# KRIPTANALISIS ALGORITMA M.S.H. BISWAS (RABIN-BISWAS) MENGGUNAKAN ALGORITMA *BABY STEP GIANT STEP* DAN METODE *CIPHERTEXT ONLY ATTACK*

**SKRIPSI**

**THORIQ AUFAR NUBLI**

**20140170**

**PROGRAM STUDI S-1 ILMU KOMPUTER**

**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

**UNIVERSITAS SUMATERA UTARA**

**MEDAN**

**2024**

# KRIPTANALISIS ALGORITMA M.S.H. BISWAS (RABIN-BISWAS) MENGGUNAKAN ALGORITMA *BABY STEP GIANT STEP* DAN METODE *CIPHERTEXT ONLY ATTACK*

**SKRIPSI**

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

**THORIQ AUFAR NUBLI**

**20140170**

**PROGRAM STUDI S-1 ILMU KOMPUTER**

**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

**UNIVERSITAS SUMATERA UTARA**

**MEDAN**

**2024**

## PERSETUJUAN

| | |
|---|---|
| Judul | : KRIPTANALISIS ALGORITMA M.S.H. BISWAS (RABIN-BISWAS) MENGGUNAKAN ALGORITMA *BABY STEP GIANT STEP* DAN METODE *CIPHER-TEXT ONLY ATTACK* |
| Kategori | : SKRIPSI |
| Nama | : THORIQ AUFAR NUBLI |
| Nomor Induk Mahasiswa | : 201401070 |
| Program Studi | : SARJANA (S1) ILMU KOMPUTER |
| Fakultas | : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA |
| Komisi Pembimbing | : |

Pembimbing 2

Fauzan Nurahmadi S.Kom., M.Cs
NIP. 198512292018051001

Pembimbing 1

Dr. Muhammad Anggi Budiman S.T., M.Comp.Sc., M.E.M.
NIP. 197510082008011011

Diketahui/disetujui oleh
Program Sarjana S1 Ilmu Komputer
Ketua,

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

# PERNYATAAN

KRIPTANALISIS ALGORITMA M.S.H. BISWAS (RABIN-BISWAS)
MENGGUNAKAN ALGORITMA *BABY STEP GIANT STEP*
DAN METODE *CIPHERTEXT ONLY ATTACK*

SKRIPSI

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

Medan, 27 November 2023

Thoriq Aufar Nubli
201401070

# PENGHARGAAN

*Bissmiillahirrahmannirrahiim*. *Alhamdulillahirabbil'alamin*, puji syukur penulis panjatkan ke hadirat Allah *Subhanahu Wa Ta'ala*, karena atas rahmat dan hidayah-Nya, penulisan laporam tuhas akhir ini dapat terselesaikan dengan baik. Dengan rasa syukur kepada Allah, Yang Maha Pengasih dan Maha Penyayang, yang telah memberikan penulis kekuatan dan petunjuk selama perjalanan penulisan skripsi ini.

Dengan rendah hati, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan banyak bantuan, meluangkan waktunya dan memberikan dukungan serta doa selama penulis menjalani proses penyusunan tugas akhir ini, yaitu:

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.
3. Bapak Dr. Mohammad Andri Budiman S.T., M.Comp.Sc., M.E.M. selaku Wakil Dekan 1 Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Dan selaku Dosen Pembimbing 1 yang memberikan banyak ilmu baru, masukan, saran, serta motivasi, dan banyak meluangkan waktunya untuk membimbing penulis dalam pengerjaan skripsi ini.
4. Ibu Dr. Amalia, ST., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara dan juga selaku Dosen Pembanding 2 yang telah memberikan masukan dan saran yang membangun kepada penulis.
5. Bapak Fauzan Nurahmadi S.Kom., M.Cs. selaku Dosen Pembimbing 2 yang selalu memberikan bimbingan, saran, masukan, serta motivasi kepada penulis selama penulisan skripsi ini.
6. Bapak Dr. Jos Timanta Tarigan S.Kom., M.Sc. selaku Dosen Pembanding 1 yang telah memberikan masukan dan saran yang membangun kepada penulis.
7. Seluruh Bapak dan Ibu Dosen beserta Staf Pegawai Program S-1 Ilmu Komputer, Fakultas Ilmu Komputer Teknologi Informasi Universitas Sumatera Utara.

8. Kedua orang tua tercinta, Ayahanda Salman S.T. dan Ibunda Mulia Pane A.Md. yang senantiasa mendo'akan penulis, memberikan dukungan dan semangat, serta kasih sayang yang tiada hentinya.
9. Sahabat-sahabat sepermainan 'Kaum Mujahiruddin' yang selalu memberi tawa dan candaan setiap harinya selama masa perkuliahan penulis.
10. Teman-teman stambuk 2020 khususnya kom B yang telah memberikan pengalaman belajar bersama yang berharga kepada penulis.
11. Setiap pihak yang telah memberikan bantuan penulis dalam pengarjaan laporan tugas akhir ini baik melalui dukungan langsung maupun tidak langsung yang sulit disebutkan satu per satu.

Semoga Allah SWT meridhoi semua pihak yang telah membantu penulis menyelesaikan laporan akhir ini dengan memberikan semangat, motivasi, perhatian, dan dukungan kepada penulis. Semoga mereka yang telah berkontribusi senantiasa menerima kasih, keberkahan, dan rahmat-Nya.

Medan, 27 November 2023

Penulis,

Thoriq Aufar Nubli

# ABSTRAK

Penyampaian informasi melalui internet rentan akan terjadinya penyadapan dikarenakan tingkat keamanannya yang relatif rendah. Untuk itu diperlukan suatu teknik khusus untuk mengamankan informasi yang hendak ditransmisikan melalui internet. Salah satu metode yang bisa dimanfaatkan yakni kriptografi. Namun, dapat mengamankan informasi saja tidak cukup bagi sebuah algoritma kriptografi. Sebuah algoritma kriptografi harus tahan terhadap serangan terhadap kriptografi guna tetap menjaga kerahasiaan informasi. Untuk itu, pada penelitian ini dilakukan kriptanalisis pada sebuah algoritma kriptografi yang bernama M.S.H. Biswas (Rabin-Biswas). Metode yang digunakan untuk mengkriptanalisis algoritma Rabin-Biswas yakni algoritma *Baby Step Giant Step* dan *ciphertext only attack*. Algoritma *Baby Step Giant Step* dapat memecahkan masalah logaritma diskrit yang ada pada algoritma Rabin-Biswas kurang dari 1 detik ketika kunci yang digunakan sebesar 36 bit. Algoritma Rabin-Biswas memiliki kelemahan utama pada proses enkripsinya. Ketika kunci yang digunakan untuk proses enkripsi $K > m^2$. Maka hanya membutuhkan *ciphertext*-nya saja untuk mendekripsinya.

**Kata Kunci:** Kriptografi, Kriptanalisis, M.S.H. Biswas (Rabin-Biswas), *Baby Step Giant Step*, *Ciphertext Only Attack.*

# ABSTRACT

Transmitting information via the internet is vulnerable to eavesdropping due to its relatively low level of security. Due to this circumstance, a unique method is necessary to protect the information intended for transmission over the internet. One technique that can be used is cryptography. However, just being able to secure information is not enough for a cryptographic algorithm. A cryptographic algorithm must be resistant to cryptographic attacks to uphold the secrecy of the information. For this reason, in this research cryptanalysis was carried out on a cryptographic algorithm called M.S.H. Biswas (Rabin-Biswas). The methods used to cryptanalyze the Rabin-Biswas algorithm are the Baby Step Giant Step algorithm and the ciphertext only attack. The algorithm known as Baby Step Giant Step is capable of addressing the discrete logarithm problem in the Rabin-Biswas algorithm in less than 1 second when the key used is 36 bits. The Rabin-Biswas algorithm has a major weakness in the encryption process. When the key used for the encryption process is $K > m^2$. So you only need the ciphertext to decrypt it.

**Keywords:** Cryptography, Cryptanalysis, M.S.H. Biswas (Rabin-Biswas), BabyStep Giant Step, Ciphertext Only Attack.

# DAFTAR ISI

PERSETUJUAN ..... ii
PERNYATAAN ..... iii
PENGHARGAAN ..... iv
ABSTRAK ..... vi
ABSTRACT ..... vii
DAFTAR ISI ..... viii
DAFTAR TABEL ..... x
DAFTAR GAMBAR ..... xi
BAB 1 PENDAHULUAN ..... 1
1.1 Latar Belakang ..... 1
1.2 Rumusan Masalah ..... 2
1.3 Batasan Penelitian ..... 3
1.4 Tujuan Penelitian ..... 3
1.5 Manfaat Penelitian ..... 3
1.6 Metode Penelitian ..... 4
1.7 Sistematika Penulisan ..... 5
BAB 2 LANDASAN TEORI ..... 6
2.1 Kriptografi ..... 6
2.2 Kriptosistem Kunci Publik ..... 7
2.3 Uji Keprimaan Algoritma AKS ..... 8
2.4 Uji Keprimaan Algoritma Fermat ..... 10
2.5 Algoritma RSA ..... 12
2.6 Algoritma Rabin ..... 13
2.7 Algoritma $AA_{\beta}$ ..... 13
2.8 Algoritma Rabin-*p* ..... 14
2.9 Algoritma M.S.H. Biswas (Rabin-Biswas) ..... 15
2.10 Kriptanalisis ..... 18
2.11 Serangan terhadap Kriptografi ..... 19

2.12 Algoritma *Baby Step Giant Step* ..... 20
2.13 Penelitian Relevan ..... 23
BAB 3 ANALISIS DAN PERANCANGAN ..... 25
3.1 Analisis ..... 25
3.1.1 Analisis Masalah ..... 25
3.1.2 Analisis Kebutuhan ..... 25
3.2 Perancangan Sistem ..... 27
3.2.1 Diagram Umum ..... 28
3.2.2 *Use Case Diagram* ..... 31
3.2.3 *Activity Diagram* ..... 33
3.2.4 *Sequence Diagram* ..... 37
3.2.5 Diagram alir (*flowchart*) ..... 41
BAB 4 IMPLEMENTASI DAN PENGUJIAN ..... 52
4.1 Implementasi Sistem ..... 52
4.1.1 Laman *Home* ..... 52
4.1.2 Laman Pembangkitan Kunci ..... 53
4.1.3 Laman Enkripsi ..... 53
4.1.4 Laman Dekripsi ..... 54
4.2 Pengujian Sistem ..... 55
4.2.1 Pengujian Pembangkitan Kunci ..... 55
4.2.2 Pengujian Enkripsi ..... 58
4.2.3 Pengujian Dekripsi ..... 60
4.2.4 Perhitungan Manual ..... 62
4.3 Kriptanalisis ..... 64
4.3.1 *Baby Step Giant Step* ..... 64
4.3.2 *Ciphertext Only Attack* ..... 66
4.3.3 *Brute Force Attack* ..... 66
BAB 5 PENUTUP ..... 68
5.1 Kesimpulan ..... 68
5.2 Saran ..... 69
DAFTAR PUSTAKA ..... 70

# DAFTAR TABEL

Tabel 2.1 Proses Pembangkitan Kunci ..... 17
Tabel 2.2 Proses Enkripsi M.S.H Biswas (Rabin-Biswas) ..... 17
Tabel 2.3 Proses Dekripsi M.S.H. Biswas (Rabin-Biswas) ..... 18
Tabel 2.4 Pasangan j dan aj ..... 22
Tabel 4.1 Waktu Eksekusi Algoritma AKS ..... 56
Tabel 4.2 Waktu Eksekusi Algoritma Fermat ..... 57
Tabel 4.3 Waktu Ekesekusi Enkripsi Algoritma Rabin-Biswas ..... 59
Tabel 4.4 Waktu Ekseksi Dekripsi Algoritma Rabin-Biswas ..... 61
Tabel 4.5 Perhitungan pada Pembangkitan Kunci ..... 62
Tabel 4.6 Perhitungan pada Enkripsi Pesan ..... 63
Tabel 4.7 Perhitungan pada Dekripsi Pesan ..... 64
Tabel 4.8 Waktu Eksekusi *Baby Step Giant Step* ..... 65
Tabel 4.9 Waktu Eksekusi *Brute Force Attack* ..... 67

# DAFTAR GAMBAR

Gambar 2.1 Proses Enkripsi dan Dekripsi ..... 7
Gambar 2.2 Proses pembangkitan kunci menggunakan Diffie-Hellman Key Exchange Protocol ..... 7
Gambar 3.1 Diagram Umum Pembangkitan Kunci ..... 28
Gambar 3.2 Diagram Umum Enkripsi dan Dekripsi ..... 30
Gambar 3.3 Diagram Umum Kriptnalisis ..... 31
Gambar 3.4 Use Case Diagram ..... 32
Gambar 3.5 *Activity Diagram* Pembangkitan Kunci ..... 34
Gambar 3.6 *Activity Diagram* Enkripsi ..... 35
Gambar 3.7 *Activity Diagram* Dekripsi ..... 36
Gambar 3.8 *Activity Diagram* Kriptanalisis ..... 37
Gambar 3.9 *Sequence Diagram* Pembangkitan Kunci ..... 38
Gambar 3.10 *Acitivity Diagram* Enkripsi ..... 39
Gambar 3.11 *Sequence Diagram* Dekripsi ..... 40
Gambar 3.12 Sequence Diagram Kriptanalisis ..... 41
Gambar 3.13 Diagram Alir Pembangkitan Kunci (*Sender*) ..... 43
Gambar 3.14 Diagram Alir Pembangkitan Kunci (*Recipient*) ..... 45
Gambar 3.15 Diagram Alir Enkripsi ..... 46
Gambar 3.16 Diagram Alir Dekripsi ..... 47
Gambar 3.17 Diagram Alir Kriptanalisis ..... 48
Gambar 3.18 Diagram Alir *Baby Step Giant Step Algorithm* ..... 50
Gambar 3.19 Diagram Alir *Get Secret Key* ..... 51
Gambar 4.1 Laman *Home* ..... 52
Gambar 4.2 Laman Pembangkitan Kunci ..... 53
Gambar 4.3 Laman Enkripsi ..... 54
Gambar 4.4 Laman Dekripsi ..... 54
Gambar 4.5 Pembangkitan Kunci (Pengirim) ..... 55
Gambar 4.6 Pembangkitan Kunci (Penerima) ..... 56
Gambar 4.7 Grafik Perbandingan Interval dengan Waktu Eksekusi (AKS) ..... 57

Gambar 4.8 Grafik Perbandingan Interval dengan Waktu Eksekusi (Fermat) ..... 58
Gambar 4.9 Enkripsi Pesan ..... 59
Gambar 4.10 Grafik Perbandingan Panjang Karakter dengan Waktu Eksekusi (Enkripsi) ..... 60
Gambar 4.11 Dekripsi Pesan ..... 61
Gambar 4.12 Grafik Perbandingan Panjang Karakter dengan Waktu Eksekusi (Dekripsi) ..... 62
Gambar 4.13 Grafik Perbandingan Panjang Bit dengan Waktu Eksekusi (*Baby-Step Giant-Step*) ..... 65
Gambar 4.14 Grafik Perbandingan Panjang Bit dengan Waktu Eksekusi (*Brute Force*) ..... 67

# BAB 1

# PENDAHULUAN

## 1.1 Latar Belakang

Tidak dapat disangkal lagi bahwa saat ini pengiriman informasi dapat dilakukan dengan sangat mudah. Hal ini terjadi dikarenakan pesatnya perkembangan teknologi dalam pertukaran informasi. Pertukaran informasi bisa dilakukan melalui berbagai metode, yakni menyampaikan secara langsung dan juga melalui internet. Penyampaian informasi melalui internet rentan akan terjadinya penyadapan dikarenakan tingkat keamanan yang relatif rendah. Hal ini membuat pihak yang tidak bertanggung jawab bisa saja mengambil informasi secara tidak sah bahkan merusaknya. Untuk itu diperlukan suatu teknik khusus untuk mengamankan informasi.

Salah satu metode yang dapat diterapkan untuk menjaga keamanan suatu informasi yakni kriptografi. Metode perhitungan yang berhubungan dengan keamanan informasi, seperti kebijakan kerahasiaan, integritas data, dan otentikasi, adalah fokus utama kriptografi. Sejak dahulu kriptografi dipercaya untuk menangani masalah keamanan pesan atau informasi (Munir, 2019). Kriptografi akan mengubah pesan asli *(plaintext)* menjadi pesan tidak beraturan, dikenal sebagai *ciphertext.* Proses ini disebut sebagai enkripsi, sedangkan untuk menafsirkan *ciphertext* menjadi ke pesan asli disebut sebagai dekripsi. *Ciphertext* didapat melalui perhitungan matematika berdasarkan algoritma tertentu. Salah algoritma kriptografi yakni algoritma M.S.H. Biswas (Rabin-Biswas).

M.S.H. Biswas (Rabin-Biswas) merupakan sebuah rancangan kriptosistem kunci publik yang secara efektif dapat melakukan enkripsi dan dekripsi (Biswas, 2020). Algoritma ini dikembangkan oleh Md Shamim Hossain Biswas pada tahun 2019. Algoritma ini dikembangkan berdasarkan Diffie-Hellman *key exchange protocol*, konsep pangkat aritmatik modular dari Rabin *Cryptosystem* yang dirancang oleh Michael O. Rabin pada tahun 1979, fungsi *floor*, dan fungsi nilai absolut (Biswas, 2019).

Dapat melakukan enkripsi dan dekripsi saja tidak cukup untuk sebuah kriptosistem. Sebuah kriptosistem harus tahan dari serangan terhadap kriptografi. Untuk menguji ketahanan sebuah kriptosistem dari serangan terhadap kriptografi, dapat dilakukan kriptanalisis. Kriptanalisis adalah disiplin ilmu dan seni yang berfokus pada dekripsi ciphertext menjadi plaintext tanpa memiliki pengetahuan tentang kunci yang digunakan (Munir, 2019). Sedangkan, menurut Jafri *et al.* (2021), kriptanalisis merupakan ilmu mendekripsi dan membaca pesan terenkripsi tanpa izin. Seseorang yang melakukan kriptanalisis disebut sebagai kriptanalis (cryptanalyst). Kriptanalisis akan mencoba menguraikan *ciphertext* melalui penerapan serangan terhadap kriptografi. Sarbini *et al.* (2018) menyatakan bahwa tingkat keamanan dari sebuah cryptosystem ditentukan oleh seberapa sulit bagi seorang kriptanalis untuk mendapatkan plaintext tanpa mengetahui kuncinya.

Maka dari itu, pada penelitian akan dilakukan kriptanalisis dengan menggunakan metode *Baby Step Giant Step* pada algoritma M.S.H. Biswas (Rabin-Biswas) untuk menguji ketahanan algoritma tersebut dari serangan terhadap kriptografi serta mencari celah yang dapat dieksploitasi guna mendapatkan kunci untuk mengubah *ciphertext* menjadi *plaintext.*

## 1.2 Rumusan Masalah

Sebuah kriptosistem tidak hanya tentang enkripsi dan dekripsi pesan, tetapi juga harus dapat tahan dari berbagai serangan kriptografi yang dapat membahayakan kerahasiaan pesan. Maka, dalam penelitian ini akan melakukan kriptanalisis terhadap algoritma M.S.H. Biswas (Rabin-Biswas) untuk mengetahui ketahanan algoritma ini terhadap

serangan kriptografi, serta mencari celah yang dapat dieksploitasi untuk mendapatkan kunci dan memecahkan *ciphertext.*

## 1.3 Batasan Penelitian

1. Penelitian ini hanya membahas algoritma kriptografi M.S.H. Biswas (Rabin-Biswas).
2. Tidak membahas kompleksitas dan efisiensi program.
3. Hasil akhir dari penelitian ini adalah kelemahan algoritma Rabin-Biswas dan program yang dapat memecahkan *ciphertext* hasil enkripsi dari algoritma Rabin-Biswas.

## 1.4 Tujuan Penelitian

Mencari kelemahan algoritma Rabin-Biswas serta membangun program yang dapat memecahkan *ciphertext* hasil enkripsi dari algoritma Rabin-Biswas tanpa memiliki pengetahuan tentang kunci yang dipakai dalam proses enkripsi.

## 1.5 Manfaat Penelitian

Mengetahui celah yang dapat dimanfaatkan guna memecahkan *ciphertext* hasil enkripsi dari algoritma Rabin-Biswas yang dapat dijadikan referensi untuk membangun kriptosistem baru yang dapat menahan berbagai serangan terhadap kriptografi guna menjaga kerahasiaan pesan.

## 1.6 Metode Penelitian

Langkah-langkah penelitian yang dilakukan pada proses penyusunan tugas akhir ini sebagai berikut:

1. Studi Pustaka

   Pada fase ini, penelitian akan diawali dari mengumpulkan referensi yang diperoleh dari sumber tertulis, seperti jurnal, buku, serta *proceeding*. Pencarian referensi dilakukan untuk mendapatkan informasi yang terkait dengan kriptanalis, serangan terhadap kriptografi dan algoritma Rabin-Biswas.

2. Analisis dan Perancangan

   Dalam fase ini, algoritma Rabin-Biswas akan dianalisis sebagai algoritma yang akan dikriptanalisis guna menguji ketahanan dari serangan yang dapat membahayakan kerahasiaan pesan serta mencari celah yang dapat dieksploitasi untuk mendapatkan kunci dengan melakukan perancangan diagram alir *(flowchart).*

3. Implementasi Sistem

   Dalam fase ini, akan dilakukan proses pembangunan sistem berdasarkan *flowchart* yang telah dibuat dengan mengimplementasikan kriptanalisis algoritma Rabin-Biswas ke dalam bahasa pemrograman Python.

4. Pengujian Sistem

   Dalam fase ini, akan dilakukan proses uji coba apakah sistem yang dibangun dapat menemukan celah yang dieksploitasi untuk mendapatkan kunci serta memecahkan *ciphertext.*

5. Dokumentasi

   Dalam fase, akan dilaksanakan proses penyusunan laporan yang melibatkan tahap analisis hingga pengujian dalam format skripsi.

## 1.7 Sistematika Penulisan

Dalam struktur penyusunan tugas akhir ini, terbagi menjadi lima bab, dengan setiap bab dijelaskan sebagai berikut :

**BAB 1 PENDAHULUAN**

Pada bagian ini kita akan memahami penjelasan di balik pemilihan judul, pembahasan masalah, termasuk sasaran penelitian, pemanfaatan hasil ujian, strategi penelitian, dan ikhtisar artikel.

**BAB 2 LANDASAN TEORI**

Algoritma Rabin-Biswas, Algoritma *Baby Step Giant Step*, dan Kriptosistem Kunci Publik semuanya merupakan subjek tinjauan teoritis dalam bab ini.

**BAB 3 ANALISIS DAN PERANCANGAN**

Analisis masalah dan desain perangkat lunak dijelaskan dalam bab ini.

**BAB 4 IMPLEMENTASI DAN PENGUJIAN**

Implementasi perangkat lunak hasil analisis dan perancangan yang telah dilakukan, dibahas pada bab ini.

**BAB 5 PENUTUP**

Pada bagian ini, akan dibahas rangkuman dari penjelasan bab-bab sebelumnya. Dengan berdasarkan kesimpulan tersebut, penulis akan memberikan saran yang dapat memberikan kontribusi positif untuk melengkapi serta meningkatkan pengembangan sistem yang telah disusun.

