IMPLEMENTASI KODE GOLAY MENGGUNAKAN KRIPTOSISTEM *McELIECE* DALAM MENGAMANKAN PESAN

SKRIPSI

OLEH
SOVIANA
NIM. 200601110111

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024

IMPLEMENTASI KODE GOLAY MENGGUNAKAN KRIPTOSISTEM *McELIECE* DALAM MENGAMANKAN PESAN

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
Soviana
NIM. 200601110111

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2024

IMPLEMENTASI KODE GOLAY MENGGUNAKAN KRIPTOSISTEM *McELIECE* DALAM MENGAMANKAN PESAN

SKRIPSI

Oleh
Soviana
NIM. 200601110111

Telah Diperiksa dan Disetujui untuk Diuji

Malang, 20 Juni 2024

Dosen Pembimbing I	Dosen Pembimbing II
Prof. Dr. H. Turmudi, M.Si., Ph.D.	Dr. Abdussakir, M.Pd.
NIP. 19571005 198203 1 006	NIP. 19751006 200312 1 001

Mengetahui,
Ketua Program Studi Matematika

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

IMPLEMENTASI KODE GOLAY MENGGUNAKAN KRIPTOSISTEM *McELIECE* DALAM MENGAMANKAN PESAN

SKRIPSI

Oleh
Soviana
NIM. 200601110111

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

Tanggal, 26 Juni 2024

Ketua Penguji	: Dr. Elly Susanti, M.Sc.	.....
Anggota Penguji 1	: Muhammad Khudzaifah, M.Si.	.....
Anggota Penguji 2	: Prof. Dr. H. Turmudi, M.Si., Ph.D.	.....
Anggota Penguji 3	: Dr. Abdussakir, M.Pd.	.....

Mengetahui,
Ketua Program Studi Matematika

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

PERNYATAAN KEASLIAN TULISAN

Saya bertanda tangan di bawah ini

Nama : Soviana

NIM : 200601110111

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Kode Golay Menggunakan Kriptosistem *McEliece* dalam Mengamankan Pesan

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini merupakan hasil karya sendiri, bukan pengambilan tulisan atau pemikiran orang lain yang saya akui sebagai pemikiran saya, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka di halaman terakhir. Apabila dikemudian hari terbukti skripsi ini hasil jiplakan atau tiruan, maka saya bersedia menerima sanksi yang berlaku atas perbuatan tersebut.

Malang, 26 Juni 2024

Soviana

NIM. 200601110111

MOTO

“Dan bersabarlah kamu, janji Allah adalah benar.”

-Q.S Ar Rum: 60

PERSEMBAHAN

Bismillahirrahmaanirrahim

Alhamdulillahi robbil aalamiin, segala puji bagi Allah
yang memberikan kekuatan dan pertolongan dalam melewati segala proses.
Dengan segenap hati skripsi ini dipersembahkan untuk :
Seluruh keluarga terkhusus Ayah Hidayat Condro Kusumo dan Ibu Mentik
yang memberikan kesempatan bagi peneliti untuk memilih jalan perjuangan
selama ini dan mendukung setiap langkah peneliti hingga dapat menyelesaikan
tugas akhir dengan do'a dan harapan.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur ke hadirat Allah Swt. atas segala limpahan rahmat dan karunia-Nya sehingga peneliti dapat menyelesaikan skripsi yang berjudul "Implementasi Kode Golay Menggunakan Kriptosistem *McEliece* dalam Mengamankan Pesan". Shalawat serta salam mudah-mudahan senantiasa tercurahkan kepada Nabi Muhammad saw. yang telah membawa dari jalan gelap gulita, yakni era jahiliah menuju jalan yang terang benderang, yakni *ad-dinul Islam* (agama Islam).

Skripsi ini dibuat oleh peneliti untuk memenuhi salah satu syarat memperoleh gelar sarjana matematika dari Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Peneliti mengucapkan terima kasih kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. 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 penguji dan ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Prof. Dr. H. Turmudi, M.Si., Ph.D. selaku dosen pembimbing I yang telah memberikan berbagai pengetahuan, nasihat, arahan, serta motivasi kepada peneliti.
5. Dr. Abdussakir, M.Pd., selaku dosen pembimbing II yang telah memberikan bimbingan, nasihat, ilmu, serta arahan kepada peneliti.
6. Muhammad Khudzaifah, M.Si., selaku anggota penguji I yang telah bersedia menguji dan memberikan banyak ilmu dan saran kepada peneliti.
7. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
8. Orang tua peneliti dan seluruh keluarga yang selalu senantiasa mendoakan, memberikan dukungan, semangat, serta kasih sayang sehingga peneliti dapat menyelesaikan tugas akhir.

9. Seluruh mahasiswa Program Studi Matematika angkatan 2020 yang telah memberikan bantuan dan mendukung dalam berbagai keadaan.

Semoga Allah Swt. memberikan balasan atas segala kebaikan yang diberikan kepada peneliti. Peneliti berharap agar skripsi ini bermanfaat dan dapat menambah pengetahuan.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 26 Juni 2024

Peneliti

DAFTAR ISI

DAFTAR TABEL

ABSTRAK

Soviana. 2024**. Implementasi Kode Golay Menggunakan Kriptosistem *McEliece* dalam Mengamankan Pesan.** Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. H. Turmudi M.Si., Ph.D. (II) Dr. Abdussakir, M.Pd.

Kata Kunci : Kode Golay, Algoritma *McEliece*, Pembentukan kunci, Enkripsi, Dekripsi.

Penggunaan kode Golay dalam implementasi Algoritma *McEliece* untuk meningkatkan keamanan pesan dalam komunikasi. Skema kriptosistem *McEliece* menjadi kriptanalisis yang handal sampai sekarang dan kode Golay dikembangkan untuk mampu mengoreksi tiga kesalahan (*triple error*). Tujuan penelitian ini untuk mengetahui proses implementasi pembangkitan kunci, enkripsi, dan dekripsi kode Golay dalam kriptosistem *McEliece*. Pembangkitan kunci dalam algoritma *McEliece* akan dimodifikasi dengan kode Golay hingga menghasilkan matriks generator sebagai kunci publik. Proses enkripsi pada algoritma *McEliece* dilakukan untuk memperoleh *ciphertext* atau pesan tersembunyi. Selanjutnya, dilakukan proses dekripsi pada algoritma *McEliece* dan pengoreksian *error* dengan menggunakan kode Golay. Beberapa hasil yang didapatkan dari penelitian ini, yaitu kunci publik $G' = S \cdot G \cdot P$, proses enkripsi menghasilkan *cipherteks* $(C_i' = C_i + e)$, dan proses dekripsi yang mengembalikan pesan pada bentuk semula.

ABSTRACT

Soviana. 2024. **Implementation of the Golay Code Using the McEliece Cryptosystem in Securing Messages.** Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisor: (I) Prof. Dr. H. Turmudi M.Si., Ph.D. (II) Dr. Abdussakir, M.Pd.

Keywords: Golay Code, McEliece algorithm, key establishment, encryption, decryption.

The use of Golay codes in the implementation of the McEliece Algorithm to improve the security of messages in communication. The McEliece cryptosystem scheme has been a reliable cryptanalysis until now and the Golay code was developed to be able to correct triple errors. The purpose of this research is to find out the implementation process of key generation, encryption, and decryption of Golay code in McEliece cryptosystem. The key generation in the McEliece algorithm will be modified with the Golay code to produce a generator matrix as a public key. The encryption process in the McEliece algorithm is carried out to obtain the ciphertext or hidden message. Then the decryption process is carried out in the McEliece algorithm and error correction using the Golay code. Some of the results obtained from this research are the public key $G' = S \cdot G \cdot P$, the encryption process produces ciphertext ($C_i' = C_i + e$), and the decryption process that returns the message to its original form.

مستخلص البحث

صوفيانا. ٢٠٢٤. تنفيذ كود جولاي باستخدام نظام التشفير مكإليس لتأمين الرسائل. البحث العلمي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول : (١) البروفيسور, الدكتور. تورمودي، الماجستير،الحاج المشرف الثاني (٢) الدكتور, عبد الشاكر،الماجستير.

الكلمات المفتاحية: خوارزمية مكإليس (*McEliece*)، كود جولاي (*Golay*)، تكوين المفتاح، التشفير، فك التشفير.

لتحسين أمان الرسائل في الاتصال. *McEliece* في تنفيذ خوارزمية *Golay* استخدام كود *Golay* تحليلا موثوقا به حتى الآن ، وقد تم تطوير كود *McEliece* كان مخطط نظام التشفير ليكون قادرا على تصحيح الأخطاء الثلاثية. الغرض من هذا البحث هو معرفة عملية تنفيذ يتم تعديل إنشاء .*McEliece* في نظام تشفير *Golay* شفرات توليد المفاتيح والتشفير وإزالة لإنشاء مصفوفة مولد كمفتاح عام. *Golay* باستخدام رمز *McEliece* المفاتيح في خوارزمية للحصول على نص مشفر أو رسائل *McEliece* يتم تنفيذ عملية التشفير على خوارزمية لتصحيح الأخطاء *McEliece* مخفية ، بعد ذلك ، يتم تنفيذ عملية فك التشفير على خوارزمية بعض النتائج التي تم الحصول عليها من هذه الدراسة هي المفتاح .*Golay* باستخدام كود وعملية فك $(C_i' = C_i + e)$،*نصا مشفرا* $S \cdot G \cdot P$ العام 'G، وعملية التشفير التي تولد التشفير التي تعيد الرسالة إلى شكلها الأصلي.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi pada era digital saat ini semakin pesat. Seiring dengan perkembangan teknologi informasi di Indonesia mulai diterapkan teknologi informasi untuk mengolah data, memproses, mendapatkan, menyusun, menyimpan, memanipulasi data dalam berbagai cara untuk menghasilkan informasi yang berkualitas, yaitu informasi yang relevan, akurat, dan tepat waktu yang digunakan untuk keperluan pribadi, bisnis, dan pemerintahan dan merupakan informasi yang strategis untuk mengambil keputusan. Informasi yang diolah harus dijaga kerahasiannya dan keamanannya karena informasi atau data-data yang dimiliki merupakan dokumen yang sangat berharga dan rentan terhadap tindakan kejahatan yang dilakukan oleh pihak-pihak yang tidak bertanggung jawab. Apabila data yang dimiliki dicuri atau disalahgunakan akan memberikan dampak kerugian (Manullang et al., 2017).

Informasi merupakan hal yang harus dipastikan keamanannya karena dengan kemajuan teknologi yang ada suatu informasi bisa didapatkan dengan mudah jika tingkat keamananya rendah. Informasi yang bersifat rahasia dan sensitif perlu dijaga keamanannya agar tidak dapat diakses oleh pihak lain yang tidak berhubungan atas informasi tersebut (Munir, 2019). Salah satu kejahatan yang diakibatkan oleh kurangnya tingkat keamanan informasi adalah penyadapan. Kejahatan seperti ini sering terjadi pada informasi yang didistribusikan melalui internet. Tingkat kejahatan seperti penyadapan akan semakin tinggi jika tidak

diimbangi dengan tingkat keamanan yang tinggi. Kasus penyadapan adalah pembobolan informasi mengenai data nasabah bank, pencurian dokumen negara, dan juga penyadapan surat-surat penting milik negara yang dilakukan oleh pihak-pihak yang tidak bertanggungjawab. Hal ini menjadikan peringatan bagi kita bahwa keamanan informasi pada saat ini sudah menjadi suatu kebutuhan (Munir, 2019). Oleh karena itu dibutuhkan sebuah alat untuk memperkuat tingkat keamanan terhadap informasi-informasi tersebut, salah satunya dengan memanfaatkan ilmu kriptografi.

Kriptografi merupakan salah satu cabang ilmu matematika yang dapat digunakan untuk meningkatkan keamanan informasi. Kriptografi merupakan solusi yang tepat untuk mengatasi masalah keamanan (Satir dan Kendirli, 2022). Sebuah ilmu dan seni untuk menjaga kerahasiaan sebuah pesan dengan cara mengubah pesan kedalam bentuk sandi hingga maknanya tidak dapat dipahami merupakan fungsi dari kriptografi. Teknik untuk mengubah informasi yang dabat dibaca/teks asli *(plaintext)* menjadi kode-kode tertentu disebut sebagai enkripsi *(encryption)* dan hasilnya disebut *ciphertext*. Sedangkan teknik untuk mengubah *ciphertext* menjadi *plaintext* disebut dekripsi *(decryption).* Algoritma yang digunakan untuk proses enkripsi dan dekripsi adalah algoritma kriptografi *(cryptographic algorithm)* atau sering disebut *cipher* (Munir, 2019). Algoritma kriptografi ini bekerja dengan menggunakan kunci *(key)* seperti kata, nomor maupun frase tertentu.

