HYBRID CRYPTOSYSTEM MENGGUNAKAN *RIVEST CODE 4 (RC4)* DAN *H-RABIN* PADA PENGAMANAN GAMBAR

SKRIPSI

VINA ADLINA RAMAYANI

201401071

PROGRAM STUDI S-1 ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2024

HYBRID CRYPTOSYSTEM MENGGUNAKAN *RIVEST CODE 4 (RC4)* DAN *H-RABIN* PADA PENGAMANAN GAMBAR

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana Ilmu Komputer

VINA ADLINA RAMAYANI
201401071

PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024

PERSETUJUAN

Judul	: *HYBRID CRYPTOSYSTEM* MENGGUNAKAN *RIVEST CODE 4 (RC4)* DAN *H-RABIN* PADA PENGAMANAN GAMBAR
Kategori	: SKRIPSI
Nama	: VINA ADLINA RAMAYANI
Nomor Induk Mahasiswa	: 201401071
Program Studi	: SARJANA (S-1) ILMU KOMPUTER
Fakultas	: ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA
Komisi Pembimbing	:

Medan, 11 Juni 2024

Pembimbing 2

Anandhini Medianty Nababan
S. Kom., M. T.
NIP. 199304132021022001

Pembimbing 1

Dian Rachmawati S.Si., M.Kom.
NIP. 198307232009122004

Diketahui/Disetujui Oleh
Program Studi S-1 Ilmu Komputer
Ketua,

Dr. Amalia ST., M.T.
NIP. 197812212014042001

PERNYATAAN

HYBRID CRYPTOSYSTEM MENGGUNAKAN *RIVEST CODE 4 (RC4)* DAN *H-RABIN* PADA PENGAMANAN GAMBAR

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 11 Juni 2024

Vina Adlina Ramayani
201401071

PENGHARGAAN

Bismillahirrahmanirrahim, segala puji syukur dipanjatkan kepada Allah *Subhanahu Wa Ta'ala* atas segala limpahan rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini sebagai syarat untuk mendapatkan gelar Sarjana Komputer di Program Studi S-1 Ilmu Komputer, Universitas Sumatera Utara. Tidak lupa shalawat serta salam tetap tercurahkan kepada Rasulullah *Shalallaahu 'Alayhi Wasallam* yang telah mengeluarkan umat manusia dari kegelapan menuju zaman terang benderang saat ini.

Penuli sadar penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, dan bimbingan dari banyak pihak. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada:

1. Bapak Prof. Dr. Muryanto Amin S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara dan Dosen Pembimbing Akademik yang telah memberi banyak dukungan dan motivasi kepada penulis.
3. Ibu Dr. Amalia, S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Dian Rachmawati S.Si., M.Kom. selaku Dosen Pembimbing I yang telah memberi banyak bimbingan, masukan, bantuan serta meluangkan waktu yang berharga untuk penulis selama penyusunan skripsi ini.
5. Ibu Anandhini Medianty Nababan S.Kom., M.T. selaku Dosen Pembimbing II yang telah memberikan bimbingan dan masukan serta memberikan semangat kepada penulis dalam penyusunan skripsi ini.
6. Ibu Dr. Ir. Elviawaty Muisa Zamzami S.T., M.T., M.M., IPU. selaku Dosen Pembanding I yang telah memberi saran dan kritik yang membangun kepada penulis terhadap penyusunan skripsi ini.
7. Bapak Dr. Mohammad Andri Budiman S.T., M.Comp.Sc., M.E.M.. selaku Dosen Pembanding II yang telah memberi saran dan kritik yang membangun kepada penulis terhadap penyusunan skripsi ini.

8. Seluruh bapak dan ibu dosen Fasilkom-TI USU, khususnya dosen Program Studi S-1 Ilmu Komputer yang telah mendidik dan memberi wawasan serta moral yang berharga, baik di bangku perkuliahan maupun setelah lulus.
9. Ibunda tercinta, ibu Zuraidah Nasution yang selalu mendoakan, memberikan semangat, memberikan dukungan moril dan materil serta kasih sayang yang begitu besar kapada penulis. Dukungan ibunda menjadi salah satu faktor pendorong penulis selalu semangat dalam penyusunan skripsi.
10. Kakak Nurus Syahri Nasution dan Adik Vinando Syahputra yang selalu memberikan semangat dan mendoakan penulis.
11. Keluarga besar penulis yang selelu memberikan dukungan dan doa untuk penulis.
12. Seluruh pegawai dan staf Fasilkom-TI USU yang telah memberi bantuan selama masa perkuliahan.
13. Sahabat dari semester 1 hingga detik ini, Azzahra Mumtaza (075) dan Devi Megarita Br Sibarani (136) yang memberikan banyak dorongan dan motivasi, terima kasih untuk seluruh tawa, suka dan duka, serta kisah kasih di masa perkuliahan.
14. Sahabat menjalani dunia per-S.Kom-an, ‘Ciwi-ciwi Kom B’, dan seluruh teman stambuk 2020 yang ramai, meriah, dan akan selalu terkenang, terima kasih untuk seluruh pembelajaran barunya, baik dari aspek akademis maupun bekal dalam menjalani kehidupan dunia dan akhirat.
15. Sahabat berkembang sejak putih abu-abu, ‘Sadukat’ yang senantiasa memberi memori hangat, semangat yang membara, dan terima kasih untuk pelukan di saat suka dan duka.

Dan seluruh pihak yang telah memberi dukungan serta doa baik yang tidak dapat penulis sebutkan satu per-satu. Semoga Allah *Subhanahu Wa Ta‘ala* senantiasa melimpahkan keberkahan serta kebaikan atas semua dukungan yang telah diberikan kepada penulis dan hasil penelitian ini dapat memberi manfaat maupun inspirasi untuk kedepannya.

Medan, 11 Juni 2024

Penulis,

Vina Adlina Ramayani

ABSTRAK

Penggunaan satu buah algoritma kriptografi masih sangat rentan keamanannya sehingga diperlukan kombinasi dua buah algoritma kriptografi untuk menghasilkan *ciphertext* yang sulit untuk dimengerti oleh pihak yang tidak berwenang dalam proses pengiriman pesan. Selain itu kemajuan teknologi memberikan kemudahan dalam melakukan pertukaran informasi dengan hanya ditransmisikan melalui internet sehingga menyebabkan banyak ancaman terhadap pesan tersebut, seperti ancaman dimodifikasi, pemalsuan, sampai penyalahgunaan informasi tersebut. *Hybrid Cryptosystem* adalah gabungan dari algoritma simetris dan asimetris yang memanfaatkan kecepatan pemrosesan data dari algoritma simetris serta kemudahan transfer kunci dari algoritma asimetris. Algoritma *Rivest Code 4* adalah salah satu algoritma kunci simetris, dimana proses enkripsi dan dekripsi dengan kunci yang sama yang memiliki mode kerja *stream cipher*. Algoritma *H-Rabin* adalah salah satu algoritma asimetri yang memanfaatkan bilangan prima random untuk membangkitkan kunci publik. File yang digunakan adalah *file* gambar berformat *.png*. Pada proses enkripsi dengan algoritma RC4 berhasil dalam pengamanan file gambar, sehingga file gambar hasil enkripsi atau *ciphertext* yang dihasilkan tidak dapat terbaca. Sedang proses enkripsi kunci dengan algoritma *H-Rabin* berhasil menghasilkan *cipherkey* sehingga kunci asli tidak diketahui oleh pihak lain. Dari hasil yang diproleh disimpulkan bahwa melakukan proses kriptografi dengan menggunakan metode *Hybrid Cryptosystem* yang mengkombinasikan algoritma *Rivest Code 4* dan algoritma *H-Rabin* berhasil mengembalikan keutuhan pesan seperti semula dan menjaga keamanan file gambar dari pihak yang tidak berwenang.

Kata Kunci: *Gambar, Kriptografi, Hybrid Cryptosystem, Rivest Code 4, H-Rabin*

HYBRID CRYPTOGRAPHY SYSTEM USING RIVEST CODE 4 AND H-RABIN FOR IMAGE SECURITY

ABSTRACT

The use of one cryptographic algorithm is still very vulnerable to security so a combination of two cryptographic algorithms is needed to produce a ciphertext that is difficult for unauthorized parties to understand in the process of sending messages. In addition, technological advances make it easy to exchange information by only being transmitted via the internet, causing many threats to the message, such as the threat of modification, forgery, and misuse of the information. Hybrid Cryptosystem is one of the combinations of symmetry and asymmetry algorithms by utilizing the advantages of data processing speed by symmetric algorithms and ease of key transfer using asymmetric algorithms. The Rivest Code 4 algorithm is one of the symmetric key algorithms, where the encryption and decryption process with the same key has a stream cipher working mode. The H-Rabin algorithm is a form of asymmetric encryption that utilizes random prime numbers to generate public keys. The file used is a .png format image file. The encryption process with the RC4 algorithm is successful in securing image files, so that the encrypted image file or the resulting ciphertext cannot be read. While the key encryption process with the H-Rabin algorithm successfully produces a cipherkey so that the original key is not known by other parties. Based on the results acquired, it can be concluded that carrying out the cryptographic process using the Hybrid Cryptosystem method which combines the Rivest Code 4 algorithm and the H-Rabin algorithm succeeds in restoring the integrity of the message as before and maintaining the security of image files from unauthorized parties.

Keywords: *Picture, Cryptography, Hybrid Cryptosystem, Rivest Code 4, H-Rabin.*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Aspek kerahasiaan dan keamanan data telah menjadi poin kritis dalam ekosistem sistem informasi modern. Dengan adanya kemunculan teknologi internet dan multimedia, semakin banyak upaya dilakukan untuk menjaga, mengamankan, dan menyembunyikan informasi yang tersimpan dalam berbagai jenis file digital, baik itu berupa teks atau gambar. Perkembangan teknologi yang pesat memberikan dorongan yang signifikan bagi semua lapisan masyarakat untuk melakukan pertukaran informasi, termasuk data gambar. Saat ini, data gambar tidak lagi harus disimpan dalam media fisik seperti *DVD*, *CD*, atau *hard disk*. Penggunaan internet memungkinkan untuk mentransmisikan data gambar secara langsung melalui saluran-saluran publik yang tersedia. Hal ini menciptakan lahan baru untuk berbagi informasi secara efisien, namun juga menimbulkan tantangan baru terkait dengan keamanan dan privasi data.

Data gambar berfungsi sebagai representasi visual dari berbagai informasi atau data, memudahkan pengguna multimedia untuk memahami atau mengingatnya. Namun, semakin banyak penggunaan data gambar, semakin tinggi risiko penyalahgunaannya. Informasi yang disampaikan melalui gambar sangat rentan terhadap penyerangan seperti pemalsuan, modifikasi, serta penyalahgunaan gambar untuk hal negatif. Oleh karena itu, perlindungan terhadap integritas dan keaslian data gambar menjadi sangat penting dalam era digital ini. Untuk melindungi data gambar tersebut diperlukan suatu metode pengamanan data untuk mengamankan data gambar dari penggunaan yang tidak sah atau jatuh tidak ke tangan penerima seharusnya. Dalam proses mengamankan gambar diperlukan algoritma kriptografi untuk mengenkripsi dan dekripsi gambar. Sehingga gambar tidak akan terbaca karena gambar asli telah diubah menjadi gambar yang tidak terbaca. Dan melakukan poses dekripsi untuk mengembalikan ke gambar asli.

Kriptografi adalah bidang studi yang menggunakan prinsip-prinsip matematika dalam pengamanan dan penyandi-an data melalui proses enkripsi dan

dekripsi. Dalam teknik ini, data diubah menjadi serangkaian kode tertentu, menghasilkan urutan huruf acak yang tidak dapat dibaca tanpa proses dekripsi yang sesuai (Hamsyar & Basri, 2022). Hal ini dilakukan untuk menentukan hanya penerima tertentu yang bisa memproses informasi yang dikirim. Kriptografi simetris dan asimetris adalah dua jenis kriptografi.

Kriptografi simetris menawarkan kecepatan pada saat mengenkripsi dan mendekripsi pesan karena menggunakan satu kunci untuk kedua proses tersebut. Namun, kelemahannya terletak pada risiko proses penyaluran kunci yang tidak aman, karena kunci yang serupa dipakai untuk mengenkripsi dan mendekripsi pesan. Sementara itu, Keunggulan kriptografi asimetris terletak pada keamanan pengiriman kunci yang dihasilkan dari proses enkripsi dan dekripsi menggunakan dua kunci yang berbeda. Namun, kriptografi asimetris sering kali lebih lambat dibandingkan dengan kriptografi simetris. Oleh karena itu, diterapkan suatu metode yaitu menggabungkan kedua konsep ini untuk menciptakan sistem keamanan yang lebih baik, memanfaatkan kecepatan kriptografi kunci simetris dan keamanan distribusi kunci dari kriptografi kunci asimetris (Ramadani, Diana, & Sauda, 2020).

Metodologi *hybrid cryptosystem* adalah pendekatan untuk memperkuat keamanan suatu file atau informasi dengan menggabungkan kriptografi simetris dan asimetris. Dengan memadukan kedua algoritma ini, diciptakanlah sebuah teknik yang lebih kuat dan mampu menutupi kelemahan yang dimiliki masing-masing algoritma tersebut. Dalam penelitian ini, akan diterapkan suatu pendekatan yang menggunakan algoritma kriptografi *Rivest Code 4* sebagai kunci simetris dan algoritma *H-Rabin* sebagai kunci asimetris. Algoritma *Rivest Code 4* digunakan untuk melakukan proses enkripsi dan dekripsi data gambar, sedangkan algoritma *H-Rabin* digunakan untuk enkripsi dan dekripsi kunci *Rivest Code 4.*

1.2 Rumusan Masalah

Pada latar belakang yang telah dijelaskan, maka penulis menyimpulkan rumusan masalah dalam penelitian ini adalah diperlukannya sistem pengamanan data berupa gambar untuk menghindari dari penyalahgunaan data gambar tersebut dari orang yang tidak berwenang. Dan juga diperlukan algoritma yang efisien dikombinasikan dengan algoritma H-Rabin yang memiliki masalah pada efisiensi dalam proses

enkripsi data besar, karena ketika kunci yang dibangkitkan panjang, maka kunci hasil enkripsi akan jauh semakin panjang.

1.3 Batasan Masalah

Penelitian ini membatasi beberapa ruang lingkup sebagai berikut:

1. Penelitian ini hanya fokus pada teknik penyandian dan pendeskripsian *file* gambar menggunakan algoritma *Rivest Code 4* dan algoritma *H-Rabin.*
2. Jenis *file* yang digunakan berekstensi *Portable Network Graphics* (.*png).
3. Tidak membahas mengenai serangan-serangan yang terjadi terhadap kriptografi.
4. Kunci yang digunakan adalah berupa angka, bukan kombinasi huruf, angka dan karakter unik.
5. Bahasa pemrograman yang diterapkan menggunakan bahasa C#.

1.4 Tujuan Penelitian

Penelitian ini memiliki tujuan membuat aplikasi agar dapat memberikan keamanan dalam proses pengamanan gambar dengan mengkombinasikan algoritma kunci simetri yaitu algoritma Rivest Code 4 dan kunci asimetri yaitu algoritma H-Rabin.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini diharapkan dapat meningkatkan keamanan serta memproteksi kerahasian data gambar agar terhindar dari orang yang tidak berwenang terhadap gambar tersebut dengan mengkombinasilan dua algoritma kriptografi yaitu simetris dan asimetris.

1.6 Metodologi Penelitian

Beberapa metode yang digunakan untuk penelitian ini yaitu sebagai berikut:

1. Studi Pustaka

   Tahapan ini dilakukan dengan mencari referensi dari berbagai sumber yang dapat dipercaya, termasuk buku, jurnal, *e-book*, artikel ilmiah, makalah, skripsi, serta situs internet terkait topik-topik seperti kriptografi, *Hybrid Cryptosystem*, algoritma *Rivest Code 4* (RC4), dan algoritma *H-Rabin*.

2. Analisis dan Perancangan Sistem

   Pada tahap ini, dilakukan analisis terhadap kebutuhan penelitian yang didapatkan dari penerapan algoritma *Rivest Code 4* dan algoritma *H-Rabin*. Analisis ini mencakup perancangan diagram alur (*flowchart*), *diagram Ishikawa* (*fishbone*), *use-case diagram*, *activity diagram*, *sequence diagram*, serta *user interface*.

3. Implementasi Sistem

   Pada tahap ini, rancangan yang telah disusun akan diterapkan dalam sebuah sistem menggunakan bahasa C#.

4. Pengujian Sistem

   Tahapan ini dilakukan dengan menguji sistem yang telah dibuat untuk menganalisis *Hybrid Cryptosystem* denngan menggunakan algoritma *Rivest Code 4* dan algoritma *H-Rabin*.

5. Dokumentasi Sistem

   Pada tahapan ini, penulis menyusun laporan yang mencakup semua tahapan dari analisis hingga pengujian dalam format penulisan penelitian.

1.7 Penelitian Relevan

Adapun penelitian terdahulu yang relevan dengan penelitian yang akan dilakukan dalam penelitian ini antara lain:

1. Berdasarkan penelitian (Hutapea, 2021) File gambar sangat rentan terhadap penyerangan seperti pemalsuan, modifikasi, serta penyalahgunaan gambar untuk hal negatif. Maka dari itu, dibutuhkan metode atau algotitma kriptografi *Rabin* untuk melindungi *file* gambar, mempertahankan keaslian dan kerahasiaannya. Proses pengamanan *file* gambar dilakukan dengan melakukan enkripsi terhadap gambar, sehingga menghasilkan *cipher image*. Dari hasil enkripsi tersebut akan memberikan dampak pembesaran ukuran *file* sehingga diperlukan algortima dekompresi untuk memperkecil ukuran *file*.
2. Berdasarkan penelitian (Maulana & R. Mahdalena, 2021) Perlindungan data pribadi siswa SMA Swasta Jaya Krama Beringin menggunakan algoritma RC4. Penelitian ini menemukan bahwa algoritma RC4 dapat digunakan untuk mengimplementasikan sistem enkripsi untuk melindungi informasi pribadi siswa karena dapat mengenkripsi data yang diinputkan sehingga dapat

digunakan untuk meningkatkan keamanan data. Selain itu, sistem yang dibuat dapat didekripsi dengan kunci yang digunakan untuk mengenkripsi *ciphertext*.

3. Berdasarkan penelitian (Rachmawati & Mohammad, 2018) Enkripsi yang dilakukan dengan kunci publik biasanya akan mengenkripsi kunci rahasia dari algoritma enkripsi lain. Pada penelitian ini algoritma kunci publik tidak digunakan untuk mengenkripsi kunci rahasia dari algoritma enkripsi lain, melainkan digunakan langsung untuk mengenkripsi *plaintext*. Adapun algoritma kunci publik yang digunakan adalah algoritma H-Rabin sehingga menghasilkan *ciphertext* yang lebih panjang, untuk itu dibutuhkan satu algoritma kompresi yang dapat digunakan untuk mengkompresi *ciphertext* yang dihasilkan dari algoritma H-Rabin. Hasil dari penelitian ini menunjukkan bahwa ukuran dari *ciphertext* dapat dikurangi dengan rasio kompresi dari 1,94 ke 1 dan penghematan ruang sekitar 48%. Hasil penelitian ini juga menunjukkan prospek dari penggunaan algoritma enkripsi kunci publik saja pada enkripsi pesan yang besar tanpa harus menggabungkan algoritma tersebut dengan algoritma simetris dengan mengompres *ciphertext* dengan menggunakan algoritma kompresi sebelum mentransmisinya.
4. Berdasarkan penelitian (Mohammed & Lahieb, 2020) pengamanan gambar berdasarkan *stream cipher* RC4 dan model pembangkitan kunci baru menggunakan henon dan metode *chaotic maps*. Skema penelitian ini terdiri dari *confusion stage* dan *diffusion stage*. Dimana algoritma *chaotic maps* digunakan pada tahap *confusion* untuk menghasilkan kunci dari algoritma RC4 yang nantinya digunakan pada tahap *diffusion*. Hasil penelitian menunjukkan bahwa skema yang digunakan dapat menahan serangan *brute-force*, selain itu perbandingan enkripsi RC4 dengan enkripsi RC4 yang didukung algoritma *chaotic maps* meningkatkan keamanan algoritma enkripsi.