# BAB 2

# LANDASAN TEORI

## 2.1 Kriptografi

Kata kriptografi sendiri berasal dari bahasa Yunani, yaitu *cryptós* (rahasia) dan *gráphein* (menulis), dengan asumsi digabungkan menjadi "komposisi rahasia". Dilihat dari ungkapannya, kriptografi dapat diartikan sebagai suatu strategi yang dapat digunakan untuk mengubah data menjadi sesuatu yang bersifat pribadi.

Metode perhitungan yang berhubungan dengan keamanan informasi, seperti kebijakan kerahasiaan, integritas data, dan otentikasi, adalah fokus utama kriptografi (Munir, 2019). Pada prosesnya, pesan asli atau biasa disebut sebagai *plaintext*, akan dikonversi menjadi sebuah pesan yang telah dienkripsi sehingga kehilangan makna aslinya atau biasa disebut sebagai *ciphertext*.

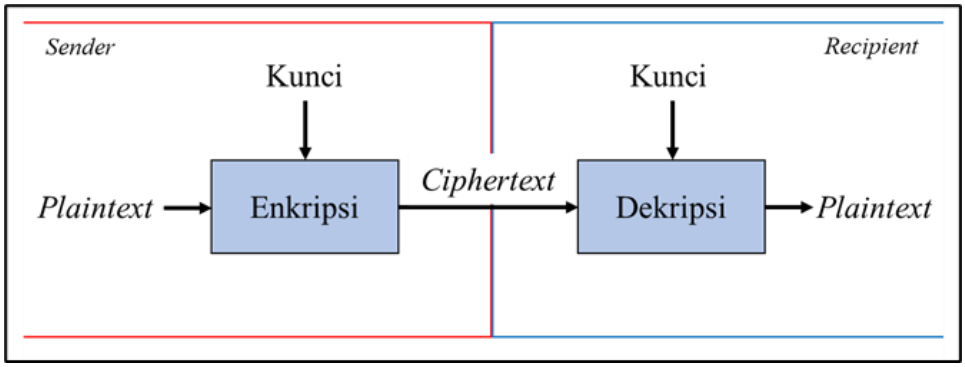
Pada kriptografi, terdapat dua proses utama, yaitu :

1. Enkripsi

   Enkripsi merupakan fase menyandikan pesan asli *(plaintext)* yang nantinya akan berubah menjadi *ciphertext*. Pada tahap ini, pesan asli akan diubah menjadi kode acak sehingga maknanya sulit untuk dikenali oleh pihak yang tidak sah.

2. Dekripsi

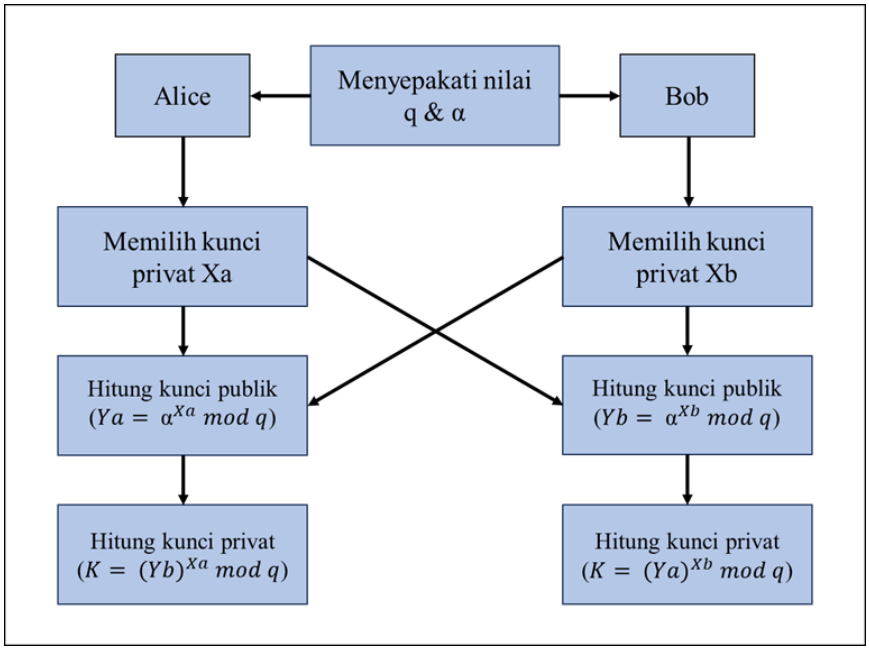
   Proses menafsirkan *ciphertext* ke bentuk aslinya, yakni pesan asli *(plaintext),* disebut dekripsi. Pada tahap ini, *ciphertext* akan diubah kembali menjadi pesan asli *(plaintext),* guna penerima *(recipient)* dapat memahami makna asli dari pesan yang dikirimkan.




**Gambar 2.1 Proses Enkripsi dan Dekripsi**

## 2.2 Kriptosistem Kunci Publik

Kriptosistem kunci publik adalah teknik kriptografi yang menggunakan dua kunci terpisah untuk menyelesaikan cara pengacakan dan penguraian pesan yang paling umum. Ada dua jenis kunci yang digunakan dalam kriptosistem ini, yaitu kunci publik khusus dan kunci rahasia. Meskipun kunci privat hanya dapat diakses oleh *sender* dan *recipient*, kunci publik dapat diakses oleh semua orang. Salah satu metode yang cukup terkenal dalam pembangkitan kunci untuk kriptosistem kunci publik ini adalah *Diffie-Hellman Key Exchange Protocol.*




**Gambar 2.2 Proses pembangkitan kunci menggunakan Diffie-Hellman Key Exchange Protocol**

Di bidang kriptografi, Protokol Pertukaran Kunci Diffie-Hellman memungkinkan dua pihak menyepakati kunci rahasia yang nantinya akan digunakan untuk mengenkripsi dan mendekripsi pesan. Strategi ini ditemukan pada tahun 1976 oleh Whitfield Diffie dan Martin Hellman. Protokol ini melibatkan penggunaan metode kunci publik untuk menyederhanakan pertukaran kunci rahasia tanpa mengirimkan kunci itu sendiri melalui jalur yang tidak aman. Pendekatan ini menjadi dasar bagi berbagai protokol kriptografi modern yang digunakan untuk menjadi privasi dan keamanan data dalam berkomunikasi secara daring.

*Pada Diffie-Hellman Key Exchange Protocol, sender* dan *recipient* akan saling berbagi kunci publik yang masing-masing telah mereka bangkitkan. Kunci publik tersebut akan dipergunakan pada fase pembangkitan kunci privat, yang nantinya akan dipakai dalam tahap enkripsi dan dekripsi pesan .

## 2.3 Uji Keprimaan Algoritma AKS

Algoritma AKS dikembangkan oleh tiga metematikawan India, yakni Manindra Agrawal, Neeraj Kayal, dan Nitin Saxena pada tahun 2002. AKS merupakan algoritma deterministik dalam pembuktian keprimaan suatu angka. AKS memanfaatkan teknik-teknik matematika yang canggih, seperti struktur aljabar polinomial dan teorema turunan, untuk memberikan keprimaan suatu bilangan secara eksak tanpa mengandalkan asumsi yang rumit. Berikut adalah langkah-langkah uji keprimaan menggunakan algoritma AKS (Agrawal, *et al.*, 2002).

**Masukan** : bilangan bulat $n > 1$

1. Jika $n = a^b$ dimana $a > 1$ dan $b > 1$, keluaran komposit.
2. Temukan $r$ terkecil yang memenuhi $ord_r(n) > (log_2 n)^2$
3. Jika $1 < gcd(a, n) < n$ untuk setiap $a \leq r$, keluaran komposit.
4. Jika $n \leq r$, keluaran prima.
5. Untuk semua $0 < a < \lfloor \sqrt{\phi(r)} \log_2 n \rfloor$, jika $((X + a)^n \neq X^n + a (mod\ X^r - 1, n)$, keluaran komposit.
6. Keluaran prima.

Contoh uji keprimaan bilangan bulat $n$ menggunakan algoritma AKS. Misalkan $n = 3$.

Langkah 1

Apakah $n = a^b$?

$a = 2, b = 2 \rightarrow a^b = 2^2 = 4$

Tidak ada $a^b = n$.

Langkah 2

Cari $r$ yang memenuhi $ord_r(n) > (log_2 n)^2$.

$(log_2 n)^2 = (log_2 3)^2 = 2.5$

$r = 2 \rightarrow ord_r(n) = ord_2(3) = \ldots?$

| e | $3^e$ mod r<br>$3^e$ mod 2 |
|---|---|
| 1 | 1 |

$e = 1 \rightarrow 1 < (log_2 n)^2$

$r = 3 \rightarrow ord_r(n) = ord_3(3) = \ldots?$

| e | $3^e$ mod r<br>$3^e$ mod 3 |
|---|---|
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |

$e =$ Tidak ada

$r = 4 \rightarrow ord_r(n) = ord_4(3) = \ldots?$

| e | $3^e$ mod r<br>$3^e$ mod 4 |
|---|---|
| 1 | 3 |
| 2 | 1 |

$e = 2 \rightarrow 2 < (log_2 n)^2$

$r = 5 \rightarrow ord_r(n) = ord_5(3) = \ldots?$

| e | $3^e$ mod r<br>$3^e$ mod 5 |
|---|---|
| 1 | 3 |
| 2 | 4 |
| 3 | 2 |
| 4 | 1 |

$e = 4 \rightarrow 4 > (log_2 n)^2$

Langkah 3

Cari $gcd(a,n)$ dimana $a \leq r$.

$$a \leq r \rightarrow a \leq 5$$
$$gcd(a,n) \rightarrow a = 2 \rightarrow gcd(2,3) = 1$$
$$\rightarrow a = 3 \rightarrow gcd(3,3) = 3$$
$$\rightarrow a = 4 \rightarrow gcd(4,3) = 1$$
$$\rightarrow a = 5 \rightarrow gcd(5,3) = 1$$

Tidak ada $gcd(a,n)$ yang $> 1$ dan $< n$.

Langkah 4

Cek apakah $n \leq r$.

$$3 \leq 5$$

$\therefore$ Ya, $n \leq r$. Maka 3 adalah prima.

## 2.4 Uji Keprimaan Algoritma Fermat

Uji keprimaan algoritma Fermat adalah salah satu metode dalam teori bilangan yang dipergunakan untuk menguji apakah suatu angka dapat dikategorikan sebagai bilangan prima atau tidak. Algoritma ini didasarkan pada Prinsip Fermat, yang menyebutkan jika $p$ adalah bilangan prima serta $a$ adalah bilangan asli yang relatif prima terhadap $p$, maka $a^{(p-1)} \equiv 1 \ (mod\ p)$. Dengan demikian, apabila suatu bilangan $p$ adalah prima, maka $a^{(p-1)}$ akan memberikan hasil yang kongruen dengan 1 modulo $p$. Berikut adalah langkah-langkah umum dari uji keprimaan algoritma Fermat:

**Masukan:** Bilangan bulat $n > 3$ dan $k$

1. Pilih $a$ secara acak, dimana $2 < a < n - 2$
2. Jika $a^{n-1} \not\equiv 1 \ (mod\ n)$, kembalikan ‘komposit’.
3. Ulangi langkah 1 dan 2 sebanyak $k$ kali. Jika keluaran tidak pernah ‘komposit’, kembalikan ‘kemungkinan prima’.

Contoh uji keprimaan bilangan bulat $n$ menggunakan algoritma Fermat. Misalkan $n = 17$ dan $k = 3$.

Karena $k = 3$, maka pengujian dilakukan sebanyak 3 kali.

Uji 1

$a = 8$

Cek apakah $a^{n-1} \not\equiv 1 \ (mod\ n)$

$a^{n-1}\ mod\ n = \ldots?$

$8^{17-1}\ mod\ 17 = 8^{16}\ mod\ 17 \rightarrow 16_{(10)} = 10000$

- bit 1: $z = x * z^2\ mod\ n = 8 * 1^2\ mod\ 17 = 8$
- bit 0: $z = z^2\ mod\ n = 8^2\ mod\ 17 = 13$
- bit 0: $z = z^2\ mod\ n = 13^2\ mod\ 17 = 16$
- bit 0: $z = z^2\ mod\ n = 16^2\ mod\ 17 = 1$
- bit 0: $z = z^2\ mod\ n = 1^2\ mod\ 17 = 1$

$8^{17-1} \equiv 1(mod\ n)$ ✓

Uji 2

$a = 4$

Cek apakah $a^{n-1} \not\equiv 1 \ (mod\ n)$

$a^{n-1}\ mod\ n = \ldots?$

$4^{17-1}\ mod\ 17 = 4^{16}\ mod\ 17 \rightarrow 16_{(10)} = 10000$

- bit 1: $z = x * z^2\ mod\ n = 4 * 1^2\ mod\ 17 = 4$
- bit 0: $z = z^2\ mod\ n = 4^2\ mod\ 17 = 16$
- bit 0: $z = z^2\ mod\ n = 16^2\ mod\ 17 = 1$
- bit 0: $z = z^2\ mod\ n = 1^2\ mod\ 17 = 1$
- bit 0: $z = z^2\ mod\ n = 1^2\ mod\ 17 = 1$

$4^{17-1} \equiv 1(mod\ n)$ ✓

Uji 3

$a = 13$

Cek apakah $a^{n-1} \not\equiv 1 \ (mod\ n)$

$a^{n-1}\ mod\ n = \ldots?$

$13^{17-1}\ mod\ 17 = 13^{16}\ mod\ 17 \rightarrow 16_{(10)} = 10000$

- bit 1: $z = x * z^2\ mod\ n = 13 * 1^2\ mod\ 17 = 13$
- bit 0: $z = z^2\ mod\ n = 13^2\ mod\ 17 = 16$

- bit 0: $z = z^2 \ mod \ n = 16^2 \ mod \ 17 = 1$
- bit 0: $z = z^2 \ mod \ n = 1^2 \ mod \ 17 = 1$
- bit 0: $z = z^2 \ mod \ n = 1^2 \ mod \ 17 = 1$

$4^{17-1} \equiv 1(mod \ n)$ ✓

∴ Hasil 3 pengujian menunjukkan $a^{n-1} \equiv 1 \ (mod \ n)$, jadi 17 adalah kemungkinan prima.

## 2.5 Algoritma RSA

Dalam makalah ilmiah tahun 1978 berjudul "*A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*" tiga peneliti dari MIT (*Massachusetts Institute of Technology*) — Ron Rivest, Adi Shamir, dan Leonard Adleman — memperkenalkan algoritma RSA. Tujuan utama dari pengenalan algoritma ini adalah untuk menggantikan algoritma National Bureau of Standards (NBS) dengan tingkat keamanan rendah.

Penelitian Rivest, *et al.* (1978) yang berjudul *"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems",* mengatakan bahwa algoritma RSA terinspirasi dari karya-karya sebelumnya oleh Diffie dan Hellman. Algoritma RSA merupakan implementasi dari *"Public-Key Cryptosystem",* sebuah gagasan elegan yang ditemukan oleh Diffie dan Hellman. Berikut adalah rumus enkripsi dan dekripsi dari algoritma ini (Rivest, *et al.*, 1978).

***Encryption Algorithm:***

$$E(M) = M^e \ (mod \ n)$$

***Decryption Algorithm:***

$$D(C) = C^d \ (mod \ n)$$

## 2.6 Algoritma Rabin

Algoritma Rabin dipublikasikan pada tahun 1979 melalui karya ilmiah yang berjudul *"Digitalized Signatures and Public Key Function as Intractable as Factorization",* oleh seorang ilmuwan asal *MIT (Massachusetts Institute of Technology)* yang bernama Michael O. Rabin. Karya ilmiah ini diterbitkan di Laboratorium Ilmu Komputer *MIT*, tempat yang sama dengan diterbitkannya algoritma RSA. Jadi, tidak heran jika melihat banyak kesamaan antara dua kriptosistem ini. Berikut adalah rumus enkripsi dan dekripsi dari algoritma ini (Rabin, 1979).

***Encryption Algorithm:***

$$C = M^2 \ mod \ n$$

***Decryption Algorithm:***

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

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

Namun, algoritma Rabin ini memiliki kelemahan, yakni pada proses pengenkripsian pesan menggunakan algoritma ini, akan menghasilkan 4 *ciphertext* yang berbeda dan dari keempat *ciphertext* yang dihasilkan tersebut, hanya ada 1 yang benar. Dan pihak-pihak yang terlibat proses kriptografi tidak ada satupun yang tau *ciphertext* mana yang jika didekripsi akan kembali ke bentuk semula *(plaintext).*

Berdasarkan kekurangan tersebut, mendorong para kriptografer di kemudian hari untuk menyempurnakan algoritma Rabin ini. Diantaranya algoritma $AA_\beta$, Algoritma Rabin-P, dan Algoritma Rabin-Biswas.

## 2.7 Algoritma $AA_\beta$

Algoritma $AA_\beta$ pertama kali diperkenalkan pada tahun 2013 melalui karya ilmiah yang berjudul, *"A New Efficient Asymmetric Cryptosystem Based on the Integer Factorization Problem of* $N = p^2q$*"* karya Ariffin, Asbullah, Abu dan Mahad. Algoritma $AA_\beta$ mampu memanfaatkan masalah modulo akar kuadrat dan mampu mengatasi skenario kegagalan dekripsi yang terjadi pada algoritma Rabin (Arifin, *et al., 2013)*.

Berikut adalah skema untuk pembangkitan kunci, enkripsi, dan dekripsi yang digunakan pada algoritma $AA_\beta$ (Arifin, *et al.,* 2013).

***Key Generation Algorithm:***

1. Tentukan 2 bilangan prima acak dan berbeda untuk p dan q yang memenuhi kondisi berikut:
   $2 < p, q < 2^{k+1}$ dan $p, q \equiv 3\ (mod\ 4)$ atau bisa disebut sebagai *blum integer*.
2. Hitung $A_2 = p^2 q$
3. Tentukan sebuah bilangan bulat acak untuk $A_1$ yang memenuhi kondisi berikut:
   $2^{3k+4} < A_1 < 2^{3k+6}$ dan $gdc(A_1, A_2) = 1$
4. Hitung d, dengan rumus $A_1 d \equiv 1\ (mod\ A_2)$
5. Kunci publik adalah $A_1$ dan $A_2$ dan kunci privatnya adalah d, p, dan q.

***Encryption Algorithm:***

1. Pilih sebuah *plaintext* m yang memenuhi kondisi $2^{2k-2} < m < 2^{2k-1}$ dimana $gcd(m, A_2) = 1$
2. Pilih sebuah *plaintext* t dimana $2^{4k} < t < 2^{4k+1}$
3. Hitung $c = A_1 m^2 + A_2 t$

***Decryption Algorithm:***

1. Hitung $w \equiv cd\ (mod A_2)$
2. Hitung $m_p \equiv w^{\frac{p+1}{4}}\ (mod\ p)$
3. Hitung $m_q = w^{\frac{q+1}{4}}\ (mod\ q)$
4. Hitung $t \equiv \frac{c - A_1 m^2}{A_2}$

## 2.8 Algoritma Rabin-*p*

Algoritma Rabin-*p* diperkenalkan pertama kali pada tahun 2013 lewat karya ilmiah yang berjudul, *"Rabin-p Cryptosystem: Practical and Efficient Method for Rabin based Encryption Scheme"*. Algoritma Rabin-*p* berhasil memecahkan masalah pada proses enkripsi algoritma Rabin yang menghasilkan 4 *ciphertext*, dimana pada algoritma Rabin-*p* pada proses dekripsinya hanya menghasilkan satu hasil, yakni pesan asli

(Budiman *et al.,* 2020). Berikut adalah skema untuk *key generation*, enkripsi dan dekripsi untuk algoritma Rabin-*p* (Asbullah, *et al.,* 2013).

***Key Generation Algorithm:***

1. Bangkitkan 2 bilangan prima acak dan berbeda untuk p dan q yang memenuhi kondisi $p \equiv 3\ (mod\ 4)$ dan $q \equiv 3\ (mod\ 4)$ atau bisa disebut sebagai *blum integer*.
2. Hitung $N = p^2q$
3. N sebagai kunci publik dan p sebagai kunci privat.

***Encryption Algorithm:***

1. Pilih *plaintext* $m < 2^{2k-1}$ dimana $gcd(m, N) = 1$
2. Hitung $c \equiv m^2\ (mod\ N)$

***Decryption Algorithm:***

1. Hitung $w \equiv$ c $(mod\ p)$
2. Hitung $m_p \equiv w^{\frac{p+1}{4}}\ (mod\ p)$
3. Hitung $i \equiv \frac{c-{m_p}^2}{p}$
4. Hitung $j = \frac{i}{2m_p}\ (mod\ P)$
5. Hitung $m_1 = m_p + jp$
6. Jika $m_1 < \frac{p^2}{2}$, maka $m = m_1$
7. Selain itu, $m = p^2 - m_1$

## 2.9 Algoritma M.S.H. Biswas (Rabin-Biswas)

Algoritma M.S.H. Biswas (Rabin-Biswas) ini dirancang oleh seseorang bernama Md Shamim Hossain Biswas. M.S.H pada tahun 2019. M.S.H. Biswas (Rabin-Biswas) merupakan sebuah rancangan kriptosistem kunci publik yang secara efektif dapat melakukan enkripsi dan dekripsi (Biswas, 2020). Algoritma ini dikembangkan berdasarkan protokol pertukaran kunci Diffie-Hellman, konsep pangkat aritmatik modular dari *Michael O. Rabin Cryptosystem*, fungsi *floor*, dan fungsi nilai absolut (Biswas, 2019). Adapun rumus enkripsi, dekripsi, dan pembangkitan kunci publik dari algoritma ini adalah sebagai berikut (Biswas, 2019).

***Key Generation Algorithm:***

$$\boldsymbol{K = (Y_b)^{xa} \bmod q}$$
$$\boldsymbol{= \left(\alpha^{xb} \bmod q\right)^{xa} \bmod q}$$
$$\boldsymbol{= \left(\alpha^{xb}\right)^{xa} \bmod q}$$
$$\boldsymbol{= \alpha^{xb\,xa} \bmod q}$$
$$\boldsymbol{= (\alpha^{xa})^{xb} \bmod q}$$
$$\boldsymbol{= (\alpha^{xa} \bmod q)^{xb} \bmod q}$$
$$\boldsymbol{= (Y_a)^{xb} \bmod q}$$

***Encryption Algorithm:***

$$Q = \llcorner\, \boldsymbol{m^2 / K}\, \lrcorner$$
$$\boldsymbol{R = m^2 \bmod K}$$
$$\boldsymbol{C = (Q, R)}$$

***Decryption Algorithm:***

$$D = |\sqrt{Q.K + R}|$$

Pada proses pembangkitan kunci, digunakan *Diffie-Hellman Key Exchange Protocol* untuk membangkitkan kunci publik. Kunci publik $(Ya, Yb)$ digunakan dalam pembangkitan kunci rahasia $(K)$ yang akan dipakai pada fase enkripsi. Dimana q adalah bilangan prima, $\alpha$ adalah akar primitif dari $q$, serta $xa$, $xb$ yang merupakan kunci privat yang dibangkitkan oleh *sender* dan *recipient*. Pada proses enkripsi akan menggunakan fungsi *floor* dan juga modulus, guna mendapatkan *ciphertext*. *Ciphertext* akan dikirim kepada *recipient*, yang selanjutnya akan dilakukan proses dekripsi. Pada proses dekripsi digunakan fungsi nilai absolut untuk mendapatkan pesan asli *(plaintext).*

Berikut contoh dari Algoritma M.S.H. Biswas (Rabin-Biswas) pada sebuah karakter:

1. Sebagai contoh untuk pembangkitan kunci, bangkitkan $q$ dan $\alpha$, dimana $q$ adalah bilangan prima dan $\alpha$ adalah akar primitif dari $q$. Selanjutnya masing-masing A dan B memilih kunci privat $(Xa, Xb)$, dimana $Xa$ dan $Xb$ lebih kecil dari $q$. Lalu hitung kunci publik $(Ya,\ Yb)$ dengan rumus $Ya = \alpha^{\mathrm{Xa}}\ mod\ q$ dan $Yb = \alpha^{\mathrm{Xb}}\ mod\ q$. Langkah terakhir hitung kunci rahasia $(K)$ dengan rumus $K = (Yb)^{\mathrm{Xa}}\ mod\ q = (Ya)^{\mathrm{Xb}}\ mod\ q$. Pada Tabel 2.1 ditunjukkan bagaimana proses pembangkitan kunci.