Berdasarkan jenis kuncinya, algoritma kriptografi dapat dibagi menjadi dua kelompok yaitu algoritma simetri *(private key algorithm)* dan algoritma asimetri *(public key algorithm).* Algoritma simetri adalah algoritma yang menggunakan

kunci enkripsi yang sama dengan kunci dekripsinya. Pada algoritma ini, pengirim dan penerima harus menyetujui suatu kunci tertentu yang dinamakan kunci rahasia *(secret key).* Contohnya adalah *Data Encryption Standard (DES), Advanced Encryption Standard (AES), Blowfish* dan lain-lain. Sedangkan, algoritma asimetri didesain sedemikian sehingga kunci yang digunakan untuk enkripsi berbeda dengan kunci untuk dekripsi. Kunci yang digunakan untuk enkripsi disebut kunci publik *(public key)* dan dapat diketahui oleh orang lain. Sedangkan, kunci untuk dekripsi dinamakan kunci rahasia atau sering disebut sebagai *private key* dan hanya diketahui oleh pemiliknya. Contohnya adalah ElGamal *Rivest-Shamir-Adleman (RSA), Elliptic Curve Cryptography (ECC)* dan lain-lain. Contoh kriptografi asimetri lainnya adalah *McEliece*, yaitu kriptosistem kunci publik yang diperkenalkan pertama kali oleh R.J. McEliece pada tahun 1978.

Algoritma enkripsi kunci public *McEliece* berdasarkan pada kode pemeriksaan kesalahan *(error-correcting codes).* Ide yang menjadi dasar bentuk enkripsi *McEliece* adalah pertama memilih sebuah kode khusus algoritma pengkodean yang efisien dan kemudian menyembunyikan kode tersebut sebagai kode linier general. Karena masalah pengkodean sebuah kode linier sembarang adalah masalah yang sukar/kompleks, dekripsi kode original/asli dapat disajikan sebagai kunci privat *(private key),* sedangkan deskripsi kode yang ditransformasikan disajikan sebagai kunci publik *(public key)* (Ilmiyah, 2019). Oleh karena itu, seseorang dapat dengan sengaja menambahkan kesalahan ke *codeword* untuk mengaburkan/mengenkripsi pesan dan mengoreksi kesalahannya pada saat mendekripsi pesan. Dengan penambahan kesalahan tersebut, kriptosistem ini tidak dapat diserang dengan menggunakan komputer kuantum.

Seiring perkembangan teknologi, berbagai algoritma kriptografi yang diciptakan sebelumnya seperti Diffie-Helman *key exchange*, enkripsi kunci publik RSA, *Algebraic Homomorphic*, Buchmann-Williams *key exchange*, dan kurva eliptik telah digagalkan oleh komputasi komputer kuantum. Komputer kuantum menurut Bernstein et al. (2009) adalah komputer yang dijalankan berdasarkan hukum mekanika kuantum dan dapat menyelesaikan perhitungan data yang besar dengan jauh lebih cepat dibanding komputer konvensional. Kriptosistem yang masih tergolong aman diantaranya adalah enkripsi kunci publik NTRU, kriptosistem *McEliece*, dan enkripsi kunci publik *Lattice-based*. Chen (2016) mengatakan bahwa algoritma kriptosistem yang banyak digunakan saat ini, yaitu RSA dan kurva eliptik telah digagalkan algoritma Shor yang dapat menyelesaikan masalah faktorisasi prima dengan mudah.

Pada tahun 1978, *McEliece* merumuskan suatu algoritma kunci asimetris dan memanfaatkan kode pengoreksi *error* yaitu kode Goppa dalam mekanismenya yang kemudian dikenal sebagai kriptosistem *McEliece*. Roering (2013) mendesain kriptosistem *McEliece* berukuran 460 Kb dengan kode Goppa. Gagasannya yaitu dengan menambahkan *error* pada *codeword* sehingga pesan menjadi tersandi dan *error* akan dikoreksi ketika penerima mendekripsi pesan. Enkripsi dan dekripsi kriptosistem *McEliece* memanfaatkan matriks generator kode. Seiring waktu, muncul berbagai kode yang dapat dimanfaatkan dalam suatu kriptosistem, di antaranya kode *Bose-Chaudhuri-Hocquenghem* (BCH), kode Reed Solomon, kode Reed Muller, kode Goppa, kode Golay, kode Hamming, dan kode Hadamard (Oktavia dan Utomo, 2023).

Skema enkripsi *McEliece* menjadi kriptanalisis yang handal sampai sekarang. *McEliece* juga dapat dicatat sebagai skema enkripsi kunci publik pertama yang menggunakan randomisasi dalam proses *encoding*. *McEliece* memiliki beberapa kelebihan dan kekurangan. Adapun kelebihannya, yaitu sistemnya elegan, mudah dipahami dan keamanannya telah teruji sejak tahun 1978 (Jochemsz, 2002), kriptosistem *McEliece* memiliki waktu eksekusi yang cepat jika dibandingkan dengan RSA (Siim, 2015), kriptosistem *McEliece* dinilai lebih efisien dan aman untuk diaplikasikan dalam komputer kuantum. Adapun kelemahannya, yaitu ukuran kunci publiknya terlalu panjang (Jochemsz, 2002).

Kode Golay merujuk pada sekelompok kode pengoreksi kesalahan yang ditemukan oleh matematikawan Amerika, Marcel J. E. Golay. Kode Golay dikembangkan untuk mendeteksi dan memperbaiki kesalahan dalam transmisi data. Kode Golay mampu mengoreksi tiga kesalahan (*triple error*). Terdapat dua kode Golay yang umumnya dikenal yaitu kode Golay *binary* 23 dan kode Golay *binary* 24. Kode Golay *binary* 23 (Golay (23, 12, 7)) merupakan kode yang memiliki panjang 23 bit dan digunakan untuk mendeteksi dan memperbaiki kesalahan pada blok data berukuran 12 bit serta jarak Hamming minimum sebesar 7. Kode ini pertama kali diusulkan oleh Golay pada tahun 1949. Kode Golay *binary* 24 (Golay (24, 12, 8)) adalah perluasan dari kode Golay *binary* 23. Kode ini memiliki panjang 24 bit dan mampu mendeteksi dan memperbaiki kesalahan pada blok data berukuran 12 bit serta jarak Hamming minimum sebesar 8. Diperkenalkan oleh Golay pada tahun 1954, kode ini termasuk dalam kelompok "kode *binary* siklik" dan telah digunakan dalam berbagai aplikasi, termasuk komunikasi nirkabel dan penyimpanan data. Kode Golay dikenal karena efisiensinya dalam mendeteksi dan

memperbaiki sejumlah besar kesalahan dengan *overhead* yang relatif kecil. Oleh karena itu, kode-kode ini sering digunakan dalam aplikasi yang ketepatan dan keandalan transmisi data sangat penting (Aini & Irawanto, 2011).

Selanjutnya, peneliti memilih untuk melakukan modifikasi kode Golay menggunakan kriptosistem *McEliece* dikarenakan kode Golay, meskipun efektif dalam mendeteksi dan memperbaiki kesalahan dalam transmisi data, memiliki kelemahan seperti *overhead* bit yang tinggi, kompleksitas implementasi yang dapat mempengaruhi kinerja, dan keterbatasan dalam menangani jenis kesalahan tertentu. Di sisi lain, kriptosistem *McEliece* menonjol dengan keunggulan keamanan tinggi terhadap serangan kriptografi, keamanan kunci publik yang tangguh, dan tahan terhadap serangan pemutusan kunci. Selain itu, kriptosistem *McEliece* lebih efisien secara komputasional, memiliki fleksibilitas dalam panjang kunci, dan tetap tahan terhadap perkembangan algoritma kuat. Proses keamanan dan penyampaian suatu informasi melalui enkripsi dan dekripsi merupakan hal yang penting, hingga terdapat pada salah satu hadits riwayat al-Bukhari dan Muslim dari Aisyah, sebagai berikut:

"*Sesungguhnya al-Haris bin Hisyam radiyallahu'anh bertanya kepada Rasulullah sallallahu'alaih wa sallam: Wahai Rasul bagaimana cara wahyu menghampirimu? Nabi bersabda, kadang-kadang seperti lonceng, dan model ini yang paling berat bagiku, lalu terinspirasi dan aku memahaminya. Kadangkala malaikat menyerupakan dirinya dengan manusia lalu berbicara dan aku memahaminya. 'Aisyah radiyallah 'anha bercerita bahwa ia pernah menyaksikan beliau ketika turun wahyu di hari yang sangat dingin tapi malah mengucur keringat dari dahinya*" (Riwayat al-Bukhari dan Muslim dari Aisyah)

Berdasarkan terjemahan hadits di atas menekankan bahwa salah satu penyampaian wahyu dapat diperoleh melalui suara gemerincing lonceng yang kuat. Cara tersebut yang paling berat bagi Rasulullah saw. apabila wahyu yang turun kepada Rasulullah saw. dengan cara tersebut, maka perlu mengumpulkan segala

kekuatan, kesadarannya untuk menerima, menghafal, dan memahaminya. Dalam hal ini, *al-wahyu* (wahyu) adalah kata masdar (infinitif). Dia menunjuk pada dua pengertian dasar, yaitu, tersembunyi dan cepat. Oleh sebab itu, dikatakan bahwa wahyu ialah informasi secara tersembunyi dan cepat yang khusus ditujukan kepada orang tertentu tanpa diketahui orang lain (Indriani, 2021).

Setelah mengetahui kriptosistem *McEliece* dan kode Golay, maka peneliti tertarik untuk melakukan penelitian mengenai implementasi kode Golay menggunakan kriptosistem *McEliece* dalam mengamankan pesan.

1.2 Rumusan Masalah

Berdasarkan latar belakang, rumusan masalah penelitian ini sebagai berikut:

1. Bagaimana proses implementasi pembangkitan kunci kode Golay dalam kriptosistem *McEliece* ?
2. Bagaimana proses enkripsi hasil implementasi kode Golay dalam kriptosistem *McEliece* ?
3. Bagaimana proses dekripsi hasil implementasi kode Golay dalam kriptosistem *McEliece*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan penelitian ini sebagai berikut:

1. Untuk mengetahui proses implementasi pembangkitan kunci kode Golay dalam kriptosistem *McEliece*.
2. Untuk mengetahui proses enkripsi hasil implementasi kode Golay dalam kriptosistem *McEliece*.

3. Untuk dekripsi hasil implementasi kode Golay dalam kriptosistem *McEliece*.

1.4 Manfaat Penelitian

Berdasarkan tujuan penelitian, skripsi ini dapat memberikan manfaat kepada siapapun, khususnya pembaca dan peneliti antara lain:

1. Bagi peneliti

   Mengetahui cara mengamankan pesan menggunakan implementasi kode Golay dalam kriptosistem *McEliece.*

2. Bagi pembaca dan peneliti selanjutnya

   a. Dapat menambah wawasan tentang ilmu kriptosistem khususnya kode Golay pada *McEliece.*

   b. Mengetahui keamanan pesan dengan menggunakan kode Golay pada *McEliece.*

   c. Sebagai referensi bagi peneliti selanjutnya dalam pembangkitan kunci kode Golay dalam kriptosistem *McEliece.*

3. Bagi institusi

   a. Sebagai media pembelajaran bagi para mahasiswa khususnya mata kuliah Kriptografi dan Teori Pengkodean.

   b. Mengimplementasikan materi khususnya mata kuliah Kriptografi dan Teori Pengkodean dalam dunia teknologi.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah kode Golay (24, 12, 8) untuk mengoreksi tiga kemungkinan kesalahan di kriptosistem *McEliece*.

