[0] = 0, S[1] = 1,...,S[255] = 255. Kenudian inisialisasi *array* lain (*S-Box* lain), misal *array* K dengan panjang 256. Isi *array* K dengan kunci yang diulangi sampai seluruh *array* K[0], K[1],...,K[255] terisi seluruhnya.

Algoritma RC4 secara umum sebagai berikut :

1. Instalasi larik *state* sepanjang 256 *byte*.
2. Memilih kunci rahasia yang ingin digunakan (panjang kunci 1 *byte* sampai 256 *byte* , semakin panjang semakin baik).
3. Menjelankan algoritma KSA (*Key Scheduling Algorithm*) dan PGRA (*Pseudo Random Generation Algorithm*).
4. *Key stream* yang dihasilkan dioperasikan dengan operasi logika *XOR* dengan plainteks.
5. Proses tersebut menghasilkan chiperteks dan plainteks.

Terdapat dua tahapan untuk membangkitkan aliran kunci algoritma RC4 yaitu :

**1. *Key scheduling algorithm* (KSA)**

Merupakan tahapan pemberian nilai awal berdasarkan kunci enkripsi. *State* dari nilai awal tersebut berupa *array* dengan representasi permutasi 256 *byte* (dengan indeks 0 sampai dengan 255) dinamakan *array* S. Menggunakan rentang tersebut karena RC4 mengenkripsi pada mode *byte*(255 =28 dan 8 bit = 1 *byte*). Artinya maksimal panjang kunci yang dapat tersimpan pada *array* U adalah 256 karakter. Permutasi terhadap nilai *array* S dilakukan dengan *pseudo-code* berikut :

```
j <- 0
for i <- 0 to 255 do
  j <- (j + S[i] + K[i mod Lenght (K)] mod 256
  swap (S[i], S[j])
end
```

**Gambar 2.2** *Pseudo-code* KSA

**2. *Pseudo-random generator algorithm* (PRGA)**

Tahap selanjutnya hasil dari *array* yang telah melalui KSA akan diproses kembali pada PRGA(*Pseudo-Random Generator Algorithm*). Pada tahap PRGA terjadi modifikasi *state* dan *output* sebuah *byte* dari aliran kunci, dimana *array* S beroperasi dengan array U yang selanjutnya akan menghasilkan *keystream*. Nilai S[i] dan S[j] diambil dan dijumlahkan dengan *modulo* 256 untuk memembangkitkan aliran kunci. Hasil dari perhitungan tersebut akan menjadi indeks S[indeks] yang menjadi aliran kunci K yang kemudian digunakan untuk

mengenkripsi plainteks ke-aliran kunci K yang kemudian digunakan untuk mengenkripsi plainteks ke-*idx*.

Setiap putaran bagian *keystream* sebesar 1 *byte*(dengan nilai antara 0 sampai dengan 255) *dioutput*-kan oleh PRGA berdasarkan state S. Berikut adalah PRGA dalam bentuk *pseudo-code* :

```
i <- 0
j <- 0
for idx <- 0 to Length (P) – 1 do
    i <- (i + 1) mod 256
    j <- (j + S[i]) mod 256
    swap (S[i], S[j])    //pertukaran nilai S[i] dan S[j]
    t <- (S[i] + S[j]) mod 256
    u <- S[t]    //keystream
    c <- u XOR p [idx]    //operasi XOR untuk enkripsi plain-image
end
```

**Gambar 2.3** *Pseudo-code* PRGA

## F. Algoritma Elgamal

Algoritma ElGamal adalah sebuah algoritma untuk kriptografi kunci publik. Algoritma ini dibuat oleh Taher ElGamal pada tahun 1985. Algoritma ini memiliki keamanan yang terletak pada kesulitan dalam menghitung logaritma diskrit. Algoritma ElGamal merupakan algoritma dalam kriptografi yang termasuk dalam kategori algoritma asimetris. Keamanan algoritma ElGamal terletak pada kesulitan

penghitungan logaritma diskret pada bilangan *modulo* prima yang besar sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sukar.

Algoritma ElGamal tidak dipatenkan. Tetapi, algoritma ini didasarkan pada algoritma Diffie – Hellman, sehingga hak paten algoritma Diffie – Hellman juga mencakup algoritma ElGamal. Karena hak paten algoritma Diffie – Hellman berakhir pada bulan April 1997, maka algoritma ElGamal dapat diimplementasikan untuk aplikasi komersil.

Algoritma ElGamal mempunyai kunci publik berupa tiga pasang bilangan dan kunci rahasia berupa satu bilangan. Algoritma ini mempunyai kerugian pada *ciphertext*-nya yang mempunyai panjang dua kali lipat dari *plaintext*-nya. Adapun property bilangan elgamal ebagai berikut :

1. Bilangan prima, *p*.
2. Bilangan acak, *g* (dengan syarat $g < p$).
3. Bilangan acak privasi/rahasia, *x* (dengan syarat $x < g < p$).
4. $Y = g^x \bmod p$.
5. *Plainimage*.
6. Bilangan acak privasi/rahasia, *k* (dengan syarat $1< k < p-2$).
7. *Cipherimage*.

Algoritma ElGamal terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi.

**1. Pembangkita kunci**

Algoritma ElGamal merupakan algoritma kunci *public*. Dalam pembuatan kunci, ada beberapa langkah yang harus dilakukan.

a. Pilih sembarang bilangan prima, *p.*

b. Pilih 2 bilangan acak, yaitu *g* dan *x* .

c. Hitung $y = g^x \bmod p$.

Setelah proses tersebut, maka akan terbentuk dua buah kunci, yaitu kunci *public* (berisi *y*,*g*,*p*) dan kunci *private* (*x*,*p*).

**2. Proses enkripsi**

Dalam algoritma enkripsi, kunci yang digunakan adalah kunci *public* (*y*,*g*,*p*). Berikut tahapan-tahapan pada proses enkripsi :

a. Dapatkan *plainimage* yang ingin dienkripsi (m).

b. Pilih bilangan acak yang bersifat privasi/rahasia, *k.*

c. Lakukan enkripsi pada *plainimage*, $C = y^k m \bmod p$.

Setelah proses tersebut, maka C merupakan hasil enkripsi yang disebut *cipherimage.*

**3. Proses dekripsi**

Kunci untuk proses dekripsi adalah kunci privat (*x*,*p*). Berikut tahapan-tahapan pada proses dekripsi :

a. Hitung $a = g^k \bmod p$.

b. Gunakan *x* untuk menghitung *invers* a, $(a^x)^{-1} = a^{p-1-x} \bmod p$.

c. Lakukan dekripsi pada *cipherimage*, *plainimage* = $C\ (a^x)^{-1} \bmod p$.

Setelah proses tersebut, maka *plainimage* akan kembali didapatkan.

Pada kriptografi visual dengan algoritma elgamal ini, dapat menggunakan salah satu dri beberapa metode enkripsi, yaitu meotde N-*segment* RGB, yaitu melakukan enkripsi pada tiap-tiap nilai rgb pada piksel tersebut. Contoh misalkan

dipilih 1 *pixel* yang terdiri dari nilai r, nilai g, nilai b, maka akan dilakukan proses enkripsi pada tiap nilainya yaitu nilai r, nilai g, dan nilai b, sehingga proses perhitungan akan semakin sukar karena akan melakukan enkripsi pada tiap nilai-nilai piksel pada suatu gambar.

Diantara beberapa kekurangan dari algoritma Elgamal yang digunakan untuk kriptografi visual adalah sulitnya untuk menyepakati metode yang digunakan pada pemilihan piksel dalam proses melakukan enkripsi pada nilai-nilai piksel pada suatu gambar.

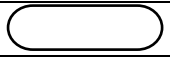
## *G. Flowchart*

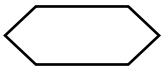
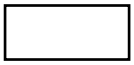
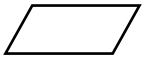
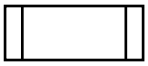
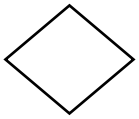
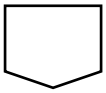
*Flowchart* adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

Dalam perancangan *flowchart* sebenarnya tidak ada rumus atau patokan yang bersifat mutlak (pasti). Hal ini didasari oleh *flowchart* (bagan alir) adalah sebuah gambaran dari hasil pemikiran dalam menganalisa suatu permasalahan dalam komputer. Karena setiap analisa akan menghasilkan hasil yang bervariasi antara satu dan lainnya. Kendati begitu secara garis besar setiap perancangan *flowchart* selalu terdiri dari tiga bagian, yaitu *input*, proses dan *output*.

Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *flowchart* :

**Tabel 2.6** *Flowchart*

| No. | Simbol | Nama | Fungsi |
|---|---|---|---|
| 1 |  | *Terminator* | Permulaan/akhir program |
| 2 |  | Garis Alir (*Flow Line*) | Arah aliran program |

| No. | Simbol | Nama | Fungsi |
|---|---|---|---|
| 3 |  | *Preparation* | Proses inisialisasi program/pemberian nilai awal |
| 4 |  | Proses | Proses perhitungan/proses pengolahan data |
| 5 |  | *Input*/*Output* Data | Proses *input*/*output* data, parameter, informasi |
| 6 |  | *Predefined process (sub program)* | Permulaan *sub* program/proses menjalankan *sub* program |
| 7 |  | *Decision* | Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk data langkah selanjutnya |
| 8 |  | *On Page Connector* | Penghubung bagian-bagian *flowchart* yang berada pada halaman berbeda |
| 9 |  | *Off Page Connector* | Penghubung bagian-bagian *flowchart* yang pada halaman berbeda |

## H. *Unified Modelling Language* (UML)

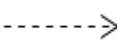
*Unified Modeling Language* (UML) adalah "sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem". *Unified Modelling Language* adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks.

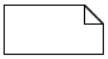
*Unified Modeling Language* merupakan metodologi untuk mengembangkan sistem OOP dan sekelompok *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah

organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.

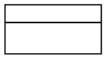
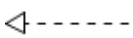
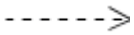
Seperti bahasa - bahasa lainnya, UML mendefinisikan notasi dan Syntax/semantik. Setiap bentuk memiliki makna tertentu, dan UML *Syntax* mendefinisikan bagaimana bentuk – bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya : *Grady Booch OOD (Object – Oriented Design), Jim Rumbaugh* OMT *(Object Modeling Technique)* dan Ivar Jacobson OOSE *(Object – Oriented Software Engineering).* Dalam UML sendiri ada beberapa garis besarnya, khususnya :

**Tabel 2.7** *Symbol Use Case Diagram*

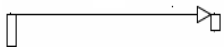
| No | GAMBAR | NAMA | KETERANGAN |
|---|---|---|---|
| 1 |  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *Use Case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*Independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*Descendent*) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (*Ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *Use Case* sumber secara Eksplisit. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |

| No | GAMBAR | NAMA | KETERANGAN |
|---|---|---|---|
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor* |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

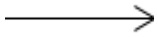
**Tabel 2.8** *Symbol Class Diagram*

| No. | GAMBAR | NAMA | KETERANGAN |
|---|---|---|---|
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor* |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

**Tabel 2.9** *Symbol Sequence Diagram*

| No | GAMBAR | NAMA | KETERANGAN |
|---|---|---|---|
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

**Tabel 2.10** *Symbol State Chart Diagram*

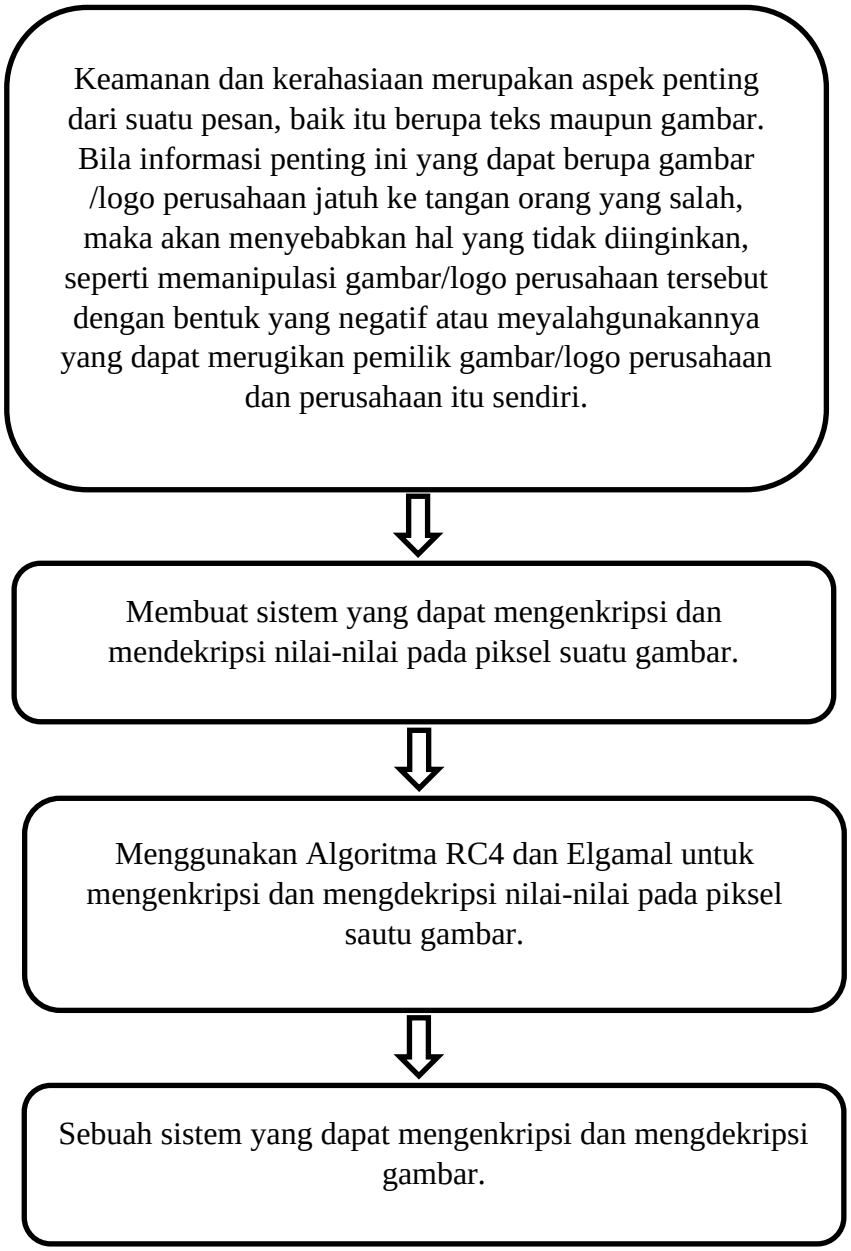
| No | GAMBAR | NAMA | KETERANGAN |
|---|---|---|---|
| 1 |  | *State* | Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |
| 2 |  | *Initial Pseudo State* | Bagaimana objek dibentuk atau diawali |
| 3 |  | *Final State* | Bagaimana objek dibentuk dan dihancurkan |
| 4 |  | *Transition* | Sebuah kejadian yang memicu sebuah *state* objek dengan cara memperbaharui satu atau lebih nilai atributnya |
| 5 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 6 |  | *Node* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

**Tabel 2.11** *Symbol Activity Diagram*

| No. | GAMBAR | NAMA | KETERANGAN |
|---|---|---|---|
| 1 |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |

| No. | GAMBAR | NAMA | KETERANGAN |
|---|---|---|---|
| 4 |  | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| 5 |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

## I. Kerangka Pikir



# BAB III

## METODE PENELITIAN

### A. Jenis Penelitian

Dalam pembuatan penelitian ini digunakan metode deskriptif yang menggambarkan fakta-fakta dan informasi secara sistematis, faktual dan akurat. Penelitian ini dilakukan melalui internet yang dapat memberikan sumber data dan pengetahuan mengenai sistem yang diteliti, kemudian mencocokkan dengan kemungkinan yang terjadi dalam usaha penyelesaian masalah.

### B. Waktu dan Tempat Penelitian

Penelitian ini dilakukan di kota Parepare, adapun waktu yang dipergunakan untuk pelaksanaan penelitian adalah ± 3 bulan.

### C. Alat dan Bahan

Untuk melakukan proses penelitian dalam pembuatan aplikasi, maka diperlukan perangkat keras dan perangkat lunak guna mendukung kegiatan penelitian tersebut. Berikut ini merupakan penjelasan dari *hardware* dan *software* yang digunakan dalam pembuatan aplikasi enkripsi gambar.

**1. Perangkat keras**

Perangkat keras yang digunakan untuk membuat aplikasi enkripsi dan dekripsi gambar dapat dilihat pada table berikut :

**Tabel 3.1** Spesifikasi Perangkat Keras

| **Spesifikasi** | |
|---|---|
| Merk laptop | Acer *Notebook* PC |
| *Processor* laptop | *AMD E1-2500 APU with Radeon (TM) HD Graphic (2 CPUs), ~1.4GHz* |
| RAM laptop | 2 (2048) GB |

**2. Perangkat lunak**

Perangkat lunak yang digunakan untuk membuat aplikasi enkripsi dan dekripsi gambar dapat dilihat sebagai berikut :

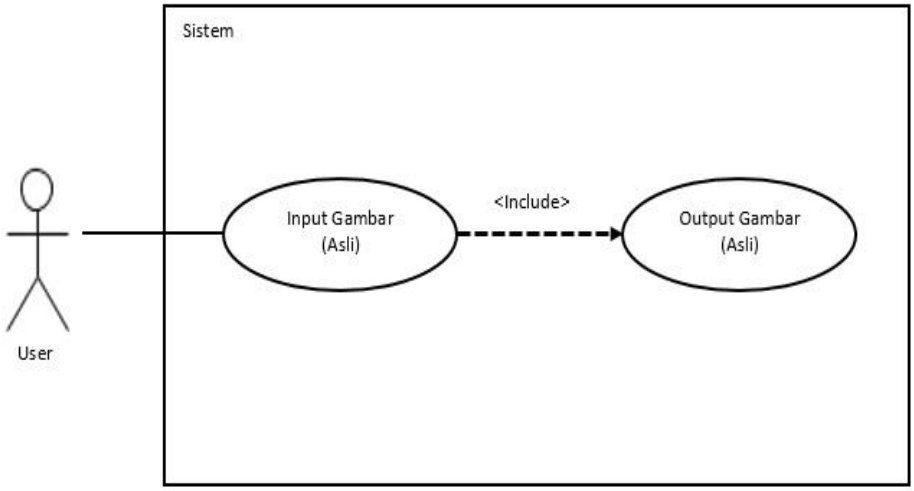
**Tabel 3.2** Spesifikasi Perangkat Lunak

| **Spesifikasi** | |
|---|---|
| Sistem operasi | Windows 7 |
| *Tool* pemrograman | Matlab |
| Bahasa pemrograman | Matlab |

## D. Perancangan Sistem

Perancangan sistem yang digunakan untuk mengenkripsi dan mendekripsi gambar ini meliputi sistem yang berjalan dan yang diusulkan, serta perancangan sistem enkripsi dan dekripsi dengan algoritma RC4 dan Elgamal, serta racangan enkripsi dan dekripsi gabungan antara algoritma RC4 dan Elgamal.

## 1. Sistem yang berjalan

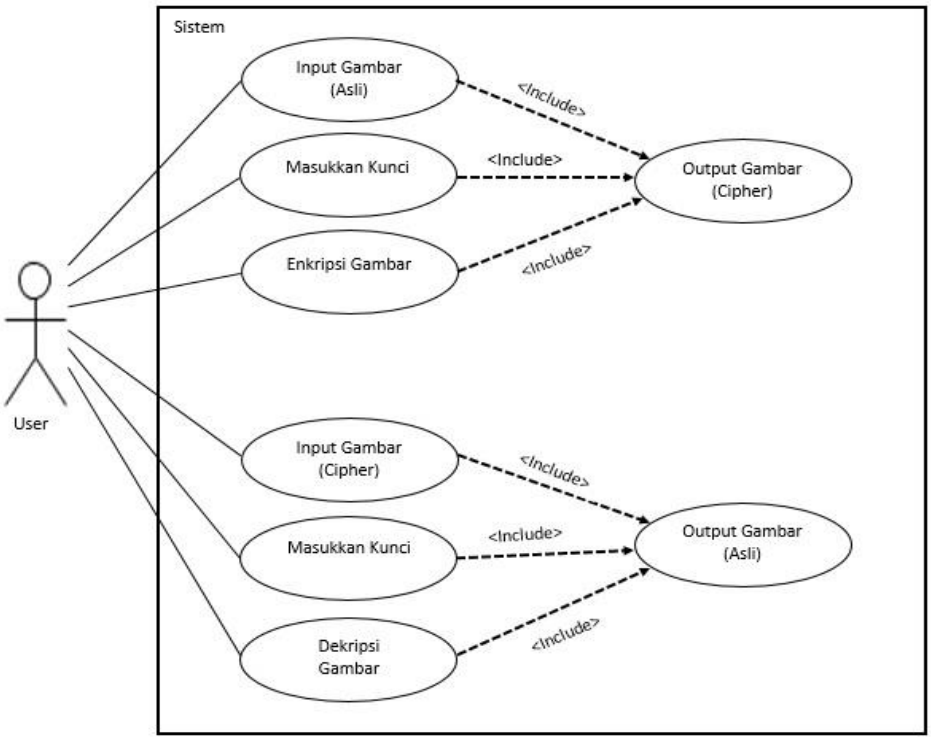



**Gambar 3.1** *Use Case Diagram* Sistem Yang Berjalan

**Tabel 3.3** Keterangan *Use Case Diagram* Sistem Yang Berjalan

| NO. | Nama *Use Case* | Keterangan *Use Case* |
|---|---|---|
| 1 | *User* | Aktor |
| 2 | *Input* Gambar (Asli) | Proses pertama ketika *user* memasukkan gambar asli |
| 3 | *Output* Gambar (Asli) | Proses ketika *user* selesai melakukan proses dan menghasilkan gambar yang tetap pada bentuk asli |

**2. Sistem yang diusulkan**

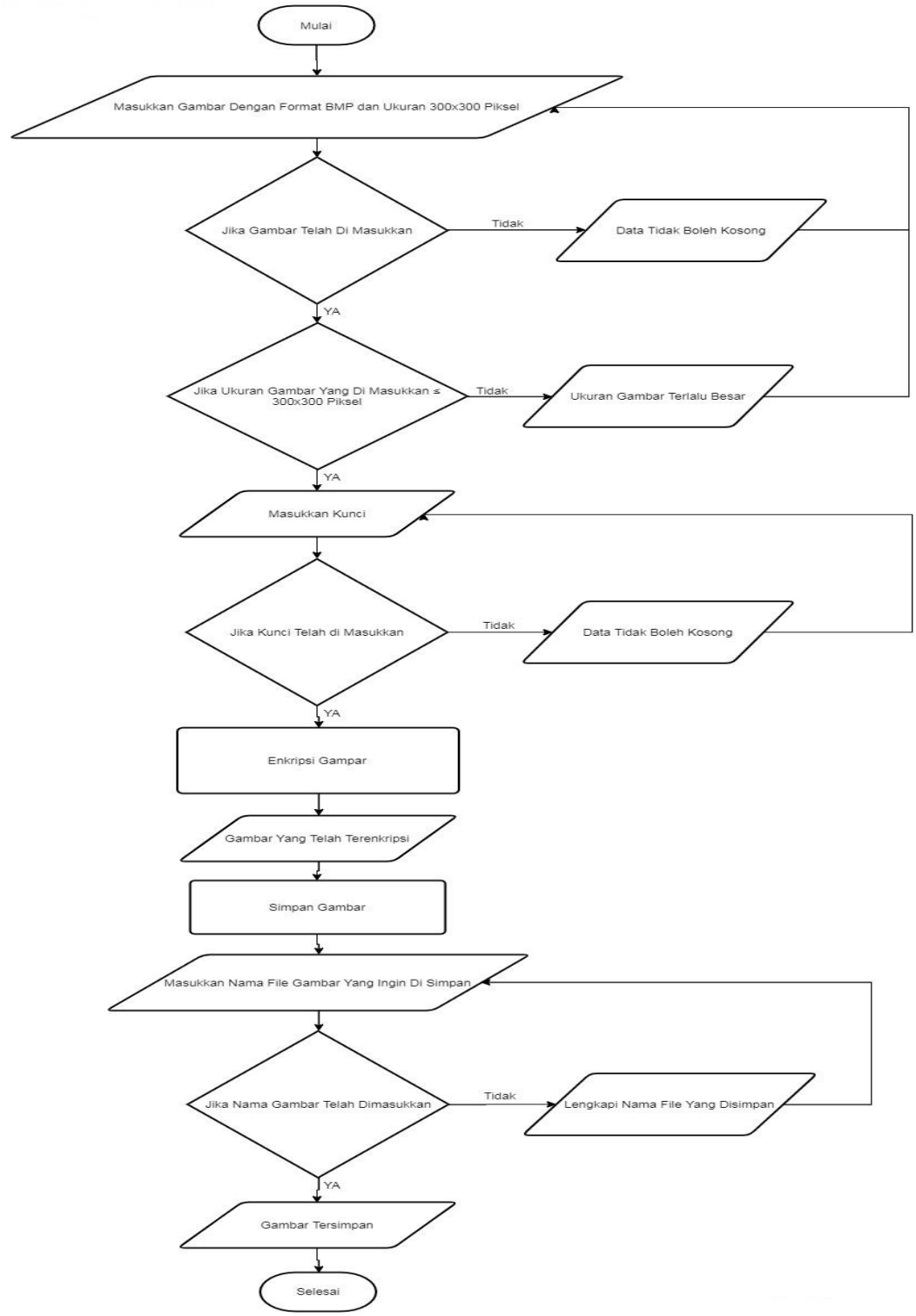



**Gambar 3.2** *Use Case Diagram* Sistem Yang Diusulkan

**Tabel 3.4** Keterangan *Use Case Diagram* Sistem Yang Diusulkan

| No. | Nama *Use Case* | Keterangan *Use Case* |
|---|---|---|
| 1 | *User* | Aktor |
| 2 | *Input* Gambar (Asli) | Proses pertama ketika *user* memasukkan gambar asli |
| 3 | Masukkan Kunci | Proses ketika *user* memasukkan kunci sebelum melakukan enkripsi gambar |
| 4 | Enkripsi Gambar | Proses berikutnya setelah *user* memasukkan kunci yaitu enkripsi gambar |
| 5 | *Output* Gambar (*Cipher*) | Merupakan hasil dari proses enkripsi gambar yang berupa *cipher* |
| 6 | Dekripsi Gambar | Proses berikutnya setelah *user* memasukkan kunci yaitu dekripsi gambar |
| 7 | *Output* Gambar (Asli) | Merupakan hasil dari proses dekripsi gambar yang telah berupa bentuk gambar asli |