1.8 Sistematika Penulisan

Tahapan penulisan skripsi yang digunakan untuk penelitian ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini mencakup penjelasan mengenai latar belakang pemilihan judul, rumusan, batasan masalah, tujuan, manfaat, dan metodologi penelitian, penelitian relevan, dan sistematika penulisan skripsi.

BAB 2 LANDASAN TEORI

Bab ini menjelaskan tinjauan teoritis yang berhubungan dengan penelitian, seperti pendahuluan kriptografi, *Hybrid CryptoSystem*, Algoritma *Rivest Code 4 (RC4)*, Algoritma *H-Rabin*, serta contoh perhitungan masing-masing algoritma.

BAB 3 ANALISIS DAN PERANCANGAN

Bab ini bertujuan untuk menjelaskan analisis terhadap algoritma yang akan digunakan, serta melakukan perancangan diagram yang diperlukan, seperti diagram alir (*flowchart*), untuk memvisualisasikan proses algoritma dengan lebih jelas dan sistematis.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan tentang penerapan algoritma yang berasal dari hasil analisis serta perancangan yang sudah dilakukan sebelumnya.

BAB 5 KESIMPULAN DAN SARAN

Bab ini merangkum kesimpulan berdasarkan uraian yang terdapat dalam beberapa bab sebelumnya, dengan tujuan memberikan saran yang berguna sebagai masukan untuk menyempurnakan pengembangan sistem yang telah dibangun.

BAB 2
LANDASAN TEORI

2.1 Gambar

2.1.1 Definisi Gambar

Salah satu konsep penting dari gambar adalah gambar digital, yang menjadi elemen penting dalam konteks multimedia dan memiliki peran krusial dalam menyampaikan informasi (Munir R., 2004). Gambar digital memiliki kekayaan informasi, yang membedakannya dari teks. Data gambar Memvisualisasikan bermacam informasi, sehingga gambar memilki peran penting untuk menyampaikan pesan, ekspresi, dan komunikasi visual (Hutapea, 2021).

Terdapat banyak format gambar yang umum digunakan contohnya seperti, format JPEG, PNG, HEIF, RAW, SVG, dan lainnya. Dalam penelitian ini berfokus dalam format gambar PNG.

Gambar 2.1 Contoh Gambar

2.1.2 Format Gambar PNG

PNG adalah singkatan dari "*Portable Network Graphics"* merupakan format gambar yang umum dipergunakan untuk menyimpan gambar dengan kualitas tingggi tanpa mengopresi secara berlebihan, sehingga menjaga ketajaman dan detail gambar. PNG juga lebih jelas terlihat dibandingkan JPEG walaupun ukuran file PNG lebih besar dari format JPEG untuk gambar yang

kompleks. Keunggulann format PNG dalam hal kualitas dan dukungan untuk transparansi membuatnya menjadi pilihan yang popular dan banyak konteks untuk desain dan web

2.1.3 Format Gambar SVG

SVG atau *Scalable Vector Graphics* merupakan format file grafis berbasi vektor yang digunakan untuk menyimpan grafik berbasi vector dengan skalabilitas tinggi. Berbeda dengan format seperti PNG dan JPEG yang terdiri dari grid piksel, SVG menggunkan koordinat matematika untuk mendefinisikan bentu, garis, dan warna sehingga memnugkinkan untuk diubah ukuran nya tanpa kehilangan kualitas. Format ini sangaat cocok untuk grafik yang perlu ditampilkan seperti logo, ikon, dan diagram.

2.2 Kriptografi

Kriptografi memiliki akar kata dari Bahasa Yunani, yaitu kripto yang berarti rahasia, dan graphia yang berarti tulisan. Secara sederhana, kriptografi dapat diartikan sebagai ilmu dan seni dalam menjaga kerahasiaan pesan saat proses pengiriman melalui sender hingga pesan sampai diterima oleh receiver. Tujuannya yaitu untuk mengamankan pesan dari orang yang tidak berhak menerima pesan tersebut, yang dicapai melalui penggunaan algoritma kriptografi (Siahaan dan Mesran, 2020).

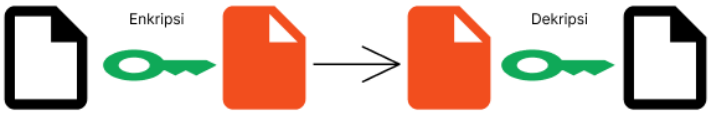
Terdapat dua komponen utama dalam kriptografi yaitu enkripsi dan dekripsi. Enkripsi adalah proses pengubahan *plaintext* atau data ke dalam bentuk yang tidak terbaca agara hanya penerima yang dituju yang dapat membacanya. Sementara dekripsi adalah proses mengembalikan data yang telah dienkripsi ke bentuk aslinya dengan menggunakan kunci yang tepat. Pesan yang telah dienkripsi disebut sebagai *ciphertext*, sedangkan setelah didekripsi, pesan tersebut kembali menjadi *plaintext* (pesan asli). Ada beberapa jenis kriptografi yang digunakan untuk melindungi informasi dalam berbagai konteks, yaitu simetris dan asimetris. Jenis riptografi simetris fokus pada keamanan komunikasi dengan menggunakan kunci rahasia yang sama antara pengirim dan

penerima, sedangkan kriptografi asimetris menggunakan kunci publik dan privat.

2.2.1 Kriptografi Simetris

Kriptografi simetris merupakan algoritma kriptografi di mana penyandian dan dekripsi pesan dilakukan dengan menggunakan kunci yang sama. Algoritma ini mengubah teks biasa, atau pesan, menjadi bentuk yang tidak dapat dibaca, atau *ciphertext*, dengan menggunakan kunci yang sama. Kunci ini kemudian digunakan kembali untuk mengembalikan pesan ke bentuk aslinya melalui proses dekripsi. Kriptografi kunci simetris memiliki dua mode, yaitu *block cipher* dan *stream cipher* (Suhandinata, 2019). Contoh algoritma kriptografi simetris adalah RC4.

Kelebihan dari kriptografi simetris yaitu, proses lebih cepat bila dibandingkan dengan kunci asimetris. Namun terdapat pula kekurangannya yaitu keamanan kunci yang digunakan untuk enkripsi dan dekripsi sangat rentan terhadap serangan sehingga kriptografi kunci simetris kurang aman digunakan untuk pengamanan pesan yang bersifat rahasia atau sensitif.



Gambar 2.2 Kunci Simetris

2.2.2 Kriptografi Asimetris

Kriptografi asimetris merupakan bentuk kriptografi dengan menggunakan sepasang kunci yang berbeda pada setiap proses yaitu, prose enkripsi menggunkan kunci publik hasil dibangkitkan dari penerima, sedangkan dekripsi dilakukan dengan kunci privat yang didapat dari hasil mengacak beberapa bilangan. Ini berarti enkripsi pesan hanya dapat dilakukan dengan

kunci publik dan dekripsi pesan hanya dapat dilakukan dengan kunci privat bukan kunci publik sehingga kerahasiaan kunci dapat dijaga(Putra & al, 2023).

Kriptografi asimetris memiliki beberapa keuntungan dibandingkan dengan kriptografi simetris, termasuk kemampuan untuk mengamankan proses pertukaran kunci tanpa perlu kehadiran fisik atau saluran komunikasi yang aman. Namun, kriptografi simetris memerlukan waktu yang lebih lama untuk mengenkripsi pesan, sehingga yang dienkripsi adalah kunci simetris daripada pesan asli. Selanjutnya pesan akan dikirim setelah dienkripsi dengan kunci simetris (Basri, 2016). Bentuk kriptografi kunci asimetris adalah algoritma H-Rabin.

Gambar 2.3 Kunci Asimetris

2.3 Hybrid Cryptosystem

Hybrid cryptosystem merupakan gabungan dari kriptografi simetris dan asimetris yang menggabungkan keuntungan dari kedua sistem tersebut untuk meningkatkan keamanan dan efisiensi dalam pertukaran informasi yang aman. Dalam *hybrid cryptosystem*, pesan atau data pertama kali dienkripsi menggunakan kriptografi simetris dengan sebuah kunci acak yang disebut kunci sesi. Proses enkripsi ini relatif cepat karena kriptografi simetris cenderung lebih efisien dalam hal waktu komputasi. Kemudian, lakukan enkripsi terhadap kunci sesi menggunakan kriptografi asimetris dengan *public key* dan pesan yang telah dienkripsi bersama dengan kunci sesi dikirimkan ke penerima. Penggabungan kedua metode ini mengakibatkan peningkatkan kecepatan tanpa mengurangi kenyamanan dan keamanan (Jamaludin & R, 2020).

Kunci publik dibangkitkan oleh penerima lalu pengirim pesan mengenkripsi kunci simetris menggunakan kunci publik yang diberikan oleh penerima, kemudian dikirim bersama pesan terenkripsi. Untuk melakukan dekripsi

ciphertext, penerima pesan mendekripsi kunci rahasia dengan kunci privat yang dimilikinya, lalu penerima mendekripsi *ciphertext* dengan kunci yang telah didekripsi tersebut.

2.4 Algoritma Rivest Code 4 (RC4)

RC4 merupakan algoritma enkripsi aliran dimana ia menghasilkan aliran bit yang terus menerus yang kemudian dilakukan operasi XOR dengan pesan asli untuk mendapatkan teks terenkripsi (*ciphertext*), atau dengan teks terenkripsi untuk mendapatkan pesan asli kembali. Algoritma ini memakai kunci yang serupa dalam melakukan proses kriptografi, juga RC4 terkenal karena kecepatannya dalam mengenkripsi data. Berikut ini adalah tiga tahap utama dalam menggunakan algortima RC4:

Key Scheduling Algorithm (KSA)

Proses KSA langkah yang dilakukan untuk menghasilkan kunci internal yang akan dipergunakan selama proses enkripsi dan dekripsi. Algoritma ini melibatkan pengacakan atau permutasi array S, yang terdiri dari 256 byte, berdasarkan kunci yang diberikan. Adapun proses *Key Scheduling Algorithm* pada RC4 yaitu:

1. Menginisialisasi Array S (S-Box) dengan pernyataan atau intruksi dibawah:

   For i = 0 to 255
   S[Box] = i

2. Selanjutnya adalah lakukan proses inialisasi S-Box untuk Array K (kunci) dengan pernyataan atau instruksi berikut:

   For i = 0 to 255

   K[i] = Kunci [i mod Panjang kunci]

3. Tukarkan posisi Array S berdasarkan Array K dengan cara mengganti isi array S[i] dengan S[j] serperti persamaan berikut:

   i = 0; j = 0
   for i = 0 to 255
   j = (j + S[i] + K[i]) mod panjang array

swap S[i] dan S[j]

***Pseudo Random Generation Algorithm* (PRGA)**

Proses ini digunakan untuk membangkitkan *keystream* menggunakan table pada array S-Box. Hasil *keystream* ini akan dilakukan operasi XOR dengan pesan asli. Adapun persamaannya adalah sebagai berikut:

i = (i + 1) mod 256

j = (j + S[i]) mod 256

swap S[i] dan S[j]

t = (S[i] + S[j]) mod 256

K = S[t]

Proses Enkripsi dan Dekripsi

Kedua tahap ini dilakukan dengan melakukan operasi XOR terhadap *byte* data dengan *keystream* untuk menghasilkan nilai *chipertext* (proses enkripsi). Sedangkan, untuk mendapatkan pesan asli, lakukan operasi XOR antara pesan terenkripsi dan kunci yang sama seperti pada tahap enkripsi

Contoh Perhitungan

Berikut adalah contoh perhitungan pada proses enkripsi RC4

Tahap *Key scheduling Algorithm* (KSA)

1. Inisalisasi array S sesuai panjang 2 *byte*

Indeks	0	1
S	0	1

2. Inisasisi array kunci (K[i]) sesuai panjang kunci

Indeks	0	1
K	2	3

3. Lakukan pertukaran posisi terhadap nilai di dalam array S[i] dengan S[j] seperti berikut:

Perulangan pertama, untuk nilai i = 0

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

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

j = (0 + 0 + 2) mod 2 = 0

Tukarkan isi array S[0] dengan S[0]

Indeks	0	1
S	0	1

Perulangan kedua, untuk nilai j = 0 dan i = 1

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

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

j = (0 + 1 + 3) mod 2 = 1

Tukarkan isi array S[1] dengan S[0]

Indeks	0	1
S	1	0

Tahap *Pseudo Random Generation Algorithm* (PRGA)

Untuk mendapatkan *ciphertext*, maka terlebih dahulu dilakukan pembangkitan kunci seperti berikut:

- perulangan pertama, i = 0 ; j = 0

  i = (0 + 1) mod 2 = 1 j = (j + S[1]) mod 2

  j = (j + S[1] mod 2

  j = (0 + 0) mod 2 = 0

  j = 0

- Melakukan penukaran isi array S[0] dengan S[1]

Index	0	1
S	1	0

t = (S[i] + S[j]) mod 2

t = (S[0] + S[1]) mod 2

t = (1 + 0) mod 2

t = 1

K = S[t] = 1

- Perulangan kedua, i = 1 ; j = 0

i = (1 + 1) mod 2 = 0 j = (j + S[0]) mod 2

j = (j + S[0]) mod 2

j = (0 + 1) mod 2

j = 1

- Melakukan Penukaran isi array S[0] dengan S[1]

Index	0	1
S	0	1

t = (S[i] + S[j]) mod 2

t = (S[0] + S[1]) mod 2

t = (0 + 1) mod 2

t = 1

K = S[t] = 1

Dari proses PRGA, maka hasil pembangkit kunci adalah 11. Lalu dikonversi ke biner.

Tabel 2.1 Konversi Keystream ke Biner

Keystream		
Desimal	1	1
Biner	000000001	00000001

Proses enkripsi dan dekripsi

Untuk melakukan enkripsi pada *plaintext* lakukan operasi XOR-kan terhadap kunci. Namun sebelum itu ubah terlebih dahulu *plaintext* ke biner.

Plaintext : NV

Kunci : 11

Tabel 2.2 Konversi *Plaintext* Ke Biner

Plaintext		
Simbol	N	V
Desimal	078	086
Biner	01001110	01010110

Tabel 2.3 Proses XOR *Keystream* dan *Plaintext*

K ⊕ P		
Keystream	00000001	00000001
Plaintext	01001110	01010110
	01001111	01010111
Ciphertext	O	W

Dari tabel diatas, dapat diketahui hasil dari enkrips RC4 adalah OW

Untuk melakukan dekripsi, lakukan operasi XOR pada *ciphertext* dengan kunci yang digunakan pada proses enkripsi, seperti berikut:

Tabel 2.4 Proses XOR *Ciphertxet* dan *Keystream*

P ⊕ K		
Ciphertext	O	W
Biner	01001111	01010111
Keystream	00000001	00000001
	01001110	01010110
Plaintext	N	V

Dari tabel diatas, dapat diketahui hasil dekripsi adalah NV (*plaintext*).

2.5 Algoritma H-Rabin

Algoritma H-Rabin termasuk kriptografi asimetri. Algoritma H-Rabin adalah sebuah perubahan dari algoritma Rabin, yang memiliki daya tahan yang lebih rumit dibandingkan algoritma Rabin. Kriptografi H-Rabin menggunakan tiga bilangan prima acak dengan kunci public n = pxqxr. Dimana p, q, r adalah bilangan prima acak. Sedangkan pada algoritma Rabin, hanya menggunakan dua kunci bilangan

prima acak n=pxq (Rachmawati, Amalia, & Triska, 2020). Algoritma H-Rabin bekerja dengan tiga tahap utama, yaitu sebagai berikut:

***Key Generation* (Pembangkitan Kunci)**

Tahap pertama pada algoritma H-Rabin adalah membangkitkan *public key* dan *private key*. Berikut ini adalah langkah-langkah dalam membangkitkan kunci tersebut:

1. Tentukan 3 buah bilangan bulat prima p, q dan r ($p \neq q \neq r$), dimana ketiga bilangan tersebut konguren dengan 3 mod 4, $p \equiv q \equiv r \equiv 3\ (mod\ 4)$. Rahasiakan ketiga bilangan tersebut dan dijadikan sebagai kunci privat.
2. Hitung *n*, dimana *n* adalah kunci publik yang dibangkitkan dari ***n=p×q×r***

Proses Enkripsi

Algoritma H-Rabin hanya menggunakan kunci publik n selama proses enkripsi. *Ciphertext* didapatkan dengan persamaan, $C \equiv m_e^2\ (mod\ n)$ dimana m_e sebagai *plaintext* atau pesan yang dikirim yang sudah di *extend* dengan nilai m dan n adalah kunci publik.

Proses Dekripsi

Setelah penerima mendapatkan *chipertext* dari pengirim, maka penerima dapat melakukan dekripsi untuk mengembalikan nilai *chipertext* tersebut ke *plaintext* atau pesan awal sebelum di enkripsi. Berikut adalah langkah-langkah dalam melakukan dekripsi:

1. Terima nilai c dan kunci privat yaitu p, q, dan r
2. Hitung nilai $m_p, m_{q,} m_r$ dengan persamaan berikut:

$$m_p = C^{(\frac{p+1}{4})}(mod\ p)$$

$$m_q = C^{(\frac{q+1}{4})}(mod\ q)$$

$$m_r = C^{(\frac{r+1}{4})}(mod\ r)$$

3. Hitung nilai-nilai berikut:

   pmp = mp (mod p)

   mmp = -mp (mod p)

   pmq = mq (mod q)

   mmq = -mq (mod q)

pmr = mr (mod r)

mmr = -mr (mod r)

4. Menghitung 3 hasil kemungkinan hasil dekripsinya dengan menggunakan metode *Chinese Remainder Theorem* (CRT) seperti dibawah ini:

$$b_1 = (\frac{n}{p})^{-1} \ (mod\ p)$$

$$b_2 = (\frac{n}{q})^{-1} \ (mod\ q)$$

$$b_3 = (\frac{n}{r})^{-1} \ (mod\ r)$$

5. Cari nilai x_1 sampai x_8 dengan cara berikut:

$$x_1 = \left(pm_p . b_1 . \frac{n}{p} + \ pm_q . b_2 . \frac{n}{q} + \ pm_r . b_3 . \frac{n}{r}\right) (mod\ n)$$

$$x_2 = \left(mm_p . b_1 . \frac{n}{p} + \ pm_q . b_2 . \frac{n}{q} + \ pm_r . b_3 . \frac{n}{r}\right) (mod\ n)$$

$$x_3 = \left(pm_p . b_1 . \frac{n}{p} + \ mm_q . b_2 . \frac{n}{q} + \ pm_r . b_3 . \frac{n}{r}\right) (mod\ n)$$

$$x_4 = \left(pm_p . b_1 . \frac{n}{p} + \ pm_q . b_2 . \frac{n}{q} + \ mm_r . b_3 . \frac{n}{r}\right) (mod\ n)$$

$x_5 = (n - \ x_1)$

$x_6 = (n - \ x_2)$

$x_7 = (n - \ x_3)$

$x_8 = (n - \ x_4)$

6. Nilai x_1 sampai x_8 modulokan dengan nilai m_e
7. Jika hasil dari modulo bernilai 0, maka bagikan nilai X tersebut dengan m_e. Lalu hasil bagi di konversi ke nilai ASCII. Maka di dapatlah hasil dari dekripsi yaitu *plaintext* atau pesan sebelum di enkripsi.

Contoh Perhitungan

Tahap *Key Generation*

Menentukan bilangan prima acak p, q, r. p = 251, q = 383, r = 479

Kunci publik n = pxqxr = 251 x 383 x 479 = 46047707

Tahap Enkripsi

Plaintext pada contoh adalah 11 dengan nilai ASCII 4949 dan nilai biner 1001101010101.

