IMPLEMENTASI PAILLIER CRYPTOSYSTEM DAN MATRIKS PERMUTASI PADA KEAMANAN PESAN TEKS

SKRIPSI

OLEH
WAHYU SETYO NUGROHO
NIM. 18610067

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

IMPLEMENTASI PAILLIER CRYPTOSYSTEM DAN MATRIKS PERMUTASI PADA KEAMANAN PESAN TEKS

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S. Mat)

Oleh:
Wahyu Setyo Nugroho
NIM. 18610067

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

IMPLEMENTASI PAILLIER CRYPTOSYSTEM DAN MATRIKS PERMUTASI PADA KEAMANAN PESAN TEKS

SKRIPSI

Oleh
Wahyu Setyo Nugroho
NIM. 18610067

Telah Disetujui Untuk Diuji

Malang, 13 Juni 2025

Dosen Pembimbing I	Dosen Pembimbing II
Muhammad Khudzaifah, M.Si.	Erna Herawati, M.Pd.
NIPPPK. 19900511 202321 1 029	NIPPPK. 19760723 202321 2 006

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

IMPLEMENTASI PAILLIER CRYPTOSYSTEM DAN MATRIKS PERMUTASI PADA KEAMANAN PESAN TEKS

SKRIPSI

Oleh
Wahyu Setyo Nugroho
NIM. 18610067

Telah Dipertahankan di Depan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Malang, 25 Juni 2025

Ketua Penguji	: Dr. Elly Susanti, M.Sc	.....
Anggota Penguji 1	: Mohammad Nafie Jauhari, M.Si	.....
Anggota Penguji 2	: Muhammad Khudzaifah, M.Si	.....
Anggota Penguji 3	: Erna Herawati, M.Pd.	.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Wahyu Setyo Nugroho

NIM : 18610067

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Paillier Cryptosystem dan Matriks Permutasi pada Kemanan Pesan Teks

Menyatakan dengan sebenar-benarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan dan pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 25 Juni 2025

Yang membuat pernyataan,

Wahyu Setyo Nugroho

NIM. 18610067

HALAMAN MOTTO

“Belajar bukan tentang menjadi yang terbaik, tapi menjadi lebih baik dari sebelumnya”

KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakatuh.

Alhamdulillah, puji syukur peneliti panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat serta hidayah-Nya sehingga peneliti mampu menyelesaikan draf skripsi yang berjudul "Implementasi Paillier Cryptosystem Dan Matriks Permutasi Pada Keamanan Pesan Teks" dengan baik, dalam rangka mendapat gelar sarjana Matematika di Fakultas sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Shalawat serta salam semoga tetap tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah mengarahkan kita dari zaman kegelapan menuju zaman yang terang benerang yakni agama islam seperti yang kita rasakan sekarang ini.

Dalam proses mengerjakan skripsi ini, banyak bimbingan, masukan, serta arahan yang diterima oleh penulis. Oleh karena itu penulis ingin mengucapkan terimakasih melalui halaman ini kepada :

1. Prof. Dr. H. M. Zainuddin, M.A., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Elly Susanti, M.Sc., selaku Ketua Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

4. Muhammad Khuzaifah, M.Si, selaku Dosen Pembimbing I yang telah memberikan banyak ilmu, bimbingan, arahan, nasihat dan motivasi kepada penulis selama perkuliahan sampai penulisan skripsi ini.
5. Erna Herawati, M.Pd, selaku Dosen pembimbing II yang telah memberi bimbingan, arahan serta nasihat kepada penulis.
6. Seluruh Dosen Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
7. Orang tua dan seluruh keluarga yang selalu memberi dukungan baik dalam bentuk moral maupun material, serta selalu mendoakan untuk kelancaran penulisan proposal ini.
8. Seluruh teman teman yang selalu menemani dan memberi semangat dalam proses pengerjaan proposal ini.

Akhir kata, penulis berharap semoga Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu dalam proses penulisan skripsi ini dan semoga skripsi ini membawa manfaat baik bagi penulis maupun pembaca.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Malang, 25 Juni 2025

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAK

Nugroho, Wahyu S. 2025**. Implementasi Paillier Cryptosystem dan Matriks Permutasi pada Keamanan Pesan Teks.** Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Muhammad Khudzaifah, M.Si. (II) Erna Herawati, M.Pd.

Kata Kunci: Kriptografi, Paillier Cryptosystem, Matriks Permutasi, Enkripsi, Homomorfik, Python

Keamanan pesan merupakan aspek fundamental dalam sistem komunikasi digital modern. Penelitian ini mengkaji dan mengimplementasikan kombinasi algoritma Paillier Cryptosystem yang bersifat homomorfik aditif dengan metode Matriks Permutasi guna meningkatkan kerahasiaan dan resistensi terhadap analisis kriptografi. Penelitian dilakukan secara kualitatif deskriptif melalui simulasi dalam bahasa pemrograman Python. Pesan teks dikonversi ke kode ASCII, kemudian dienkripsi menggunakan algoritma Paillier, dan hasil *ciphertext* dipermutasi menggunakan matriks acak untuk meningkatkan kebingungan (confusion). Proses dekripsi dilakukan dengan invers permutasi dan penguraian *ciphertext* menjadi *plaintext* semula. Hasil menunjukkan bahwa kombinasi dua metode ini mampu meningkatkan tingkat keamanan pesan teks secara signifikan. Penelitian ini tidak hanya relevan secara teknis, tetapi juga mendukung nilai-nilai amanah dan integritas data sesuai prinsip syariah.

ABSTRACT

Nugroho, Wahyu S. 2025. **Implementation of Paillier Cryptosystem and Permutation Matrix in Securing Text Messages**. Thesis. Department of Mathematics, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisors: (I) Muhammad Khudzaifah, M.Si., (II) Erna Herawati, M.Pd.

Keywords: Cryptography, Paillier Cryptosystem, Permutation Matrix, Homomorphic, Encryption, Python

Message security is a critical component of modern digital communication systems. This study implements a hybrid approach combining the Paillier Cryptosystem, an additive homomorphic encryption method, with a Permutation Matrix technique to enhance the confidentiality and resistance of encrypted messages against cryptanalysis. A descriptive qualitative methodology is employed, utilizing Python for simulation and implementation. Plaintext messages are converted into ASCII codes, encrypted via the Paillier algorithm, and then rearranged using randomized permutation matrices to increase the complexity and confusion of the ciphertext. Decryption is carried out by applying the inverse of the permutation matrix followed by Paillier decryption to restore the original plaintext. The results demonstrate that the integration of these two cryptographic techniques significantly improves message security. Furthermore, this study not only provides technical contributions to the field of cryptography but also reinforces ethical values of trust and data integrity consistent with Islamic principles.

مستخلص البحث

نوكروهو، وحي ستييا. ٢٠٢٥. **تنفيذ نظام تشفير *Paillier* ومصفوفة التقليب على أمان الرسائل النصية.** اليحث العلمي. برقسم الرياضيات ، كلية العلوم والتكنولوجيا ، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (١) محمد حذيفة، الماجستيرفي العلوم (٢) إيرناهراواتي، الماجستيرة في تعليم اللغة العربية

الكلمات الأساسية: التشفير ، نظام التشفير *Paillier* ، مصفوفة التقليب ، التشفير متماثل ، بايثون

يعد أمان الرسائل جانبا أساسيا من أنظمة الاتصالات الرقمية الحديثة. بحثت هذه الدراسة وتنفذ الجمع بين خوارزمية نظام تشفير *Paillier* المتجانسة المضافة مع طريقة مصفوفة التقليب لزيادة السرية ومقاومة تحليل التشفير. تم إجراء البحث بطريقة وصفية نوعية من خلال المحاكاة بلغة برمجة بايثون. يتم تحويل الرسائل النصية إلى كود *ASCII* ، ثم تم تشفيرها باستخدام خوارزمية *Paillier* ، وتم تغيير نتائج النص المشفر باستخدام مصفوفة عشوائية لزيادة الالتباس. تم عملية فك التشفير عن طريق التقليب العكسي وتحلل النص المشفر إلى النص العادي الأصلي. تظهر النتائج أن الجمع بين هاتين الطريقتين قادر على زيادة مستوى أمان الرسائل النصية بشكل كبير. هذا البحث ليس وثيق الصلة من الناحية الفنية فحسب ، بل يدعم أيضا قيم الثقة وسلامة البيانات وفقا لمبادئ الشريعة الإسلامية.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Manusia terlahir sebagai makhluk sosial. Maka dari itu manusia membutuhkan komunikasi untuk saling berinteraksi dan memahami satu sama lain. Cara manusia berkomunikasi terus mengalami perkembangan seiring kemajuan teknologi dan ilmu pengetahuan. Salah satu cara manusia berkomunikasi yaitu melalui tulisan yang berfungsi untuk menyampaikan pesan dari penulis kepada pembacanya.

Menurut (Ariyus, 2008), pesan terbagi menjadi dua jenis, yaitu pesan yang bersifat rahasia dan tidak rahasia. Pesan rahasia adalah pesan yang dikirim secara khusus agar tidak diketahui orang lain. Cara mengirim pesan rahasia melalui enkripsi, sandi sunstitusi, atau aplikasi dengan kunci rahasia. Pesan tidak rahasia adalah pesan yang tidak penting atau tidak terlalu diperhatikan sehingga orang lain dapat menggandakannya. Cara mengirimnya pun secara biasa, yaitu melalui obrolan. Dari uraian tersebut maka pesan yang bersifat rahasia perlu dijaga keamanannya agar terhindar dari kejahatan-kejahatan internet (Stallings, 2017). Maka diperlukan solusi agar pesan atau informasi sampai kepada penerima dengan aman.

Dalam surat An-Nisa' ayat 58 yang artinya: "Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil" (QS. An-Nisa': 58). Berdasarkan arti surat di atas dapat diambil kesimpulan bahwa Allah SWT memerintahkan umatnya untuk "amanat"

kepada yang berhak.

Amanat adalah sesuatu yang Allah SWT. percayakan kepada hamba-Nya untuk dilaksanakan dengan sebaik-baiknya, yang meliputi amanat Allah kepada hamba-Nya, amanat seseorang kepada sesama, dan terhadap dirinya sendiri. Sejalan dengan nilai-nilai amanah tersebut, negara juga memiliki tanggung jawab untuk melindungi informasi yang bersifat pribadi dan penting bagi setiap warganya. Dalam konteks kenegaraan, perlindungan terhadap data pribadi penduduk memiliki landasan konstitusional dalam Undang-Undang Dasar Negara Republik Indonesia Tahun 1945, khususnya Pasal 28G ayat (1) yang menyatakan bahwa: "Setiap orang berhak atas perlindungan diri pribadi, keluarga, kehormatan, martabat, dan harta benda yang di bawah kekuasaannya, serta berhak atas rasa aman dan perlindungan dari ancaman ketakutan…" Dalam era digital saat ini, data pribadi seperti Nomor Induk Kependudukan (NIK), alamat, dan informasi biometrik termasuk dalam kategori hak atas privasi yang wajib dilindungi.

Sebagai penguatan terhadap amanat konstitusi tersebut, pemerintah telah menerbitkan Undang-Undang Nomor 27 Tahun 2022 tentang Perlindungan Data Pribadi (UU PDP) yang mengatur secara komprehensif hak subjek data, kewajiban pengendali data, serta mekanisme keamanan informasi. Oleh karena itu, dalam dunia teknologi informasi, penerapan metode kriptografi seperti enkripsi merupakan bagian penting dalam menjaga kerahasiaan dan integritas data pribadi agar tidak disalahgunakan oleh pihak yang tidak berwenang. Teknologi enkripsi tidak hanya memenuhi aspek teknis keamanan, tetapi juga mencerminkan komitmen terhadap nilai etis dan hukum yang berlaku dalam masyarakat.

Salah satu ilmu matematika yang dapat menjaga keamanan pesan adalah kriptografi. Kriptografi merupakan salah satu cabang ilmu yang mempelajari penulisan secara rahasia untuk melindungi kerahasiaan pesan (Stallings, 2017). Kriptografi menggunakan algoritma kriptografi untuk mencapai tujuan keamanan informasi. Cara kerja algoritma ini yaitu mengubah teks asli (plaintext) menjadi teks kode (ciphertext) yang disebut dengan enkripsi. Sedangkan pengertian dekripsi sendiri yaitu proses pengubahan teks yang telah dienkripsi ke bentuk asalnya (plaintext). Algoritma yang digunakan untuk proses dekripsi dan enkripsi tentu berbeda. Pada algoritma kriptografi juga dibutuhkan kunci untuk melakukan proses enkripsi dan dekripsi (Ariyus, 2008).

Algoritma kriptografi dibagi menjadi dua yaitu algoritma simetris dan algoritma asimetris. Pembagian algoritma ini berdasarkan kunci yang dibutuhkan. Setiap algoritma memiliki keunggulan masing-masing. Salah satu metode enkripsi kriptografi yaitu enkripsi homomorfik. Enkripsi homomorfik adalah salah satu metode enkripsi yang memungkinkan operasi pada data terenkripsi tanpa harus melakukan dekripsi terlebih dahulu (Gentry, 2009). Jenis enkripsi homomorfik sendiri dibagi menjadi dua, yaitu enkripsi homomorfik sebagian dan enkripsi homomorfik penuh. Enkripsi homomorfik dapat melindungi informasi sensitif sehingga dapat diaplikasikan untuk mengamankan pesan. Pada penelitian ini akan dipilih metode enkripsi sebagian menggunakan algoritma Paillier cryptosystem.

Algoritma Paillier cryptosystem merupakan salah satu kriptografi asimetris yang bersifat probabilistik dan memiliki sifat homomorfisme aditif (Paillier, 1999). Algoritma ini menggunakan dua kunci yang berbeda yaitu kunci publik untuk mengenkripsi data dan juga kunci privat untuk mendekripsikan data.

Kelebihan Paillier cryptosystem yaitu memiliki sifat homomorfisme aditif, yang artinya pesan yang akan dikirim dapat ditambahkan ketika dienkripsi dan pihak lain tidak dapat mendekripsikan dengan tepat. Sehingga algoritma ini cocok diaplikasikan untuk dapat menjaga keamanan pesan sehingga pesan yang bersifat rahasia tetap terjaga kerahasiaannya dan aman dari pencurian data.

Untuk lebih meningkatkan keamanan pesan, maka digunakan metode tambahan berupa Matriks Permutasi. Matriks permutasi berfungsi untuk mengacak susunan elemen *ciphertext* sehingga walaupun berhasil disadap, pesan tidak dapat langsung dikenali tanpa informasi matriks invers (Dewi Suryani, 2019). Kombinasi antara algoritma Paillier dengan teknik permutasi ini diharapkan mampu meningkatkan tingkat kerahasiaan dan kebingungan (confusion) dalam proses kriptografi modern (Ayu Wahyuni, 2020).

Teknologi terus dikembangkan untuk mendapatkan hasil yang optimal dan cara yang efektif. Maka dari itu penulis ingin menghubungkan Paillier cryptosystem dengan Matriks Permutasi untuk meningkatkan keamanan pesan. Dimana Matriks Permutasi dalam Paillier cryptosystem diharapkan dapat membuat proses enkripsi dan dekripsi menjadi aman. Berdasarkan uraian yang telah dikemukakan, maka peneliti tertarik untuk mengkaji lebih dalam penelitian yang berjudul "Implementasi Paillier Cryptosystem dan Matriks Permutasi pada keamanan pesan Teks".

1.2 Rumusan Masalah

Adapun rumusan masalah penelitian berdasarkan latar belakang sebagai berikut:

1. Bagaimana implementasi algoritma *Paillier cryptosystem* dan Matriks permutasi pada keamanan pesan teks?
2. Bagaimana implementasi algoritma Paillier cryptosystem dan Matriks permutasi pada keamanan pesan teks dalam program python?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, maka tujuan dari penelitian ini sebagai berikut:

1. Untuk mengetahui proses implementasi algoritma Paillier cryptosystem dan Matriks permutasi untuk mengamankan pesan teks.
2. Untuk mengetahui implementasi algoritma Paillier cryptosystem dan Matriks permutasi untuk mengamankan pesan teks dalam program python.

1.4 Manfaat Penelitian

Adapun manfaat penelitian berdasarkan tujuan penelitian sebagai berikut:

1. Bagi Penulis

   Mengetahui cara mengamankan pesan menggunakan metode enkripsi homomorfik sebagian dengan pengaplikasian algoritma *Paillier cryptosystem* dan Matriks permutasi.

2. Bagi Pembaca dan Peneliti Selanjutnya
   a. Dapat menambah wawasan tentang ilmu kriptografi khususnya enkripsi homomorfik menggunakan algoritma *Paillier cryptosystem* dan Matriks permutasi.
   b. Mengetahui keamanan pesan teks menggunakan enkripsi homomorfik sebagian dengan pengaplikasian algoritma Paillier cryptosystem dan Matriks permutasi.

c. Sebagai referensi bagi peneliti selanjutnya dalam memodifikasi enkripsi homomorfik sebagian menggunakan algoritma Paillier cryptosystem dan Matriks permutasi.

3. Bagi Institusi
   a. Sebagai media pembelajaran bagi para mahasiswa khususnya mata kuliah kriptografi.
   b. Mengimplementasikan materi khususnya mata kuliah kriptografi dalam dunia teknologi.

1.5 Batasan Masalah

Berikut batasan masalah penelitian sebagai berikut:

1. Data yang dienkripsi adalah data pesan teks.
2. Algoritma yang akan digunakan adalah Paillier cryptosystem dan Matriks permutasi.
3. Implementasi pengamanan pesan ini dilakukan menggunakan software python.

1.6 Definisi Istilah

Berikut istilah yang digunakan dalam penelitian sebagai berikut:

1. Pesan adalah informasi tertulis berupa teks.
2. Plaintext adalah pesan asli yang mudah dipahami.
3. Ciphertext adalah sebuah pesan yang tidak memiliki arti atau makna dan telah melalui proses enkripsi.
4. Enkripsi adalah suatu proses pada pesan asli (plaintext) yang diubah menggunakan algoritma tertentu menjadi pesan rahasia yang tidak dimengerti (ciphertext).

5. Ciphertext adalah fungsi matematika yang digunakan untuk proses enkripsi dan dekripsi.
6. Dekripsi adalah proses pengubahan pesan rahasia yang tidak dapat terbaca (ciphertext) menggunakan algoritma tertentu menjadi pesan yang dapat dibaca (plaintext).
7. Algoritma adalah proses yang harus dilakukan dalam perhitungan atau operasi pemecahan masalah
8. Kunci adalah parameter yang digunakan untuk proses enkripsi dan dekripsi.