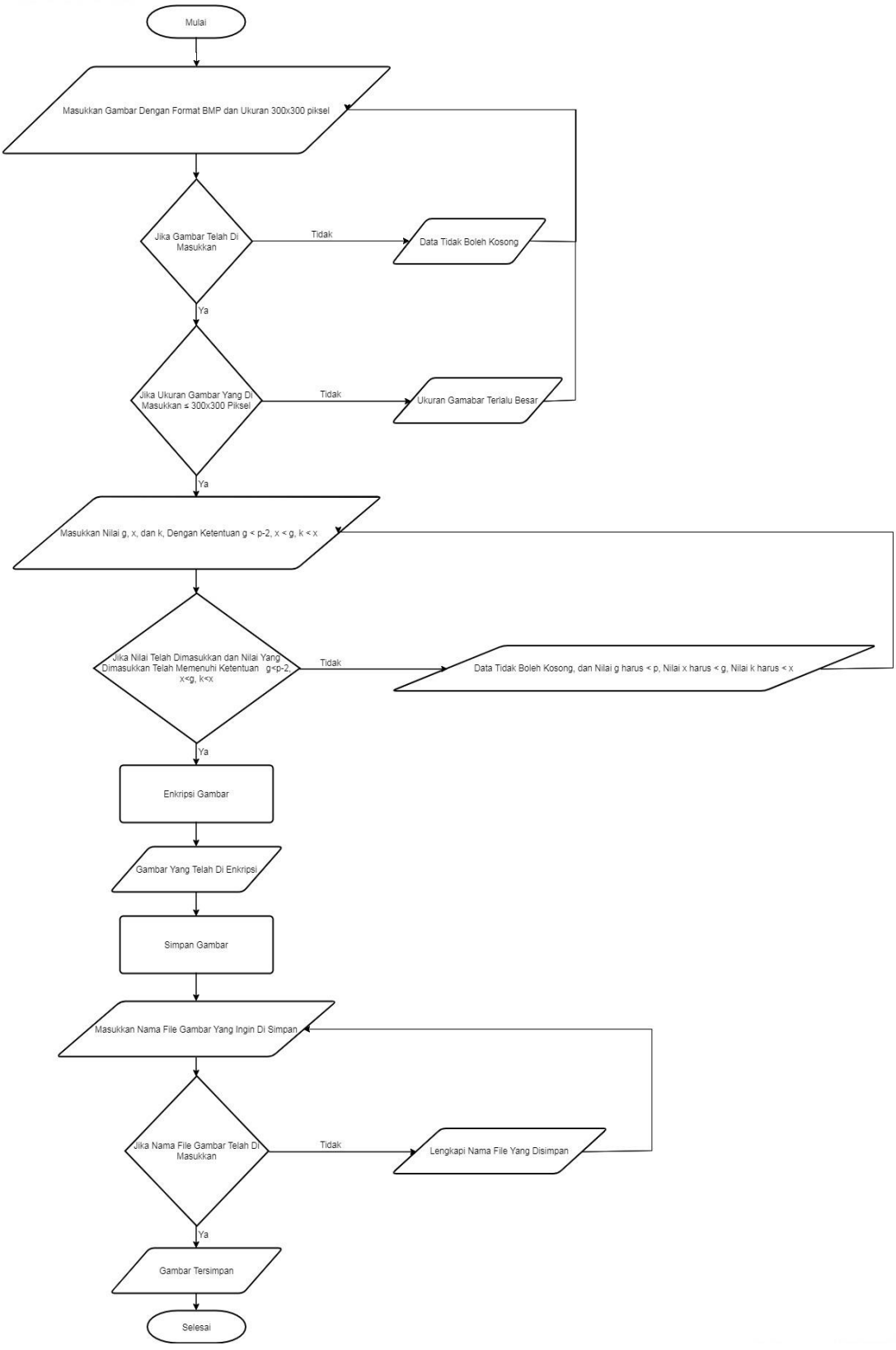
## 3. Flowchart



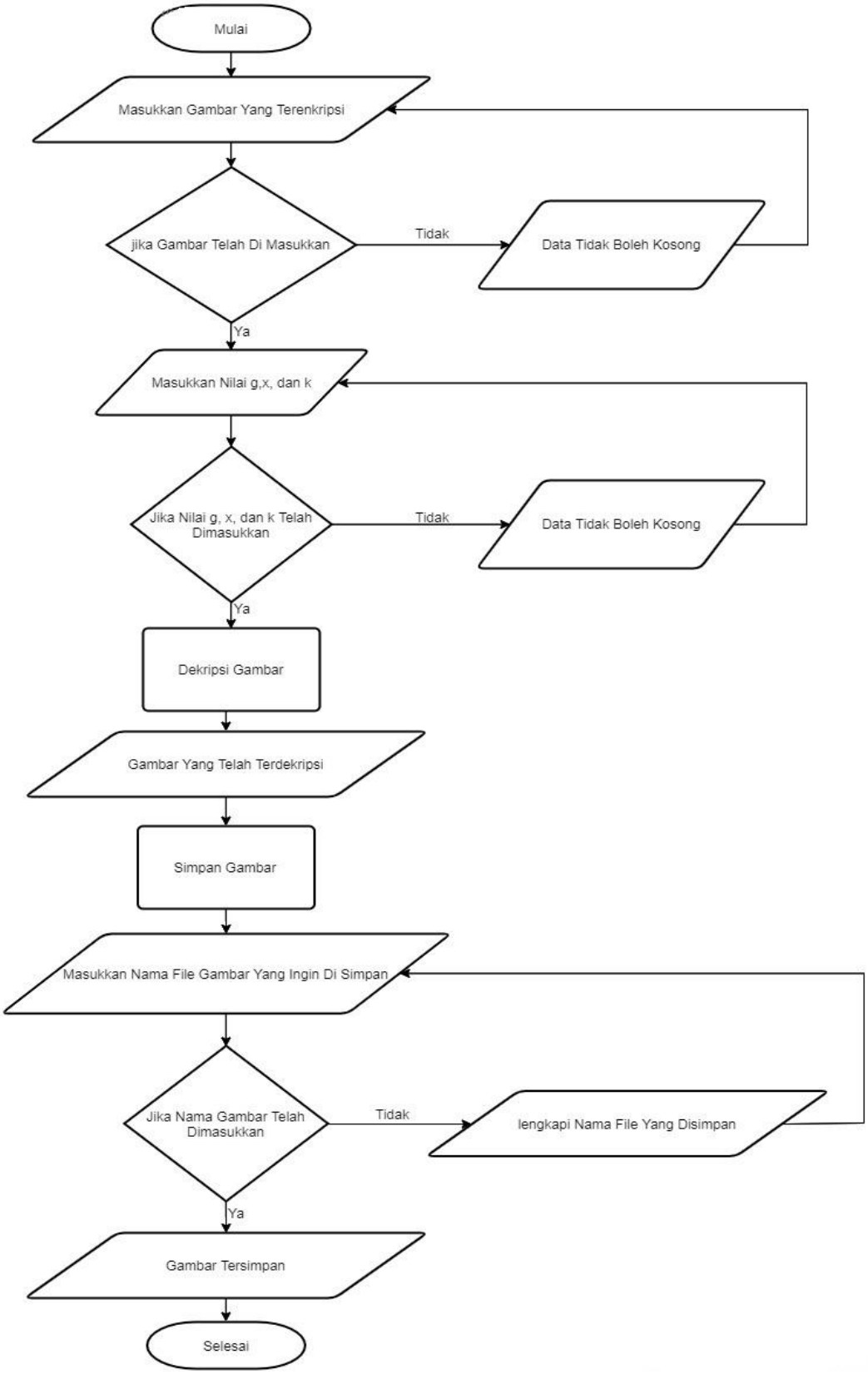

**Gambar 3.3** *Flowchart* Enkripsi Algoritma RC4



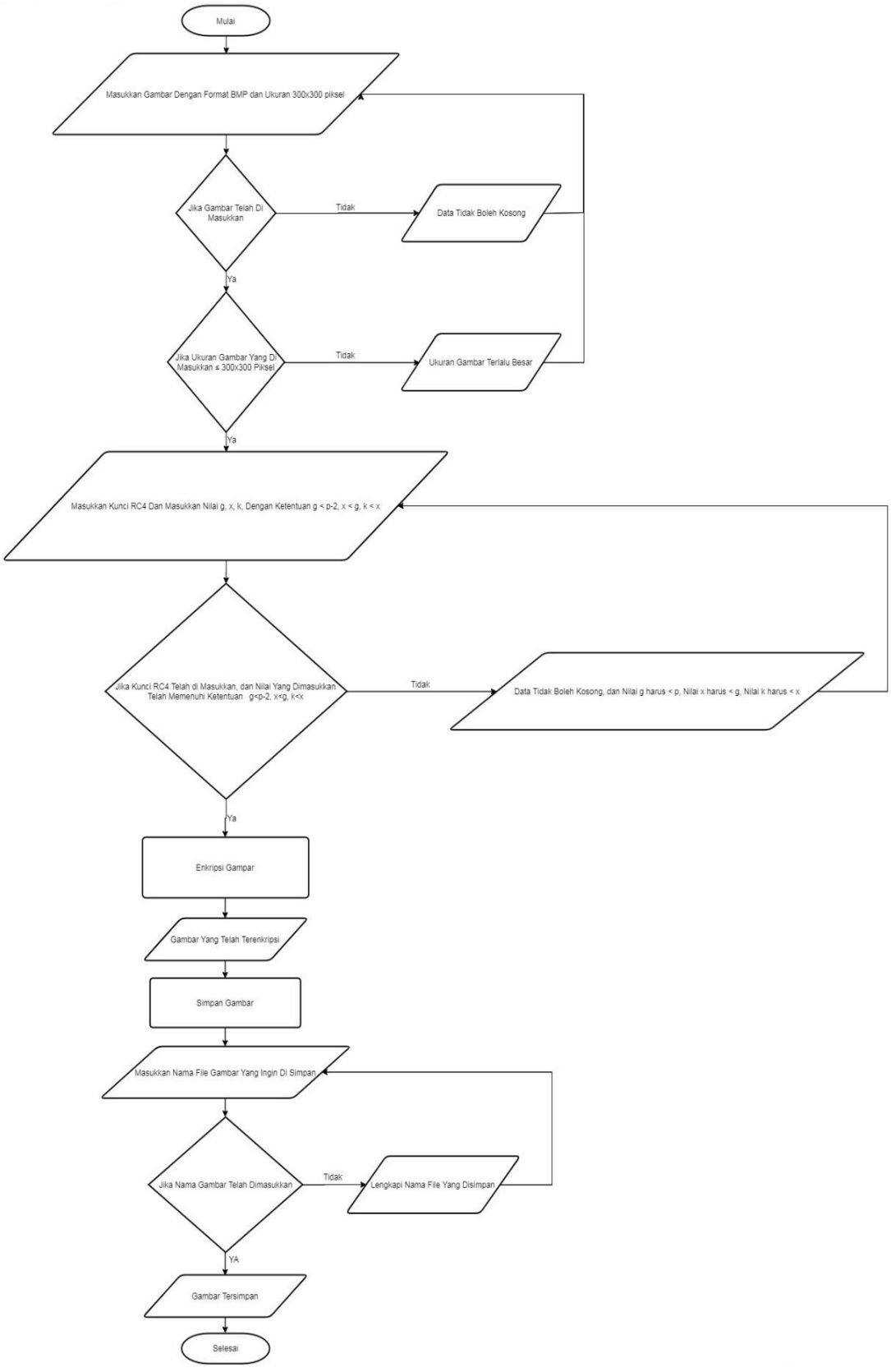

**Gambar 3.4** *Flowchart* Dekripsi Algoritma RC4



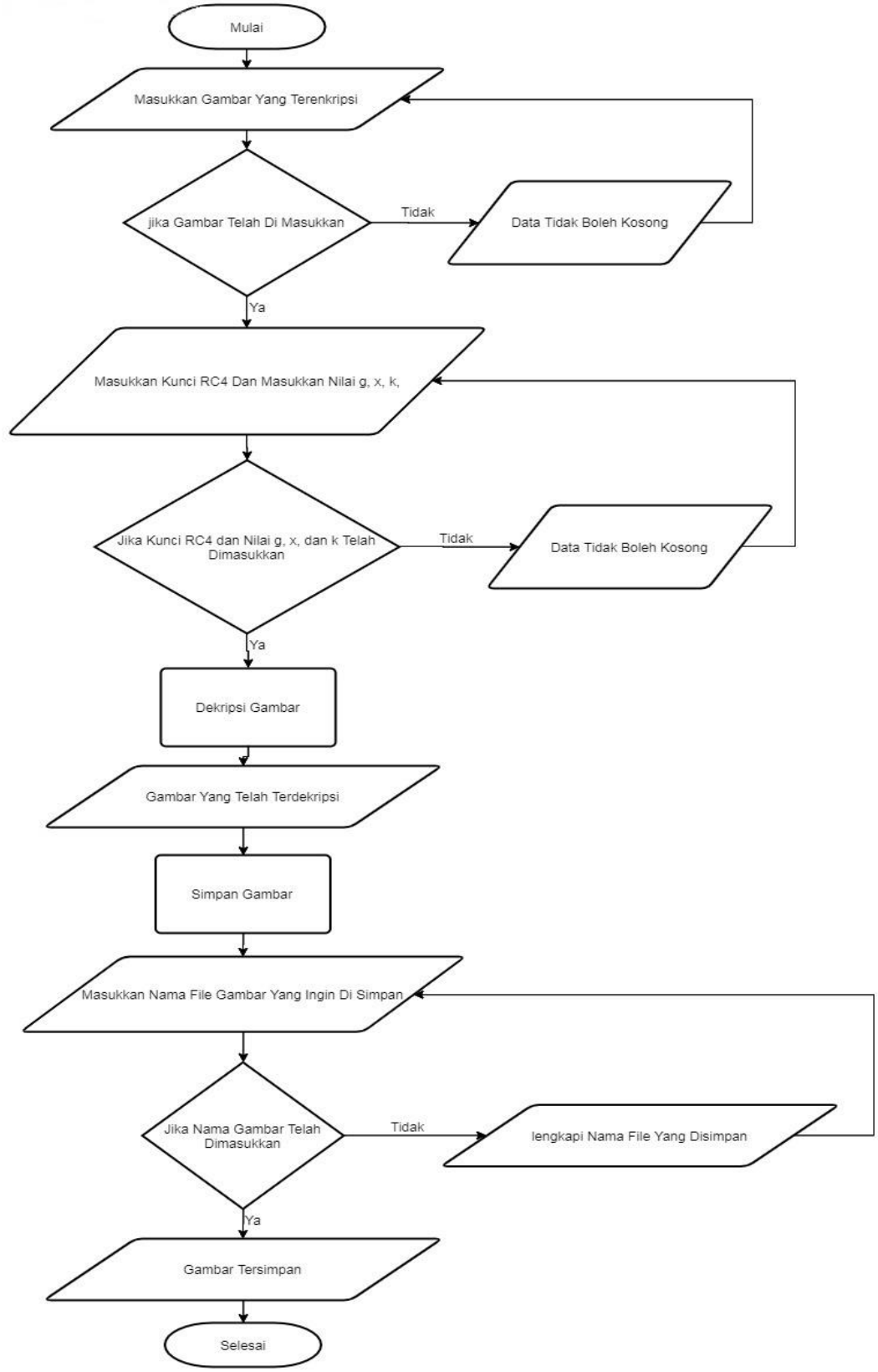

**Gambar 3.5** *Flowchart* Enkripsi Algoritma Elgamal




**Gambar 3.6** *Flowchart* Dekripsi Algoritma Elgamal




**Gambar 3.7** *Flowchart* Enkripsi Gabungan Algoritma RC4 Dan Elgamal




**Gambar 3.8** *Flowchart* Dekripsi Gabungan Algoritma RC4 Dan Elgamal

## E. Metode Pengumpulan Data

Metode pengumpulan data yang tepat yaitu dengan mempertimbangkan penggunaannya berdasarkan jenis data dan sumbernya. Data yang *objektif* dan relevan dengan pokok permasalahan penelitian merupaka indikator keberhasilan suatu penelitian. Pengumpulan data pada penelitian ini dilakukan dengan cara melakukan studi pustaka yaitu mengumpulkan data dengan mempelajari masalah yang berhubungan dengan objek yang diteliti, bersumber dari buku-buku pedoman, literatur yang disusun oleh para ahli untuk melengkapi data yang diperlukan dalam penelitian baik secara *offline* maupun *online*. Adapun jurnal dan beberapa buku yang digunakan dalam mengumpulkan data-data yang dibutuhkan yaitu buku – buku yang berkaitan dengan enkripsi dan dekripsi gambar dengan menggunakan algoritma RC4 dan Elgamal.

## F. Metode Pengujian

Setelah tahap perancangan aplikasi telah dibuat serta beberapa kali uji coba pada aplikasi, selanjutnya dilakukan pengujian, yang mana pada tahap pengujian dilakukan dalam 3 bagian yaitu, pengujian enkripsi denkripsi gambar pada algoritma yang pertama yaitu RC4, kemudian melakukan pengujian enkripsi dan dekripsi pada algoritma Elgamal, kemudian setelah itu tahap yang ketiga melakukan percobaan enkripsi dekripsi dengan menggabungkan dua algoritma yaitu RC4 dan Elgamal. Selain pengujian di atas dilakukan pula pengujian *black box* dan juga pengujian *white box*.

**1. Metode *Black Box***

Yaitu memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

**2. Metode *White Box***

Yaitu pengujian terhadap cara kerja perangkat lunak itu sendiri yaitu prosedur programnya (basis *path*) atau proses *looping* (pengulangan).

# BAB IV

# HASIL DAN PEMBAHASAN

## A. Hasil

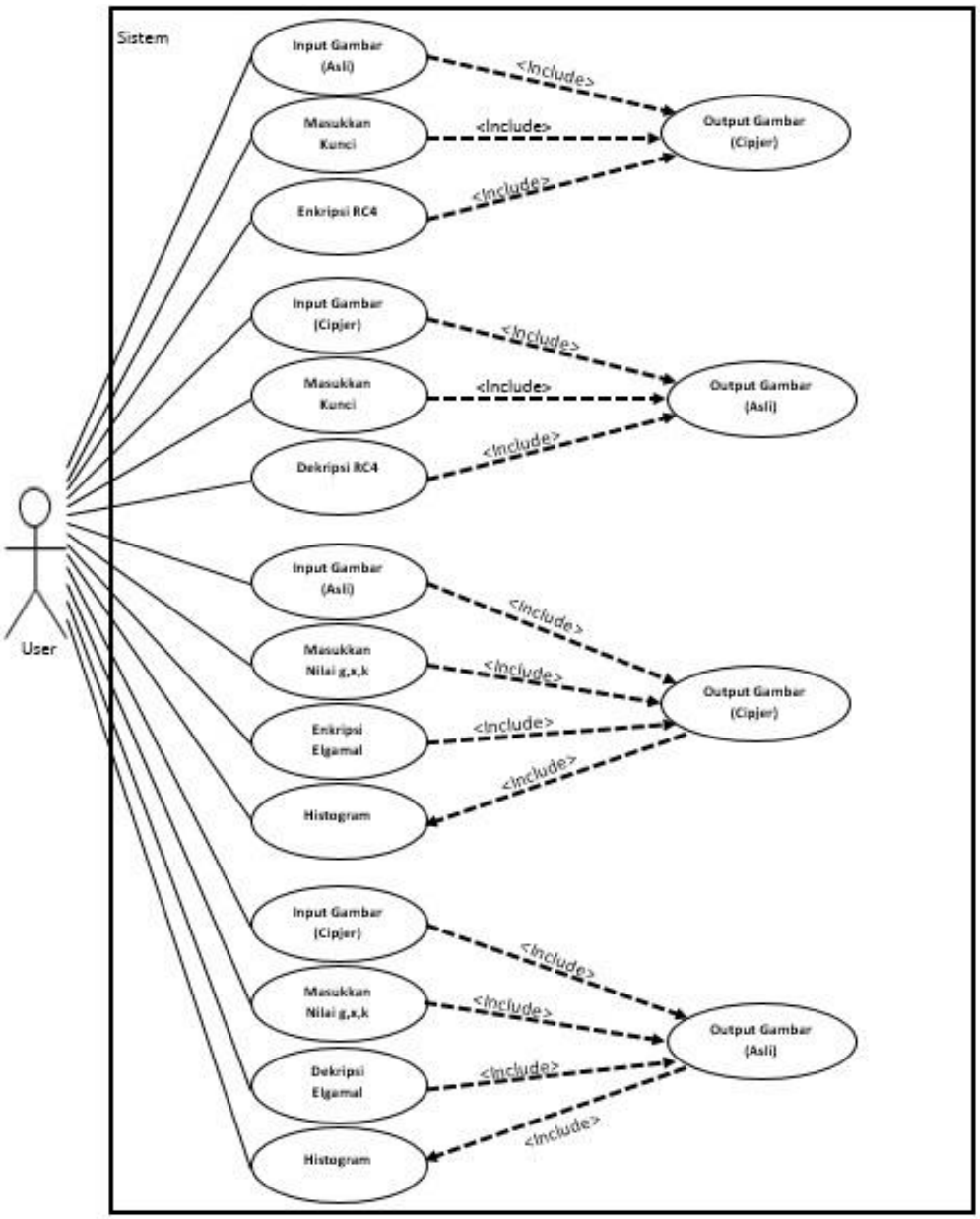
Pada bagian ini akan dipaparkan mengenai perancangan sistem dengan menggunakan model UML (*Unified Modelling Language*) dan perancangan aplikasi enkripsi dan dekripsi gambar.

### 1. Perancangan model UML (*Unified Modelling Language*)

Sistem yang penulis rancang adalah sistem berupa enkripsi dan dekripsi gambar. Adapun dalam melakukan perancangan sistem adalah dengan memanfaatkan diagram UML berupa *Use case diagram, Activity diagram,* dan *Seqeuence diagram*.

#### a. *Use case diagram*

*Use case diagram* berfungsi untuk menjelaskan alur sistem jika dilihat menurut pandangan orang yang berada diluar sistem.