**Tabel 2.1 Proses Pembangkitan Kunci**

| Alice (Sender) | | Eve (Eavesdropper) | | Bob (Receiver) | |
|---|---|---|---|---|---|
| *Known* | *Unknown* | Known | Unknown | Known | Unknown |
| q = 331 | | ✓ | | ✓ | |
| α = 3 | | ✓ | | ✓ | |
| Xa = 281 | Xb = 151 | | Xa & Xb | Xb = 151 | Xa = 281 |
| $Ya = 3^{281} \bmod 331$ | | | | $Yb = 3^{151} \bmod 331$ | |
| $K = 209^{281} \bmod 331$<br>K = 107 | | 209 | 158 | $K = 158^{151} \bmod 331$<br>K = 107 | |

2. Sebagai contoh untuk enkripsi, dengan kunci $K = 107$ dan sebuah pesan *(plaintext)* yaitu "THORIQ AUFAR". Pada Tabel 2.2 ditunjukkan bagaimana proses enkripsi *plaintext* tersebut.

**Tabel 2.2 Proses Enkripsi M.S.H Biswas (Rabin-Biswas)**

| *Plaintext* (m) | Kode ASCII | $Q = \lfloor m^2/K \rfloor$ | $R = m^2 \bmod K$ | C = (Q, R) |
|---|---|---|---|---|
| T | 84 | 65 | 101 | (65, 101) |
| H | 72 | 48 | 48 | (48, 48) |
| O | 79 | 58 | 35 | (58, 35) |
| R | 82 | 62 | 90 | (62, 90) |
| I | 73 | 49 | 86 | (49, 86) |
| Q | 81 | 61 | 34 | (61, 34) |
| (spasi) | 32 | 9 | 61 | (9, 61) |
| A | 65 | 39 | 52 | (39, 52) |
| U | 85 | 67 | 56 | (67, 56) |
| F | 70 | 45 | 85 | (45, 85) |
| A | 65 | 39 | 52 | (39, 52) |
| R | 82 | 62 | 90 | (62, 90) |

3. Sebagai contoh untuk dekripsi, dengan kunci $K = 107$ dan sebuah pesan asli *(plaintext)* yang telah diubah menjadi *ciphertext*, pihak *recipient* akan melakukan dekripsi pada *ciphertext* untuk mengetahui pesan asli yang dikirim oleh *sender*. Maka proses dekripsi *ciphertext* ditunjukkan seperti pada Tabel 2.3 sebagai berikut:

**Tabel 2.3 Proses Dekripsi M.S.H. Biswas (Rabin-Biswas)**

| *Ciphertext* (Q, R) | $D = \sqrt{Q * K + R}$ | *Plaintext* (m) |
|---|---|---|
| (65, 101) | 84 | T |
| (48, 48) | 72 | H |
| (58, 35) | 79 | O |
| (62, 90) | 82 | R |
| (49, 86) | 73 | I |
| (61, 34) | 81 | Q |
| (9, 61) | 32 | (spasi) |
| (39, 52) | 65 | A |
| (67, 56) | 85 | U |
| (45, 85) | 70 | F |
| (39, 52) | 65 | A |
| (62, 90) | 82 | R |

## 2.10 Kriptanalisis

Kriptanalisis merupakan cabang ilmu ataupun seni yang ditujukan untuk memecahkan *ciphertext* menjadi ke bentuk semula yakni *plaintext* tanpa pengetahuan mengenai kunci yang digunakan dalam proses enkripsi. Orang yang mempraktikkan kriptanalisis disebut sebagai kriptanalis.

Kriptanalis pertama dikemukakan pada abad ke-9 oleh ilmuwan Arab yang memiliki nama Abu Yusuf Ibnu Ishaq Ibnu As-Sabbah Ibnu ‘Omran Ibnu Ismail Al-Kindi, atau yang lebih dikenal sebagai Al-Kindi (Munir, 2019). Pada saat itu, Al-Kindi menemukan sebuah teknik yang saat ini dikenal sebagai analisis frekuensi, yakni sebuah metode untuk menguraikan teks terenkripsi dengan merujuk pada seberapa sering karakter muncul dalam pesan. Pendekatan ini melibatkan penelitian terhadap seberapa sering huruf-huruf tertentu muncul dalam Al-Quran.

## 2.11 Serangan terhadap Kriptografi

Menurut Munir (2019), terdapat beberapa jenis serangan terhadap kriptografi, yaitu :

1. Berdasarkan keterlibatan penyerang dalam komunikasi
   a. Serangan pasif *(passive attack)*

      Dalam jenis serangan ini, pihak yang melakukan serangan tidak terlibat pada proses interaksi antara *sender* dan *recipient*. Penyerang berfokus pada kegiatan penyadapan untuk mengoleksi sebanyak mungkin informasi. Beberapa metode yang umum meliputi :

      - *Wiretapping*

        Pihak yang melakukan serangan menghalangi data yang sedang dikirim melalui saluran kabel komunikasi dengan memanfaatkan perangkat keras guna mendapatkan akses pada sambungan tersebut.
      - *Electromagnetic Eavesdropping*

        Pihak yang melakukan serangan mengintersep data yang sedang dikirim melalui saluran nirkabel, seperti melalui frekuensi radio dan gelombang mikro.
      - *Acoustic Eavesdropping*

        Merekam gelombang suara yang timbul dari ucapan.

   b. Serangan aktif *(active attack)*

      Dalam jenis serangan ini, pihak yang melakukan serangan terlibat dalam intervensi komunikasi dan berusaha memanipulasi sistem untuk keuntungannya sendiri. Salah satu metode yang dapat digunakan ialah man-in-the-middle attack.
2. Berdasarkan teknik yang digunakan untuk menemukan kunci
   a. *Brute force attack*

      Sering disebut juga dengan *exhaustive* attack. Dikeranakan serangan ini melelahkan sebab mencoba setiap kemungkingan yang ada.

   b. *Analytical attack*

      Jenis serangan ini akan berfokus pada analisis matematis atau statistik dari algoritma kriptografi atau kunci yang digunakan.
3. Berdasarkan ketersediaan data
   a. *Ciphertext-only attack*, kriptanalis hanya mendapatkan *ciphertext*.

b. *Known-plaintext attack*, selain mendapatkan *ciphertext*, kriptanalis juga mendapatkan *plaintext*.
c. *Chosen-plaintext attack*, kriptanalis dapat memilih dengan sengaja suatu teks asli tertentu untuk dienkripsi, khususnya yang dapat memberikan petunjuk atau arahan lebih jelas terkait penemuan kunci.
d. *Adaptive-chosen-plaintext attack*, kriptanalis memilih blok besar dari teks asli, mengenkripsinya, dan kemudian menggunakan hasil serangan sebelumnya untuk memilih blok yang lebih kecil untuk didekripsi. Proses ini berulang.
e. *Chosen-ciphertext attack*, kriptanalis memiliki kemampuan untuk memilih berbagai *ciphertext* yang akan didekripsi dan mendapatkan akses terhadap teks asli yang telah diuraikan.
f. *Chosen-key attack*, Kriptanalis dapat menentukan kunci yang tepat untuk mendekripsi pesan karena mereka menyadari hubungan antara berbagai kunci tersebut.
g. *Rubber-hose cryptanalysis*, kriptanalis akan mengintimidasi atau mengirim pesan ancaman, bahkan mungkin menggunakan tindakan ekstrem untuk memaksa pihak yang memiliki kunci untuk memberikannya guna melakukan dekripsi pesan tersebut..

## 2.12 Algoritma *Baby Step Giant Step*

Algoritma *Baby Step Giant Step* adalah sebuah metode efisien untuk menyelesaikan masalah logaritma diskrit atau persamaan modular diskrit dalam konteks teori bilangan dan kriptografi. Algoritma ini dapat menemukan nilai $x$ yang memenuhi $\alpha^x \equiv \beta \ (mod\ n)$. Dengan pendekatan yang inovatif, algoritma ini memecah masalah logaritma diskrit menjadi dua langkah kunci, yakni *Baby-Step* dan *Giant-Step*. Berikut adalah langkah-langkah dalam algoritma *Baby Step Giant Step* untuk menyelesaikan permasalahan diskrit (Shanks, 1971).

**Masukan** : $n$, $\alpha$, $\beta$, dimana n adalah modulus, $\alpha$ adalah generator dari $n$, dan $\beta$ adalah hasil modulus.

1. Hitung $m = \lceil \sqrt{n} \rceil$

2. Untuk semua $j$ dimana $0 \leq j < m$, hitung $aj = \alpha^j \ mod \ n$ dan masukkan $(j, aj)$ ke sebuah tabel.
3. Hitung $am = \alpha^{-m} \ mod \ n$
4. Tetapkan $y = \beta$
5. Untuk semua $i$ dimana $0 \leq i < m$ :
   a. Cek apakah $y$ ada pada komponen kedua $(aj)$ dalam tabel.
   b. Jika ada, kembalikan $i * m \ + \ j \ mod \ n$
   c. Jika tidak ada, $y = y * am \ mod \ n$

Berikut adalah contoh penggunaan algoritma *Baby Step Giant Step* dalam menyelasaikan permasalahan logaritma diskrit.

$n = 13$

$\alpha = 2$

$\beta = 7$

Berapa $x$ yang memenuhi $\alpha^x \equiv \beta \ (mod \ n)$?

Langkah 1

$$m = \lceil \sqrt{n} \rceil = \lceil \sqrt{13} \rceil = \lceil 3.605 \rceil = 4$$

Langkah 2

$$aj_0 \rightarrow j = 0 \rightarrow aj = a^j \ mod \ n$$
$$= 2^0 \ mod \ 13$$
$$= 1$$
$$aj_1 \rightarrow j = 1 \rightarrow aj = a^j \ mod \ n$$
$$= 2^1 \ mod \ 13$$
$$= 2$$

$$aj_2 \rightarrow j = 2 \rightarrow aj = a^j \ mod \ n$$
$$= 2^2 \ mod \ 13$$
$$= 4$$
$$aj_3 \rightarrow j = 3 \rightarrow aj = a^j \ mod \ n$$
$$= 2^3 \ mod \ 13$$
$$= 8$$

**Tabel 2.4 Pasangan j dan aj**

| j | aj |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 4 |
| 3 | 8 |

Langkah 3

$$am = a^{-m} \; mod \; n$$
$$= 2^{-4} \; mod \; 13$$
$$= 2^{-4} 2^{\phi(13)} \; mod \; 13$$
$$= 2^{-4} 2^{12} \; mod \; 13$$
$$= 2^{8} \; mod \; 13$$
$$= 256 \; mod \; 13$$
$$= 9$$

Langkah 4

$$y = \beta$$
$$y = 7$$

Langkah 5

$i = 0 \rightarrow$ Cek apakah $y = 7$ ada dalam tabel pada kolom $aj$? Tidak ada. Maka :

$$y = y * am \; mod \; n$$
$$= 7 * 9 \; mod \; 13$$
$$= 63 \; mod \; 13$$
$$y = 11$$

$i = 1 \rightarrow$ Cek apakah $y = 11$ ada dalam tabel pada kolom $aj$? Tidak ada. Maka :

$$y = y * am \; mod \; n$$
$$= 11 * 9 \; mod \; 13$$
$$= 99 \; mod \; 13$$
$$y = 8$$

$i = 2 \rightarrow$ Cek apakah $y = 8$ ada dalam tabel pada kolom $aj$? Ada pada $j = 3$. Maka :

$$x = i * m + j \; mod \; n$$
$$= 2 * 4 + 3 \; mod \; 13$$
$$= 8 + 3 \; mod \; 13$$

$= 11 \; mod \; 13$

$= 11$

∴ Nilai $x$ yang memenuhi $2^x \equiv 7 \; (mod \; 13)$ adalah 11.

## 2.13 Penelitian Relevan

1. Penelitian Kota & Aissi (2022) yang berjudul *"Implementation of the RSA algorithm and its cryptanalysis"*, menyatakan bahwa kriptosistem kunci publik yang mereka kriptanalisis, yakni algoritma RSA, berhasil dikriptanalisis menggunakan algoritma pecahan lanjutan. Metode ini dapat menemukan kunci dengan kurang dari $n^{0.25}$. Namun metode ini tidak dapat bekerja jika *e* yang dipilih lebih besar dari $(pq)^{1.5}$, dimana *e* adalah kunci publik.
2. Penelitian Pramitasari (2022) yang berjudul "ALGORITMA OPTIMASI CHAOS PADA RIDGE POLYNOMIAL NEURAL NETWORk UNTUK KRIPTANALISIS KUNCI PUBLIK ELGAMAL", menyatakan bahwa metode yang ia gunakan dalam proses kriptanalisis pada kriptosistem kunci publik El Gamal dapat menemukan kunci privat dengan baik. Pada prosesnya, beberapa variabel dapat mempengaruhi tingkat keberhasilan dan menghasilkan nilai *error*, hal ini tidak bergantung pada perbedaan *learning rate* dan banyaknya *order* PSN.
3. Penelitian Mahad, *et al.* (2022), yang berjudul *"Cryptanalysis of RSA-Variant Cryptosystem Generated by Potential Rogue CA Methodology"*, pada penelitian tersebut, mereka melakukan kriptanalisis pada salah satu varian ataupun modifikasi dari algoritma RSA, yakni Murru-Saettone RSA. Mereka menyatakan bahwa ketika $|Z - \psi| < \frac{p-q}{p+q} N^{1/4}$ atau $|Z - \psi| < N$ terpenuhi, kriptosistem ini sangat rentan terhadap serangan. Pada penelitian ini mereka berhasil melakukan kriptanalisis dengan menggunakan algoritma pecahan lanjutan untuk mendapatkan faktor dari modulus N tanpa mengetahui kunci privat maupun kunci publik.
4. Penelitian Grari, *et al.* (2021) yang berjudul *"Cryptanalysis of Merkle-Hellman cipher using ant colony optimization"*, pada penelitian tersebut mereka memperkenalkan metode baru yang dapat digunakan untuk kriptanalisis algoritma Merkle-Hellman, yakni *ant colony optimization*. Metode dapat bekerja dengan

baik, dengan hasil terbaik diperoleh dengan prosedur MH MACO. Namun metode ini masih perlu dikembangkan lagi untuk melakukan kriptanalisis yang lebih efisien.

5. Penelitian Susilo, *et al.* (2021) yang berjudul *"Divide and capure: An improved cryptanalysis of the encryption standart algorithm RSA"*, mereka memperkenalkan serangan baru yang dapat melakukan kriptanalisis algoritma RSA dengan melakukan peningkatan dari *Wiener attack*. Dengan sebuah paramater $t$, serangan mereka dapat memecah algoritma RSA dengan $d < \sqrt{t\,(2\sqrt{2} + 8/3\,N^{75}/\sqrt{e}}$ dengan *running* time $O(tlog(N))$. Serangan mereka sangat cocok untuk kasus saat $e$ jauh lebih kecil dari $N$.

# BAB 3

# ANALISIS DAN PERANCANGAN

## 3.1 Analisis

Analisis dan perancangan merupakah sebuah langkah-langkah yang dilakukan dalam pengembangan sistem. Tujuan analisis adalah untuk membuat sebuah gambaran secara umum terhadap perangkat lunak. Hasil analisis sendiri berfungsi untuk menjadi pedoman penting untuk proses desain atau perancangan perangkat lunak untuk memenuhi kebutuhan sistem yang akan dikembangkan.

### 3.1.1 Analisis Masalah

Kriptografi adalah metode yang digunakan untuk mengamankan sebuah pesan digital. Namun, sebuah algoritma kriptografi tidak hanya tentang mengamankan sebuah pesan (enkripsi dan dekripsi), tetapi sebuah algoritma kriptografi juga harus tahan terhadap serangan kriptografi agar pesan asli hanya dapat diakses ataupun dibaca oleh pihak-pihak yang terverifikasi saja.

Daripada itu, pada penelitian ini, akan dilakukan pengujian terhadap keamanan suatu algoritma kriptografi yang bernama M.S.H. Biswas (Rabin-Biswas) untuk mengetahui apakah algoritma ini merupakan sebuah algoritma yang aman untuk digunakan dalam mengamankan sebuah pesan.

### 3.1.2 Analisis Kebutuhan

Analisis kebutuh dapat dibedakan menjadi dua aspek utama, yaitu analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional. Analisis kebutuhan fungsional merincikan mengenai fitur-fitur yang berkaitan dengan hasil akhir yang ingin dituju dari

sistem yang akan dikembangkan dalam penelitian ini. Sementara itu, analisis kebutuhan non-fungsional menjelaskan fitur tambahan seperti kinerja, kemudahan pengguna dan fitur lain guna mendukung sistem yang akan dibangun menjadi lebih baik.

#### 3.1.2.1 Analisis Kebutuhan Fungsional

Apa yang perlu dicapai oleh sistem dan apa yang dilakukannya ditentukan oleh persyaratan fungsional. Uraian tentang langkah-langkah yang bisa dijalankan oleh sistem yang akan dibangun dalam penelitian ini dimasukkan dalam kebutuhan fungsional.

Berikut adalah kebutuhan fungsional pada sistem yang akan dibangun:

1. Melakukan pembangkitan kunci
   Sistem mampu melakukan pembangkitan kunci yang akan dipakai pada tahap enkripsi dan juga dekripsi. Baik itu kunci privat, kunci publik, dan kunci rahasia.
2. Menerima masukan *plaintext*
   Sistem mampu membaca masukan *string* yang dimasukkan oleh pengguna.
3. Memberi keluaran *ciphertext*
   Sitem akan menghasilkan *ciphertext* berupa *array* hasil enkripsi algoritma Rabin-Biswas.
4. Menerima masukan *ciphertext*
   Sistem mampu membaca masukan *array ciphertext* yang dimasukkan oleh pengguna.
5. Memberi keluaran *plaintext*
   Sistem akan menghasilkan *plaintext* hasil dekripsi algoritma Rabin-Biswas.
6. Enkripsi pesan
   Sistem mampu melakukan enkripsi masukan *string* yang dimasukkan oleh pengguna dengan kunci yang dihasilkan oleh sistem menggunakan algoritma Rabin-Biswas.
7. Dekripsi pesan
   Sistem mampu melakukan dekripsi *ciphertext* yang berupa *array* dengan kunci yang dihasilkan oleh sistem menggunakan algoritma Rabin-Biswas.

8. Melakukan kriptanalisis

   Sistem mampu memecahkan *ciphertext* hasil enkripsi algoritma Rabin-Biswas tanpa memiliki pengetahuan tentang kunci yang digunakan pada tahap enkripsi.

#### 3.1.2.2 Analisis Kebutuhan Non-fungsional

Kebutuhan non-fungsional ialah pernyataan yang menguraikan fitur-fitur tambahan yang mempengaruhi kinerja sistem yang akan dibangun, tetapi tidak mencakup rincian fitur-fitur pokok sistem.

Berikut adalah aspek-aspek non-fungsional yang perlu diperhatikan pada sistem yang akan dikembangkan :

1. Kinerja

   Sitem mampu melakukan proses enkripsi dengan cepat serta melakukan proses dekripsi untuk mendapatkan pesan asli secara efektif.

2. Kemudahan pengguna

   Antarmuka yang dirancang nantinya akan mudah untuk dipahami dan tidak membingungkan pengguna.

3. Perangkat keras

   Pengujian yang dilakukan dalam penelitian ini dilakukan dengan spesifikasi seperti berikut.

   - Processor : Intel® Core™ i7-9750H
   - RAM : 16 GB 2667 MHz
   - GPU : NVDIA GeForce 1660 Ti

## 3.2 Perancangan Sistem

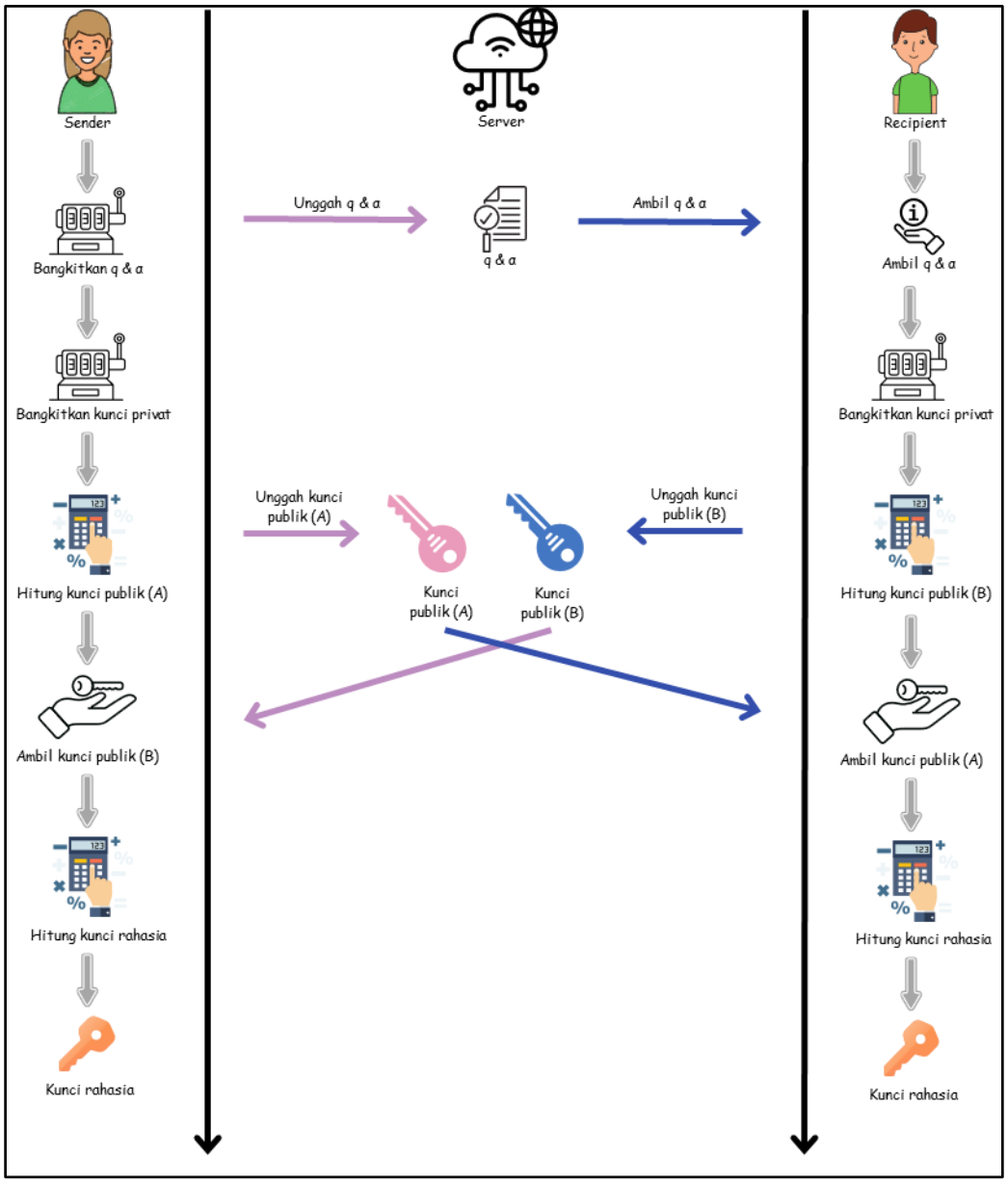
Perancangan sistem adalah proses yang dilakukan untuk merancang sistem dengan tujuan membuat sistem tersebut beroperasi dengan fokus pada meningkatkan tingkat efektivitas dan efisiensi. Dalam konteks penelitian ini, perancangan sistem akan memanfaatkan berbagai jenis diagram, termasuk diagram umum, diagram *use case*, diagram aktivitas, diagram urutan, dan diagram alir (*flowchart)*.