BAB II
KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 Keterbagian

keterbagian atau divisibility adalah konsep dasar yang menyatakan hubungan pembagian antara dua bilangan bulat. Untuk $a, b \in \mathbb{Z}$, dengan $a \neq 0$, dikatakan bahwa "a membagi b" atau $a|$b jika terdapat bilangan bulat $k \in \mathbb{Z}$ sehingga:

$$b = a \cdot k$$

Dalam hal ini:

a disebut sebagai pembagi dari b

b disebut sebagai kelipatan dari a

Jika tidak ada bilangan bulat k yang memenuhi persamaan tersebut, maka a tidak membagi b, ditulis $a \nmid$ b. (Burton, 2011)

Contoh: $4 \mid 20$, Karena $20 = 4 \cdot 5$

Teorema 1

Jika $a \mid b$ maka $a \mid bc$ untuk setiap $c \in \mathbb{Z}$

Bukti:

$a|b$, maka $b = a \cdot k$ untuk suatu $k \in \mathbb{Z}$ (definisi)

bc $= (a \cdot$ k$) \cdot$ c untuk setiap $c \in \mathbb{Z}$ (kalikan kedua ruas dengan c)

bc $= a \cdot$ (k $\cdot$ c) untuk setiap $kc \in \mathbb{Z}$ (asosiatif perkalian)

Terbukti bahwa $a \mid bc$

Teorema 2

Jika $a \mid b$ dan $b \mid c$, maka $a \mid c$

Bukti:

$a \mid b$ maka $b = a \cdot k$ untuk suatu $k \in \mathbb{Z}$	(definisi)
$b \mid c$ maka $c =$ b $\cdot l$ untuk suatu $l \in \mathbb{Z}$	(definisi)
$c = (a \cdot k) \cdot l$	(subtitusi b pada $b \mid c$)
$c = a \cdot (k \cdot l)$ untuk $kl \in \mathbb{Z}$	(asosiatif perkalian)
Terbukti $a \mid c$	

Teorema 3

Jika $a \mid b$ *dan* $a \mid c$, maka $a \mid$ (m$\cdot b + n \cdot c$) untuk setiap $m, n \in \mathbb{Z}$

$a \mid b$ maka $b = a \cdot k$ untuk $k \in \mathbb{Z}$	(definisi)
$a \mid c$ maka $c = a \cdot l$ untuk $l \in \mathbb{Z}$	(definisi)
Misal $m \cdot b + n \cdot c$ dengan $m, n \in \mathbb{Z}$	(teorema 3)
$m \cdot b + n \cdot c = m \cdot (a \cdot k) + n \cdot (a \cdot l)$	(subtitusikan nilai b dan c)
$m \cdot b + n \cdot c = a \cdot (m \cdot k) + a \cdot (n \cdot l)$	(asosiatif perkalian)
$m \cdot b + n \cdot c = a(m \cdot k + n \cdot l)$ dengan $m, k, n, l \in \mathbb{Z}$	(faktorkan a dari kedua suku)
Terbukti $a \mid$ (m$\cdot b + n \cdot c$)	

Teorema 4

Jika $a \mid b$ dan $a \mid c$, maka $\gcd(a, b) = |a|$

$a \mid b$ maka $b = ak$ untuk $k \in \mathbb{Z}$	(definisi)
Misal $d \mid a$ dan $d \mid b$	(definisi)
Karena $d \mid ak$ maka $d \mid b$	(asosiatif perkalian)
Semua pembagi a membagi b	(implikasi dari definisi pembagi)

Himpunan pembagi bersama a dan b = himpunan pembagi a	(definisi)
Pembagi terbesar dari a adalah $\|a\|$	(definisi)
Terbukti Jika $a \mid b$ dan $a \mid c$, maka $\gcd(a,b) = \|a\|$	

Teorema 5

Jika gcd(*a*,*n*) =1, maka *a* memiliki invers modulo n

gcd(*a*,*n*) =1	(definisi)
a dan n saling prima	(definisi)
$\exists x, y \in \mathbb{Z}, a \cdot x + n \cdot y = 1$	(identitas bezout)
$a \cdot x + n \cdot y \equiv 1 \; mod \; n$	(modulo kedua ruas)
$n \cdot y \equiv 0 \; mod \; n$	(karena $n \mid n \cdot y$)
$a \cdot x \equiv 1 \; mod \; n$	(pengurangan)
x adalah invers dari *a* mod *n*	(definisi invers modulo)
$\exists a^{-1} \equiv x \bmod n$	(terbukti)

Teorema 6

Jika $a \equiv b \; (mod \; n)$ maka:

1. $a + c \equiv b + c \; (mod \; n)$
2. $a \cdot c \equiv b \cdot c \; (mod \; n)$

Bukti: $a + c \equiv b + c \; (mod \; n)$

$a \equiv b \; (mod \; n)$	(definisi)
$n \mid (a - b)$	(definisi kongruensi)
$(a + c) - (b + c) = a - b$	(Penjumlahan)

$n|(a+c-b-c) = a-b$ (selisih tetap sama)

$a+c \equiv b+c \ (mod\ n)$ (terbukti)

Bukti: $a \cdot c \equiv b \cdot c \ (mod\ n)$

$a \equiv b \ (mod\ n)$ (definisi)

$n|(a-b)$ (definisi kongruensi)

$(a-c)-(b-c) = (a-b) \cdot c$ (distributif)

$n|(a-b)c$ (karena $n|(a-b)$)

$a \cdot c \equiv b \cdot c \ (mod\ n)$ (terbukti)

2.1.2 Aritmetika Modular

Aritmetika modular merupakan cabang dari teori bilangan yang membahas operasi bilangan bulat dalam sistem bilangan terbatas, yang disebut sistem modulo. Dalam sistem ini, operasi seperti penjumlahan, pengurangan, dan perkalian dilakukan dengan mengambil sisa pembagian terhadap suatu bilangan positif *m*, yang disebut sebagai modulus.

Definisi Aritmetika Modular

Jika m adalah bilangan bulat positif, maka himpunan bilangan bulat modulo m dinotasikan dengan $\mathbb{Z}_m$. Dua bilangan bulat a dan b dikatakan kongruen modulo m jika m membagi selisih $a - b$, ditulis sebagai $a \equiv b \ (mod\ m)$. Artinya, a dan b berada dalam kelas sisa yang sama. Sebagaimana dijelaskan oleh (Kurnia D. A., 2015),"dalam sistem bilangan modulo, kita mengelompokkan bilangan ke dalam kelas sisa berdasarkan hasil bagi pembagi tertentu."

Sistem Bilangan Modulo

Dalam aritmetika modular, himpunan bilangan yang relevan dinyatakan sebagai Z_m, yaitu himpunan bilangan bulat dari 0 hingga $m - 1$, di mana m adalah

modulus. Setiap bilangan bulat akan direpresentasikan oleh sisa pembagiannya terhadap m. Misalnya, dalam sistem modulo 6, kita memiliki:

$$\mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}$$

Setiap bilangan bulat akan memiliki pasangan kongruen dalam sistem ini.

Sebagai contoh: $8 \equiv 2\ (mod\ 6)$ karena 8 dan 2 memiliki sisa pembagian yang sama saat dibagi 6.

Operasi dalam Aritmetika Modular

1. Operasi Dasar Modular

Tiga operasi utama dalam aritmetika modular adalah penjumlahan, pengurangan, dan perkalian. Operasi-operasi ini dilakukan seperti biasa, namun hasil akhirnya diambil sebagai sisa modulo m. Secara umum berlaku sifat:

$$(a + b) mod\ m = [(a\ mod\ m) + (b\ mod\ m)] mod\ m$$

$$(a - b) mod\ m = [(a\ mod\ m) - (b\ mod\ m)] mod\ m$$

$$(a \cdot b)\ mod\ m = [(a\ mod\ m) \cdot (b\ mod\ m)]\ mod\ m$$

Contoh:

Misalkan diberikan a = 11 dan b = 7 dengan modulus m = 6:

$$(a + b)\ mod\ 6 = (11 + 7)\ mod\ 6 = 18\ mod\ 6 = 0$$

$$a - b\ mod\ 6 = (11 - 7)\ mod 6 = 4\ mod\ 6 = 4$$

$$a \cdot b\ mod\ 6 = 11 \cdot 7\ mod\ 6 = 77\ mod\ 6 = 5$$

Sehingga:

$$11 + 7 \equiv 0\ (mod\ 6)$$

$$11 - 7 \equiv 4\ (mod\ 6)$$

$$11 \cdot 7 \equiv 5\ (mod\ 6)$$

2. Eksponensial Modular

Eksponensial modular adalah operasi menghitung pangkat bilangan bulat kemudian mengambil hasilnya dalam modulo tertentu, ditulis sebagai $a^b \ mod \ m$. Operasi ini menjadi sangat penting dalam perhitungan dengan bilangan besar, terutama dalam bidang komputasi dan teori kriptografi (Trappe, 2006).

Untuk efisiensi perhitungan, digunakan metode eksponensiasi kuadrat berulang *(exponentiation by squaring)*. Metode ini memecah pangkat menjadi bentuk biner dan melakukan kuadrat bertahap untuk mengurangi jumlah operasi.

Contoh: Hitung $2^{10} \ mod \ 17$ dengan eksponensiasi kuadrat:

$2^2 = 4$

$2^4 = (2^2)^2 = 16$

$2^8 = (2^4)^2 = 256 \rightarrow 256 \, mod \, 17 = 1$

Gabungkan: $2^{10} = 2^8 \cdot 2^2 = 1 \cdot 4 = 4 \, mod \, 17$

Maka, $2^{10} \equiv 4 \ (mod \ 17)$

Aritmetika modular menyediakan kerangka penting dalam perhitungan bilangan bulat di ruang terbatas. Operasi-operasi ini menjadi dasar dalam pengembangan sistem keamanan digital dan kriptografi modern. Pemahaman terhadap operasi dasar dan eksponensial modular sangat penting sebagai fondasi untuk memahami relasi kongruensi dan sistem bilangan kriptografi.

2.1.3 Kongruensi

Kongruensi adalah suatu relasi yang menunjukkan bahwa dua bilangan memiliki sisa pembagian yang sama jika dibagi oleh bilangan bulat positif tertentu (modulus). Notasi yang digunakan adalah:

$$a \equiv b \ (mod \ m)$$

yang artinya bilangan bulat a dan b memiliki selisih yang habis dibagi oleh m, atau secara formal:

$$m|(a - b)$$

Relasi kongruensi memiliki sifat-sifat matematis yang menunjukkan bahwa relasi ini termasuk relasi ekuivalensi. Relasi ekuivalensi berarti relasi tersebut bersifat refleksif, simetris, dan transitif. Ketiga sifat ini penting untuk memahami struktur kelas-kelas kongruensi yang terbentuk dari sistem bilangan modulo. Pembahasan berikut akan menjelaskan masing-masing sifat tersebut beserta pembuktiannya secara formal.

1. Refleksif

Setiap bilangan bulat *x* memenuhi: $x \equiv x \ (mod\ m)$

Bukti:

Berdasarkan definisi kongruensi, $x \equiv y \ (mod\ m)$ berlaku jika *m* membagi selisih $x - y$. Untuk kasus refleksif:

$$x - x = 0$$

Karena 0 selalu habis dibagi m (untuk $m \neq 0$), maka $m \mid 0$ dengan demikian:

$$x \equiv x \ (mod\ m)$$

2. Simetris

Jika $x \equiv y \ (mod\ m)$, maka $y \equiv x \ (mod\ m)$

Bukti:

Diketahui bahwa $x \equiv y \ (mod\ m)$ artinya $m \mid (x - y)$, maka terdapat $z \in \mathbb{Z}$ sehingga:

$x - y = m \cdot z$

Dengan mengalikan kedua ruas dengan -1:

$-(x - y) = -m \cdot z$

$y - x = (-z) \cdot m$

Karena $(-z) \in \mathbb{Z}$, maka $y \equiv x \ (mod\ m)$

3. Transitif

Jika $x \equiv y \ (mod\ m)$ dan $y \equiv z \ (mod\ m)$, maka $x \equiv z \ (mod\ m)$

Bukti:

Misalkan:

$x \equiv y \ (mod\ m)$, maka $x - y = m \cdot a$ untuk $a \in \mathbb{Z}$

$y \equiv z \ (mod\ m)$, maka $y - z = m \cdot b$ untuk $b \in \mathbb{Z}$

Jumlahkan kedua persamaan:

$(x - y) + (y - z) = m \cdot a + m \cdot b$

$x - z = m \cdot (a + b)$

Karena $(a + b) \in \mathbb{Z}$, maka $x \equiv z \ (mod\ m)$

Dengan demikian, relasi kongruensi memenuhi ketiga sifat dasar relasi ekuivalensi, yaitu refleksif, simetris, dan transitif. Sifat-sifat ini menjadikan relasi kongruensi penting dalam pembentukan himpunan bilangan modulo yang membagi bilangan bulat menjadi kelas-kelas ekivalen. Hal ini menjadi dasar dalam banyak aplikasi matematika, termasuk sistem kriptografi modern yang menggunakan operasi-operasi modular untuk menjamin keamanan pesan. (Burton, 2011)

2.1.4 Kongruensi Kuadrat

Kongruensi kuadrat merupakan salah satu bentuk khusus dari kongruensi dalam teori bilangan. Konsep ini penting dalam kajian matematika dan kriptografi karena berkaitan dengan apakah suatu bilangan merupakan hasil kuadrat dari bilangan lain dalam sistem bilangan modulo tertentu. Pemahaman tentang

kongruensi kuadrat sangat berguna dalam pembuktian bilangan prima, kriptografi berbasis residu kuadrat, dan sistem enkripsi dalam beberapa algoritma (Trappe, 2006).

Menurut Kurnia (Kurnia D. A., 2015), kongruensi kuadrat dapat dipahami sebagai bentuk kesetaraan hasil kuadrat dalam sistem bilangan modulo, yang sering digunakan dalam pengujian bilangan prima. Secara formal, diberikan bilangan bulat a dan modulus n, Dimana n adalah anggota bilangan asli positif, dikatakan bahwa a merupakan residu kuadrat modulo n jika terdapat bilangan bulat x sedemikian sehingga:

$$x^2 \equiv a \ (mod\ n)$$

Dalam hal ini, bilangan a disebut sebagai residu kuadrat modulo n. Sebaliknya, jika tidak terdapat bilangan bulat x yang memenuhi kongruensi tersebut, maka a disebut sebagai non-residu kuadrat modulo n.

Contoh:

Pertimbangkan sistem modulo 7. Perhitungan kuadrat dari bilangan 1 hingga 6 adalah sebagai berikut:

$$1^2 \equiv 1 \ (mod\ 7)$$

$$2^2 \equiv 4 \ (mod\ 7)$$

$$3^2 \equiv 2 \ (mod\ 7)$$

$$4^2 \equiv 2 \ (mod\ 7)$$

$$5^2 \equiv 4 \ (mod\ 7)$$

$$6^2 \equiv 1 \ (mod\ 7)$$

Dari hasil tersebut, bilangan 1, 2, dan 4 adalah residu kuadrat modulo 7 karena ada bilangan bulat yang kuadratnya kongruen dengan nilai-nilai tersebut modulo 7.

Bilangan 3, 5, dan 6 tidak termasuk residu kuadrat dalam modulo 7 karena tidak ada bilangan *x* yang memenuhi $x^2 \equiv 3, 5, atau\ 6\ (mod\ 7)$.

Jumlah Residu Kuadrat Modulo Prima

Jika p adalah bilangan prima, maka terdapat tepat $\frac{(p-1)}{2}$ residu kuadrat berbeda dalam sistem modulo *p*. Hal ini disebabkan oleh fakta bahwa kuadrat dari bilangan *x* dan $-x$ akan menghasilkan nilai yang sama dalam modulo *p*. Karena hanya diperlukan satu perwakilan dari setiap pasangan *(x, −x),* maka jumlah residu kuadrat menjadi separuh dari total bilangan bulat tak nol dalam sistem tersebut (Burton, 2011).

Teorema

Jika *p* adalah bilangan prima, maka terdapat tepat $\frac{(p-1)}{2}$ residu kuadrat yang berbeda modulo *p*.

Bukti:

Misalkan $f(x) = x^2\ mod\ p$ untuk $x = 1, 2, \ldots, p-1$. Karena p adalah prima, setiap nilai *x* dan $-x \in \mathbb{Z}$ menghasilkan kuadrat yang sama:

$$(-x)^2 = x^2$$

Sehingga nilai kuadrat hanya unik untuk setiap pasangan $(x, -x)$. Karena terdapat $(p-1)$ bilangan bulat dari 1 hingga $p-1$, maka jumlah nilai kuadrat unik adalah $\frac{(p-1)}{2}$ (Rosen, 2012).

Kongruensi kuadrat memberikan dasar penting dalam teori bilangan dan kriptografi. Masalah menentukan apakah suatu bilangan merupakan residu kuadrat memiliki keterkaitan langsung dengan algoritma keamanan dan perhitungan

modular. Oleh karena itu, konsep ini menjadi bagian integral dalam pembentukan sistem enkripsi berbasis bilangan bulat (Kurnia D. A., 2015).

2.1.5 Sejarah Kriptografi

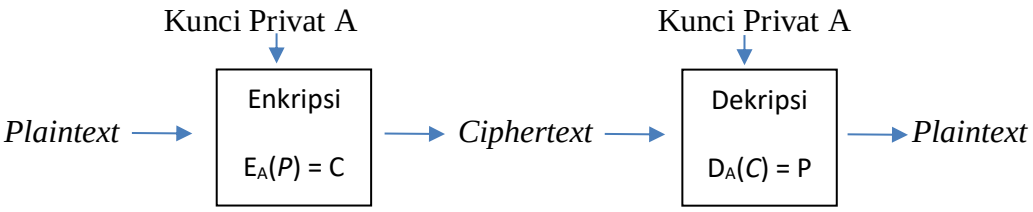
Kriptografi berasal dari bahasa Yunani yaitu “kryptos” artinya rahasia dan “graphein” artinya tulisan. Kriptografi adalah ilmu matematika yang mempelajari teknik mengamankan informasi seperti menjaga kerahasiaan data, integritas data, dan keabsahan data. Kriptografi sudah digunakan sejak awal tahun 400 sebelum masehi oleh tantara sparta di Yunani. Mereka menggunakan alat yang dinamakan *scytale*. *Scytale* terbuat dari kertas panjang dari daun papyrus yang dililitkan pada sebuah silinder dari diameter tertentu. Biasanya pesan ditulis secara horizontal atau dalam bentuk baris per baris dan jika pita dilepaskan maka huruf-huruf di dalamnya telah tersusun secara acak membentuk pesan rahasia.

Kriptografi mulai berkembang sejak adanya perkembangan peralatan komputer digital. Kriptografi modern beroperasi pada string biner. Selain memberikan aspek keamanan confidentiality, kriptografi modern juga memberikan aspek keamanan lain seperti otentikasi dan integritas data. Pada sistem kriptografi modern, kekuatan kriptografinya terletak pada kunci. Parameter yang digunakan untuk transformasi enkripsi dan dekripsi disebut kunci. Kunci biasanya berupa string atau deretan bilangan dan deretan karakter. Aturan untuk menjalan proses enkripsi dan dekripsi pada kriptogarfi yang berupa fungsi matematika disebut algoritma kriptografi.

2.1.6 Klasifikasi Kriptografi

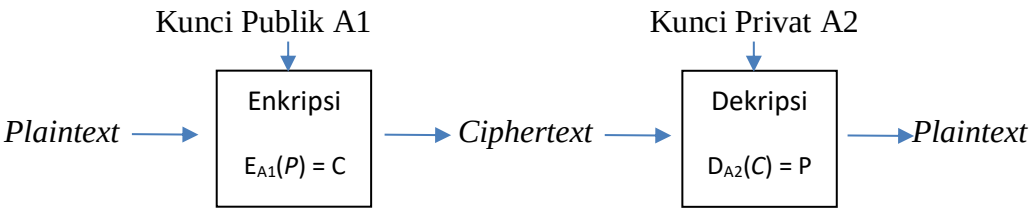
Menurut (Schneier, 1996), algoritma kriptografi berdasarkan kuncinya dibedakan menjadi dua, yaitu algoritma simetris dan algoritma asimetris. Algoritma

simetris sering disebut algoritma klasik karena pada proses enkripsi dan dekripsinya menggunakan kunci yang sama. Pada proses pengiriman pesan menggunakan algoritma simetri, pengirim pesan harus memberitahukan kunci kepada penerima pesan agar dapat mendeskrisikan pesan yang diterimanya. Adapun kelebihan dari algoritma simetris yaitu proses yang sederhana, membutuhkan data komputasi yang lebih sedikit, dan lebih cepat dibandingkan dengan enkripsi asimetris. Namun algorima simetris juga memiliki kelemahan, yaitu diperlukan kunci yang berbeda apabila si penerima pesan adalah orang yang berbeda. Adapun contoh algoritma simetris antara lain DES (Data Encryption Standard), IDEA, GOST, Blowfish, Serpent, RC2, RC4, RC5, Rijndael, dan lain-lain. Adapun skema kriptografi algoritma simetris dapat dilihat pada Gambar 2.1.



Gambar 1 Skema Algoritma Simetris

Algoritma asimetris merupakan algoritma yang menggunakan kunci berbeda untuk mendekripsikan proses enkripsi dan dekripsinya, yaitu kunci privat untuk proses dekripsi dan kunci publik untuk proses enkripsi. Kunci publik dapat disebarluaskan atau bersifat umum, sedangkan kunci privat perlu disimpan dan hanya boleh diketahui oleh penerima dan pengirim pesan. Penerapan algoritma ini sering digunakan pada pembuatan tanda tangan digital serta pembuatan dan distribusi kunci sesi. Adapun contoh algoritma asimetris yaitu RSA, ECC, Diffie-Hellman, ElGamal, DSA, dan XTR. Adapun skema kriptografi algoritma asimetris dapat dilihat pada Gambar 2.2 berikut.