1. Menghitung nilai m_e

   m = 65

   $m_e = m \mid m$ atau m x 8193

   $m_e = 1001101010101 \mid 1001101010101$ atau 4949 x 8193

   $m_e = 40547157$

2. Menghitung *ciphertext* atau C

   $C \equiv m_e^2 \ (mod\ n)$

   $C = \ 40547157^2 (mod\ 46047707)$

   C = 36036494

Tahap Dekripsi

1. Penerima menerima *ciphertext* atau C dari pengirim yaitu 21584
2. Menghitung nilai $m_p, m_{q,} m_r$

   $m_p = 36036494^{\left(\frac{251+1}{4}\right)} (mod\ 251) = 115$

   $m_q = 36036494^{\left(\frac{383+1}{4}\right)} (mod\ 383) = 96$

   $m_r = 36036494^{\left(\frac{479+1}{4}\right)} (mod\ 479) = 193$
3. Menghitung nilai – nilai berikut:

   pmp = mp (mod p) = 115 (mod 251) = 115

   mmp = -mp (mod p) = -115 (mod 251) = 136

   pmq = mq (mod q) = 96 (mod 383) = 96

   mmq = -mq (mod q) = -96 (mod 383) = 287

   pmr = mr (mod r) = 193 (mod 479) = 193

   mmr = -mr (mod r) = -193 (mod 479) = 286

4. Menghitung kemungkinan nilai b1,b2, dan b3 dengan *Chinese Remainder Theorem* (CRT)

   $b_1 = \left(\frac{46047707}{251}\right)^{-1} (mod\ 251) = (183457)^{-1} (mod\ 251)$

   $b_2 = \left(\frac{46047707}{383}\right)^{-1} (mod\ 383) = (120229)^{-1} (mod\ 383)$

   $b_3 = \left(\frac{46047707}{479}\right)^{-1} (mod\ 479) = (96133)^{-1} (mod\ 479)$

$b_1 = 183457x_1 \equiv 1\ (mod\ 251)$

$b_1 = 227x_1 \equiv 1\ (mod\ 251)$

$b_1 = x_1 = 115$

$b_2 = 120229x_2 \equiv 1\ (mod\ 383)$

$b_2 = 350x_2 \equiv 1\ (mod\ 383)$

$b_2 = x_2 = 58$

$b_3 = 96133x_3 \equiv 1\ (mod\ 479)$

$b_3 = 333x_1 \equiv 1\ (mod\ 479)$

$b_3 = x_3 = 187$

5. Menghitung nilai x_1 - x_8

x_1= (115.115.183457 + 96.58.120229 + 193.187.96133)
(mod 46047707)
= 2641506

x_2= (136.115.183457 + 96.58.120229 + 193.187.96133)
(mod 46047707)
= 8987191

x_3= (115.115.183457 + 287.58.120229 + 193.187.96133)
(mod 46047707)
= 22928965

x_4= (115.115.183457 + 96.58.120229 + 286.187.96133)
(mod 46047707)
= 40547157

$x_5 = n - x_1 = 46047707 - 2641506 = 43406201$

$x_6 = n - x_2 = 46047707 - 8987191 = 37060516$

$x_7 = n - x_3 = 46047707 - 22928965 = 23118742$

$x_8 = n - x_4 = 46047707 - 40547157 = 5500550$

6. Menghitung hasil modulo $x_1 - x_8$ dengan 8193

   $x_1 = 2641506 \; mod \; 8193 = 3360$

   $x_2 = 8987191 \; mod \; 8193 = 7663$

   $x_3 = 22928965 \; mod \; 8193 = 4951$

   $x_4 = 40547157 \; mod \; 8193 = 0$

   $x_5 = 43406201 \; mod \; 8193 = 7880$

   $x_6 = 37060516 \; mod \; 8193 = 3577$

   $x_7 = 23118742 \; mod \; 8193 = 6289$

   $x_8 = 5500550 \; mod \; 8193 = 3047$

   Dari hasil perhitungan sebelumnya, dapat diamati bahwa nilai dari $x_4 \; modulo \; 8194 = 0$ maka nilai x_4 dibagi dengan 8193 agar mendapatkan nilai *plaintext*

   $x_4 = 40547157$ / 8193 = 4949 = 11

2.6 Algoritma Extended Euclidien

Algoritma Extended Euclidean adalah algoritma yang digunakan untuk menemukan solusi dari persamaan Diophantine

$$ax + by = gcd(a,b)$$

di mana a dan b adalah bilangan bulat tertentu. Selain itu, algoritma ini juga digunakan untuk menemukan invers modular. Algoritma ini disebut "extended" karena selain menghitung faktor persekutuan terbesar gcd (a, b), juga menghasilkan koefisien x dan y yang memenuhi persamaan diophantine tersebut. Dengan demikian, algoritma ini memberikan informasi yang lebih lengkap daripada algoritma Euclidean biasa.

Contoh : berapa inverse dari 3 (mod 7)

ax + by = gcd(a,b)

3x + 7y = 1

x	y	d	k
0	1	7	
1	0	2	2
-2	1	1	

$m^{-1} \equiv x \equiv -2$

$m^{-1} \equiv -2 \equiv 5 \pmod 7$

2.7 Teori Modular Exponentiation

Teori Modular Exponentiation adalah teknik dalam matematika dan ilmu komputer yang digunakan untuk menghitung kekuatan bilangan dengan modulus, terutama saat berhadapan dengan bilangan yang sangat besar. Ini sangat penting dalam kriptografi, terutama dalam algoritma seperti RSA, Diffie-Hellman, dan algoritma-algoritma lainnya yang menggunakan operasi eksponensial dalam grup modular. Modular Exponentiation adalah operasi yang dinyatakan sebagai:

$$C = a^b \bmod m$$

di mana a adalah basis, b adalah eksponen, m adalah modulus, dan c adalah hasil dari operasi tersebut.

Contoh Perhitungan :

Hitunglah $3^4 \bmod 5$.

Penyelesaian : a = 3, b = 4, m = 5

a. Itersi 1 : b (genap)

   a = (3 x 3) mod 5 = 9 mod 5 = 4

   b = 4/2 = 2

b. Iterasi 2 : b = 2 (genap)

   a = (4 x 4) mod 5 = 16 mod 5 = 1

   b = 2/2 = 1

c. Iterasi 3 : b = 1 (ganjil)

   a = (1 x 1) mod 5 = 1

   b = 1 / 2 = 0

   ketika hasil perhitungan sudah 0, maka hasil didapat yaitu 1

BAB 3
ANALISIS DAN PERANCANGAN

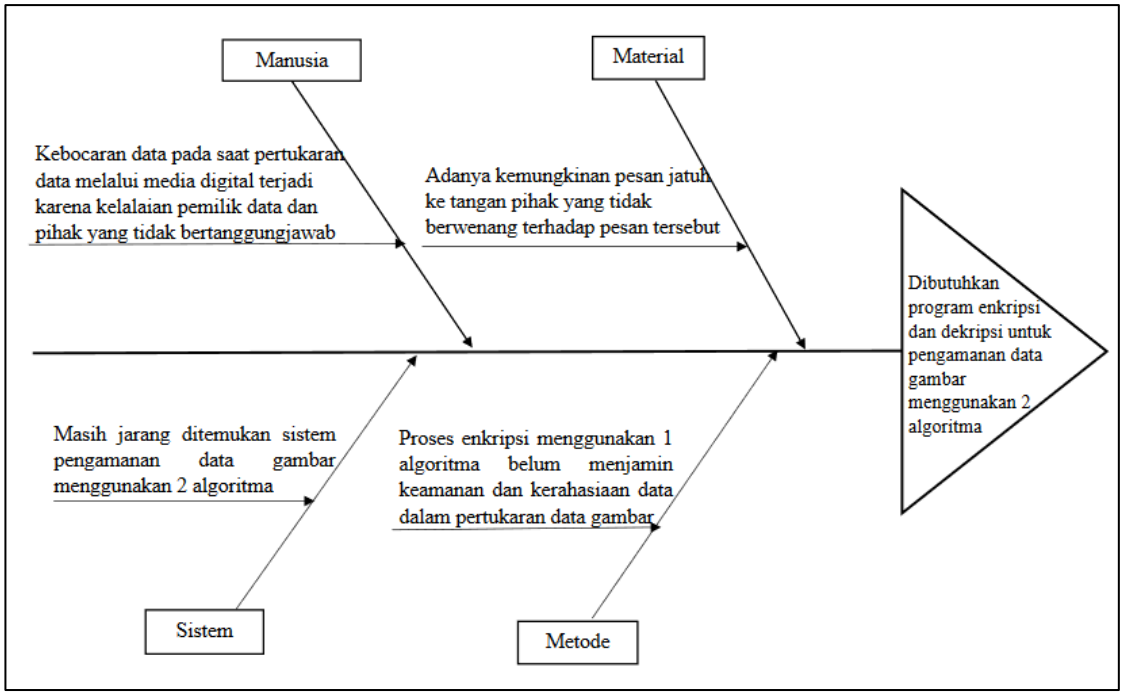
3.1 Analisis

Analisis merupakan proses menguraikan atau mengelompokkan sesuatu menjadi bagian tertentu untuk mendapatkan pemahaman arti keseluruhan. Tujuan analisis adalah untuk mendapatkan hasil akhir dari proses pengamatan yang telah dilakukan. Hasil analisis digunakan sebagai landasan untuk membangun sebuah sistem. Penelitian ini menggunkan dua tahapan analisis, yaitu analisis masalah serta analisis kebutuhan.

3.1.1 Analisis masalah

Permasalahan pada keamanan data adalah satu hal yang sangat penting, terutama kemanan pada proses pengiriman data berupa gambar melalui media digital. Kemajuan teknologi yang sangat pesat menyebabkan pihak yang tidak berwenang dengan mudah memodifikasi dan mengambil keuntungan dari hal tersebut.

Faktor masalah utama yang akan diselesaikan dalam penelitian ini digambarkan secara umum menggunakan *Fishbone Diagram* pada gambar 3.1. Terlihat bahwa ada beberapa penyebab masalah yang diilustrasikan sebagai tulang ikan yaitu terdiri dari manusia, material, sistem dan metode. Sedangkan kepala ikan mengilustrasikan simpulan masalah yang akan diselesaikan yaitu pengamanan ganda untuk menjaga kerahasiaan gambar.



Gambar 3.1 Diagram Ishikawa

3.1.2 Analisis kebutuhan

Analisis kebutuhan terdiri dari dua bagian utama, yaitu fungsional dan non-fungsional.

1. Kebutuhan fungsional

   Kebutuhan fungsional merupakan spesifikasi fungsionalitas yang perlu ada dan harus dilakukan pada sistem agar mencapai tujuan utama. Penelitian ini memiliki kebutuhan fungsional utama, yaitu:

   a. Menerima masukan kunci

      Sistem mampu menerima dan membaca *input* kunci untuk enkripsi dengan algoritma *Rivest Code* 4 (RC4) dari *user.*

   b. Menerima masukan gambar

      Sistem mampu mencari dan membaca *file* gambar yang berekstensi *Portable Network Graphics (.png)* yang tersimpan pada direktori pengguna.

   c. Menerima keluaran gambar terenkripsi

      Sistem mampu menerima *file* gambar hasil enkripsi yang berkstensi *Portable Network Graphics (.png).*

d. Menerima masukan kunci publik

   Sistem dapat menerima dan membaca *public key* yang dimasukkan oleh pengguna.

e. Enkripsi Pesan

   Sistem mampu mengenkripsi pesan berupa gambar menggunakan algoritma RC4 sehingga gambar tidak terbaca

f. Enkripsi Kunci

   Sistem mampu mengenkripsi kunci yang digunakan pada enkripsi sebelumnya sehingga menghasilkan *cipherkey*.

g. Membangkitkan kunci publik

   Sistem mampu menghasilkan kunci publik hasil dari mengacak 3 bilangan prima.

h. Dekripsi kunci

   Sistem mampu mendekripsi *cipherkey* dengan menggunakan kunci rahasia dan algoritma *H-Rabin*.

i. Dekripsi pesan

   Sistem mampu mendekripsi atau mengembalikan nilai pesan berupa gambar dengan menggunakan kunci yang telah didekripsi dan menggunakan algoritma RC4.

j. Simpan *file*

   Sistem mampu menyimpan *file* hasil enkripsi dan dekripsi

2. Kebutuhan non-fungsional

   Kebutuhan non-fungsional merupakan spesifikasi tambahan yang mendukung kinerja sistem menjadi lebih baik lagi. Penelitian ini memiliki beberapa kebutuhan non-fungsional, yaitu:

   1. User Interface

      Sistem dirancang dengan sederhana agar pengguna tidak bingung saat menggunakannya.

   2. Kontrol

      Sistem memiliki kontrol yaitu menampilkan pesan eror jika inputan *user* tidak sesuai.

3. Performa

   Sistem dapat menangani banyak proses enkripsi dan dekripsi pada saat sistem di *running.*

4. Kualitas

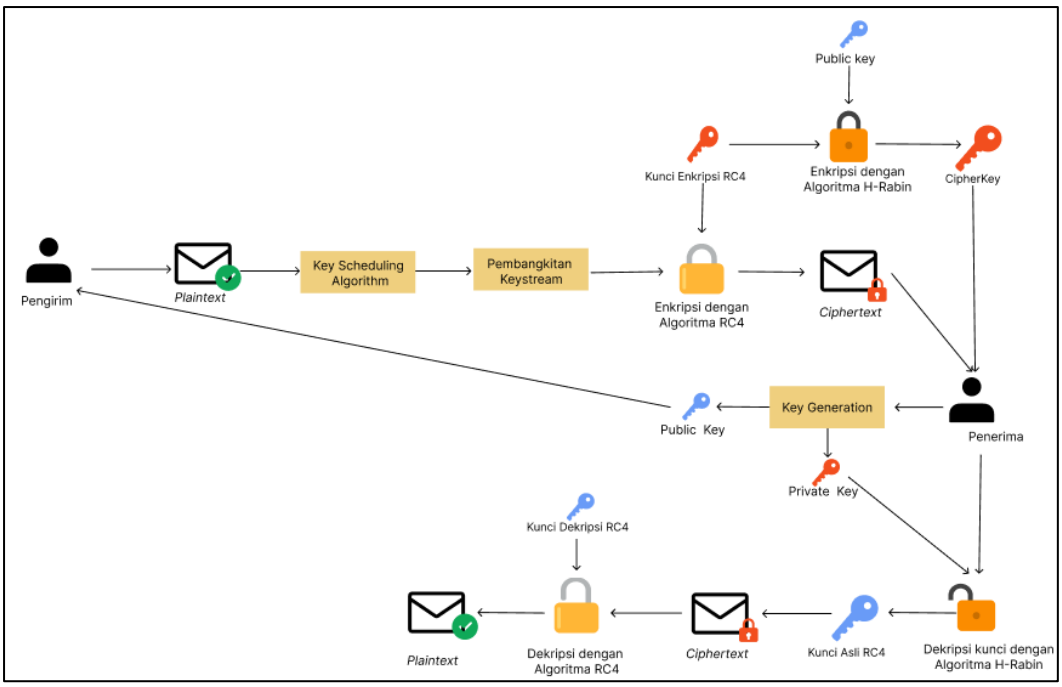
   Kualitas yang baik dari sistem yaitu sistem mampu menunjukkan waktu proses enkripsi dan dekripsi pada setiap tahapan dalam satuan *millisecond* (ms).

3.2 Perancangan Sistem

Sistem dirancang atas dasar analisis yang telah dilakukan terhadap penelitian sebelumnya. Pemodelan sistem menggunakan berbagai diagram, termasuk diagram umum, diagram *use case*, *activity diagram*, *sequence diagram*, dan *user interface*.

3.2.1 Diagram umum penelitian

Diagram umum penelitian digunakan untuk menjelasakan secara kesulurahan serta membri gambaran umum bagaimana cara kerja dari sistem yang dibangun.



Gambar 3.2 Diagram Umum

Berdasarkan Gambar 3.2 menunjukkan konsep perancangan dan bagaimana cara sistem yang akan dibangun bekerja. Urutan dari alur sistem yaitu

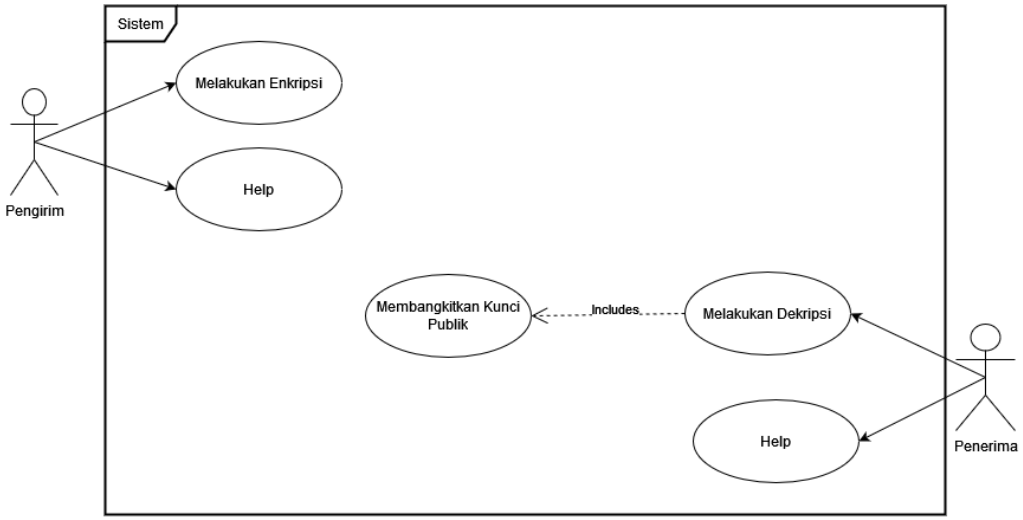
terdapat proses enkripsi pesan, enkripsi kunci, dekripsi kunci, dan dekripsi pesan. Berikut runtutan proses yang terjadi pada diagram umum:

1. Pengirim menyiapkan pesan atau *plaintext* yang akan dikirim kepada penerima. Pesan terlebih dahulu dienkripsi dengan algoritma *Rivest Code 4* (RC4) sebelum dikirimkan dengan memberikan kunci rahasia RC4. Hasil dari enkripsi akan menghasilkan *ciphertext* yang akan dikirim kepada penerima.
2. Pengirim melakukan enkripsi melakukan enkripsi terhadap kunci RC4 yang digunakan untuk mengenkripsi pesan sebelumnya menggunkan kunci publik yang didapat dari penerimaa. Enkripsi kunci ini dilakukan menggunakan algoritma *H-Rabin*, sehingga menghasilkan *cipherkey* yang akan dikirim kepada penerima bersama dengan *ciphertext*.
3. Penerima yang sudah mendapatkan *cipherkey* dan *ciphertext* terlebih dahulu melakukan dekripsi terhadap *cipherkey* yang didapat dengan kunci rahasia yang dimilikinya. Dekripsi *cipherkey* ini dilakukan dengan algoritma H-Rabin sehingga menghasilkan kunci asli dari RC4 yang digunakan pada saat pengirim melakukan enkripsi.
4. Setelah penerima melakukan dekripsi *cipherkey*, selanjutnya penerima melakukan dekripsi *ciphertext* dengan kunci yang sudah berhasil di dekripsi. Dekripsi *ciphertext* ini dilakukan dengan algoritma RC4.

3.2.2 Use case Diagram

Use case diagram digunakan sebagai gambaran dari interaksi antar pengguna dan sistem. Gambar 3.3 terdapat dua aktor yaitu *sender* dan *receiver*. *Sender* dan *receiver* bisa melakukan proses enkripsi dan dekripsi yang dilakukan dengan algoritma *Rivest Code 4* (RC4) dan algoritma *H-Rabin*. Pengirim dapat menjalankan sistem dengan cara menginputkan *plaintext* yang berupa file gambar, serta meginputkan kunci rahasia yang digunakan untuk mengenkripsi pesan tersebut menggnakan algoritma RC4. Setalah itu pengirim juga perlu menginputkan kunci public yang diterima dari penerima pesan untuk melakukan enkripsi kunci RC4. Sehingga sistem mengahasilkan *ciphertext* dan *cipherkey*. Dan begitu pula hal sebaliknya yang dilakukan penerima yaitu menginputkan

cipherkey untuk di dekripsi terlebih dahulu, lalu menginputkan ciphertext untuk di dekripsi sehingga menghasilkan pesan asli.