### 3.2.1 Diagram Umum

Diagram umum ialah gambaran umum yang mendefinisikan seperti apa sistem yang dikembangkan nantinya. Diagram umum dalam penelitian ini terbagi menjadi 3, yakni diagram umum pembangkitan kunci, enkripis dan dekripsi, serta kripatanlisis.

#### 3.2.1.1 Diagram Umum Pembangkitan Kunci

Diagram umum pada fase pembangkitan kunci ditunjukkan pada Gambar 3.1 dibawah ini.



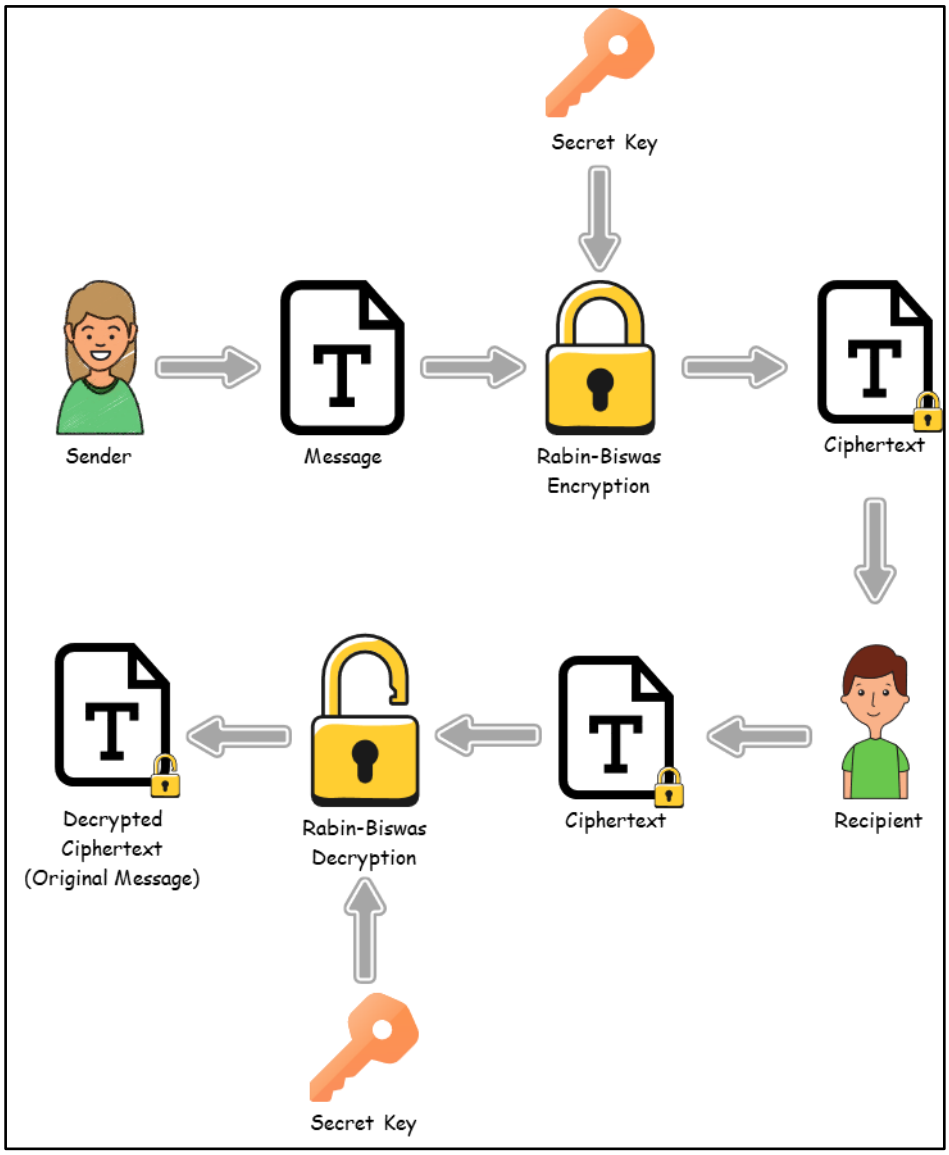

**Gambar 3.1 Diagram Umum Pembangkitan Kunci**

Pada Gambar 3.1 menunjukkan bagaimana proses pembangkitan kunci yang digunakan pada algoritma Rabin-Biswas. Berikut adalah langkah-langkah yang terjadi pada Gambar 3.1.

1. Pengirim (*sender*) akan membangkitkan $q$ dan $\alpha$ yang akan digunakan nantinya untuk seluruh proses dalam pembangkitan kunci. Lalu, *sender* akan mengunggah nilai $q$ dan $\alpha$ tersebut ke server. Hal ini dilakukan agar *recipient* dapat mengakses nilai $q$ dan $\alpha$ juga. Nilai $q$ dan $\alpha$ dibutuhkan untuk proses pembangkitan kunci publik dan kunci rahasia.
2. Kedua belah pihak, baik *sender* ataupun *recipient* akan membangkitkan kunci privat guna membangkitkan kunci publik. Setelah itu, kedua bilah pihak akan melakukan pembangkitan kunci publik menggunakan $q$, $\alpha$, dan juga kunci privat masing-masing pihak.
3. Setelah kunci publik dibangkitkan, kedua belah pihak mengunggah kunci publik mereka ke server. Setelahnya, *sender* akan mengambil kunci publik *recipient* (B). Begitu juga sebaliknya, *recipient* akan mengambil kunci publik *sender* (A). Diperlukan kunci publik untuk menghasilkan kunci pribadi yang nantinya akan dipakai dalam fase atau tahap enkripsi dan dekripsi.
4. Langkah terakhir adalah membangkitkan kunci rahasia. Kunci rahasia yang dihasilkan kedua pihak adalah sama.

### 3.2.1.2 Diagram Umum Enkripsi dan Dekripsi

Pada Gambar 3.2 ditunjukkan graf umum untuk fase atau tahap enkripsi dan dekripsi.




**Gambar 3.2 Diagram Umum Enkripsi dan Dekripsi**

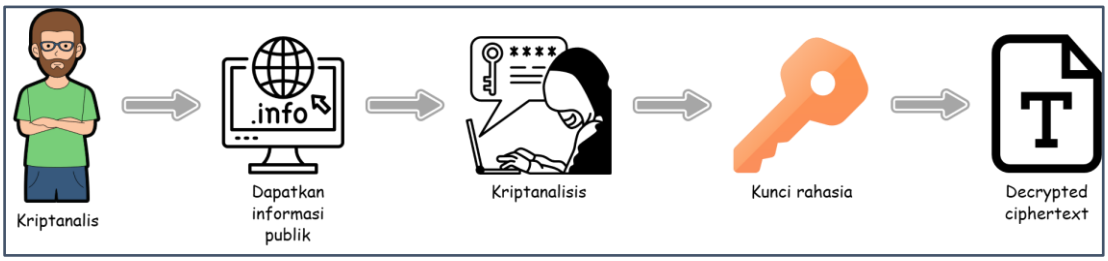
Pada Gambar 3.2 menunjukkan bagaimana fase enkripsi dan dekripsi berjalan pada sistem yang dikembangkan. Berikut adalah langkah-langkah yang terjadi pada Gambar 3.2.

1. Pihak *sender* akan memasukkan *message* yang akan dienkripsi menggunakan algoritma Rabin-Biswas. Lalu, *sender* juga akan memasukkan kunci rahasia yang telah dibangkitkan sebelumnya. Kunci rahasia tersebut akan digunakan untuk mengenkripsi *message* yang sebelumnya sudah dimasukkan.

2. Setelah *message* berhasil dienkripsi, *sender* akan mengirimkan *ciphertext* kepada *recipient* (boleh menggunakan jalur yang tidak aman).
3. Setelah menerima *ciphertext*, *recipient* akan melakukan proses dekripsi menggunakan algoritma Rabin-Biswas untuk mendapatkan *orginal message*. Proses dekripsi juga menggunakan kunci rahasia yang sama yang dibangkitkan sebelumnya.

#### 3.2.1.3 Diagram Umum Kriptanalisis

Pada Gambar 3.3 menunjukkan graf umum untuk fase atau tahap kriptanalisis.



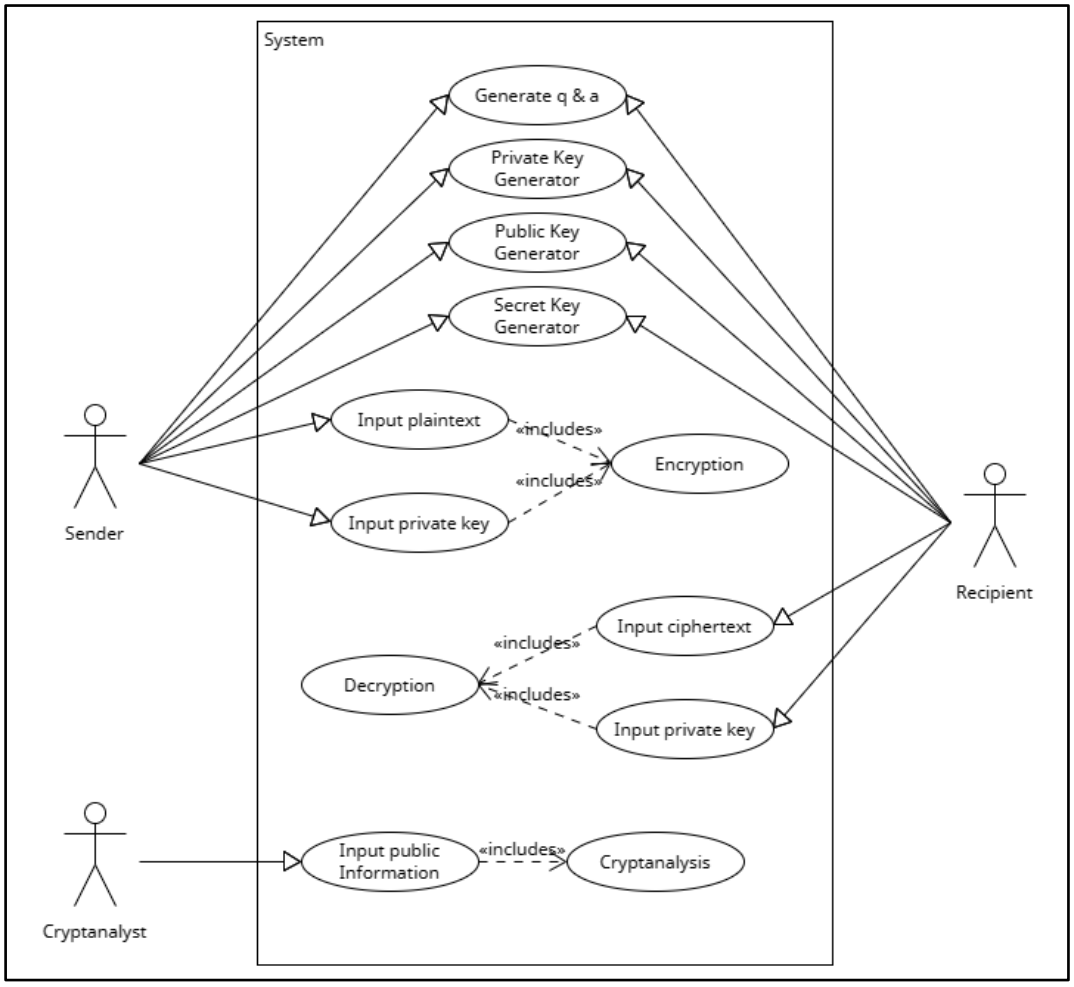

**Gambar 3.3 Diagram Umum Kriptnalisis**

Pada Gambar 3.3 menunjukkan bagaimana proses kriptanalisis akan dilakukan pada sistem yang akan dikembangkan. Berikut merupakan langkah-langkah yang ditunjunkkan pada Gambar 3.3.

1. Kriptanalisis akan mengumpulkan semua informasi publik yang digunakan pada proses enkripsi, lalu memasukkan semua informasi tersebut ke dalam sistem.
2. Dari informasi publik tersebut akan dilakukan proses kriptanalisis untuk melakukan dekripsi pada *ciphertext* tanpa memiliki pengetahuan tentang kunci yang dipakai pada tahap enkripsinya.

### 3.2.2 *Use Case Diagram*

*Use case diagram* merupakan jenis graf pada pemodelan perangkat lunak yang digunakan untuk memvisualisasikan hubungan antara sistem perangkat lunak dan berbagai aktor (pengguna atau entitas eksternal lainnya) yang terlibat dalam sistem. *Use case diagram* membantu mengidentifikasi, mendeskripsikan, serta memahami bagaimana sistem yang akan dibangun beroperasi dalam konteks situasi dunia nyata. *Use case diagram* untuk sistem yang akan dikembangkan pada penelitian ini digambarkan pada Gambar 3.4.




**Gambar 3.4 Use Case Diagram**

Pada Gambar 3.4 terdapat tiga aktor yang dapat menjalankan sistem yang akan dibangun nantinya, yakni *sender*, *recipient*, dan juga kriptanalis. *Sender* dan *recipient* dapat melakukan proses pembangkitan kunci pada sistem, baik itu pembangkitan q dan α, pemabangkitan kunci privat, pembangkitan kunci publik, maupun pembangkitan kunci rahasia yang akan dipakai di tahap enkripsi dan dekripsi pesan. *Sender* dapat menjalankan sistem dengan memasukkan *plaintext* serta kunci rahasia yang telah dibangkitkan pada tahap pembangkitan kunci untuk melakukan tahap enkripsi menggunakan algoritma Rabin-Biswas pada *plaintext* sehingga sistem dapat mengkonversi *plaintext* menjadi bentuk *ciphertext*. *Recipent* dapat mengembalikan *ciphertext* menjadi ke bentuk semula (*plaintext*) dengan memasukkan *ciphertext* dan kunci rahasia sehingga sistem dapat menjalankan proses dekripsi *ciphertext* menggunakan algoritma Rabin-Biswas. *Cryptanalyst* dapat melakukan proses kriptanalisis pada sistem hanya dengan menginputkan informasi publik yang digunakan

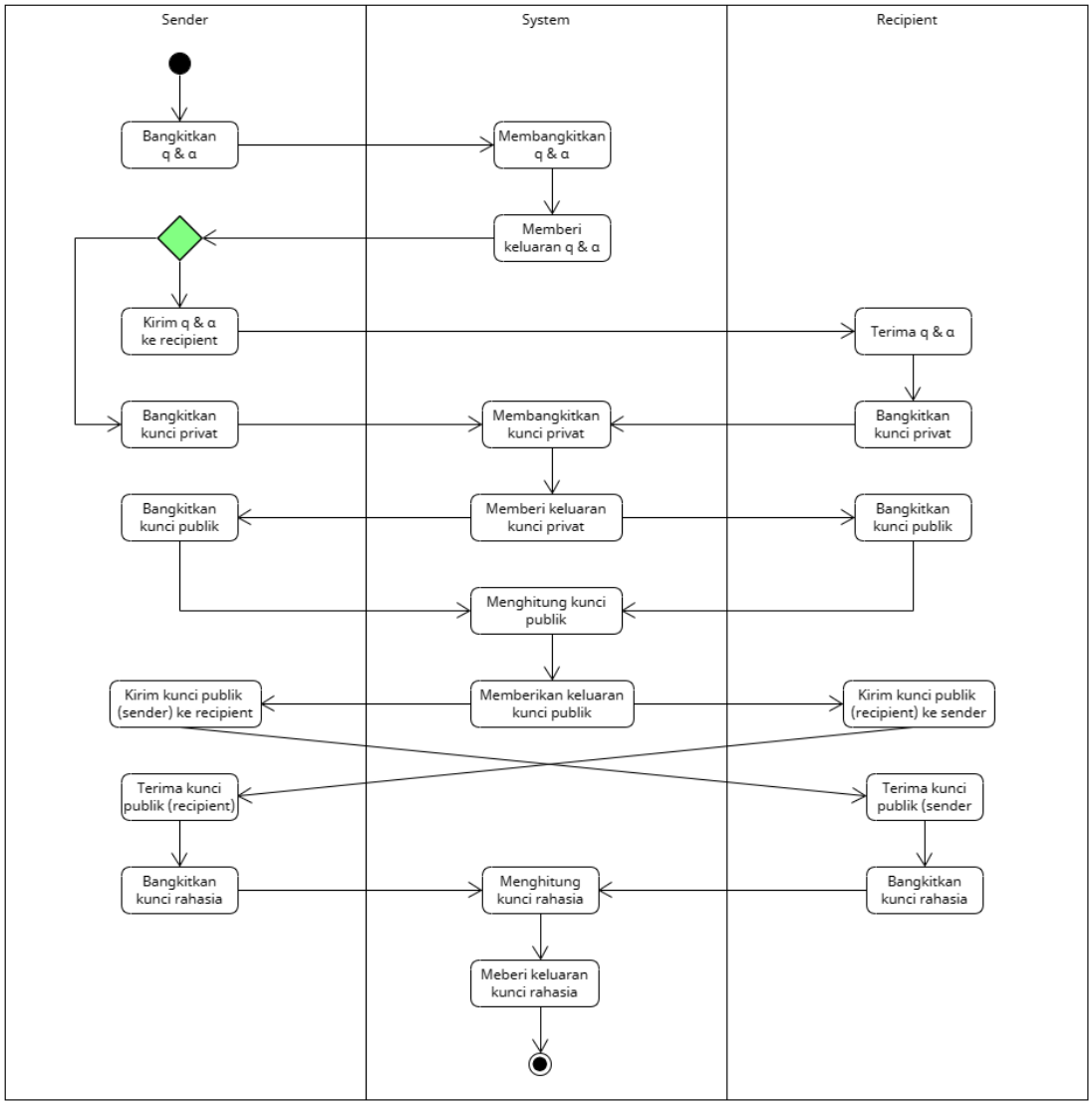
pada proses enkripsi untuk dapat memecahkan *ciphertext* hasil enkripsi algoritma Rabin-Biswas tanpa mengetahui kunci yang digunakan pada proses enkripsinya.

### 3.2.3 *Activity Diagram*

*Activity diagram* ialah graf pada pemodelan perangkat lunak yang digunakan untuk mengilustrasikan aliran aktivitas dalam suatu sistem atau tahapan. Diagram ini memberikan bantuan dalam memvisualisasikan urutan langkah-langkah atau aktivitas yang dilakukan pada suatu proses. Pada sistem yang dikembangkan pada penelitian ini, terdapat empat buah *activity diagram* yakni *activity diagram* pada tahap pembangkitan kunci, *activity diagram* pada tahap enkripsi, *activity diagram* tahap dekripsi, dan juga *activity diagram* pada tahap kriptanalisis.

#### 3.2.3.1 *Activity Diagram* Pembangkitan Kunci

Pada Gambar 3.5 menggambarkan graf aktivitas pada fase pembangkitan kunci yang digunakan di sistem yang dibuat pada penelitian ini. *Acitivity diagram* di fase ini memiliki tiga buah kotak, yakni kotak paling kiri menunjukan aktivitas yang dilakukan *sender* pada sistem, lalu kotak tengah menunjukkan respon apa yang saja yang dilakukan oleh sistem, dan kotak paling kanan menunjukkan apa yang dilakukan *recipient* pada sistem.



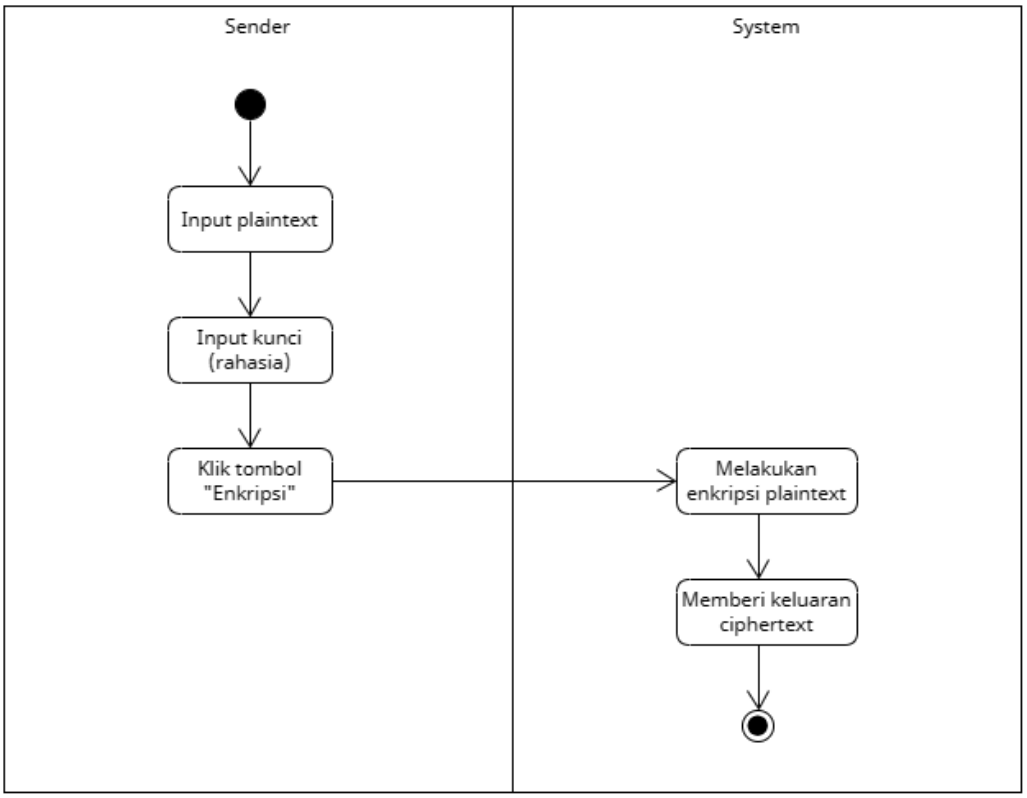

**Gambar 3.5 *Activity Diagram* Pembangkitan Kunci**

*Sender* akan melakukan pembangkitan q dan α sehingga sistem akan merespon untuk melakukan proses tersebut dan mengembalikan nilai q dan α yang telah dibangktikan kepada *sender*. Lalu nilai q dan α tersebut akan dikirim oleh *sender* ke *recipient* yang akan digunakan untuk proses pembangkitan kunci publik masing-masing pihak. Setelah *recipient* menerima q dan α, kedua belah pihak (*sender* dan *recipient*) akan membangkitkan kunci privat mereka masing-masing dan dilanjutkan pembangkitan kunci publik masing-masing pihak. Sistem akan melakukan proses pembangkitan kunci publik lalu mengembalikannya ke masing-masing pihak. Setelah kunci publik dibangkitkan, kedua belah pihak melakukan pertukaran kunci publik mereka. Kunci publik ini akan dipakai untuk fase pembangkitan kunci rahasia. Sistem

akan melakukan proses pemabangkitakn kunci rahasia dan mengembalikan keluaran sebuah kunci rahasia yang akan dipakai di fase enkripsi dan dekripsi.