1.6 Definisi Istilah

Penelitian ini menggunakan banyak istilah sehingga peneliti memberikan definisi dari istilah pada bagian awal agar pembaca lebih mudah untuk memahami, berikut adalah istilah-istilah yang peneliti gunakan:

1. *Encoding* adalah proses dalam kriptografi untuk menyandikan pesan asli *(Plaintext)* menjadi sebuah pesan yang dikodekan *(Ciphertext)*.
2. *Decoding* adalah proses dalam kriptografi untuk mengubah kembali pesan yang telah dikodekan *(Ciphertext)* menjadi pesan yang asli *(Plaintext).*
3. *Plaintext* adalah pesan asli yang akan dikirim kepada seseorang.
4. *Ciphertext* adalah pesan yang telah dikodekan dan siap untuk dikirimkan kepada seseorang.
5. Kunci/*key* adalah komponen penting yang berbentuk parameter untuk melakukan proses enkripsi atau *encoding* maupun dekripsi atau *decoding* pada pesan.
6. *Cryptosystem*/sistem kriptografi adalah keseluruhan proses dari kriptografi itu sendiri yang meliputi proses enkripsi, dekripsi, *plaintext*, *ciphertext*, dan kunci.
7. Modifikasi pembangkit kunci adalah pengubahan proses pembentukan kunci pada suatu algoritma dari bentuk semula.

BAB II

KAJIAN TEORI

2.1 Teori Bilangan

Bilangan dan teori bilangan (*number theory*) adalah satu dari kajian matematika yang tertua. Keadaan tertua ini dapat dilacak kembali berdasarkan keperluan bilangan datang lebih awal dari keperluan bentuk atau bangun (*shape*) dalam geometri. Masyarakat pada zaman kuno memerlukan kuantitas untuk berbagai keperluan. Beberapa keperluan kuantitas antara lain adalah (1) membilang, (2) menyatakan banyak atau jumlah, terhadap apa yang mereka miliki atau yang mereka peroleh, misalnya jumlah harta, banyaknya binatang hasil buruan, banyaknya buah yang diambil, banyaknya anak, (3) membandingkan banyak atau jumlah, dan (4) melakukan tukar-menukar barang (misalnya "barter" hasil bumi akibat belum ada "mata uang" sebagai alat untuk "jual-beli"). Pada saat zaman kuno ini, keperluan utama bilangan adalah untuk membilang. Kajian tentang sifat-sifat bilangan diduga belum mereka lakukan. Secara sistematis, sesuai dengan kaidah penyelidikan, mereka dipandang belum mengembangkan operasi bilangan beserta sifat-sifatnya, serta belum mencari pola bilangan dan nama-nama tertentu dari bilangan-bilangan khusus (misalnya bilangan perfek, bilangan prima, bilangan abundan, bilangan defisien, barisan bilangan). Bilangan itu sendiri dikenal, dipahami, dan digunakan tanpa perlu mengetahui penjelasannya, artinya tidak ada definisi tentang bilangan karena bilangan dipandang sebagai *undefined term*, yaitu istilah yang tidak didefinisikan. Bilangan merupakan kreasi budaya manusia yang

tumbuh dan berkembang selama ribuan tahun di berbagai belahan dunia (Muhsetyo, 2014).

Dalam pengertian yang sederhana, teori bilangan berkaitan dengan kajian bilangan bulat dan sifat-sifatnya. Ini berarti bahwa dalam pembahasan teori bilangan tidak dijumpai adanya pembahasan tentang bilangan pecahan dan bilangan desimal. Demikian pula, teori bilangan tidak membahas tentang bilangan irasional, serta bilangan-bilangan yang terkait dengan algoritma dan perbandingan trigonometri. Kajian bilangan bulat ini antara lain terkait dengan faktor atau pembagi, faktor persekutuan, faktor persekutuan terbesar, kelipatan, kelipatan persekutuan, kelipatan persekutuan terkecil, keprimaan, persamaan *Diophatine* (persamaan yang memerlukan penyelesaian berupa bilangan bulat, kekongruenan, dan model pengkodean. Bilangan-bilangan yang khas, misalnya bilangan prima, bilangan Mersenne, bilangan abundan, bilangan defisien, dan bilangan bersekawan merupakan bagian pembahasan yang melengkapi keseluruhan pembahasan dalam teori bilangan. Fungsi-fungsi khas, misalnya fungsi banyak pembagi, fungsi jumlah pembagi, dan fungsi Euler (Muhsetyo, 2014).

2.1.1 Keterbagian

Pembagian bilangan bulat merupakan bahan pelajaran matematika yang sudah diberikan diberikan di sekolah dasar. Bahan pelajaran ini diperluas penggunaannya sampai pada pemfaktoran prima, faktor persekutuan terbesar, kelipatan persekutuan terkecil, dan keterbagian oleh bilangan tertentu, misalnya keterbagian oleh 2, 3, dan 9.

Keterbagian (*divisibility*) merupakan dasar pengembangan teori bilangan,

sehingga konsep-konsep ketebagian akan banyak digunakan dalam sebagain besar uraian atau penjelasan matematis tentang pembuktian teorema. Jika suatu bilangan bulat dibagi oleh suatu bilangan bulat yang lain, maka hasil baginya adalah suatu bilangan bulat atau suatu bilangan yang tidak bulat, misalnya jika 40 dibagi 8, maka hasil baginya adalah bilangan bulat 8; tetapi jika 40 dibagi 16, maka hasilnya 2,5. Keadaan inilah yang memberikan gagasan tentang perlunya definisi keterbagian (Muhsetyo, 2014).

Definisi 2.1

Suatu bilangan bulat q habis dibagi oleh suatu bilangan bulat $p \neq 0$ jika ada suatu bilangan bulat x sehingga $q = px$. Dengan notasi, sebagai berikut:

1. $p|q$ dibaca p membagi q, p faktor dari q, q habis dibagi p, atau q kelipatan dari p
2. $p \nmid q$ dibaca p tidak membagi q, p bukan faktor dari q, q tidak habis dibagi p, atau q bukan kelipatan dari p

Contoh 2.1

1. $6|18$ sebab ada bilangan bulat 3 sehingga $18 = 6 \cdot 3$
2. $12 \nmid 15$ sebab tidak ada bilangan bulat x sehingga $15 = 12 \cdot x$
3. $5|-30$ sebab ada bilangan bulat -6 sehingga $-30 = 5 \cdot (-6)$
4. $-4|20$ sebab ada bilangan bulat 5 sehingga $20 = (-4) \cdot 5$

Berdasarkan Definisi 2.1 di atas jelas bahwa faktor-faktor suatu bilangan bisa merupakan bilangan bulat positif atau merupakan bilangan bulat negatif. Dengan demikian, faktor-faktor dari:

1. 6, adalah $1, -1, 2, -2, 3, -3, 6,$ dan -6
2. 15, adalah $1, -1, 3, -3, 5, -5, 15$ dan -15

Beberapa sifat sederhana keterbagian adalah :

1. $1|p$ untuk setiap $p \in Z$
2. $p|0$ untuk setiap $p \in Z$ dan $p \neq 0$
3. $p|p$ untuk setiap $p \in Z$ dan $p \neq 0$
4. Jika $p|q$, maka kemungkinan hubungan antara p dan q adalah $p < q$, $p = q$, atau $p > q$ (misalnya $3|6,\ 3|3,$ atau $3|-3$)

Teorema 2.1

Jika $p,\ q\ \in Z$ dan $p|q$, maka $p|qr$ untuk semua $r \in Z$

Bukti:

Diketahui bahwa $p|q$, maka menurut Definisi 2.1 ada suatu $x \in Z$ sehingga $q - px$ berarti $qr = pxr$, atau $qr = p(x.r)$ dengan $xr \in Z$ (sebab $x \in Z$ dan $r \in Z$) sesuai dengan Definisi 2.1 karena $qr = p(xr)$ maka $p|qr$

Teorema 2.2

Jika $p,\ q,\ r \in Z,\ p|q$ dan $q|r$, maka $p|r$

Bukti:

Diketahui $p|q$ dan $q|r$, maka menurut Definisi 2.1, tentu ada $x, y, \in Z$ sehingga $q = px$ dan $r = qy, r = qy$ dan $q = px,$ maka $r = (px)y$ atau $r = p(xy)$ dengan $x, y \in Z$. Sesuai dengan Definisi 2.1 karena $r = p(xy),$ maka $p|r$.

2.1.2 Bilangan Prima

Bilangan prima adalah bilangan asli yang hanya mempunyai tepat 2 pembagi positif, yaitu satu dan bilangan itu sendiri. Contoh bilangan prima yaitu 2, 3, 5, 7, 11, dan 13. Bilangan asli yang mempunyai lebih dari 2 pembagi positif

disebut bilangan komposit. Contoh bilangan komposit yaitu 4, 6, 8, 9, dan 10. (Sukirman, 2006).

Definisi 2.2

Untuk setiap bilangan komposit *n*, maka terdapat bilangan prima p sehingga $p|n$ dan $p \leq \sqrt{n}$. Jadi, jika tidak ada bilangan prima p yang dapat membagi *n* dengan $p \leq \sqrt{n}$, maka *n* adalah bilangan prima.

Contoh 2.2

Tentukan apakah bilangan-bilangan berikut merupakan bilangan prima atau bilangan komposit

1. 157
2. 221

Penyelesaian:

1. Bilangan-bilangan prima yang kurang dari atau sama dengan $\sqrt{157}$ adalah 2, 3, 5, 7, 11. Karena tidak ada di antara bilangan-bilangan tersebut yang dapat membagi 157, maka 157 merupakan bilangan prima.
2. Bilangan-bilangan prima yang kurang dari atau sama dengan $\sqrt{221}$ adalah 2, 3, 5, 7, 11, 13. Karena $13|221$, maka 221 adalah bilangan komposit.

2.1.3 Faktor Persekutuan Terbesar (FPB)

Definisi 2.3

Misalkan a dan b adalah bilangan bulat yang keduanya tidak nol. Faktor persekutuan terbesar dari a dan b adalah bilangan bulat positif terbesar d sehingga $d|a$ dan $d|b$. Faktor persekutuan terbesar dari a dan b dinotasikan dengan (a, b) (Lee, 2010).

Contoh 2.3

1. FPB dari 12 dan 18

   Faktor positif dari 12: 1, 2, 3, 4, 6, 12

   Faktor positif dari 18: 1, 2, 3, 6, 9, 18

   Faktor persekutuan adalah 1, 2, 3, dan 6. Jadi, $(12, 18) = 6$.

2. FPB dari 24 dan 36

   Faktor positif dari 24: 1, 2, 3, 4, 6, 8, 12, 24

   Faktor positif dari 36: 1, 2, 3, 4, 6, 9, 12, 18, 36

   Faktor persekutuan adalah 1, 2, 3, 4, 6, dan 12. Jadi, $(24, 36) = 1$.

2.1.4 Relatif Prima

Definisi 2.4

Dua bilangan $a, b \in Z$ dikatakan relatif prima jika $(a,\ b) = 1$ (Basri, 2021).

Contoh 2.4

1. Bilangan 5 dan 6 adalah relatif prima karena (5, 6) = 1
2. Bilangan 8 dan 12 bukan relatif prima karena (8, 12) ≠ 1

2.1.5 Kongruensi

Definisi 2.5

Ditentukan $p,\ q,\ m$ adalah bilangan-bilangan bulat dan $m \neq 0,\ p$ disebut kongruen dengan q modulo m, ditulis $p \equiv q(mod\ m)$, jika dan hanya jika $m|(p - q)$. Jika $m \nmid (p - q)$ maka ditulis $p \not\equiv q\ (mod\ m)$, dibaca p tidak kongruen q modulo m (Muhsetyo, 2014).

Contoh 2.5

1. $10 \equiv 6\ (mod\ 2)$ sebab $2|(10-6)$ atau $2|4$
2. $13 \equiv$ -5 $(mod\ 9)$ sebab $9|(13-(-5))$ atau $9|18$
3. $107 \equiv 2\ (mod\ 15)$ sebab $7|(107-2)$ atau $15|104$

Teorema 2.3

Jika $p,\ q,\ r,$ dan m adalah bilangan bulat dam $m > 0$ sedemikian hingga $p \equiv q(mod\ m)$, maka:

1. $p + r \equiv q + r\ (mod\ m)$
2. $p - r \equiv q - r\ (mod\ m)$
3. $pr \equiv qr\ (mod\ m)$

Bukti:

1. Diketahui $p \equiv q\ (mod\ m)$, maka $m|(p-q)$. Selanjutnya, dapat ditentukan bahwa $p-q=(p+r)-(q+r)$, berarti $m|(p-q)$ berakibat $m|(p+q)-(q+r)$. Dengan demikian $p + r \equiv q + r\ (mod\ m)$.
2. Ingat bahwa $p - r \equiv q - r\ (mod\ m)$.
3. Diketahui $p \equiv q\ (mod\ m)$, maka $m|(p-q)$, dan menurut teorema keterbagian, $m|r(p-q)$ untuk sebarang bilangan bulat r, dengan demikian $m|(pr-qr)$. Jadi, $pr \equiv qr\ (mod\ m)$.