Gambar 3.3 *Use Case* Diagram

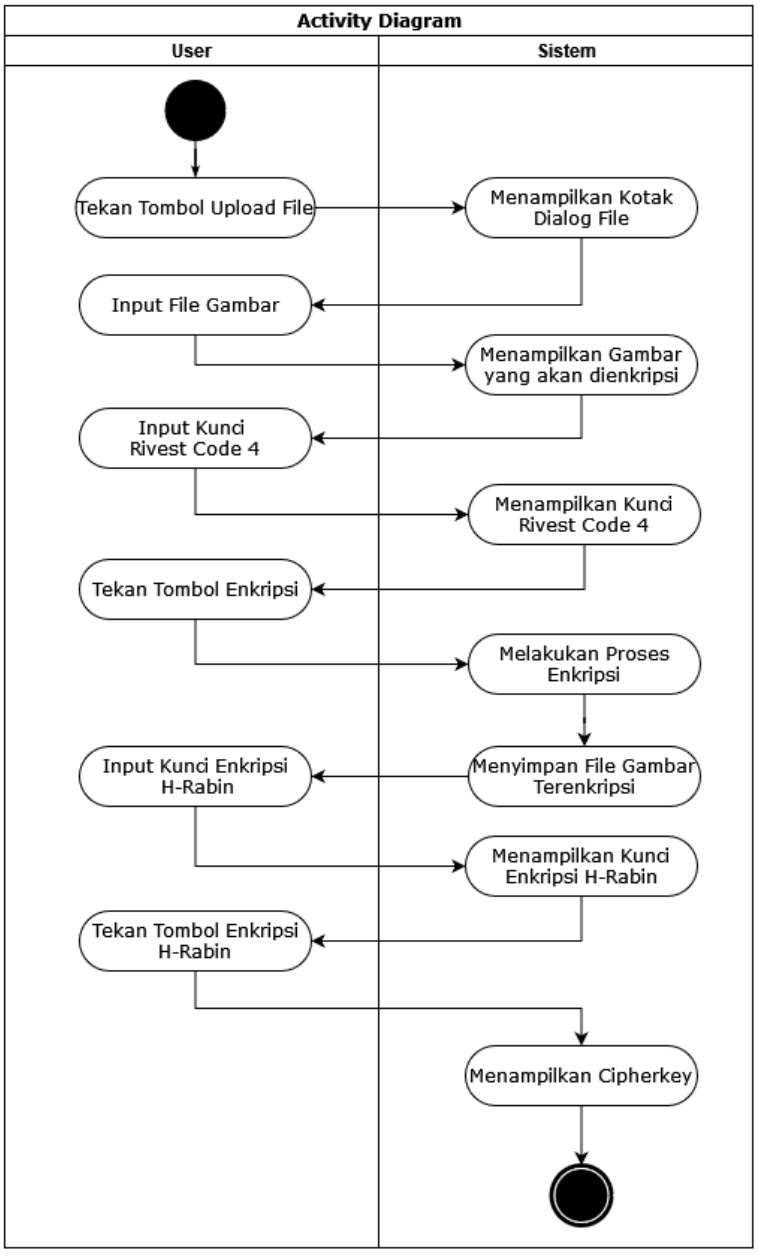
3.2.3 Activity Diagram

Activity diagram menggambarkan alur kerja berdasarkan sistem yang akan dibangun. Diagram ini dibangun untuk memberikan pehamaan dari proses secara keseluruan serta menjelaskan interaksi antar beberapa *use case*. Pada sistem ini terdapat 3 activity diagram yaitu activity diagram untuk pengirim, activity diagram untuk penerima, activity diagram bantuan.

3.2.3.1 Activity Diagram Pengirim

Gambar 3.4 menunjukkan bagaimana pengirim melakukan enkripsi pada sistem yang akan dibangun. Diagram tersebut menunjukkan kegiatan yang dilakukan oleh pengirim pada kotak sebelah kiri, sementara kotak di sebelah kanan menggambarkan respons sistem terhadap kegiatan yang dilakukan pengirim terhadap sistem Pengirim mencari dan menginputkan *file* gambar sehingga sistem akan merespon dan menampilkan kotak dialog *file* cari. Setelah itu pengirim memasukkan kunci rahasia *Rivest Code* 4 (RC4) kemudian sistem akan menampilkan kunci. Pengirim menekan tombol enkripsi RC4 sehingga sistem akan merepon dan melakukan enkripsi, hasil enkripsi akan disimpan. Pengirim dapat melanjutkan proses enkripsi kunci dengan menginputkan kunci public,

sehingga sistem akan menampilkan kunci publik. Lalu pengirim menekan tombol enkripsi *H-Rabin*, maka sistem akan mengenkkripsi kunci dan menampilkan *cipherkey*.

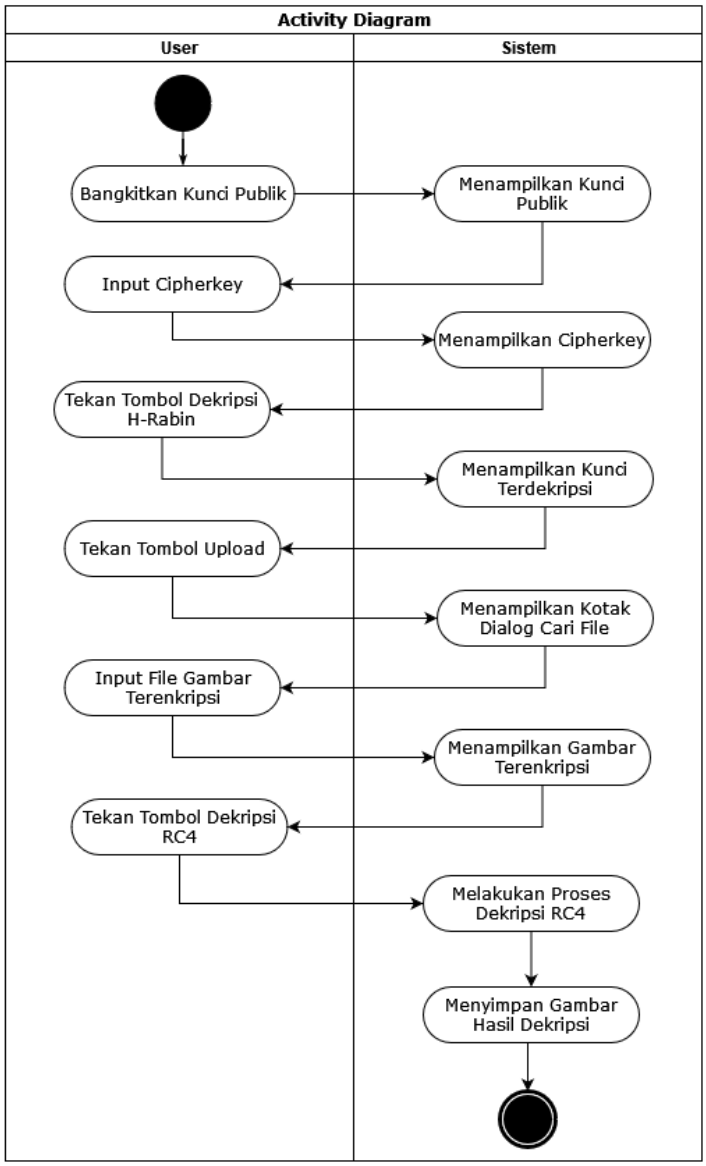


Gambar 3.4 Activity Diagram Pengirim

3.2.3.2 Activity Diagram Penerima

Activity diagram penerima akan menampilkan proses dekripsi yang dilakukan terhadap sistem yang dibangun. Penerima harus membangkitkan kunci publik sehingga sistem akan merespon menampilkan kunci publik, setelah itu

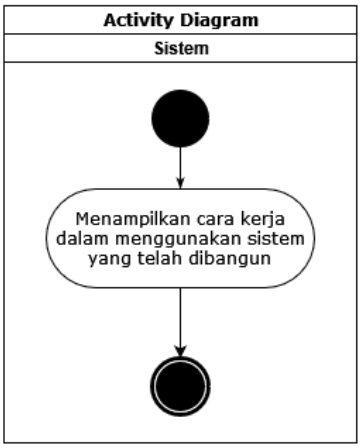
penerima menginputkan *cipherkey* untuk didekripsi lalu sistem akan merespon menampilkan *cipherkey*. Proses selanjutnya penerima dapat menekan tombol dekripsi *H-Rabin* sehingga sistem merespon menampilkan hasil dekripsi yaitu kunci rahasia RC4. Setelah itu penerima menekan tombol *upload* untuk menginputkan *file* gambar terdekripsi lalu sistem akan menampilkan dialog *file* cari, penerima akan menginputkan gambar dan sistem menampilkan gambar. Penerima dapat melanjutkan ke proses selanjutnya yaitu menekan tombol dekripsi RC4 untuk melakukan dekripsi yang akan direspon dan dilakukan oleh sistem serta menyimpan *file* yang sudah terdekripsi. *Activity diagram* penerima dapat dilihat pada gambar 3.5.



Gambar 3.5 Activity Diagram Penerima

3.2.3.3 Activity Diagram Bantuan

Pada gambar 3.6 sistem akan menampilkan cara kerja menggunakan aplikasi yang dibangun.



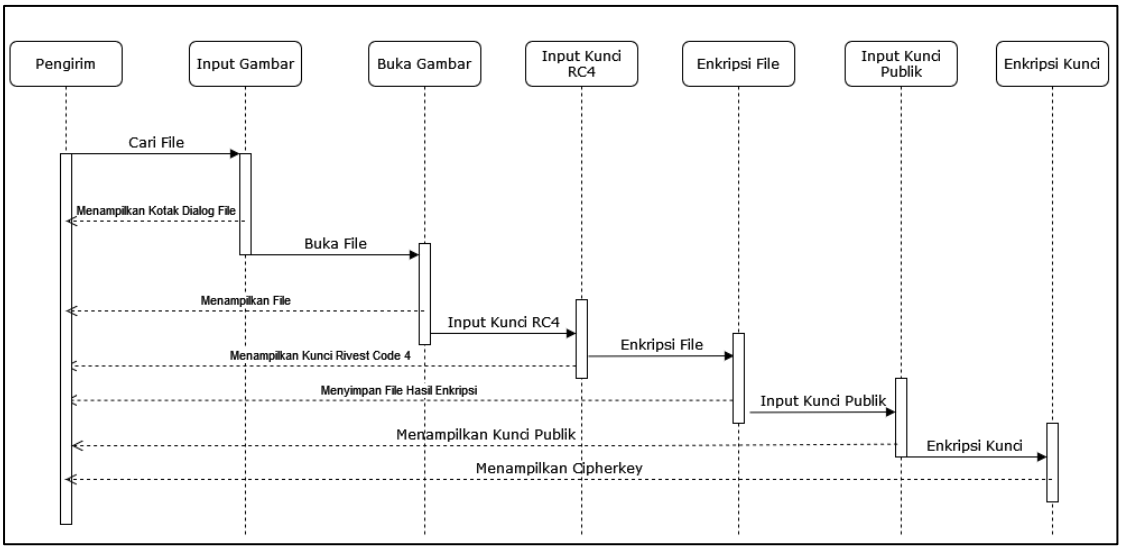
Gambar 3.6 Activity Diagram Bantuan

3.3 *Sequence Diagram*

Sequence Diagram menjelaskan hubungan antar komponen sistem yang satu dengan lainnya dalam urutan tertentu dan digambarkan dalam diagram. Diagram ini terdiri atas dimensi waktu (vertikal) dan dimensi objek (horizontal) pada sistem. Interaksi antara user ke sistem digambarkan dengan garis panah horizontal dan interaksi dari sistem ke user digambarkan dengan garis panah putus-putus horizontal. *User* yang terdapat dalam diagram ini adalah pengirim dan penerima.

3.3.1 Sequence Diagram Enkripsi (Pengirim)

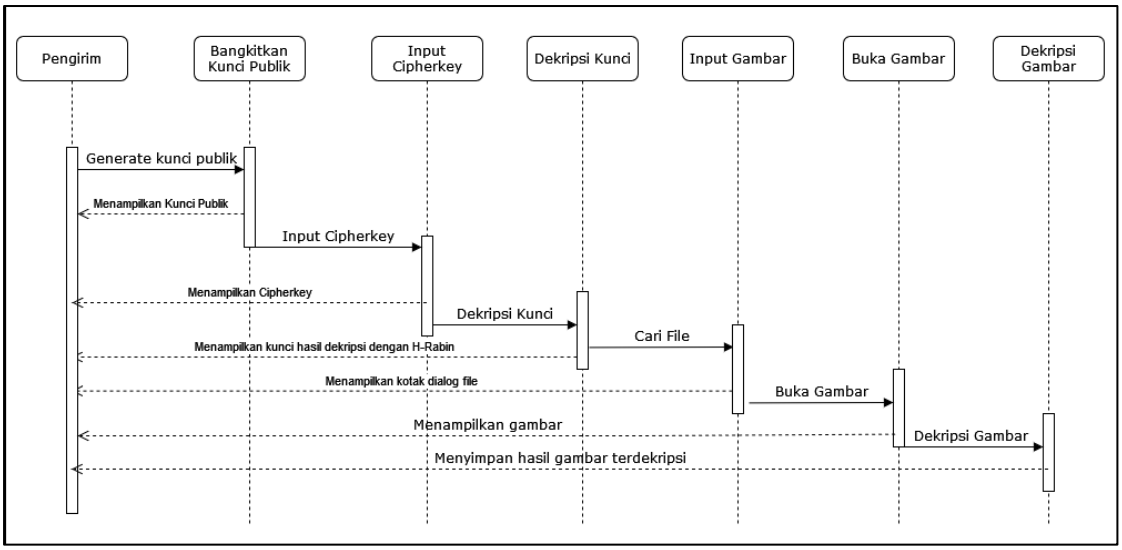
Sequence diagram ditunjukkan pada gambar 3.7 untuk interaksi antara sistem dan pengirim pada tahap enkripsi gambar dan kunci.



Gambar 3.7 *Sequence* Diagram Enkripsi (Pengirim)

3.3.2 Sequence Diagram Dekripsi (Penerima)

Pada gambar 3.8 dapat dilihat sequence diagram untuk interaksi antara sistem dan penerima pada tahap dekripsi kunci dan gambar.



Gambar 3.8 *Sequence* Diagram Dekripsi (Pengirim)

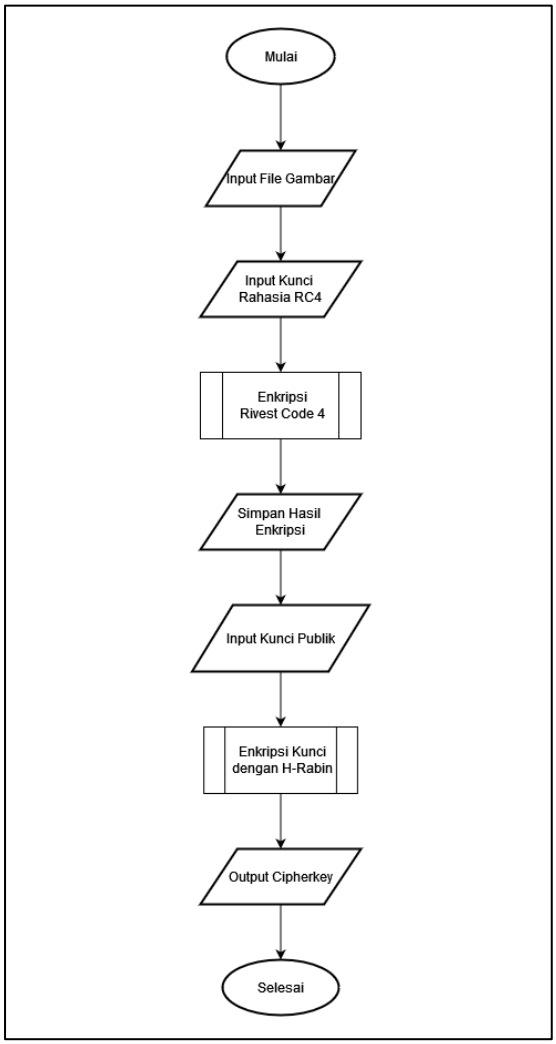
3.4 Diagram Alir (*Flowchart*)

Flowchart merupakan sebuah diagram atau bagan yang menggambarkan tahapan dalam penyelesaian masalah yang digambarkan dengan simbol – simbol tertentu yang mudah untuk dipahami. *Flowchart* bertujuan untuk menyelesaikan masalah secara sistematis sehingga mendapatkan Solusi. Adapun *flowchart*

dalam penelitian ini adalah *flowchart* enkripsi, *flowchart* dekripsi, *flowchart* pembangkitan kunci *public, flowchart* enkripsi dan dekripsi *Rivest Code* 4 (RC4),, dan *flowchart* enkripsi dan dekripsi H-Rabin.

3.4.1 Flowchart Enkripsi (Pengirim)

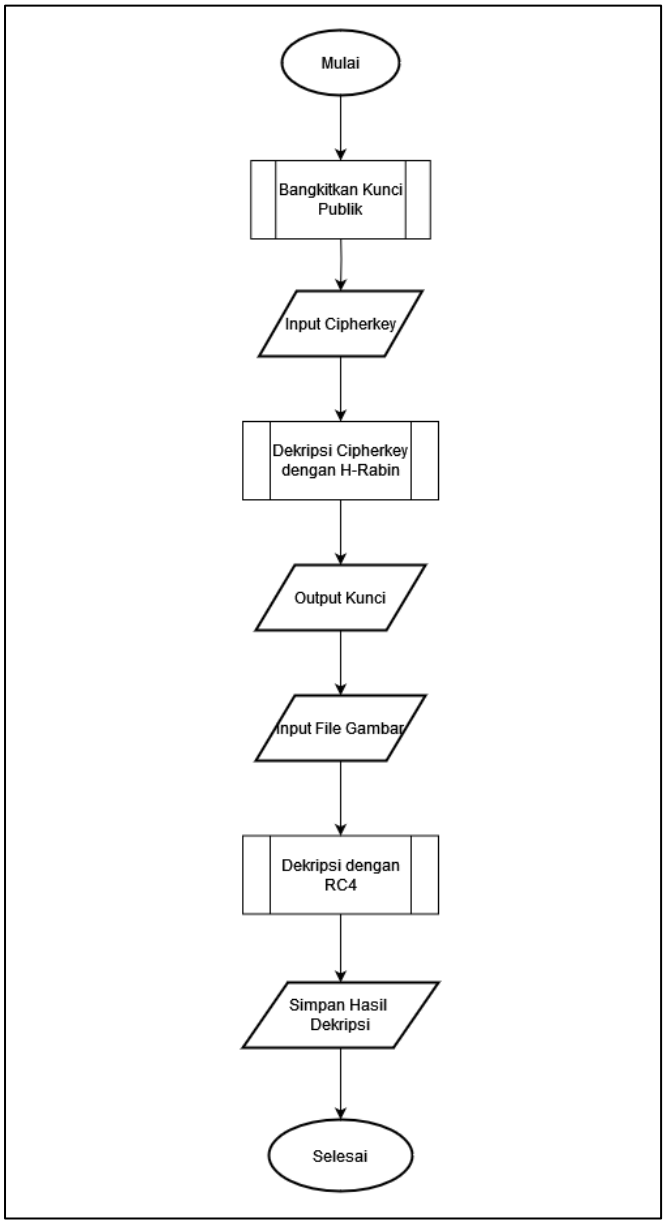
Gambar 3.9 merupakan flowchart untuk enkripsi yang dilakukan pengirim. Flowchart enkripsi menunjukkan proses enkripsi yang terdapat pada sistem yang dibangun. Proses enkripsi diawali dengan pengirim menginputkan gambar dan kunci rahasia RC4, kemudian dilakukan enkripsi dengan algoritma Rivest Code 4 (RC4) lalu gambar terenkripsi disimpan. Setelah itu pengirim menginputkan kunci public untuk melakukan enkripsi kunci rahasia RC4 sehingga diperoleh cipherkey hasil enkripsi dengan algoritma H-Rabin.