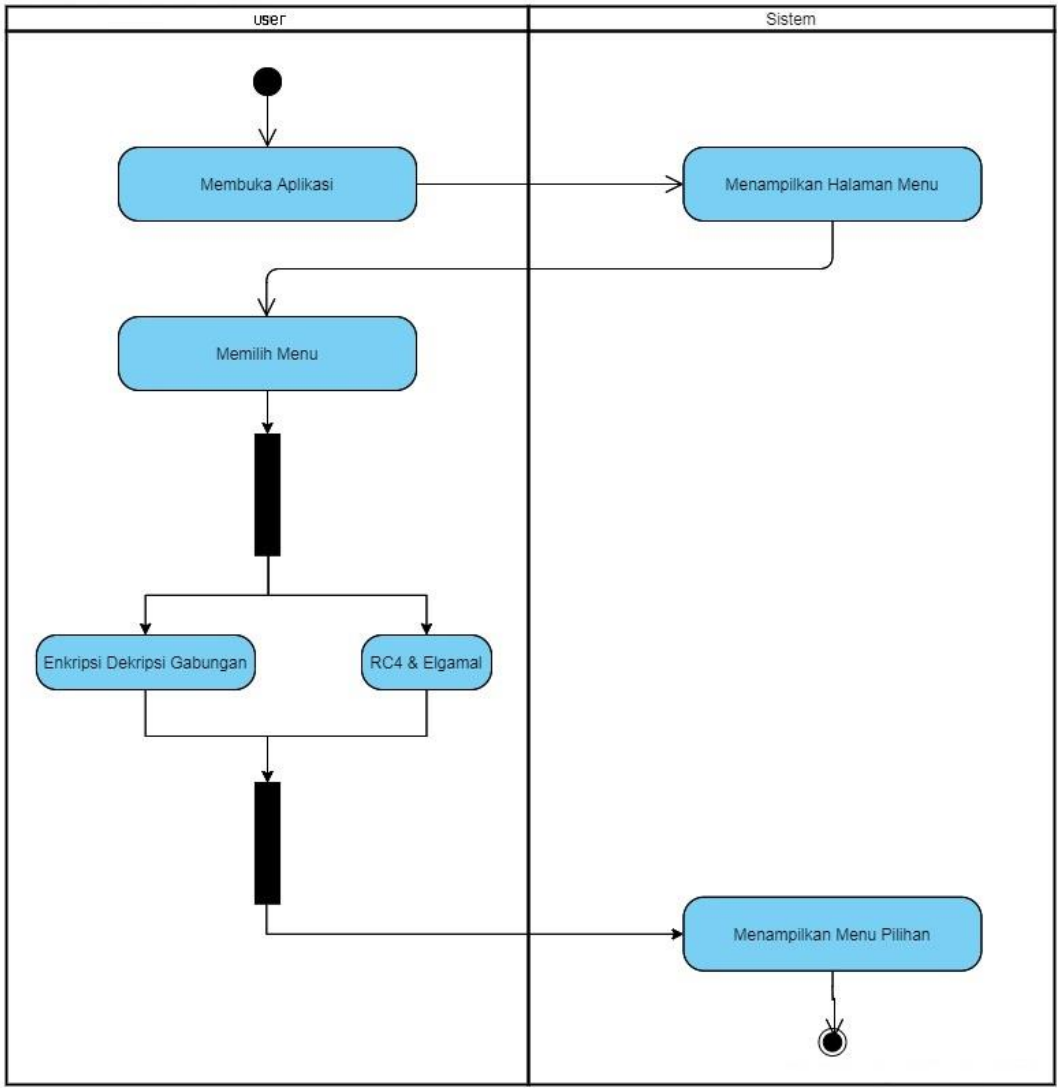

**Gambar 4.1** *Use Case Diagram* Umum

**Tabel 4.1** Penjelasan *Use Case Diagram* Umum

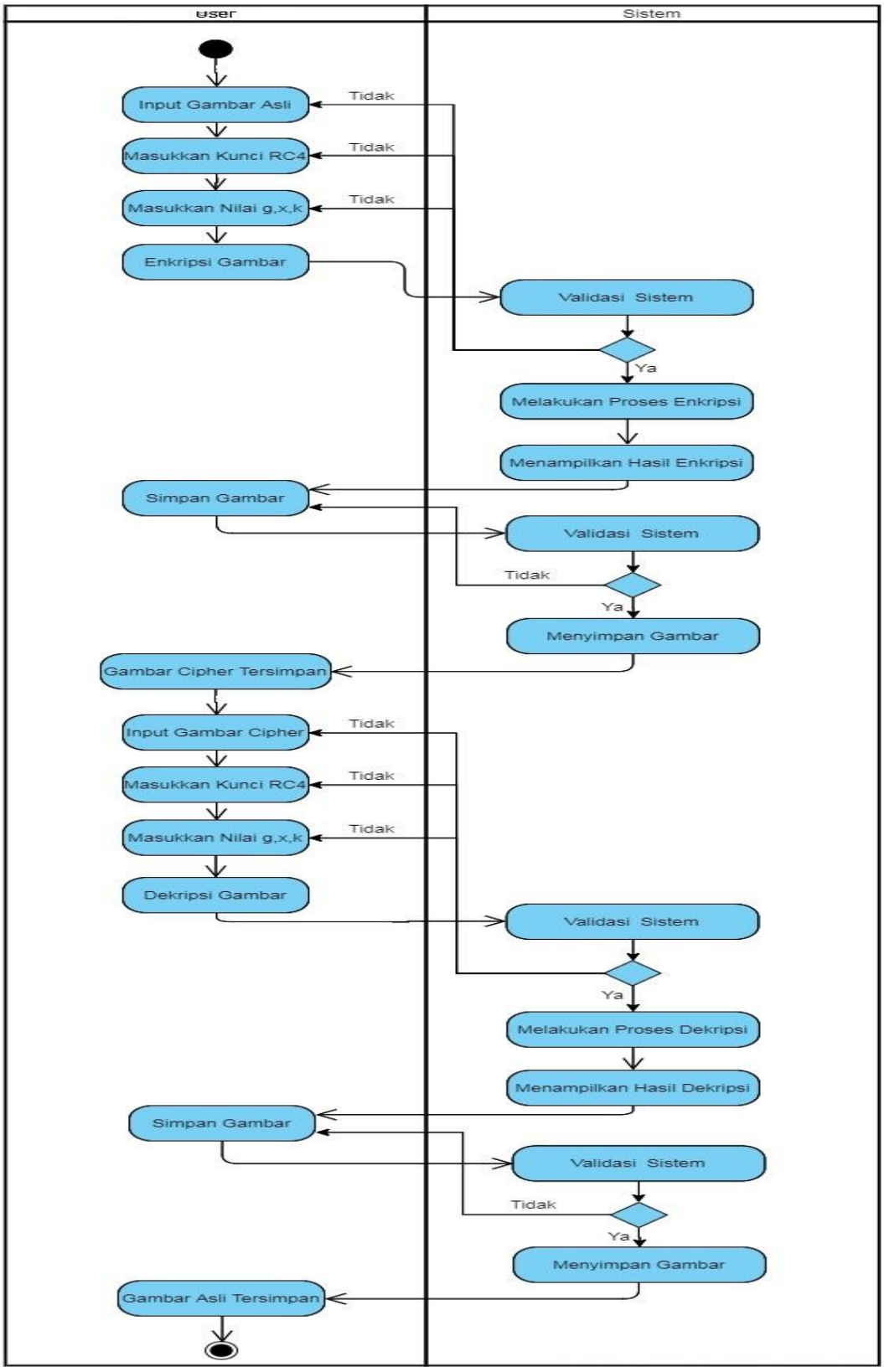
| **No.** | **Nama *Use Case*** | **Keterangan *Use Case*** |
|---|---|---|
| 1 | *User* | Aktor |
| 2 | *Input* Gambar (Asli ) | Proses pertama ketika *user* memasukkan gambar sebelum di proses |
| 3 | Masukkan Kunci | Proses *user* memasukkan kunci untuk algoritma RC4 |
| 4 | Enkripsi RC4 | Proses enkripsi gambar menggunakan RC4 |
| 5 | Input Gambar (*Cipher*) | Proses ketika *user* memasukkan gambar yang telah di proses atau terenkripsi |
| 6 | *Output* Gambar (*Cipher*) | Proses ketika gambar telah selesai di proses atau terenkripsi |
| 7 | Dekripsi RC4 | Proses dekripsi menggunakan RC4 |
| 8 | *Output* Gambar (Asli) | Proses ketika gambar telah didekripsi atau dikembalikan ke bentuk aslinya |
| 9 | Enkripsi Elgamal | Proses enkripsi menggunakan Elgamal |
| 10 | Masukkan Nilai g,x,k | Proses user memasukkan nilai sebelum proses enkripsi Elgamal |
| 11 | Dekripsi Egamal | Proses dekripsi menggunakan Elgamal |
| 12 | Histogram | Proses untuk menampilkan histogram yang berisi nilai-nilai pada piksel gambar |

**b. *Aktivity diagram***

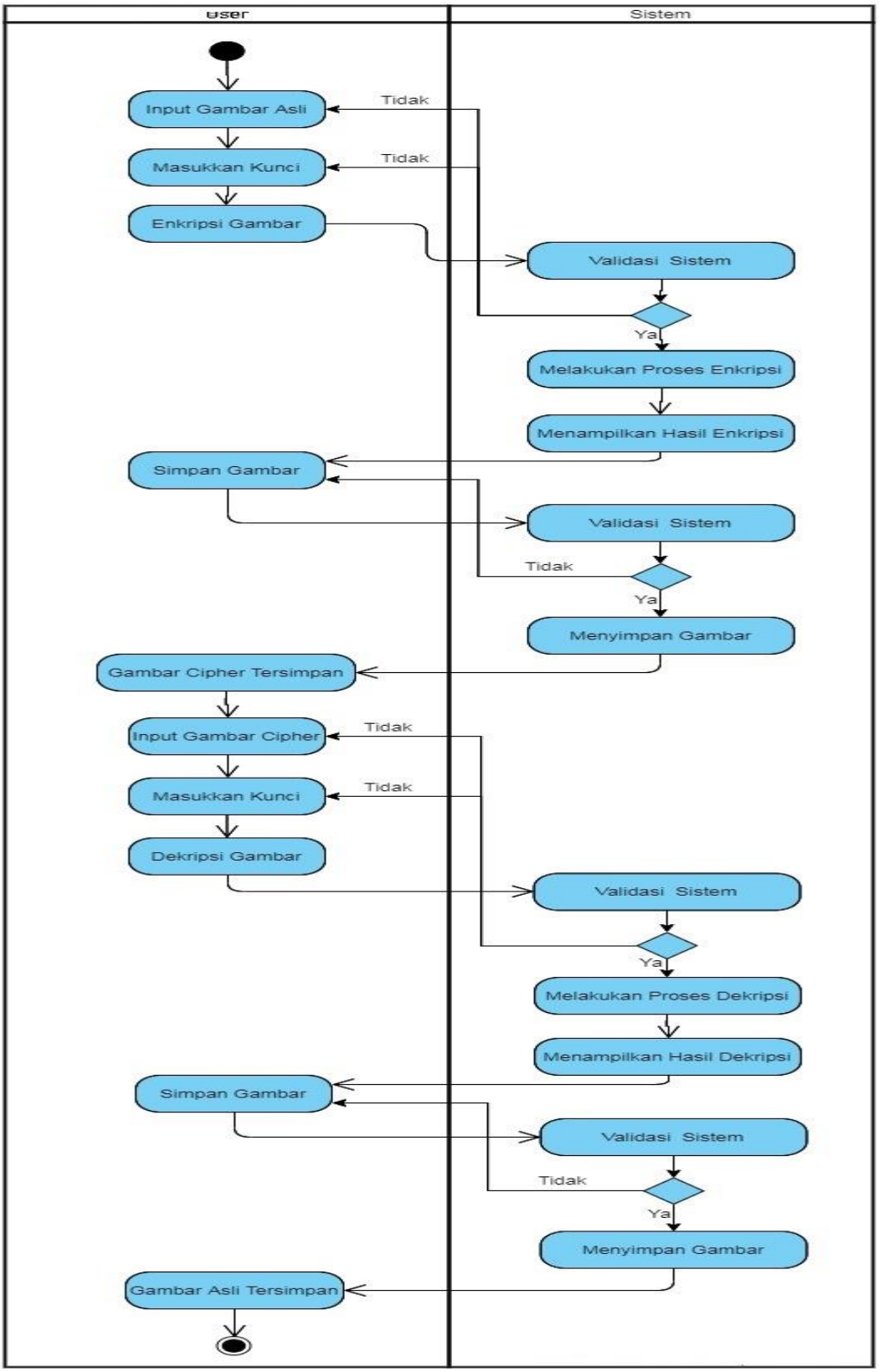
*Activity diagram* memodelkan alur kerja (*workflow*) sebuah proses dan urutan dalam suatu proses. *Activity diagram* pada sistem yang penulis rancang antara lain :



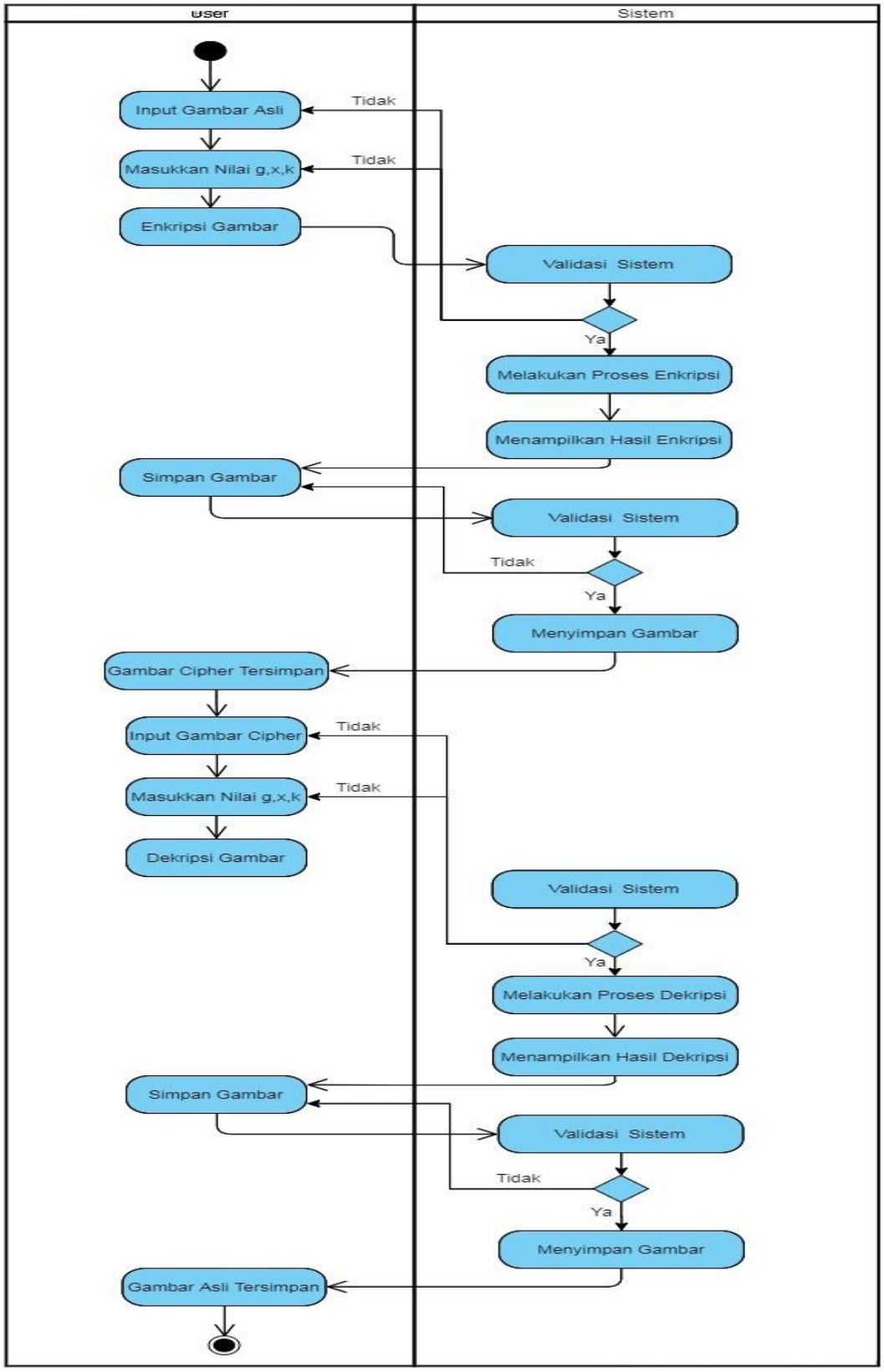

**Gambar 4.2** *Activity Diagram* Menu




**Gambar 4.3** *Activity Diagram* Gabungan RC4 dan Elgamal



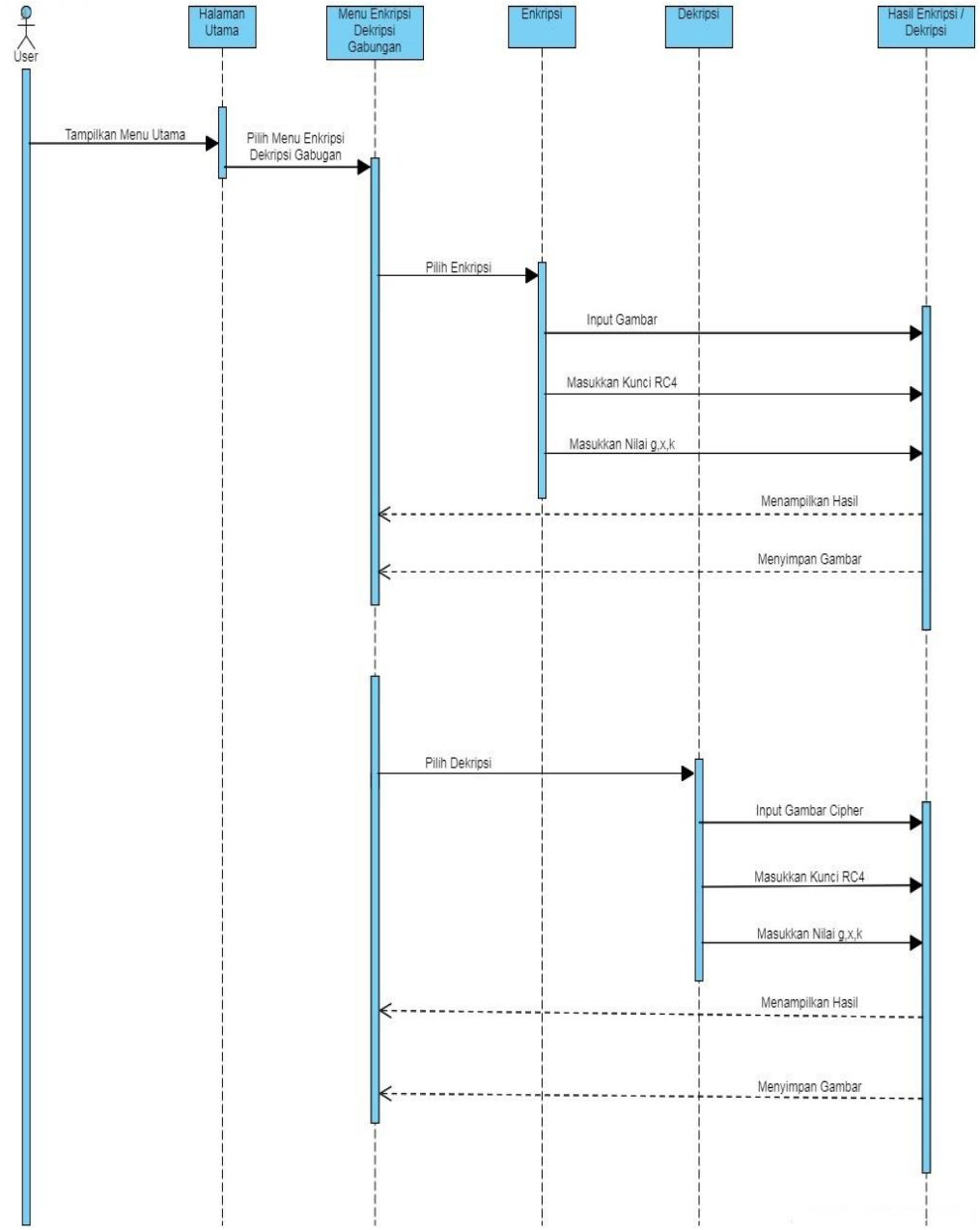

**Gambar 4.4** *Activity Diagram* Algoritma *RC4*



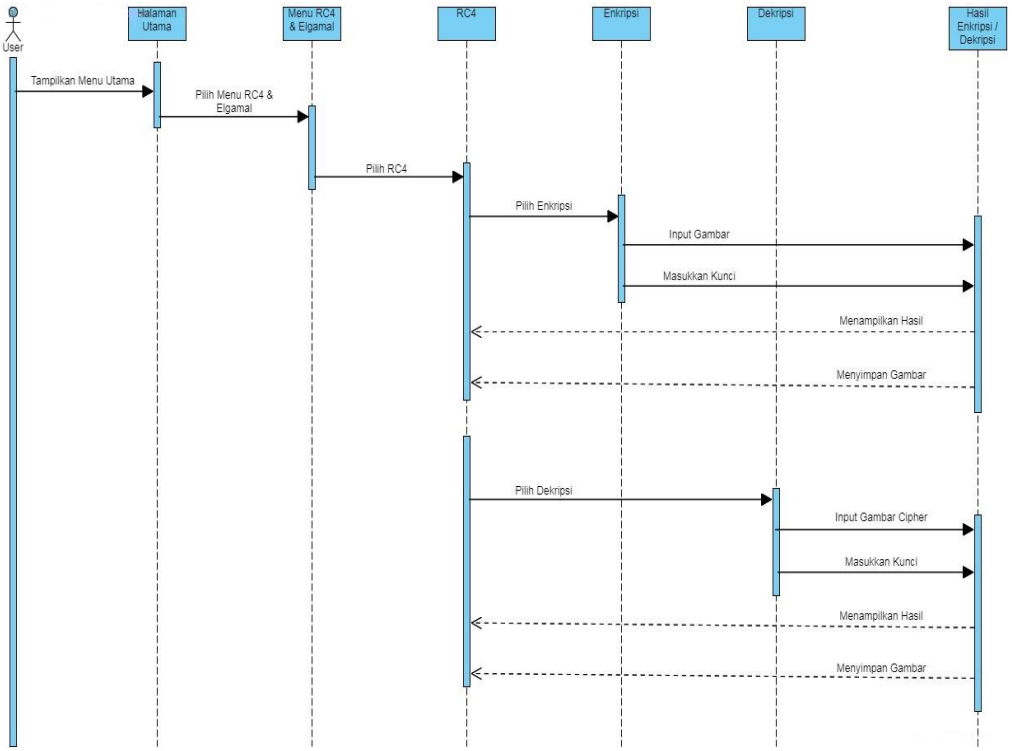

**Gambar 4.5** *Activity Diagram* Algoritma Elgamal

### c. *Sequnce diagram*

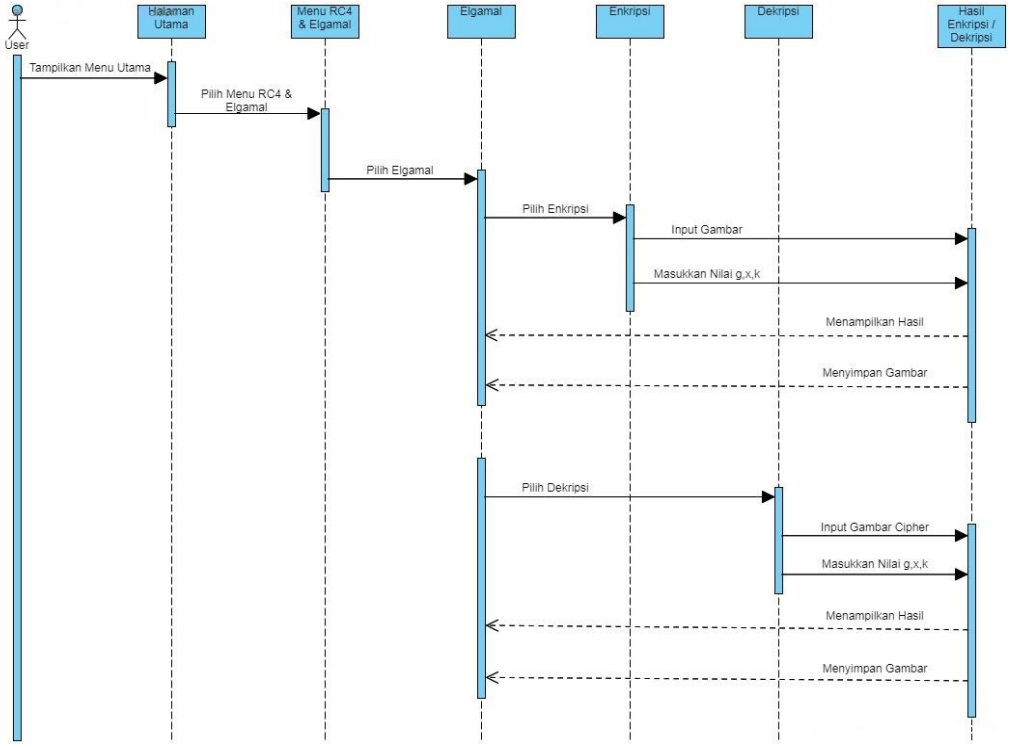
*Sequence diagram* adalah suatu diagram yang menggambarkan interaksi antara objek dan mengkondisikan komunikasi antara objek-objek tersebut. Berikut *squence diagram* aplikasi yang akan dibangun.