2.1.6 Fungsi Euler

Teorema 2.4

Jika $a, m \in Z$ dan $m > 0$ sehingga $(a, m) = 1$, maka:

$$a^{\emptyset(m)} \equiv\ 1\ (mod\ m)$$

Bukti:

Misalkan bahwa $\{x_1,\ x_2, \dots ,\ x_{\emptyset(m)}\}$ adalah suatu sistem residu tereduksi modulo m dengan unsur-unsur bilangan bulat positif kurang dari m dan a relatif prima dengan m, yakni $(a,\ m) = 1$, maka $\{ax_1,\ ax_2,\ \dots ,\ ax_{\emptyset(m)}\}$ juga merupakan suatu sistem residu tereduksi modulo m. Dengan demikian, residu-residu positif terkecil dari $ax_1,\ ax_2 , \dots ,\ ax_{\emptyset(m)}$ adalah bilangan-bilangan bulat yang terdapat pada $x_1,\ x_2,\ \dots ,\ x_{\emptyset(m)}$ dengan urutan tertentu. Akibatnya, dapat mengalikan semua suku dari masing-masing sistem residu tereduksi, sehingga diperoleh:

$$ax_1, ax_2 , \dots , ax_{\emptyset(m)} \equiv x_1, x_2, \dots , x_{\emptyset(m)}\ (mod\ m)$$

dengan demikian dapat ditentukan bahwa:

$$a^{\emptyset(m)} x_1 \cdot x_2 \dots x_{\emptyset(m)} \equiv\ x_1 \cdot x_2 \dots\ x_{\emptyset(m)}\ (mod\ m)$$

$\{x_1,\ x_2, \dots ,\ x_{\emptyset(m)}\}$ adalah suatu sistem tereduksi modulo m, maka berlaku $(x_i,\ m) = 1$, yaitu $(x_1, m) = (x_2, m) = \cdots x_{\emptyset(m)},\ m) = 1$.

Maka, ditentukan bahwa $\left(x_1 \cdot\ x_2\ \dots\ x_{\emptyset(m)}, m\right) = 1$.

Dari dua keadaan, berikut:

$$a^{\emptyset(m)} x_1 \cdot x_2 \dots x_{\emptyset(m)} \equiv\ x_1 \cdot x_2 \dots\ x_{\emptyset(m)}\ (mod\ m) \text{ dan}$$

$$\left(x_1 \cdot\ x_2\ \dots\ x_{\emptyset(m)}, m\right) = 1$$

dapat ditentukan bahwa:

$$a^{\emptyset(m)} = a \cdot a^{\emptyset(m)-1} \equiv\ 1\ (mod\ m)$$

Jadi, $a^{\emptyset(m)-1}$ adalah invers dari a modulo m (Muhsetyo, 2014).

2.2 Lapangan

Definisi 2.6

Lapangan adalah himpunan tak kosong *F* dengan dua operasi '+' (disebut penjumlahan) dan '.' (disebut perkalian) yang memenuhi aksioma berikut. Untuk semua $a, b, c \in F$.

1. $a + b \in F$ dan $a \cdot b \in F$ berada di *F* (*F* tertutup untuk $(+)$ dan $(\cdot)$)
2. Komutatif : $a + b = b + a$

   $a \cdot b = b \cdot a$
3. Asosiatif : $(a + b) + c = a + (b + c)$

   $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
4. Distributif : $a \cdot (b + c) = a \cdot b + a \cdot c$

   $(a \cdot b) + c = a \cdot c + b \cdot c$

   Selanjutnya, dua elemen identitas berbeda 0 dan 1 harus ada di F, maka:
5. $a + 0 = 0 + a = a$ untuk semua $a \in F$
6. $a \cdot 1 = 1 \cdot a = a$ dan $a \cdot 0 = 0 \cdot a = 0$ untuk semua $a \in F$
7. Untuk setiap *a* di *F*, terdapat elemen invers penjumlahan (-*a*) di *F* sehingga $(-a) + a = a + (-a) = 0$
8. Untuk setiap $a \neq 0$ di F, terdapat elemen invers perkalian a^{-1} di *F* sehingga $a \cdot a^{-1} = a^{-1} \cdot a = 1$ (Ling & Xing, 2004).

2.2.1 *Finite Field* Bilangan Prima (GF(p))

Finite field atau dikenal juga dengan *Galois field* adalah *field* yang banyak anggotanya berhingga. *Finite field* dipakai secara luas di kriptografi, misalnya sistem sandi simetri AES (*Advanced Encryption Standart*) (Sadikin, 2012). *Finite*

field dengan struktur sederhana adalah *finite field* yang nilai ordernya adalah bilangan prima dinotasikan dengan GF(*p*). GF(*p*) terdiri dari himpunan bilangan $\mathbb{Z}_p$ dengan *p* bilangan prima yaitu himpunan integer {0, 1, …, *p*-1} dan 2 operasi aritmetika (penjumlahan dan perkalian) modular *p* (Sadikin, 2012).

Contoh 2.6

Buatlah tabel penjumlahan dan perkalian untuk GF(5).

Penyelesaian:

GF(5) memiliki tabel penjumlahan dan perkalian seperti Tabel 2.1 dan Tabel 2.2. Pasangan invers penjumlahan dan perkalian diberikan oleh Tabel 2.3. GF(5) adalah *field*.

Tabel 2.1 Operasi Penjumlahan pada GF(5)

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Tabel 2.2 Operasi Perkalian pada GF(5)

×	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Tabel 2.3 Pasangan Invers Penjumlahan dan Perkalian pada GF(5)

Invers Penjumlahan				
0	1	2	3	4
0	4	3	2	1
Invers Perkalian				
0	1	2	3	4
-	1	3	2	4

2.2.2 Ruang Vektor atas Lapangan Hingga

Definisi 2.7

Misalkan $\boldsymbol{F}_q$ lapangan berhingga dengan orde q. Himpunan tak kosong V, bersama dengan operasi penjumlahan (vektor) + dan perkalian skalar dengan elemen-elemen dari $\boldsymbol{F}_q$ adalah ruang vektor (atau ruang linier) atas $\boldsymbol{F}_q$ jika memenuhi semua syarat berikut. Untuk semua $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in V$ dan untuk semua λ, $\mu \in \boldsymbol{F}_q$ (Ling dan Xing, 2004).

1. $\boldsymbol{u} + \boldsymbol{v} = \boldsymbol{v} + \boldsymbol{u} \in V$
2. $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$
3. Terdapat $\mathbf{0} \in V$ dengan sifat $\mathbf{0} + \boldsymbol{v} = \boldsymbol{v} + \mathbf{0} = \mathbf{0}$ untuk semua $\boldsymbol{v} \in V$
4. Untuk setiap $\boldsymbol{u} \in V$ terdapat elemen dari V disebut $-\boldsymbol{u}$, sehingga

   $\boldsymbol{u} + (-\boldsymbol{u}) = (-\boldsymbol{u}) + \boldsymbol{u} = \mathbf{0}$
5. $\boldsymbol{u} + \boldsymbol{v} = \boldsymbol{v} + \boldsymbol{u}$
6. $\lambda \boldsymbol{v} \in V$
7. $\lambda(\boldsymbol{u} + \boldsymbol{v}) = \lambda \boldsymbol{u} + \lambda \boldsymbol{v}$

   $(\lambda + \mu)\boldsymbol{u} = \lambda \boldsymbol{u} + \mu \boldsymbol{u}$

8. $\lambda(\mu)\boldsymbol{u} = \lambda(\mu\boldsymbol{u})$
9. Jika 1 adalah identitas perkalian dari $\boldsymbol{F}_q$, maka $1\boldsymbol{u} = \boldsymbol{u}$.

2.3 Kode Linier

Definisi 2.8

Kode linier adalah kode koreksi kesalahan yang digunakan untuk mendeteksi dan mengoreksi kesalahan pada *codeword.* Kode linier dengan panjang n dan berdimensi k atas F adalah subruang berdimensi k dari F^n maka disebut (n, k)-*code*. Jika jarak minimum dari kode adalah d, maka disebut (n, k, d)-*code* (Trappe, 2006). Kode Golay $(24, 12, 8)$ adalah kode linier yang dapat mendeteksi hingga tiga kesalahan *codeword* dengan menghitung *syndrome* pada proses dekripsi.

2.4 Kode Biner

Definisi 2.9

Diberikan kode alfabet $F_2 = \{0, 1\}$. Kode atas F_2 dinamakan kode biner. Simbol kode yang digunakan untuk suatu kode biner adalah 0 dan 1 (Ling dan Xing, 2004).

Contoh 2.7

1. $C_1 = \{00,\ 01, 10, 11\}$ adalah kode-(2, 4)
2. $C_2 = \{000,\ 011, 101, 110\}$ adalah kode-(3, 4)
3. $C_3 = \{0011, 0101, 1010, 1100, 1001, 0110\}$ adalah kode-(4, 6)

2.5 Matriks Generator dan Matriks Paritas

1. Matriks Generator (G): Matriks ini digunakan untuk menghasilkan kata-kata kode dari kode linier. Dalam kode blok linier, setiap kata kode dapat dihasilkan dengan mengalikan vektor dengan matriks generator. Matriks generator kode Golay dengan bentuk $(I_{12}|A)$.
2. Matriks Paritas (H): Matriks ini digunakan untuk melakukan pemeriksaan kesalahan pada kata-kata kode. Jika matriks generator G berbentuk standar, maka matriks pemeriksa paritas untuk C adalah matriks transpose. Matriks paritas kode Golay dengan bentuk $(I_{12}|A)^T$ (Ling dan Xing, 2004).

Contoh 2.8

Matriks $G = \begin{pmatrix} 1&0&1&0&1 \\ 0&1&0&1&1 \\ 0&0&1&1&1 \end{pmatrix}$ adalah matriks generator untuk kode linier C dengan $A = F_2$ dan matriks $H = \begin{pmatrix} 1&0&0 \\ 0&1&0 \\ 1&0&1 \\ 0&1&1 \\ 1&1&1 \end{pmatrix}$ adalah matriks *parity-check* untuk kode linier C.

2.6 Kode ASCII

ASCII (*American Standard Code for Information Interchange*) 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).

2.7 Kode Golay

Kode Golay pertama kali ditemukan oleh Golay di tahun 1949. Terdapat dua kode Golay yang umumnya dikenal yaitu kode Golay binary 23 dan kode Golay binary 24. Dua kode biner Golay $\mathcal{G}_{23}$ dan $\mathcal{G}_{24}$ merupakan kode yang digunakan oleh Voyager I dan Voyager II wahana antariksa selama 1979-1981 untuk memberikan koreksi kesalahan pada transmisi kembali ke Bumi gambar berwarna Yupiter dan Saturnus. Bentuk dari matriks pembangkit kunci kode Golay yang berukuran 12×24, yaitu sebagai berikut:

$$G = (I_{12}|A)$$

$$G = \begin{pmatrix} 100000000000111011100010 \\ 010000000000101101110001 \\ 001000000000110110111000 \\ 000100000000101011011100 \\ 000010000000100101101110 \\ 000001000000100010110111 \\ 000000100000110001011011 \\ 000000010000111000101101 \\ 000000001000111100010110 \\ 000000000100101110001011 \\ 000000000010110111000101 \\ 000000000001011111111111 \end{pmatrix}$$

dengan I_{12} adalah matriks identitas berukuran 12×12 dan A adalah matriks paritas berukuran 12×12. Matriks identitas digunakan agar bit-bit asli pesan tetap utuh dalam kode yang dihasilkan, sedangkan matriks paritas ditambahkan untuk memungkinkan deteksi dan koreksi kesalahan, serta sindrom R dengan vektor kolom s berdimensi 12 yang didefinisikan oleh $s = HR^T$ atau $s = RH^T$ (Ling & Xing, 2004). Kombinasi ini memberikan kemampuan koreksi kesalahan yang kuat pada kode Golay.