### 3.2.3.2 *Activity Diagram* Enkripsi

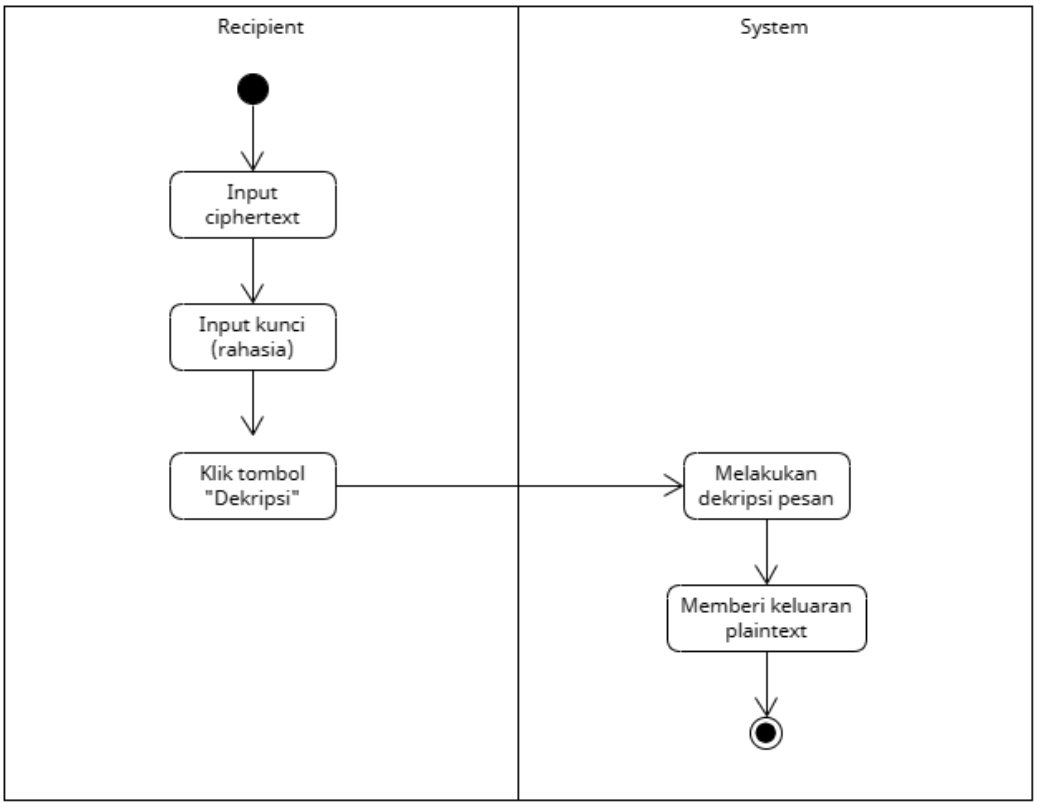
Pada Gambar 3.6 menunjukkan fase aktiviats pada fase penyandian pesan yang digunakan pada sistem yang dikembangkan pada penelitian ini. *Activity diagram* di fase ini memiliki dua buah kotak, yakni kotak sebelah kiri menunjukkan aktivitas yang dilakukan *sender* pada sistem dan kota sebelah kanan menunjukkan respon apa yang dilakukan oleh sistem. *Sender* akan memasukkan *plaintext* yang akan dienkripsi menggunakan algoritma Rabin-Biswas beserta kunci rahasia. Sistem akan merespon dengan melakukan proses enkripsi terhadap *plaintext* yang dimasukkan oleh *sender* sebelumnya. Sistem akan memberi keluaran *ciphertext* hasil enkripsi menggunakan algoritma Rabin-Biswas.




**Gambar 3.6 *Activity Diagram* Enkripsi**

### 3.2.3.3 *Activity Diagram* Dekripsi

Pada Gambar 3.7 menunjukkan graf aktivitas pada fase dekripsi pesan yang digunakan pada sistem yang dibuat pada penelitian ini. *Activity diagram* pada fase ini memiliki 2 buah kotak, yakni kotak sebelah kiri menunjukkan aktivitas yang dilakukan *recipient* pada sistem dan kotak sebelah kanan menujukkan respon apa saja yang dilakukan oleh sistem. *Recipient* akan memasukkan *ciphertext* dan kunci rahasia untuk memulai fase dekripsi. Sistem akan melakukan fase menafsirkan *ciphertext* tersebut menggunakan algoritma Rabin-Biswas. Setelah selesai, sistem akan mengeluarkan sebuah *plaintext*.

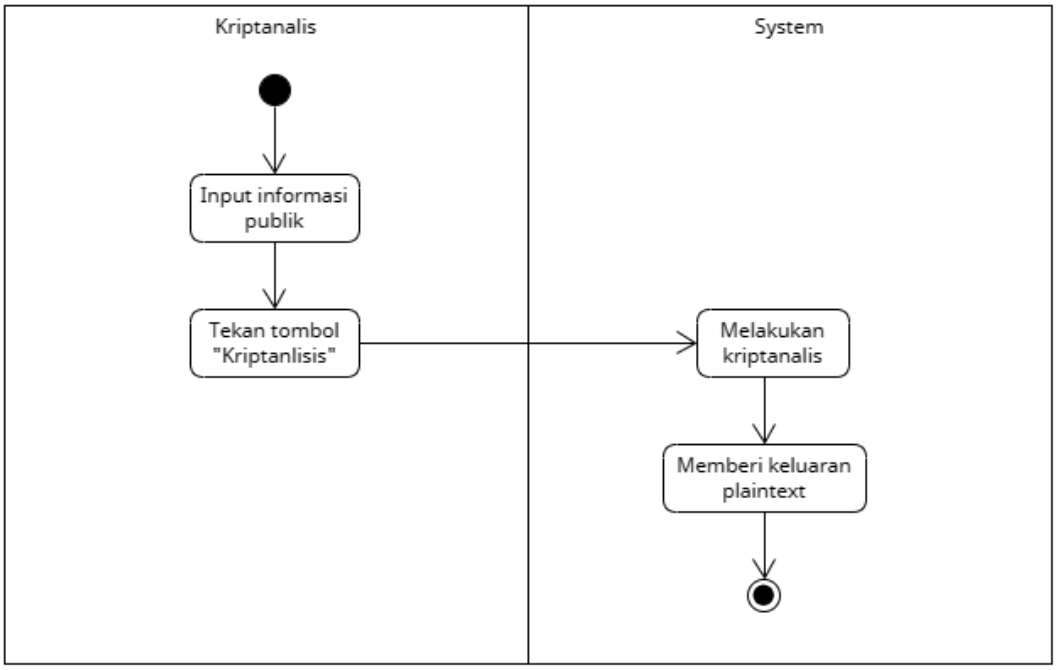



**Gambar 3.7 *Activity Diagram* Dekripsi**

### 3.2.3.4 Activity Diagram Kriptanalisis

Pada Gambar 3.8 menunjukkan graf aktivitas pada fase kriptanalisis yang digunakan pada sistem yang dikembangkan pada penelitian ini. *Activity diagram* di fase ini memiliki dua buah kotak, yakni sebelah kiri menunjukkan aktivitas yang dilakukan oleh kriptanalis pada sistem dan kotak sebelah kanan menunjukkan respon apa yang dilakukan oleh sistem. Kriptanalisi harus memasukkan informasi publik yang didapat dari proses pembangkitan kunci dan enkripsi sebelumnya. Setelah itu sistem akan

merespon dengan melakukan proses kriptanalisis menggunakan informasi yang dimasukkan oleh kriptanalis untuk memecahkan *ciphertext* dan mendapatkan *plaintext*. Setalah proses kriptanalisis selesai, sistem akan memberikan keluaran berupa *plaintext*.



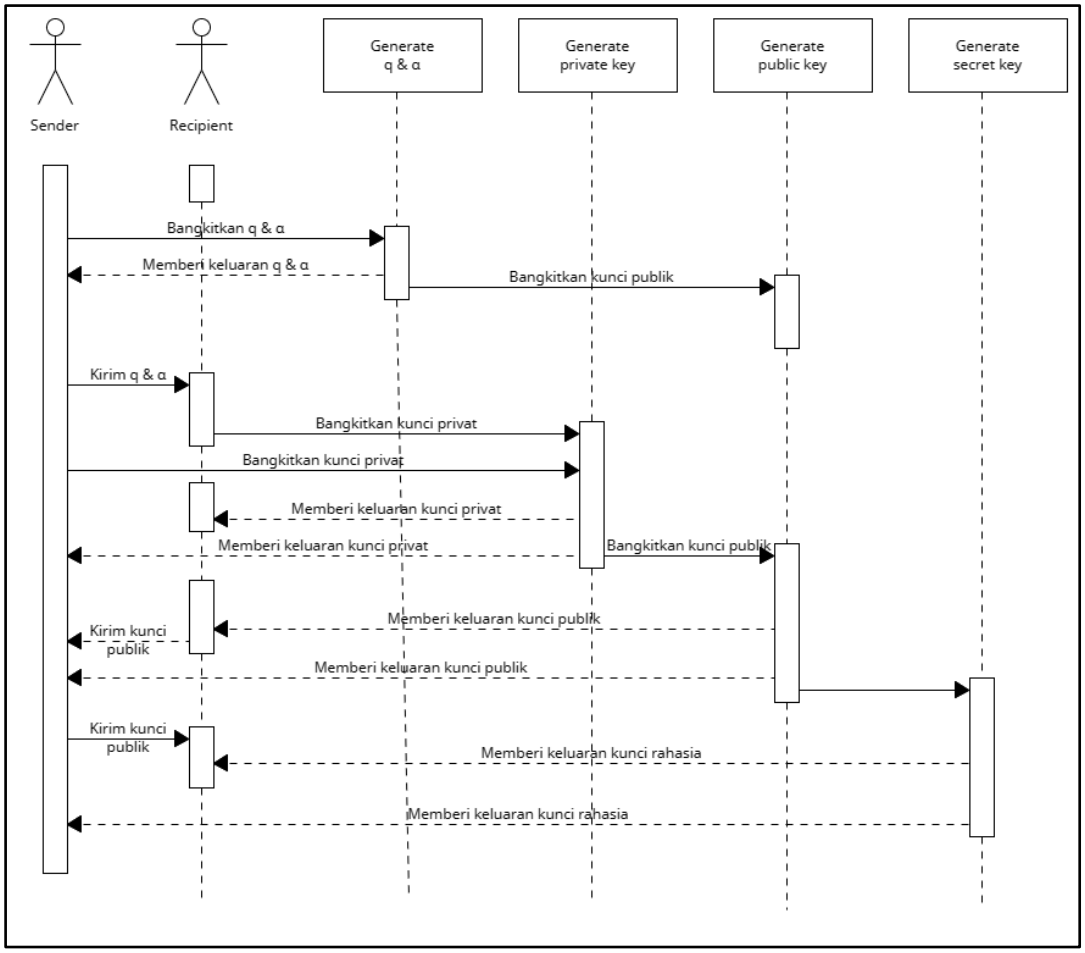

**Gambar 3.8 *Activity Diagram* Kriptanalisis**

### 3.2.4 *Sequence Diagram*

Diagram urutan ialah graf pemodelan perangkat lunak yang digunakan untuk memvisualisasikan hubungan antar objek dalam suatu sistem dalam urutan waktu tertentu. Diagram ini memvisualisasikan bagaimana objek berhubungan satu dengan lainnya dan berinteraksi dalam konteks *use case*. Pada penelitian ini, sistem yang dibuat memiliki empat diagram urutan, yakni diagram urutan pembangkitan kunci, diagram urutan enkripsi, diagram urutan dekripsi, dan diagram urutan kriptanalis.

### 3.2.4.1 *Sequence Diagram* Pembangkitan Kunci

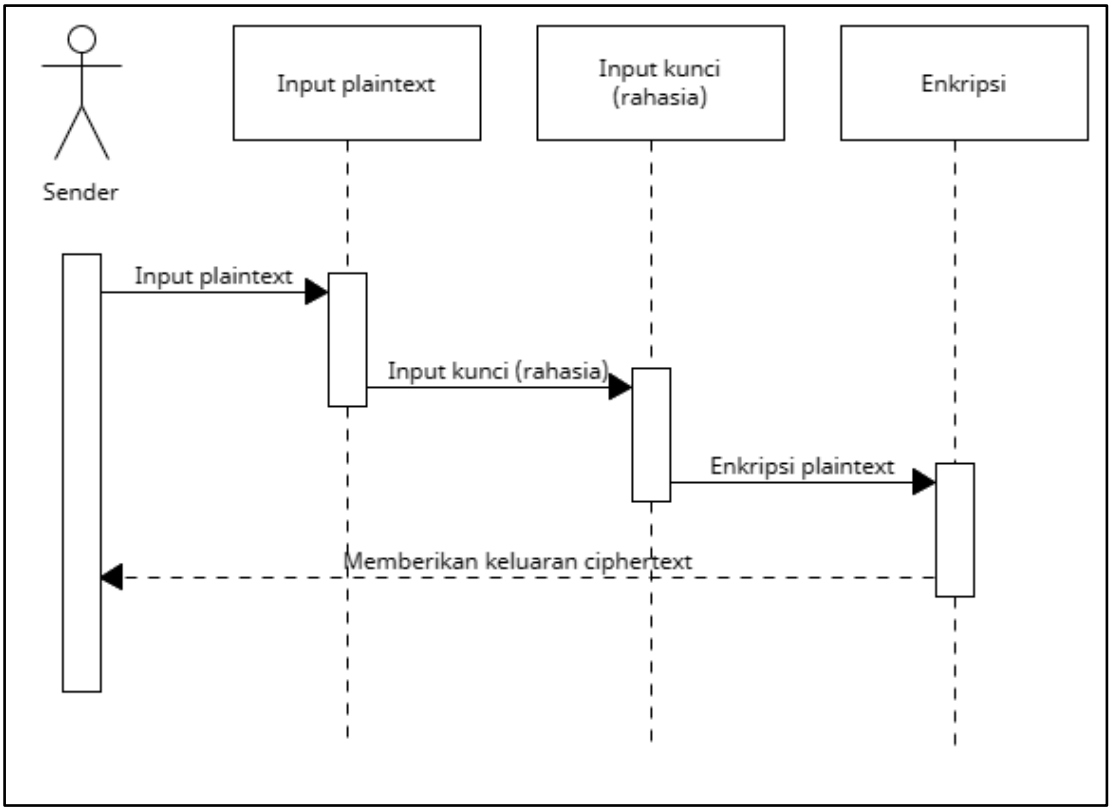
Pada Gambar 3.9 menunjukkan bagaimana diagram urutan unutk interaksi antara *sender, recipient*, dan sistem dalam fase pembangkitan kunci.




**Gambar 3.9 *Sequence Diagram* Pembangkitan Kunci**

### 3.2.4.2 *Sequence Diagram* Enkripsi

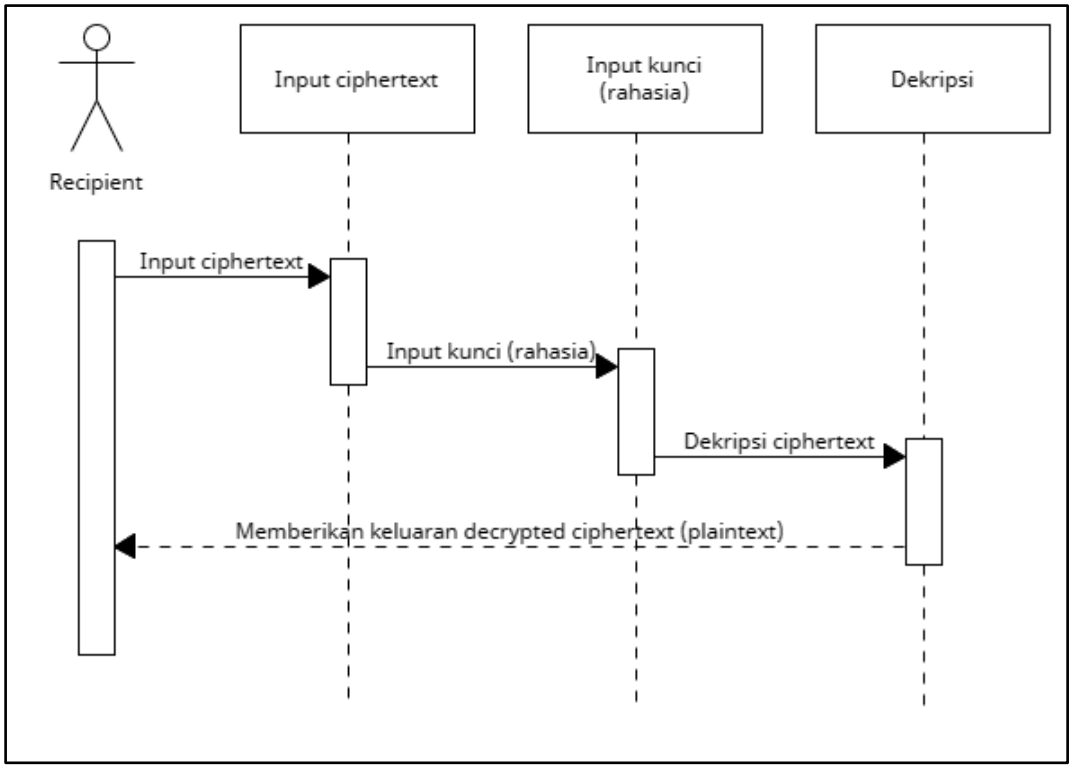
Pada Gambar 3.10 menunjukkan bagaimana *sequence diagram* untuk interaksi antara *sender* dan sistem pada fase penyandian.




**Gambar 3.10 *Acitivity Diagram* Enkripsi**

### 3.2.4.3 *Sequence Diagram* Dekripsi

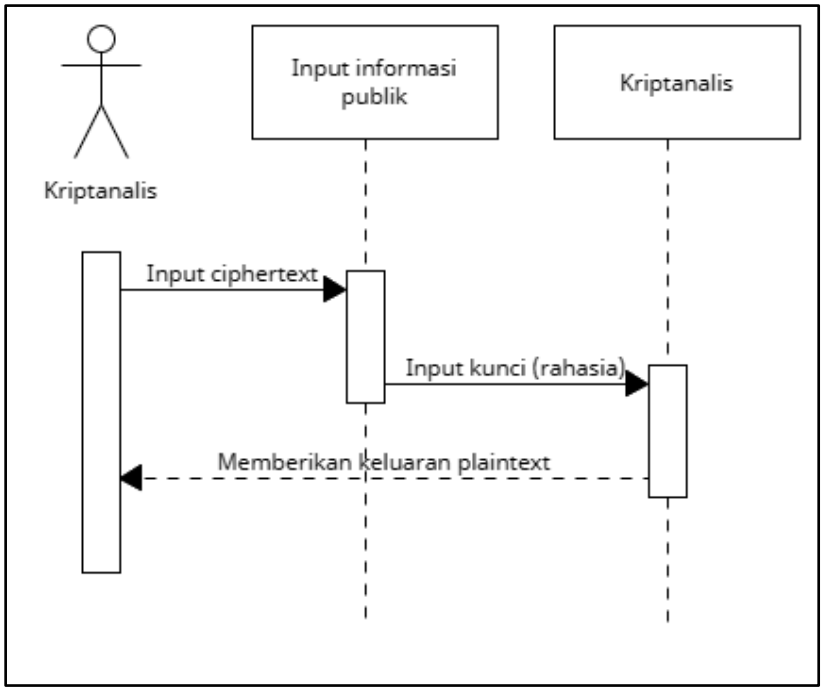
Pada Gambar 3.11 menunjukkan bagaimana diagram urutan untuk interaksi antara *recipient* dan sistem pada fase dekripsi.




**Gambar 3.11 *Sequence Diagram* Dekripsi**

#### 3.2.4.4 *Sequence Diagram* Kriptanalisis

Pada Gambar 3.12 menunjukkan bagaimana *sequence diagram* untuk interaksi antara kriptanalis dan sistem pada fase kriptanalisis.




**Gambar 3.12 Sequence Diagram Kriptanalisis**

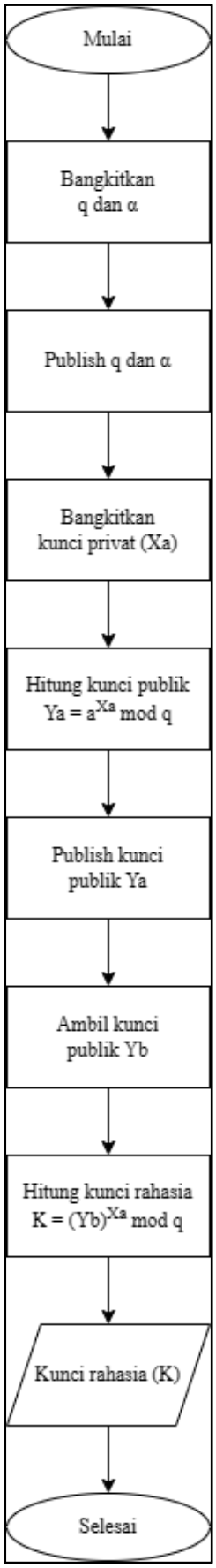
### 3.2.5 Diagram alir (*flowchart*)

*Flowchart* adalah representasi visual dari proses yang digunakan dalam suatu proses atau sistem. Diagram alir digunakan untuk memodelkan, merancang, dan mendokumentasikan alur kerja, algoritma, atau proses dengan cara yang mudah dimengerti bagi pihak yang terlibat dalam proses tersebut. Diagram alir pada penelitian ini terdiri dari diagram alir pembangkitan kunci (*sender*), diagram alir pembangkitan kunci (*recipient*), diagram alir enkripsi, diagram alir dekripsi, dan diagram alir kriptanalisis.