**Gambar 4.6** *Sequence Diagram* Gabungan RC4 dan Elgamal



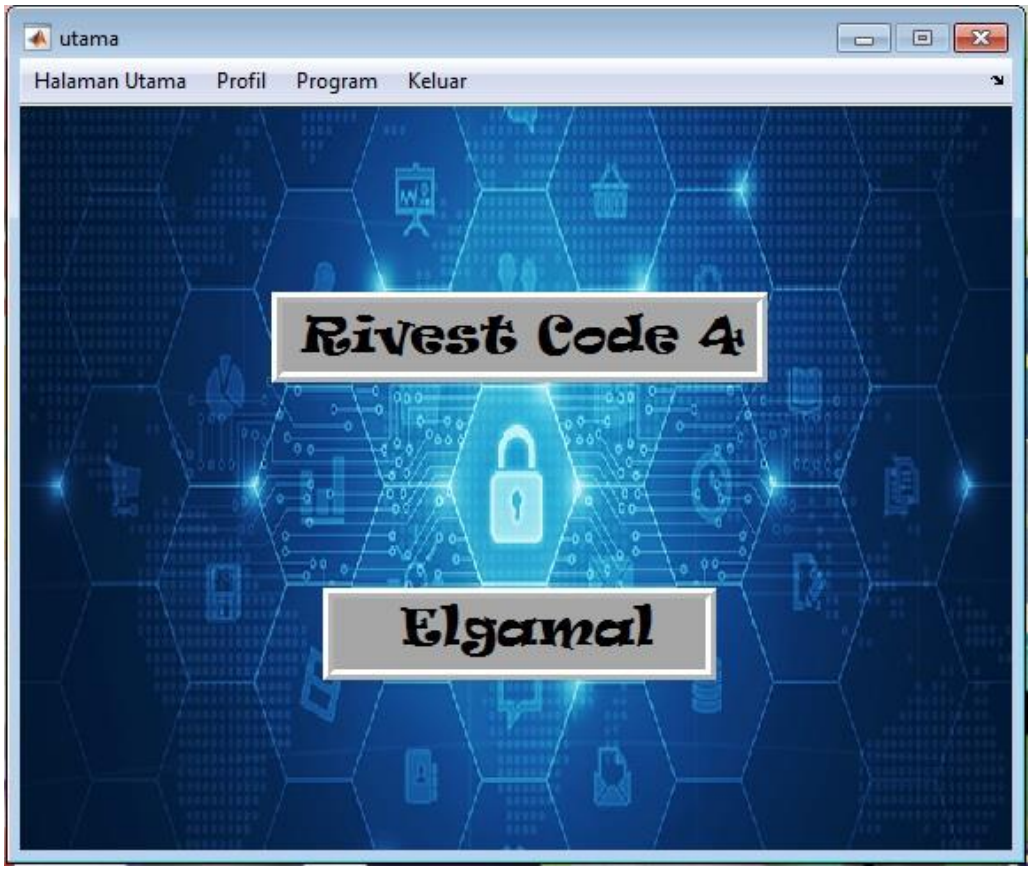

**Gambar 4.7** *Sequence Diagram* RC4




**Gambar 4.8** *Sequence Diagram* Elgamal

## 2. Perancangan aplikasi

### a. Halaman awal




**Gambar 4.9** Halaman Awal

Merupakan halaman awal dari aplikasi ketika pertama kali dibuka yang menampilkan nama algoritma yang digunakan, serta beberapa pilihan menu yaitu profil, program, dan keluar.

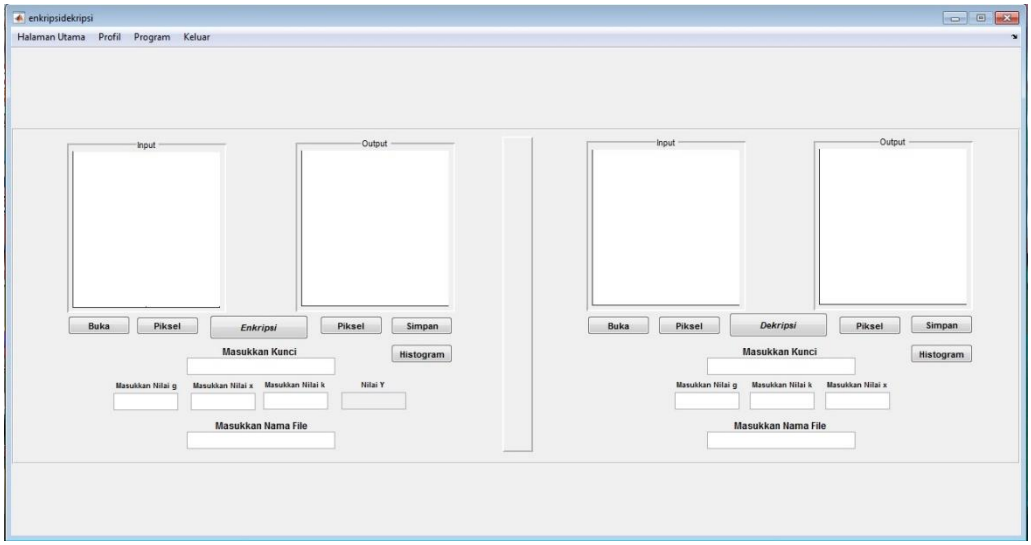
**b. Profil**




**Gambar 4.10** Profil

Profil merupakan tampilan yang berisikan nama, nim, jurusan, dan nama Universitas dari pembuat atau perancang aplikasi.

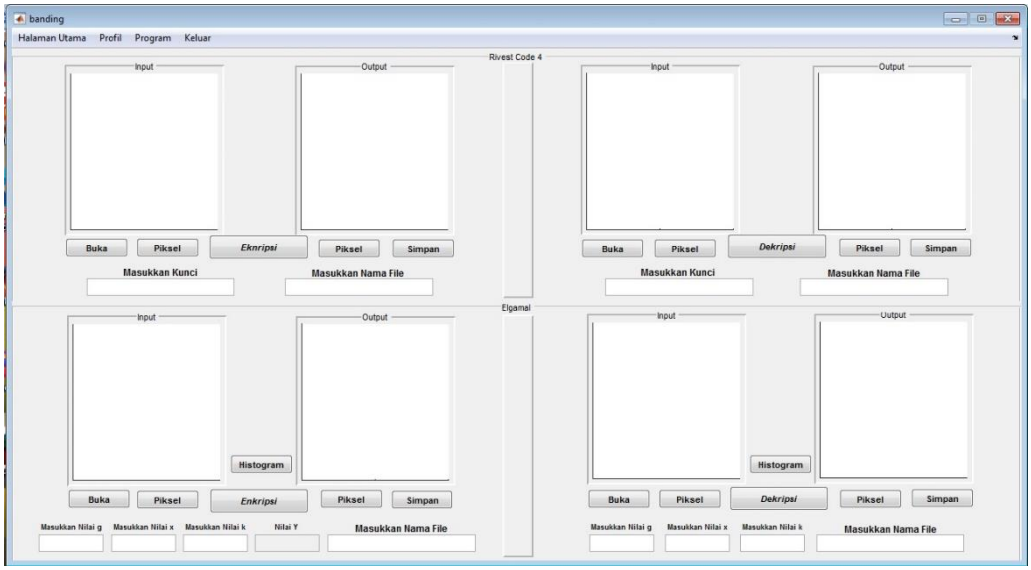
### c. Gabungan algoritma RC4 dan Elgamal




**Gambar 4.11** Gabungan RC4 dan Elgamal

Gabungan algoritma RC4 dan Elgamal merupakan tampilan yang berisi halaman yang digunakan untuk melakukan proses enkripsi dan dekripsi gambar yang menggabungkan dua algoritma yaitu RC4 dan Elgamal dalam satu proses.

### d. Algoritma RC4 dan Elgamal




**Gambar 4.12** Algoritma RC4 dan Elgamal

Pada tampilan ini terbagi menajdi dua bagian, yaitu pada bagian atas halaman merupakan enkripsi dan dekripsi gambar yang menggunakan algoritma RC4, sedangkan pada bagian bawah halaman merupakan enkripsi dan dekripsi gambar yang menggunakan algoritma Elgamal.

## B. Pembahasan

Pada bagian ini akan dibahas mengenai implementasi yang berupa contoh perhitungan manual, kemudian akan dibahas juga mengenai pengujian sistem yaitu *black box* dan *white box*.

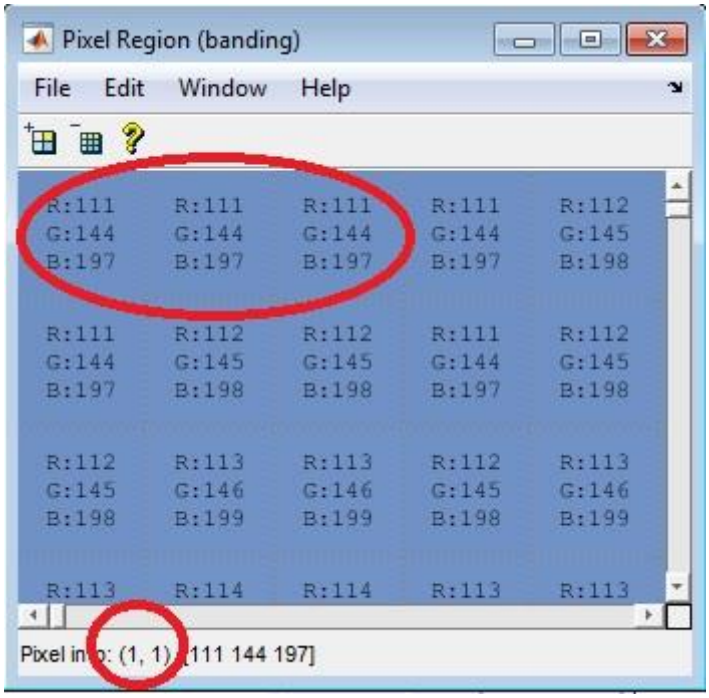
### 1. Implementasi

Implementasi adalah suatu penerapan atau tindakan yang dilakukan berdasarkan rencana yang telah disusun atau dibuat dengan cermat dan terperinci sebelumnya. Adapun pembahasan pada bagian ini akan membahas mengenai pengimplementasian algoritma RC4 dan Elgamal pada proses enrkipsi dan dekripsi pada citra digital yang berupa gambar 2 dimensi, dan akan diambil 3 piksel yang terdiri dari *red*, *green*, dan *blue* sebagai sampel untuk perhitungan manual pada algoritma RC4 dan Elgamal.

Berikut gambar dan 3 piksel yang dijadikan sebagai sampel :



**Gambar 4.13** Gambar Sampel




**Gambar 4.14** Nilai Piksel Pada Gambar Sampel

**a. Algoritma RC4**

Proses enkripsi pada algortima RC4 terdapat 3 tahap yaitu KSA, PRGA dan proses peng-*XOR*-an.

Proses pertama, *key schedulling algorithm* (KSA) :

1) Inisialisasi larik S dan K

Agar iterasi atau perulangan tidak terlalu banyak, maka akan digunakan S = 4.

Larik S = $S_0 = 0$, $S_1 = 1$, $S_2 = 2$, $S_3 = 3$

*Key* = $K_0 = 2$, $K_1 = 5$, $K_2 = 7$, $K_3 = 3$

2) Permutasi Larik S

**Iterasi 1**

i = 0

j = 0

j = ( j + S[i] + K[mod (panjang K)] ) mod 4

*Swap* S[i] dengan S[j]

j = ( j + S[0] + K[0 mod 4] ) mod 4

j = ( 0 + 0 + K[0] ) mod 4

j = ( 0 + 0 + 2 ) mod 4

j = 2

*Swap* S[0] dengan S[2]

Larik S: 0 1 2 3

Hasil Larik S : 2 1 0 3

**Iterasi 2**

i = 1

j = 2

j = ( j + S[i] + K[mod (panjang K)] ) mod 4

*Swap* S[i] dengan S[j]

j = ( j + S[1] + K[1 mod 4] ) mod 4

j = ( 2 + 1 + K[1] ) mod 4

j = ( 2 + 1 + 5 ) mod 4

j = 0

*Swap* S[1] dengan S[0]

Larik S: 2 1 0 3

Hasil Larik S : 1 2 0 3

**Iterasi 3**

i = 2

j = 0

j = ( j + S[i] + K[mod (panjang K)] ) mod 4

*Swap* S[i] dengan S[j]

j = ( j + S[2] + K[2 mod 4] ) mod 4

j = ( 0 + 0 + K[2] ) mod 4

j = ( 0 + 0 + 7 ) mod 4

j = 3

*Swap* S[2] dengan S[3]

Larik S: 1 2 0 3

Hasil Larik S : 1 2 3 0

**Iterasi 4**

i = 3

j = 3

j = ( j + S[i] + K[mod (panjang K)] ) mod 4

*Swap* S[i] dengan S[j]

j = ( j + S[3] + K[3 mod 4] ) mod 4

j = ( 3 + 0 + K[3] ) mod 4

j = ( 3 + 0 + 3 ) mod 4

j = 2

*Swap* S[3] dengan S[2]

Larik S: 1 2 3 0

Hasil Larik S : 1 2 0 3

Proses kedua, *pseudo-random generation algorithm* (PRGA) :

Pada proses PRGA terdapat 2 tahap, yaitu permutasi dan pembangkitan kunci atau *keystream.* Metode yang digunakan dalam proses pembangkitan kunci atau *keystream* adalah sesuai dengan panjangnya baris pada piksel suatu gambar, oleh karena itu iterasi akan berulang sampai 3 kali karena jumlah piksel yang dijadikan sampel sebanyak 3.

Hasil akhir dari permutasi larik S=4, yaitu $S_0 = 1$, $S_1 = 2$, $S_2 = 0$, $S_3 = 3$.

**Iterasi 1**

i = 0

j = 0

i = ( i+1 ) mod 4

i = ( 0+1 ) mod 4

i = 1

j = ( j+S[i] ) mod 4

j = ( 0 + S[1] ) mod 4

j = ( 0 + 2 ) mod 4

j = 2

*Swap* S[1] dengan S[2]

Larik S = 1 2 0 3

Hasil larik S = 1 0 2 3

K = S ( S[i] + S[j]) mod 4

K = S ( 0 + 2 ) mod 4

$K_1$ = S[K]

$K_1$ = S[2]

$K_1$ = 2

**Iterasi 2**

i = 1

j = 2

i = ( i+1 ) mod 4

i = ( 1+1 ) mod 4

i = 2

j = ( j+S[i] ) mod 4

j = ( 2 + S[2] ) mod 4

j = ( 2 + 2 ) mod 4

j = 0

*Swap* S[2] dengan S[0]

Larik S = 1 0 2 3

Hasil larik S = 2 0 1 3

K = S ( S[i] + S[j]) mod 4

K = S ( 1 + 2 ) mod 4

$K_2$ = S[K]

$K_2$ = S[3]

$K_2$ = 3

**Iterasi 3**

i = 2

j = 0

i = ( i+1 ) mod 4

i = ( 2+1 ) mod 4

i = 3

j = ( j+S[i] ) mod 4

j = ( 0 + S[3] ) mod 4

j = ( 0 + 3 ) mod 4

j = 3

*Swap* S[3] dengan S[3]

Larik S = 2 0 1 3

Hasil larik S = 2 0 1 3

K = S ( S[i] + S[j]) mod 4

K = S ( 3 + 3 ) mod 4

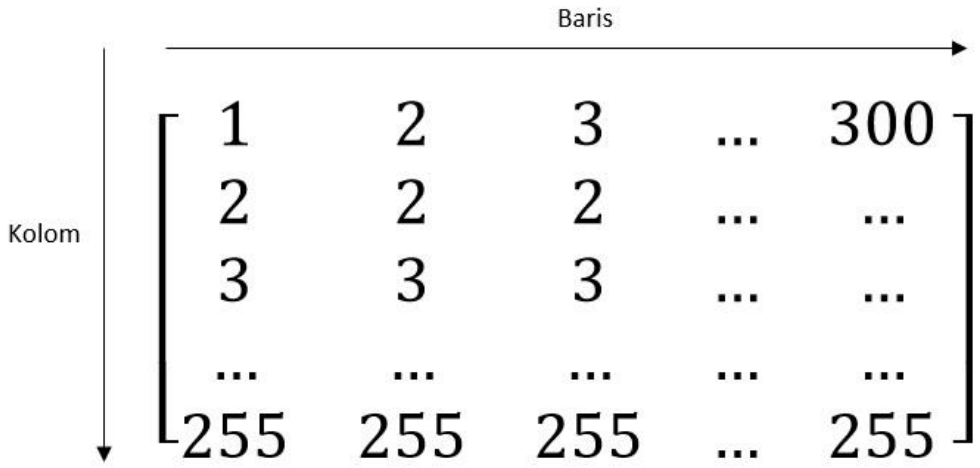
$K_3$ = S[K]

$K_3$ = S[2]

$K_3$ = 1

Proses ketiga, proses peng-*XOR*-an :

Pada proses ini terdapat 3 tahap, tahap yang pertama adalah mengubah nilai *red*, *green*, *blue* dan *key stream* ke bentuk biner, tahap yang kedua adalah meng-*XOR*-kan nilai *red*, *green* dan *blue* dengan nilai *keystream*, kemudian tahap yang terakhir adalah mengubah kembali bilangan biner ke desimal. Metode yang digunakan pada peng-*XOR*-an adalah N-Piksel RGB, yaitu dengan pemilihan dilakukan ke beberapa piksel sekaligus.




**Gambar 4.15** Baris dan Kolom

Misal pada gambar 4.15 dipilih piksel baris 1 kolom 1 sampai ke piksel baris 1 ke kolom 255, piksel baris 2 kolom 1 sampai piksel baris 2 kolom 255, dan begitu seterusnya sampai baris 300 kolom ke 255. Jadi metode yang digunakan adalah mengambil nilai piksel perkolom.

1) Mengubah nilai RGB ke bentuk biner.

Piksel 1

R = 111 = 01101111

G = 144 = 10010000

B = 197 = 11000101

Piksel 2

R = 111 = 01101111

G = 144 = 10010000

B = 197 = 11000101

Piksel 3

R = 111 = 01101111

G = 144 = 10010000

B = 197 = 11000101

Kunci / *keystream*

$K_1$ = 2 = 00000010

$K_2$ = 3 = 00000011

$K_3$ = 1 = 00000001

2) Proses *XOR*

Piksel 1

R = 111 = 01101111

$K_1$ = 2 = 00000010

---

= 01101101

G = 144 = 10010000

$K_1$ = 2 = 00000010

---

= 10010010

B = 197 = 11000101

$K_1$ = 2 = 00000010

---

= 11000111

Piksel 2

R = 111 = 01101111

$K_2$ = 3 = 00000011

---

= 01101100

G = 144 = 10010000

$K_2$ = 3 = 00000011

---

= 10010011

B = 197 = 11000101

$K_2$ = 3 = 00000011

---

= 11000110

Piksel 3

R = 111 = 01101111

$K_3$ = 1 = 00000001

---

= 01101110

G = 144 = 10010000

$K_3$ = 1 = 00000001

---

= 10010001

B = 197 = 11000101

$K_3$ = 1 = 00000001

---

= 11000100

3) Mengubah nilai biner ke bentuk desimal.

Piksel 1

R = 01101101 = 109

G = 10010010 = 146

B = 11000111 = 199

Piksel 2

R = 01101100 = 108

G = 10010011 = 147

B = 11000110 = 198

Piksel 3

R = 01101110 = 110

G = 10010001 = 145

B = 11000100 = 196

Jadi nilai dari *cipherimage* adalah :

Piksel 1

R = 109

G = 146

B = 199

Piksel 2

R = 108

G = 147

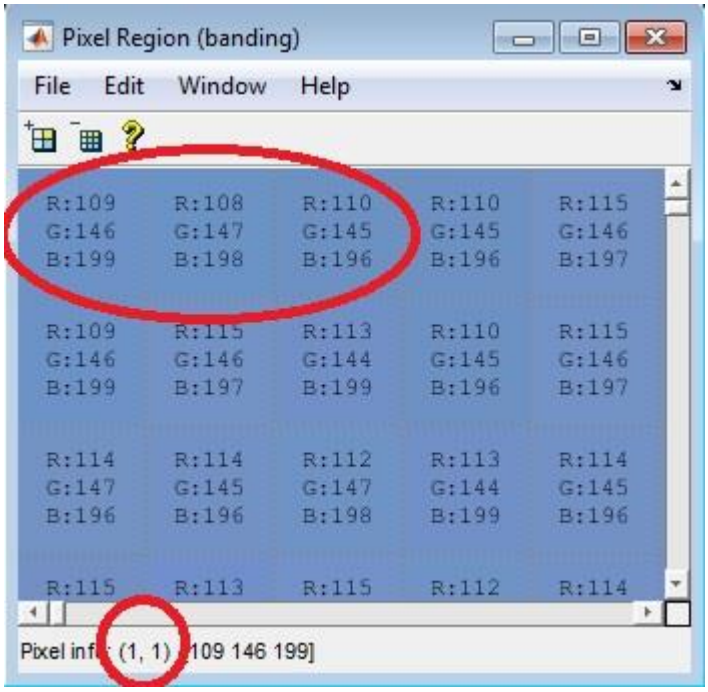
B = 198

Piksel 3

R = 110

G = 145

B = 196




**Gambar 4.16** *Cipher* Nilai Piksel Pada Gambar Sampel Dengan RC4

Berikut hasil gambar *cipher* yang menggunakan larik S = 255 :



**Gambar 4.17** Gambar *Cipher* Secara Keseluruhan

Adapun pada proses dekripsi, yaitu dengan cara meng-*XOR*-kan kembali *cipherimage* dengan *keystream*.

1) Mengubah nilai RGB ke bentuk biner

Piksel 1

R = 109 = 01101101

G = 146 = 10010010

B = 199 = 11000111

Piksel 2

R = 108 = 01101100

G = 147 = 10010011

B = 198 = 11000110

Piksel 3

R = 110 = 01101110

G = 145 = 10010001

B = 196 = 11000100

2) Proses *XOR*

Piksel 1

R = 109 = 01101101

$K_1$ = 2 = 00000010

---

= 01101111

G = 146 = 10010010

$K_1$ = 2 = 00000010

---

= 10010000

B = 199 = 11000111

$K_1$ = 2 = 00000010

---

= 11000101

Piksel 2

R = 108 = 01101100

$K_2$ = 3 = 00000011

---

= 01101111

G = 147 = 10010011

$K_2$ = 3 = 00000011

---

= 10010000

B = 198 = 11000110

$K_2$ = 3 = 00000011

---

= 11000101

Piksel 3

R = 110 = 01101110

$K_3$ = 1 = 00000001

---

= 01101111

G = 145 = 10010001

$K_3$ = 1 = 00000001

---

= 10010000

B = 196 = 11000100

$K_3$ = 1 = 00000001

---

= 11000101

3) Mengubah nilai biner ke bentuk desimal

Piksel 1

R = 01101111 = 111

G = 10010000 = 144

B = 11000101 = 197

Piksel 2

R = 01101111 = 111

G = 10010000 = 144

B = 11000101 = 197

Piksel 3

R = 01101111 = 111

G = 10010000 = 144

B = 11000101 = 197

Jadi nilai dari proses dekripsi adalah :

Piksel 1

R = 111

G = 144

B = 197

Piksel 2

R = 111

G = 144

B = 197

Piksel 3

R = 111

G = 144

B = 197

Berikut gambar dekripsi secara keseluruhan :



**Gambar 4.18** Dekripsi Secara Keseluruhan

**b. Algoritma Elgamal**

Proses enkripsi pada algoritma Elgamal terdapat 2 tahap, yaitu *generate key process* dan enkripsi. Metode yang digunakan dalam mengenkripsi nilai piksel pada gambar adalah dengan pemilihan yang dilakukan pada beberapa piksel sekaligus. Namun pemrosesan dilakukan *persegment* sesuai dengan *segment* rgb-nya, yaitu pemrosesan dilakukan pada tiap nilai piksel yaitu *red, green, blue* dan pemrosesan dilakukan hanya sekali saja.

Metode ini sama dengan metode *N-pixel segment* rgb, hanya saja pada proses *N-pixel segment* rgb dilakukan sebanyak 3 kali pemrosesan *persegment-nya*.