2.8 Kriptografi

Kriptografi merupakan salah satu cabang ilmu matematika yang dapat digunakan untuk meningkatkan keamanan informasi. Kriptografi merupakan solusi yang tepat untuk mengatasi masalah keamanan (Satir & Kendirli, 2022). Sebuah ilmu dan seni untuk menjaga kerahasiaan sebuah pesan dengan cara mengubah pesan ke dalam bentuk sandi hingga maknanya tidak dapat dipahami merupakan fungsi dari kriptografi (Munir, 2019). Teknik untuk mengubah informasi yang dabat dibaca/teks asli *(plaintext)* menjadi kode–kode tertentu disebut sebagai enkripsi *(encryption)* dan hasilnya disebut *ciphertext*. Sedangkan teknik untuk mengubah *ciphertext* menjadi *plaintext* disebut dekripsi *(decryption).* Algoritma yang digunakan unruk proses enkripsi dan dekripsi adalah algoritma kriptografi *(cryptographic algorithm)* atau sering disebut *cipher*. Algoritma kriptografi ini bekerja dengan menggunakan kunci *(key)* seperti kata, nomor maupun frase tertentu (Munir, 2019).

Kriptografi merupakan gabungan dari kata *cryptos* dan *graphein* yang berasal dari bahasa Yunani yang memiliki arti rahasia dan tulisan. Kriptografi secara harfiah berarti tulisan rahasia (Munir, 2019). Kriptografi merupakan ilmu yang mempelajari metode untuk menyamarkan pesan sehingga pesan hanya dapat dibaca oleh penerima (Jamaludin et al., 2022). Menurut (Menezes et al., 1996), kriptografi merupakan disiplin ilmu yang di dalamnya mempelajari teknik-teknik matematika yang memiliki hubungan dengan keamanan suatu informasi.

Kriptografi menjadi syarat penting dalam keamanan teknologi informasi, terlebih dalam pengiriman pesan rahasia. Proses pengiriman pesan rahasia rentan mengalami penyerangan seperti penyadapan, pemutusan komunikasi, pengubahan

isi pesan dan lain sebagainya. Keamanan dalam pengiriman pesan dapat meningkat dengan adanya kriptografi, pengamanan pesan dilakukan dengan mengubah pesan dalam bentuk sandi dengan menggunakan sebuah algoritma dan kunci tertentu yang hanya diketahui oleh pihak-pihak yang berwenang.

Kriptografi membentuk sistem yang disebut sebagai sistem kriptografi (*cryptosystem*). Sistem kriptografi adalah himpunan yang terdiri dari lima bagian sebagai berikut:

1. *Plaintext*

   Plaintext atau teks asli adalah pesan asli yang dapat terbaca. *Plaintext* merupakan *input* untuk algoritma enkripsi.

2. *Secret Key*

   Secret key adalah kunci rahasia yang merupakan *input* bagi algoritma enkripsi pula. *Secret key* merupakan nilai yang bebas terhadap teks asli (*plaintext*) dan menentukan hasil *output* untuk algoritma enkripsi.

3. *Ciphertext*

   Ciphertext merupakan *output* dari algoritma enkripsi. *Ciphertext* juga diartikan sebagai pesan yang tersembunyi karena sudah melalui proses enkripsi. *Ciphertext* yang acak dan sulit dipahami dapat dihasilkan dengan menggunakan algoritma enkripsi yang baik.

4. Algoritma *Encoding*

   Terdapat dua *input* yang dibutuhkan dalam algoritma *encoding*, yaitu *plaintext* dan *secret key*. Pada proses algoritma enkripsi, *plaintext* ditransformasi sehingga menghasilkan *ciphertext*.

5. Algoritma *Decoding*

Terdapat dua *input* yang dibutuhkan dalam algoritma *decoding*, yaitu *ciphertext* dan *secret key*. Algoritma *decoding* mengembalikan *ciphertext* menjadi *plaintext* jika *secret key* yang digunakan pada algoritma *decoding* dan algoritma enkripsi sama (Sadikin, 2012).

2.9 Kriptosistem *McEliece*

Algoritma enkripsi kunci publik *McEliece* berdasarkan pada kode pemeriksaan kesalahan (*error-correcting codes*). Ide yang menjadi dasar bentuk enkripsi *McEliece* adalah pertama memilih sebuah kode khusus algoritma pengkodean yang efisien dan kemudian menyembunyikan kode tersebut sebagai kode linier general. Karena masalah pengkodean sebuah kode linier sembarang adalah masalah yang sukar/kompleks, dekripsi kode original/asli dapat disajikan sebagai kunci privat (*private key*), sedangkan dekripsi kode yang ditransformasikan disajikan sebagai kunci publik *(public key*) (Ilmiyah, 2019).

Kriptosistem *McEliece* menjadi kriptanalisis yang handal sampai sekarang. Skema enkripsi *McEliece* juga dapat dicatat sebagai skema enkripsi kunci publik pertama yang menggunakan randomisasi dalam proses enkripsi. Di samping itu, sistem *McEliece* juga lebih cepat dari kriptosistem kunci publik lainnya. Meskipun sangat efisien, skema enkripsi kunci publik *McEliece* hanya menerima sedikit perhatian dalam praktik, hal ini disebabkan beberapa permasalahan, di antaranya adalah karena kunci publiknya sangat besar mempunyai panjang 2^{19} bit, perluasan data juga besar dan *ciphertext* yang dihasilkan dua kali lebih panjang daripada

plaintext (Anggoro et al., 2020). Proses dari sistem kriptografi *McEliece* dapat disimulasikan sebagai berikut:

1. Proses Pembentukan Kunci
    a. Alice memilih generator matriks dari kode linier biner yang dapat memperbaiki *error t*. Kode yang dipilih dapat berfungsi sebagai algoritma *decoding* yang efisien.
    b. Alice memilih secara acak matriks biner S yang memiliki invers berukuran $k \times k$.
    c. Alice memilih secara acara matriks permutasi P berukuran $n \times n$.
    d. Alice menghitung matriks G' dengan ukuran $k \times n$, yaitu:

$$G' = S \cdot G \cdot P$$

    e. Kunci publik Alice adalah (G', t) dan kunci privatnya adalah (S, G, P).
2. Proses Enkripsi
    a. Misalkan Bob ingin mengirim pesan ke Alice dengan kunci publik. Maka, Bob melakukan *encoding* terhadap pesan m menjadi deretan biner dengan panjang k.
    b. Bob menghitung vektor $c' = m \cdot G'$.
    c. Bob membangkitkan vektor e sebesar n-bit secara acak yang mengandung t angka 1, agar hasil enkripsi semakin acak.
    d. Bob menghitung pesan rahasia $c = c' + e$.

    Contoh enkripsi, sebagai berikut:

    a. Misalkan nilai $k = 4$ dan $n = 7$.
    b. Pilih sembarang matriks A berukuran $k \times (n - k)$ atau 4×3.

$$A = \begin{bmatrix} 111 \\ 101 \\ 000 \\ 001 \end{bmatrix}$$

c. Pilih matriks generator berukuran $k \times n$ dan $G = [I_k|A]$.

$$G = \begin{bmatrix} 1000111 \\ 0100101 \\ 0010000 \\ 0001001 \end{bmatrix}$$

d. Pilih matriks non-singular S berukuran $k \times k$ atau 4×4.

$$S = \begin{bmatrix} 0011 \\ 0110 \\ 0001 \\ 1111 \end{bmatrix}$$

e. Pilih matriks permutasi P berukuran $n \times n$ atau 7×7.

$$P = \begin{bmatrix} 0000100 \\ 0000010 \\ 1000000 \\ 0001000 \\ 0000001 \\ 0100000 \\ 0010000 \end{bmatrix}$$

f. Simpan matriks S, G, dan P sebagai *private key*.

g. Hitung $G' = S \cdot G \cdot P$, yang merupakan *public key*.

$$G' = S \cdot G \cdot P = \begin{bmatrix} 1011000 \\ 1010011 \\ 0011000 \\ 1101101 \end{bmatrix}$$

h. Menentukan m yaitu 1110.

i. Pilih sembarang vektor $e = [0011100]$ berukuran $1 \times n$ atau 1×7.

j. Lakukan proses enkripsi untuk memperoleh *ciphertext c* dengan cara:

$$c = m \cdot G' + e$$

$$c = [1110] \cdot \begin{bmatrix} 1011000 \\ 1010011 \\ 0011000 \\ 1101101 \end{bmatrix} + [0011100] = [1001110]$$

Kemudian, pesan berupa *ciphertext c* dikirim ke Alice sebagai penerima pesan.

3. Proses Dekripsi

   a. Misalkan Alice telah menerima pesan rahasia c dari Bob. Alice menghitung (P^{-1}), invers dari P.

   b. Alice menghitung $c' = c(P^{-1})$.

   c. Alice menggunakan algoritma *decoding* untuk kode C untuk proses *decoding* c' menjadi m'.

   d. Alice menghitung $m = m' \cdot S^{-1}$

   Contoh dekripsi, sebagai berikut:

   a. Menghitung (P^{-1}), invers dari P.

$$P^{-1} = \begin{bmatrix} 0010000 \\ 0000010 \\ 0000001 \\ 0001000 \\ 1000000 \\ 0100000 \\ 0000100 \end{bmatrix}$$

   b. Menghitung $c' = c(P^{-1})$, sehingga

$$c' = [1001110] \cdot \begin{bmatrix} 0010000 \\ 0000010 \\ 0000001 \\ 0001000 \\ 1000000 \\ 0100000 \\ 0000100 \end{bmatrix} = [1111000]$$

   c. Penerima menggunakan algoritma *decoding* untuk kode c untuk proses *decoding* c' menjadi m' menggunakan kode Hamming dan

diperoleh $m' = [0100101]$.

d. Kemudian dari m', diambil 4 digit pertamanya, sehingga menjadi $m' = [0100]$.

e. Setelah itu, untuk mendapatkan pesan asli, pengirim menghitung $m = m' \cdot S^{-1}$. Sehingga diperoleh

$$S^{-1} = \begin{bmatrix} 1001 \\ 1110 \\ 1010 \\ 0010 \end{bmatrix}$$

$$m = m' \cdot S^{-1} = [0100] \cdot \begin{bmatrix} 1001 \\ 1110 \\ 1010 \\ 0010 \end{bmatrix} = [1110]$$

f. Proses dekripsi selesai.

2.10 Kajian Integrasi Topik dengan al-Quran Hadits

Konsep dalam al-Quran yang berhubungan dengan ilmu kriptografi terdapat pada terjemahan surat *al-Kahfi* [18] ayat 94-98 yang berbunyi (Kemenag, 2023):

Artinya : *"Hai Dzulkarnain , sesungguhnya Ya'juj dan Ma'juj itu orang-orang yang membuat kerusakan di muka bumi, maka dapatkah kami memberikan sesuatu pembayaran kepadamu, supaya kamu membuat dinding antara kami dan mereka?."* (QS. *al-Kahfi*/18:94). *Dzulkarnain berkata: "Apa yang telah dikuasakan oleh Tuhanku kepadaku terhadapnya adalah lebih baik, maka tolonglah aku dengan kekuatan (manusia dan alat-alat), agar aku membuatkan dinding antara kamu dan mereka,*(QS. *al-Kahfi*/18:95). *berilah aku potongan-potongan besi." Hingga apabila besi itu telah sama rata dengan kedua (puncak) gunung itu, berkatalah Dzulkarnain: "Tiuplah (api itu)." Hingga apabila besi itu sudah menjadi (merah seperti) api, diapun berkata: "Berilah aku tembaga (yang mendidih) agar aku kutuangkan ke atas besi panas itu."*(QS. *al-Kahfi*/18:96).*Maka, mereka (Ya'juj dan Ma'juj) tidak mampu mendakinya dan tidak mampu (pula) melubanginya.*(QS. *al-Kahfi*/18:97). *Dia (Zulqarnain) berkata, "(Tembok ini adalah rahmat dari Tuhanku. Apabila janji Tuhanku telah tiba, Dia akan menjadikannya hancur luluh. Janji Tuhanku itu benar."*(QS. *al-Kahfi*/18:98).

Dalam ayat ini dijelaskan secara tidak eksplisit mengenai keamanan informasi. Namun praktik dan prinsip *security* atau keamanan secara tersirat terkandung di dalamnya seperti contoh tentang kisah Zulkarnain dengan Ya'juj dan Ma'juj. Pada saat itu Zulkarnain diminta untuk membangun dinding tinggi dan tebal yang tidak dapat ditembus oleh Ya'juj dan Ma'juj untuk melindungi kaumnya dari kejahatan mereka. Zulkarnain pun kemudian membangun dinding yang terbuat dari bahan tembaga dan besi panas. Dinding tersebut digunakan untuk memenuhi kebutuhan kaum yang membutuhkan keamanan tersebut. Kisah tersebut tersirat dalam al-Qur'an surah *al-Kahfi* ayat 94-98. Konsep dinding tembaga dan besi panas tersebut kemudian dapat diadopsi dalam dunia digitalisasi yang populer dengan sebutan *Cyber Security* (Keamanan Siber). Fungsi *security* ini tidak lain adalah untuk menangkal adanya serangan *Cyber Crime* seperti pengaksesan dari pihak-pihak yang tidak dikehendaki terhadap data yang dimiliki oleh seseorang.