Gambar 3.9 *Flowchart* Enkripsi (Penerima)

3.4.2 Flowchart Dekripsi (Penerima)

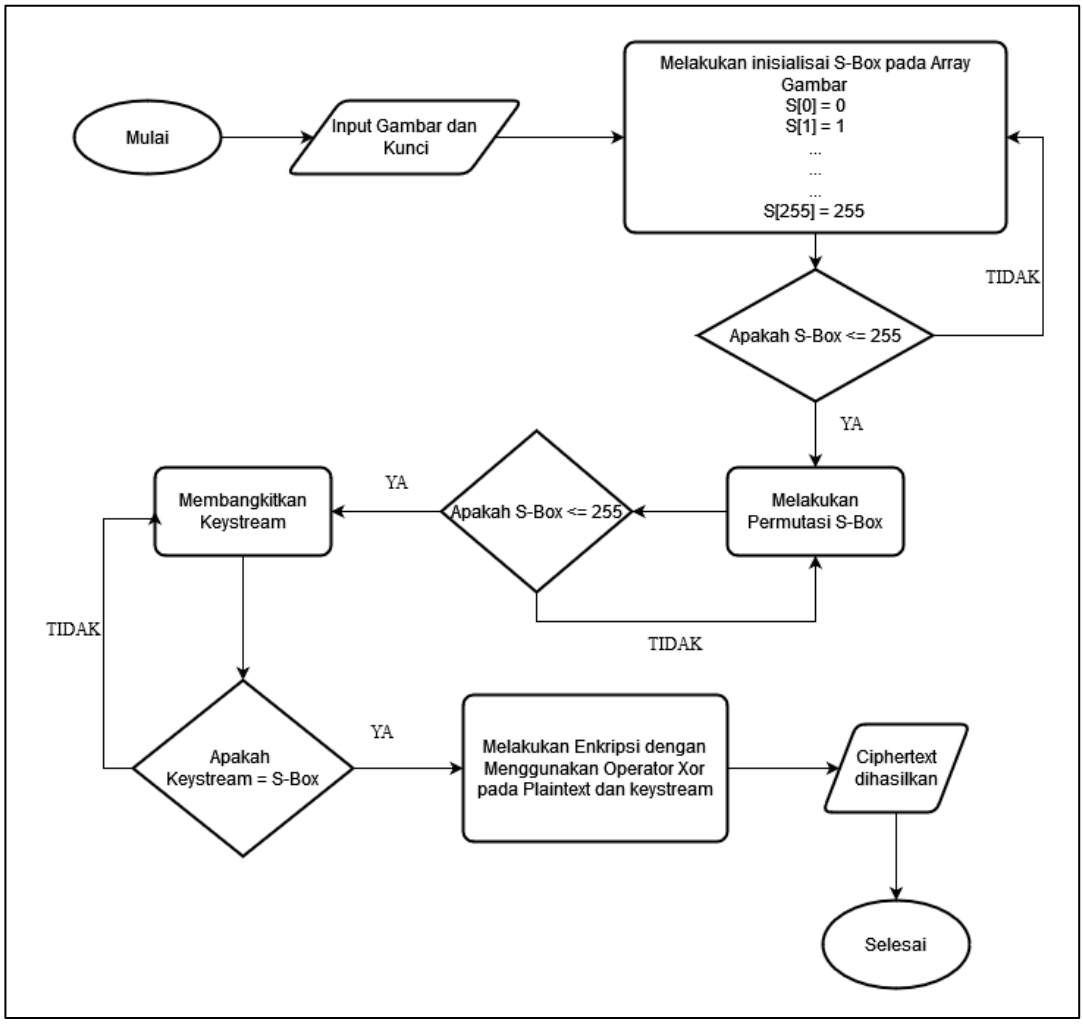
Gambar 3.10 merupakan *flowchart* untuk dekripsi yang dilakukan penerima, *flowchart* menjelaskan proses dekripsi yang terdapat pada sistem. Proses dekripsi diawali dengan penerima menginputkan *cipherkey* lalu dilakukan dekripsi dengan algoritma *H-Rabin* untuk mendapatkan kunci asli. Selanjutnya penerima menginputkan gambar terenkripsi untuk dilakukan dekripsi denganalgoritma RC4 menggunakan kunci yang sudah didekripsi, sehingga diperoleh gambar asli yang berhasil didekripsi lalu gambar tersebut langsung disimpan.



Gambar 3.10 *Flowchart* Dekripsi (Penerima)

3.4.3 Flowchart Enkripsi dan Dekripsi Algoritma Rivest Code 4 (RC4)

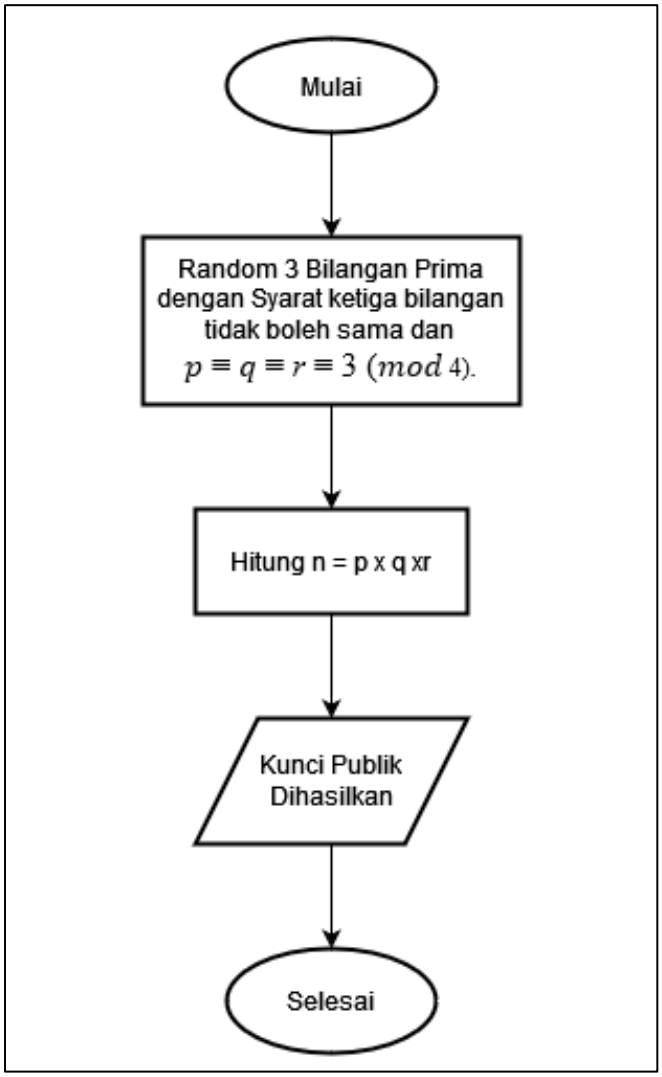
Proses enkripsi dan dekripsi *Rivest Code 4* sama, menggunakan operator XOR, jadi tidak ada perbedaan. Proses enkripsi dan dekripsi RC4 diawali dengan menginputkan gambar dan kunci, lalu sistem akan membaca byte data pada gambar. Proese selanjutnya adalah proses inisialisai S-Box pada array gambar dan array kunci, lalu terdapat kondisi apakah S-Box elbih kecil dari 256, jika iya maka dilakukan lagi proses inisalisasi sampai memenuhi kondisi. Proses selanjutnya adalah permutasi nilai-nilai S-Box pada proses ini juga terdapat kondisi bilamana jika tidak memenuhi kondisi maka proses permutasi akan terus dilanjutkan sampai kondisi terpenuhi. Jika kondisi sudah terpenuhi prose selanjutnya adalah pembankitan keystream. Bila keystream sudah memenuhi kondisi maka proses dilanjutkan ke tahap enkripsi ataupun dekripsi dengan menggunakan operator XOR, sehingga ciphertext ataupun *plaintext* dihasilkan.



Gambar 3.11 *Flowchart* Enkripsi dan Dekripsi Algoritma *Rivest Code 4*

3.4.4 Flowchart Pembangkitan Kunci Publik

Gambar 3.12 merupakan *flowchart* untuk pembangkitan kunci publik yang dilakukan oleh penerima. Proses pembangkitan kunci dimulai dengan menciptakan kunci rahasia melalui pengacakan tiga bilangan prima yang memenuhi syarat, setelah kunci rahasia didapatkan maka dilakukan operasi perkalian antara kunci rahasia untuk menghasilkan kunci publik.

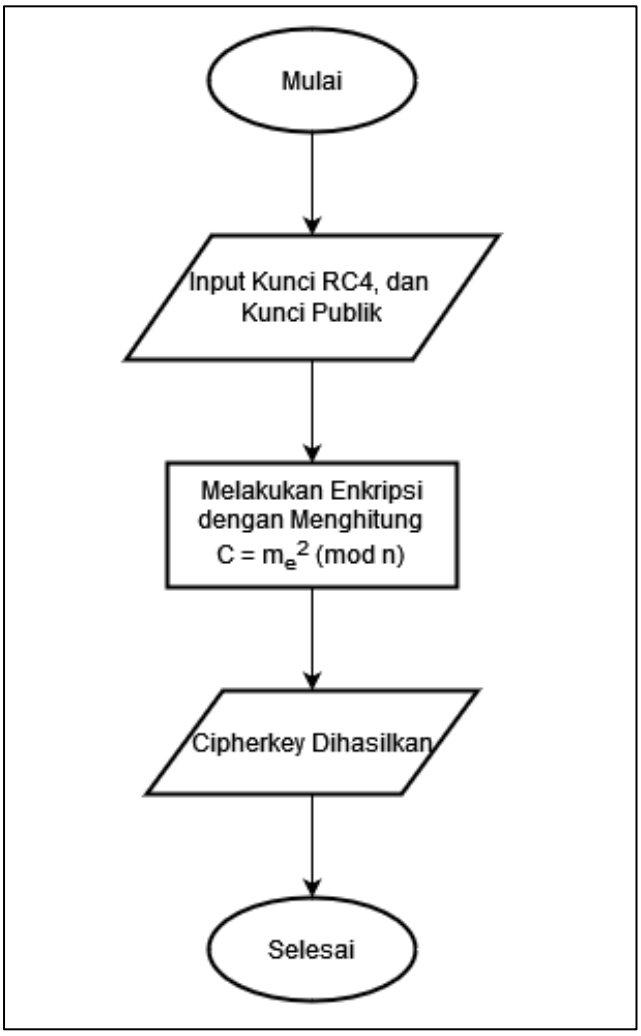


Gambar 3.12 *Flowchart* Pembangkitan Kunci Publik

3.4.5 Flowchart Enkripsi Algoritma H-Rabin

Proses enkripsi diawali dengan pengirim menginputkan kunci rahasia RC4 dan kunci publik yang diterima dari penerima. Proses selanjutnya adalah melakukan

enkripsi dengan menghitung C = me2 (mod n) sehingga *cipherkey* didapatkan. Berikut gambar *flowchart* enkripsi *H-Rabin*.

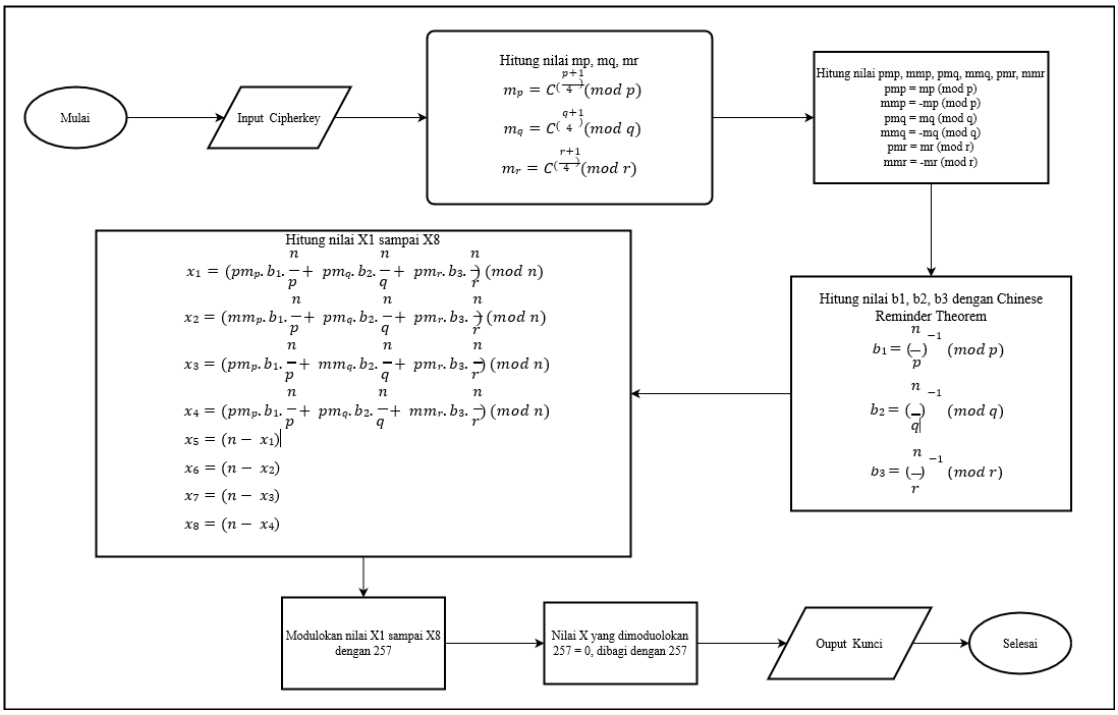


Gambar 3.13 *Flowchart* Enkripsi *Algoritma H-Rabin*

3.4.6 Flowchart Dekripsi Algoritma H-Rabin

Gambar 3.14 merupakan *flowchart* proses dekripsi dengan algoritma *H-Rabin* yang dilakukan penerima. Prose dekripsi diawali dengan menginputkan *cipherkey* lalu menghitung nilai mp, mq, mr dengan kunci rahasia yang dibangkitkan untuk membangkitkan kunci public. Setalah itu menghitung masing-masing niali pmp, mmp, pmq, mmq, pmr, mmr. Proses selanjutnya menghitung nilai b1, b2, b3 dengan menggunakan *Chinese reminder thoreme*, lalu hitung 8 kemungkinan hasil dekripsi yaitu X1 sampai X8, setalah itu modulokan nilai X1 sampai X8 dengan 257. Jika hasil X1 sampai X8 yang

dimodulokan dengan 257 sama dengan 0, maka dibagi dengan 257, sehingga didapatlah kunci asli.



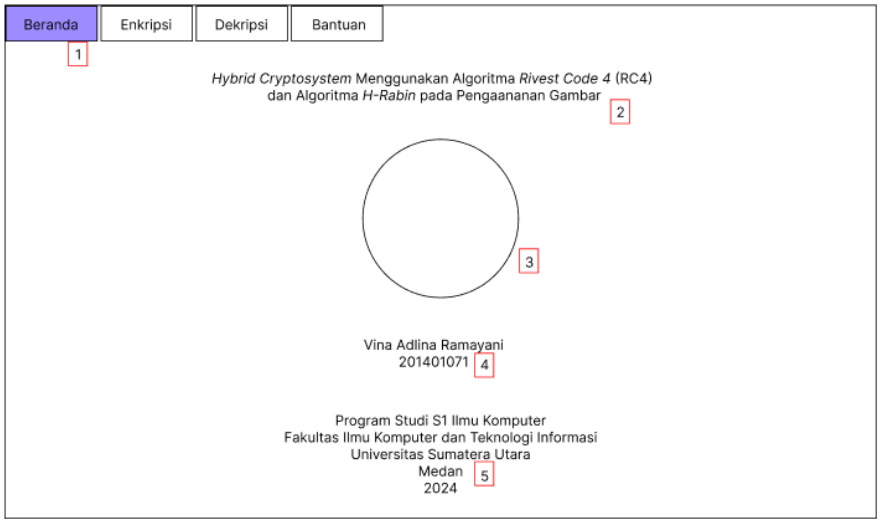
Gambar 3.14 *Flowchart* Dekripsi Algortima *H-Rabin*

3.5 *User Interface*

User Interface merupakan antarmuka yang digunakan untuk memberikan suatu gambaran kepada pengguna bagaimana cara berinteraksi dengan menggunakan tampilan yang ada di layar komputer.

3.5.1 Rancangan Menu Beranda

Pada program terdapat 4 menu list yaitu beranda, enkripsi, dekripsi, dan bantuan. Pada menu atau halaman beranda berisikan judul sistem, logo universitas, identitas penulis, dan keterangan program studi.



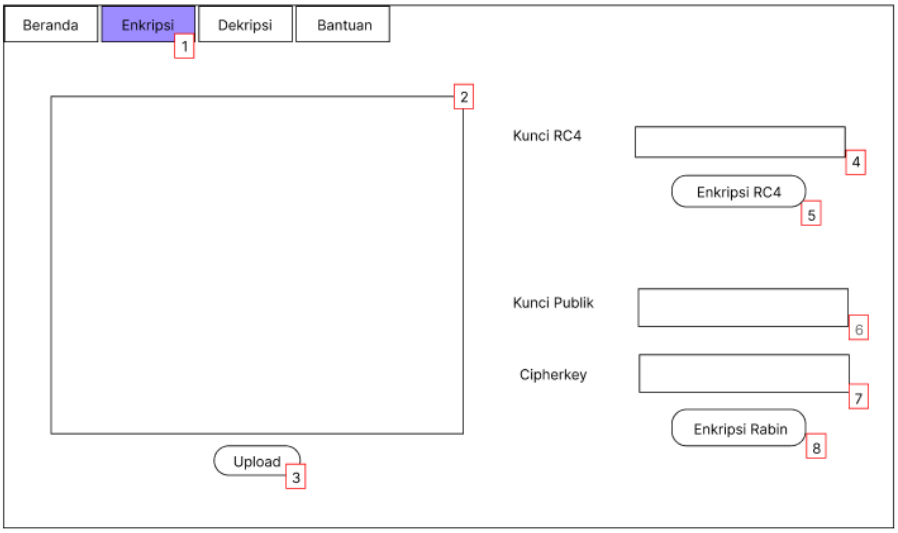
Gambar 3.15 Antarmuka Halaman Beranda

Berikut adalah keterangan gambar 3.15 :

1. *Menustrip* digunakan untuk menampilkan menu
2. *Label* digunakan untuk menampilkan judul peneilitian.
3. *Picturebox* digunakan untuk menampilkan logo universitas.
4. *Label* digunakan untuk menampilkan program studi.