Proses pertama, yaitu *generate key*.

p = 257

g = 99

x = 76

$y = g^x \ mod \ p$

$= 99^{76} \ mod \ 257$

y = 120

Proses yang kedua, yaitu enkripsi dengan menggunakan rumus :

$y^k$ **plainimage mod* p

Piksel 1

R = 111

G = 144

B = 197

Piksel 2

R = 111

G = 144

B = 197

Piksel 3

R = 111

G = 144

B = 197

Nilai k = 17

Untuk piksel 1

R = $120^{17}$ *111 *mod* 257

= 44

G = $120^{17}$ *144 *mod* 257

= 196

B = $120^{17}$ *197 *mod* 257

= 4

Untuk piksel 2

R = $120^{17}$ *111 *mod* 257

= 44

G = $120^{17}$ *144 *mod* 257

= 196

B = $120^{17}$ *197 *mod* 257

= 4

Untuk piksel 3

R = $120^{17}$ *111 *mod* 257

= 44

G = $120^{17}$ *144 *mod* 257

= 196

B = $120^{17}$ *197 *mod* 257

= 4

Piksel 1

R = 44

G = 196

B = 4

Piksel 2

R = 44

G = 196

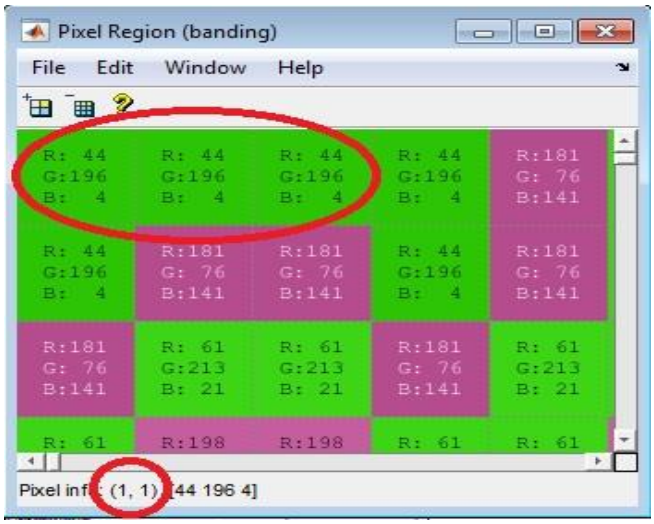
B = 4

Piksel 3

R = 44

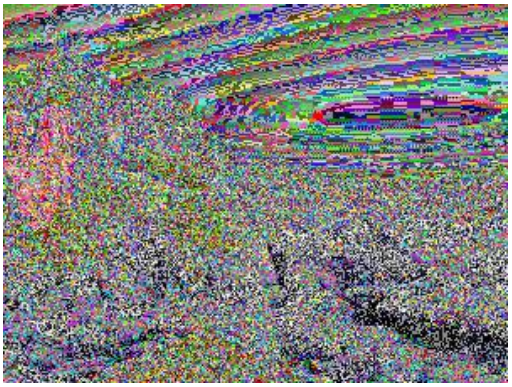
G = 196

B = 4




**Gambar 4.19** *Cipher* Nilai Piksel Pada Gambar Sampel Dengan Elgamal

Berikut gambar *cipher* secara keseluruhan :



**Gambar 4.20** *Cipher* Secara Keseluruhan

Adapun proses dekripsi pada algoritma Elgamal , yaitu :

1) Mencari nilai ‘a’ dengan menggunakan rumus :

$a = g^k \ mod \ p$

$= 99^{17} \ mod \ 257$

$= 139$

2) Mencari *invers* dari nilai ‘a’ dengan menggunakan rumus :

$(a)^{-1} = a^{p-1-x}$ *mod* p

$= 139^{257-1-76}$ *mod* 257

= 242

3) Melakukan proses dekripsi pada tiap nilai piksel dengan menggunakan rumus :

*Cipherimage**$(a)^{-1}$ *mod* p

Untuk piksel 1

R = 44*242 *mod* 257

= 111

G = 196*242 *mod* 257

= 144

B = 4*242 *mod* 257

= 197

Untuk piksel 2

R = 44*242 *mod* 257

= 111

G = 196*242 *mod* 257

= 144

B = 4*242 *mod* 257

= 197

Untuk piksel 3

R = 44*242 *mod* 257

= 111

G = 196*242 *mod* 257

= 144

B = 4*242 *mod* 257

= 197

Jadi nilai dari proses dekripsi adalah :

Piksel 1

R = 111

G = 144

B = 197

Piksel 2

R = 111

G = 144

B = 197

Piksel 3

R = 111

G = 144

B = 197

Berikut gambar dekripsi secara keseluruhan :



**Gambar 4.21** Dekripsi Secara Keseluruhan

**c. Gabungan algoritma RC4 dan Elgamal**

Proses enkripsi pada gabungan algoritma RC4 dan Elgamal dilakukan dengan cara terlebih dahulu melakukan enkripsi dengan menggunakan algoritma RC4 kemudian hasil enkripsi tersebut kembali dienkripsi menggunakan algoritma Elgamal. Adapun proses dekripsi, terlebih dahulu proses dekripsi menggunakan algoritma Elgamal kemudian hasil dekripsi tersebut kembali didekripsi menggunakan algoritma RC4.

Adapun nilai piksel yang dijadikan sampel sebagai berikut :

Piksel 1

R = 111

G = 144

B = 197

1) Proses enkripsi

Melakukan enkripsi menggunakan algoritma RC4 :

Larik S = 4, larik K = 2573, maka $K_1 = 2$

Mengubah nilai RGB dan $K_1$ kebiner :

R = 111 = 01101111

G = 144 = 10010000

B = 197 = 11000101

$K_1$ = 2 = 00000010

Proses *XOR* :

R = 111 = 01101111

$K_1$ = 2 = 00000010

---

= 01101101

G = 144 = 10010000

$K_1$ = 2 = 00000010

---

= 10010010

B = 197 = 11000101

$K_1$ = 2 = 00000010

---

= 11000111

Mengubah bilangan biner ke desimal

R = 01101101 = 109

G = 10010010 = 146

B = 11000111 = 199

Hasil enkripsi RC4 :

R = 109

G = 146

B = 199

Melakukan enkripsi menggunakan algoritma Elgamal :

p = 257, g = 99, x = 76, maka nilai y = 120, k = 17

Melakukan enkripsi menggunakan rumus :

$y^k$ **plainimage mod* p

R = $120^{17}$ *109 *mod* 257

= 27

G = $120^{17}$ *146 *mod* 257

= 213

B = $120^{17}$ *199 *mod* 257

= 21

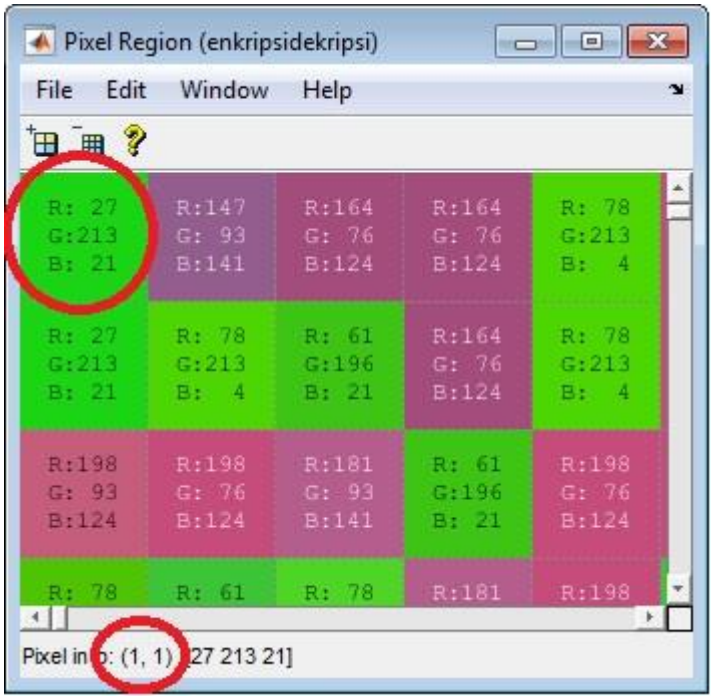
Jadi *cipherimage* dari gabungan agloritma RC4 dan Elgamal adalah :

Piksel 1

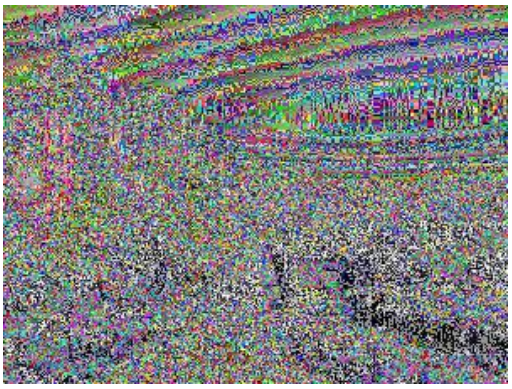
R = 27

G = 213

B = 21




**Gambar 4.22** *Cipher* Nilai Piksel Pada Gambar Sampel Dengan Gabungan



**Gambar 4.23** *Cipher* Secara Keseluruhan

2) Proses dekripsi

Melakukan dekripsi menggunakan algoritma Elgamal :

a = 139, maka nilai $(a)^{-1}$ = 242

Melakukan dekripsi menggunakan rumus :

*Cipherimage*$*(a)^{-1}$ *mod* p

R = 27*242 *mod* 257

= 109

G = 213*242 *mod* 257

= 146

B = 21*242 *mod* 257

= 199

Melakukan dekripsi menggunakan algoritma RC4 :

R = 109

G = 146

B = 199

$K_1 = 2$

Mengubah nilai RGB dan $K_1$ kebiner :

R = 109 = 01101101

G = 146 = 10010010

B = 199 = 11000111

$K_1$ = 2 = 00000010

Proses *XOR* :

R = 109 = 01101101

$K_1$ = 2 = 00000010

---

= 01101111

G = 146 = 10010010

$K_1$ = 2 = 00000010

---

= 10010000

B = 199 = 11000111

$K_1$ = 2 = 00000010

= 11000101

Mengubah nilai biner kedesimal :

Piksel 1

R = 01101111 = 111

G = 10010000 = 144

B = 11000101 = 197



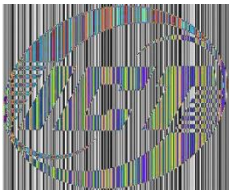
**Gambar 4.24** Dekripsi Secara Keseluruhan

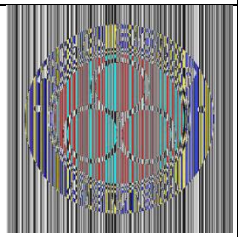
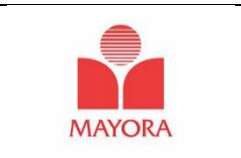
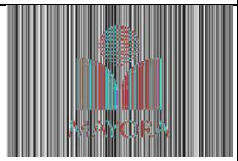
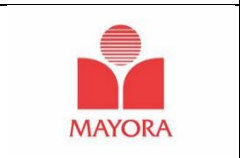
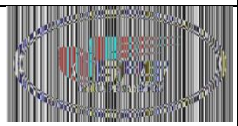
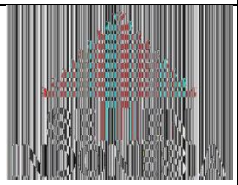
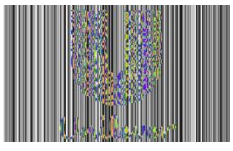
**d. Hasil percobaan**

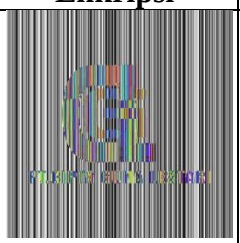
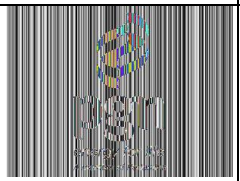
1) Algoritma RC4

Inisialisasi larik S yang digunakan adalah S = 0-255, dengan nilai K = 2573.

**Tabel 4.2** Hasil Percobaan Menggunakan RC4

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 1 |  |  |  | Berhasil |

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 2 |  |  |  | Berhasil |
| 3 |  |  |  | Berhasil |
| 4 |  |  |  | Berhasil |
| 5 |  |  |  | Berhasil |
| 6 |  |  |  | Berhasil |
| 7 |  |  |  | Berhasil |
| 8 |  |  |  | Berhasil |

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 9 |  |  |  | Berhasil |
| 10 |  |  |  | Berhasil |

**Tabel 4.3** Ukuran Piksel Hasil Percobaan Menggunakan RC4

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 1 | 300x246 | 300x246 | 300x246 | Sesuai |
| 2 | 300x165 | 300x165 | 300x165 | Sesuai |
| 3 | 300x300 | 300x300 | 300x300 | Sesuai |
| 4 | 300x200 | 300x200 | 300x200 | Sesuai |
| 5 | 300x152 | 300x152 | 300x152 | Sesuai |
| 6 | 300x177 | 300x177 | 300x177 | Sesuai |
| 7 | 300x230 | 300x230 | 300x230 | Sesuai |
| 8 | 300x179 | 300x179 | 300x179 | Sesuai |
| 9 | 300x300 | 300x300 | 300x300 | Sesuai |
| 10 | 300x225 | 300x225 | 300x225 | Sesuai |

Pada hasil percobaan enkripsi dan dekripsi menggunakan algoritma RC4 yang ditunjukkan pada tabel diatas terlihat gambar yang sebelum dienkripsi, setelah dienkripsi dan setelah didekripsi. Pada percobaan ini gambar yang telah dienrkipsi kemudian dikembalikan kebentuk aslinya atau didekripsi dapat kembali secara sempurna tanpa adanya perubahan warna pada nilai piksel maupun perubahan ukuran piksel baik itu setelah enkripsi maupun setelah didekripsi.

2) Algoritma Elgamal

Berikut nilai-nilai yang digunakan pada percobaan 1:

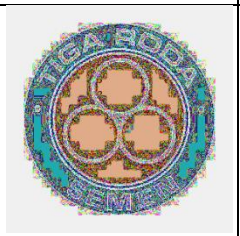
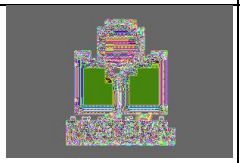
p = 257

g = 99

x = 76

k = 17

**Tabel 4.4** Hasil Percobaan Menggunakan Elgamal percobaan 1

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 1 |  |  |  | Berhasil |
| 2 |  |  |  | Berhasil |
| 3 |  |  |  | Berhasil |
| 4 |  |  |  | Berhasil |
| 5 |  |  |  | Berhasil |

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 6 |  |  |  | Berhasil |
| 7 |  |  |  | Berhasil |
| 8 |  |  |  | Berhasil |
| 9 |  |  |  | Berhasil |
| 10 |  |  |  | Berhasil |

**Tabel 4.5** Ukuran Piksel Hasil Percobaan Menggunakan Elgamal Percobaan 1

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 1 | 300x246 | 300x246 | 300x246 | Sesuai |
| 2 | 300x165 | 300x165 | 300x165 | Sesuai |
| 3 | 300x300 | 300x300 | 300x300 | Sesuai |
| 4 | 300x200 | 300x200 | 300x200 | Sesuai |
| 5 | 300x152 | 300x152 | 300x152 | Sesuai |
| 6 | 300x177 | 300x177 | 300x177 | Sesuai |
| 7 | 300x230 | 300x230 | 300x230 | Sesuai |
| 8 | 300x179 | 300x179 | 300x179 | Sesuai |
| 9 | 300x300 | 300x300 | 300x300 | Sesuai |

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 10 | 300x225 | 300x225 | 300x225 | Sesuai |

Pada hasil percobaan 1 enkripsi dan dekripsi menggunakan algoritma Elgamal yang ditunjukkan pada tabel diatas terlihat gambar yang sebelum dienkripsi, setelah dienkripsi dan setelah didekripsi. Pada percobaan ini gambar yang telah dienrkipsi kemudian dikembalikan kebentuk aslinya atau didekripsi dapat kembali secara sempurna tanpa adanya perubahan warna pada nilai piksel maupun perubahan ukuran piksel baik itu setelah enkripsi maupun setelah didekripsi.

Berikut nilai-nilai yang digunakan pada percobaan 2:

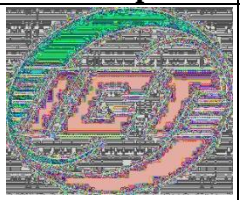
p = 257

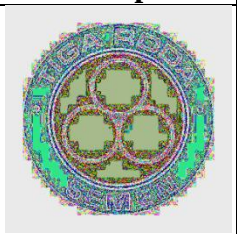
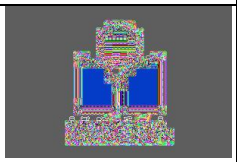
g = 187

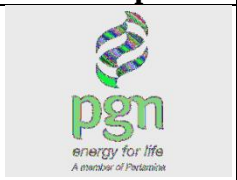
x = 173

k = 73

**Tabel 4.6** Hasil Percobaan Menggunakan Elgamal Percobaan 2

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 1 |  |  |  | Tidak Sesuai |
| 2 |  |  |  | Berhasil |

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 3 |  |  |  | Berhasil |
| 4 |  |  |  | Berhasil |
| 5 |  |  |  | Berhasil |
| 6 |  |  |  | Berhasil |
| 7 |  |  |  | Berhasil |
| 8 |  |  |  | Berhasil |
| 9 |  |  |  | Tidak Sesuai |

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 10 |  |  |  | Tidak Sesuai |

**Tabel 4.7** Ukuran Piksel Hasil Percobaan Menggunakan Elgamal Percobaan 2

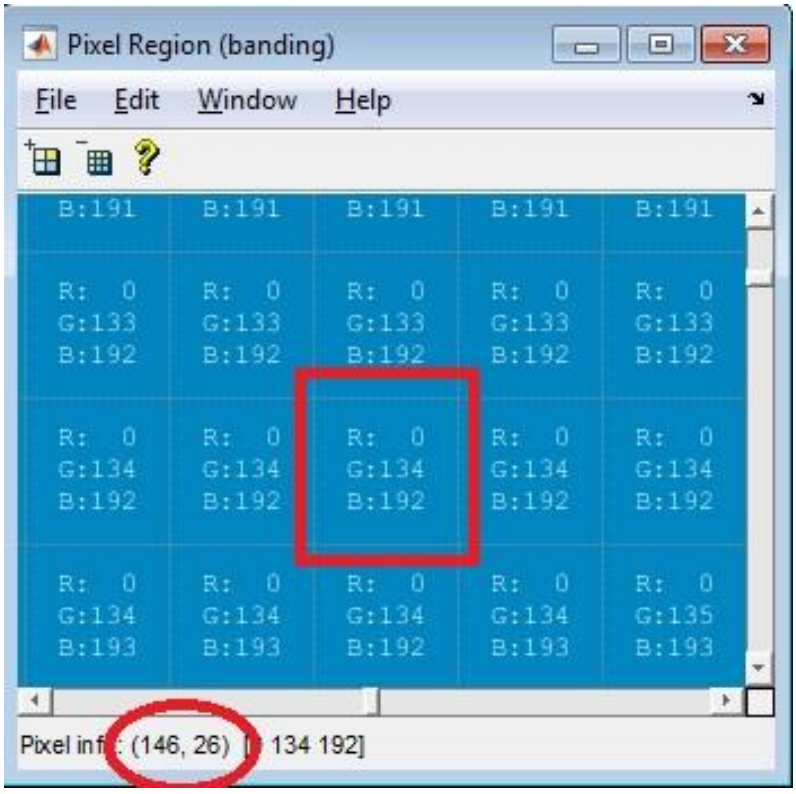
| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 1 | 300x246 | 300x246 | 300x246 | Sesuai |
| 2 | 300x165 | 300x165 | 300x165 | Sesuai |
| 3 | 300x300 | 300x300 | 300x300 | Sesuai |
| 4 | 300x200 | 300x200 | 300x200 | Sesuai |
| 5 | 300x152 | 300x152 | 300x152 | Sesuai |
| 6 | 300x177 | 300x177 | 300x177 | Sesuai |
| 7 | 300x230 | 300x230 | 300x230 | Sesuai |
| 8 | 300x179 | 300x179 | 300x179 | Sesuai |
| 9 | 300x300 | 300x300 | 300x300 | Sesuai |
| 10 | 300x225 | 300x225 | 300x225 | Sesuai |

Pada hasil percobaan 2 enkripsi dan dekripsi menggunakan algoritma Elgamal yang ditunjukkan pada tabel diatas terlihat gambar yang sebelum dienkripsi, setelah dienkripsi dan setelah didekripsi. Pada percobaan ini gambar yang telah dienrkipsi kemudian dikembalikan kebentuk aslinya atau didekripsi terdapat 3 gambar yang tidak kembali kebentuk aslinya yaitu pada gambar nomor 1, 9 dan 10.

Hal ini dikarenakan pada saat proses enkripsi pada nilai piksel gambar tesrebut, hasil dari proses enkripsi tersebut menghasilkan nilai yang melibihi nilai maksimal pada nilai tiap rgb-nya, yaitu lebih besar dari 255. Kemudian setiap proses yang nilainya melebihi 255 maka akan diubah ke nilai terdekat dari nilai

maksimal tiap r, g, b yaitu 0-255. Fungsi ini dinamakan *uint8* atau *Unsigned 8-bit integer* pada matlab yang nilainya adalah 0 sampai $2^8$-1 atau 255.

Agar lebih jelasnya, kita akan ambil 1 piksel pada gambar nomor 1 pada baris ke 146 kolom 26 sebagai sampel yang akan digunakan untuk perhitungan manual.




**Gambar 4.25** Nilai Awal Pada Piksel Gambar Nomor 1

p = 257

g = 187

x = 173

k = 73

$y = g^x \bmod p$

$= 187^{173} \bmod 257$

= 35

Piksel (146, 26)

R = 0

G = 134

B = 192

*Cipherimage* = $y^k$ **plainimage mod* p

R = $35^{73}$ *0 *mod* 257

= 0

G = $35^{73}$ *134 *mod* 257

= 256

= *uint8* 256 = 255

B = $35^{73}$ *192 *mod* 257

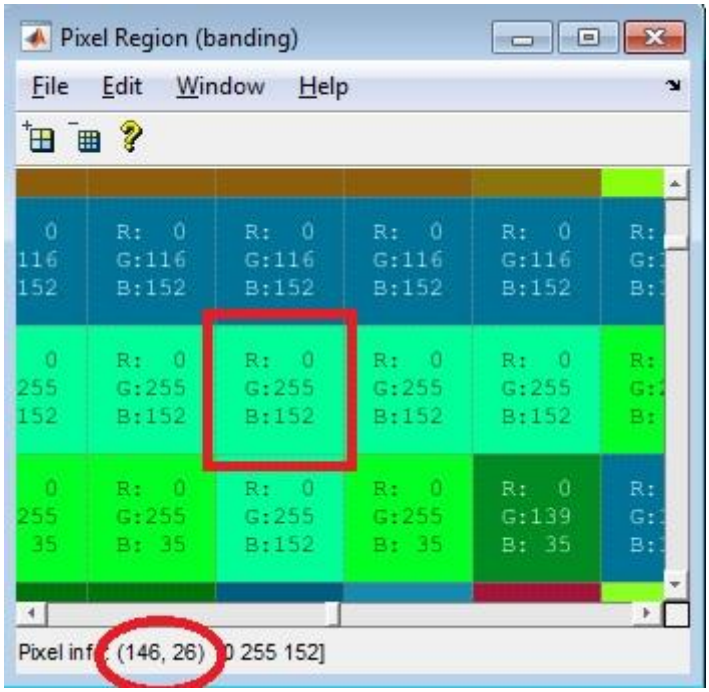
= 152

Jadi nilai dari *cipherimage* piksel (146, 26) :

R = 0

G = 255

B = 152




**Gambar 4.26** Nilai Piksel Pada *Cipherimage* nomor 1

Pada gambar diatas menampilkan nilai piksel yang sesuai dengan perhitungan manual yaitu nilai yang melibihi 255 maka akan diubah kenilai terdekat dari nilai maksimal rgb yaitu 0-255. Maka nilai yang diubah inilah yang akan digunakan untuk proses dekripsi sehingga hasilnya tidak sesuai dengan nilai piksel pada gambar semula. Untuk lebih jelasnya, maka kita akan lakukan proses dekripsi pada nilai *cipherimage* secara manual.