Security (keamanan) menurut Hukum Islam juga berkaitan dengan Asas Keamanan dan Keselamatan. Dalam hukum Islam ada lima hal yang wajib dijaga dan dipelihara (*al-dharuriyyat al-khamsah*), yaitu: (1) memelihara agama (*hifdh al-din*), (2) memelihara jiwa (*hifdh al-nafs*), (3) memelihara akal (*hifdh al-aql*), (4) memelihara keturunan (*hifdh nasl*), dan (5) memelihara harta (*hifdh al-maal*). Jika ditelaah dari proses awal hingga dalam praktik keamanan siber yang diterapkan telah mengaplikasikan dan memelihara dari *al-dharuriyyat alkhamsa* (Munawarah & Yusuf, 2022).

2.11 Kajian Topik dengan Teori Pendukung

Seiring dengan perkembangan zaman, informasi semakin mudah untuk diakses dimanapun dan kapanpun. Akan tetapi dengan kemudahan akses tersebut tidak menutup kemungkinan bahwa suatu informasi juga akan lebih mudah untuk dicuri dan disalahgunakan. Terlebih lagi dengan semakin maraknya penggunaan media sosial dalam kegiatan sehari-hari akan memudahkan penyadap untuk mencuri informasi yang didistribusikan melalui internet khususnya media sosial. Untuk mengantisipasi hal-hal yang tidak diinginkan maka perlu adanya bantuan dengan memanfaatkan cabang dari ilmu matematika yaitu kriptografi dan teori pengkodean.

Kriptografi dan teori pengkodean dapat membantu menyamarkan sebuah informasi agar tidak diketahui orang lain selain yang berhak menerimanya. Penerapan ilmu kriptografi tidak terlepas dari konsep matematika dalam bidang aljabar. Seperti penggunaan teori bilangan dalam penghitungan kunci dalam algoritma kriptografi, kemudian penggunaan matriks generator dalam proses pembangkitan kunci. Dalam penelitian ini digunakan kode Golay dan kriptosistem *McEliece* untuk mengamankan pesan teks. Kedua hal tersebut digunakan agar mendapatkan tingkat keamanan yang lebih tinggi.

Peneliti mengimplementasikan pembangkitkan kunci pada kode Golay dengan memodifikasinya menggunakan kriptosistem *McEliece*, sehingga pembangkit kunci tersebut memiliki formula yang berbeda dengan pembangkit kunci yang asli. Kunci publik dan kunci privat adalah dua kunci yang dimiliki kode Golay, dalam penelitian ini peneliti hanya membangkitkan kunci publik. Setelah proses modifikasi kunci sudah selesai, kemudian dilanjutkan dengan mengenkripsi

plaintext dengan menggunakan kriptosistem *McEliece* hingga mendapatkan *ciphertext*. Selanjutnya, untuk mengembalikan kembali *ciphertext* hingga menjadi *plaintext* seperti semula, peneliti melakukan proses dekripsi menggunakan kriptosistem *McEliece*.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Peneliti pada penelitian ini menggunakan algoritma kualitatif karena pada penelitian ini melakukan studi literatur yaitu kegiatan dalam penelitian yang bertujuan untuk mengembangkan aspek teoritis maupun aspek manfaat praktis dengan menemukan referensi dan hasil-hasil riset yang berkaitan dengan bidang ilmu. Data yang dikumpulkan dalam penelitian ini bersifat deskriptif dalam bentuk kata-kata seperti jurnal, laporan hasil penelitian, buku yang relevan, tesis, jurnal, artikel, skripsi dan lain sebagainya yang mendukung penelitian ini.

3.2 Pra Penelitian

Proses pra penelitian ini diawali dengan mencari referensi penelitian melalui sumber penelitian dari jurnal, artikel, buku yang relevan serta semacamnya. Kemudian, peneliti menganalisis suatu masalah untuk mendapatkan algoritma yang sesuai. Selanjutnya, peneliti merancang algoritma yang akan digunakan pada proses penelitian selanjutnya.

3.3 Tahapan Penelitian

1. Proses Pembentukan Kunci
    a. Bangkitkan generator matriks dari kode Golay (24, 12, 8) yang dapat mengkodekan $(k = 12)$ bit data dalam satu *word* dengan panjang $(n = 24)$ dan $(t = 3)$ kesalahan yang dapat diperbaiki

b. Bangkitkan matriks P dengan ukuran 24×24 yang merupakan matriks permutasi

c. Bangkitkan matriks *S* dengan ukuran 12×12 yang merupakan matriks non-singular yang mempunyai invers

d. Hitung matriks (G') menggunakan rumus, sebagai berikut:

$$G' = S \cdot G \cdot P$$

dengan G' tersebut merupakan kunci publik dari kriptosistem ini.

2. Proses Enkripsi

Pada proses enkripsi diperlukan kunci publik dan kesalahan yang akan ditambahkan dengan bobot maksimal tiga karena kode Golay hanya bisa mengoreksi tiga kesalahan. Untuk melakukan proses *encoding*, pengirim membuat pesan *m* kemudian mengubah pesan tersebut ke biner, setelah menjadi biner, dilakukan pembagian panjang kode biner menjadi panjang dari *k*. Setelah itu kode tersebut dikalikan dengan G' sehingga menjadi *codeword* dengan panjang *n*. Kemudian ditambahkan kesalahan pada *codeword* tersebut sehingga menjadi *ciphertext*. Algoritma *encoding*, sebagai berikut:

a. Hitung *codeword*:

$$C_i = m \cdot G'$$

b. Tambahkan kesalahan pada *codeword*:

$$C_i' = C_i + e$$

3. Proses Dekripsi

Pada proses dekripsi diperlukan invers matriks *P*, invers matriks *S*, dan proses *decoding* untuk kode Golay diperpanjang. Untuk melakukan proses

decoding penerima melakukan perkalian vektor *c* terhadap invers matriks *P*. Algoritma *decoding*, sebagai berikut:

a. Hitung nilai $y_i = C_i' \cdot P^{-1}$

b. Hilangkan vektor kesalahan menggunakan fungsi *decoding*:

$$y_i' = y_i + e'$$

c. Hitung m' dengan mengalikan y_i' dan S^{-1}

d. Diperoleh m atau proses dekripsi berhasil.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembentukan Kunci

Algoritma pembentukan kunci pembangkit dalam algoritma *McEliece* memiliki peranan penting dalam memastikan tingkat keamanan dan efektivitas sistem kriptografi. Dengan tahapan awal menetapkan nilai n, k, dan d, matriks generator G menjadi dasar dari kunci publik. Keutamaan algoritma *McEliece* terletak pada kemampuannya menggunakan matriks permutasi acak P dan matriks non-singular S yang dibangkitkan secara acak untuk membentuk kunci tambahan G'. Proses ini menghasilkan ketidakteraturan struktual yang meningkatkan keamanan kunci publik, menjadikan algoritma *McEliece* sebagai pendekatan yang kuat dalam melindungi pesan.

4.1.1 Algoritma Pembentukan Kunci pada Algoritma *McEliece*

1. *Input* pembentukan kunci, sebagai berikut:
    a. Bangkitkan generator matriks dari kode Golay (24, 12, 8) yang dapat mengkodekan $(k = 12)$, $(n = 24)$, dan $(t = 3)$ kesalahan yang dapat diperbaiki oleh matriks (G) berukuran (24×12)
    b. Bangkitkan matriks (P) dengan ukuran (24×24) yang merupakan matriks permutasi
    c. Bangkitkan matriks (S) dengan ukuran (12×12) yang merupakan matriks non-singular yang mempunyai invers.

2. Proses pembentukan kunci, sebagai berikut:

   Hitung matriks (G') menggunakan rumus, berikut:

$$G' = S \cdot G \cdot P$$

3. *Output* pembentukan kunci, sebagai berikut:

   Matriks (G') yang akan digunakan sebagai kunci publik. Matriks $(S), (G),$ dan (P) sebagai kunci privat.

4.1.2 Simulasi Pembentukan Kunci pada Algoritma *McEliece*

Dalam tahap awal sistem kriptografi *McEliece*, penerima mengawalinya dengan membangkitkan kunci kode Golay untuk menentukan parameter kunci $(n), (k),$ dan (d) dengan nilai $(n = 24), (k = 12),$ dan $(d = 8)$. Selanjutnya, matriks generator (G), ditentukan sebagai berikut:

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

Setelahnya, dilakukan pembangkitan matriks (S) secara acak dengan ukuran (12×12) dan matriks (P) dengan ukuran (24×24).

$$S = \begin{bmatrix} 110001100100 \\ 011000010100 \\ 100111111110 \\ 010010000011 \\ 000110111100 \\ 111000001101 \\ 000000111000 \\ 111000011001 \\ 010100100001 \\ 100101110001 \\ 100011110110 \\ 110011011011 \end{bmatrix}$$

$$P = \begin{bmatrix} 000000000000000000000001 \\ 010000000000000000000000 \\ 000000000000010000000000 \\ 000000000000000000000100 \\ 000000100000000000000000 \\ 000010000000000000000000 \\ 000000000000000010000000 \\ 000000000010000000000000 \\ 000001000000000000000000 \\ 000000000001000000000000 \\ 000000000000000000000010 \\ 001000000000000000000000 \\ 100000000000000000000000 \\ 000000000000000000001000 \\ 000100000000000000000000 \\ 000000000000000000100000 \\ 000000001000000000000000 \\ 000000000100000000000000 \\ 000000000000100000000000 \\ 000000000010000000000000 \\ 000000000000000001000000 \\ 000000000000000100000000 \\ 000000000000000000010000 \\ 000000010000000000000000 \end{bmatrix}$$

Selanjutnya, matriks generator (G') dihitung dengan mengalikan matriks $(S), (G),$ dan (P).

$$G' = S \cdot G \cdot P$$

$$= \begin{bmatrix} 010010011101100110101001 \\ 011100010011110101011000 \\ 001011110011000010100111 \\ 111100110000001000000010 \\ 101101100011000010111100 \\ 011001011101111101111001 \\ 101101011110100010110000 \\ 010101011010111001001001 \\ 110100000000101110000100 \\ 101110001110101010000101 \\ 000110111011000011010011 \\ 111111111010001001011011 \end{bmatrix}$$

Dengan demikian, matriks generator (G') dengan ukuran (12×24) berhasil dibentuk dan dapat digunakan sebagai kunci publik pada tahap enkripsi dalam algoritma *McEliece*. Proses ini menunjukkan kerangka kerja yang kompleks dan kuat dalam membangun kunci pada algoritma *McEliece*, suatu langkah penting dalam menjaga keamanan informasi melalui sistem kriptografi kunci publik.

4.2 Proses Enkripsi

Proses enkripsi pada algoritma *McEliece* membuktikan kehandalan dan ketangguhan sistem dalam melindungi informasi sensitif. Melalui konversi pesan teks menjadi representasi biner dengan panjang 8 bit ke 12 bit, langkah awal ini memastikan bahwa setiap karakter pesan memiliki representasi yang unik. Selanjutnya, matriks generator G' yang telah dibentuk menjadi kunci publik digunakan untuk menghasilkan blok-blok *ciphertext*, menjadikan pesan tidak mudah terbaca oleh pihak yang tidak berhak. Penambahan *error* acak pada setiap blok *ciphertext* memberikan lapisan keamanan, melengkapi proses enkripsi ini

sebagai langkah kritis dalam menjaga kerahasiaan pesan dalam sistem kriptografi *McEliece*.