3.5.2 Rancangan Menu Enkripsi

Halaman enkripsi menampilkan proses enkripsi dilakukan pengirim dengan menggunakan algoritma *Rivest Code 4* (RC4) dan algoritma *H-Rabin.*



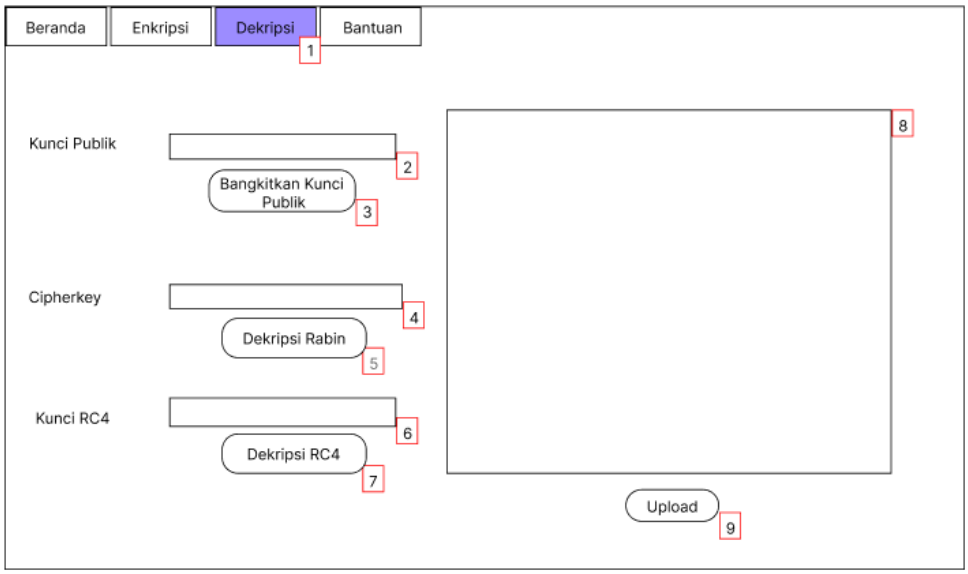
Gambar 3.16 *User Interface* Halaman Enkripsi

Berikut adalah keterangan gambar 3.16:

1. *Menustrip*, digunakan untuk menampilkan menu enkripsi.
2. *Picturebox*, digunakan untuk menampilkan gambar yang pengguna.
3. *Button*, digunakan untuk mencari dan menginputkan gambar.
4. *Textbox*, digunakan untuk memasukkan dan menampilkan input kunci RC4 dari user.
5. *Button*, untuk proses enkripsi RC4 terhadap file gambar yang sudah diinputkan.
6. *Textbox*, digunakan untuk memasukkan dan menampilkan kunci public yang didapat dari penerima.
7. *Textbox*, digunakan untuk menampilkan hasil enkripsi kunci dengan H-Rabin.
8. *Button*, digunakan untuk melakukan proses enkripsi kunci dengan algoritma H-Rabin.

3.5.3 Rancangan Menu Dekripsi

Halaman dekripsi menampilkan proses pembangkitan kunci dan proses dekripsi yang dilakukan oleh penerima dengan algoritma *H-Rabin*, dan algoritma *Rivest Code 4* (RC4).

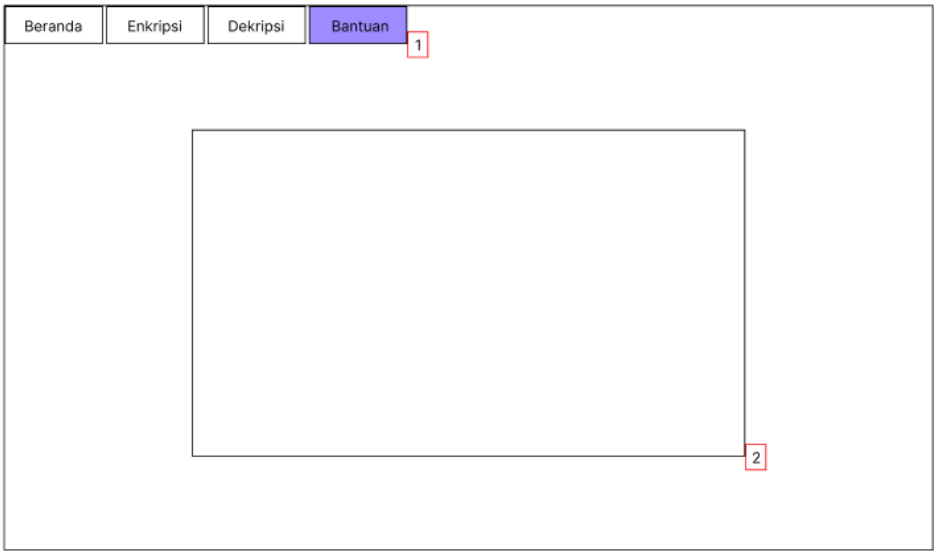


Gambar 3.17 *User Interface* Halaman Dekripsi

1. *MenuStrip*, digunakan untuk menampilkan menu dekripsi.
2. *Textbox*, digunakan untuk menampilkan kunci public yang dibangkitkan.
3. *Button*, digunakan untuk melakukan tahap pembangkitan kunci.
4. *Textbox*, digunakan untuk untuk memasukkan dan menampilkan kunci hasil enkripsi (*cipherkey*).
5. *Button*, digunakan untuk melakukan proses dekripsi kunci dengan *H-Rabin* untuk mendapatkan kunci asli.
6. *Textbox*, digunakan untuk menampilkan kunci asli RC4 atau kunci hasil dekripsi.
7. *Button*, digunakan untuk melakukan proses dekripsi gambar.
8. *Picturebox*, digunakan ntuk menampilkan gambar.
9. *Button*, digunakan untk mencari dan memasukkan gambar.

3.5.4 Rancangan Menu Bantuan

Halaman bantuan menampilkan cara kerja dan petunjuk penggunaan sistem yang dirancang untuk memberikan kemudahan pada user.



Gambar 3.18 Antarmuka Halaman Bantuan

Berikut adalah keterangan gambar 3.18:

1. *Menustrip*, digunkan untuk menampilkan menu bantuan.
2. *Picturebox*, digunakan untuk menampilkan halaman tentang aplikasi yang berisi mengenai petunjuk menggunakan sistem.

BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi dan uji sistem dilakukan setelah proses analisis dan perancangan selesai. Pada penelitian ini melibatkan sebuah software untuk melakukan implementasi yaitu sharpdevelop dengan bahasa pemrograman C#. Pengujian dilakukan dengan mencoba melakukan proses kriptografi pada sistem. Sistem ini terdapat empat halaman, termasuk halaman beranda, halaman enkripsi, halaman dekripsi, dan halaman bantuan.

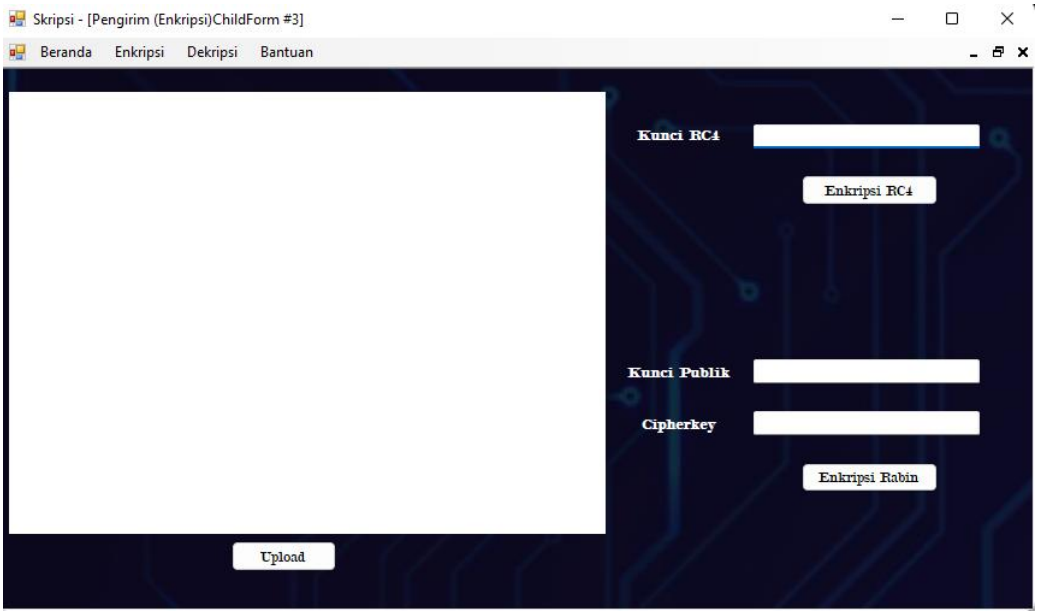
4.1.1 Halaman Beranda

Halaman ini akan tampil jika *user* memilih menu beranda pada strip menu diatas. Gambar 4.1 menunjukkan tampilan halaman beranda.

Gambar 4.1 Implementasi Halaman Beranda

Pada Gambar 4.1 menunjukkan halaman beranda untuk menampilkan informasi berupa judul penelitian, nama penulis, dan informasi prodi penulis.

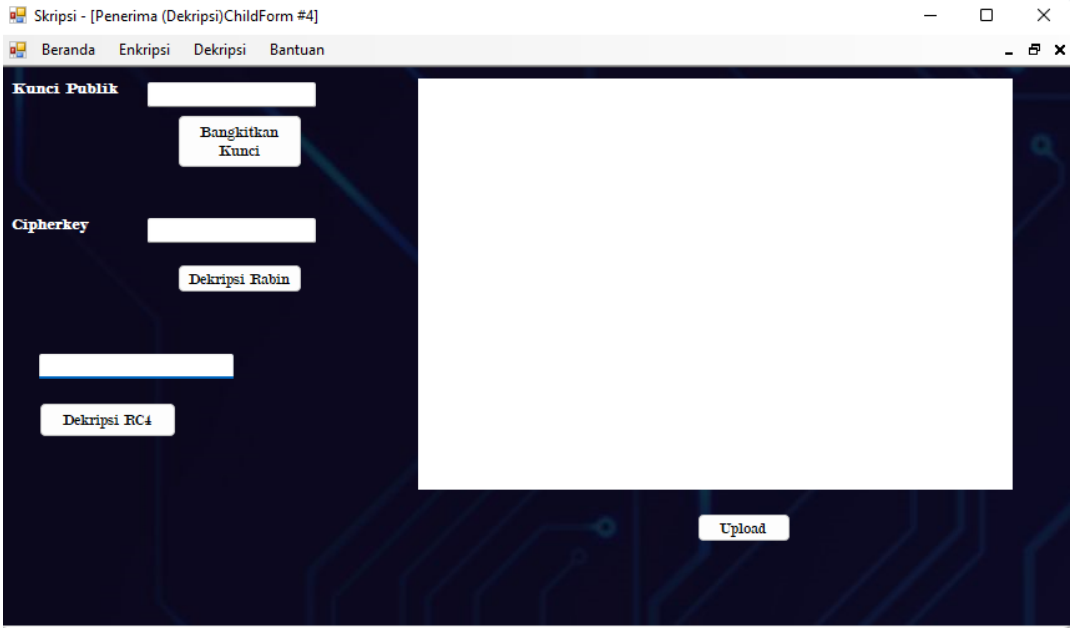
4.1.2 Halaman Enkripsi (Pengirim)



Gambar 4.2 Implementasi Halaman Enkripsi (Pengirim)

Gambar diatas memperlihatkan halaman enrkipsi untuk melakukan tahap enkripsi pada gambar. Halaman enkripsi terdapat proses lokasi pesan yang berfungsi untuk memasukkan *file* yang akan dienkripsi. Terdapat *Textbox* untuk memasukkan nilai dari kunci *Rivest Code 4* (RC4), *Textbox* untuk memasukkan nilai dari kunci publik, dan *Textbox* untuk menampilkan kunci RC4 yang dienkripsi dengan algoritma *H-Rabin*.

4.1.3 Halaman Dekripsi (Penerima)

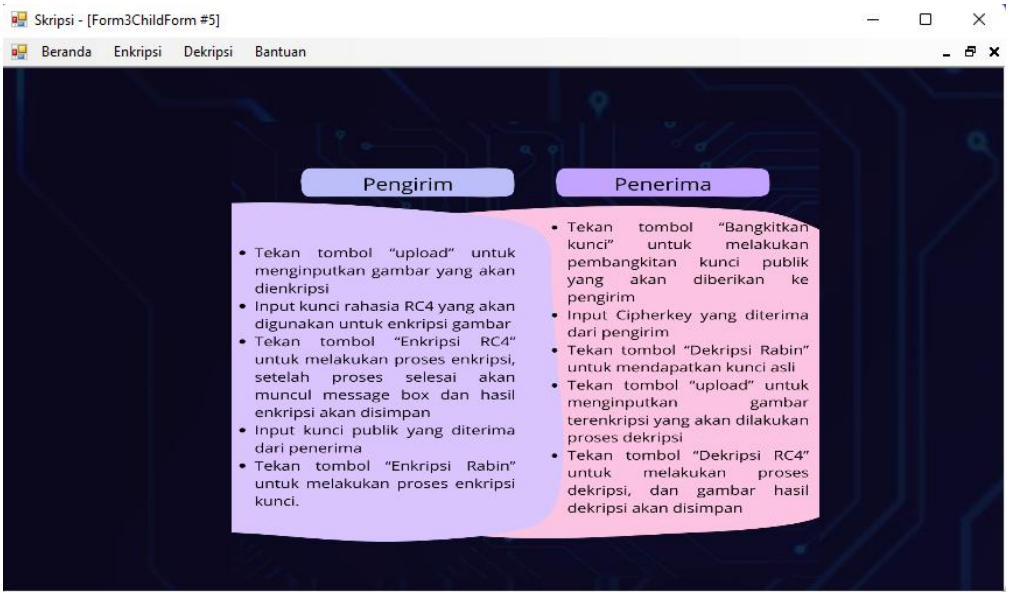


Gambar 4.3 Implementasi Halaman Dekripsi (Penerima)

Gambar diatas memperlihatkan halaman penerima, halaman penerima untuk melakukan proses pembangkitan kunci dan dekripsi *file* gambar. Pada halaman dekripsi terdapat proses lokasi *file* yang berfungsi memasukkan *file* yang telah dienkripsi dengan algoritma RC4. Terdapat *Textbox* untuk menampilkan nilai kunci publik yang berhasil dibangkitkan, terdapat juga *Textbox* untuk memasukkan kunci hasil enkripsi dengan algoritma *H-Rabin*, dan *Textbox* untuk menampilkan kunci asli hasil dekripsi.

4.1.4 Halaman Bantuan

Halaman ini akan tampil jika pengguna memilih menu bantuan pada strip menu. Tampilan halaman Bantuan ditunjukkan pada gambar 4.4.



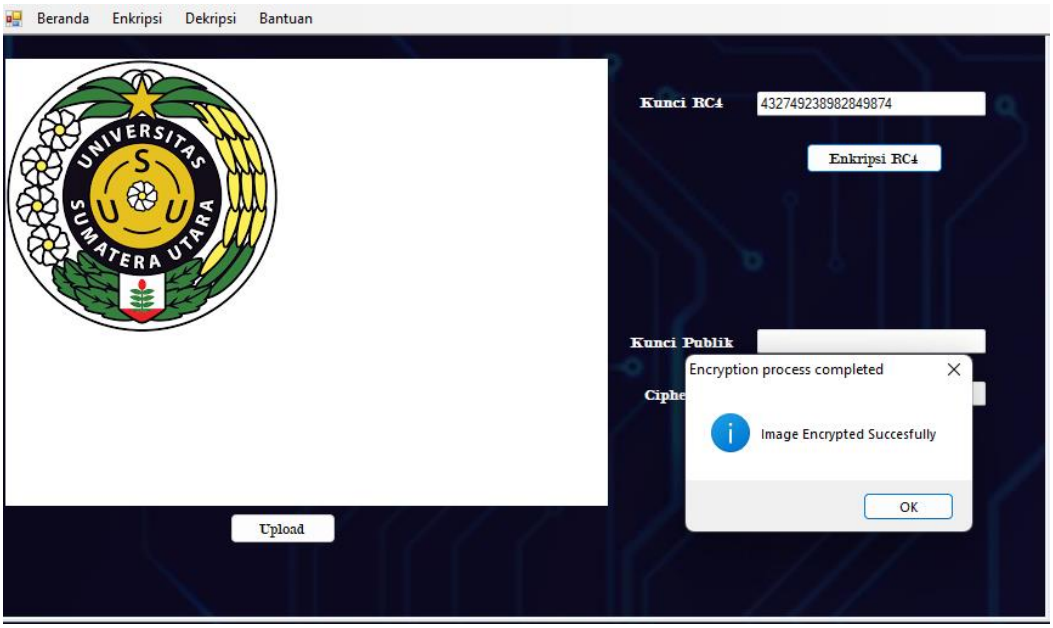
Gambar 4.4 Implementasi Halaman Bantuan

4.2 Pengujian Sistem

Tahapan ini dilakukan pengujan pada sistem untuk mengetahui apakah sistem ini berhasil melakukan enkripsi dengan algoritma *Rivest Code 4* (RC4) serta mengembalikan pesan itu kembali dengan melakukan proses dekripsi dengan algoritma *H-Rabin*.

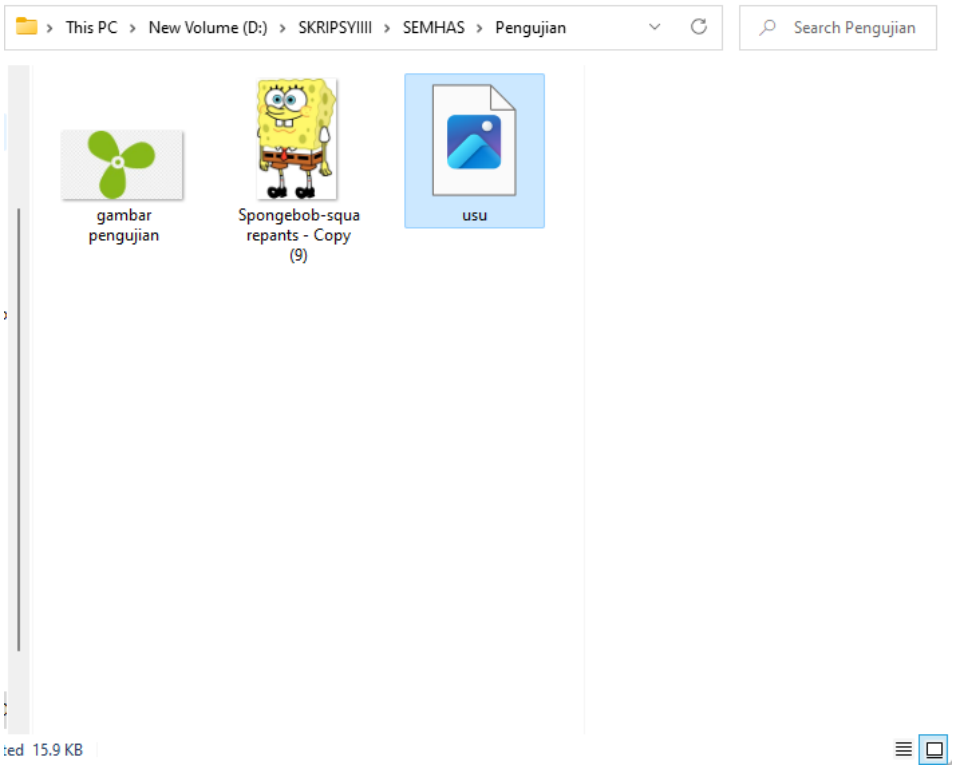
4.2.1 Pengujian Enkripsi

Tahapan ini dilakukan dengan menekan tombol "*Upload*" untuk memasukkan *file* gambar yang akan diuji, kemudian masukkan kunci RC4. Setalah proses enkripsi selesai gambar akan tersimpan. Setalah itu lakukan enkripsi kunci dengan menginputkan kunci publik yang diterima dari penerima.