Gambar 2 Skema Algoritma Asimetris

2.1.7 Metode Substitusi

Metode substitusi merupakan salah satu konsep dasar dalam kriptografi yang merujuk pada proses penggantian suatu elemen dengan elemen lain berdasarkan aturan tertentu. Pada sistem kriptografi modern, substitusi tidak hanya dilakukan antar karakter dalam bentuk huruf, tetapi juga dalam bentuk representasi numerik, yang mendukung pengolahan data secara matematis (Paar & Pelzl, 2021)

Salah satu bentuk substitusi numerik yang umum adalah melalui representasi ASCII (American Standard Code for Information Interchange). Dalam sistem ini, setiap karakter dalam teks diubah menjadi bilangan bulat yang unik dan terstandarisasi.

Transformasi ini bersifat deterministik dan satu-ke-satu, memungkinkan pesan teks diubah menjadi urutan bilangan yang dapat diproses lebih lanjut oleh algoritma kriptografi. Dalam literatur modern, proses ini dikenal sebagai substitusi numerik, karena mengubah simbol menjadi bilangan tanpa kehilangan makna posisi atau struktur (Wang & Li, 2020)

Dalam penelitian ini, metode substitusi tidak digunakan sebagai teknik kriptografi mandiri seperti pada sandi klasik, melainkan sebagai tahap praproses yang sangat penting. Setiap karakter dari pesan teks dikonversi terlebih dahulu ke nilai ASCII, sebelum diproses oleh algoritma Paillier Cryptosystem. Dengan demikian, substitusi dalam konteks ini bersifat implisit dan fungsional, yaitu

mengubah domain teks ke bentuk bilangan bulat yang kompatibel dengan sistem kriptografi berbasis matematika.

2.1.8 Metode Transposisi

Metode transposisi merupakan salah satu teknik dalam kriptografi yang bertujuan untuk menyamarkan pesan dengan cara mengubah urutan elemen-elemen penyusunnya, tanpa mengubah isi atau nilai dari elemen tersebut. Berbeda dengan metode substitusi yang mengganti elemen dengan simbol lain, transposisi mempertahankan isi pesan tetapi mengacak struktur penyusunannya agar tidak mudah dikenali (Paar & Pelzl, 2021)

Dalam penerapannya, metode transposisi dapat dilakukan baik pada karakter teks maupun pada representasi numerik dari data. Pada konteks kriptografi modern, transposisi lebih efektif jika dilakukan terhadap bilangan hasil enkripsi, karena memberikan lapisan perlindungan tambahan terhadap pola-pola yang mungkin masih tampak pada *ciphertext*. Salah satu pendekatan yang umum dan efisien untuk melakukan transposisi numerik adalah melalui penggunaan matriks permutasi, yaitu matriks khusus yang menyusun ulang posisi elemen-elemen dalam suatu vektor tanpa mengubah nilainya.

Dalam penelitian ini, metode transposisi digunakan sebagai langkah pasca-enkripsi, setelah proses pengamanan pesan dengan algoritma Paillier Cryptosystem. Hasil enkripsi berupa deretan bilangan kemudian disusun dalam bentuk vektor dan posisinya diacak menggunakan matriks permutasi. Langkah ini menambah keamanan dengan menyulitkan analisis terhadap pola distribusi bilangan dalam *ciphertext*, tanpa mengubah nilai enkripsi yang dihasilkan.

Dengan demikian, metode transposisi memberikan lapisan keamanan tambahan terhadap susunan *ciphertext*, dan berfungsi melengkapi perlindungan isi pesan yang diberikan oleh algoritma enkripsi.

2.1.9 Enkripsi Homomorfik

Enkripsi homomorfik adalah suatu komputasi pada cipherteks tanpa harus mendekripsikan cipherteks itu terlebih dahulu. Hasil cipherteks yang didekripsikan akan memberikan hasil yang sama dengan operasi serupa bila dilakukan pada *plaintext*nya. Secara matematis enkripsi homomorfik adalah sebuah *cryptosystem* untuk melakukan operasi pada cipherteks menggunakan fungsi enkripsi yang bersifat homomorfik. Adapun jenis operasi pada enkripsi homomorfik yaitu penjumlahan dan pengurangan. Suatu homomorfik dikatan aditif jika:

$$E(x + y) = E(x) \otimes E(y)$$

dimana E adalah fungsi enkripsi, x dan y adalah *plaintext*, dan $\otimes$ menyatakan operasi yang bergantung pada chiper yang digunakan (operasi $+$,$\times$, atau operasi lainnya). Sebuah algoritma enkripsi homomorfik dikatakan multiplikatif jika:

$$E(x \cdot y) = E(x) \otimes E(y)$$

dimana E adalah fungsi enkripsi, x dan y adalah *plaintext*, dan $\otimes$ menyatakan operasi yang bergantung pada chiper yang digunakan (operasi $+$,$\times$, atau operasi lainnya).

Enkripsi homomorfik dibagi menjadi dua yaitu enkripsi homomorfik sebagian (*partially homomorphic encryption*) dan enkripsi homomorfik keseluruhan (*fully homomorphic encryption*). Dikatakan enkripsi homomorfik sebagian karena hanya dilakukan satu jenis operasi tertentu pada cipertext, yaitu penjumlahan atau perkalian saja. Sedangkan enkripsi homomorfik keseluruhan

adalah enkripsi homomorfik yang dapat dilakukan dua jenis operasi pada cipertext, yaitu penjumlahan dan perkalian. Sehingga enkripsi homomorfik sebagian hanya memiliki satu sifat saja yaitu sifat aditif atau sifat multiplikatif. Adapun contoh algoritma enkripsi homomorfik sebagian yaitu algoritma RSA, algoritma ElGamal, dan algoritma Paillier (Munir, Enkripsi Homomorfik, 2021).

Enkripsi homomorfik penuh adalah enkripsi homomorfik yang bila didekripsikan dapat dilakukan dua operasi pada cipherteksnya yaitu operasi penjumlahan dan perkalian, dimana hasilnya akan sama dengan penjumlahan atau perkalian *plaintext*nya. Enkripsi homomorfik penuh dapat memiliki sifat aditif dan multiplikatif sekaligus. Operasi penjumlahan dan perkalian dapat dinyatakan sebagai operasi AND dan XOR dalam *string biner*.

2.1.10 Kode ASCII

ASCII (American Standard Code for Information Interchange) yang merupakan salah satu standar yang banyak digunakan pada komputer dan perangkat komunikasi, untuk merepresentasikan sebuah karakter. Kode ASCII memiliki komposisi bilangan biner sebanyak 8 bit. Mulai dari 00000000 hingga 11111111. Total kombinasi yang dihasilkan adalah 256, dimulai dari kode 0 hingga 255, terdiri dari abjad a-z dan A-Z, angka 0-9, beberapa tanda baca yang umum digunakan, dan beberapa karakter kontrol (Kurnia, 2013).

2.1.11 Algoritma Paillier Cryptosystem

Algoritma *Paillier cryptosystem* merupakan salah satu algoritma asimetris probabilistik pada kriptografi kunci publik yang dikembangkan oleh Pasca Paillier pada tahun 1999. Keamanan algortima ini didasarkan pada sulitnya memecahkan persoalan perhitungan *n-residue class*. *Paillier cryptosystem* berdasarkan pada

Composite Residuosity, yaitu jika diberikan $x \in Z_{N^2}^*\{x \in Z | \, 1 \leq x < N^2$ dan $\gcd(x, N) = 1$.

Elemen *x* dikatakan sebuah residu ke-N jika terdapat elemen lain, $y \in Z_{N^2}^*$ yang memenuhi: $x = y^N mod N^2$. Setiap *x* memiliki tepat N buah akar dari $Z_{N^2}^*$atau memiliki *N* buah solusi berbeda. Oleh karena itu, pasti akan sulit menentukan mana solusi yang sebenarnya. Berikut rumus hasil perkalian dari dua buah cipherteks yang akan menghasilkan penjumlahan dari *plaintext*nya:

$$D(E(m_1, r_1) \cdot E(m_2, r_2) mod\ N^2 = m_1 + m_2\ mod\ N$$

Sedangkan jika dekripsi dari cipherteks yang dipangkatkan dengan *plaintext*nya akan menghasilkan perkalian kedua *plaintext* dan dirumuskan sebagai berikut.

$$D(E(m_1, r_1)^{m_2} mod\ N^2) = m_1 \cdot m_2\ mod\ N$$

Paillier cryptosystem merupakan jenis kriptografi berbasis *keypair*, yaitu penerima mendapatkan kunci privat dan publik sehingga pesan yang dienkripsi menggunakan kunci publik serta didekripsi menggunakan kunci privat. Kelebihan algoritma ini sama seperti algoritma homomorfik aditif, yaitu pihak lain tidak dapat mendekripsikan isi pesan dengan benar (Jost, Ha Lam, & Smeets, 2015).

2.1.11.1. Pembangkitan Kunci Algoritma Paillier Cryptosystem

Diperlukan kunci publik dan privat untuk melakukan proses enkripsi dan dekripsi pesan teks pada algoritma *Paillier cryptosystem*. Kunci publik untuk proses enkripsi teks dan kunci privat untuk proses dekripsi pesan. Adapun mekanisme pembangkitan kunci tersebut sebagai berikut.

1. Siapkan teks sebagai plainteks
2. Pilih dua bilangan prima sembarang misal p dan q yang memenuhi syarat (gcd) dengan: $gcd\big(pq, (p-1)(q-1)\big) = 1$

3. Hitung nilai N yang merupakan hasil perkalian nilai p dan q.

   $|Z_N^*| = \varphi(N) = (p-1)(q-1)$

   $|Z_N^*| = \varphi(N^2) = N \cdot \varphi(N)$

4. Hitung nilai $\lambda = lcm(p-1, q-1)$

5. Pilih bilangan bulat acak $g \in Z_{N^2}^*$, dengan syarat $1 \leq g < N^2$, serta memenuhi $gcd(g, N^2) = 1$ dan $gcd(L(g^\lambda \bmod N^2), N) = 1$

   Notasi $Z_{N^2}^*$ menyatakan himpunan bilangan bulat dari 1 hingga $N^2 - 1$ yang relatif prima terhadap N^2, yaitu:$Z_{N^2}^* = \{x \in Z_{N^2}^* | \, gcd(x, N^2) = 1\}$. Dengan kata lain, semua elemen dalam $Z_{N^2}^*$ memiliki invers modulo N^2. Hal ini penting karena pemilihan g dari himpunan ini menjamin bahwa fungsi dekripsi dapat dihitung dengan benar menggunakan properti homomorfik dari Paillier Cryptosystem. Jika g tidak berasal dari $Z_{N^2}^*$, maka fungsi $L(g^\lambda \bmod N^2)$ kemungkinan tidak memiliki invers modulo N, yang akan menyebabkan proses dekripsi gagal.

6. Pilih bilangan acak r, yang memenuhi syarat $1 \leq r < N$ dan $gcd(r, N) = 1$

7. Hitung nilai μ menggunakan basis g yang telah dipilih, dengan rumus $\mu = \left(L(g^\lambda \bmod N^2)\right)^{-1} \bmod N$, dimana fungsi $L(u)$ didefinisikan sebagai $L(u) = (\frac{u-1}{N})$ dengan $u = g^\lambda \bmod N^2$.

Perlu diingat bahwa pemilihan bilangan prima p dan q harus secara acak dan independen satu sama lain dalam proses pembangkitan kunci publik dan privat. Adapun hasil prosedur di atas yaitu kunci publik berupa pasangan (g, N) dan kunci privat pasangan (λ, μ) (Purba, Sinaga, & Purba, 2019)

2.1.11.2. Prosedur Enkripsi dan Dekripsi Paillier Cryptosystem

Adapun mekanisme enkripsi algoritma *Paillier cryptosystem* sebagai berikut.

1. Misalkan *m* pesan yang akan dienkripsi dengan syarat $0 \leq m < N$
2. Pilih dua bilangan prima sembarang misal p dan q yang memenuhi syarat: $gcd\big(p \cdot q, (p-1)(q-1)\big) = 1$
3. Hitung nilai $N = p \cdot q$
4. Hitung nilai $\lambda = lcm(p-1, q-1)$
5. Pilih bilangan bulat acak g dengan syarat $\gcd(L\big(g^{\lambda}\ mod\ N^2\big), N) = 1$
6. Pilih bilangan bulat acak r dengan syarat $0 \leq r < N$ dan $gcd(r, N) = 1$.
7. Hitung cipherteks dari m dengan persamaan $c = g^m \cdot r^N\ mod\ N^2$. Dimana c adalah residu ke-n dalam modulus N^2 dan dilambangkan $[c]_g$.

Adapun mekanisme dekripsi algoritma *Paillier cryptosystem* sebagai berikut.

1. Misalkan c cipherteks yang akan didekripsi.
2. Hitung nilai untuk μ dari basis g yang nilainya memenuhi syarat $\mu = \Big(L\big(g^{\lambda} mod\ N^2\big)\Big)^{-1}\ mod\ N$ Dimana menggunakan fungsi $L(u) = (\frac{u-1}{N})$ dengan $u\ =\ g^{\lambda}\ mod\ N^2$
3. Hitung plainteks dari c dengan persamaan $m = L\big(c^{\lambda}\ mod\ N^2\big) \cdot \mu\ mod\ N$ atau dengan persamaan lain yaitu $m = \frac{L\big(c^{\lambda}\ mod\ N^2\big)}{L\big(g^{\lambda}\ mod\ N^2\big)}\ mod\ N$

(Mulya, Rismawati, & Trisanto, 2020).

Simulasi Enkripsi Paillier Cryptosystem pada pesan teks "BACA"

1. Representasi huruf menggunakan ASCII.

Tabel 1 Simulasi Representasi ASCII

HURUF	REPRESENTASI
B	66
A	65
C	67
A	65

2. Pilih dua bilangan prima: $p = 17,\ q = 19$
3. Hitung nilai N dan N^2:

   $N = p \cdot q = 17 \cdot 19 = 323$

   $N^2 = 104329$
4. Hitung nilai $\lambda = lcm(p-1, q-1) = lcm(16,18) = 144$
5. Pilih $g \in Z^*_{N^2}$, kita ambil: $g = 324$

   Cek apakah g valid: Harus memenuhi: $gcd(L(g^\lambda \ mod\ N^2), N) = 1$

   Ambil $u = g^\lambda \ mod\ N^2$

   $u = g^\lambda \ mod\ N^2 = 324144 \ mod\ 3232 = 46513$

   Maka $L(g^\lambda \ mod\ N^2) = L(46513) = \frac{46513-1}{323} = \frac{46512}{323} = 144$

   Karena $gcd\ (144{,}323) = 1$, maka valid.

   Dari hasil perhitungan diatas ditemukan bahwa:

   Kunci public (N, g) = (323, 324)
6. Pilih $r \in Z^*_{N^2}$, dengan syarat $0 \leq r < N$ dan $gcd(r, N) = 1$.
7. Enkripsi dengan rumus: $c = g^m \cdot r^N \ mod\ N^2$

Tabel 2 Simulasi Enkripsi Algoritma *Paillier Cryptosystem*

Plaintext	**M**	**R**	**Enkripsi**	***Ciphertext***

B	66	2	$c = g^m \cdot r^N \, mod \, N^2$ $= 324^{66} \cdot 2^{323} \, mod \, 323^2$ $= 26154$	$c = 26154$
A	65	3	$c = g^m \cdot r^N \, mod \, N^2$ $= 324^{65} \cdot 3^{323} \, mod \, 323^2$ $= 18557$	$c = 18557$
C	67	4	$c = g^m \cdot r^N \, mod \, N^2$ $= 324^{67} \cdot 4^{323} \, mod \, 323^2$ $= 19461$	$c = 19461$
A	65	5	$c = g^m \cdot r^N \, mod \, N^2$ $= 324^{65} \cdot 5^{323} \, mod \, 323^2$ $= 46535$	$c = 46535$

Simulasi Dekripsi Paillier Cryptosystem pada pesan teks

1. Ciphertext yang dihasilkan dari proses enkripsi di atas adalah: (26154, 18557, 19461, 46535)
2. Hitung nilai untuk μ dari basis g yang nilainya memenuhi syarat $\mu = \left(L\left(g^{\lambda} mod \, N^2\right)\right)^{-1} mod \, N$ dengan fungsi $L(u) = (\frac{u-1}{N})$

   Hitung $\mu = \left(L\left(g^{\lambda} mod \, N^2\right)\right)^{-1} mod \, N$, Dimana dari perhitungan diatas didapatkan nilai nilai $u = g^{\lambda} mod \, N^2$ = 46513 dan N = 323

   Kemudian hitung fungsi $L(\mathrm{u}) = \frac{u-1}{N} = \frac{46513-1}{323} = \frac{46512}{323} = 144$

   Sehingga $\mu = 144^{-1} \, mod \, 323$ = 83 (karena gcd(323, 83) = 1 maka invers modularnya ada, sehingga didapatkan $144 \cdot 83 \equiv 1 \bmod 323$)

   Dari hasil perhitungan diatas ditemukan bahwa:

Kunci privat (λ, μ) = (144, 83)

3. Dekripsi menggunakan rumus: $m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$ dimana $L(u) = (\frac{u-1}{N})$ dengan $u\ =\ c^{\lambda}\ mod\ N^2$

Tabel 3 Simulasi Dekripsi Algoritma *Paillier Cryptosystem*

Cipher text (c)	**Dekripsi**	**Plain text**
26154	$m = L\left(c^{\lambda}\ mod\ N^2\right)\ \cdot\ \mu\ mod\ N$ $= L(26154^{144}\ mod\ 323^2)\ \cdot\ \mu\ mod\ N$ $= L(44252)\ \cdot\ \mu\ mod\ N$ Dimana: L (44252) = $\frac{44252-1}{323} = \frac{44251}{323}$ = 137 Sehingga: $m =\ 137\ \cdot\ \mu\ mod\ N$ $=\ 137\ \cdot\ 83\ mod\ 323$ $=\ 11371\ mod\ 323$ $=\ 66$	B
18557	$m = L\left(c^{\lambda}\ mod\ N^2\right)\ \cdot\ \mu\ mod\ N$ $= L(18557^{144}\ mod\ 323^2)\ \cdot\ \mu\ mod\ N$ $= L(102069)\ \cdot\ \mu\ mod\ N$ Dimana: L (102069) = $\frac{102069-1}{323} = \frac{102068}{323}$ = 316 Sehingga: $m =\ 316\ \cdot\ \mu\ mod\ N$	A

	$= 316 \cdot 83 \, mod \, 323$ $= 26228 \, mod \, 323$ $= 65$	
19461	$m = L\left(c^{\lambda} \, mod \, N^2\right) \cdot \mu \, mod \, N$ $= L(19461^{144} \, mod \, 323^2) \cdot \mu \, mod \, N$ $= L(90764) \cdot \mu \, mod \, N$ Dimana: L (90764) = $\frac{90764-1}{323} = \frac{90763}{323} = 281$ Sehingga: $m = 281 \cdot \mu \, mod \, N$ $= 281 \cdot 83 \, mod \, 323$ $= 23323 \, mod \, 323$ $= 67$	C
46535	$m = L\left(c^{\lambda} \, mod \, N^2\right) \cdot \mu \, mod \, N$ $= L(46535^{144} \, mod \, 323^2) \cdot \mu \, mod \, N$ $= L(102069) \cdot \mu \, mod \, N$ Dimana: L (102069) = $\frac{102069-1}{323} = \frac{102068}{323} = 316$ Sehingga: $m = 316 \cdot \mu \, mod \, N$ $= 316 \cdot 83 \, mod \, 323$ $= 26228 \, mod \, 323$ $= 65$	A

2.1.12 Matriks Permutasi

Matriks permutasi merupakan matriks biner khusus yang digunakan untuk merepresentasikan perubahan posisi atau permutasi elemen dalam suatu vektor atau matriks. Ciri utama dari matriks ini adalah setiap baris dan setiap kolom hanya mengandung satu elemen bernilai 1, sedangkan elemen lainnya bernilai 0. Struktur ini membuat matriks permutasi mampu menyusun ulang elemen-elemen suatu vektor tanpa mengubah nilai elemennya.