4.2.1 Algoritma Enkripsi Menggunakan *McEliece*

1. *Input* enkripsi menggunakan *McEliece*, sebagai berikut:
   a. Menentukan *plaintext* yang akan dikirimkan
   b. Menentukan vektor *e* berukuran 24 bit yang memiliki *t* elemen non-nol (vektor dengan panjang 24 dan bobot 3).
2. Proses enkripsi menggunakan *McEliece*, sebagai berikut:
   a. Mengubah *plaintext* ke bentuk *binary m* berdasarkan tabel ASCII
   b. Kode biner dibagi menjadi blok-blok dengan panjang 12 bit
   c. Hitung *codeword* $C_i = m_i \cdot G'$
   d. Tambahkan kesalahan pada *codeword* $C_i' = C_i + e$.
3. *Output* enkripsi menggunakan *McEliece*, sebagai berikut:
   Matriks *y* merupakan *ciphertext*.

4.2.2 Simulasi Proses Enkripsi Menggunakan *McEliece*

Selanjutnya pada bagian ini pengirim mengubah *plaintext* ke bentuk biner berdasarkan tabel ASCII dengan panjang 8 bit. Sebagai contoh pengirim akan mengirimkan *plaintext* "Sekar" dengan $e = [000000010000001000001000]$, berdasarkan tabel ASCII didapatkan pesan, sebagai berikut:

Sekar = 01010011011001010110101101100000101110010

S = $[01010011]$

e = $[01100101]$

k = $[01101011]$

a = [01100001]

r = [01110010]

Selanjutnya dilakukan penambahan bit *padding* dari 8 bit ke 12 bit dikarenakan pesan *m* modulo *k* tidak sama dengan nol. Oleh karena itu, dilakukan *padding* pada *plaintext* terlebih dahulu agar dapat dienkripsi, sebagai berikut:

m =010100110110010101101011011000010111001000000000000000000000

Kemudian membagi kode biner ke beberapa blok m_i dengan panjang 12 bit.

$$m_1 = [010100110110]$$

$$m_2 = [010101101011]$$

$$m_3 = [011000010111]$$

$$m_4 = [001000000000]$$

$$m_5 = [000000000000]$$

Setelah itu, *plaintext* m_i yang telah diubah ke bentuk biner dikalikan dengan matriks generator G' untuk menghasilkan matriks C_i yang merupakan *codeword*.

$$C_1 = m_1 \cdot G' = [110000010010001111110101]$$

$$C_2 = m_2 \cdot G' = [011001100001000110011111]$$

$$C_3 = m_3 \cdot G' = [010101110101101110111011]$$

$$C_4 = m_4 \cdot G' = [001011110011000010100111]$$

$$C_5 = m_5 \cdot G' = [000000000000000000000000]$$

Kemudian C_i ditambahkan *error* acak e dengan panjang 24 bit, dalam hal ini $e = [000000010000001000001000]$ dan didapatkan nilai C_i' sebagai berikut:

$$C_1' = C_1 + e = [110000000010000111111101]$$

$$C_2' = C_2 + e = [011001110001001110010111]$$

$$C_3' = C_3 + e = [010101100101100110110011]$$

$$C_4' = C_4 + e = [001011100011001010101111]$$

$$C_5' = C_5 + e = [000000010000001000001000]$$

Sehingga dari penjumlahan di atas diperoleh *Ciphertext* dari pesan Sekar = [110000000010000111111101011001110001001110010111010101100101100110011100101110001100101010111100000001000000100000 1000].

4.3 Proses Dekripsi dan Pengoreksian *Error*

Langkah-langkah untuk mengembalikan pesan ke bentuk semula. Dimulai setelah *ciphertext* diterima. Langkah awal dalam proses dekripsi ini adalah mengindentifikasi invers matriks (P) dan (S) yang sebelumnya digunakan pada tahap pembentukan kunci. Selanjutnya, invers matriks (P) ini akan dikalikan dengan setiap blok *ciphertext*. Kemudian masuk ke tahap pengoreksian *error* untuk mengalikan blok *ciphertext* pertama dengan matriks *parity check H*. Setelah tahap pengoreksian *error, ciphertext* yang telah dikoreksi *error* dikalikan dengan invers dari matriks (S) dan hasilnya akan diubah ke dalam bentuk teks kembali berdasarkan tabel ASCII.

4.3.1 Algoritma Dekripsi Menggunakan Algoritma *McEliece* dengan Kode Golay

1. *Input* dekripsi menggunakan *McEliece*, sebagai berikut:

   a. *Ciphertext* C_i' dengan panjang (12 bit).

   b. Menghitung invers dari matriks P berukuran (24×24).

c. Menghitung invers dari matriks S berukuran (12×12).

d. Menentukan matriks *parity check H*.

2. Proses dekripsi menggunakan *McEliece*, sebagai berikut:

a. Menentukan (y_i) dengan mengalikan *ciphertext* (C_i') dengan matriks (P^{-1}).

b. Mencari *syndrome* $S(y_i)$ dengan mengalikan matriks *parity check* H^T dengan matriks (C_1'), sebagai berikut:

$$S(y_i) = C_1' \cdot H^T$$

c. Mencari letak bit yang *error*.

d. Mengoreksi bit yang *error*.

e. Menentukan (m_i') dengan mengalikan matriks (y_i') yang telah dikoreksi dengan matriks (S^{-1}).

f. *Binary* (m_i') yang diperoleh dikembalikan ke dalam bentuk teks berdasarkan tabel ASCII.

3. *Output* dekripsi menggunakan *McEliece*, sebagai berikut:

Menghasilkan teks sesuai dengan pesan awal.

4.3.2 Simulasi Proses Dekripsi Algoritma *McEliece* dengan Kode Golay

Proses dekripsi dengan pengoreksian *error* dapat dilakukan setelah menerima *ciphertext.* Selanjutnya, penerima menghitung nilai dari P^{-1}, sebagai berikut:

$$P^{-1} = \begin{bmatrix} 000000000000010000000000 \\ 010000000000000000000000 \\ 000000000001000000000000 \\ 000000000000000100000000 \\ 000001000000000000000000 \\ 000000010000000000000000 \\ 000010000000000000000000 \\ 000000000000000000000001 \\ 000000000000000001000000 \\ 000000000000000000100000 \\ 000000010000000000000000 \\ 000000001000000000000000 \\ 000000000000000000010000 \\ 001000000000000000000000 \\ 000000000001000000000000 \\ 000000000000000000000100 \\ 000000100000000000000000 \\ 000000000000000000001000 \\ 000000000000000010000000 \\ 000000000000000000000010 \\ 000000000000001000000000 \\ 000100000000000000000000 \\ 000000000010000000000000 \\ 100000000000000000000000 \end{bmatrix}$$

Setelah nilai P^{-1} didapatkan, *ciphertext* (C_i') dikalikan dengan P^{-1} untuk menghasilkan matriks y_i', sebagai berikut:

$$y_1 = C_1' \cdot P^{-1} = [110100110000011010001110]$$

$$y_2 = C_2' \cdot P^{-1} = [110110101111100000000111]$$

$$y_3 = C_3' \cdot P^{-1} = [110010101110000110110110]$$

$$y_4 = C_4' \cdot P^{-1} = [100111111111101010000000]$$

$$y_5 = C_5' \cdot P^{-1} = [000000000001001000000001]$$

Setelah itu, memasuki tahap pengoreksian *error* dengan mencari nilai dari *syndrome* $S(y_i)$. Untuk mencari nilai dari *syndrome* $S(y_i)$, matriks *parity check* H^T akan dikalikan dengan *binary* (y_i), sebagai berikut:

$$S(y_i) = C_i' \cdot H^T$$

Matriks *parity check* H didapatkan dari matriks G, matriks generator berbentuk $G = [I_{12}|A]$ dengan I_{12} adalah matriks identitas berukuran 12×12 dan A adalah matriks paritas berukuran 12×12. Matriks identitas digunakan agar bit-bit asli pesan tetap utuh dalam kode yang dihasilkan, sedangkan matriks paritas ditambahkan untuk memungkinkan deteksi dan koreksi kesalahan. Kombinasi ini memberikan kemampuan koreksi kesalahan yang kuat pada kode Golay dan matriks *parity check* $H^T = [I_{12}|A]^T$. Maka didapatkan matriks H^T berukuran 12×24.

$$G = \left[\begin{array}{cc} \underbrace{\begin{matrix} 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{matrix}}_{I_{12}} & \underbrace{\begin{matrix} 1&0&1&0&1&1&1&0&0&0&1&1 \\ 1&1&1&1&1&0&0&1&0&0&1&0 \\ 1&1&0&1&0&0&1&0&1&0&1&1 \\ 1&1&0&0&0&1&1&1&0&1&1&0 \\ 1&1&0&0&1&1&0&1&1&0&0&1 \\ 0&1&1&0&0&1&1&0&1&1&0&1 \\ 0&0&1&1&0&0&1&1&0&1&1&1 \\ 1&0&1&1&0&1&1&1&1&0&0&0 \\ 0&1&0&1&1&0&1&1&1&1&0&0 \\ 0&0&1&0&1&1&0&1&1&1&1&0 \\ 1&0&1&1&1&0&0&0&1&1&0&1 \\ 0&1&0&1&1&1&0&0&0&1&1&1 \end{matrix}}_{A} \end{array}\right]$$

$$H^T = \begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&1\\
1&1&1&1&1&0&0&1&0&0&1&0\\
0&1&1&1&1&1&0&0&1&0&0&1\\
1&1&0&0&0&1&1&1&0&1&1&0\\
0&1&1&0&0&0&1&1&1&0&1&1\\
1&1&0&0&1&0&0&0&1&1&1&1\\
1&0&0&1&1&1&0&1&0&1&0&1\\
1&0&1&1&0&1&1&1&1&0&0&0\\
0&1&0&1&1&0&1&1&1&1&0&0\\
0&0&1&0&1&1&0&1&1&1&1&0\\
0&0&0&1&0&1&1&0&1&1&1&1\\
1&1&1&1&0&0&1&0&0&1&0&1\\
1&0&1&0&1&1&1&0&0&0&1&1
\end{bmatrix}$$

Setelah, matriks transpose *parity check* H didapatkan selanjutnya mencari *syndrome* $S(y_i)$ dengan mengalikan matriks transpose *parity check* H dan C_1'.

$$S(y_i) = C_1' \cdot H^T$$

$$= [110000000010000111111101] \cdot \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&1&1&1&1&0&0&1&0&0&1&0 \\ 0&1&1&1&1&1&0&0&1&0&0&1 \\ 1&1&0&0&0&1&1&1&0&1&1&0 \\ 0&1&1&0&0&0&1&1&1&0&1&1 \\ 1&1&0&0&1&0&0&0&1&1&1&1 \\ 1&0&0&1&1&1&0&1&0&1&0&1 \\ 1&0&1&1&0&1&1&1&1&0&0&0 \\ 0&1&0&1&1&0&1&1&1&1&0&0 \\ 0&0&1&0&1&1&0&1&1&1&1&0 \\ 0&0&0&1&0&1&1&0&1&1&1&1 \\ 1&1&1&1&0&0&1&0&0&1&0&1 \\ 1&0&1&0&1&1&1&0&0&0&1&1 \end{bmatrix}$$

$$= [100011110101]$$

Jika nilai dari *syndrome* $S(y_i) \neq 0$, maka hal ini menunjukkan bahwa terjadi *error* pada y_i. Untuk mencari letak posisi bit yang salah dengan menambahkan $S(y_i)$ dan setiap baris pada matriks A.