$a = g^k$ *mod* p

$= 187^{73}$ *mod* 257

$= 23$

$(a)^{-1} = a^{p-1-x}$ *mod* p

$= 23^{257-1-173}$ *mod* 257

$= 123$

*Plainimage* = *cipherimage* $*(a)^{-1}$ *mod* p

R = 0 *123 *mod* 257

= 0

G = 255 *123 *mod* 257

= 11

B = 152 *123 *mod* 257

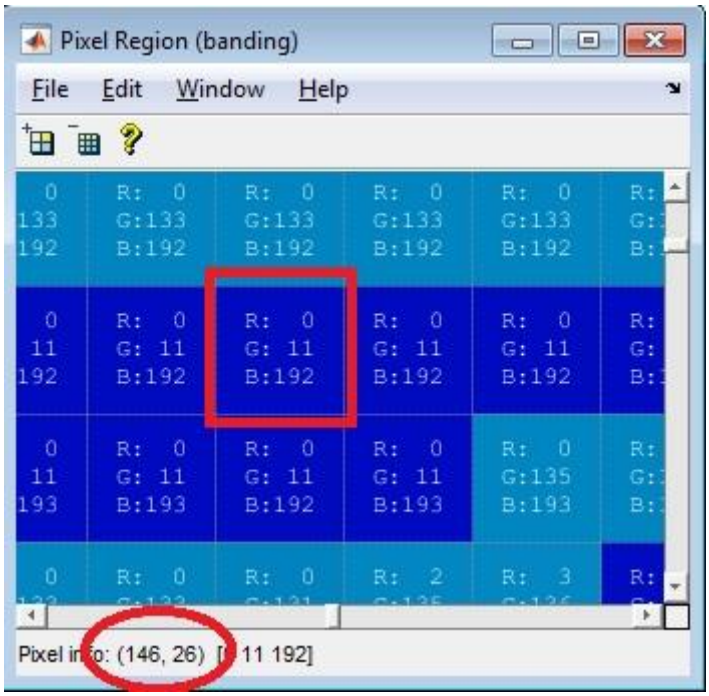
= 192

Jadi nilai dari *plainimage* piksel (146, 26) :

R = 0

G = 11

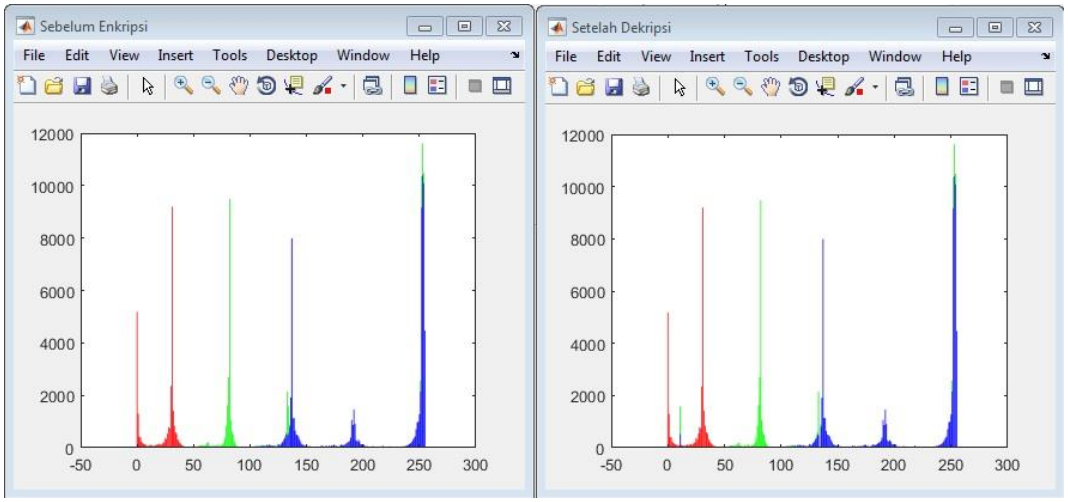
B = 192



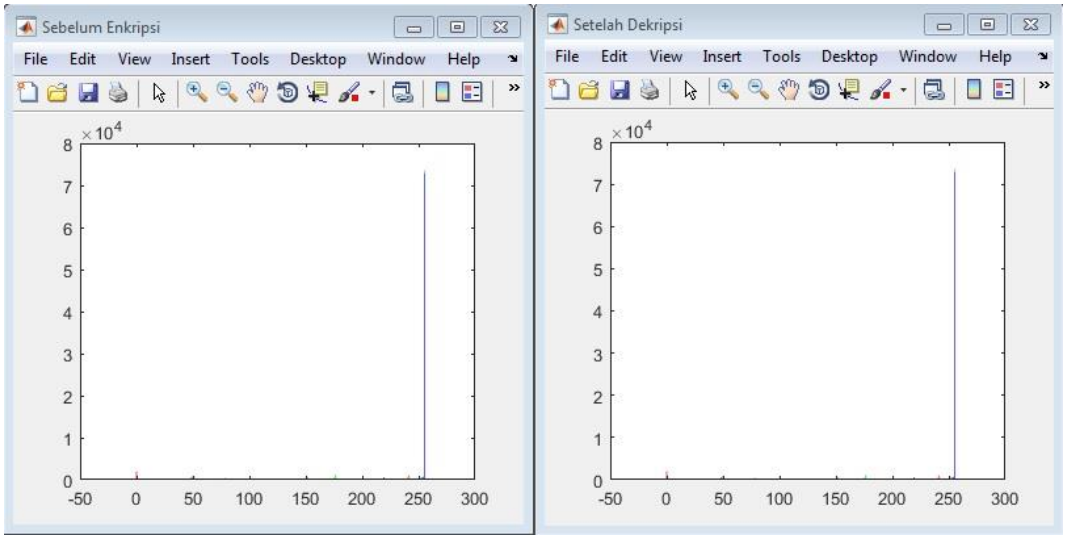

**Gambar 4.27** Nilai Piksel *Plainimage* nomor 1

Pada gambar diatas menunjukkan bahwa nilai piksel yang telah didekripsi tidak dapat kembali sama seperti sebelum dilakukan proses enkripsi pada gambar nomor 1.

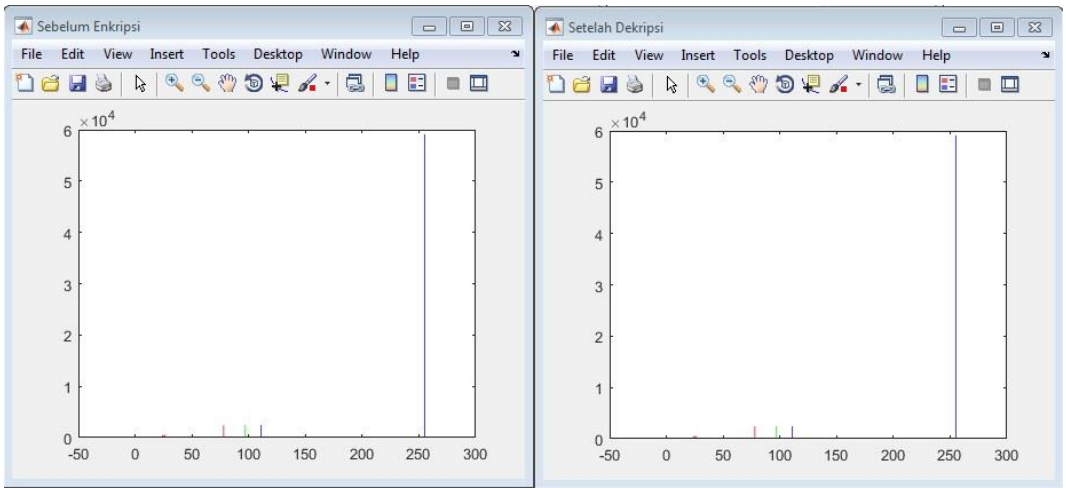
Berikut histogram yang menampilkan nilai rgb pada gambar sebelum enkripsi dan setelah didekripsi, yang digunakan untuk membandingkan nilai piksel pada kedua gambar, apakah terjadi perubahan yang signifikan atau hanya perubahan kecil pada nilai piksel kedua gambar.




**Gambar 4.28** Histogram Perbandingan Nilai Piksel Gambar nomor 1




**Gambar 4.29** Histogram Perbandingan Nilai Piksel Gambar nomor 9

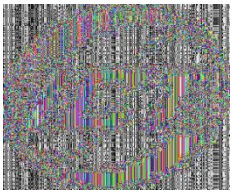
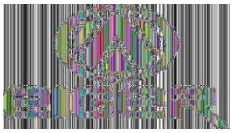
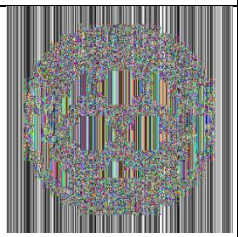
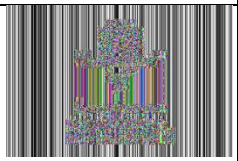
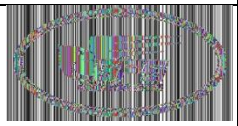
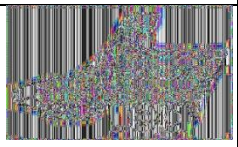
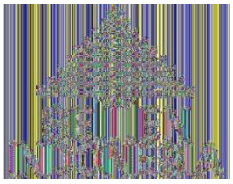


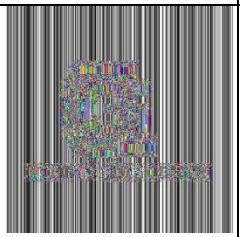
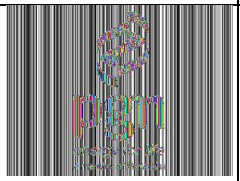

**Gambar 4.30** Histogram Perbandingan Nilai Piksel Gambar nomor 10

Dari perbandingan histogram diatas menunjukkan sedikit perbedaan pada kedua nilai piksel sebelum enkripsi dan setelah dekripsi. Jika dilihat dari perbandingan histogram diatas terdapat perbedaan, namun perbedaan nilai piksel yang ditunjukkan pada histogram tersebut tidak begitu signifikan, dan juga jika dilihat dari tabel gambar, maka gambar dari hasil dekiprisi masih terlihat sama dengan gambar sebelum dekripsi dikarenakan perbedaan atau perubahan nilai piksel yang sangat sedikit.

3) Gabungan algoritma RC4 dan Elgamal

**Tabel 4.8** Hasil Percobaan Menggunakan Gabungan RC4 dan Elgamal

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 1 |  |  |  | Tidak Sesuai |
| 2 |  |  |  | Berhasil |
| 3 |  |  |  | Berhasil |
| 4 |  |  |  | Tidak Sesuai |
| 5 |  |  |  | Berhasil |
| 6 |  |  |  | Tidak Sesuai |
| 7 |  |  |  | Tidak Sesuai |

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 8 |  |  |  | Berhasil |
| 9 |  |  |  | Tidak Berhasil |
| 10 |  |  |  | Tidak Berhasil |

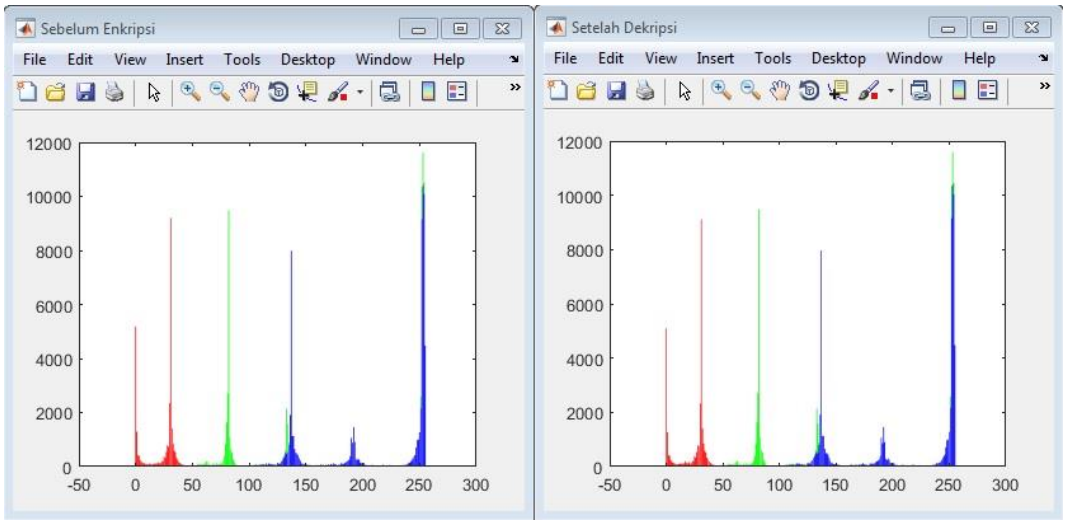
**Tabel 4.9** Ukuran Piksel Hasil Percobaan Menggunakan Gabungan RC4 dan Elgamal

| NO. | Sebelum Enkripsi | Setelah Enkripsi | Setelah Dekripsi | Ket. |
|---|---|---|---|---|
| 1 | 300x246 | 300x246 | 300x246 | Sesuai |
| 2 | 300x165 | 300x165 | 300x165 | Sesuai |
| 3 | 300x300 | 300x300 | 300x300 | Sesuai |
| 4 | 300x200 | 300x200 | 300x200 | Sesuai |
| 5 | 300x152 | 300x152 | 300x152 | Sesuai |
| 6 | 300x177 | 300x177 | 300x177 | Sesuai |
| 7 | 300x230 | 300x230 | 300x230 | Sesuai |
| 8 | 300x179 | 300x179 | 300x179 | Sesuai |
| 9 | 300x300 | 300x300 | 300x300 | Sesuai |
| 10 | 300x225 | 300x225 | 300x225 | Sesuai |

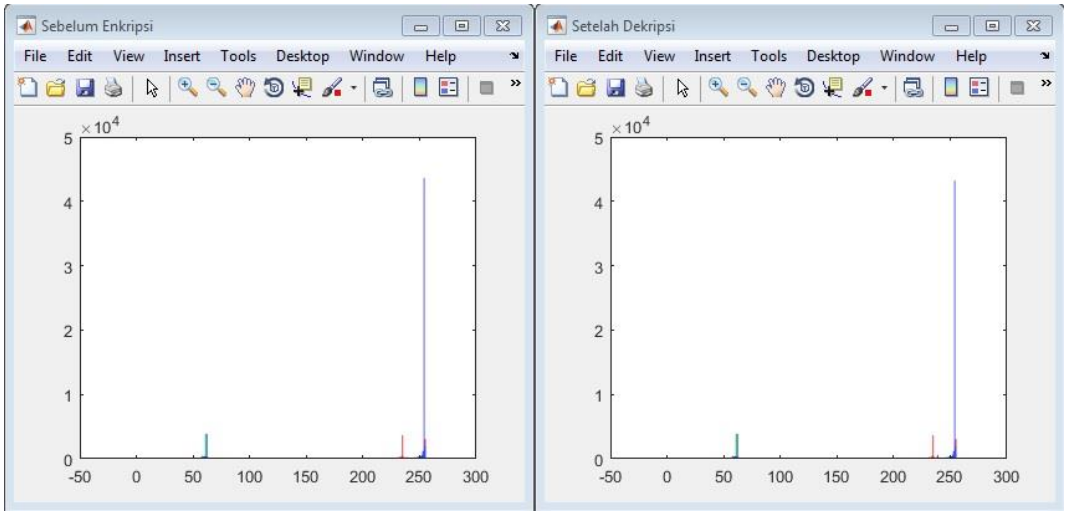
Pada hasil percobaan enkripsi dan dekripsi menggunakan gabungan algoritma RC4 dan Elgamal yang ditunjukkan pada tabel diatas terlihat gambar yang sebelum dienkripsi, setelah dienkripsi dan setelah didekripsi. Pada percobaan ini terdapat 6 gambar yang ketika didekripsi hasilnya tidak sesuai ketika sebelum

enrkipsi. Hal ini dikarenakan nilai hasil enkripsi algoritma Elgamal yang lebih dari 255 maka akan diubah kenilai terdekat dari nilai maksimal r,g,b yaitu 0-255.

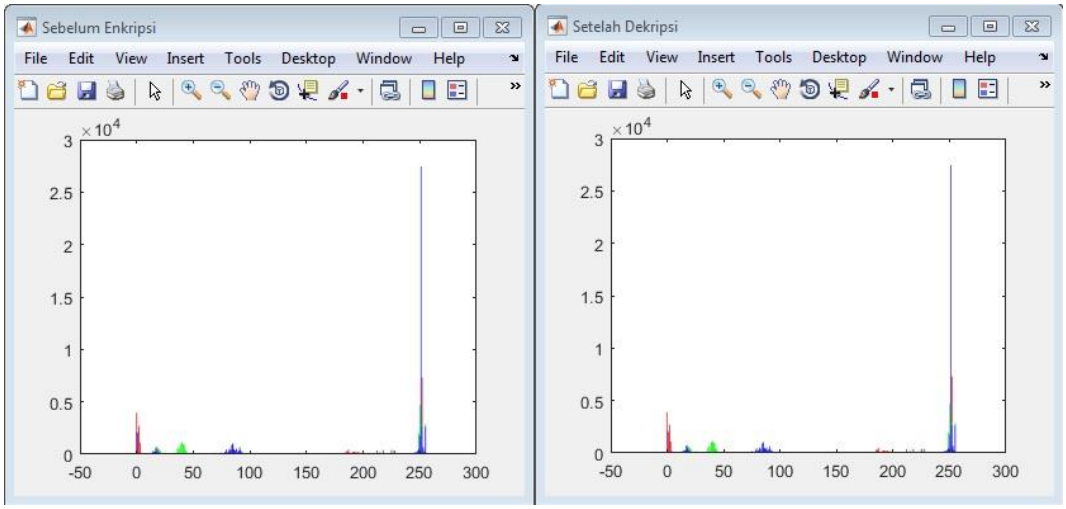
Berikut histogram perbandingan nilai piksel pada 6 gambar tersebut.



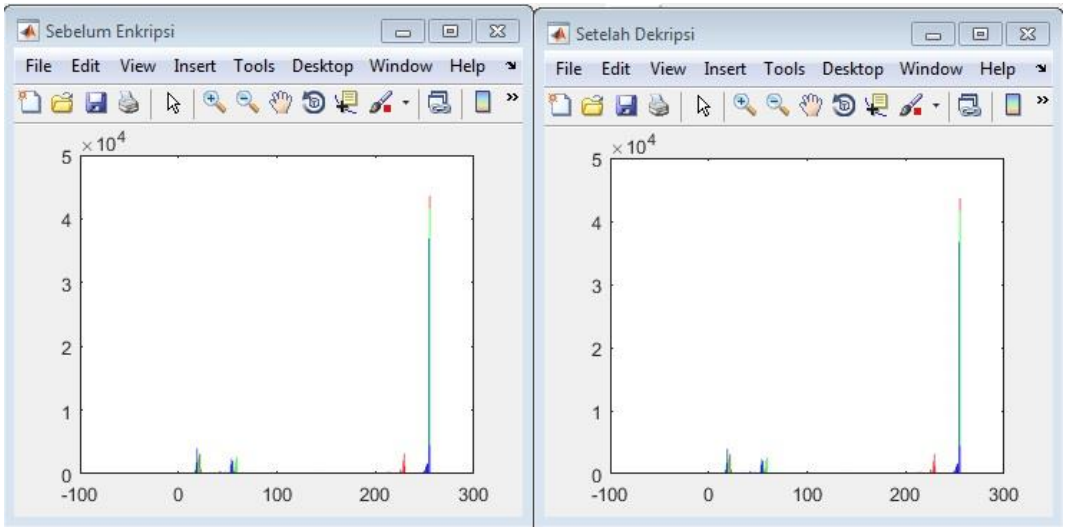

**Gambar 4.31** Histogram Perbandingan Nilai Piksel Gambar nomor 1



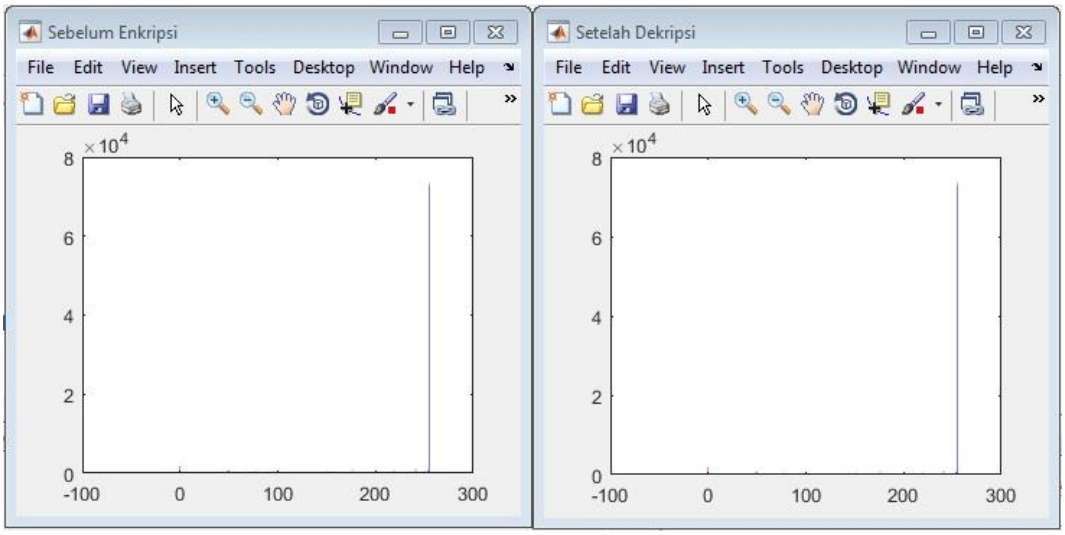

**Gambar 4.32** Histogram Perbandingan Nilai Piksel Gambar nomor 4



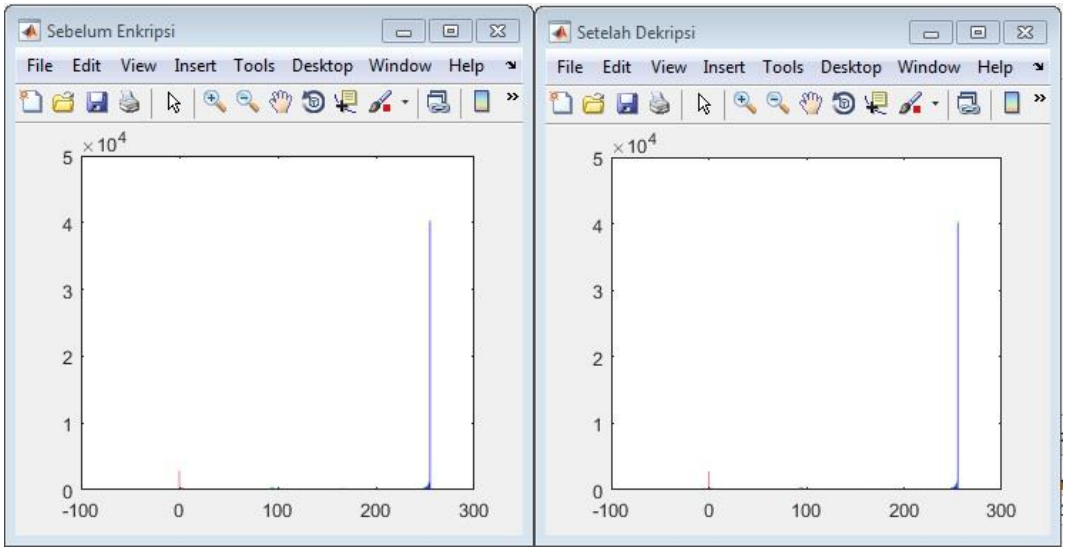

**Gambar 4.33** Histogram Perbandingan Nilai Piksel Gambar nomor 6




**Gambar 4.34** Histogram Perbandingan Nilai Piksel Gambar nomor 7




**Gambar 4.35** Histogram Perbandingan Nilai Piksel Gambar nomor 9




**Gambar 4.36** Histogram Perbandingan Nilai Piksel Gambar nomor 10

## 2. Pengujian

Pengujian aplikasi dilakukan dengan menggunakan metode pengujian yaitu pengujian *blackbox* dan pengujian *whitebox*.