Matriks permutasi biasanya dibentuk dengan menyusun ulang baris-baris dari matriks identitas. Dalam bentuk ini, posisi angka 1 dalam tiap baris berpindah ke kolom tertentu, menunjukkan perpindahan posisi suatu elemen. Ketika matriks permutasi dikalikan dengan sebuah vektor, hasilnya adalah vektor baru dengan elemen-elemen yang telah berpindah tempat sesuai dengan pola permutasi yang dibawa oleh matriks tersebut.

Secara matematis, sebuah matriks permutasi $P \in \mathbb{R}^{nxn}$ memiliki properti berikut:

1. Untuk setiap $i \in \{1,2,.,n\}$, terdapat tepat satu $j \in \{1,2,,n\}$ sehingga $P_{ij} = 1$
2. Semua elemen lain dalam baris dan kolom tersebut adalah nol.
3. Matriks permutasi adalah ortogonal, sehingga berlaku $PT = P- 1$
4. Determinan dari matriks permutasi adalah +1 atau –1, tergantung dari jenis permutasinya (genap atau ganjil).

Matriks permutasi digunakan dalam berbagai konteks komputasi dan matematika, termasuk:

1. Pengaturan ulang baris atau kolom dari sebuah matriks.
2. PA : menghasilkan matriks baru dengan baris-baris A disusun ulang.

3. AP : menghasilkan matriks baru dengan kolom-kolom A disusun ulang.
4. Faktorisasi matriks, seperti dalam LU decomposition dengan pivoting.
5. Pemrograman linier dan grafika komputer, di mana posisi data sering diubah dengan efisien menggunakan operasi permutasi.

Contoh:

Jika kita memiliki vektor $C = [1, 2, 3, 4]$

Matriks permutasi: $P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Maka: $C \; x \; P = \begin{bmatrix} 2 \\ 4 \\ 1 \\ 3 \end{bmatrix}$, Sehingga C'= [2, 4, 1, 3]

Matriks ini merepresentasikan bahwa:

1. Baris pertama berpindah ke posisi kedua
2. Baris kedua ke posisi keempat
3. Baris ketiga ke posisi pertama
4. Baris keempat ke posisi ketiga

Ketika matriks ini dikalikan dengan vektor kolom x ∈ $\mathbb{R}4$, maka akan dihasilkan vektor dengan urutan elemen yang telah dipermutasi sesuai C.

2.2 Kajian Integrasi Keagamaan dengan Kriptografi

Amanah berasal dari Bahasa Arab yaitu *amina – ya'manu – amānah* yang bermakna dasar aman atau kepercayaan. Bentuk *amānah* merupakan *masdar* dari *fi'il māḍī amina* dengan pola *fa'ālah*, yang dalam ilmu *ṣarf* menunjukkan sifat atau keadaan yang menetap. Derivasi lain dari akar ini antara lain *āmin* sebagai *ism fā'il* dan *ma'mūn* sebagai *ism maf'ūl* (Muhammad, 2007). Secara bahasa amanah adalah

sesuatu yang dipercayakan atau kepercayaan. Lawan kata dari amanah adalah khianat. Sebagaimana konsep amanah terdapat di dalam QS. An-Nisa' ayat 58 :

إِنَّ ٱللَّهَ يَأۡمُرُكُمۡ أَن تُؤَدُّواْ ٱلۡأَمَٰنَٰتِ إِلَىٰٓ أَهۡلِهَا وَإِذَا حَكَمۡتُم بَيۡنَ ٱلنَّاسِ أَن تَحۡكُمُواْ بِٱلۡعَدۡلِۚ إِنَّ ٱللَّهَ نِعِمَّا يَعِظُكُم بِهِۦٓۗ إِنَّ ٱللَّهَ كَانَ سَمِيعَۢا بَصِيرٗا

Yang artinya "Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil" (QS. An-Nisa': 58).

Dalam tafsir Ibnu Katsir (Katsir, 2017) Ali bin Abi Thalhah berkata dari Ibnu 'Abbas r.a: Amanah adalah kewajiban dari Allah SWT. yang ditawarkan kepada langit, bumi, dan gunung-gunung, dimana jika mereka menunaikannya mereka akan diberikan balasan mulia dan jika diabaikan, mereka akan dibalas hina. Merekapun semua tidak siap menerimanya dan khawatir mengabaikannya sebagai tanda pengagungan kepada agama Robbnya. Amanah inipun kemudian disampaikan kepada Nabi Adam a.s. (dengan semua kandungan dan konsekuensinya), dan Nabi Adam a.s. menerimanya." (Katsir, 2017).

Berdasarkan ayat di atas, konsep amanah telah dijelaskan dengan sebaik mungkin dimana amanah merupakan kepercayaan yang diberikan oleh Allah SWT. kepada makhluk-Nya untuk dijalankan baik berupa perintah maupun larangan. Hal ini sesuai dengan konsep keamanan pesan yang harus dijaga kerahasiaannya agar pesan yang dikirim oleh pengirim dapat diterima dengan baik oleh penerima pesan.

Dalam era informasi dan transformasi digital yang semakin masif, teknologi kriptografi telah menjadi pondasi penting dalam sistem keamanan digital modern. Salah satu aspek utama dari teknologi ini adalah integritas data, yaitu jaminan bahwa data atau informasi yang dikirimkan atau disimpan tidak mengalami perubahan, modifikasi, atau manipulasi oleh pihak yang tidak berwenang. Integritas

menjadi unsur penting dalam berbagai layanan digital, seperti transaksi keuangan elektronik, sistem pemerintahan digital (e-government), tanda tangan digital, serta pengelolaan data pribadi maupun kelembagaan.

Meskipun bersumber dari pengembangan ilmu pengetahuan dan teknologi modern, konsep ini sejatinya selaras dengan nilai-nilai fundamental dalam Islam. Ajaran Islam secara eksplisit menekankan pentingnya kejujuran, amanah, serta larangan terhadap pengubahan atau penyembunyian kebenaran. Hal ini ditegaskan dalam firman Allah SWT, "Wahai orang-orang yang beriman! Janganlah kamu mengkhianati Allah SWT dan Rasul SAW, dan (jangan pula) mengkhianati amanah-amanah yang dipercayakan kepadamu, sedangkan kamu mengetahui." (Q.S. Al-Anfal: 27). Selain itu, dalam hadist Nabi Muhammad SAW yang artinya, "Barang siapa menipu, maka ia bukan termasuk golongan kami." (H.R. Muslim). Kedua dalil ini menunjukkan bahwa menjaga kebenaran dan keutuhan informasi merupakan bagian dari ajaran Islam yang sangat ditekankan, bahkan menjadi ciri utama seorang muslim yang jujur dan dapat dipercaya.

Dalam konteks ini, konsep integritas digital melalui kriptografi dapat dianalisis melalui pendekatan hukum islam, khususnya metode qiyas. Qiyas merupakan proses analogi hukum yang digunakan ketika suatu fenomena baru tidak ditemukan secara langsung dalam nash (Al-Qur'an atau Al-hadist), tetapi memiliki kesamaan illat (alasan hukum) dengan fenomena lama yang sudah memiliki ketetapan hukum. Jika ditinjau dari prinsip ini, integritas digital yang dijaga melalui kriptografi memiliki illat yang sama dengan tindakan Nabi Muhammad SAW dalam menjaga dokumen, akad, atau informasi penting di masa beliau, yakni mencegah penyalahgunaan informasi dan memastikan keabsahan suatu transaksi

atau pesan. Pada masa Rasulullah, keabsahan dan keutuhan data dijaga melalui tulisan tangan, saksi, dan kepercayaan antar individu.

Dalam era sekarang hal itu dilakukan melalui sistem digital yang kompleks dan akurat secara matematis. Karena memiliki illat yang sama, yaitu menjaga kebenaran dan mencegah penipuan, maka menjaga integritas melalui kriptografi dapat dipandang sebagai bentuk modern dari nilai-nilai syariah. Bahkan, dalam konteks tertentu seperti perlindungan dokumen zakat, akad wakaf digital, dan transaksi daring yang bernilai hukum, menjaga integritas dapat bernilai wajib, sebab kerusakan atau kebocoran data dapat menimbulkan kerugian besar bagi individu maupun masyarakat. Ketika hak orang lain bergantung pada keutuhan suatu informasi digital, menjaga integritasnya adalah bagian dari amanah yang harus dipenuhi.

Dengan demikian, kriptografi bukan sekadar teknologi modern yang bersifat teknis, melainkan juga memiliki dimensi etika dan hukum dalam pandangan Islam. Menjaga integritas data melalui teknologi ini dapat dipandang sebagai upaya untuk mengamalkan perintah syariat dalam bentuk baru yang kontekstual. Hal ini menunjukkan bahwa Islam adalah agama yang tidak hanya relevan di masa lalu, tetapi juga fleksibel terhadap perkembangan zaman, termasuk dalam menerima dan mengarahkan pemanfaatan teknologi. Oleh karena itu, integrasi antara teknologi kriptografi dan nilai-nilai syariah tidak hanya diperlukan dalam tataran praktis, tetapi juga penting dikembangkan dalam kerangka keilmuan yang sistematis agar teknologi modern benar-benar mampu mendukung terbentuknya masyarakat digital yang amanah, adil, dan bermartabat.

2.3 Kajian Topik dengan Teori Pendukung

Penelitian ini disusun menggunakan beberapa teori pendukung seperti algoritma *paillier cryptosystem* yang termasuk salah satu enkripsi homomorfik sebagian pada kriptografi. Adapun kunci dari algoritma ini terletak pada proses enkripsi dan dekripsinya. Penulis ingin menghubungkan *paillier cryptosystem* dengan matriks permutasi untuk meningkatkan keamanan pesan teks. Dimana Matriks permutasi dalam *Paillier cryptosystem* diharapkan dapat membuat proses enkripsi dan dekripsi menjadi lebih aman. Langkah awal untuk proses enkripsi dan dekripsinya yaitu menginput plainteks. Kemudian inputkan juga *key generator* berupa nilai p dan q berupa bilangan prima ukuran besar secara acak. Jika semua data telah terinput maka proses perhitungan enkripsi dan dekripsi akan dilakukan menggunakan algoritma *paillier cryptosystem*. Adapun hasil *output* dari program ini yaitu berupa plainteks.

BAB III
METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan yaitu penelitian kualitatif yaitu penelitian yang bersifat deskriptif dan cenderung menggunakan analisis. Data yang akan diteliti bersifat deskriptif dan dikumpulkan dalam bentuk text.

3.2 Pra Penelitian

Pra penelitian merupakan tahapan awal dalam pelaksanaan suatu penelitian yang bertujuan untuk membangun pondasi teoritis, konseptual, dan teknis sebagai dasar pelaksanaan penelitian utama. Tahap ini sangat penting agar pelaksanaan penelitian dapat berjalan secara sistematis, terarah, dan sesuai dengan tujuan yang telah ditetapkan. Menurut (Moloeng, 2017), pra-penelitian dimaksudkan untuk memahami lebih jauh objek penelitian serta untuk menentukan pendekatan dan strategi yang tepat dalam menjawab rumusan masalah.

Dalam penelitian ini, pra-penelitian dilakukan melalui serangkaian kegiatan yang mencakup identifikasi permasalahan, kajian literatur, perumusan tujuan, serta perencanaan teknis. Adapun penjelasan masing-masing kegiatan dalam tahap pra penelitian dijabarkan sebagai berikut:

3.3 Identifikasi Permasalahan

Peneliti memulai kegiatan pra-penelitian dengan mengidentifikasi permasalahan yang relevan dalam bidang keamanan informasi, khususnya pada sistem pengamanan pesan teks. Pesan teks yang dikirimkan melalui media komunikasi digital rentan terhadap serangan pihak ketiga seperti penyadapan, pengubahan isi pesan, atau pencurian data. Oleh karena itu, diperlukan sebuah

sistem kriptografi yang dapat menjamin kerahasiaan, integritas, dan keaslian data. Hal ini sejalan dengan prinsip dasar keamanan informasi yang dikemukakan oleh (Stallings, 2017),yaitu confidentiality, integrity, dan authenticity.

3.2.1 Studi Literatur

Selanjutnya, peneliti melakukan kajian literatur untuk memperdalam pemahaman terkait teori-teori yang mendasari penelitian. Kajian dilakukan terhadap sumber-sumber ilmiah seperti jurnal, buku teks, dan laporan penelitian yang membahas konsep kriptografi modern, kriptografi homomorfik, dan teknik permutasi data. Algoritma Paillier dipilih karena memiliki sifat homomorfik yang memungkinkan operasi matematika dilakukan pada data terenkripsi tanpa perlu melakukan dekripsi terlebih dahulu (Katz, 2014). Selain itu, penggunaan matriks permutasi ditujukan untuk meningkatkan kerahasiaan data dengan cara mengacak posisi karakter atau blok *ciphertext* sebelum dikirimkan.

3.2.2 Perumusan Topik, Tujuan, dan Rumusan Masalah

Berdasarkan hasil studi literatur dan identifikasi masalah, peneliti merumuskan topik penelitian dengan fokus pada penerapan dua metode kriptografi dalam satu sistem, yaitu Paillier Cryptosystem dan matriks permutasi. Peneliti menyusun tujuan penelitian untuk mengetahui bagaimana algoritma tersebut dapat diimplementasikan dan sejauh mana efektivitasnya dalam menjaga keamanan pesan teks. Rumusan masalah disusun secara spesifik agar dapat dijawab melalui proses implementasi dan analisis deskriptif.

3.2.3 Perencanaan Teknis Penelitian

Pada tahap ini, peneliti merancang aspek teknis penelitian yang mencakup pemilihan alat bantu dan perangkat lunak, perancangan alur enkripsi dan dekripsi,

serta format input dan output data. Bahasa pemrograman Python digunakan sebagai alat bantu implementasi karena fleksibilitasnya dan ketersediaan pustaka kriptografi yang lengkap. Perencanaan ini juga mencakup penyusunan tahapan implementasi, pengujian, dan dokumentasi hasil, yang semuanya mendukung pencapaian tujuan penelitian.

Dengan tahapan pra-penelitian yang matang, peneliti dapat melanjutkan ke proses implementasi algoritma secara lebih terstruktur, serta memperoleh hasil yang relevan dan valid sesuai dengan tujuan penelitian.

3.4 Tahapan Penelitian

Penelitian ini terdiri dari tiga tahapan yaitu tahap pembangkitan kunci, tahap enkripsi, dan tahap dekripsi. Pada tahap pembangkitan kunci menghasilkan bilangan prima acak yang nantinya digunakan sebagai kunci privat dan kunci publik. Pada tahap enkripsi akan mengubah plainteks menjadi cipherteks, sedangkan pada tahap dekripsi akan mengubah cipherteks menjadi plainteks. Berikut tahapan-tahapan pada implementasi Paillier cryptosystem dan Matriks permutasi:

1. Proses Enkripsi

a. Menentukan plainteks

b. Konversi plaintext ke numerik

c. Proses pembangkitan kunci:

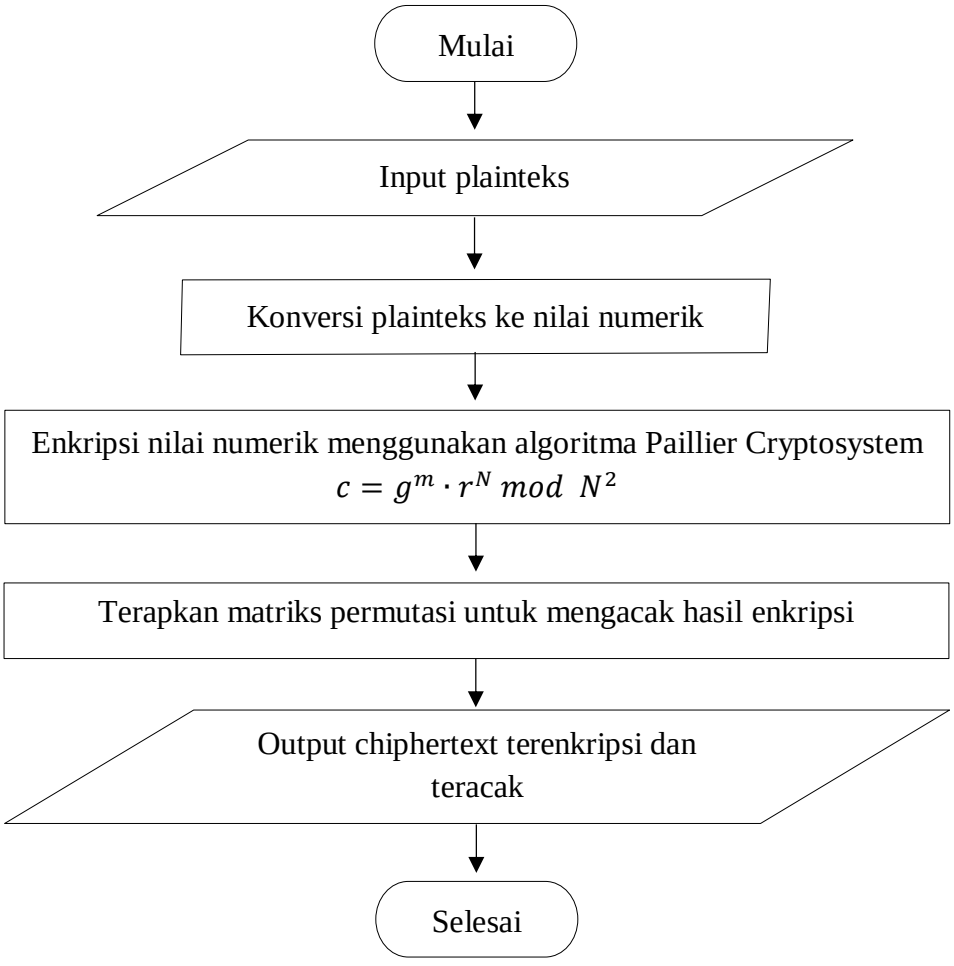
   1. Pilih dua bilangan prima sembarang misal p dan q yang memenuhi syarat $gcd\big(p \cdot q, (p-1)(q-1)\big) = 1$.
   2. Hitung nilai N yang merupakan hasil perkalian nilai p dan q
   3. Hitung nilai $\lambda = lcm(p-1, q-1)$.