#### 3.2.5.1 Diagram Alir Pembangkitan Kunci (*Sender*)

Pada Gambar 3.13 menunjukkan diagram alir untuk fase atau proses pembangkitan kunci untuk pihak *sender*. Diagram alir ini menjelaskan bagaimana nantinya proses pembangkitan kunci (*sender*) terjadi pada sistem yang akan dibangun. Fase

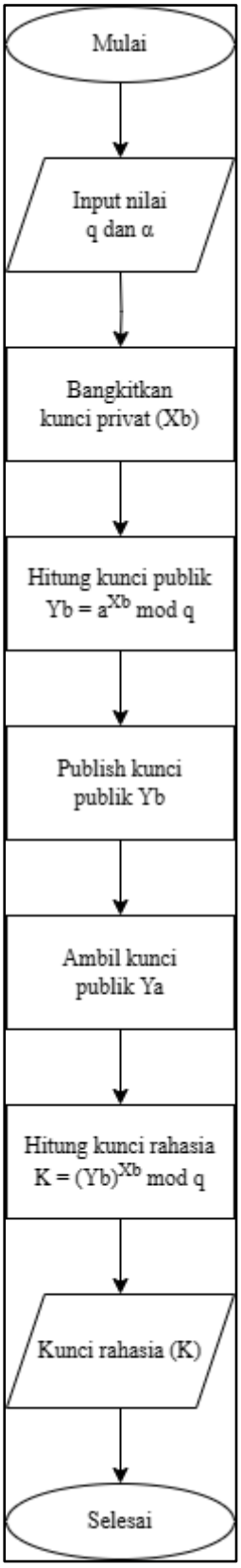
pembangkitan kunci diawali dengan pembangkitan nilai $q$ dan $\alpha$. Setelah itu, kirim nilai $q$ dan $\alpha$ tersebut ke pihak *recipient*. Lalu bangkitkan kunci privat yang akan dipakai untuk pembangkitan kunci publik. Bangkitkan kunci publik menggunakan nilai $q$, $\alpha$, dan kunci privat yang telah dibangkitkan sebelumnya. Setelah kunci publik dibangkitkan, lakukan pertukaran kunci publik dengan *recipient*. Bangkitkan kunci rahasia menggunakan $q$, $\alpha$, dan kunci publik milik *recipient*. Dan keluaran dari fase ini adalah kunci rahasia.




**Gambar 3.13 Diagram Alir Pembangkitan Kunci (*Sender*)**

#### 3.2.5.2 Diagram Alir Pembangkitan Kunci (*Recipient*)

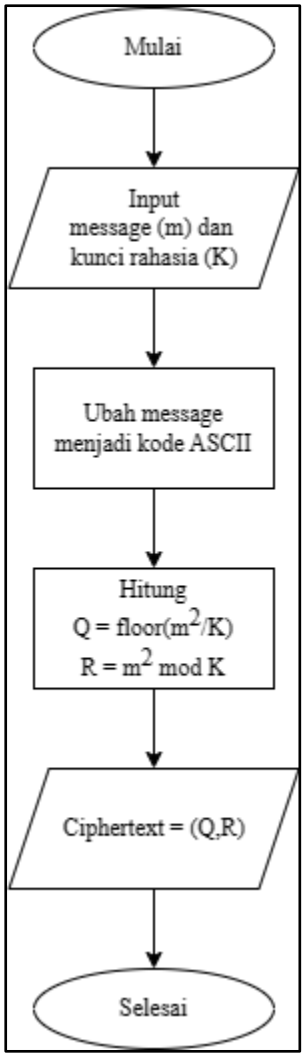
*Flowchart* untuk fase atau tahap pembangkitan kunci untuk pihak *recipient* digambarkan pada Gambar 3.14. Diagram alir ini menjelaskan bagaimana nantinya proses pembangkitan kunci (*recipient*) terjadi pada sistem yang akan dibangun. Fase pembangkitan kunci diawali dengan memasukkan nilai q dan α yang dikirim oleh *sender*. Lalu bangkitkan kunci privat yang akan dipakai untuk pembangkitan kunci publik. Bangkitkan kunci publik menggunakan nilai q, α, dan kunci privat yang telah dibangkitkan sebelumnya. Setelah kunci publik dibangkitkan, lakukan pertukaran kunci publik dengan *sender*. Bangkitkan kunci rahasia menggunakna q, α, dan kunci publik milik *sender*. Dan keluaran dari fase ini adalah kunci rahasia.




**Gambar 3.14 Diagram Alir Pembangkitan Kunci (*Recipient*)**

### 3.2.5.3 Diagram Alir Enkripsi

Pada Gambar 3.15 menjelaskan *flowchart* untuk fase enkripsi. Diagram alir ini menjelaskan bagaimana nantinya proses enkripsi terjadi pada sistem yang akan dibangun. Proses enkripsi diawali dengan memasukkan *plaintext* dan juga kunci yang telah dibangkitkan sebelumnya. Lalu *plaintext* akan diubah terlebih dahulu ke dalam bentuk kode ASCII. Setelah itu akan dilakukan proses enkripsi menggunakan algoritma Rabin-Biswas. Setelah proses enkripsi selesai, keluaran dari fase ini adalah sepasang *ciphertext* (Q, R).

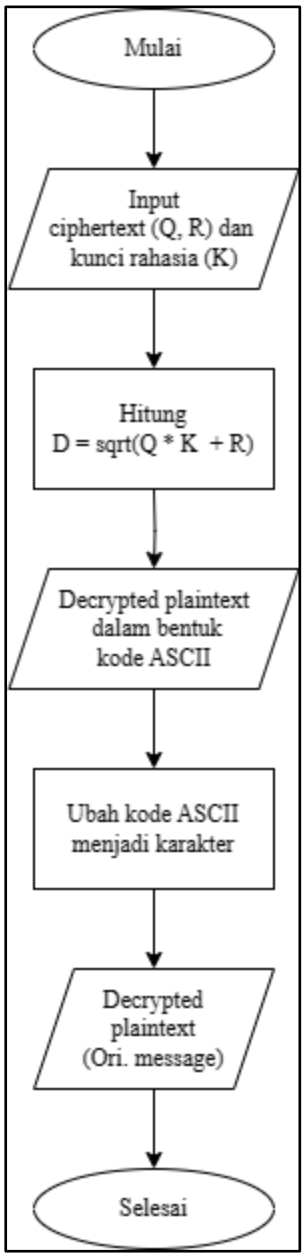



**Gambar 3.15 Diagram Alir Enkripsi**

### 3.2.5.4 Diagram Alir Dekripsi

*Flowchart* pada fase atau proses dekripsi digambarkan pada Gambar 3.16. Diagram alir ini menjelaskan bagaimana nantinya proses dekripsi terjadi pada sistem yang akan

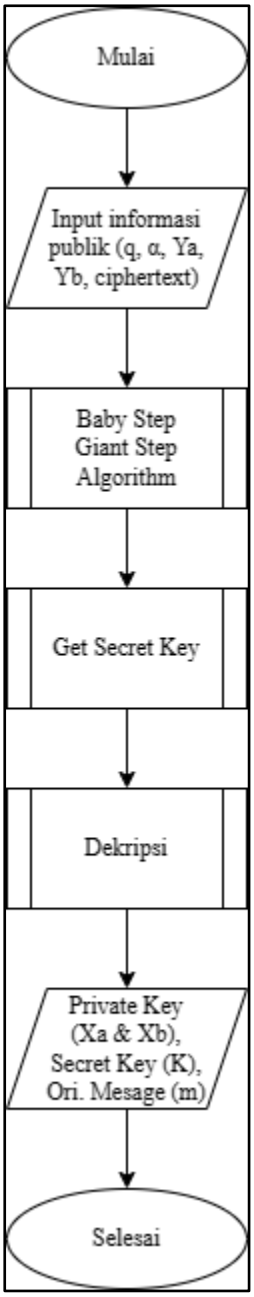
dibangun. Proses dekripsi diawali dengan memasukkan *ciphertext* dan kunci rahasia yang telah dibangkitkan sebelumnya. Lalu akan dimulai fase dekripsi *ciphertext* tersebut menggunakan algoritma Rabin-Biswas. Keluaran dari proses ini adalah *plaintext* yang masih dalam bentuk kode ASCII. Untuk itu dilakukan pengkonversian kode ASCII menjadi bentuk simbol atau karakter. Setelah proses konversi kode ASCII selesai, keluar akhirnya adalah *plaintext*.




**Gambar 3.16 Diagram Alir Dekripsi**

### 3.2.5.5 Diagram Alir Kriptanalisis

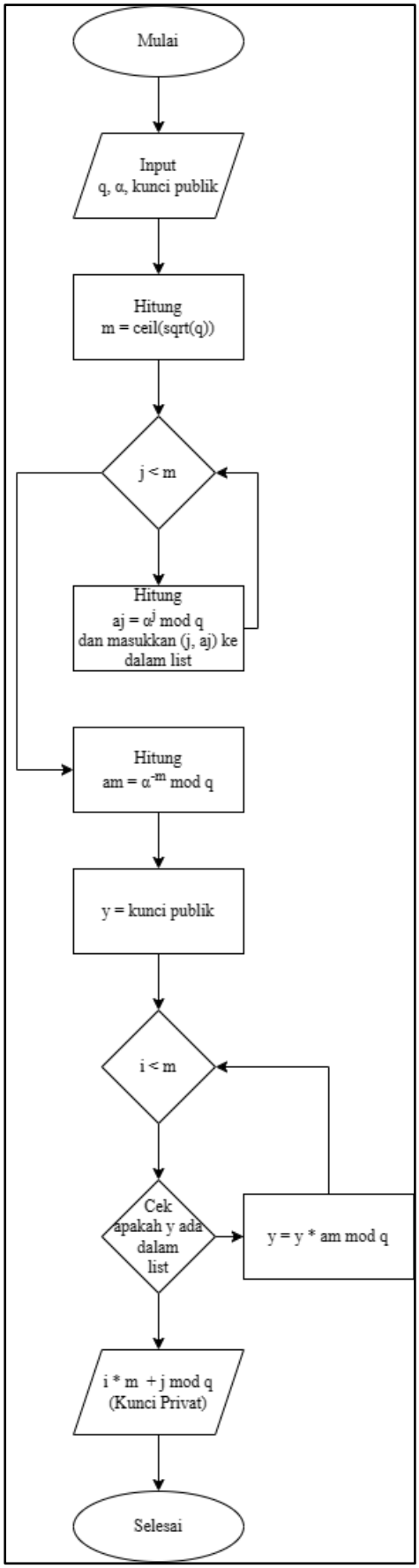
*Flowchart* untuk proses kriptanalisis digambarkan pada Gambar 3.17. *Flowchart* ini menjelaskan bagaimana nantinya proses kriptanalisis berjalan pada sistem dikembangkan. Masukkan yang dibutuhkan di proses ini ada empat, yakni $q, \alpha, Ya, Yb,$ dan *ciphertext*. Lalu akan dilanjutkan pada ketiga sub program, yaitu sub program *Baby Step Giant Step Algorithm*, sub program *Get Secret Key*, dan sub program Dekripsi. Hingga keluaran yang diberikan adalah dua buah kunci privat milik *sender* dan *recipient*, kunci rahasia dan juga pesan asli.




**Gambar 3.17 Diagram Alir Kriptanalisis**

#### 3.2.5.6 Diagram Alir *Baby Step Giant Step Algorithm*

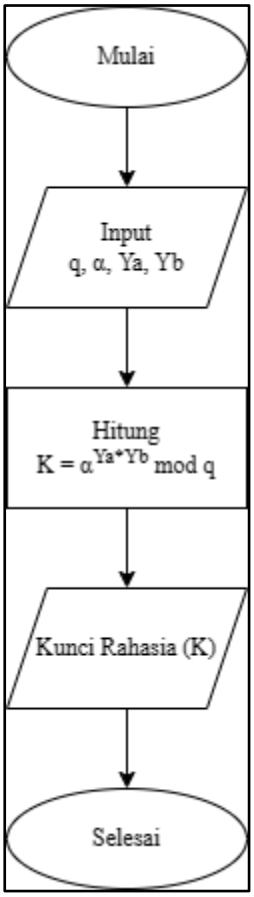
*Flowchart* untuk sub program dari diagram alir kriptanalisis digambarkan pada Gambar 3.18, yakni *Baby Step Giant Step Algorithm*. Diagram ini menjelaskan bagaimana proses algoritma *Baby Step Giant Step* bekerja nantinya pada sistem yang dikembangkan. Masukan yang dibutuhkan di tahapan ini ada tiga, yaitu $q$, $\alpha$, dan kunci publik. Lalu dilanjutkan dengan menghitung $m$ dimana $m$ akan digunakan sebagai paramater untuk perulangan yang akan dilakukan pada langkah selanjutnya. Lalu lakukan *for loop* dengan paramater $j$, ketika $j$ lebih kecil dari $m$, hitung $aj = \alpha^j \ mod \ q$ dan masukkan $(j, \ aj)$ ke dalam sebuah list. Setelah *for loop* selesai, hitung $am = a^{-m} \ mod \ q$ dan tetapkan nilai $y$ sama dengan nilai masukan kunci publik. Lalu lakukan *for loop* dengan paramater $i$, ketika $i$ lebih kecil dari $m$, cek apakah $y$ ada dalam list. Jika $y$ ada dalam list, kembalikan nilai $i * m + j \ mod \ q$. Jika $y$ tidak ada dalam list, perbaharui nilai $y$ dengan rumus $y = y * am \ mod \ q$.




**Gambar 3.18 Diagram Alir *Baby Step Giant Step Algorithm***

### 3.2.5.7 Diagram Alir Get Secret Key

Pada Gambar 3.19 menunjukkan diagram alir untuk sub program dari diagram alir kriptanalisis, yakni *Get Secret Key*. Diagram alir ini menjelaskan bagaimana proses untuk mendapatkan kunci privat pada tahapan kriptanalisis. Masukkan dari proses ini ada empat, yaitu $q$, $\alpha$, $Ya$, dan $Yb$. Lalu dilanjutkan tahap penghitungan kunci privat dengan rumus $K = \alpha^{Ya*Yb} \ mod \ q$. Dan akan memberikan keluaran kunci privat ($K$).




**Gambar 3.19 Diagram Alir *Get Secret Key***

# BAB 4

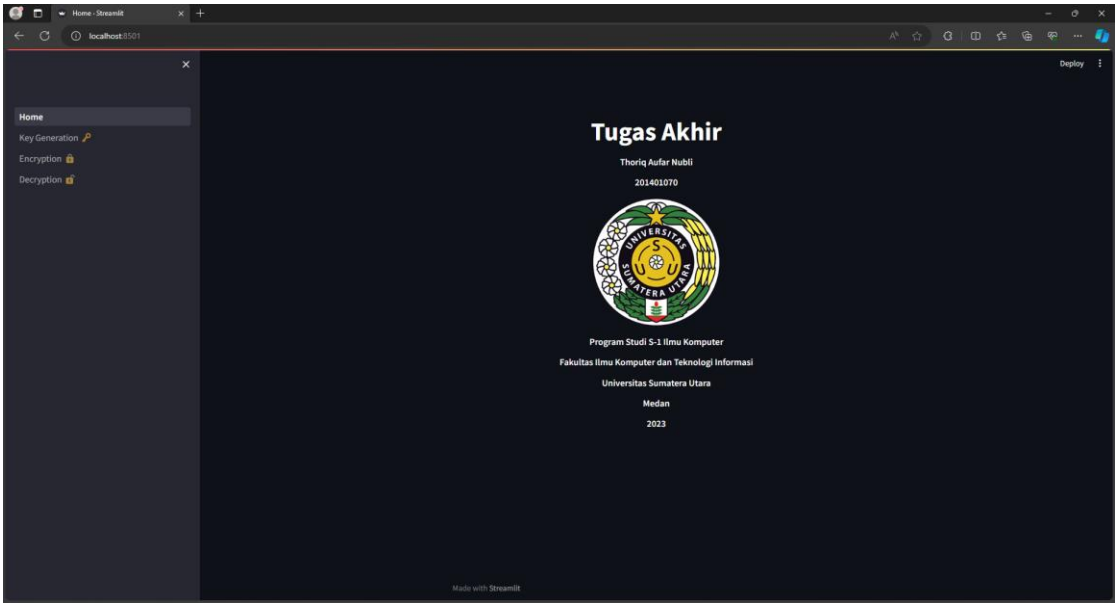
# IMPLEMENTASI DAN PENGUJIAN

## 4.1 Implementasi Sistem

Dalam penelitian ini, sistem dikembangkan menggunakan bahasa pemrograman Python sebagai *back-end* dan *framework* streamlit sebagai *front-end* dengan *Visual Studio Code* sebagai IDE. Pada sistem yang dikembangkan memiliki 4 laman, yakni laman *Home*, laman Pembangkitan Kunci, laman Enkripsi, dan laman Dekripsi.

### 4.1.1 Laman *Home*

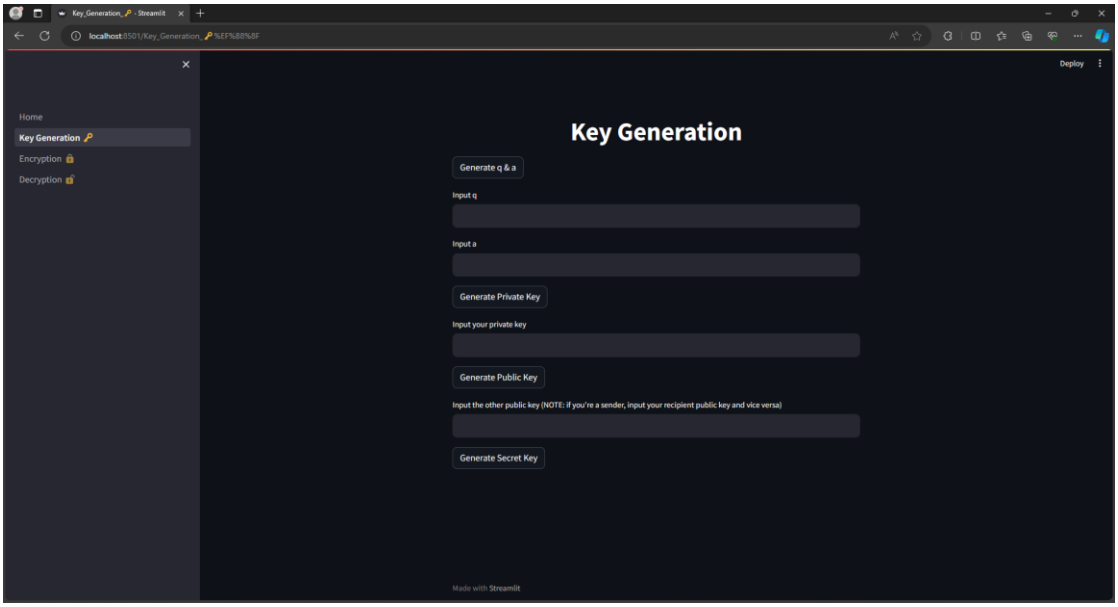
Laman *Home* adalah laman yang ditampilkan pertama kali ketika website dibuka. Laman *Home* ditunjukkan pada Gambar 4.1.




**Gambar 4.1 Laman *Home***

### 4.1.2 Laman Pembangkitan Kunci

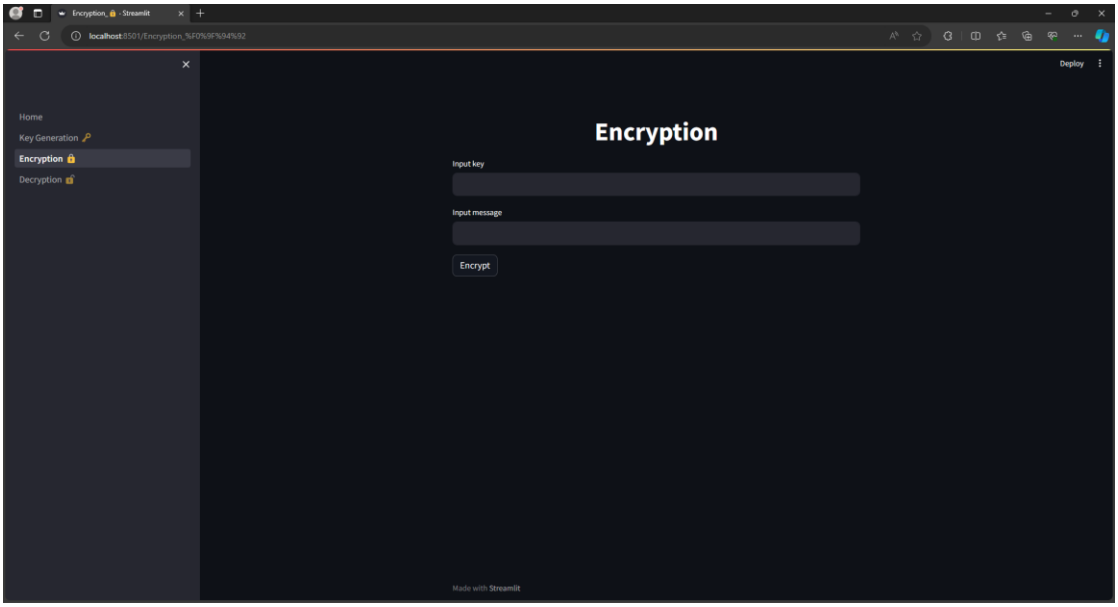
Pada Gambar 4.2 menampilkan laman Pembangkitan Kunci. Halaman ini diakses oleh kedua belah pihak yang terlibat dalam kriptografi untuk pembangkitan kunci yang akan dipakai di fase enkripsi (penyandian pesan) dan dekripsi (penafsiran *ciphertext)*.




**Gambar 4.2 Laman Pembangkitan Kunci**

### 4.1.3 Laman Enkripsi

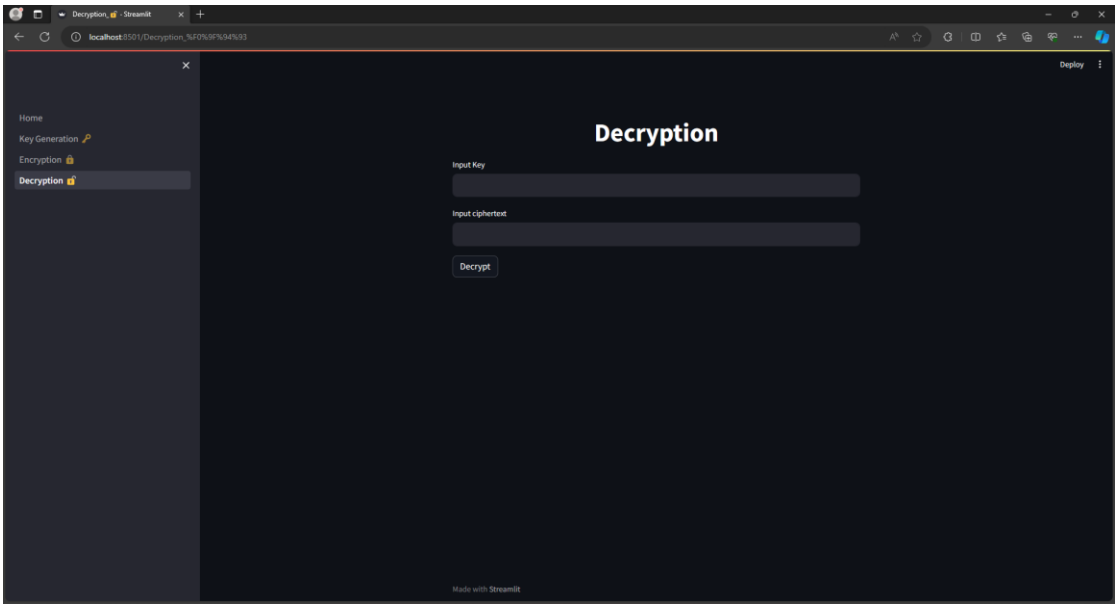
Pada Gambar 4.3 menunjukkan laman Enkripsi atau penyandian pesan. Halaman ini diakses oleh *sender* untuk melakukan enkripsi terhadap pesan yang ingin dikirimkan ke *recipient.* Keluaran dari halaman ini adalah *ciphertext*.