### a. Pengujian *blackbox*

1) Halaman Enkripsi

**Tabel 4.10** Menekan Tombol Enkripsi Pada Saat Data Kosong

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol enrkipsi pada saat belum memasukkan gambar, kunci RC4, dan nilai-nilai Elgamal | ✓ | Berhasil, karena tampil pesan validasi tentang data tidak boleh kosong |
| *Screenshot* | | |
|  | | |

**Tabel 4.11** Ukuran Gambar Terlalu Besar

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika gambar yang dimasukkan memiliki ukuran yang lebih besar dari 300x300 piksel | ✓ | Berhasil, karena tampil pesan validasi tetang ukuran gambar terlalu besar |

| *Screenshot* |
| --- |
|  |



**Tabel 4.12** Menekan Tombol Piksel dan Histogram Saat Gambar Kosong

| Test Faktor | Hasil | Kesimpulan |
| --- | --- | --- |
| Jika menekan tombol piksel dan histogram pada saat gambar masih kosong | ✓ | Berhasil, karena tampil pesan validasi tentang silahkan masukkan gambar |
| *Screenshot* | | |
|  | | |

**Tabel 4.13** Menekan Tombol Enkripsi Saat Nilai g Lebih Besar Dari 257

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol enkripsi pada saat nilai g lebih besar dari 257 | ✓ | Berhasil, karena tampil pesan validasi tentang nilai g harus lebih kecil dari 257 |
| *Screenshot* | | |
|  | | |

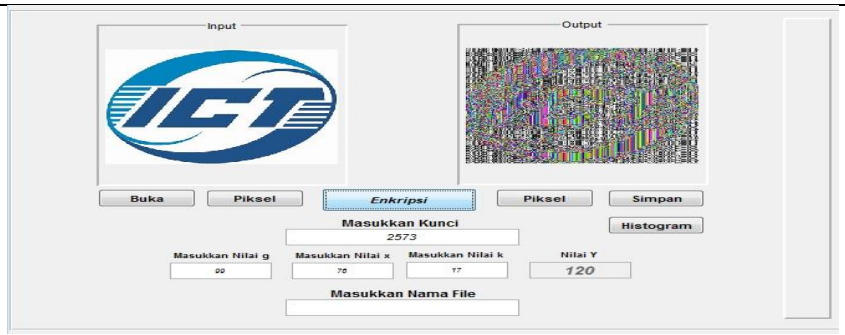
**Tabel 4.14** Menekan Tombol Enkripsi Saat Nilai x Lebih Besar Dari g

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol enkripsi pada saat nilai x lebih besar dari nilai g | ✓ | Berhasil, karena tampil pesan validasi tentang nilai x harus lebih kecil dari nilai g |
| *Screenshot* | | |
|  | | |

**Tabel 4.15** Menekan Tombol Enkripsi Saat Nilai k Lebih besar Dari x

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol enkripsi pada saat nilai k lebih besar dari nilai x | ✓ | Berhasil, karena tampil pesan validasi tentang nilai k harus lebih kecil dari nilai x |
| *Screenshot* | | |
|  | | |

**Tabel 4.16** Proses Enkripsi Berhasil

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika proses enkripsi berhasil | ✓ | Berhasil, karena tidak tampil pesan validasi |
| *Screenshot* | | |
|  | | |

**Tabel 4.17** Menekan Tombol Simpan Saat Data Masih Kosong

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol simpan pada saat data masih kosong | ✓ | Berhasil, karena tampil pesan validasi tentang lakukan proses enkripsi |
| *Screenshot* | | |
|  | | |

**Tabel 4.18** Menekan Tombol Simpan Saat Nama *File* Masih Kosong

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol simpan pada saat nama *file* masih kosong | ✓ | Berhasil, karena tampil pesan validasi tentang lengkapi nama *file* |
| *Screenshot* | | |
|  | | |

**Tabel 4.19** Berhasil Menyimpan Gambar

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika berhasil menyimpan gambar hasil enrkipsi | ✓ | Berhasil, karena tampil pesan validasi tentang berhasil menyimpan gambar |
| *Screenshot* | | |
|  | | |

2) Halaman Dekripsi

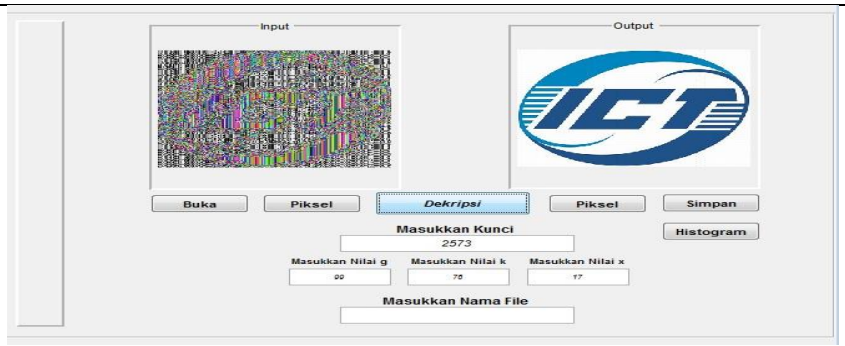
**Tabel 4.20** Menekan Tombol Dekripsi Saat Data Masih Kosong

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol dekripsi pada saat data masih kosong | ✓ | Berhasil, karena tampil pesan validasi tentang data tidak boleh kosong |
| *Screenshot* | | |
|  | | |

**Tabel 4.21** Menekan Tombol Piksel dan Histogram Saat Gambar Kosong

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol piksel dan histogram pada saat gambar masih kosong | ✓ | Berhasil, karena tampil pesan validasi tentang silahkan masukkan gambar |
| *Screenshot* | | |
|  | | |

**Tabel 4.22** Proses Dekripsi Berhasil

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika proses dekripsi berhasil | ✓ | Berhasil, karena tidak tampil pesan validasi |
| *Screenshot* | | |
|  | | |

**Tabel 4.23** Menekan Tombol Simpan Saat Data Masih Kosong

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol simpan pada saat data masih kosong | ✓ | Berhasil, karena tampil pesan validasi tentang lakukan proses dekripsi |
| *Screenshot* | | |
|  | | |

**Tabel 4.24** Menekan Tombol Simpan Saat Nama *File* Masih Kosong

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika menekan tombol simpan pada saat nama *file* masih kosong | ✓ | Berhasil, karena tampil pesan validasi tentang lengkapi nama *file* |
| *Screenshot* | | |
|  | | |

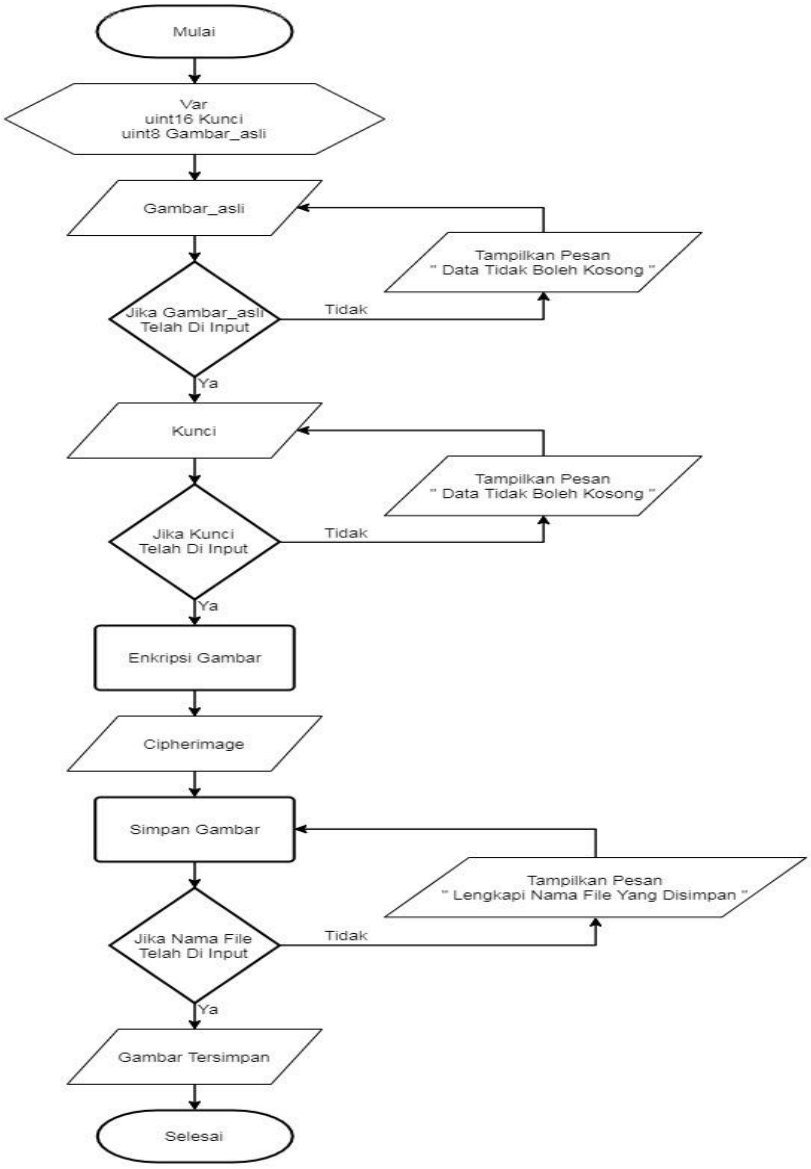
**Tabel 4.25** Berhasil Menyimpan Gambar

| Test Faktor | Hasil | Kesimpulan |
|---|---|---|
| Jika berhasil menyimpan gambar hasil dekripsi | ✓ | Berhasil, karena tampil pesan validasi tentang berhasil menyimpan gambar |
| *Screenshot* | | |
|  | | |

**b. Pengujian *whitebox***

1) Algoritma RC4

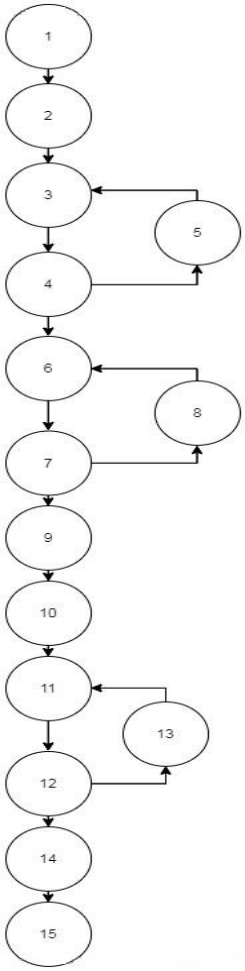
a) *Flowchart* dan *flowgraph* enkripsi RC4




**Gambar 4.37** *Flowchart* enkripsi RC4

Berdasarkan gambar diatas proses enkripsi RC4 dimulai dengan sistem yang akan melakukan inisiasi kemudian proses berikutnya adalah *user* memasukkan gambar asli yang berupa logo perusahaan dan kunci, namun apabila gambar dan kunci belum dimasukkan maka akan tampil pesan seperti gambar diatas. Kemudian proses berikutnya adalah proses enkripsi gambar dan akan menghasilkan *cipherimage*. Setelah itu langkah berikutnya adalah menyimpan gambar yang telah

terenkripsi, namun apabila nama *file* belum dimasukkan maka akan tampil pesan seperti gambar diatas, jika telah dimasukkan maka gambar akan berhasil tersimpan.




**Gambar 4.38** *Flowgraph* enkripsi RC4

Berdasarkan gambar 4.37 maka dapat dilakukan proses perhitungan sebagai berikut :

Menghitung *Cyclomatic Complexcity* V(G) dari *edge* dan *node*

Dengan rumus V(G) = E – N + 2

Dengan *edge* = 17

Dengan *node* = 15

Dengan predikat *node* (P) = 2

Penyelesaian :

V(G) = E – N + 2

= 17 – 15 + 2

= 4

Predikat *node* (P) = P + 1

= 2 + 1

= 3

Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 3

*Independent path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 12 – 14 – 15

*Path* 2 = 1 – 2 – 3 – 4 – 5 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 12 – 14 – 15

*Path* 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 6 – 7 – 9 – 10 – 11 – 12 – 14 – 15

*Path* 4 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 12 – 13 – 11 – 12 – 14 – 15

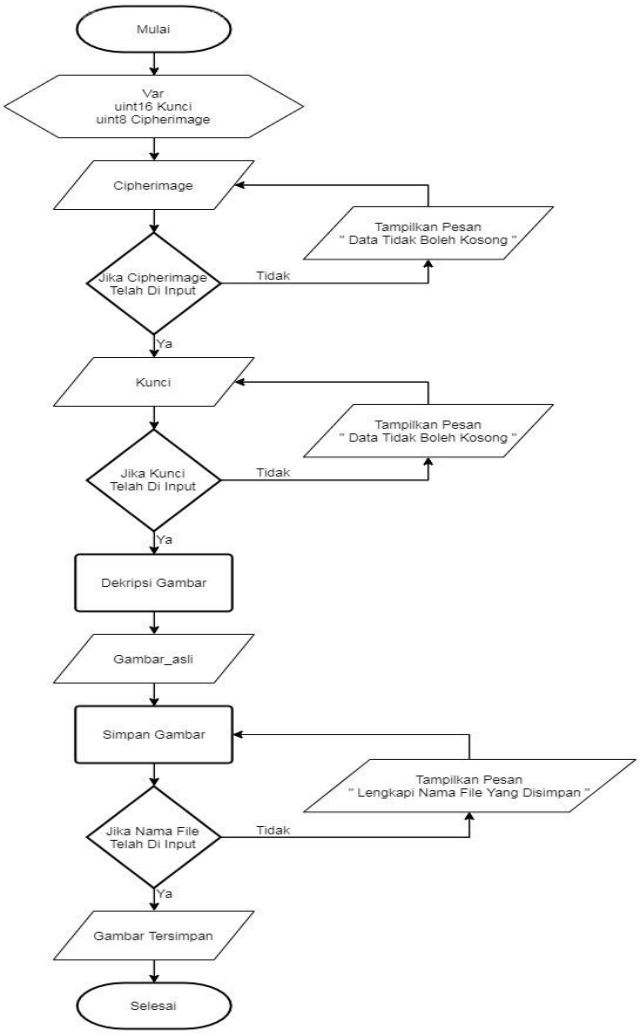
Grafik matriks pada *flowgraph* diatas adalah :

**Tabel 4.26** Grafik Matriks Enkripsi RC4

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **E-1** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 1 | | | | | | | | | | | | | | 1-1=0 |
| 2 | | | 1 | | | | | | | | | | | | | 1-1=0 |
| 3 | | | | 1 | | | | | | | | | | | | 1-1=0 |
| 4 | | | | | 1 | 1 | | | | | | | | | | 2-1=1 |
| 5 | | | 1 | | | | | | | | | | | | | 1-1=0 |
| 6 | | | | | | | 1 | | | | | | | | | 1-1=0 |
| 7 | | | | | | | | 1 | 1 | | | | | | | 2-1=1 |
| 8 | | | | | | 1 | | | | | | | | | | 1-1=0 |
| 9 | | | | | | | | | | 1 | | | | | | 1-1=0 |
| 10 | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 11 | | | | | | | | | | | | 1 | | | | 1-1=0 |
| 12 | | | | | | | | | | | | | 1 | 1 | | 2-1=1 |
| 13 | | | | | | | | | | | 1 | | | | | 1-1=0 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | E-1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14 | | | | | | | | | | | | | | | 1 | 1-1=0 |
| 15 | | | | | | | | | | | | | | | | |
| S U M E+1 | | | | | | | | | | | | | | | | 3+1=4 |

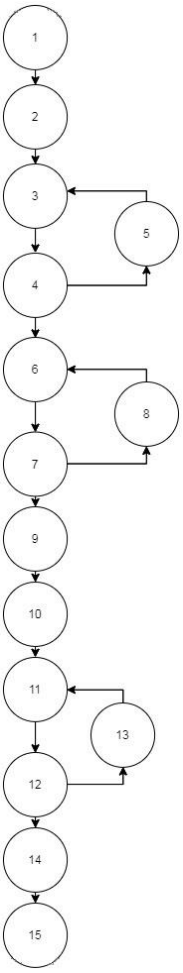
b) *Flowchart* dan *flowgraph* dekripsi RC4




**Gambar 4.39** *Flowchart* dekripsi RC4

Berdasarkan gambar diatas proses dekripsi RC4 dimulai dengan sistem yang akan melakukan inisiasi kemudian proses berikutnya adalah *user* memasukkan *cipherimage* dan kunci, namun apabila cipherimage dan kunci belum dimasukkan

maka akan tampil pesan seperti gambar diatas. Kemudian proses berikutnya adalah proses dekripsi gambar dan akan menghasilkan gambar asli. Setelah itu langkah berikutnya adalah menyimpan gambar yang telah terdekripsi, namun apabila nama *file* belum dimasukkan maka akan tampil pesan seperti gambar diatas, jika telah dimasukkan maka gambar akan berhasil tersimpan.




**Gambar 4.40** *Flowgraph* dekripsi RC4

Berdasarkan gambar 4.39 maka dapat dilakukan proses perhitungan sebagai berikut :

Menghitung *Cyclomatic Complexcity* V(G) dari *edge* dan *node*

Dengan rumus V(G) = E – N + 2

Dengan *edge* = 17

Dengan *node* = 15

Dengan predikat *node* (P) = 2

Penyelesaian :

V(G) = E – N + 2

= 17 – 15 + 2

= 4

Predikat *node* (P) = P + 1

= 2 + 1

= 3

Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 3

*Independent path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 12 – 14 – 15

*Path* 2 = 1 – 2 – 3 – 4 – 5 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 12 – 14 – 15

*Path* 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 6 – 7 – 9 – 10 – 11 – 12 – 14 – 15

*Path* 4 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 12 – 13 – 11 – 12 – 14 – 15

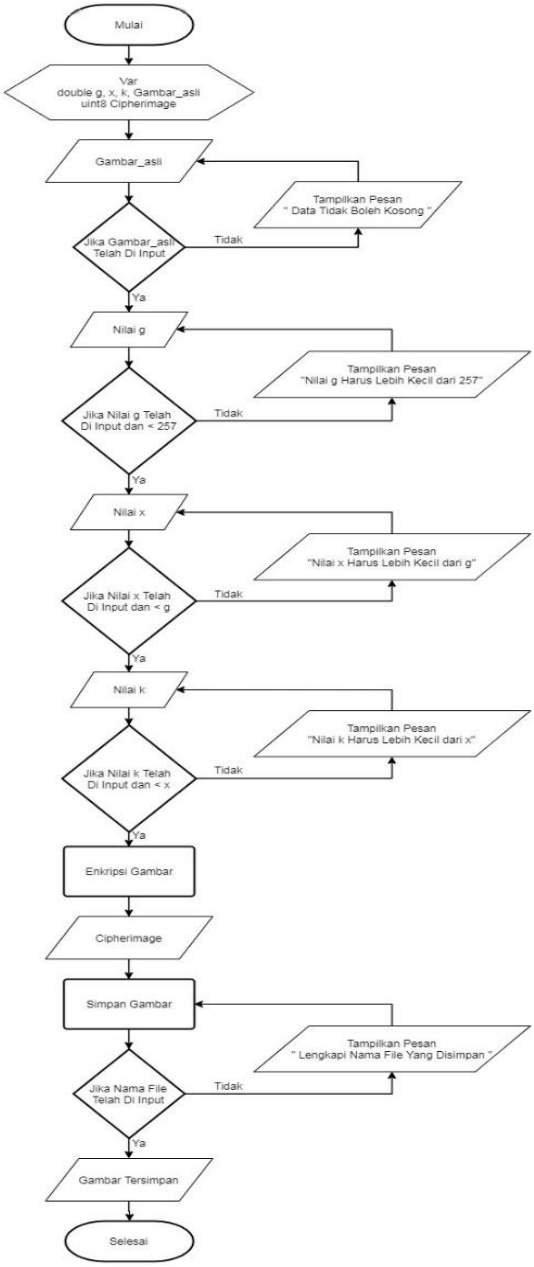
Grafik matriks pada *flowgraph* diatas adalah :

**Tabel 4.27** Grafik Matriks Dekripsi RC4

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **E-1** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 1 | | | | | | | | | | | | | | 1-1=0 |
| 2 | | | 1 | | | | | | | | | | | | | 1-1=0 |
| 3 | | | | 1 | | | | | | | | | | | | 1-1=0 |
| 4 | | | | | 1 | 1 | | | | | | | | | | 2-1=1 |
| 5 | | | 1 | | | | | | | | | | | | | 1-1=0 |
| 6 | | | | | | | 1 | | | | | | | | | 1-1=0 |
| 7 | | | | | | | | 1 | 1 | | | | | | | 2-1=1 |

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **E-1** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | | | | | | 1 | | | | | | | | | | 1-1=0 |
| 9 | | | | | | | | | | 1 | | | | | | 1-1=0 |
| 10 | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 11 | | | | | | | | | | | | 1 | | | | 1-1=0 |
| 12 | | | | | | | | | | | | | 1 | 1 | | 2-1=1 |
| 13 | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 14 | | | | | | | | | | | | | | | 1 | 1-1=0 |
| 15 | | | | | | | | | | | | | | | | |
| S U M E+1 | | | | | | | | | | | | | | | | 3+1=4 |

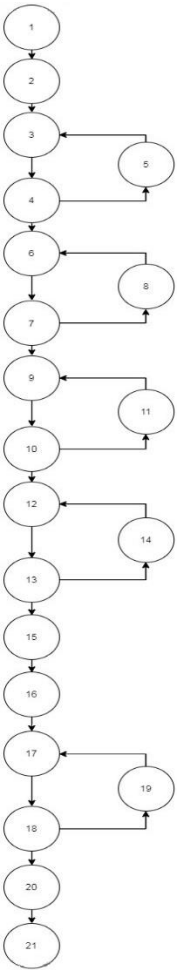
c) *Flowchart* dan *flowgraph* enkripsi Elgamal




**Gambar 4.41** *Flowchart* enkripsi Elgamal

Berdasarkan gambar diatas proses enkripsi Elgamal dimulai dengan sistem yang akan melakukan inisiasi kemudian proses berikutnya adalah *user* memasukkan gambar yang berupa logo dan nilai g, x, dan k, namun apabila gambar

dan nilai-nilai tersebut belum dimasukkan maka akan tampil pesan seperti gambar diatas. Kemudian proses berikutnya adalah proses enkripsi gambar dan akan menghasilkan *cipherimage*. Setelah itu langkah berikutnya adalah menyimpan gambar yang telah terenkripsi, namun apabila nama *file* belum dimasukkan maka akan tampil pesan seperti gambar diatas, jika telah dimasukkan maka gambar akan berhasil tersimpan.




**Gambar 4.42** *Flowgraph* enkripsi Elgamal

Berdasarkan gambar 4.41 maka dapat dilakukan proses perhitungan sebagai berikut :

Menghitung *Cyclomatic Complexcity* V(G) dari *edge* dan *node*

Dengan rumus V(G) = E – N + 2

Dengan *edge* = 25

Dengan *node* = 21

Dengan predikat *node* (P) = 4

Penyelesaian :

V(G) = E – N + 2

= 25 – 21 + 2

= 6

Predikat *node* (P) = P + 1

= 4 + 1

= 5

Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 5

*Independent path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

*Path* 2 = 1 – 2 – 3 – 4 – 5 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

*Path* 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

*Path* 4 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

*Path* 5 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 14 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

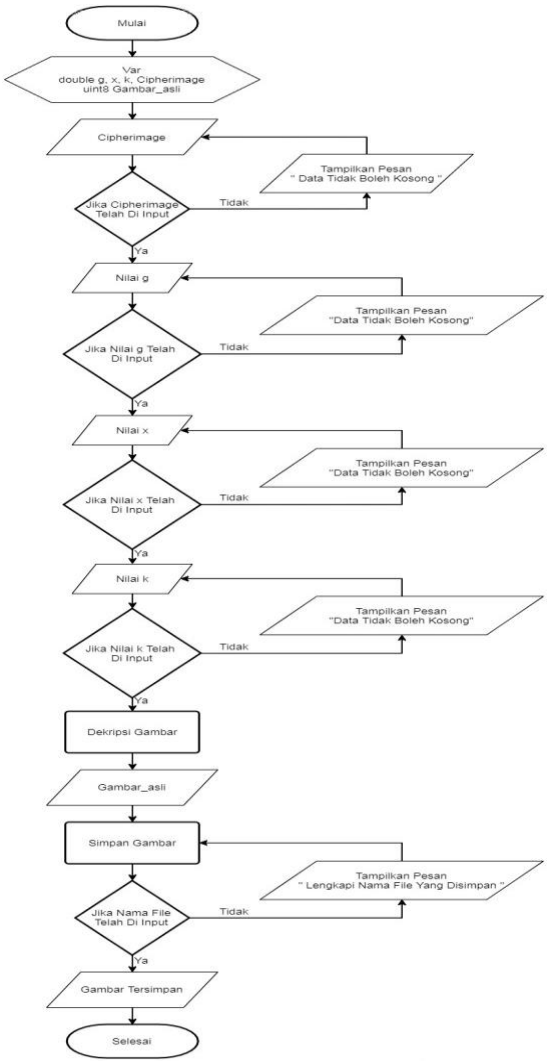
*Path* 6 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 19 – 17 – 18 – 20 – 21

Grafik matriks pada *flowgraph* diatas adalah :

**Tabel 4.28** Grafik Matriks Enkripsi Elgamal

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **E-1** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 1 | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 2 | | | 1 | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 3 | | | | 1 | | | | | | | | | | | | | | | | | | 1-1=0 |
| 4 | | | | | 1 | 1 | | | | | | | | | | | | | | | | 2-1=1 |
| 5 | | | 1 | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 6 | | | | | | | 1 | | | | | | | | | | | | | | | 1-1=0 |
| 7 | | | | | | | | 1 | 1 | | | | | | | | | | | | | 2-1=1 |
| 8 | | | | | | 1 | | | | | | | | | | | | | | | | 1-1=0 |
| 9 | | | | | | | | | | 1 | | | | | | | | | | | | 1-1=0 |
| 10 | | | | | | | | | | | 1 | 1 | | | | | | | | | | 2-1=1 |
| 11 | | | | | | | | | 1 | | | | | | | | | | | | | 1-1=0 |
| 12 | | | | | | | | | | | | | 1 | | | | | | | | | 1-1=0 |
| 13 | | | | | | | | | | | | | | 1 | 1 | | | | | | | 2-1=1 |
| 14 | | | | | | | | | | | | 1 | | | | | | | | | | 1-1=0 |
| 15 | | | | | | | | | | | | | | | | 1 | | | | | | 1-1=0 |
| 16 | | | | | | | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 17 | | | | | | | | | | | | | | | | | | 1 | | | | 1-1=0 |
| 18 | | | | | | | | | | | | | | | | | | | 1 | 1 | | 2-1=1 |
| 19 | | | | | | | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 20 | | | | | | | | | | | | | | | | | | | | | 1 | 1-1=0 |
| 21 | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| SUM E+1 | | | | | | | | | | | | | | | | | | | | | | 5+1=6 |

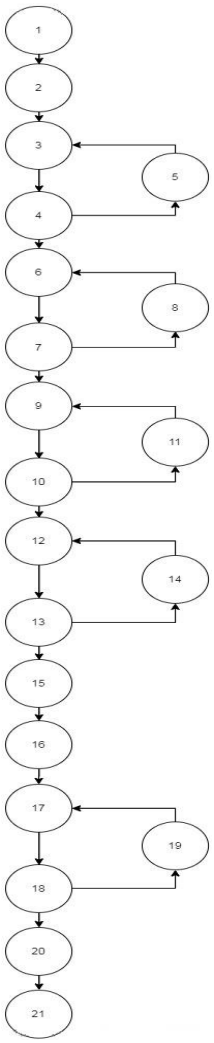
d) *Flowchart* dan *flowgraph* dekripsi Elgamal




**Gambar 4.43** *Flowchart* dekripsi Elgamal

Berdasarkan gambar diatas proses dekripsi Elgamal dimulai dengan sistem yang akan melakukan inisiasi kemudian proses berikutnya adalah *user* memasukkan *cipherimage* dan nilai g, x, dan k, namun apabila gambar dan nilai-nilai tersebut belum dimasukkan maka akan tampil pesan seperti gambar diatas. Kemudian proses berikutnya adalah proses dekripsi gambar dan akan menghasilkan gambar asli. Setelah itu langkah berikutnya adalah menyimpan gambar yang telah

terdekripsi, namun apabila nama *file* belum dimasukkan maka akan tampil pesan seperti gambar diatas, jika telah dimasukkan maka gambar akan berhasil tersimpan.