4. Pilih bilangan bulat acak $g \in Z_{N^2}^*$, dengan syarat $1 \leq g < N^2$, serta memenuhi $gcd(g, N^2) = 1$ dan $gcd(L(g^\lambda \ mod \ N^2), N) = 1$
5. Pilih bilangan bulat acak r dengan syarat $0 \leq r < N$ dan $gcd(r, N) = 1$.
6. Hitung nilai μ menggunakan basis g yang telah dipilih, dengan rumus $\mu = \left(L\left(g^\lambda mod \ N^2\right)\right)^{-1} mod \ N$, dimana fungsi $L(u)$ didefinisikan sebagai $L(u) = (\frac{u-1}{N})$ dengan $u \ = \ g^\lambda \ mod \ N^2$.

d. Enkripsi menggunakan algoritma Paillier Cryptosystem $c = g^m \cdot r^N \ mod \ N^2$

e. Terapkan matriks permutasi untuk mengacak hasil enkripsi

f. Output chiphertext terenkripsi dan teracak



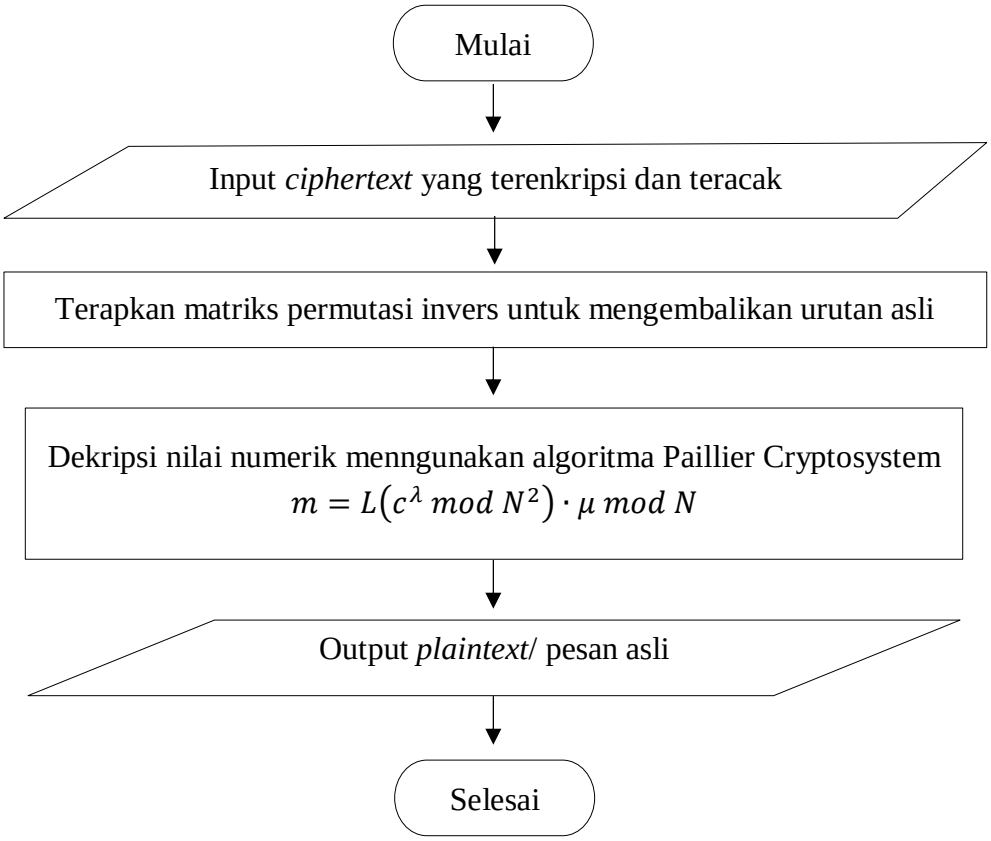
Gambar 3 Flowchart Enkripsi Algoritma Paillier Cryptosystem

2. Proses Dekripsi

a. Terapkan invers matriks permutasi untuk mengembalikan urutan asli

b. Dekripsi nilai numerik menngunakan algoritma Paillier Cryptosystem $m = L\left(c^{\lambda} \bmod N^2\right) \cdot \mu \bmod N$

c. Keluaran *plaintext*/ pesan asli



Gambar 4 Flowchart Dekripsi Paillier Cryptosystem

BAB IV
HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dari implementasi algoritma kriptografi Paillier dan teknik Matriks Permutasi dalam mengamankan pesan teks. Tujuan dari penelitian ini adalah untuk mengimplementasikan algoritma Paillier Cryptosystem dan Matriks Permutasi, baik melalui perhitungan manual untuk menunjukkan mekanisme secara rinci, maupun dengan bantuan simulasi menggunakan bahasa pemrograman Python untuk menguji implementasi pada skala lebih besar.

Algoritma Paillier merupakan kriptosistem asimetris yang bersifat homomorfik aditif, memungkinkan operasi matematika dilakukan langsung pada *ciphertext*. Sementara itu, Matriks Permutasi digunakan untuk mengacak urutan *ciphertext* guna memperkuat keamanan data. Kombinasi kedua teknik ini diimplementasikan untuk menunjukkan efektivitas dalam menjaga kerahasiaan pesan teks.

Proses implementasi dimulai dengan mengubah karakter ke dalam bentuk ASCII, mengenkripsinya menggunakan Paillier, dan mengacak hasil enkripsi menggunakan Matriks Permutasi. Selanjutnya dilakukan proses dekripsi untuk memperoleh kembali pesan asli. Setiap tahapan dijelaskan secara manual per huruf agar transparan dan terukur, kemudian diikuti oleh simulasi Python yang menunjukkan hasil serupa dalam skala lebih besar dan otomatis.

Pada bab ini akan membahas tentang proses enkripsi dan dekripsi pesan teks menggunakan algoritma paillier cryptosystem dan pemanfaatan matriks permutasi dalam pengamanan pesan teks.

4.1 Implementasi Matriks Permutasi pada Paillier Cryptosystem

Untuk meningkatkan tingkat keamanan pesan terenkripsi, dilakukan penerapan matriks permutasi setelah proses enkripsi dengan algoritma Paillier Cryptosystem. Matriks permutasi bertujuan untuk menyusun ulang posisi elemen-elemen *ciphertext* dalam blok-blok tertentu sehingga menghasilkan susunan yang acak dan tidak mudah dianalisis secara langsung. Strategi ini secara kriptografis memperkuat kerahasiaan.

4.1.1. Algoritma Enkripsi

1. Buat *Plaintext* yang akan dienkripsi, berupa rangkaian karakter.
2. Konversi ke representasi numerik, setiap karakter pada plaintext dikonversikan ke dalam bentuk numerik menggunakan standar kode ASCII.
3. Pembangkitan Kunci Paillier
   a. Pilih dua bilangan prima secara acak, yaitu p dan q, yang independen satu sama lain.
   b. Hitung nilai $N = (p \cdot q)$ dan N^2.
   c. Hitung nilai $\lambda = lcm(\mathrm{p}-1, \mathrm{q}-1)$.
   d. Pilih bilangan bulat acak $g \in Z^*_{N^2}$, dengan syarat $1 \leq g < N^2$ serta memenuhi $gcd(g, N^2) = 1$ dan $gcd(L(g^{\lambda} \bmod N^2), N) = 1$
   e. Hitung nilai μ menggunakan basis g yang telah dipilih, dengan rumus $\mu = \left(L\left(g^{\lambda} mod\ N^2\right)\right)^{-1} mod\ N$, dimana fungsi $L(u)$ didefinisikan sebagai $L(u) = (\frac{u-1}{N})$ dengan $u = g^{\lambda} \bmod N^2$.

   Hasil dari proses ini adalah kunci publik (N, g) dan kunci privat (λ, μ).
4. Enkripsi dengan Algoritma Paillier

Untuk setiap nilai numerik m, Pilih bilangan acak $r \in Z_{N^2}^*$

Hitung ciphertext dengan rumus: $c = g^m \cdot r^N \, mod \, N^2$

5. Segmentasi Ciphertext

   Seluruh ciphertext yang diperoleh dibagi menjadi blok-blok berukuran tetap, misalnya berukuran 1×5, agar dapat diproses secara modular.

6. Penerapan Matriks Permutasi

   Setiap blok ciphertext dikalikan dengan matriks permutasi biner berukuran sesuai, menghasilkan urutan elemen ciphertext yang teracak: $C' = C \times P$. Matriks permutasi yang digunakan berbeda untuk setiap blok guna menghindari pola berulang.

7. Output Ciphertext Terenkripsi dan Teracak

   Hasil akhir dari proses enkripsi adalah kumpulan ciphertext yang telah melalui transformasi paillier dan permutasi posisi elemen. ciphertext inilah yang siap untuk disimpan atau dikirim secara aman.

4.1.2. Simulasi Enkripsi Pesan

Enkripsi dalam algoritma Paillier merupakan proses matematis yang mengubah informasi *plaintext* menjadi *ciphertext* dengan menggunakan kunci publik. Proses ini dirancang sedemikian rupa agar hanya dapat dibalik oleh pihak yang memiliki kunci privat, serta memiliki sifat homomorfik aditif yang memungkinkan operasi matematika dilakukan langsung pada data terenkripsi.

Paillier termasuk dalam kelompok kriptografi asimetris, sehingga skema kuncinya terdiri dari dua bagian yaitu kunci publik yang digunakan untuk enkripsi dan kunci privat yang digunakan untuk dekripsi. Dalam konteks komunikasi

rahasia, pengirim pesan hanya memerlukan kunci publik untuk mengamankan pesan, sedangkan kunci privat disimpan oleh penerima.

Dalam proses enkripsi pesan, pengirim menentukan pesan asli (*plaintext*) yang akan disandikan menggunakan algoritma paillier cryptosystem. Berikut ini langkah-langkah dalam proses enkripsi pesan:

1. Menentukan *plaintext*. Misalkan pesan yang dikirim berisi kalimat **18610067 MATEMATIKA UINMA**
2. Mengkonversi plaintext menjadi kode ASCII 125.

Tabel 4 Simulasi Representasi ASCII

Plaintext	Kode ASCII	*Plaintext*	Kode ASCII
1	49	M	77
8	56	A	65
6	54	T	84
1	49	I	73
0	48	K	75
0	48	A	65
6	54	Spasi	32
7	55	U	85
Spasi	32	I	73
M	77	N	78
A	65	M	77
T	84	A	65
E	69		

3. Pembangkitan kunci

   a. Pilih dua bilangan prima ($p\ dan\ q$): p = 163, q = 191

   b. Hitung nilai N dan N^2:

      $N = p \cdot q = 163 \cdot 191 = 31133$

      $N^2 = 311332 = 969263689$

   c. Hitung nilai $\lambda = lcm(p-1, q-1) = lcm(162, 190) = 15390$

   d. Pilih $g \in Z_{N^2}$, kita ambil: $g = 31134$

      Validasi nilai g, Harus memenuhi: $gcd\left(L\left(\mathrm{g}^{\lambda} \bmod N^2\right), N\right) = 1$

      $u = (g^{\lambda}\ mod\ N^2) = (31134^{15390} \bmod 31133^2) = 479136871$

      $L(g^{\lambda}\ mod\ N^2) = L(479136871) = \frac{479136871-1}{31133} = \frac{479136870}{31133} = 15390$

      Sehingga $gcd(15390,31133) = 1$

      Dari hasil perhitungan diatas ditemukan bahwa:

      Kunci public $(N, g) = (31133, 31134)$

   e. Pilih $r \in Z_{N^2}^*$, dengan syarat $0 \leq r < N$ dan gcd(r, N) = 1

4. Selanjutnya akan dilakukan proses enkripsi pesan pada algoritma paillier cryptosystem dengan rumus $c = g^m \cdot r^N\ mod\ N^2$.

Tabel 5 Simulasi Enkripsi Pesan

Plain text	m	r	Enkripsi	*Ciphertext*
1	49	2	$c = g^m \cdot r^N\ mod\ N^2$ $= 31134^{49} \cdot 2^{31133}\ mod\ 31133^2$ $= 850723011$	$c = 850723011$
8	56	3	$c = g^m \cdot r^N\ mod\ N^2$	$c = 512828523$

			$= 31134^{56} \cdot 3^{31133}\ mod\ 31133^2$ $= 512828523$	
6	54	4	$c = g^m \cdot r^N\ mod\ N^2$ $= 31134^{54} \cdot 4^{31133}\ mod\ 31133^2$ $= 356366282$	$c = 356366282$
1	49	5	$c = g^m \cdot r^N\ mod\ N^2$ $= 31134^{49} \cdot 5^{31133}\ mod\ 31133^2$ $= 570833799$	$c = 570833799$
0	48	6	$c = g^m \cdot r^N\ mod\ N^2$ $= 31134^{48} \cdot 6^{31133}\ mod\ 31133^2$ $= 862752353$	$c = 862752353$
0	48	7	$c = g^m \cdot r^N\ mod\ N^2$ $= 31134^{48} \cdot 7^{31133}\ mod\ 31133^2$ $= 750883227$	$c = 750883227$
6	54	8	$c = g^m \cdot r^N\ mod\ N^2$ $= 31134^{54} \cdot 8^{31133}\ mod\ 31133^2$ $= 205497622$	$c = 205497622$
7	55	9	$c = g^m \cdot r^N\ mod\ N^2$ $= 31134^{55} \cdot 9^{31133}\ mod\ 31133^2$ $= 823557145$	$c = 823557145$
Spasi	32	10	$c = g^m \cdot r^N\ mod\ N^2$ $= 31134^{32} \cdot$ $10^{31133}\ mod\ 31133^2$	$c = 532690126$

			$= 532690126$	
M	77	11	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{77} \cdot 11^{31133} \, mod \, 31133^2$ $= 197698592$	$c = 197698592$
A	65	12	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{65} \cdot 12^{31133} \, mod \, 31133^2$ $= 442837872$	$c = 442837872$
T	84	13	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{84} \cdot 13^{31133} \, mod \, 31133^2$ $= 812521579$	$c = 812521579$
E	69	14	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{69} \cdot 14^{31133} \, mod \, 31133^2$ $= 397468337$	$c = 397468337$
M	77	15	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{77} \cdot 15^{31133} \, mod \, 31133^2$ $= 745697381$	$c = 745697381$
A	65	16	$c = g^m \cdot r^N \, mod \, N^2$	$c = 947202143$

			$= 31134^{65} \cdot$ $16^{31133} \ mod \ 31133^2$ $= 947202143$	
T	84	17	$c = g^m \cdot r^N \ mod \ N^2$ $= 31134^{84} \cdot$ $17^{31133} \ mod \ 31133^2$ $= 757440499$	$c = 757440499$
I	73	`18	$c = g^m \cdot r^N \ mod \ N^2$ $= 31134^{73} \cdot$ $18^{31133} \ mod \ 31133^2$ $= 593199099$	$c = 593199099$
K	75	19	$c = g^m \cdot r^N \ mod \ N^2$ $= 31134^{75} \cdot$ $19^{31133} \ mod \ 31133^2$ $= 606849109$	$c = 606849109$
A	65	20	$c = g^m \cdot r^N \ mod \ N^2$ $= 31134^{65} \cdot$ $20^{31133} \ mod \ 31133^2$ $= 626111816$	$c = 626111816$
Spasi	32	21	$c = g^m \cdot r^N \ mod \ N^2$ $= 31134^{32} \cdot$ $21^{31133} \ mod \ 31133^2$ $= 155244901$	$c = 155244901$

U	85	22	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{85} \cdot$ $22^{31133} \, mod \, 31133^2$ $= 163131862$	$c = 163131862$
I	73	23	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{73} \cdot$ $23^{31133} \, mod \, 31133^2$ $= 783163452$	$c = 783163452$
N	78	24	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{78} \cdot$ $24^{31133} \, mod \, 31133^2$ $= 523876378$	$c = 523876378$
M	77	25	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{77} \cdot$ $25^{31133} \, mod \, 31133^2$ $= 598428719$	$c = 598428719$
A	65	26	$c = g^m \cdot r^N \, mod \, N^2$ $= 31134^{65} \cdot$ $26^{31133} \, mod \, 31133^2$ $= 781097392$	$c = 781097392$

Dari perhitungan proses enkripsi menggunakan algoritma paillier cryptosystem diatas diperoleh *ciphertext* :

(850723011, 512828523, 356366282, 570833799, 862752353, 750883227, 205497622, 823557145, 532690126, 197698592, 442837872, 812521579,

397468337, 745697381, 947202143, 757440499, 593199099, 606849109, 626111816, 155244901, 163131862, 783163452, 523876378, 598428719, 781097392).

5. Segmentasi ciphertext

Setelah proses enkripsi selesai dilakukan menggunakan algoritma Paillier Cryptosystem, setiap elemen *ciphertext* kemudian diproses lebih lanjut melalui mekanisme permutasi blok. Langkah ini dilakukan untuk meningkatkan kompleksitas penyusunan data terenkripsi dan memberikan lapisan perlindungan tambahan terhadap kemungkinan analisis pola oleh pihak yang tidak berwenang. Teknik yang diterapkan dalam implementasi ini adalah membagi *ciphertext* menjadi beberapa blok kecil, masing-masing terdiri dari lima elemen, yang selanjutnya diproses menggunakan matriks permutasi berukuran 5 × 5.

Setiap blok *ciphertext* yang terbentuk dianggap sebagai sebuah vektor berdimensi 1×5. Selanjutnya, dilakukan permutasi terhadap posisi elemen dalam vektor tersebut dengan menggunakan matriks permutasi yang valid, yaitu matriks biner di mana setiap baris dan kolom memiliki tepat satu elemen bernilai 1. Dalam implementasi ini, setiap blok menggunakan matriks permutasi yang berbeda untuk menghindari pola berulang yang dapat mengurangi efektivitas keamanan.

Dari Proses enkripsi diatas didapatkan:

a. Blok *ciphertext* awal (sebelum permutasi)

(850723011, 512828523, 356366282, 570833799, 862752353, 750883227, 205497622, 823557145, 532690126, 197698592, 442837872, 812521579, 397468337, 745697381, 947202143, 757440499, 593199099, 606849109,

626111816, 155244901, 163131862, 783163452, 523876378, 598428719, 781097392)

b. Ciphertext disegmentasi menjadi blok-blok berdimensi 1 x 5 untuk setiap tahap permutasi

Blok 1: [850723011, 512828523, 356366282, 570833799, 862752353]

Blok 2: [750883227, 205497622, 823557145, 532690126, 197698592]

Blok 3: [442837872, 812521579, 397468337, 745697381, 947202143]

Blok 4: [757440499, 593199099, 606849109, 626111816, 155244901]

Blok 5: [163131862, 783163452, 523876378, 598428719, 781097392]

6. Penerapan matriks permutasi

Permutasikan setiap blok dengan matriks acak 5 x 5:

a. Blok 1 sebelum permutasi :

[850723011, 512828523, 356366282, 570833799, 862752353]

Matriks permutasi 1

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Blok 1 Setelah permutasi:

[356366282, 512828523, 570833799, 850723011, 862752353]

b. Blok 2 sebelum permutasi:

[750883227, 205497622, 823557145, 532690126, 197698592]

Matriks permutasi 2

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Blok 2 setelah permutasi:

[197698592, 205497622, 532690126, 823557145, 750883227]

c. Blok 3 sebelum permutasi:

[442837872, 812521579, 397468337, 745697381, 947202143]

Matriks permutasi 3

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Blok 3 setelah permutasi:

[812521579, 397468337, 442837872, 947202143, 745697381]

d. Blok 4 sebelum permutasi:

[757440499, 593199099, 606849109, 626111816, 155244901]

Matriks permutasi 4

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Blok 4 setelah permutasi:

[626111816, 155244901, 757440499, 593199099, 606849109]

e. Blok 5 sebelum permutasi:

[163131862, 783163452, 523876378, 598428719, 781097392]

Matriks permutasi 5

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Blok 5 setelah permutasi:

[781097392, 163131862, 598428719, 523876378, 783163452]

7. Output ciphertext terenkripsi dan teracak

(356366282, 512828523, 570833799, 850723011, 862752353, 197698592, 205497622, 532690126, 823557145, 750883227, 812521579, 397468337, 442837872, 947202143, 745697381, 626111816, 155244901, 757440499, 593199099, 606849109, 781097392, 163131862, 598428719, 523876378, 783163452)

4.1.3. Algoritma Dekripsi

1. Masukkan *ciphertext* hasil enkripsi dan permutasi yang akan didekripsi.
2. Segmentasi *ciphertext* ke dalam blok-blok berukuran tetap (misalnya 1× 5) sesuai dengan segmentasi yang digunakan saat proses enkripsi. Konsistensi segmentasi sangat penting untuk keberhasilan invers permutasi.
3. Terapkan invers dari matriks permutasi yang digunakan pada saat enkripsi untuk setiap blok ciphertext, sehingga diperoleh urutan elemen ciphertext yang sesuai dengan hasil enkripsi awal: $C = C' \times P^{-1}$.
4. Dekripsi menggunakan rumus: $m = L\left(c^{\lambda} \bmod N^2\right) \cdot \mu \bmod N$ dimana fungsi $L(u)$ didefinisikan sebagai $L(u) = (\frac{u-1}{N})$ dengan $u = c^{\lambda} \bmod N^2$
5. Setiap nilai numerik hasil dekripsi dikonversikan kembali ke karakter ASCII untuk membentuk plaintext.
6. Gabungkan seluruh karakter hasil dekripsi untuk memperoleh pesan asli dalam bentuk teks.

4.1.4. Simulasi Dekripsi Pesan

Setelah proses permutasi dilakukan terhadap setiap blok *ciphertext*, posisi elemen-elemen dalam blok telah berubah sesuai pola yang ditentukan oleh matriks

permutasi. Karena urutan elemen *ciphertext* berperan penting dalam proses dekripsi, maka perlu dilakukan langkah invers permutasi untuk mengembalikan susunan elemen ke bentuk awal. Langkah ini tidak mengubah nilai *ciphertext*, namun berfungsi untuk memastikan bahwa proses dekripsi dapat dilakukan secara akurat dan menghasilkan *plaintext* yang sesuai dengan pesan asli.