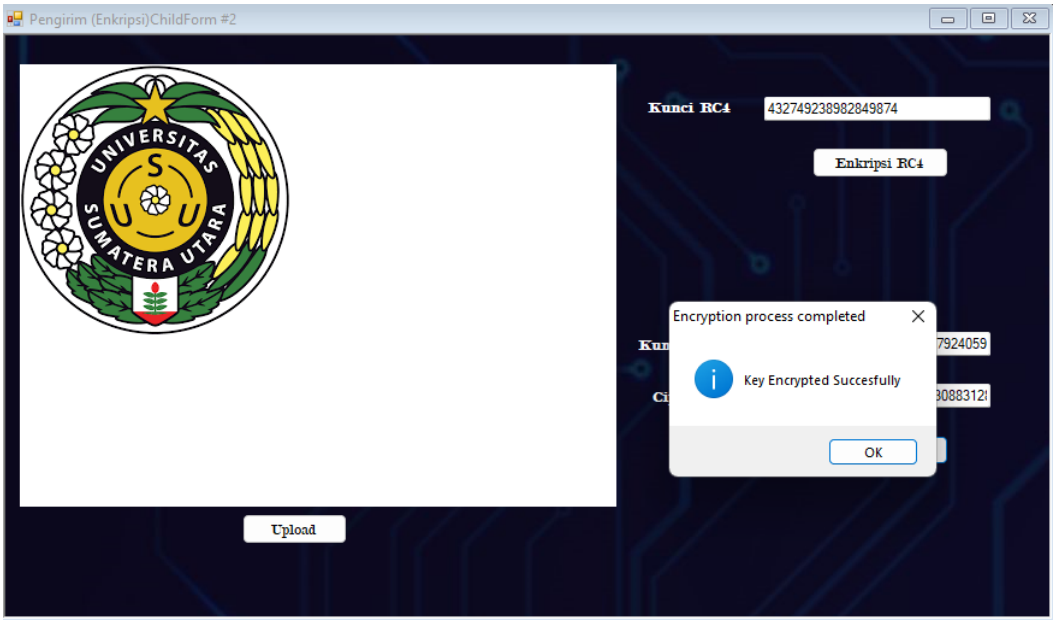


Gambar 4.5 Hasil Pengujian Enkripsi Gambar dengan Algoritma RC4

Pada Gambar 4.6 menunjukkan *output* enkripsi *file* gambar yang dilakukan dengan kunci rahasia milik pengguna, dimana hasil dekripsi akan tersimpan pada file direktori seperti pada gambar 4.6. Gambar hasil dekripsi tidak akan terbaca karena bit data pada gambar sudah terenkripsi.



Gambar 4.6 Gambar terenkripi terismpan pada direktori pengguna



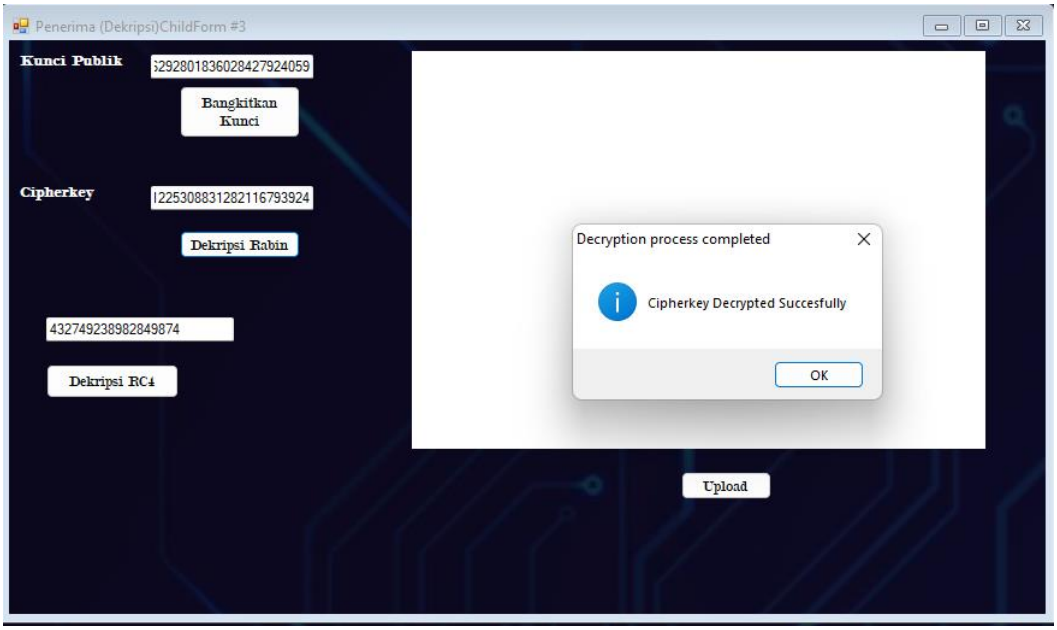
Gambar 4.7 Enkripsi Kunci RC4 dengan Algoritma *H-Rabin*

Pada gambar 4.7 adalah lanjutan proses enkripsi yaitu melakukan enkripsi terhadap kunci rahasia RC4, enkripsi dilakukan dengan kunci publik yang didapat dari penerima sehingga hasil enkripsi menampilkan *cipherkey*.

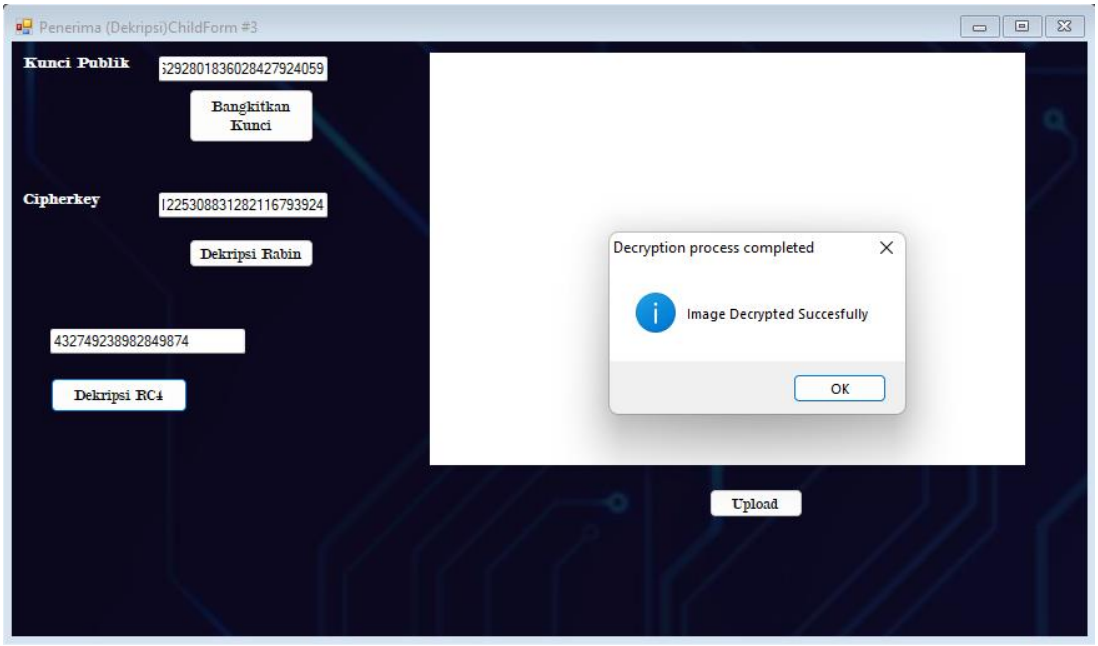
4.2.2 Pengujian Dekripsi

Tahapan ini dilakukan dengan menekan tombol "bangkitkan kunci" untuk melakukan proses pembangkitan kunci terlebih dahulu. Setelah itu masukkan *cipherkey* dari pengirim lalu tekan tombol enkripsi rabin untuk melakukan proses dekripsi kunci. Setelah kunci berhasil didekripsi, masukkan gambar hasil enkripsi untuk dilakukan dekripsi dengan kunci asli yang didapatkan sehinga mendapatkan gambar asli.

Pada gambar 4.8 terlihat hasil dekripsi *cipherkey* yang dilakukan dengan algoritma H-Rabin, dan menampilkan kunci asli RC4 yang akan digunakan untuk melakukan dekripsi pada *file* gambar yang terenkripsi.

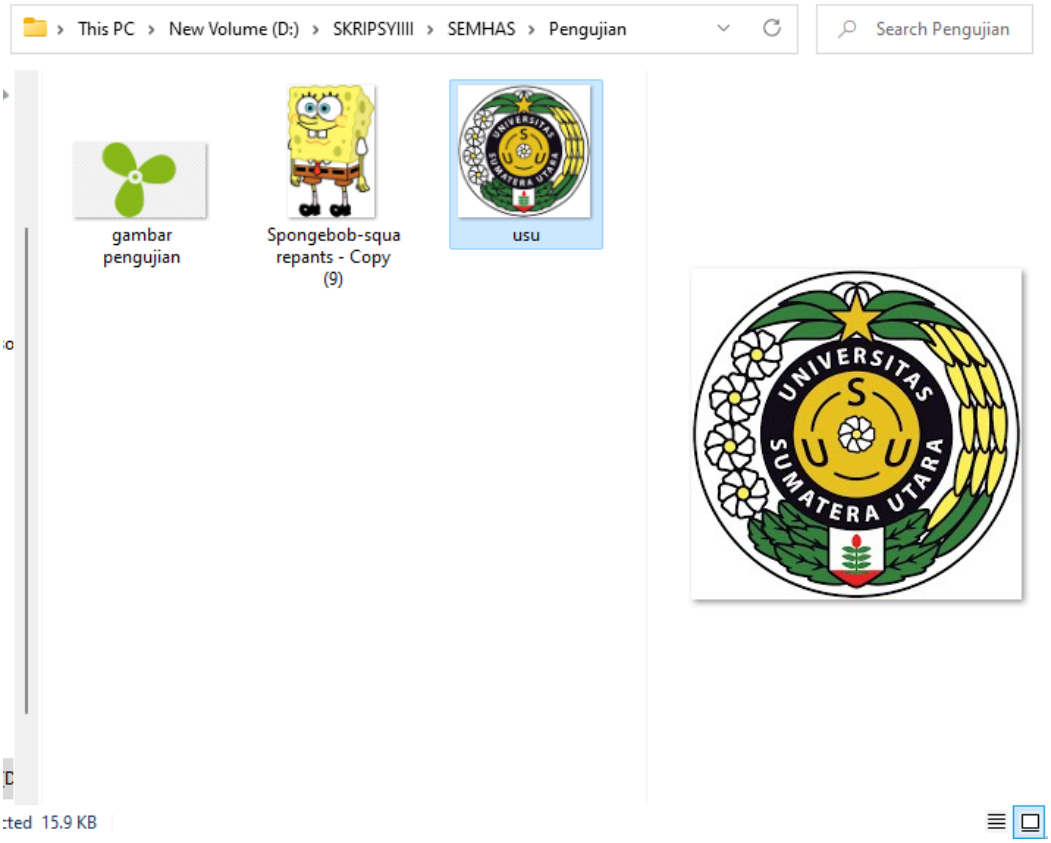


Gambar 4.8 Hasil Pengujian Dekripsi *Cipherkey* dengan Algoritma H-Rabin



Gambar 4.9 Hasil Pengujian Dekripsi Gambar dengan Algoritma RC4

Pada gambar 4.9 dilakukan dekripsi terhadap gambar dengan menggunakan kunci yang berhasil didapat dari hasil dekripsi sebelumnya, hasil dekripsi gambar akan disimpan pada direktori pengguna seperti pada gambar 4.10.



Gambar 4.10 Gambar Hasil Dekripsi Tersimpan Pada Direktori Pengguna

Dapat dilihat dari skema kerja pengujian sistem implementasi Algoritma *Rivest Code 4* (RC4) dan algoritma *H-Rabin* menunjukkan kesesuaian pesan asli dengan *plaintext* hasil dekripsi yang telah dilakukan. Proses tersebut terbukti bahwa kombinasi kedua algoritma *Rivest Code* 4 dan algoritma *H-Rabin* berhasil diimplemantasikan dan tidak merusak keutuhan data.

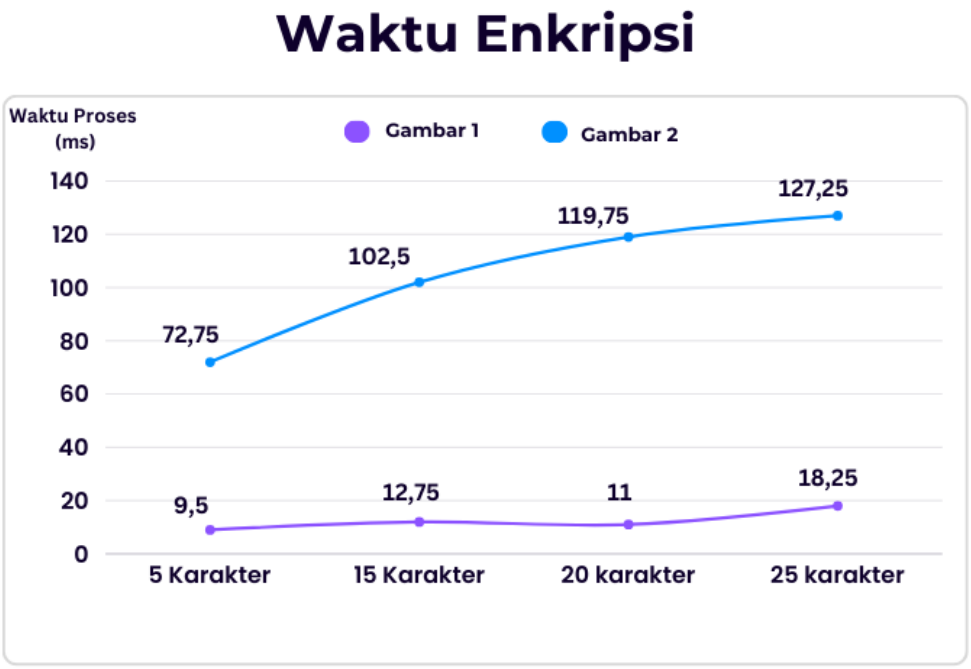
4.2.3 Pengujian Runnning Time Program

Tahapan ini dilakukan untuk mengetahui waktu yang diperlukan ketika proses enkripsi dan dekripsi pada setiap gambar dengan ukuran yang berbeda dan panjang kunci yang berbeda.

1. Pengujian Running time program berdasarkan panjang kunci dan bessar ukuran gambar pada enkripsi dan dekripsi RC4

Tabel 4.1 Waktu Enkripsi Program RC4

	Panjang Kunci	Uji 1 (ms)	Uji 2 (ms)	Uji 3 (ms)	Uji 4 (ms)	Rata-Rata
Gambar 1 (4,3 kb)	5	7	11	7	13	9,5
	15	13	13	12	13	12,75
	20	10	11	13	10	11
	25	41	13	11	8	18,25
Gambar 2 (5,98 mb)	5	86	92	89	84	72, 75
	15	86	96	128	100	102,5
	20	124	99	129	127	119, 75
	25	130	129	128	123	127, 5

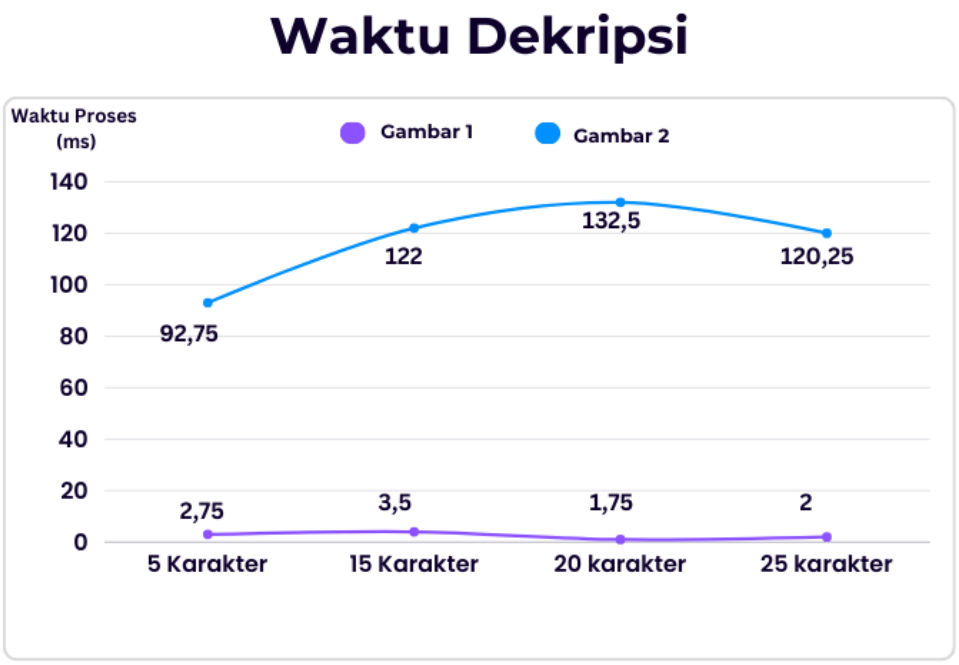


Gambar 4.11 Grafik Waktu Eksekusi Program Pada Gambar 1 (4,3kb)

Dari tabel dan gambar dapat disimpulkan panjang kunci pada proses enkripsi RC4 tidak terlalu mempengaruhi waktu proses, seperti pada gambar 1 dengan ukuran 4,3 kb perbedaan waktu hanya berbeda beberapa milidetik dengan panjang kunci yang berbeda 10 karakter.

Tabel 4.2 Waktu Dekripsi Program RC4

	Panjang Kunci	Uji 1 (ms)	Uji 2 (ms)	Uji 3 (ms)	Uji 4 (ms)	Rata-Rata
Gambar 1 (4,3 kb)	5	6	2	1	2	2, 75
	15	2	2	8	2	3, 5
	20	2	1	2	2	1, 75
	25	2	2	2	2	2
Gambar 2 (5,98 mb)	5	94	97	90	90	92, 75
	15	95	128	137	128	122
	20	146	129	125	130	132, 5
	25	130	103	120	128	120, 25



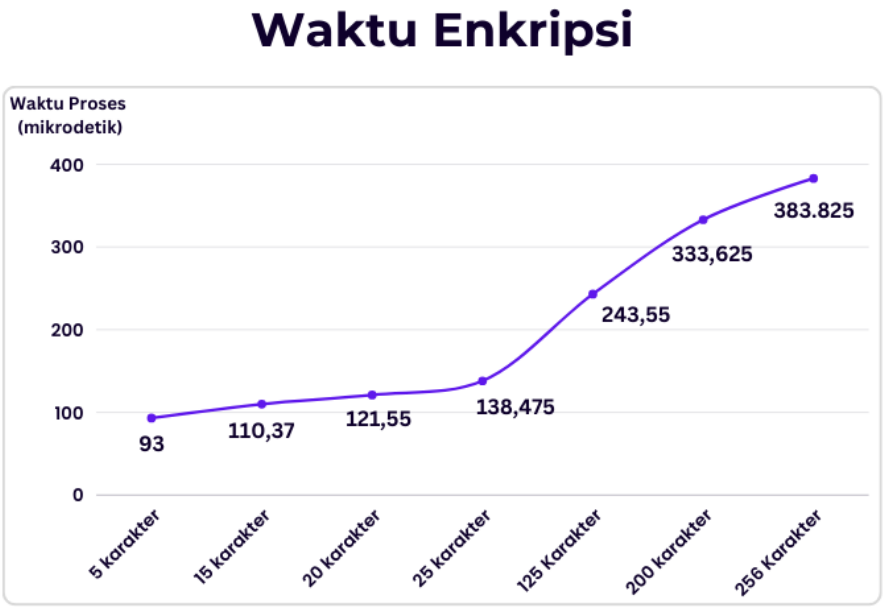
Gambar 4.12 Grafik Waktu Eksekusi Program Pada Gambar 2 (5,98mb)

Dari hasil pada uji program dapat disimpulkan bahwa ukuran gambar yang akan di proses mempengaruhi waktu yang ada terhadap proses enkripsi dan dekripsi. Jika ukuran gambar semakin besar maka semakin lama waktu yang diperlukan. Namun jika kunci yang diinputkan pengguna semakin panjang, maka waktu proses enkripsi dan dekripsi semakin cepat.