**Gambar 4.3 Laman Enkripsi**

### 4.1.4 Laman Dekripsi

Pada Gambar 4.4 menunjukkan laman Dekripsi atau penafsiran *ciphertext*. Halaman ini diakses oleh *recipient* untuk melakukan dekripsi terhadap *ciphertext* yang didapat dari *sender*. Keluaran dari halaman ini adalah pesan asli.



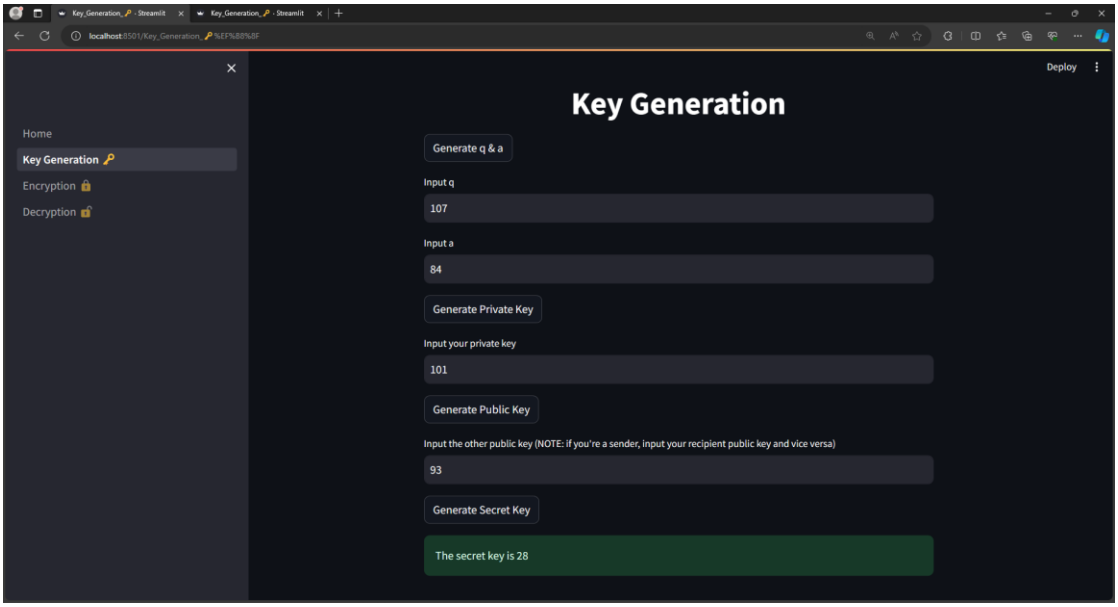

**Gambar 4.4 Laman Dekripsi**

## 4.2 Pengujian Sistem

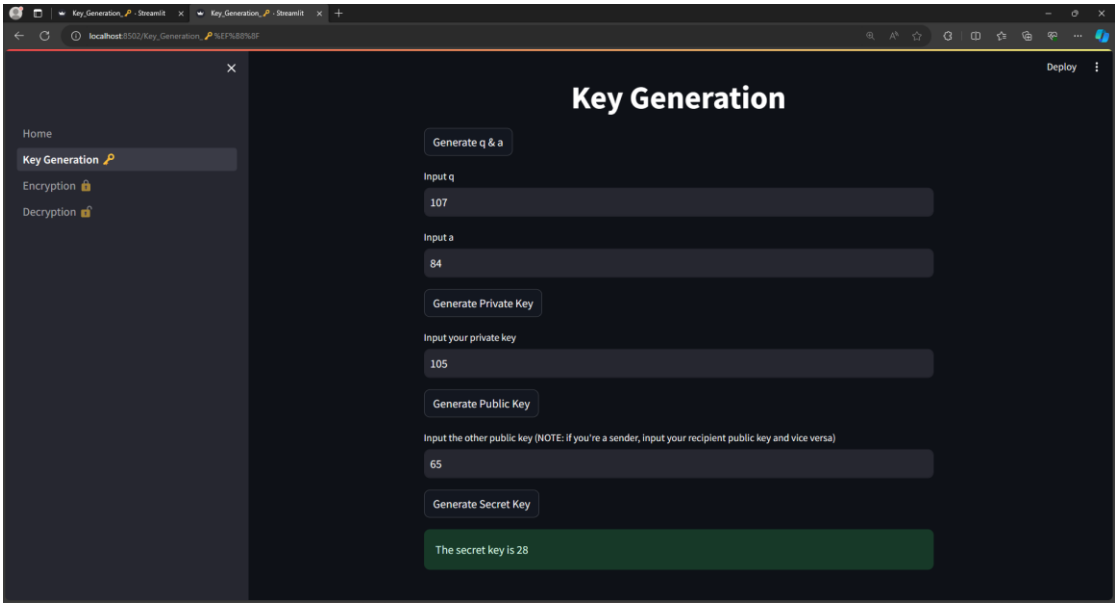
Dalam fase ini, sistem yang berhasil dikembangkan akan diuji guna memastikan bahwa sistem dapat melakukan enkripsi pada pesan serta mengembalikannya menjadi kebentuk semula dengan menggunakan algoritma Rabin-Biswas.

### 4.2.1 Pengujian Pembangkitan Kunci

Dakan tahap ini dimulai dengan menekan tombol *"Generate q & a"* untuk membangkitkan $q$ dan $\alpha$. Unggah $q$ dan $\alpha$ ke server agar pihak lain yang terlibat dalam kriptografi dapat mengambilnya. Setelah itu masukkan nilai $q$ dan $\alpha$ yang telah dibangkitkan ke kolom *"Input q"* dan *"Input a"*. Lalu tekan tombol *"Generate Private Key"* untuk membangkitkan kunci privat dan masukkan kunci privat yang telah dibangkitkan ke kolom *"Input your private key"*. Selanjutnya bangkitkan kunci publik dengan menekan tombol *"Generate Public Key"*. Unggah kunci publik ke server dan ambil kunci publik pihak lain yang terlibat dalam kriptografi untuk melakukan pembangkitan kunci rahasia. Lakukan pembangkitan kunci rahasia dengan menekan tombol *"Generate Secret Key"*.




**Gambar 4.5 Pembangkitan Kunci (Pengirim)**



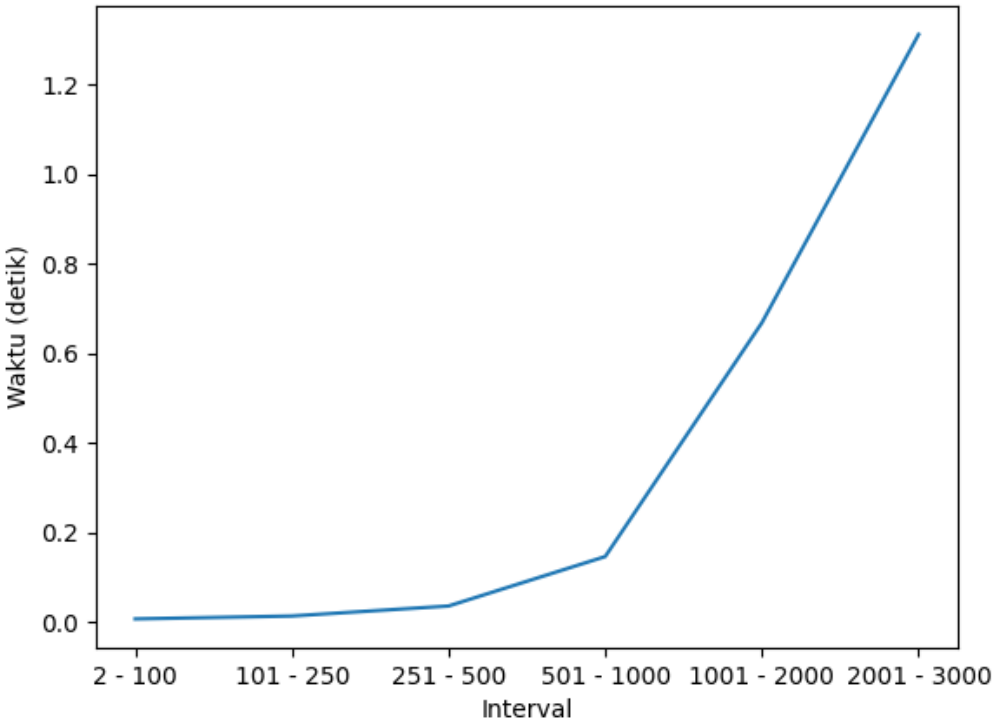

**Gambar 4.6 Pembangkitan Kunci (Penerima)**

Pada Gambar 4.5 dan Gambar 4.6 menjelaskan bahwa keluaran kunci rahasianya sama. Hal ini membuktikan bahwa sistem berhasil melakukan proses pembakitan kunci dengan baik.

Pada Gambar 4.5 menunjukkan bagaimana pembangkitan kunci pada sisi pengirim dilakukan. Pembangkitan bilangan prima ($q$) pada sistem dilakukan dengan cara memilih angka secara acak dalam interval tertentu, lalu menguji keprimaan angka tersebut dengan menggunakan algoritma AKS (Agrawal-Kayal-Saxena). Waktu eksekusi untuk menguji keprimaan sebuah angka dengan menggunakan algoritma AKS ditunjukkan pada Tabel 4.1.

**Tabel 4.1 Waktu Eksekusi Algoritma AKS**

| Interval | Jumlah Bilangan Prima | Waktu (detik) |
|---|---|---|
| 2 -100 | 25 | 0.00694584846496582 |
| 101 - 250 | 28 | 0.013449430465698242 |
| 251 - 500 | 42 | 0.035733699798583984 |
| 501 - 1000 | 73 | 0.14615750312805176 |
| 1001 - 2000 | 135 | 0.668210506439209 |
| 2001 - 3000 | 127 | 1.3116991519927979 |

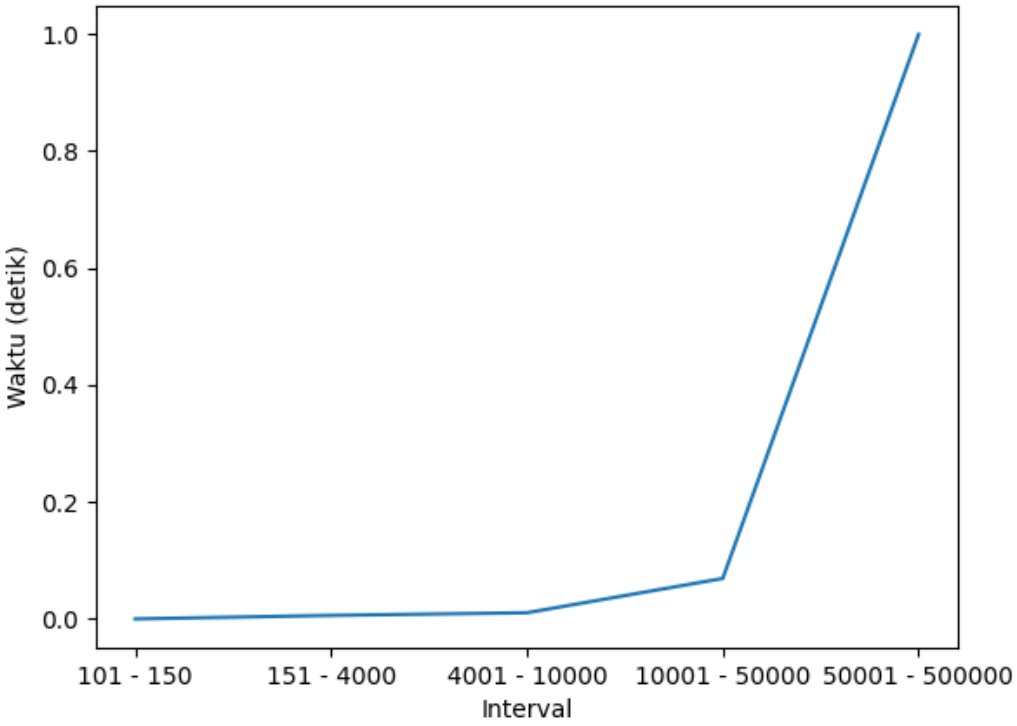



**Gambar 4.7 Grafik Perbandingan Interval dengan Waktu Eksekusi (AKS)**

Pada Tabel 4.2 menunjukkan waktu eksekusi algoritma Fermat's ketika menguji kerprimaan setiap angka yang ada dalam interval tertentu. Jika dibandigkan dengan algoritma AKS, algoritma Fermat's memiliki waktu eksekusi yang lebih cepat. William Stallings (2022) pada bukunya yang berjudul *"Cryptography and Network Security: Principle and Practice"* juga mengatakan bahwa algoritma AKS tampaknya tidak se-efisien algoritma Miller-Rabin.

**Tabel 4.2 Waktu Eksekusi Algoritma Fermat**

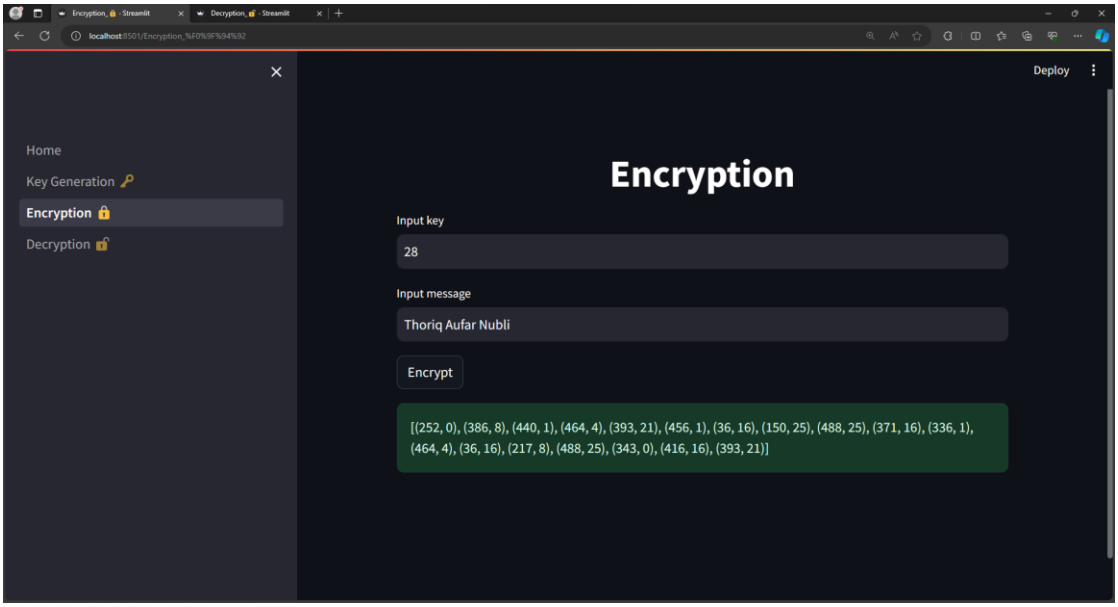
| Interval | Kemungkinan Prima | Waktu (detik) |
|---|---|---|
| 101 - 150 | 10 | 0.0 |
| 151 - 4000 | 517 | 0.006013154983520508 |
| 4001 - 10000 | 680 | 0.010533571243286133 |
| 10001 - 50000 | 3907 | 0.0691530704498291 |
| 50001 - 500000 | 36416 | 0.9989256858825684 |




**Gambar 4.8 Grafik Perbandingan Interval dengan Waktu Eksekusi (Fermat)**

### 4.2.2 Pengujian Enkripsi

Dalam fase ini, pengujian dimulai dengan memasukkan kunci rahasia yang telah dibangkitkan sebelumnya pada kolom *"Input key"*. Lalu masukkan pesan yang ingin dienkripsi pada kolom *"Input message"*. Setelah kedua komponen tersebut dimasukkan, tekan tombol *"Encrypt"* untuk mengenkripsi pesan yang dimasukkan. Dapat dilihat pada Gambar 4.9 keluaran yang dihasilkan adalah pasangan *ciphertext* $(Q, R)$. Kirimkan pasangan *ciphertext* ini ke pihak lain yang terlibat kriptografi.

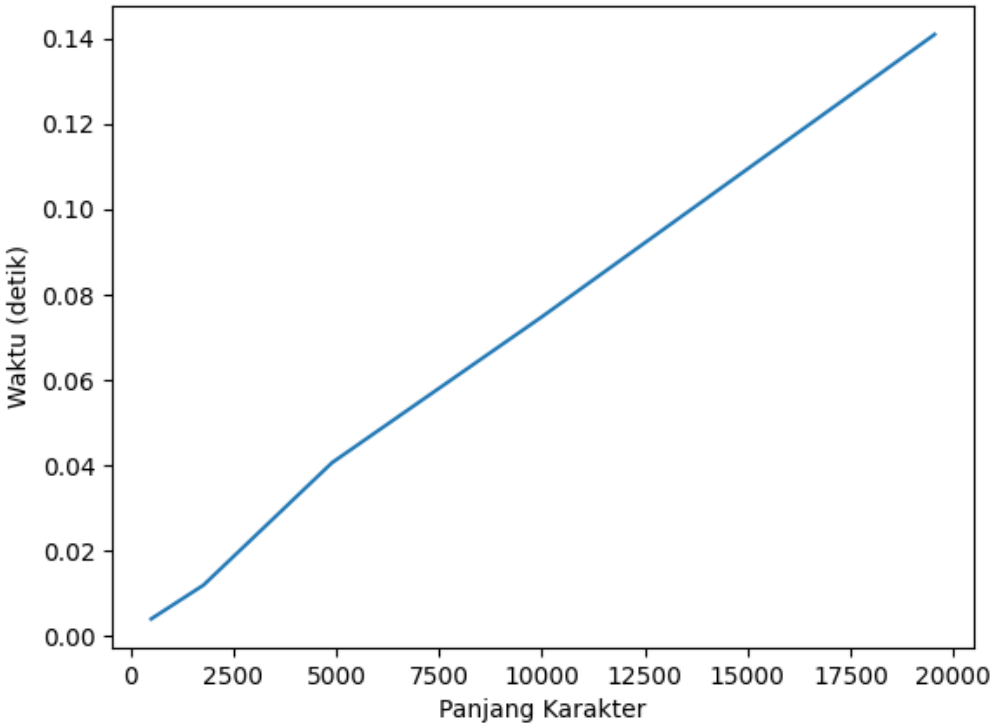



**Gambar 4.9 Enkripsi Pesan**

Pada Tabel 4.3 menunjukkan waktu eksekusi untuk proses enkripsi menggunakan algoritma Rabin-Biswas. Pengujian dilakukan dengan panjang karakter yang berbeda.

**Tabel 4.3 Waktu Ekesekusi Enkripsi Algoritma Rabin-Biswas**

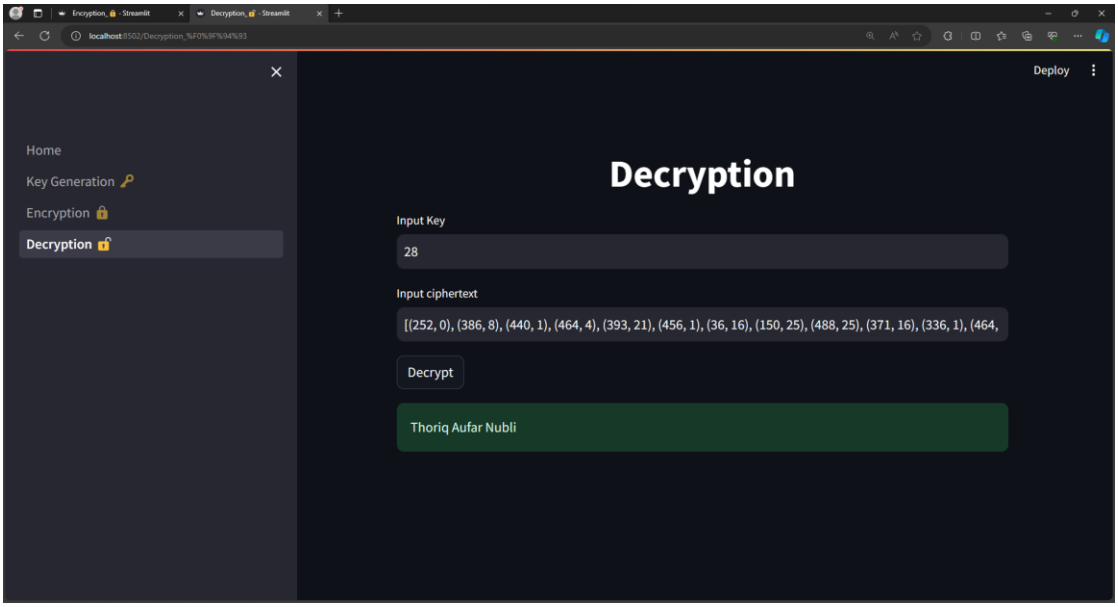
| Panjang Karakter | Waktu (detik) |
|---|---|
| 500 | 0.0040781497955322266 |
| 1770 | 0.011978626251220703 |
| 4900 | 0.04068326950073242 |
| 10000 | 0.07483291625976562 |
| 19552 | 0.14098715782165527 |




**Gambar 4.10 Grafik Perbandingan Panjang Karakter dengan Waktu Eksekusi (Enkripsi)**

### 4.2.3 Pengujian Dekripsi

Dalam tahap ini, pengujian dimulai dengan memasukkan kunci rahasia pada kolom *"Input Key"*. Lalu masukkan pasangan *ciphertext* $(Q, R)$ yang didapat dari pihak lain yang terlibat kriptografi. Tekan tombol *"Decrypt"* untuk melakukan dekripsi *ciphertext*. Dapat dilihat keluaran pada Gambar 4.11 tidak berbeda dengan pesan yang dimasukkan pada Gambar 4.9.