1. Input *ciphertext*

   (356366282, 512828523, 570833799, 850723011, 862752353, 197698592, 205497622, 532690126, 823557145, 750883227, 812521579, 397468337, 442837872, 947202143, 745697381, 626111816, 155244901, 757440499, 593199099, 606849109, 781097392, 163131862, 598428719, 523876378, 783163452)

2. Segmentasi *ciphertext*

Blok 1: [356366282, 512828523, 570833799, 850723011, 862752353]

Blok 2: [197698592, 205497622, 532690126, 823557145, 750883227]

Blok 3: [812521579, 397468337, 442837872, 947202143, 745697381]

Blok 4: [626111816, 155244901, 757440499, 593199099, 606849109]

Blok 5: [781097392, 163131862, 598428719, 523876378, 783163452]

3. Inver matriks permutasi

   a. Blok 1 setelah permutasi:

   [356366282, 512828523, 570833799, 850723011, 862752353]

   Matriks invers permutasi 1

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Blok 1 setelah invers permutasi:

[850723011, 512828523, 356366282, 570833799, 862752353]

b. Blok 2 setelah permutasi:

[197698592, 205497622, 532690126, 823557145, 750883227]

Matriks invers permutasi 2

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Blok 2 setelah invers permutasi:

[750883227, 205497622, 823557145, 532690126, 197698592]

c. Blok 3 setelah permutasi:

[812521579, 397468337, 442837872, 947202143, 745697381]

Matriks invers permutasi 3

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Blok 3 setelah invers permutasi:

[442837872, 812521579, 397468337, 745697381, 947202143]

d. Blok 4 setelah permutasi:

[626111816, 155244901, 757440499, 593199099, 606849109]

Matriks invers permutasi 4

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Blok 4 setelah invers permutasi:

[757440499, 593199099, 606849109, 626111816, 155244901]

e. Blok 5 setelah permutasi:

[781097392, 163131862, 598428719, 523876378, 783163452]

Matriks invers permutasi 5

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Blok 5 setelah invers permutasi:

[163131862, 783163452, 523876378, 598428719, 781097392]

Ciphertext setelah inverse permutasi :

(850723011, 512828523, 356366282, 570833799, 862752353, 750883227, 205497622, 823557145, 532690126, 197698592, 442837872, 812521579, 397468337, 745697381, 947202143, 757440499, 593199099, 606849109, 626111816, 155244901, 163131862, 783163452, 523876378, 598428719, 781097392)

Dengan diterapkannya invers permutasi pada setiap blok *ciphertext*, susunan elemen telah berhasil dikembalikan ke urutan awal sebagaimana hasil enkripsi asli. Proses ini memastikan bahwa *ciphertext* telah berada dalam struktur yang benar dan siap untuk didekripsi tanpa kehilangan integritas informasi.

4. Dekripsi dengan Algoritma Paillier

Setelah susunan elemen *ciphertext* dikembalikan ke urutan semula melalui proses invers permutasi, struktur data terenkripsi telah sesuai dengan format hasil enkripsi awal. Dengan kondisi tersebut, tahap selanjutnya adalah melakukan

dekripsi menggunakan algoritma Paillier. Proses ini bertujuan untuk mengubah *ciphertext* kembali menjadi *plaintext* dengan menerapkan kunci privat dan operasi matematis yang menjadi inti dari skema Paillier, sehingga pesan asli dapat direkonstruksi secara tepat.

Berikut mekanisme dekripsi algoritma paillier cryptosystem:

Hitung nilai untuk μ dari basis g yang nilainya memenuhi syarat $\mu = \left(L\left(g^{\lambda} mod\ N^2\right)\right)^{-1} mod\ N$ dengan fungsi $L(u) = (\frac{u-1}{N})$.

Hitung $\mu = \left(L\left(g^{\lambda} mod\ N^2\right)\right)^{-1} mod\ N$, Dimana dari perhitungan diatas didapatkan nilai nilai $u = g^{\lambda} mod\ N^2$ = 479136871 dan N = 31133

Kemudian hitung fungsi $L(\mathrm{u}) = \frac{u-1}{N} = \frac{479136871-1}{31133} = \frac{479136870}{31133} = 15390$

Sehingga $\mu = 15390^{-1}\ mod\ 31133$=7673 (karena $gcd(15390, 31133) = 1$ maka invers modularnya ada, sehingga didapatkan $15390\ x\ 7673 \equiv 1\ mod\ 31133$)

Kunci privat (λ, μ) = (15390, 7673)

Dekripsi menggunakan rumus: $m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$ dimana $L(u) = (\frac{u-1}{N})$ dengan $u\ =\ c^{\lambda}\ mod\ N^2$

Tabel 6 Simulasi Dekripsi Pesan

Cipher text (c)	**Dekripsi**
850723011	$m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$ $= L(850723011^{15390}\ mod\ 31133^2) \cdot \mu\ mod\ N$ $= L(215378095) \cdot \mu\ mod\ N$ Dimana:

	$L\ (215378095) = \frac{215378095-1}{31133} = \frac{215378094}{31133} = 6918$ Sehingga: $m = 6918 \cdot \mu\ mod\ N$ $= 6918 \cdot 7673\ mod\ 31133$ $= 54081814\ mod\ 31133$ $= 49$
512828523	$m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$ $= L(512828523^{15390}\ mod\ 31133^2) \cdot \mu\ mod\ N$ $= L(661545118) \cdot \mu\ mod\ N$ Dimana: $L\ (661545118) = \frac{661545118-1}{31133} = \frac{661545117}{31133} = 21249$ Sehingga: $m = 21249 \cdot \mu\ mod\ N$ $= 21249 \cdot 7673\ mod\ 31133$ $= 163043577\ mod\ 31133$ $= 56$
356366282	$m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot\ \mu\ mod\ N$ $= L(356366282^{15390}\ mod\ 31133^2) \cdot \mu\ mod\ N$ $= L(672535067) \cdot \mu\ mod\ N$ Dimana: $L\ (672535067) = \frac{672535067-1}{31133} = \frac{672535066}{31133} = 21602$ Sehingga: $m = 21602 \cdot \mu\ mod\ N$

	$= 21062 \cdot 7673 \; mod \; 31133$ $= 165752146 \; mod \; 31133$ $= 54$
570833799	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(570833799^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(215378095) \cdot \mu \; mod \; N$ Dimana: L (215378095) = $\frac{215378095-1}{31133} = \frac{215378094}{31133}$ = 6918 Sehingga: $m = 6918 \cdot \mu \; mod \; N$ $= 6918 \cdot 7673 \; mod \; 31133$ $= 53081814 \; mod \; 31133$ $= 49$
862752353	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(862752353^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(705504914) \cdot \mu \; mod \; N$ Dimana: L (705504914) = $\frac{705504914-1}{31133} = \frac{705504913}{31133}$ = 22661 Sehingga: $m = 22661 \cdot \mu \; mod \; N$ $= 22661 \cdot 7673 \; mod \; 31133$ $= 173877853 \; mod \; 31133$ $= 48$

750883227	$m = L\left(c^{\lambda} \bmod N^2\right) \cdot \mu \bmod N$ $= L(750883227^{15390} \bmod 31133^2) \cdot \mu \bmod N$ $= L(705504914) \cdot \mu \bmod N$ Dimana: $L\ (705504914) = \frac{705504914-1}{31133} = \frac{705504913}{31133} = 22661$ Sehingga: $m = 22661 \cdot \mu \bmod N$ $= 22661 \cdot 7673 \bmod 31133$ $= 173877853 \bmod 31133$ $= 48$
205497622	$m = L\left(c^{\lambda} \bmod N^2\right) \cdot \mu \bmod N$ $= L(205497622^{15390} \bmod 31133^2) \cdot \mu \bmod N$ $= L(672535067) \cdot \mu \bmod N$ Dimana: $L\ (672535067) = \frac{672535067-1}{31133} = \frac{672535066}{31133} = 21602$ Sehingga: $m = 21602 \cdot \mu \bmod N$ $= 21602 \cdot 7673 \bmod 31133$ $= 165752146 \bmod 31133$ $= 54$
823557145	$m = L\left(c^{\lambda} \bmod N^2\right) \cdot \mu \bmod N$ $= L(823557145^{15390} \bmod 31133^2) \cdot \mu \bmod N$ $= L(182408248) \cdot \mu \bmod N$

	Dimana: $L\ (182408248) = \frac{182408248-1}{31133} = \frac{182408247}{31133} = 5859$ Sehingga: $m = 5859 \cdot \mu\ mod\ N$ $= 5859 \cdot 7673\ mod\ 31133$ $= 44956107\ mod\ 31133$ $= 55$
532690126	$m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$ $= L(532690126^{15390}\ mod\ 31133^2) \cdot \mu\ mod\ N$ $= L(793424506) \cdot \mu\ mod\ N$ Dimana: $L\ (793424506) = \frac{793424506-1}{31133} = \frac{793424505}{31133} = 25485$ Sehingga: $m = 25485 \cdot \mu\ mod\ N$ $= 25485 \cdot 7673\ mod\ 31133$ $= 195546405\ mod\ 31133$ $= 32$
197698592	$m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$ $= L(197698592^{15390}\ mod\ 31133^2) \cdot \mu\ mod\ N$ $= L(61518809) \cdot \mu\ mod\ N$ Dimana: $L\ (61518809) = \frac{61518809-1}{31133} = \frac{61518808}{31133} = 1976$ Sehingga:

	$m = 1976 \cdot \mu \; mod \; N$ $= 1976 \cdot 7673 \; mod \; 31133$ $= 15161848 \; mod \; 31133$ $= 77$
442837872	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(442837872^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(127458503) \cdot \mu \; mod \; N$ Dimana: $L\,(127458503) = \frac{127458503-1}{31133} = \frac{127458502}{31133} = 4094$ Sehingga: $m = 4094 \cdot \mu \; mod \; N$ $= 4094 \cdot 7673 \; mod \; 31133$ $= 31413262 \; mod \; 31133$ $= 65$
812521579	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(812521579^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(507685832) \cdot \mu \; mod \; N$ Dimana: $L\,(507685832) = \frac{507685832-1}{31133} = \frac{507685831}{31133} = 16307$ Sehingga: $m = 16307 \cdot \mu \; mod \; N$ $= 16307 \cdot 7673 \; mod \; 31133$ $= 125123611 \; mod \; 31133$

	$= 84$
397468337	$m = L\left(c^{\lambda} \bmod N^2\right) \cdot \mu \bmod N$ $= L(397468337^{15390} \bmod 31133^2) \cdot \mu \bmod N$ $= L(105478605) \cdot \mu \bmod N$ Dimana: $L\ (105478605) = \frac{105478605-1}{31133} = \frac{105478604}{31133} = 3388$ Sehingga: $m = 3388 \cdot \mu \bmod N$ $= 3388 \cdot 7673 \bmod 31133$ $= 25996124 \bmod 31133$ $= 69$
745697381	$m = L\left(c^{\lambda} \bmod N^2\right) \cdot \mu \bmod N$ $= L(745697381^{15390} \bmod 31133^2) \cdot \mu \bmod N$ $= L(61518809) \cdot \mu \bmod N$ Dimana: $L\ (61518809) = \frac{61518809-1}{31133} = \frac{61518808}{31133} = 1976$ Sehingga: $m = 1976 \cdot \mu \bmod N$ $= 1976 \cdot 7673 \bmod 31133$ $= 15161848 \bmod 31133$ $= 77$
947202143	$m = L\left(c^{\lambda} \bmod N^2\right) \cdot \mu \bmod N$ $= L(947202143^{15390} \bmod 31133^2) \cdot \mu \bmod N$

	$= L(127458503) \cdot \mu \; mod \; N$ Dimana: $L\,(127458503) = \frac{127458503-1}{31133} = \frac{127458503}{31133} = 4094$ Sehingga: $m = \; 4094 \cdot \mu \; mod \; N$ $= \; 4094 \cdot 7673 \; mod \; 31133$ $= \; 31413262 \; mod \; 31133$ $= \; 65$
757440499	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(757440499^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(507685832) \cdot \mu \; mod \; N$ Dimana: $L\,(507685832) = \frac{507685832-1}{31133} = \frac{507685831}{31133} = 16307$ Sehingga: $m = \; 16307 \cdot \mu \; mod \; N$ $= \; 16307 \cdot 7673 \; mod \; 31133$ $= \; 125123611 \; mod \; 31133$ $= \; 84$
593199099	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(593199099^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(83498707) \cdot \mu \; mod \; N$ Dimana: $L\,(83498707) = \frac{83498707-1}{31133} = \frac{83498706}{31133} = 2682$

	Sehingga: $m = 2682 \cdot \mu \; mod \; N$ $= 2682 \cdot 7673 \; mod \; 31133$ $= 20578986 \; mod \; 31133$ $= 73$
606849109	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(606849109^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(72508758) \cdot \mu \; mod \; N$ Dimana: $L\,(72508758) = \frac{72508758-1}{31133} = \frac{72508757}{31133} = 2329$ Sehingga: $m = 2329 \cdot \mu \; mod \; N$ $= 2329 \cdot 7673 \; mod \; 31133$ $= 17870417 \; mod \; 31133$ $= 75$
626111816	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(626111816^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(127458503) \cdot \mu \; mod \; N$ Dimana: $L\,(127458503) = \frac{127458503-1}{31133} = \frac{127458502}{31133} = 4094$ Sehingga: $m = 4094 \cdot \mu \; mod \; N$ $= 4094 \cdot 7673 \; mod \; 31133$

	$= 31413262 \; mod \; 31133$ $= 65$
155244901	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(155244901^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(793424506) \cdot \mu \; mod \; N$ Dimana: $L\,(793424506) = \frac{793424506-1}{31133} = \frac{793424505}{31133} = 25485$ Sehingga: $m = 25485 \cdot \mu \; mod \; N$ $= 25485 \cdot 7673 \; mod \; 31133$ $= 195546405 \; mod \; 31133$ $= 32$
163131862	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$ $= L(163131862^{15390} \; mod \; 31133^2) \cdot \mu \; mod \; N$ $= L(17559013) \cdot \mu \; mod \; N$ Dimana: $L\,(17559013) = \frac{17559013-1}{31133} = \frac{17559012}{31133} = 564$ Sehingga: $m = 564 \cdot \mu \; mod \; N$ $= 564 \cdot 7673 \; mod \; 31133$ $= 4327572 \; mod \; 31133$ $= 85$
783163452	$m = L\left(c^{\lambda} \; mod \; N^2\right) \cdot \mu \; mod \; N$

	$= L(783163452^{15390}\ mod\ 31133^2) \cdot \mu\ mod\ N$ $= L(83498707) \cdot \mu\ mod\ N$ Dimana: $L\ (83498707) = \frac{83498707-1}{31133} = \frac{83498706}{31133} = 2682$ Sehingga: $m = 2682 \cdot \mu\ mod\ N$ $= 2682 \cdot 7673\ mod\ 31133$ $= 20578986\ mod\ 31133$ $= 73$
523876378	$m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$ $= L(523876378^{15390}\ mod\ 31133^2) \cdot \mu\ mod\ N$ $= L(540655679)$ x $\mu\ mod\ N$ Dimana: $L\ (540655679) = \frac{540655679-1}{31133} = \frac{540655678}{31133} = 17366$ Sehingga: $m = 17366 \cdot \mu\ mod\ N$ $= 17366 \cdot 7673\ mod\ 31133$ $= 133249318\ mod\ 31133$ $= 78$
598428719	$m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$ $= L(598428719^{15390}\ mod\ 31133^2) \cdot \mu\ mod\ N$ $= L(61518809) \cdot \mu\ mod\ N$ Dimana:

	$L\ (61518809) = \frac{61518809-1}{31133} = \frac{61518808}{31133} = 1976$ Sehingga: $m = 1976 \cdot \mu\ mod\ N$ $= 1976 \cdot 7673\ mod\ 31133$ $= 15161848\ mod\ 31133$ $= 77$
781097392	$m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$ $= L(781097392^{15390}\ mod\ 31133^2) \cdot \mu\ mod\ N$ $= L(127458503) \cdot \mu\ mod\ N$ Dimana: $L\ (127458503) = \frac{127458503-1}{31133} = \frac{127458502}{31133} = 4094$ Sehingga: $m = 4094 \cdot \mu\ mod\ N$ $= 4094 \cdot 7673\ mod\ 31133$ $= 31413262\ mod\ 31133$ $= 65$

Hasil Dekripsi: 49, 56, 54, 49, 48, 48, 54, 55, 32, 77, 65, 84, 69, 77,65, 84, 73, 75, 65, 32, 85, 73, 78, 77, 65

5. Konversi numerik ke karakter

Tabel 7 Simulasi Representasi ASCII ke Teks

Kode ASCII	*Plaintext*	Kode ASCII	*Plaintext*
49	1	77	M
56	8	65	A

54	6	84	T
49	1	73	I
48	0	75	K
48	0	65	A
54	6	32	Spasi
55	7	85	U
32	Spasi	73	I
77	M	78	N
65	A	77	M
84	T	65	A
69	E		

6. Output *Plaintext*

18610067 MATEMATIKA UINMA

4.2 Simulasi dengan menggunakan program python

Pada subbab ini, dilakukan simulasi proses pembangkitan kunci, enkripsi, transposisi *ciphertext*, dan dekripsi kembali pesan teks menggunakan bahasa pemrograman Python. Tujuan dari simulasi ini adalah untuk membuktikan keabsahan implementasi praktis dari algoritma Paillier Cryptosystem yang bersifat homomorfik aditif, serta penerapan Matriks Permutasi sebagai mekanisme pengacakan posisi elemen *ciphertext* guna meningkatkan keamanan pesan.

Kode program disusun secara modular dan sistematis, berdasarkan tahapan algoritmik yang telah dijabarkan dalam kajian teori sebelumnya. Dimulai dari proses pembangkitan kunci hingga rekonstruksi pesan asli, simulasi ini bertujuan

memberikan pemahaman konseptual dan praktis terhadap implementasi algoritma kriptografi secara numerik

4.2.1. Simulasi Enkripsi dengan menggunakan program python

Sebelum dilakukan proses enkripsi, pesan teks terlebih dahulu direpresentasikan ke dalam bentuk bilangan bulat menggunakan standar ASCII. Representasi ini diperlukan agar setiap karakter dalam pesan dapat diproses oleh algoritma Paillier Cryptosystem.

Konversi dilakukan menggunakan bahasa pemrograman Python, dengan memanfaatkan fungsi ord() untuk memperoleh nilai ASCII dari setiap karakter. Berikut tabel 8 yang berisi potongan kode program dan tabel 9 yang berisi hasil keluaran pada terminal yang menunjukkan proses representasi tersebut.

Tabel 8 Kode Program Representasi ASCII

```
def text_to_ascii(text):
    return [ord(c) for c in text]
plaintext = input("Masukkan plaintext: ")
ascii_plain = text_to_ascii(plaintext)
print(f"Plaintext ASCII: {ascii_plain}")
```

Tabel 9 Output Terminal Representasi ASCII

```
Masukkan plaintext: 18610067 MATEMATIKA UINMA
Plaintext ASCII: [49, 56, 54, 49, 48, 48, 54, 55, 32, 77,
65, 84, 69, 77, 65, 84, 73, 75, 65, 32, 85, 73, 78, 77,
65]
```

Pada tabel 9 ditunjukan bahwa *plaintext* berhasil dikonversi ke bentuk bilangan ASCII. Setelah pesan dikonversi ke bentuk bilangan ASCII, dilakukan simulasi pembangkitan kunci algoritma *Paillier Cryptosystem* dengan pendekatan input manual terhadap bilangan prima p dan q. Nilai-nilai tersebut dimasukkan langsung oleh pengguna melalui program Python, sehingga memungkinkan kontrol penuh terhadap parameter awal yang digunakan dalam proses kriptografi.

Setelah menerima masukan, program akan menghitung nilai modulus $N = p \cdot q$, serta $\lambda = lcm(p-1, q-1)$ secara numerik. Selanjutnya, nilai g juga dimasukkan secara manual, kemudian diverifikasi apakah memenuhi syarat validitas untuk digunakan dalam algoritma Paillier. Nilai μ diperoleh dengan menghitung invers dari $\mu = \left(L\left(g^{\lambda} mod\ N^2\right)\right)^{-1} mod\ N$, menggunakan fungsi $L(u) = (\frac{u-1}{N})$ dengan $u = g^{\lambda}\ mod\ N^2$.