2. Pengujian Running time program berdasarkan panjang kunci dan bessar ukuran gambar pada enkripsi dan dekripsi H-Rabin

Tabel 4.3 Waktu Enkripsi Program H-Rabin

Panjang Kunci	Uji 1 (μs)	Uji 2 (μs)	Uji 3 (μs)	Uji 4 (μs)	Rata-Rata
5	94,7	98,3	90,4	88,6	93
15	112,2	114,4	102,4	112,5	110,37
20	126	113,8	117,2	129,2	121,55
25	125,9	126	163,2	138,8	138,475
125	237,5	240	253,8	242,9	243,55
200	335	333,8	330,5	335,2	333,625
256	389	384,8	373,9	387,6	383.825

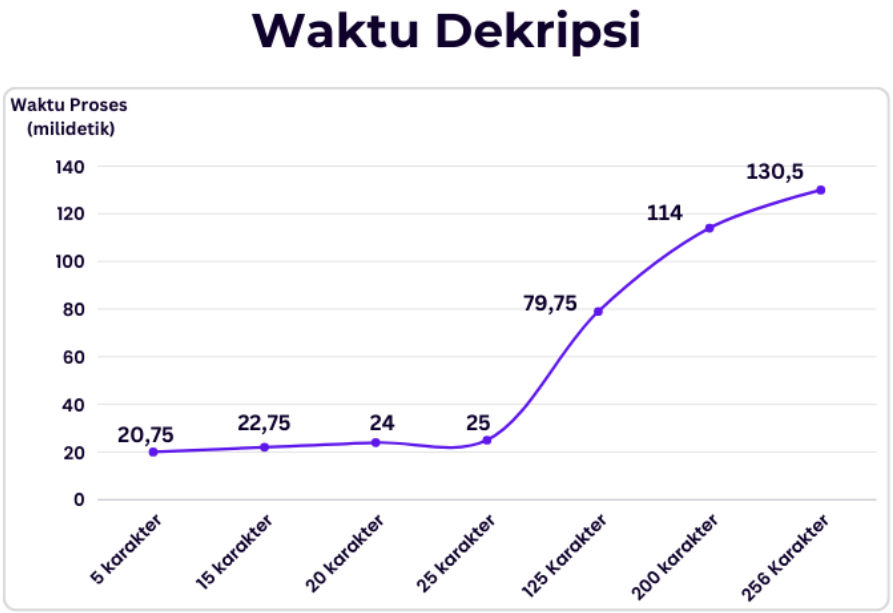


Gambar 4.13 Grafik Waktu Eksekusi Program Enkripsi H-Rabin

Dari hasil pengujian yang telah dilakukan disimpulkan bahwa waktu proses enkripsi jauh lebih cepat dibandingkan dengan waktu proses dekripsi yang dijelaskan pada tabel dan grafik dibawah. Pengujian pada enkripsi kunci dengan H-Rabin dilakukan dengan satuan mikrodetik. Disimpulkan juga bahwa semakin panjang kunci maka semakin panjang waktu yang dibutuhkan dalam melakukan proses enkripsi.

Tabel 4.4 Waktu Enkripsi H-Rabin

Panjang Kunci	Uji 1 (ms)	Uji 2 (ms)	Uji 3 (ms)	Uji 4 (ms)	Rata-Rata
5	20	22	22	19	20,75
15	23	24	22	22	22,75
20	25	24	23	24	24
25	25	26	24	25	25
125	78	82	79	80	79.75
200	113	108	115	120	114
256	127	132	140	123	130,5



Gambar 4.14 Grafik Waktu Eksekusi Program Dekripsi H-Rabin

Dari hasil uji program diatas dapat disimpulkan bahwa hasil dekripsi pada algoritma H-Rabin berkaitan dengan panjang karakter kunci, semakin panjang kunci semakin lama pula waktu yang dibutuhkan. Pengujian pada dekripsi kunci dengan algoritma H-Rabin dilakukan dengan menggunakan satuan milidetik pada proses dekripsi

4.2.4 Pengujian Runnning Time Program dengan Teori Carmen

Teori Cormen merupakan teori yang digunakan untuk memperkirakan dan menghitung kompleksitas waktu dalam menganalisis suatu algoritma dengan memanfaatkan bentuk tabel dan notasi θ (theta).

1. Kompleksitas Waktu RC4

Tabel 4.5 Kompleksitas Fungsi RC4

```
FUNCTION RC4(input, key)
    FOR i FROM 0 TO 255
        SET box[i] TO i
    END FOR
    FOR i FROM 0 TO 255
        SET j TO (key[i MOD
key.LENGTH] + box[i] + j) MOD 256
        SET x TO box[i]
        SET box[i] TO box[j]
        SET box[j] TO x
    END FOR

    SET j TO 0
    FOR i FROM 0 TO input.LENGTH - 1
        SET y TO i MOD 256
        SET j TO (box[y] + j) MOD 256
        SET x TO box[y]
        SET box[y] TO box[j]
        SET box[j] TO x
        SET result[i] TO input[i] XOR
box[(box[y] + box[j]) MOD 256]
    END FOR

    RETURN result
END FUNCTION
```

C	#	C#
C_1	n	C_1n
C_2	n	C_2n
C_1	n	C_1n
C_2	n	C_2n
C_2	n	C_2n
C_2	n	C_2n
C_2	n	C_2n
C_3	1	C_3
C_1	n-1	$C_1(n-1)$
C_2	n-1	$C_2(n-1)$
C_2	n-1	$C_2(n-1)$
C_2	n-1	$C_2(n-1)$
C_2	n-1	$C_2(n-1)$
C_2	n-1	$C_2(n-1)$
C_2	n-1	$C_2(n-1)$
C_4	1	C_4

$\therefore T(n) = 2C_1n + 5C_2n + C_3 + C_1n - C_1 + 6C_2n - C_2 + C_4$

$= (C_1 + 6C_2)n - C_1 - C_2 + 2C_1n + 5C_2n + C_4$

$= \theta(n)$ (linear)

2. Kompleksitas Enkripsi H-Rabin

Tabel 4.6 Kompleksitas Waktu Enkripsi H-Rabin

```
FUNCTION RabinEncrypt(plaintext, n)

    SET m TO plaintext * 257
    SET ciphertext TO m^2 MOD n

    RETURN ciphertext
END FUNCTION
```

C	#	C#
C_1	1	C_1
C_2	1	C_2
C_3	1	C_3

$\therefore T(n) = C_1 + C_2 + C_3$

$= \theta(1)$ (konstan)

3. Kompleksitas Waktu Algoritma *Generate Random BigInteger*

Tabel 4.7 Kompleksitas Waktu Generate Random BigInteger

```
FUNCTION
GenerateRandomBigInteger
    SET range TO max - min
    SET bytes TO new BYTE ARRAY
    CALL random.NextBytes(bytes)
    SET randomNumber TO new
BigInteger(bytes)
    IF randomNumber < 0 THEN
        SET randomNumber TO -
randomNumber
    END IF

    RETURN (randomNumber %
range) + min
END FUNCTION
```

C	#	C#
C_1	1	C_2
C_1	1	C_2
C_2	1	C_2
C_1	1	C_1
C_3	1	C_3
C_1	1	C_1
C_4	1	C_4

$\therefore T(n) = 4C_1 + C_2 + C_3 + C_4$

$= \theta(1)$ (konstan)

4. Kompleksitas Algoritma *Fermat's Little Fiere*

Tabel 4.8 Kompleksitas Waktu Algoritma *Fermat's Little Fiere*

```
FUNCTION IsFermatPrime
    IF number <= 1 OR number == 4 THEN
        RETURN false
    END IF

    IF number <= 3 THEN
        RETURN true
    END IF

    FOR i FROM 0 TO iterations - 1 DO
        SET a TO 2 + RANDOM() %
(number - 3)
        IF GCD(a, number) != 1 THEN
            RETURN false
        END IF

        IF ModularExponentiation(a,
number - 1, number) != 1 THEN
            RETURN false
        END IF
    END FOR
    RETURN true
END FUNCTION
```

C	#	C#
C_1	1	C_1
C_2	1	C_2
C_1	1	C_1
C_2	1	C_2
C_3	n-1	$C_3(n-1)$
C_4	n-1	$C_4(n-1)$
C_1	n-1	$C_1(n-1)$
C_2	n-1	$C_2(n-1)$
C_1	n-1.n	$C_1(n-1)$
C_2	n-1.n	$C_2(n-1)$
C_2	1	C_2

$\therefore T(n) = 2C_1 + 3C_2 + 2\ C_1n - C_1 + 2C_2n - C_2 + C_3n - C_3 + C_4n - C_4$

$= (2C_1 + 2C_2 + C3 + C_4)n.n - C_1 - C_2 - C3 - C_4 + 2C_1 + 3C_2$

$= \theta(n^2)$ (kuadratik)

5. Kompleksitas Waktu Algortima *Generate Random Prime*

Tabel 4.9 Kompleksitas Waktu Genrate Random Prime

```
FUNCTION GenerateRandomPrime
    WHILE true DO
        SET randomNumber TO
GenerateRandomBigInteger(min, max)

        IF IsFermatPrime(randomNumber,
10) AND randomNumber % 4 == 3 THEN
            RETURN randomNumber
        END IF
    END WHILE
END FUNCTION
```

C	#	C#
C_1	n	C_1n
C_2	n	C_2n
C_3	n	C_3n
C_2	n	C_2n

$\therefore T(n) = C_1n + C_2n + C_3n + C_2n$

$= \theta(n)$ (konstan)

6. Kompleksitas Waktu Algoritma Modular Exponentiation

Tabel 4.10 Kompleksitas Waktu Algoritma Modular Exponentiation

```
FUNCTION ModularExponentiation
    IF modulus == 1 THEN
        RETURN 0
    END IF
    SET result TO 1
    SET baseNumber TO baseNumber % modulus
    WHILE exponent > 0 DO
        IF exponent % 2 == 1 THEN
            SET result TO (result *
baseNumber) % modulus
        END IF
        SET exponent TO exponent >> 1
        SET baseNumber TO (baseNumber *
baseNumber) % modulus
    END WHILE
    RETURN result
END FUNCTION
```

C	#	C#
C_1	1	C_1
C_2	1	C_2
C_3	1	C_{31}
C_4	1	C_4
C_5	n	C_5n
C_1	n	C_1n
C_2	n	C_2n
C_4	n	C_4n
C_4	n	n C_4
C_3	1	C_3

$\therefore T(n) = C_1 + C_2 + 2C_3 + C_5 + C_1n + 3C_3n + C_4n$

$= \theta(n)$ (Linear)

7. Kompleksitas Algoritma *ModInverse*

Tabel 4.11 Kompleksitas Waktu Algoritma *ModInverse*

```
FUNCTION ModInverse(a, m)
    SET m0 TO m
    SET y TO 0
    SET x TO 1

    IF m == 1 THEN
        RETURN 0
    END IF

    WHILE a > 1 DO
        SET t TO m
        SET m TO a % m
        SET a TO t
        SET t TO y
        SET y TO x - q * y
        SET x TO t
    END WHILE

    IF x < 0 THEN
        x = x + m0
    END IF

    RETURN x
END FUNCTION
```

C	#	C#
C_1	1	C_1
C_1	1	C_1
C_1	1	C_1
C_2	1	C_2
C_3	1	C_3
C_4	n	C_4n
C_1	n	C_1n
C_1	n	C_1n
C_1	n	C_1n
C_1	n	C_1n
C_1	n	C_1n
C_1	n	C_1n
C_2	1	C_2
C_5	1	C_5
C_3	1	C_3

$\therefore T(n) = 3C_1 + 2C_2 + 2C_3 + C_5 + C_4n + 6C_1n$

$= \theta(n)$ (Linear)

8. Kompleksitas Waktu Dekripsi H-Rabin

Tabel 4.12 Kompleksots Waktu Dekripsi H-Rabin

```
FUNCTION Decrypt(ciphertext)
    SET P
    SET q
    SET r
    SET n = pxqxr
     SET mp TO
ModularExponentiation(ciphertext, (p + 1)
/ 4, p)
     SET mq TO
ModularExponentiation(ciphertext, (q + 1)
/ 4, q)
     SET mr TO
ModularExponentiation(ciphertext, (r + 1)
/ 4, r)


     SET pmp TO mp % p
     SET mmp TO ((-mp % p) + p) % p
     SET pmq TO mq % q
     SET mmq TO ((-mq % q) + q) % q
     SET pmr TO mr % r
     SET mmr TO ((-mr % r) + r) % r


     SET b1 TO ModInverse(n / p, p)
     SET b2 TO ModInverse(n / q, q)
     SET b3 TO ModInverse(n / r, r)


     SET X1 TO (pmp * b1 * n / p + pmq
* b2 * n / q + pmr * b3 * n / r) % n
     SET X2 TO (mmp * b1 * n / p + pmq
* b2 * n / q + pmr * b3 * n / r) % n
     SET X3 TO (pmp * b1 * n / p + mmq
* b2 * n / q + pmr * b3 * n / r) % n
     SET X4 TO (pmp * b1 * n / p + pmq
* b2 * n / q + mmr * b3 * n / r) % n
```

C	#	C#
C_1	1	C_1
C_1	1	C_1
C_1	1	C_1
C_1	1	C_1
C_2	1	C_2
C_2	1	C_2
C_2	1	C_2
C_3	1	C_3
C_3	1	C_3
C_3	1	C_3
C_3	1	C_3
C_3	1	C_3
C_3	1	C_3
C_4	1	C_4
C_4	1	C_4
C_4	1	C_4
C_5	1	C_5
C_5	1	C_5
C_5	1	C_5
C_5	1	C_5

```
    SET X5 TO n - X1
    SET X6 TO n - X2
    SET X7 TO n - X3
    SET X8 TO n - X4

    SET X9 – X16 To Mod 257

    IF X1 % 257 == 0 THEN
        SET trueValue TO X1 / 257
    ELSE IF X2 % 257 == 0 THEN
        SET trueValue TO X2 / 257
    ELSE IF X3 % 257 == 0 THEN
        SET trueValue TO X3 / 257
    ELSE IF X4 % 257 == 0 THEN
        SET trueValue TO X4 / 257
    ELSE IF X5 % 257 == 0 THEN
        SET trueValue TO X5 / 257
    ELSE IF X6 % 257 == 0 THEN
        SET trueValue TO X6 / 257
    ELSE IF X7 % 257 == 0 THEN
        SET trueValue TO X7 / 257
    ELSE IF X8 % 257 == 0 THEN
        SET trueValue TO X8 / 257
    END IF

    RETURN {trueValue}
END FUNCTION
```

C_5	1	C_5
C_5	1	C_5
C_5	1	C_5
C_5	1	C_5
C_6	1	C_6
C_7	1	C_7
C_8	1	C_8
C_7	1	C_7
C_8	1	C_8
C_7	1	C_7
C_8	1	C_8
C_7	1	C_7
C_8	1	C_8
C_7	1	C_7
C_8	1	C_8
C_7	1	C_7
C_8	1	C_8
C_7	1	C_7
C_8	1	C_8
C_7	1	C_7
C_8		C_8
C_9	1	C_9

$\therefore T(n) = 4C_1 + 3C_2 + 6C_3 + 3C_4 + 8C_5 + C6 + 8C_7 + 8C_8 + C_9$

$= \theta(1)$ (konstan)

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil analisis serta pengujian terhadap implementasi yang telah dilakukan pada penelitian *Hybrid Cryptosystem* dengan algoritma *Rivest Code 4 (RC4)* serta algoritma *H-Rabin* dalam pengamanan gambar disimpulkan bahwa:

1. Mengkombinasikan dua buah algoritma yaitu algoritma simetris dan asimetris berhasil melakukan proses kriptografi yaitu *key generation*, serta terbukti dapat melakukan proses enkripsi dan dekripsi pada file gambar.
2. Algoritma *Rivest Code 4* (RC4) berhasil melakukan enkripsi dan dekripsi pada bit gambar sehingga *file* gambar yang berhasil di enkripsi tidak dapat terbaca.
3. Kunci publik hasil dibangkitkan oleh penerima memiliki panjang lebih dari 2048 bit.
4. Algoritma RC4 memiliki kompleksitas waktu linear ($\theta(n)$), sedangkan Algoritma H-Rabin memiliki kompleksitas waktu konstan ($\theta(1)$).
5. Algoritma pembangkitan kunci seperti random bilangan besar, random bilangan prima, dan cek keprimaan bilangan memiliki kompleksitas waktu konstan ($\theta(1)$) dan linear ($\theta(n)$).
6. Proses enkripsi dan dekripsi pada RC4 tidak berpengaruh pada panjang kunci melainkan pada besar ukuran file gambar, jika ukuran gambar semakin besar maka semakin lama waktu yang dibutuhkan.

5.2 Saran

Saran untuk penelitian selanjutnya yaitu:

1. Diharapkan pada pengembangan selanjutnya dapat membangun aplikasi ini pada *platform* selain desktop, misalnya: *Android, iOS, Web*, dll.
2. Disarankan untuk melakukan penelitian lebih lanjut, mengapa jika kunci yang diinputkan lebih panjang, maka proses enkripsi ataupun dekripsi semakin cepat.
3. Disarankan penelitian selanjutnya untuk dapat menerapkan penggunaan kunci berupa password yaitu terdiri dari angka, huruf, dan karakter unik.

DAFTAR PUSTAKA

Andara, H., Fera, D., & Khairunnisa. (2022). Implementasi Kriptografi Hybrida Algoritma RSA dan Vernam Cipher Dalam Pengamanan File Text. *ALGORITMA: Jurnal Ilmu Komputer dan Informatika*, 8-15.

Basri. (2016). Kriptografi Simetris dan Asimetris Dalam Perspektif Keamanan Data dan Kompleksitas Komputasi. *Jurnal Ilmiah Ilmu Komputer*, 16-23.

Hamsyar, A., & Basri, M. (2022). Aplikasi Enkripsi Gambar Menggunakan Metode (Rivest Shamir Adleman) RSA. *JURNAL SINTAKS LOGIKA*, 39-45.

Hutapea, D. (2021). Implementasi Algoritma Kriptografi Rabin Dan Lempel-Ziv-Welch (Lzw) Dalam Pengamanan Dan Kompresi File Citra. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 213-220.

Jamaludin, J., & R, R. (2020). Rancang Bangun Pengamananan Teks Menggunakan Kombinasi Vigenere Cipher dan RSA dalam Hybrid Cryptosystem. *Prosiding Seminar Nasional Riset Dan Information Science (SENARIS) 2020*, 105-116.

Maulana, R., & R. Mahdalena, S. (2021). Implementasi Kriptografii Untuk Pengamanan Data Pribadi Siswa SMA Swasta Jaya Krama Beringin dengan Algoritma RC4. *Jurnal Nasional Komputasi dan Teknologi Informasi*, 377-383.

Mohammed, R., & Lahieb, M. J. (2020). Secure Image Encryption Scheme Using Chaotic Maps and RC4 Algorithm. *Solid State Technology*, 3449-3465.

Putra, N. B., & al, e. (2023). Analisis Enkripsi Kriptografi Asimetri Algoritma RSA Berbaasis Pemrograman Batch Pada Media Flashdik. *Jurnal Riset Sistem Informasi dan Teknik Informatika (JURASIK)*, 142-154.

Rachmawati, D., & Mohammad, A. B. (2018). A Cryptocompression System based on H-Rabin Public Key Encryption Algorithm and Elias Gamma Codes. *nternational Conference of Science, Technology, Engineering, Environmental and Ramification Researches (ICOSTEERR)*, 1910-1914.

Rachmawati, D., Amalia, & Triska, H. I. (2020). File Cryptocompression By Using H-Rabin and Golomb Rice Algorithm. *Journal of Theoretical and Applied Information Technology*, 1831-1841.

Ramadani, S., Diana, & Sauda, S. (2020). Penerapan Algoritma AES dan DSA Menggunakan Hybrid Cryptosystem untuk. *JURIKOM (Jurnal Riset Komputer)*, 523-529.

Siahaan, K. N., & Mesran. (2020). Penerapan Algoritma Venigmare Cipher dan Vernam Cipher Dalam Pengamanan Data Teks. *Jurnal Sistem Komputer dan Informatika (JSON)* , 48-52.

Suhandinata, S. e. (2019). Analisis Performa Kriptografi Hybrid Algoritma Blowfish dan Algoritma RSA. *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, 1-10.

Sulaiman, O. K. (2019). Hybrid Cryptosystem Menggunakan XOR Cipher dan Merkle–Hellman Knapsack untuk Menjaga Kerahasiaan Pesan Digital. *Jurnal Teknologi Informasi*, 169-173.

Suseno, A. Y., Sulitiyowati, N., & Purwantoro. (2021). Analisis Peningkatan Hybrid Cryptosystem Untuk Enkripsi dan Dekripsi Menggunakan Vigenere Cipher dan RSA Pada Text. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)* , 142-148.