**Gambar 4.44** *Flowgraph* dekripsi Elgamal

Berdasarkan gambar 4.43 maka dapat dilakukan proses perhitungan sebagai berikut :

Menghitung *Cyclomatic Complexcity* V(G) dari *edge* dan *node*

Dengan rumus V(G) = E – N + 2

Dengan *edge* = 25

Dengan *node* = 21

Dengan predikat *node* (P) = 4

Penyelesaian :

V(G) = E – N + 2

= 25 – 21 + 2

= 6

Predikat *node* (P) = P + 1

= 4 + 1

= 5

Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 5

*Independent path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

*Path* 2 = 1 – 2 – 3 – 4 – 5 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

*Path* 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

*Path* 4 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

*Path* 5 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 14 – 12 – 13 – 15 – 16 – 17 – 18 – 20 – 21

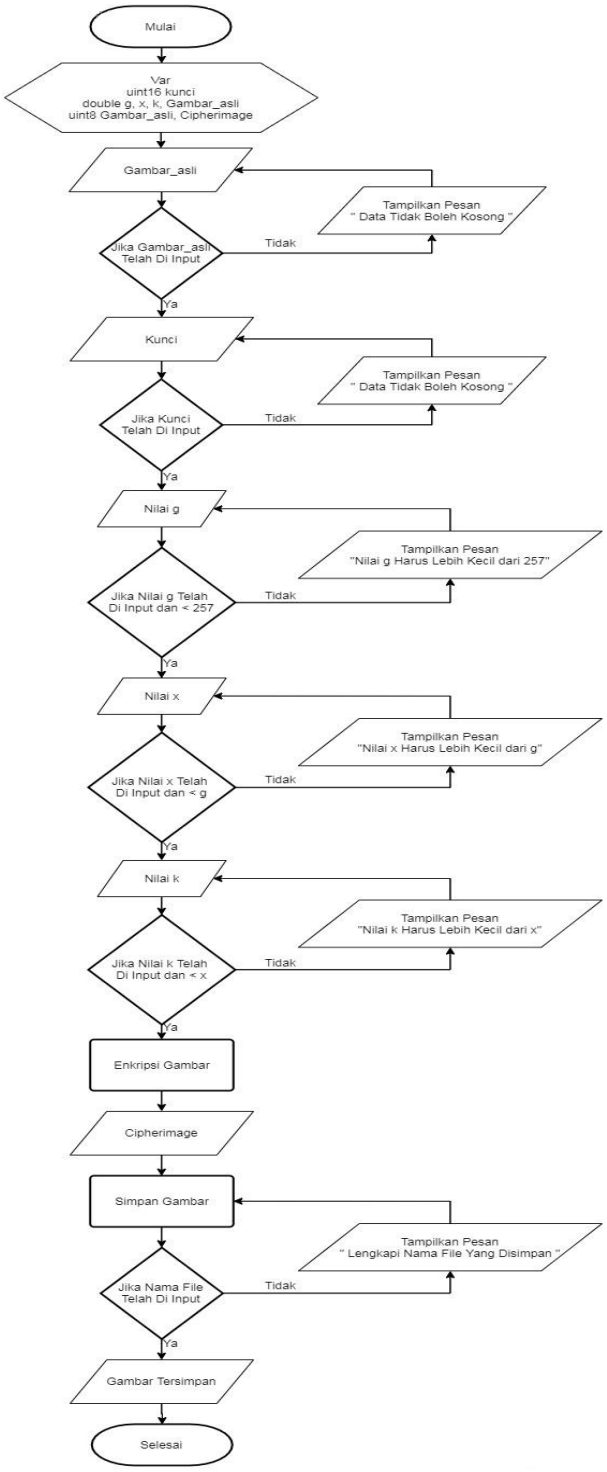
*Path* 6 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 18 – 19 – 17 – 18 – 20 – 21

Grafik matriks pada *flowgraph* diatas adalah :

**Tabel 4.29** Grafik Matriks Dekripsi Elgamal

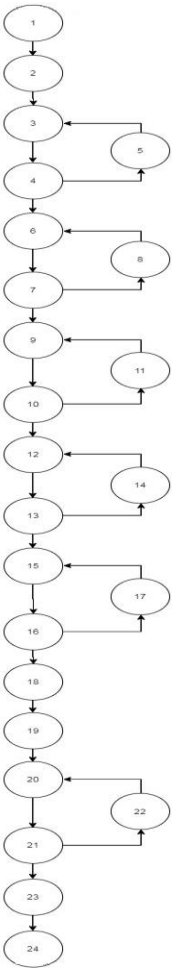
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **E-1** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 1 | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 2 | | | 1 | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 3 | | | | 1 | | | | | | | | | | | | | | | | | | 1-1=0 |
| 4 | | | | | 1 | 1 | | | | | | | | | | | | | | | | 2-1=1 |
| 5 | | | 1 | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 6 | | | | | | | 1 | | | | | | | | | | | | | | | 1-1=0 |
| 7 | | | | | | | | 1 | 1 | | | | | | | | | | | | | 2-1=1 |
| 8 | | | | | | 1 | | | | | | | | | | | | | | | | 1-1=0 |
| 9 | | | | | | | | | | 1 | | | | | | | | | | | | 1-1=0 |
| 10 | | | | | | | | | | | 1 | 1 | | | | | | | | | | 2-1=1 |
| 11 | | | | | | | | | 1 | | | | | | | | | | | | | 1-1=0 |
| 12 | | | | | | | | | | | | | 1 | | | | | | | | | 1-1=0 |
| 13 | | | | | | | | | | | | | | 1 | 1 | | | | | | | 2-1=1 |
| 14 | | | | | | | | | | | | 1 | | | | | | | | | | 1-1=0 |
| 15 | | | | | | | | | | | | | | | | 1 | | | | | | 1-1=0 |
| 16 | | | | | | | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 17 | | | | | | | | | | | | | | | | | | 1 | | | | 1-1=0 |
| 18 | | | | | | | | | | | | | | | | | | | 1 | 1 | | 2-1=1 |
| 19 | | | | | | | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 20 | | | | | | | | | | | | | | | | | | | | | 1 | 1-1=0 |
| 21 | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| SUM E+1 | | | | | | | | | | | | | | | | | | | | | | 5+1=6 |

e) *Flowchart* dan *flowgraph* enkripsi gabungan




**Gambar 4.45** *Flowchart* enkripsi gabungan

Berdasarkan gambar diatas proses enkripsi gabungan dimulai dengan sistem yang akan melakukan inisiasi kemudian proses berikutnya adalah *user* memasukkan gambar yang berupa logo perusahaan dan kunci RC4 serta nilai g, x, k, namun apabila gambar, kunci dan nilai-nilai tersebut belum dimasukkan maka akan tampil pesan seperti gambar diatas. Kemudian proses berikutnya adalah proses enkripsi gambar dan akan menghasilkan *cipherimage*. Setelah itu langkah berikutnya adalah menyimpan gambar yang telah terenkripsi, namun apabila nama *file* belum dimasukkan maka akan tampil pesan seperti gambar diatas, jika telah dimasukkan maka gambar akan berhasil tersimpan.




**Gambar 4.46** *Flowgraph* enkripsi gabungan

Berdasarkan gambar 4.45 maka dapat dilakukan proses perhitungan sebagai berikut :

Menghitung *Cyclomatic Complexcity* V(G) dari *edge* dan *node*

Dengan rumus V(G) = E – N + 2

Dengan *edge* = 29

Dengan *node* = 24

Dengan predikat *node* (P) = 5

Penyelesaian :

V(G) = E – N + 2

= 29 – 24 + 2

= 7

Predikat *node* (P) = P + 1

= 5 + 1

= 6

Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 6

*Independent path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 2 = 1 – 2 – 3 – 4 – 5 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 4 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 5 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 14 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 6 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

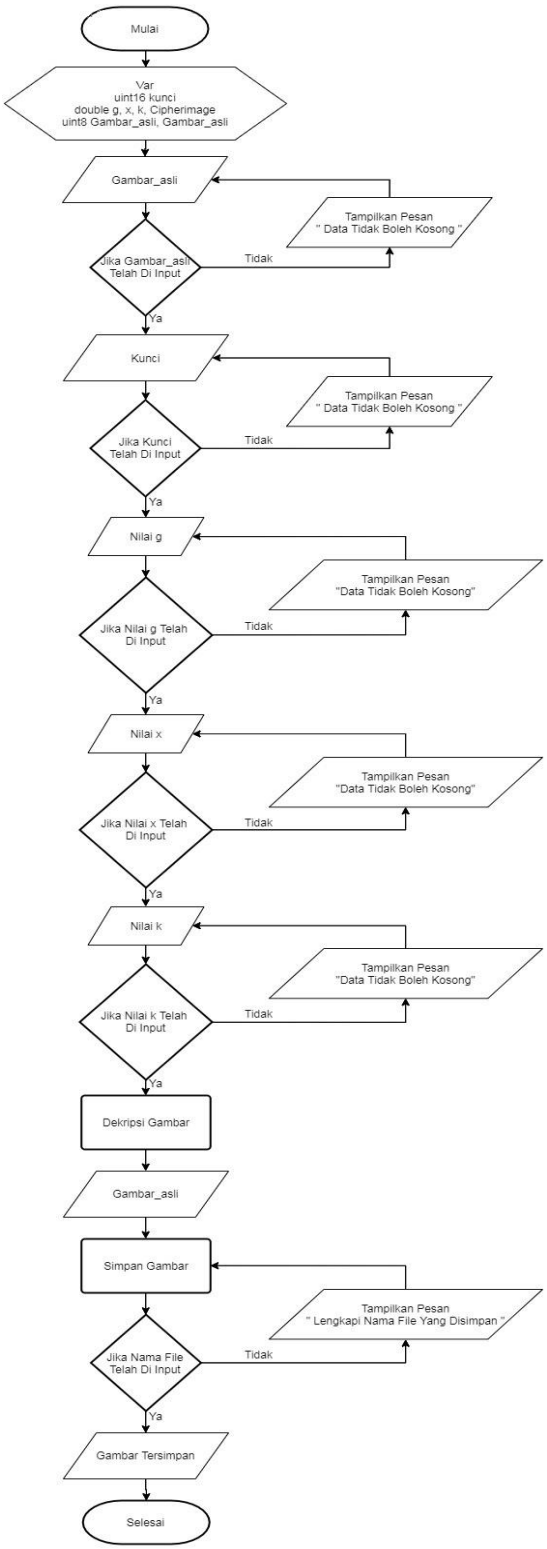
*Path* 7 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 22 – 20 – 21 – 23 – 24

Grafik matriks pada *flowgraph* diatas adalah :

**Tabel 4.30** Grafik Matriks Enkripsi Gabungan

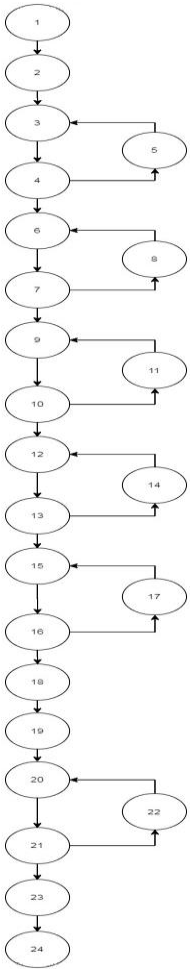
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **E-1** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 1 | | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 2 | | | 1 | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 3 | | | | 1 | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 4 | | | | | 1 | 1 | | | | | | | | | | | | | | | | | | | 2-1=1 |
| 5 | | | 1 | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 6 | | | | | | | 1 | | | | | | | | | | | | | | | | | | 1-1=0 |
| 7 | | | | | | | | 1 | 1 | | | | | | | | | | | | | | | | 2-1=1 |
| 8 | | | | | | 1 | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 9 | | | | | | | | | | 1 | | | | | | | | | | | | | | | 1-1=0 |
| 10 | | | | | | | | | | | 1 | 1 | | | | | | | | | | | | | 2-1=1 |
| 11 | | | | | | | | | 1 | | | | | | | | | | | | | | | | 1-1=0 |
| 12 | | | | | | | | | | | | | 1 | | | | | | | | | | | | 1-1=0 |
| 13 | | | | | | | | | | | | | | 1 | 1 | | | | | | | | | | 2-1=1 |
| 14 | | | | | | | | | | | | 1 | | | | | | | | | | | | | 1-1=0 |
| 15 | | | | | | | | | | | | | | | | 1 | | | | | | | | | 1-1=0 |
| 16 | | | | | | | | | | | | | | | | | 1 | 1 | | | | | | | 2-1=1 |
| 17 | | | | | | | | | | | | | | | 1 | | | | | | | | | | 1-1=0 |
| 18 | | | | | | | | | | | | | | | | | | | 1 | | | | | | 1-1=0 |
| 19 | | | | | | | | | | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 20 | | | | | | | | | | | | | | | | | | | | | 1 | | | | 1-1=0 |
| 21 | | | | | | | | | | | | | | | | | | | | | | 1 | 1 | | 2-1=1 |
| 22 | | | | | | | | | | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 23 | | | | | | | | | | | | | | | | | | | | | | | | 1 | 1-1=0 |
| 24 | | | | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| SUM E+1 | | | | | | | | | | | | | | | | | | | | | | | | | 6+1=7 |

f) *Flowchart* dan *flowgraph* dekripsi Elgamal




**Gambar 4.47** *Flowchart* dekripsi gabungan

Berdasarkan gambar diatas proses dekripsi gabungan dimulai dengan sistem yang akan melakukan inisiasi kemudian proses berikutnya adalah *user* memasukkan cipherimage dan kunci RC4 serta nilai g, x, k, namun apabila gambar, kunci dan nilai-nilai tersebut belum dimasukkan maka akan tampil pesan seperti gambar diatas. Kemudian proses berikutnya adalah proses dekripsi gambar dan akan menghasilkan gambar asli. Setelah itu langkah berikutnya adalah menyimpan gambar yang telah terenkripsi, namun apabila nama *file* belum dimasukkan maka akan tampil pesan seperti gambar diatas, jika telah dimasukkan maka gambar akan berhasil tersimpan.




**Gambar 4.48** *Flowgraph* dekripsi gabungan

Berdasarkan gambar 4.47 maka dapat dilakukan proses perhitungan sebagai berikut :

Menghitung *Cyclomatic Complexcity* V(G) dari *edge* dan *node*

Dengan rumus V(G) = E – N + 2

Dengan *edge* = 29

Dengan *node* = 24

Dengan predikat *node* (P) = 5

Penyelesaian :

V(G) = E – N + 2

= 29 – 24 + 2

= 7

Predikat *node* (P) = P + 1

= 5 + 1

= 6

Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* diatas memiliki *Region* = 6

*Independent path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 2 = 1 – 2 – 3 – 4 – 5 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 4 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 5 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 14 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 6 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 17 – 15 – 16 – 18 – 19 – 20 – 21 – 23 – 24

*Path* 7 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 22 – 20 – 21 – 23 – 24

Grafik matriks pada *flowgraph* diatas adalah :

**Tabel 4.31** Grafik Matriks Dekripsi Gabungan

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **E-1** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 1 | | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 2 | | | 1 | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 3 | | | | 1 | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 4 | | | | | 1 | 1 | | | | | | | | | | | | | | | | | | | 2-1=1 |
| 5 | | | 1 | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 6 | | | | | | | 1 | | | | | | | | | | | | | | | | | | 1-1=0 |
| 7 | | | | | | | | 1 | 1 | | | | | | | | | | | | | | | | 2-1=1 |
| 8 | | | | | | 1 | | | | | | | | | | | | | | | | | | | 1-1=0 |
| 9 | | | | | | | | | | 1 | | | | | | | | | | | | | | | 1-1=0 |
| 10 | | | | | | | | | | | 1 | 1 | | | | | | | | | | | | | 2-1=1 |
| 11 | | | | | | | | | 1 | | | | | | | | | | | | | | | | 1-1=0 |
| 12 | | | | | | | | | | | | | 1 | | | | | | | | | | | | 1-1=0 |
| 13 | | | | | | | | | | | | | | 1 | 1 | | | | | | | | | | 2-1=1 |
| 14 | | | | | | | | | | | | 1 | | | | | | | | | | | | | 1-1=0 |
| 15 | | | | | | | | | | | | | | | | 1 | | | | | | | | | 1-1=0 |
| 16 | | | | | | | | | | | | | | | | | 1 | 1 | | | | | | | 2-1=1 |
| 17 | | | | | | | | | | | | | | | 1 | | | | | | | | | | 1-1=0 |
| 18 | | | | | | | | | | | | | | | | | | | 1 | | | | | | 1-1=0 |
| 19 | | | | | | | | | | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 20 | | | | | | | | | | | | | | | | | | | | | 1 | | | | 1-1=0 |
| 21 | | | | | | | | | | | | | | | | | | | | | | 1 | 1 | | 2-1=1 |
| 22 | | | | | | | | | | | | | | | | | | | | 1 | | | | | 1-1=0 |
| 23 | | | | | | | | | | | | | | | | | | | | | | | | 1 | 1-1=0 |
| 24 | | | | | | | | | | | | | | | | | | | | | | | | | 1-1=0 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | E-1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SUM E+1 | | | | | | | | | | | | | | | | | | | | | | | | | 6+1=7 |

**Tabel 4.32** Hasil Pengujian

| NO | *Flowchart* | *Independent Path* | *Region* | *Cyclomatic Complexity* |
|---|---|---|---|---|
| 1 | Enkripsi RC4 | 4 | 3 | 4 |
| 2 | Dekripsi RC4 | 4 | 3 | 4 |
| 3 | Enkripsi Elgamal | 6 | 5 | 6 |
| 4 | Dekripsi Elgamal | 6 | 5 | 6 |
| 5 | Enkripsi Gabungan | 7 | 6 | 7 |
| 6 | Dekripsi Gabungan | 7 | 6 | 7 |

# BAB V

# KESIMPULAN DAN SARAN

## A. Kesimpulan

Kesimpulan yang dapat diambil setelah malakukan penelitian dan hasil pengujian terhadap aplikasi, maka dapat ditarik kesimpulan sebagai berikut.

1. Implementasi algoritma RC4 pada enkripsi gambar yang berupa logo perusahaan cukup baik. Hasil enkripsi dan dekripsi tidak mengalami perubahan baik itu pada ukuran maupun kualitas gambar dalam hal ini berupa *noise*.
2. Implementasi algoritma Elgamal pada enkripsi gambar yang berupa logo perusahaan kurang cocok dikarenakan pada saat proses enkripsi terdapat beberapa nilai yang melebihi nilai maksimal pada tiap nilai rgb, sehingga nilai yang melebihi tersebut dibulatkan ke nilai terdekat maksimal yaitu 255. Sehingga pada saat didekripsi terdapat beberapa nilai yang tidak sesuai dengan nilai awal yang menghasilkan *noise* pada gambar setelah didekripsi.
3. Gabungan algoritma RC4 dan Elgamal dalam enkripsi gambar yang berupa logo perusahaan sangat baik. Hasil enkripsi dari gabungan kedua algoritma tersebut menghasilkan gambar yang teracak dan sulit untuk dipecahkan. Akan tetapi pada hasil dari dekripsi terdapat sedikit

*noise* namun tidak begitu signifikan, hal ini dikarenakan pada proses enkripsi Elgamal yang nilainya melibihi nilai maksimal rgb.

## B. Saran

Pada penelitian ini penulis menyadari bahwa masih ada beberapa kekurangan yang sangat di perlukan perbaikan dan pengembangan di penelitian selanjutnya. Oleh karena itu, penulis memiliki beberapa saran untuk pengembangan selanjutnya, sebagai berikut:

1. Untuk penelitian selanjutnya diharapkan gambar yang dienkripsi tidak hanya untuk beberapa ekstensi akan tetapi bisa untuk semua eksistensi gambar. Selain itu diharapkan juga gambar yang dienkripsi bukan hanya gambar 2 dimensi akan tetapi sudah dapat mengenkripsi yang 3 dimensi.
2. Untuk penelitian selanjutnya diharapkan untuk menggabungkan algoritma yang memiliki tingkat keamanan dan kesulitan yang tinggi untuk enkripsi gambar.
3. Untuk penelitian selanjutnya diharapkan tidak menggunakan algoritma elgamal pada proses enkripsi dekripsi gambar 2 dimensi.

# DAFTAR PUSTAKA


Andono, P.N, Sutojo, T. & Muljono. (2018). *Pengolahan Citra Digital*. Semarang: Penerbit Andi.

Chumairoh, D. (2014). *Implementasi Algoritma RC4 Pada Aplikasi Smart Card Layanan Transaksi Peminjaman Buku Perpustakaan*. Malang: Universitas Islam Negeri Maulana Malik Ibrahim.

Fauzan, A. (2013). *Kriptografi Visual Dengan Memanfaatkan Algoritma Elgamal Untuk Citra Berwarna*. Makalah disajikan dalam web resmi Institut Teknologi Bandung di Bandung, Bandung 19 Mei 2013.

Kadir, A. (2019). *Dasar Pemrograman Matlab.* Yogyakarta: Penerbit Andi.

Munir, R (2019). *Krigtografi Edisi Kedua.* Bandung: Informatika.

Munir, R. (2018). *Algoritma Elgamal*. Diperoleh dari: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2017-2018/Algoritma-ElGamal-(2018).pdf. (Diakses 6 Oktober 2020).

Nisa, L. & Indriyani, T. & Ruswiansari, M. (2020). *Aplikasi Enkripsi Citra Dan Teks Menggunakan Algoritma Diffie-Hellman Dan Elgamal*. 1(1): 8-17.

Prasetyo, E. (2011). *Pengolahan Citra Digital Dan Aplikasinya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.

Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: CV. Andi Offset.

Sekarsari, G. A. (2015). *Analisa Algoritma Kriptografi RC4 Pada Enkripsi Citra Digital*. 14(4): 250-254.

Setyaningsih, E. (2015). *Kriptografi Dan Implementasinya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.

Sumihar, Y. P. (2015). *Analisa Kemampuan Algoritma Elgamal Untuk Kriptografi Citra*. 7(1): 27-44.

Tamam, T. M. & Dwiono, W. & Hartanto, T. (2010). *Penerapan Algoritma Kriptograi Elgamal Untuk Pengaman File Citra*. 4(1): 8-11.

Vahlevi, M. R. (2018). *Implementasi Algoritma RC4 Dan Rabin-Williams Dalam Skema Hybrid Cryptosystem Dalam Pengamanan File Gambar Pada Apliaksi Instant Messaging Berbasis Android*. Medan: Universitas Sumatra Utara.

Zebua, T., Ndruru, E. (2017). *Pengamanan Citra Digital Berdasarkan Modifikasi Algoritma RC4*. 4(4): 275-282.