Untuk merepresentasikan proses tersebut, berikut tabel 10 yang menampilkan potongan kode program Python yang digunakan:

Tabel 10 Kode Program Simulasi Pembangkitan Kunci

```
from math import gcd
import numpy as np
from Crypto.Util import number

def L(u, n):
    return (u - 1) // n

while True:
    p = int(input("Hi Wahyu! Masukkan bilangan prima p:
"))
    if not number.isPrime(p):
        print("p bukan bilangan prima. Coba lagi.")
        continue
    q = int(input("Masukkan bilangan prima q: "))
    if not number.isPrime(q):
        print("q bukan bilangan prima. Coba lagi.")
        continue
    if p == q:
        print("p dan q tidak boleh sama. Coba lagi.")
        continue
    break

g = int(input("Masukkan nilai g: "))

n = p * q
nsquare = n * n
lam = (p - 1) * (q - 1) // gcd(p - 1, q - 1)
mu = pow(L(pow(g, lam, nsquare), n), -1, n)
```

```
print(f"p = {p}, q = {q}, N = {n}, g = {g}, lambda =
{lam}, mu = {mu}")
print(f"Kunci Publik (g,N): {g}, {n} ")
print(f"Kunci Privat (lam, mu): {lam}, {mu} ")
```

Adapun hasil keluaran (output) yang diperoleh dari terminal setelah program dijalankan dan pengguna memasukkan nilai input, ditunjukkan pada tabel 11 berikut:

Tabel 11 Output Terminal Pembangkitan Kunci

```
Hi Wahyu! Masukkan bilangan prima p: 163
Masukkan bilangan prima q: 191
Masukkan nilai g: 31134
p = 163, q = 191, N = 31133, g = 31134, lambda = 15390, mu
= 7673
Kunci Publik (g,N): 31134, 31133
Kunci Privat (lam, mu): 15390, 7673
```

Tabel 11 memperlihatkan hasil keluaran proses pembangkitan kunci pada algoritma paillier cryptosystem, yang menghasilkan sepasang kunci, yaitu kunci publik (g, N) dan kunci privat privat (λ, μ) yang akan digunakan pada tahap enkripsi dan dekripsi pesan.

Tahap berikutnya adalah proses enkripsi menggunakan algoritma Paillier Cryptosystem. Parameter kunci publik yang digunakan telah dibangkitkan pada tahap sebelumnya, dan setiap nilai ASCII dienkripsi secara individual menggunakan rumus dasar algoritma Paillier.

Implementasi dilakukan melalui Python dengan menginput parameter N, g, dan bilangan acak r untuk setiap karakter. Proses ini menghasilkan *ciphertext* dalam bentuk bilangan bulat. Berikut tabel 12 yang menampilkan kode program dan tabel 13 yang menampilkan hasil keluaran terminal yang menunjukkan *ciphertext* dari setiap karakter pesan.

Tabel 12 Kode Program Simulasi Enkripsi Pesan

```
def encrypt_manual(m, r, g, n, nsquare):
    return (pow(g, m, nsquare) * pow(r, n, nsquare)) %
nsquare

r_vals = []
candidate = 2
while len(r_vals) < len(ascii_plain):
    if gcd(candidate, n) == 1:
        r_vals.append(candidate)
    candidate += 1

ciphertext = [encrypt_manual(m, r, g, n, nsquare) for m, r
in zip(ascii_plain, r_vals)]

print("\n-- Ringkasan Proses --")
print(f"{'Index':<5} {'Char':<5} {'ASCII':<6} {'r':<3}
{'Ciphertext':<15}")
for i, (c, a, r, cip) in enumerate(zip(plaintext,
ascii_plain, r_vals, ciphertext)):
    print(f"{i:<5} {c:<5} {a:<6} {r:<3} {cip:<15}")

print("\nCiphertext keseluruhan sebelum permutasi:")
print(ciphertext)
```

Tabel 13 Output Terminal Simulasi Enkripsi Pesan

```
-- Ringkasan Proses --
Index Char  ASCII  r   Ciphertext
0     1     49     2   850723011
1     8     56     3   512828523
2     6     54     4   356366282
3     1     49     5   570833799
4     0     48     6   862752353
5     0     48     7   750883227
6     6     54     8   205497622
7     7     55     9   823557145
8           32     10  532690126
9     M     77     11  197698592
10    A     65     12  442837872
11    T     84     13  812521579
12    E     69     14  397468337
13    M     77     15  745697381
14    A     65     16  947202143
15    T     84     17  757440499
16    I     73     18  593199099
17    K     75     19  606849109
18    A     65     20  626111816
```

```
19          32      21  155244901
20    U     85      22  163131862
21    I     73      23  783163452
22    N     78      24  523876378
23    M     77      25  598428719
24    A     65      26  781097392

Ciphertext keseluruhan sebelum permutasi:
[850723011, 512828523, 356366282, 570833799, 862752353,
750883227, 205497622, 823557145, 532690126, 197698592,
442837872, 812521579, 397468337, 745697381, 947202143,
757440499, 593199099, 606849109, 626111816, 155244901,
163131862, 783163452, 523876378, 598428719, 781097392]
```

Pada tabel 13 memperlihatkan *ciphertext* yang diperoleh dari hasil enkripsi nilai ASCII dan untuk menambah lapisan keamanan, hasil *ciphertext* tersebut kemudian diacak menggunakan matriks permutasi. Matriks ini digunakan untuk menyusun ulang posisi elemen *ciphertext* berdasarkan pola permutasi tertentu yang telah ditentukan sebelumnya.

Transposisi dilakukan dengan mengimplementasikan indeks permutasi pada array *ciphertext* menggunakan Python. Tujuannya adalah untuk mengacak urutan data sehingga meskipun *ciphertext* berhasil diakses pihak tidak berwenang, susunannya tetap sulit dikenali tanpa mengetahui pola permutasinya.

Berikut tabel 14 yang menampilkan potongan kode python dan tabel 15 menampilkan hasil tampilan terminal setelah proses permutasi diterapkan pada *ciphertext*.

Tabel 14 Simulasi Penggunaan Matriks Permutasi

```python
def build_permutation_matrix(perm):
    n = len(perm)
    matrix = np.zeros((n, n), dtype=int)
    for i, p in enumerate(perm):
        matrix[p][i] = 1
    return matrix

def split_into_blocks(data, block_size, pad_value=0):
```

```
    blocks = []
    for i in range(0, len(data), block_size):
        block = data[i:i + block_size]
        if len(block) < block_size:
            block += [pad_value] * (block_size -
len(block))
        blocks.append(block)
    return blocks

def apply_permutation(lst, perm):
    return [lst[i] for i in perm]

def inverse_permutation(lst, inv_perm):
    return [lst[i] for i in inv_perm]

def generate_permutation(n):
    from random import shuffle
    perm = list(range(n))
    shuffle(perm)
    inverse = [0] * n
    for i, p in enumerate(perm):
        inverse[p] = i
    return perm, inverse

block_size = int(input("Masukkan ukuran blok (n x n): "))
cipher_blocks = split_into_blocks(ciphertext, block_size)

permuted_blocks = []
inverse_blocks = []
```

Tabel 15 Output Terminal Simulasi Penggunaan Matriks Permutasi

```
Masukkan ukuran blok (n x n): 5

Blok ke-1: [850723011, 512828523, 356366282, 570833799,
862752353]
Matriks Permutasi:
[[0 1 0 0 0]
 [0 0 1 0 0]
 [0 0 0 0 1]
 [1 0 0 0 0]
 [0 0 0 1 0]]
Setelah Permutasi: [570833799, 850723011, 512828523,
862752353, 356366282]

Blok ke-2: [750883227, 205497622, 823557145, 532690126,
197698592]
Matriks Permutasi:
```

```
[[0 1 0 0 0]
 [0 0 0 0 1]
 [0 0 1 0 0]
 [1 0 0 0 0]
 [0 0 0 1 0]]
Setelah Permutasi: [532690126, 750883227, 823557145,
197698592, 205497622]

Blok ke-3: [442837872, 812521579, 397468337, 745697381,
947202143]
Matriks Permutasi:
[[0 0 1 0 0]
 [0 0 0 0 1]
 [0 0 0 1 0]
 [1 0 0 0 0]
 [0 1 0 0 0]]
Setelah Permutasi: [745697381, 947202143, 442837872,
397468337, 812521579]

Blok ke-4: [757440499, 593199099, 606849109, 626111816,
155244901]
Matriks Permutasi:
[[0 0 0 1 0]
 [0 1 0 0 0]
 [1 0 0 0 0]
 [0 0 0 0 1]
 [0 0 1 0 0]]
Setelah Permutasi: [606849109, 593199099, 155244901,
757440499, 626111816]

Blok ke-5: [163131862, 783163452, 523876378, 598428719,
781097392]
Matriks Permutasi:
[[0 0 0 1 0]
 [0 0 0 0 1]
 [0 1 0 0 0]
 [1 0 0 0 0]
 [0 0 1 0 0]]
Setelah Permutasi: [598428719, 523876378, 781097392,
163131862, 783163452]

Ciphertext setelah permutasi (seluruh blok):
[570833799, 850723011, 512828523, 862752353, 356366282,
532690126, 750883227, 823557145, 197698592, 205497622,
745697381, 947202143, 442837872, 397468337, 812521579,
606849109, 593199099, 155244901, 757440499, 626111816,
598428719, 523876378, 781097392, 163131862, 783163452]
```

Dari hasil tabel 15, terlihat bahwa posisi ciphertext telah berhasil diacak sesuai dengan matriks permutasi.

4.2.2. Simulasi Dekripsi dengan menggunakan program python

Setelah dilakukan proses transposisi terhadap *ciphertext*, tahap selanjutnya adalah mengembalikan susunan elemen *ciphertext* ke posisi semula melalui proses invers permutasi. Langkah ini bertujuan untuk memastikan bahwa *ciphertext* yang akan didekripsi sesuai dengan urutan aslinya sebelum dilakukan permutasi.

Invers permutasi dilakukan dengan membalik indeks posisi dari permutasi yang telah digunakan sebelumnya. Implementasi dilakukan menggunakan Python dengan cara menyusun ulang elemen-elemen *ciphertext* berdasarkan indeks asli permutasinya. Pada tabel 16 berikut akan ditampilkan potongan kode program dan pada tabel 17 akan ditampilkan hasil keluaran pada terminal dari proses invers permutasi.

Tabel 16 Kode Program Penggunaan Invers Matriks

```
for idx, block in enumerate(cipher_blocks):
    perm, inv_perm = generate_permutation(len(block))
    permuted = apply_permutation(block, perm)

    print(f"\nBlok ke-{idx+1}: {block}")
    print("Matriks Permutasi:")
    print(build_permutation_matrix(perm))
    print("Setelah Permutasi:", permuted)

    recovered = inverse_permutation(permuted, inv_perm)
    print("Matriks Invers Permutasi:")
    print(build_permutation_matrix(inv_perm))
    print("Setelah Invers:", recovered)

    permuted_blocks.extend(permuted)
    inverse_blocks.extend(recovered)

print("\nCiphertext setelah permutasi (seluruh blok):")
print(permuted_blocks)
```

```
print("\nCiphertext setelah inverse permutasi (seluruh
blok):")
print(inverse_blocks)
```

Tabel 17 Output Simulasi Penggunaan Invers Matriks

```
Matriks Invers Permutasi:
[[0 0 0 1 0]
 [1 0 0 0 0]
 [0 1 0 0 0]
 [0 0 0 0 1]
 [0 0 1 0 0]]
Setelah Invers: [850723011, 512828523, 356366282,
570833799, 862752353]

Matriks Invers Permutasi:
[[0 0 0 1 0]
 [1 0 0 0 0]
 [0 0 1 0 0]
 [0 0 0 0 1]
 [0 1 0 0 0]]
Setelah Invers: [750883227, 205497622, 823557145,
532690126, 197698592]

Matriks Invers Permutasi:
[[0 0 0 1 0]
 [0 0 0 0 1]
 [1 0 0 0 0]
 [0 0 1 0 0]
 [0 1 0 0 0]]
Setelah Invers: [442837872, 812521579, 397468337,
745697381, 947202143]

Matriks Permutasi:
[[0 0 0 1 0]
 [0 1 0 0 0]
 [1 0 0 0 0]
 [0 0 0 0 1]
 [0 0 1 0 0]]
Setelah Permutasi: [606849109, 593199099, 155244901,
757440499, 626111816]

Matriks Invers Permutasi:
[[0 0 0 1 0]
 [0 0 1 0 0]
 [0 0 0 0 1]
```

```
 [1 0 0 0 0]
 [0 1 0 0 0]]
Setelah Invers: [163131862, 783163452, 523876378,
598428719, 781097392]

Ciphertext setelah inverse permutasi (seluruh blok):
[850723011, 512828523, 356366282, 570833799, 862752353,
750883227, 205497622, 823557145, 532690126, 197698592,
442837872, 812521579, 397468337, 745697381, 947202143,
757440499, 593199099, 606849109, 626111816, 155244901,
163131862, 783163452, 523876378, 598428719, 781097392]
```

Tabel 19 menampilkan proses invers permutasi berhasil mengembalikan *ciphertext* ke urutan awal. Setelah *ciphertext* berhasil dikembalikan ke urutan awal, dilakukan proses dekripsi menggunakan algoritma Paillier Cryptosystem. Dekripsi dilakukan dengan memanfaatkan kunci privat yang telah dibangkitkan sebelumnya, yaitu pasangan nilai λ dan μ, untuk setiap elemen *ciphertext*.

Setiap nilai *ciphertext* didekripsi dengan menggunakan rumus: $m = L\left(c^{\lambda}\ mod\ N^2\right) \cdot \mu\ mod\ N$, dengan fungsi $L(u) = (\frac{u-1}{N})$ dimana $u\ =\ c^{\lambda}\ mod\ N^2$. Hasil dekripsi berupa bilangan bulat akan dikonversi kembali ke karakter teks menggunakan fungsi chr() dalam Python, sehingga pesan asli dapat dipulihkan secara utuh.

Berikut tabel 18 menampilkan potongan kode program dan tabel 19 menampilkan hasil keluaran pada terminal dari proses dekripsi hingga rekonstruksi pesan asli.

Tabel 18 Kode Program Simulasi Dekripsi Pesan

```python
def decrypt_manual(c, lam, mu, n, g, nsquare):
    u = pow(c, lam, nsquare)
    l = L(u, n)
    return (l * mu) % n

def ascii_to_text(numbers):
    return ''.join([chr(n) for n in numbers])
```

```
decrypted_ascii = [decrypt_manual(c, lam, mu, n, g,
nsquare) for c in inverse_blocks]
decrypted_ascii_clean = decrypted_ascii[:len(ascii_plain)]
decrypted_text = ascii_to_text(decrypted_ascii_clean)

print("\n-- Ringkasan Dekripsi --")
print(f"{'Index':<5} {'Decrypt':<8} {'Char Out'}")
for i, val in enumerate(decrypted_ascii_clean):
    print(f"{i:<5} {val:<8} {chr(val)}")

print("\nHasil dekripsi akhir:", decrypted_text)
if decrypted_text == plaintext:
    print("[✓] Dekripsi cocok dengan input.")
else:
    print("[!] Dekripsi tidak cocok dengan input.")
```

Tabel 19 Output Terminal Simulasi Dekripsi Pesan

```
-- Ringkasan Dekripsi --
Index Decrypt  Char Out
0     49       1
1     56       8
2     54       6
3     49       1
4     48       0
5     48       0
6     54       6
7     55       7
8     32
9     77       M
10    65       A
11    84       T
12    69       E
13    77       M
14    65       A
15    84       T
16    73       I
17    75       K
18    65       A
19    32
20    85       U
21    73       I
22    78       N
23    77       M
24    65       A
```

```
Hasil dekripsi akhir: 18610067 MATEMATIKA UINMA
[✓] Dekripsi cocok dengan input.
```

Berdasarkan hasil pada tabel 19, *ciphertext* berhasil dikembalikan menjadi teks asli yang identik dengan input awal/ *plaintext*.

4.3 Kriptografi dalam Pandangan Islam

Perkembangan teknologi informasi telah mengubah cara manusia berkomunikasi dan bertukar data. Di era digital ini, informasi dapat tersebar dalam waktu singkat. Namun, risiko penyalahgunaan data pribadi dan pesan rahasia juga semakin besar. Oleh karena itu, pengamanan informasi tidak cukup hanya dilakukan secara teknis, tetapi juga harus berlandaskan nilai moral. Islam adalah agama yang tidak hanya mengatur urusan ibadah. Islam juga membimbing manusia dalam seluruh aspek kehidupan, termasuk dalam menjaga tanggung jawab dan etika dalam penggunaan ilmu pengetahuan serta teknologi.

Salah satu nilai utama dalam Islam adalah amanah. Amanah berarti kepercayaan yang harus dijaga dan tidak boleh disalahgunakan. Dalam Al-Qur'an surat An-Nisa' ayat 58, Allah SWT berfirman: "Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil…" (RI, 2011). Ayat ini menunjukkan bahwa menjaga informasi rahasia adalah bagian dari kewajiban agama. Dalam dunia digital, bentuk amanah tersebut bisa berupa pesan teks, identitas pribadi, atau data penting lainnya yang harus dijaga dari pihak yang tidak berwenang.

Rasulullah SAW juga menekankan pentingnya menjaga rahasia pribadi. Dalam sebuah hadis disebutkan: "Apabila seseorang berbicara kepadamu lalu ia berpaling, maka itu adalah amanah" (HR. Abu Dawud, no. 4868). Hadis ini

menjelaskan bahwa menjaga pembicaraan pribadi merupakan akhlak mulia. Dengan demikian, penggunaan teknologi seperti kriptografi untuk melindungi pesan bukan hanya tindakan teknis. Ia juga merupakan bentuk pelaksanaan nilai-nilai Islam.

Penelitian ini menggunakan algoritma Paillier Cryptosystem dan Matriks Permutasi untuk mengamankan pesan teks. Algoritma Paillier bersifat homomorfik aditif. Sementara itu, Matriks Permutasi berfungsi mengacak susunan data agar sulit dikenali oleh pihak yang tidak berkepentingan. Kedua metode ini mencerminkan prinsip Islam dalam menjaga kerahasiaan, melindungi kehormatan, dan menghormati hak privasi setiap individu.

Islam tidak memandang teknologi sebagai sesuatu yang bebas nilai. Setiap teknologi, termasuk kriptografi, harus digunakan dengan niat yang benar dan tujuan yang baik. Teknologi akan bernilai ibadah jika digunakan untuk melindungi hak orang lain, menghindari kerusakan, dan menciptakan kemaslahatan. Maka, kriptografi menjadi sarana untuk menjalankan amanah dan menjaga etika digital dalam kehidupan sehari-hari.

Selain itu, Islam memiliki tujuan hukum yang dikenal sebagai maqāṣid asy-syarī‘ah. Tujuan ini meliputi penjagaan terhadap agama, jiwa, akal, keturunan, dan harta. Dalam konteks digital, kriptografi membantu menjaga kehormatan (ḥifẓ al-‘irḍ), melindungi informasi dan aset pribadi (ḥifẓ al-māl), serta menjaga keselamatan pribadi dari ancaman penyalahgunaan data (Auda, 2008). Maka, penggunaan enkripsi yang tepat menjadi bagian dari tanggung jawab moral dan spiritual.

Dapat disimpulkan bahwa kriptografi bukan sekadar alat bantu dalam teknologi informasi. Ia juga menjadi wujud nyata dari pelaksanaan nilai-nilai Islam seperti amanah, keadilan, dan perlindungan. Penerapan algoritma Paillier dan Matriks Permutasi dalam penelitian ini menunjukkan bahwa ilmu pengetahuan dan agama dapat berjalan seiring. Keduanya dapat bekerja sama untuk menciptakan manfaat yang lebih luas bagi kehidupan manusia.

BAB V
KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Implementasi algoritma Paillier Cryptosystem berhasil mengubah pesan teks (*plaintext*) menjadi *ciphertext* dengan pendekatan enkripsi homomorfik sebagian. Algoritma ini menggunakan pasangan kunci publik dan privat, serta mampu mengenkripsi pesan dengan sifat probabilistik yang kuat dan mendukung operasi penjumlahan pada *ciphertext* tanpa dekripsi terlebih dahulu. Serta penggunaan matriks permutasi setelah proses enkripsi menambah tingkat keamanan pesan dengan mengacak posisi elemen *ciphertext*. Teknik ini terbukti efektif dalam menyulitkan pola struktur *ciphertext* yang dapat dianalisis oleh pihak tidak berwenang, sehingga meningkatkan kerahasiaan pesan secara keseluruhan.
2. Implementasi dalam bahasa Python memungkinkan automatisasi proses enkripsi dan dekripsi dengan efisiensi tinggi. Melalui proses segmentasi ciphertext, penerapan matriks permutasi secara blok, dan inverse permutasi, pesan teks dapat dikembalikan ke bentuk asal (plaintext) secara akurat.