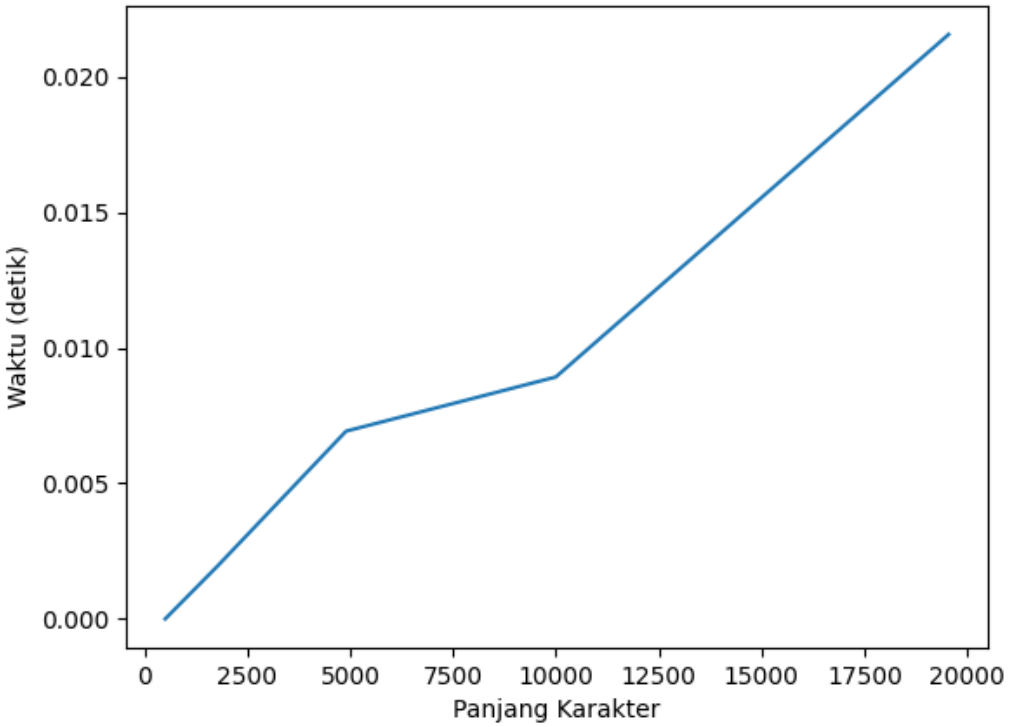



**Gambar 4.11 Dekripsi Pesan**

Waktu eksekusi untuk proses dekripsi menggunakan algoritma Rabin-Biswas ditunjukkan pada Tabel 4.4. Panjang karakter yang digunakan pada setiap pengujian adalah berbeda.

**Tabel 4.4 Waktu Ekseksi Dekripsi Algoritma Rabin-Biswas**

| Panjang Karakter | Waktu (detik) |
|---|---|
| 500 | 0.0 |
| 1770 | 0.0019435882568359375 |
| 4900 | 0.0069239139556884766 |
| 10000 | 0.008920907974243164 |
| 19552 | 0.02155327796936035 |




**Gambar 4.12 Grafik Perbandingan Panjang Karakter dengan Waktu Eksekusi (Dekripsi)**

### 4.2.4 Perhitungan Manual

Dalam fase ini dilakukan perhitungan secara manual dengan pesan yang digunakan pada pengujian ini adalah “Kriptografi 2023”.

#### 4.2.4.1 Pembangkitan Kunci

Pada Tabel 4.5 menunjukkan perhitungan manual pada tahap pembangkitan kunci algoritma Rabin-Biswas.

**Tabel 4.5 Perhitungan pada Pembangkitan Kunci**

| Alice (Sender) | | Eve (Eavesdropper) | | Bob (Receiver) | |
|---|---|---|---|---|---|
| Known | Unknown | Known | Unknown | Known | Unknown |
| $q = 29$ | | ✓ | | ✓ | |
| $\alpha = 2$ | | ✓ | | ✓ | |
| $Xa = 10$ | $Xb = 11$ | | $Xa$ & $Xb$ | $Xb = 11$ | $Xa = 10$ |
| $Ya = 2^{10} \bmod 29$ | | | | $Yb = 2^{11} \bmod 29$ | |
| $K = 18^{10} \bmod 29$<br>$K = 22$ | | 18 | 9 | $K = 9^{11} \bmod 29$<br>$K = 22$ | |

#### 4.2.4.2 Enkripsi Pesan

Pada tabel 4.6 menunjukkan perhitungan manual pada tahap enkripsi pesan dengan menggunakan rumus enkripsi algoritma Rabin-Biswas.

**Tabel 4.6 Perhitungan pada Enkripsi Pesan**

| *Plaintext* (m) | Kode ASCII | $Q = \lfloor m^2/K \rfloor$ | $R = m^2 \bmod K$ | C = (Q, R) |
|---|---|---|---|---|
| K | 75 | 255 | 15 | (255, 15) |
| r | 114 | 590 | 16 | (590, 16) |
| i | 105 | 501 | 3 | (501, 3) |
| p | 112 | 570 | 4 | (570, 4) |
| t | 116 | 611 | 14 | (611, 14) |
| o | 111 | 560 | 1 | (560, 1) |
| g | 103 | 482 | 5 | (482, 5) |
| r | 114 | 590 | 16 | (590, 16) |
| a | 97 | 427 | 15 | (427, 15) |
| f | 102 | 472 | 20 | (472, 20) |
| i | 105 | 501 | 3 | (501, 3) |
| (spasi) | 32 | 46 | 12 | (46, 12) |
| 2 | 50 | 113 | 14 | (113, 14) |
| 0 | 48 | 104 | 16 | (104, 16) |
| 2 | 50 | 113 | 14 | (113, 14) |
| 3 | 51 | 118 | 5 | (118, 5) |

#### 4.2.4.3 Dekripsi Pesan

Pada tabel 4.7 menunjukkan perhitungan manual pada tahap dekripsi pesan dengan menggunakan rumus dekripsi algoritma Rabin-Biswas.

**Tabel 4.7 Perhitungan pada Dekripsi Pesan**

| *Ciphertext* (Q, R) | $D = \sqrt{Q * K + R}$ | *Plaintext* (m) |
|---|---|---|
| (255, 15) | 75 | K |
| (590, 16) | 114 | r |
| (501, 3) | 105 | i |
| (570, 4) | 112 | p |
| (611, 14) | 116 | t |
| (560, 1) | 111 | o |
| (482, 5) | 103 | g |
| (590, 16) | 114 | r |
| (427, 15) | 97 | a |
| (472, 20) | 102 | f |
| (501, 3) | 105 | i |
| (46, 12) | 32 | (spasi) |
| (113, 14) | 50 | 2 |
| (104, 16) | 48 | 0 |
| (113, 14) | 50 | 2 |
| (118, 5) | 51 | 3 |

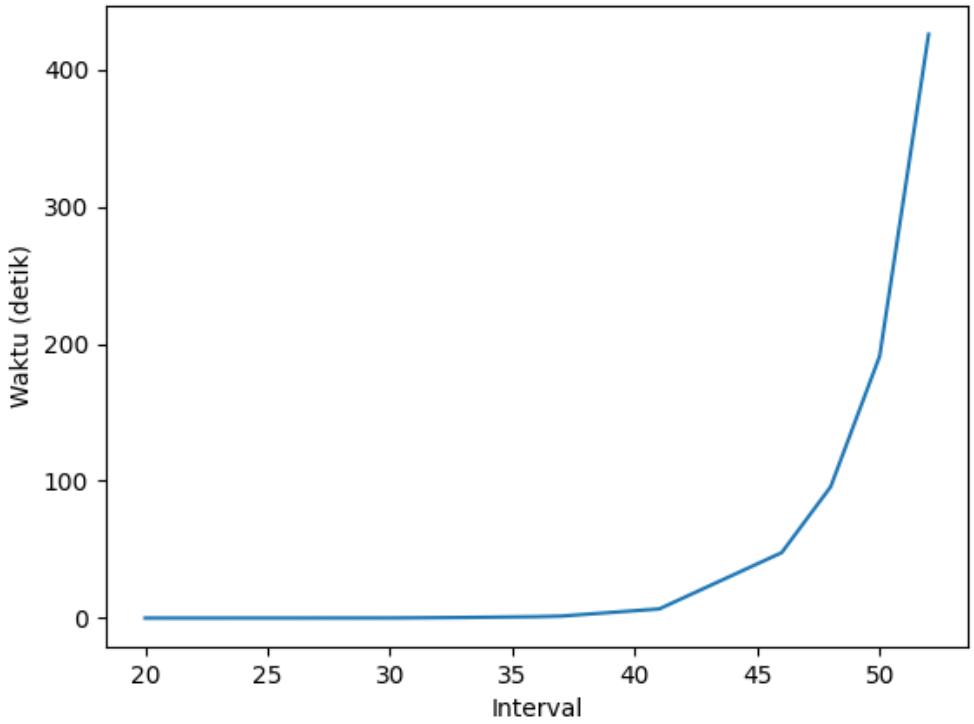
## 4.3 Kriptanalisis

### 4.3.1 *Baby Step Giant Step*

Pada tahap ini, peneliti mencoba untuk memecahkan masalah logaritma diskrit yang terdapat pada tahap pembangkitan kunci. Pada Tabel 4.8 menunjukkan waktu yang dibutuhkan untuk memecahkan masalah logaritma diskrit pada tahap pembangkitan kunci menggunakan algoritma *baby step giant step*.

**Tabel 4.8 Waktu Eksekusi *Baby Step Giant Step***

| Bit | Diketahui | | | Tidak Diketahui | Waktu (detik) |
|---|---|---|---|---|---|
| | q | α | Kunci Publik | Kunci Privat | |
| 20 | 574631 | 543362 | 239368 | 543578 | 0.0 |
| 21 | 1754173 | 1432080 | 1605011 | 1663493 | 0.0025773048400878906 |
| 26 | 62475089 | 59939851 | 7201321 | 56531706 | 0.02295684814453125 |
| 30 | 672580393 | 548122732 | 456672350 | 549681178 | 0.06055045127868652 |
| 31 | 1453779911 | 1401430654 | 891220113 | 1406480332 | 0.15481328964233398 |
| 33 | 7197337909 | 5570386732 | 5877397155 | 5926098205 | 0.4225118160247803 |
| 36 | 48718275887 | 35313753426 | 44443802782 | 43859085605 | 0.9994480609893799 |
| 37 | 101522316823 | 100216483024 | 44406454569 | 100585441126 | 1.4496934413909912 |
| 41 | 1735122003403 | 1484604903731 | 1050015045273 | 1343740119508 | 6.673135757446289 |
| 46 | 66693904715447 | 55747916407814 | 54805890751550 | 39293092325451 | 47.74265265464783 |
| 48 | 270712274071639 | 259793047894982 | 238655079879997 | 39688883891426 | 95.77358770370483 |
| 50 | 922615262936663 | 849545799838169 | 827474071641716 | 548786789299096 | 191.04351663589478 |
| 52 | 4130859632819951 | 3790094517968443 | 1319003345788892 | 905335938493318 | 425.82457160949707 |




**Gambar 4.13 Grafik Perbandingan Panjang Bit dengan Waktu Eksekusi (*Baby-Step Giant-Step*)**

Dengan menggunakan algoritma *baby-step giant-step*, masalah logaritma diskrit yang terdapat pada tahap pembangkitan kunci algoritma Rabin-Biswas dapat dipecahkan. Jika $q$, $\alpha$, dan kunci publik yang digunakan lebih kecil dari 20 bit, masalah logaritma diskrit pada algoritma Rabin-Biswas dapat dipecahkan hanya dalam waktu 0 detik.

### 4.3.2 *Ciphertext Only Attack*

Algoritma ini bisa dibobol hanya dengan mendapatkan *ciphertext*-nya saja. Asumsikan karakter yang ingin dienkripsi adalah "z". Dengan kunci yang digunakan $K = 15000$ dan $m = z = 122$ akan dilakukan proses penyandian menggunakan algoritma Rabin-Biswas. Maka hasil enkripsinya adalah sebagai berikut.

$$Q = \left\lfloor \frac{m^2}{K} \right\rfloor = \left\lfloor \frac{122^2}{15000} \right\rfloor = 0$$

$$R = m^2 \ mod \ K = 122^2 \ mod \ 15000 = 14884 \ mod \ 15000 = 14884$$

$$C = (Q, R) = (0,14884)$$

Lalu kita lakukan dekripsi *ciphertext* tersebut dengan menggunakan rumus dekripsi algoritma Rabin-Biswas.

$$D = \sqrt{Q * K + R} = \sqrt{0 * K + 14884} = \sqrt{14884} = 122 = z$$

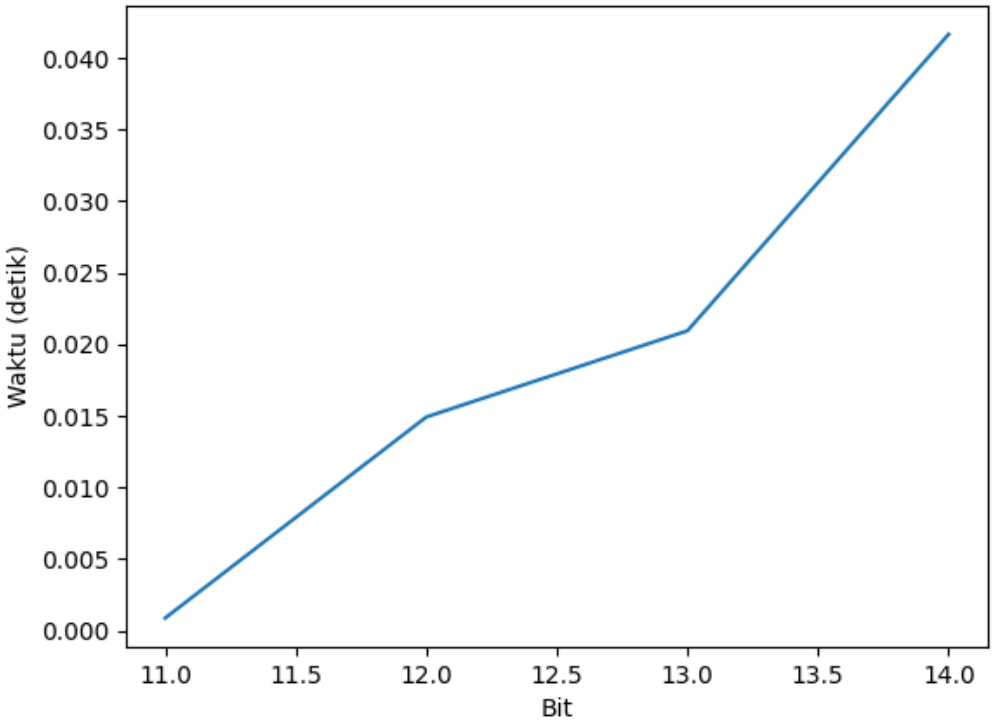
Dapat dilihat dari proses dekripsi di atas, tanpa memasukkan kunci pada proses perhitungan, tetap dapat melakukan dekripsi *ciphertext* dan hasil dekripsi sama dengan karakter awal, yakni "z". Jika dilihat pada tahap enkripsi, dapat disimpulkan jika $K > m^2$, maka dapat dipastikan pasangan *ciphertext* yang dihasilkan adalah $(0, m^2)$. Dengan itu, dapat dilakukan pemecahan *ciphertext* tanpa menggunakan kunci yang digunakan pada fase penyandian yakni dengan mengakarkuadratkan $m$.

### 4.3.3 *Brute Force Attack*

Setelah diketahui jika $K > m^2$, proses enkripsi pada algoritma ini adalah kelemahan utama algoritma ini. Untuk itu, $K$ yang digunakan pada proses enkripsi harus lebih kecil dari $m^2$. Asumsikan yang ingin dienkripsi adalah "z". Digunakannya karakter "z" sebagai contoh dikarenakan karakter "z" adalah alphabet dengan kode ASCII paling besar, yakni 122. Jadi, $K$ yang digunakan harus $K < 122^2 = K < 14884$. Pada tahap ini, peneliti akan membangkitkan beberapa dari $q$, $\alpha$, $Xa$, $Xb$, $Yb$, $Ya$ yang akan menghasilkan $K < 14884$ dan mencoba untuk melakukan *brute force attack* untuk mengetahui waktu yang dibutuhkan untuk mendapatkan kunci privat $(Xa, Xb)$ dan kunci rahasia $(K)$. Dapat dilihat pada Tabel 4.9 adalah hasil *brute force attack* pada setiap $q$, $\alpha$, $Ya$, $Yb$ yang dibangkitkan untuk $K < 14884$.

**Tabel 4.9 Waktu Eksekusi *Brute Force Attack***

| Bit | Diketahui | | | | Tidak Diketahui | | | Waktu (Detik) |
|---|---|---|---|---|---|---|---|---|
| | q | α | Ya | Yb | Xa | Xb | K | |
| 11 | 1549 | 1543 | 1286 | 276 | 1012 | 1290 | 1273 | 0.0008707046508789062 |
| 12 | 3907 | 3853 | 1422 | 3631 | 3504 | 3334 | 3419 | 0.014920473098754883 |
| 13 | 6971 | 5852 | 4474 | 843 | 6504 | 6579 | 6486 | 0.02094292640686035 |
| 14 | 14419 | 14121 | 6781 | 4863 | 14109 | 14033 | 13888 | 0.041660308837890625 |




**Gambar 4.14 Grafik Perbandingan Panjang Bit dengan Waktu Eksekusi (*Brute Force*)**

# BAB 5

# PENUTUP

## 5.1 Kesimpulan

1. Algoritma M.S.H. Biswas (Rabin-Biswas) yang diimplementasikan pada sistem dapat mengenkripsi pesan serta mendekripsi dengan sangat baik dan waktu eksekusi yang efisien. *Ciphertext* yang dihasilkan pada proses enkripsi dapat dikembalikan ke bentuk semula tanpa ada satu karakter pun yang hilang.
2. Waktu eksekusi algoritma Agrawal-Kayal-Saxena (AKS) untuk menguji keprimaan suatu angka yang diperoleh pada penilitian tidak cukup baik. Algoritma AKS mendapatkan waktu eksekusi kurang lebih 1 detik untuk menguji setiap angka dalam interval 2001-3000. Sedangankan algoritma Fermat dapat menguji keprimaan setiap angka pada interval 50001-500000 kurang dari 1 detik.
3. Algoritma *Baby-Step Giant-Step* dapat digunakan untuk memecahkan permasalahan logaritma diskrit yang terdapat pada tahap pembangkitan kunci. Dengan besar digit yang sama untuk setiap $q$, $\alpha$, dan kunci publik, algoritma ini dapat menemukan kunci privat yang memenuhi $\alpha^{kunci\ privat} \equiv kunci\ publik\ (mod\ q)$ kurang dari 1 detik untuk setiap $q$, $a$, dan kunci publik yang digunakan kurang dari 12 digit.
4. Kelemahan utama dari algoritma ini adalah pada proses enkripsinya. Ketika kunci rahasia ($K$) yang digunakan pada proses enkripsi lebih besar dari $m^2$. Maka, pasangan *ciphertext* yang dihasilkan adalah $(0, m^2)$. Untuk memecahkan *ciphertext* tersebut, dapat dilakukan dengan mengakarkuadratkan $m^2$.
5. Jika kunci rahasia ($K$) yang digunakan lebih kecil dari $m^2$, juga menjadi masalah lain. Karena jika $K$ yang digunakan kecil, komponen yang digunakan untuk menghasilkan $K$ juga kecil. Bahkan, jika dilakukan brute force attack untuk $K$ yang lebih kecil dari alfabet dengan kode ASCII paling besar, yakni “z” dengan kode

ASCII “122”. Hanya membutuhkan waktu dibawah 1 detik untuk menemukan komponen yang tidak diketahui pada saat menghasilkan $K < 122^2$ termasuk $K$ itu sendiri.

## 5.2 Saran

1. Untuk menutupi kelemahan algoritma ini pada proses enkripsi, lakukan pengelempokkan beberapa karakter menjadi satu buah blok. Gabungkan setiap kode ASCII setiap karakter yang ada dalam blok. Dengan metode ini, kunci rahasia $(K)$ yang digunakan pasti sangat besar tetapi tetap harus lebih kecil dari $m^2$. Makin banyak karakter yang dikelompokkan dalam satu blok, makin besar pula $K$ yang digunakan.
2. Disarankan untuk melakukan studi lebih lanjut guna memperoleh pemahaman lebih mendalam menyebabkan lambatnya waktu eksekusi untuk uji keprimaan suatu angka dengan menggunakan algoritma AKS.

# DAFTAR PUSTAKA


Abullah, M. A., & Ariffin, M. R. (2014). Rabin-*p* Cryptosystem: Practical and Efficient Method for Rabin based Encryption Scheme. 1-13.

Agrawal, M., Kayal, N., & Saxena, N. (2004). PRIMES is in P. *Annals of Mathematics*, 781-793.

Ariffin, M. R., Asbullah, M. A., Abu, N. A., & Mahad, Z. (2013). A New Efficient Asymmetric Cryptosystem Based on the Integer Factorization Problem of $N = p^2q$. *Malaysian Journal of Mathematical Sciences*, 19-37.

Biswas, M. S. (2019). A mathematical model for ascertaining same ciphertext generated from distinct plaintext in Michael O. Rabin Cryptosystem. *International Journal of Scientific & Engineering Research*, 596-601.

Biswas, M. S. (2020). DESIGN A NEW CRYPTOSYSTEM. *International Journal of Scientific and Research Publications (IJSRP)*, 1-85.

Budiman, M. A., Saputra, M. Y., & Handrizal. (2020). A Hybrid Cryptosystem Using Vigenère Cipher and Rabin-*p* Algorithm in Securing BMP Files. *Journal of Computing and Applied Informatics (JoCAI)*, 89-99.

Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 644-654.

Grari, H., Lamzabi, S., Azouaoui, A., & Zine-Dine, K. (2021). Cryptanalysis of Merkle-Hellman cipher using ant colony optimzation. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 490-500.

Jafri, M. D., Sarmin, N. H., Ghafar, A. H., & Asbullah, M. A. (2021). Simple Cryptanalysis on the Phony Rivest-Shamir-Adleman cryptosystem. *International Journal of Cryptology Research*, 1-12.

Kota, C. M., & Aissi, C. (2022). Implementation of the RSA algorithm and its cryptanalysis. *2002 GSW*.

Mahad, Z., Ariffin, M. R., Ghafar, A. H., & Salim, N. R. (2022). Cryptanalysis of RSA-Variant Cryptosystem Generated by Potential Rogue CA Methodology. *Symmetry 2022*, 1-14.

Munir, R. (2019). *Kriptografi*. Bandung: ITB.

Munir, R. (2019). *Serangan Terhadap Kriptografi*. Bandung: ITB.

Pramitasari, R. (2022). ALGORITMA OPTIMASI CHAOS PADA RIDGE POLYNOMIAL NEURAL NETWORK UNTUK KRIPTANALISIS KUNCI PUBLIK ELGAMAL. *Information System Journal (INFOS)*, 40-45.

Rabin, M. O. (1979). Digitalized Signatures and Public Key Functions as Intractable as Factorization. *MIT Laboratory for Computer Science*, 1-16.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 120-126.

Sarbini, I. N., Jin, W. T., Feng, K. L., Othman, M., Said, M. R., & Hung, Y. P. (2018). Garbage-man-in-the-middle (type 2) Attack on the Lucas Based El-Gamal Cryptosystem in the Elliptic Curve Group Over Finite Field. *Cryptology and Information Security Conference 2018* (pp. 35-41). Port Dickson: Institute for Mathematical Research.

Shanks, D. (1971). Class number, a theory of factorization and genera. *Proc. Symp. Pure Math.*, 415-440.

Stallings, W. (2022). *Cryptography and Network Security: Principles and Practice, Global Edition.* Pearson.

Susilo, W., Tonien, J., & Yang, G. (2021). Divide and capture: An improved cryptanalysis of the encryption standard algorithm RSA. *Computer Standards and Interfaces 74*, 1-6.