Dengan demikian, kombinasi antara algoritma Paillier Cryptosystem dan Matriks Permutasi tidak hanya mampu menjawab kebutuhan akan keamanan pesan dalam dunia digital, tetapi juga merepresentasikan integrasi antara teknologi dan

nilai-nilai keislaman. Penelitian ini menunjukkan bahwa matematika, melalui kriptografi, dapat berperan penting dalam menciptakan sistem informasi yang aman, terpercaya, dan berlandaskan etika. Keberhasilan dalam dekripsi seluruh pesan menunjukkan bahwa metode ini valid, akurat, dan layak dikembangkan lebih lanjut.

5.2 Saran

Untuk pengembangan dan penyempurnaan penelitian lebih lanjut, beberapa saran dapat diajukan sebagai berikut:

1. Penelitian lanjutan disarankan untuk mengembangkan metode permutasi yang lebih kompleks seperti penggunaan kunci dinamis pada setiap blok, atau integrasi dengan metode kriptografi lain seperti RSA atau ElGamal.
2. Implementasi algoritma dalam sistem berbasis web atau mobile dapat menjadi arah pengembangan selanjutnya, agar aplikasi pengamanan pesan teks ini dapat digunakan lebih luas dan mudah diakses oleh pengguna awam.
3. Integrasi dengan aspek autentikasi pengguna perlu dipertimbangkan, sehingga selain menjaga kerahasiaan, sistem juga dapat menjamin keaslian pengirim dan penerima pesan.
4. Penelitian ini juga dapat dikembangkan pada jenis pesan selain teks, seperti citra (gambar) atau suara, sehingga cakupannya semakin luas dalam bidang keamanan multimedia.
5. Disarankan untuk membuat antarmuka grafis (GUI) agar program Python lebih ramah pengguna, khususnya bagi kalangan non-programmer yang ingin menggunakan sistem kriptografi ini secara praktis.

DAFTAR PUSTAKA

Ariyus, D. (2008). *Pengantar Ilmu Kriptografi : Teori, Analisis, dan Implementasi.* Yogyakarta: Penerbit Andi.

Auda, J. (2008). *Maqasid Al Shariah as Philosopy of Islamic Law.* London: The International Institute of Islamic Thought.

Ayu Wahyuni, S. F. (2020). Kombinasi Alhoritma Kriptografi dan Teknik Permutasi untuk Meningkatkan Keamanan Pesan Rahasia. *Jurnal Ilmiah Teknologi Informasi Terapan*, 45-52.

Burton, D. M. (2011). *Elementary Number Theory.* McGraw-Hill Education.

Dewi Suryani, R. H. (2019). Penerapan Transposisi Matriks pada Kriptografi untuk Keamanan Data Teks. *Jurnal Informatikka dan Komputer*, 33-40.

Gentry, C. (2009). *A Fully Homomorphic Encryption Scheme.* Retrieved from https://crypto.stanford.edu/craig

Jost, C., Ha Lam, A. M., & Smeets, B. (2015). Encryption Performance Improvement of the Paillier Cryptosystem. *IACR Cryptology*, 11.

Katsir, I. I. (2017). *Tafsir Ibnu Katsir Jilid 1.* Solo: Insal Kamil.

Katz. (2014). *Introduction to Modern Cryptography.* Boca Raton: FL: Chapman and Hall/CRC.

Kurnia. (2013). Optimasi Konversi String Biner Hasil Least Significant Bit Steganography.

Kurnia, D. A. (2015). *Pengantar Teori Bilangan.* Malang: Universitas Negeri Malang Press.

Moloeng, L. J. (2017). *Metodologi Penelitian Kualitatif.* Bandung: Remaja Rosdakarya.

Muhammad, M. E. (2007). *From the Treasure of Arabic Morphology.* Pakistan: Zam Zam Publisher.

Mulya, M. F., Rismawati, N., & Trisanto, D. (2020). Analisis Dan Perancangan Simulasi Algoritma Paillier Cryptosystem Pada Pesan Text Dengan Presentation Format Binary, Octal, Hexadecimal dan Base64. *Faktor Exacta*, 13(4), 208-215.

Munawar, A. (2007). *Etika Informasi dalam Islam.* Jakarta: Raja Grafindo Persada.

Munir, R. (2019). Kriptografi. Bandung: Informatika Bandung.

Munir, R. (2021). *Enkripsi Homomorfik.* Bandung: Departemen Teknik Informatika Institut Teknologi Bandung.

Online, N. ((n.d)). *Al-Hujurat ayat 12.* Retrieved from https://quran.nu.or.id/al-hujurat/12

Online, N. ((n.d)). *An-Nisa ayat 58.* Retrieved from https://quran.nu.or.id/an-nisa/58

Paar, C., & Pelzl, J. (2021). *Understanding Cryptography: A Textbook for Student and Practitioners.* Berlin: Springer.

Paillier, P. (1999). *Public key cryptosystems based on composite degree residuosity classes.*

Purba, J. A., Sinaga, D., & Purba, S. R. (2019). Implementasi Algoritma Paillier Cryptosystem Pengamanan Audio. *Jurnal Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)*, 898-902.

RI, K. A. (2011). *Al-Qur'an dan Terjemahannya.* Jakarta: Lajnah Pentashihan Mushaf Al-Qur'an.

Rivai. (2011). *Etika Bisnis Islam.* Jakarta: Bumi Aksara.

Rosen, K. (2012). *Discrete Mathematics and its Aplications.* McGraw Hill.

Schneier, B. (1996). *Applied Cryptography 2nd Edition.* New Jersey: John Wiley & Sons.

Stalling, W. (2017). *Cryptography and Network Security: Principles and Practice (7th edition).* Boston: Pearson Education.

Stallings, W. (2017). *Cryptography and Network Security : Principles and Practice.* Boston: Pearson Education.

Trappe. (2006). *Introduction to Cryptography with Coding Theory.* Pearson Prentice Hall.

Tuasikal, M. A. ((n.d)). *Ilmu Dunia, Engkau Lebih Paham.* Retrieved from https://rumaysho.com/13101-ilmu-dunia-engkau-lebih-paham.html

Wang, Q., & Li, H. (2020). One The Numeric Representation of Textual Data for Cryptographic Applications. *Journal of Information Security and Aplications*, 52.

LAMPIRAN 1

```
from math import gcd
import numpy as np
from Crypto.Util import number

#REPRESENTASI ASCII
def text_to_ascii(text):
    return [ord(c) for c in text]
plaintext = input("Masukkan plaintext: ")
ascii_plain = text_to_ascii(plaintext)
print(f"Plaintext ASCII: {ascii_plain}")

#PEMBANGKITAN KUNCI
def L(u, n):
    return (u - 1) // n

while True:
    p = int(input("Hi Wahyu! Masukkan bilangan prima p:
"))
    if not number.isPrime(p):
        print("p bukan bilangan prima. Coba lagi.")
        continue
    q = int(input("Masukkan bilangan prima q: "))
    if not number.isPrime(q):
        print("q bukan bilangan prima. Coba lagi.")
        continue
    if p == q:
        print("p dan q tidak boleh sama. Coba lagi.")
        continue
    break

g = int(input("Masukkan nilai g: "))

n = p * q
nsquare = n * n
lam = (p - 1) * (q - 1) // gcd(p - 1, q - 1)
mu = pow(L(pow(g, lam, nsquare), n), -1, n)

print(f"p = {p}, q = {q}, N = {n}, g = {g}, lambda =
{lam}, mu = {mu}")
print(f"Kunci Publik (g,N): {g}, {n} ")
print(f"Kunci Privat (lam, mu): {lam}, {mu} ")

#ENKRIPSI
def encrypt_manual(m, r, g, n, nsquare):
```

```
    return (pow(g, m, nsquare) * pow(r, n, nsquare)) %
nsquare

r_vals = []
candidate = 2
while len(r_vals) < len(ascii_plain):
    if gcd(candidate, n) == 1:
        r_vals.append(candidate)
    candidate += 1

ciphertext = [encrypt_manual(m, r, g, n, nsquare) for m, r
in zip(ascii_plain, r_vals)]

print("\n-- Ringkasan Proses --")
print(f"{'Index':<5} {'Char':<5} {'ASCII':<6} {'r':<3}
{'Ciphertext':<15}")
for i, (c, a, r, cip) in enumerate(zip(plaintext,
ascii_plain, r_vals, ciphertext)):
    print(f"{i:<5} {c:<5} {a:<6} {r:<3} {cip:<15}")

print("\nCiphertext keseluruhan sebelum permutasi:")
print(ciphertext)

#MATRIKS PERMUTASI
def build_permutation_matrix(perm):
    n = len(perm)
    matrix = np.zeros((n, n), dtype=int)
    for i, p in enumerate(perm):
        matrix[p][i] = 1
    return matrix

def split_into_blocks(data, block_size, pad_value=0):
    blocks = []
    for i in range(0, len(data), block_size):
        block = data[i:i + block_size]
        if len(block) < block_size:
            block += [pad_value] * (block_size -
len(block))
        blocks.append(block)
    return blocks

def apply_permutation(lst, perm):
    return [lst[i] for i in perm]

def inverse_permutation(lst, inv_perm):
    return [lst[i] for i in inv_perm]
```

```
def generate_permutation(n):
    from random import shuffle
    perm = list(range(n))
    shuffle(perm)
    inverse = [0] * n
    for i, p in enumerate(perm):
        inverse[p] = i
    return perm, inverse

block_size = int(input("Masukkan ukuran blok (contoh: 3):
"))
cipher_blocks = split_into_blocks(ciphertext, block_size)

permuted_blocks = []
inverse_blocks = []

#INVERS MATRIKS
for idx, block in enumerate(cipher_blocks):
    perm, inv_perm = generate_permutation(len(block))
    permuted = apply_permutation(block, perm)

    print(build_permutation_matrix(perm))
    print("Setelah Permutasi:", permuted)

    recovered = inverse_permutation(permuted, inv_perm)
    print("Matriks Invers Permutasi:")
    print(build_permutation_matrix(inv_perm))
    print("Setelah Invers:", recovered)

    permuted_blocks.extend(permuted)
    inverse_blocks.extend(recovered)

print("\nCiphertext setelah permutasi (seluruh blok):")
print(permuted_blocks)

print("\nCiphertext setelah inverse permutasi (seluruh
blok):")
print(inverse_blocks)

#DEKRIPSI
def decrypt_manual(c, lam, mu, n, g, nsquare):
    u = pow(c, lam, nsquare)
    l = L(u, n)
    return (l * mu) % n

def ascii_to_text(numbers):
    return ''.join([chr(n) for n in numbers])
```

```
decrypted_ascii = [decrypt_manual(c, lam, mu, n, g,
nsquare) for c in inverse_blocks]
decrypted_ascii_clean = decrypted_ascii[:len(ascii_plain)]
decrypted_text = ascii_to_text(decrypted_ascii_clean)

print("\n-- Ringkasan Dekripsi --")
print(f"{'Index':<5} {'Decrypt':<8} {'Char Out'}")
for i, val in enumerate(decrypted_ascii_clean):
    print(f"{i:<5} {val:<8} {chr(val)}")

print("\nHasil dekripsi akhir:", decrypted_text)
if decrypted_text == plaintext:
    print("[✓] Dekripsi cocok dengan input.")
else:
    print("[!] Dekripsi tidak cocok dengan input.")
```

LAMPIRAN 2

```
Masukkan plaintext: 18610067 MATEMATIKA UINMA
Plaintext ASCII: [49, 56, 54, 49, 48, 48, 54, 55, 32, 77,
65, 84, 69, 77, 65, 84, 73, 75, 65, 32, 85, 73, 78, 77,
65]
Hi Wahyu! Masukkan bilangan prima p: 163
Masukkan bilangan prima q: 191
Masukkan nilai g: 31134
p = 163, q = 191, N = 31133, g = 31134, lambda = 15390, mu
= 7673
Kunci Publik (g,N): 31134, 31133
Kunci Privat (lam, mu): 15390, 7673

-- Ringkasan Proses --
Index Char  ASCII  r   Ciphertext
0     1     49     2   850723011
1     8     56     3   512828523
2     6     54     4   356366282
3     1     49     5   570833799
4     0     48     6   862752353
5     0     48     7   750883227
6     6     54     8   205497622
7     7     55     9   823557145
8           32     10  532690126
9     M     77     11  197698592
10    A     65     12  442837872
11    T     84     13  812521579
12    E     69     14  397468337
13    M     77     15  745697381
14    A     65     16  947202143
15    T     84     17  757440499
16    I     73     18  593199099
17    K     75     19  606849109
18    A     65     20  626111816
19          32     21  155244901
20    U     85     22  163131862
21    I     73     23  783163452
22    N     78     24  523876378
23    M     77     25  598428719
24    A     65     26  781097392
Ciphertext keseluruhan sebelum permutasi:
[850723011, 512828523, 356366282, 570833799, 862752353,
750883227, 205497622, 823557145, 532690126, 197698592,
442837872, 812521579, 397468337, 745697381, 947202143,
757440499, 593199099, 606849109, 626111816, 155244901,
163131862, 783163452, 523876378, 598428719, 781097392]
```

```
Masukkan ukuran blok (n x n): 5

Blok ke-1: [850723011, 512828523, 356366282, 570833799,
862752353]
Matriks Permutasi:
[[0 1 0 0 0]
 [0 0 1 0 0]
 [0 0 0 0 1]
 [1 0 0 0 0]
 [0 0 0 1 0]]
Setelah Permutasi: [570833799, 850723011, 512828523,
862752353, 356366282]
Matriks Invers Permutasi:
[[0 0 0 1 0]
 [1 0 0 0 0]
 [0 1 0 0 0]
 [0 0 0 0 1]
 [0 0 1 0 0]]
Setelah Invers: [850723011, 512828523, 356366282,
570833799, 862752353]

Blok ke-2: [750883227, 205497622, 823557145, 532690126,
197698592]
Matriks Permutasi:
[[0 1 0 0 0]
 [0 0 0 0 1]
 [0 0 1 0 0]
 [1 0 0 0 0]
 [0 0 0 1 0]]
Setelah Permutasi: [532690126, 750883227, 823557145,
197698592, 205497622]
Matriks Invers Permutasi:
[[0 0 0 1 0]
 [1 0 0 0 0]
 [0 0 1 0 0]
 [0 0 0 0 1]
 [0 1 0 0 0]]
Setelah Invers: [750883227, 205497622, 823557145,
532690126, 197698592]

Blok ke-3: [442837872, 812521579, 397468337, 745697381,
947202143]
Matriks Permutasi:
[[0 0 1 0 0]
 [0 0 0 0 1]
 [0 0 0 1 0]
 [1 0 0 0 0]
```

```
 [0 1 0 0 0]]
Setelah Permutasi: [745697381, 947202143, 442837872,
397468337, 812521579]
Matriks Invers Permutasi:
[[0 0 0 1 0]
 [0 0 0 0 1]
 [1 0 0 0 0]
 [0 0 1 0 0]
 [0 1 0 0 0]]
Setelah Invers: [442837872, 812521579, 397468337,
745697381, 947202143]

Blok ke-4: [757440499, 593199099, 606849109, 626111816,
155244901]
Matriks Permutasi:
[[0 0 0 1 0]
 [0 1 0 0 0]
 [1 0 0 0 0]
 [0 0 0 0 1]
 [0 0 1 0 0]]
Setelah Permutasi: [606849109, 593199099, 155244901,
757440499, 626111816]
Matriks Invers Permutasi:
[[0 0 1 0 0]
 [0 1 0 0 0]
 [0 0 0 0 1]
 [1 0 0 0 0]
 [0 0 0 1 0]]
Setelah Invers: [757440499, 593199099, 606849109,
626111816, 155244901]

Blok ke-5: [163131862, 783163452, 523876378, 598428719,
781097392]
Matriks Permutasi:
[[0 0 0 1 0]
 [0 0 0 0 1]
 [0 1 0 0 0]
 [1 0 0 0 0]
 [0 0 1 0 0]]
Setelah Permutasi: [598428719, 523876378, 781097392,
163131862, 783163452]
Matriks Invers Permutasi:
[[0 0 0 1 0]
 [0 0 1 0 0]
 [0 0 0 0 1]
 [1 0 0 0 0]
 [0 1 0 0 0]]
```

```
Setelah Invers: [163131862, 783163452, 523876378,
598428719, 781097392]

Ciphertext setelah permutasi (seluruh blok):
[570833799, 850723011, 512828523, 862752353, 356366282,
532690126, 750883227, 823557145, 197698592, 205497622,
745697381, 947202143, 442837872, 397468337, 812521579,
606849109, 593199099, 155244901, 757440499, 626111816,
598428719, 523876378, 781097392, 163131862, 783163452]

Ciphertext setelah inverse permutasi (seluruh blok):
[850723011, 512828523, 356366282, 570833799, 862752353,
750883227, 205497622, 823557145, 532690126, 197698592,
442837872, 812521579, 397468337, 745697381, 947202143,
757440499, 593199099, 606849109, 626111816, 155244901,
163131862, 783163452, 523876378, 598428719, 781097392]
-- Ringkasan Dekripsi --
Index Decrypt  Char Out
0     49       1
1     56       8
2     54       6
3     49       1
4     48       0
5     48       0
6     54       6
7     55       7
8     32
9     77       M
10    65       A
11    84       T
12    69       E
13    77       M
14    65       A
15    84       T
16    73       I
17    75       K
18    65       A
19    32
20    85       U
21    73       I
22    78       N
23    77       M
24    65       A
Hasil dekripsi akhir: 18610067 MATEMATIKA UINMA
[✓] Dekripsi cocok dengan input.
```

RIWAYAT HIDUP

Wahyu Setyo Nugroho, lebih dikenal sebagai Tyo, dilahirkan di Lamongan pada tanggal 07 April 1999 sebagai anak pertama dari pasangan Bapak Utomo dan Ibu Enik Idayati. Pendidikan dasar penulis diperoleh dari SDN Duriwetan dan lulus pada tahun 2011, kemudian melanjutkan pendidikan di MTsN Babat dan lulus pada tahun 2014. Penulis selanjutnya menempuh Pendidikan menengah atas di SMAN 2 Lamongan hingga lulus pada tahun 2017. Selanjutnya pada tahun 2018 penulis menempuh Pendidikan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Wahyu Setyo Nugroho
NIM : 18610067
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Paillier Cryptosystem dan Matriks Permutasi pada Keamanan Pesan Teks.
Pembimbing I : Muhammad Khudzaifah, M.Si
Pembimbing II : Erna Herawati, M.Pd

No	Tanggal	Hal	Tanda Tangan
1.	26 Februari 2025	Konsultasi Topik dan Data	1.
2.	3 Maret 2025	Konsultasi Bab I, II, dan III	2.
3.	7 Maret 2025	Konsultasi Kajian Agama Bab I dan II	3.
4.	14 April 2025	Konsultasi Bab I, II, dan III	4.
5.	23 April 2025	Konsultasi Bab I, II, dan III	5.
6.	5 Mei 2025	ACC Bab I, II, dan III	6.
7.	14 Mei 2025	ACC Kajian Agama Bab I dan II	7.
8.	19 Mei 2025	ACC Seminar Proposal	8.
9.	23 Mei 2025	Konsultasi Revisi Seminar Proposal	9.
10.	23 Mei 2025	Konsultasi Bab IV dan V	10.
11.	26 Mei 2025	Konsultasi Bab IV dan V	11.
12.	26 Mei 2025	ACC Bab IV dan V	12.
13.	26 Mei 2025	Konsultasi Kajian Agama Bab IV	13.
14.	27 Mei 2025	ACC Kajian Agama Bab IV	14.
15.	27 Mei 2025	ACC Seminar Hasil	15.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

16.	10 Juni 2024	Konsultasi Revisi Seminar Hasil	16.
17.	13 Juni 2024	ACC Sidang Skripsi	17.
18.	25 Juni 2024	ACC Keseluruhan	18.

Malang, 25 Juni 2025
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005