$s + A_1 = [100011110101] + [101011100011] = [001000010110]$, maka

$|s + A_1| = 4 > 2$

$s + A_2 = [100011110101] + [111110010010] = [011101100111]$, maka

$|s + A_2| = 8 > 2$

$s + A_3 = [100011110101] + [110100101011] = [010111011110]$, maka

$|s + A_3| = 8 > 2$

$s + A_4 = [100011110101] + [110001110110] = [010010000011]$, maka

$|s + A_4| = 4 > 2$

$s + A_5 = [100011110101] + [110011011001] = [010000101100]$, maka

$|s + A_5| = 4 > 2$

$s + A_6 = [100011110101] + [011001101101] = [111010011000]$, maka

$|s + A_6| = 6 > 2$

$s + A_7 = [100011110101] + [001100110111] = [101111000010]$, maka

$|s + A_7| = 6 > 2$

$s + A_8 = [100011110101] + [101101111000] = [001110001100]$, maka

$|s + A_8| = 5 > 2$

$s + A_9 = [100011110101] + [010110111100] = [110101001001]$, maka

$|s + A_9| = 6 > 2$

$s + A_{10} = [100011110101] + [001011011110] = [101000101011]$, maka

$|s + A_{10}| = 6 > 2$

$s + A_{11} = [100011110101] + [101110001101] = [001101111000]$, maka

$|s + A_{11}| = 6 > 2$

$s + A_{12} = [100011110101] + [010111000111] = [110100110010]$, maka

$|s + A_{12}| = 6 > 2$

$sA = 011111000110$, maka $|sA| = 7 > 3$

$sA + A_1 = [011111000110] + [101011100011] = [110100100101]$, maka

$|sA + A_1| = 6 > 2$

$sA + A_2 = [011111000110] + [111110010010] = [100001010000]$, maka

$|sA + A_2| = 3 > 2$

$sA + A_3 = [011111000110] + [110100101011] = [101011101101]$, maka

$|sA + A_3| = 8 > 2$

$sA + A_4 = [011111000110] + [110001110110] = [101110110000]$, maka $|sA + A_4| = 7 > 2$

$sA + A_5 = [011111000110] + [110011011001] = [101100011111]$, maka $|sA + A_5| = 8 > 2$

$sA + A_6 = [011111000110] + [011001101101] = [101100011111]$, maka $|sA + A_6| = 6 > 2$

$sA + A_7 = [011111000110] + [001100110111] = [010011110001]$, maka $|sA + A_7| = 6 > 2$

$sA + A_8 = [011111000110] + [101101111000] = [110010111110]$, maka $|sA + A_8| = 8 > 2$

$sA + A_9 = [011111000110] + [010110111100] = [001001111010]$, maka $|sA + A_9| = 6 > 2$

$sA + A_{10} = [011111000110] + [001011011110] = [010100011000]$, maka $|sA + A_{10}| = 4 > 2$

$sA + A_{11} = [011111000110] + [101110001101] = [110001001011]$, maka $|sA + A_{11}| = 6 > 2$

$sA + A_{12} = [011111000110] + [010111000111] = [001000000001]$, maka $|sA + A_{12}| = 2$, memenuhi.

Dari hasil perhitungan di atas, maka diperoleh $|sA + A_{12}| = 2$ yang artinya *error* dimulai dari bit ke-12. Sehingga diperoleh

$e' = [000000000001001000000001]$

Setelah posisi *error* ditemukan, langkah selanjutnya adalah pengoreksian *error* dengan menambahkan $error\ e'$ dengan bobot 1 pada bit yang salah.

$y_1' = y_1 + e' = [110100110001010010001111]$

$y_2' = y_2 + e' \ = [110110101110101000000110]$

$y_3' = y_3 + e' \ = [110010101111001110110111]$

$y_4' = y_4 + e' \ = [100111111110100010000001]$

$y_5' = y_5 + e' \ = [000000000000000000000000]$

Setelah y_i dikoreksi maka didapatkan y_i' karena y_i' dihasilkan dari *standard form generator matriks* dengan 12 bit dari kiri merupakan pesan informasi atau data bit dan 12 bit dari kanan merupakan *parity* bit. Langkah selanjutnya adalah dengan mengambil 12 bit *binary* y_i' dari kiri untuk dikalikan dengan S invers.

$m_i' = y_i' \cdot S^{-1}$

Didapatkan nilai dari invers matriks $S^{-1} = \begin{bmatrix} 011101100111 \\ 101100101000 \\ 111101111000 \\ 100000100111 \\ 101010001110 \\ 111000011010 \\ 100001011100 \\ 001001011001 \\ 101000100101 \\ 001000001001 \\ 101111111101 \\ 101101011011 \end{bmatrix}$

Kemudian y_i' dikalikan dengan invers matriks S

$$m_1' = y_1' \cdot S^{-1} = [010100110110]$$

$$m_2' = y_2' \cdot S^{-1} = [010101101011]$$

$$m_3' = y_3' \cdot S^{-1} = [011000010111]$$

$$m_4' = y_4' \cdot S^{-1} = [001000000000]$$

$$m_5' = y_5' \cdot S^{-1} = [000000000000]$$

Didapatkan hasil dari dekripsi, sebagai berikut:

010100110110010101101011011000010111001000000000000000000000

Kemudian dilakukan *unpadding*, sehingga menjadi:

$$010100110110010101101011011000010111001 0$$

Untuk mengembalikan ke dalam bentuk teks, perlu membagi pesan biner tersebut ke dalam blok-blok dengan panjang 8-bit. Berdasarkan tabel ASCII, diperoleh sebagai berikut:

$01010011 - 01100101 - 01101011 - 01100001 - 01110010$ = Sekar.

4.4 Analisis Hasil

Pembentukan kunci pada algoritma *McEliece* menunjukkan bahwa langkah-langkah yang kompleks dan kuat telah berhasil membentuk matriks generator (G') dengan ukuran 12×24 . Tahap awal penerima melibatkan algoritma *decoding*, dalam hal ini menggunakan metode kode Golay untuk menentukan parameter kunci $(n), (k),$ dan (d). Proses ini penting dalam membentuk dasar keamanan sistem kriptografi *McEliece*. Matriks generator (G) terbentuk sebagai hasil dari algoritma *decoding* dan menjadi dasar untuk pembangkitan matriks (S) dan (P) yang dibangkitkan secara acak. Penggunaan matriks permutasi acak (P) dan matriks (S) non-singular menciptakan kekacauan struktural yang memperkuat keamanan kunci publik.

Proses enkripsi pada algoritma *McEliece* menunjukkan kehandalan dan ketangguhan sistem dalam melindungi informasi sensitif. Tahap awal pengirim yang melibatkan konversi pesan teks menjadi representasi biner dengan panjang 8 bit, dan dilakukan proses *padding* kemudian membagi blok-blok dengan panjang 12 bit. Perkalian dengan matriks generator (G') sebagai kunci publik menjadikannya sulit terbaca oleh pihak yang tidak berhak. Proses penggabungan *error* acak pada

setiap blok *ciphertext* memberikan lapisan tambahan keamanan, menjadikan proses enkripsi sebagai langkah kritis dalam menjaga kerahasiaan pesan dalam sistem algoritma *McEliece*.

Proses dekripsi dan pengkoreksian *error* pada algoritma *McEliece* dengan kode Golay melibatkan beberapa langkah penerima untuk mengembalikan pesan terenkripsi ke bentuk semula. Setelah menerima *ciphertext*, langkah awal penerima adalah mengidentifikasi matriks invers P dan S yang sebelumnya digunakan dalam pembentukan kunci. Matriks invers ini digunakan untuk mendekripsi setiap blok pesan terenkripsi, dan hasilnya dikombinasikan untuk menghasilkan pesan biner dalam format ASCII. Pada tahap dekripsi, setiap blok pesan terenkripsi dikalikan dengan matriks invers P, kemudian pengkoreksian *error* dengan mencari *syndrome* $S(y_i)$ dihitung dengan mengalikan matriks transpose *parity check* H dengan matriks hasil dekripsi. Jika *syndrome* tidak sama dengan nol, itu menunjukkan adanya *error* pada pesan, dan lokasi bit yang salah dapat ditemukan. Setelah lokasi *error* ditemukan, dilakukan pengkoreksian *error* dengan menambahkan *error* pada bit yang terdeteksi. Selanjutnya, hasil dikalikan dengan matriks invers S untuk mendapatkan pesan yang telah dikoreksi. Hasil *binary* dikembalikan ke bentuk teks berdasarkan tabel ASCII. Dengan mengikuti langkah-langkah ini, algoritma *McEliece* dengan kode Golay dapat mengamankan dan mendekripsi pesan secara efektif.

Penelitian ini berhasil menunjukkan bahwa algoritma *McEliece* mampu menghadirkan langkah-langkah pengamanan pesan yang kompleks dan kuat. Algoritma *McEliece* dalam penelitian ini melibatkan langkah-langkah

pengkoreksian *error* dengan kode Golay yang menunjukkan keberhasilan dalam mengembalikan pesan terenkripsi ke bentuk semula.

4.5 Kajian Penelitian dalam Perspektif Islam

Security (keamanan) menurut hukum Islam juga berkaitan dengan Asas Keamanan dan Keselamatan. Dalam hukum Islam ada lima hal yang wajib dijaga dan dipelihara (*al-dharuriyyat alkhamsah*), yaitu: (1) memelihara agama (*hifdh al-din*), (2) memelihara jiwa (*hifdh al-nafs*), (3) memelihara akal (*hifdh al-aql*), (4) memelihara keturunan (*hifdh nasl*), dan memelihara harta (*hifdh al-maal*).Jika ditelaah dari proses awal hingga dalam praktik *cyber security* yang diterapkan telah mengaplikasikan dan memelihara dari *al-dharuriyyat al-khamsa* (Munawarah & Yusuf, 2022).

Oleh karena itu Islam selalu menganjurkan menjaga keamanan dan kemaslahatan pada data pribadi, seperti pesan yang berbentuk teks, suara, maupun gambar. Sebagaimana dalam *al-Qur'an* pada surah *al-Kahfi* [18] ayat 94-98, bahwa dalam perspektif Islam keamanan sangat penting. Ini ditekankan dalam terjemahan ayat tersebut mengenai konsep dinding tembaga dan besi panas tersebut diadopsi dalam keamanan pesan melalui proses enkripsi untuk menghalau akses dari pihak-pihak yang tidak dikehendaki dan tidak bertanggung jawab terhadap pesan yang dimiliki oleh seseorang.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan didapatkan kesimpulan yaitu:

1. Pembentukan kunci publik melibatkan matriks non-singular S berukuran 12×12 dikalikan dengan matriks generator G yang berukuran 12×24 berbentuk $[I_{12}|A]$, dengan I_{12} adalah matriks identitas berukuran 12×12 dan A adalah suatu matriks berukuran $12 \times (24 - 12)$). Kemudian dikalikan lagi dengan matriks permutasi P berukuran 24×24 dan menghasilkan kunci publik G'. Selain itu, juga menghasilkan matriks $S, G,$ dan P sebagai kunci privat.
2. Proses enkripsi dimulai dari mengubah pesan teks menjadi kode biner dengan panjang 12 bit berdasarkan tabel ASCII. Kemudian dibagi ke dalam blok blok m_i dengan panjang 12 bit. Selanjutnya, mengalikan m_i dengan matriks generator G' ($C_i = m_i \cdot G'$) dan hasilnya ditambahkan *error* dengan bobot 1 untuk menghasilkan *cipherteks* ($C_i' = C_i + e$).
3. Proses dekripsi diawali dengan mengalikan *cipherteks* dengan invers matriks permutasi ($y_i = C_i' \times P^{-1}$). Kemudian kode Golay mulai mengkoreksi dan memperbaiki data bit dengan mencari *syndrome* dari y_i . Matriks *parity check* H^T. *Syndrome* didapatkan dari matriks transpose *parity check* H yang dikalikan dengan y_i, $S(y_i) = C_1' \times H^T$. Setelah pesan selesai dikoreksi dan diperbaiki kemudian dikalikan dengan invers dari matriks non-singular S ($m_i' = y_i' \times S^{-1}$). Kemudian blok-blok digabungkan

secara berurutan dan dilakukan *unpadding* sehingga menciptakan biner dengan panjang 8 bit. Dengan mengacu pada tabel ASCII pesan asli dalam bentuk biner dapat diubah ke bentuk teks. Dengan mengimplementasikan kode Golay, algoritma *McEliece* dapat meningkatkan keandalan dan keamanan pertukaran informasi, sekaligus menghadapi potensi ancaman dari komputer kuantum.

5.2 Saran

Saran untuk penelitian selanjutnya, diharapkan dapat melakukan pengamanan pertukaran data dalam bentuk gambar, audio, ataupun video dengan menggunakan algoritma *McEliece* yang menggunakan kode Golay sebagai koreksi *error*.

DAFTAR RUJUKAN

Aini, N. A., & Irawanto, B. (2011). Kontruksi Greedy Kode Lexicographic untuk Membangun Perluasan Kode Golay (24 12,8). *Jurusan Matematika FMIPA UNDIP*, *19*, 1.

Anggoro, D., Bhagaskoro, P., Barmawi, A. M., Informatika, F., Telkom, U., Gambar, E., & Gambar, D. (2020). *Enkripsi Gambar Berbasis Grid menggunakan Kriptografi Berbasis Kode*. *7*(1), 2343–2386.

Basri, H. (2021). *Teori Bilangan*. Eureka Media Aksara. Purbalingga

Ilmiyah, N. F. (2019). Kajian Tentang Kriptosistem Mceliece Dalam Menghadapi Tantangan Komputer Kuantum Di Era Revolusi Industri 4.0. *Prosiding Seminar Nasional MIPA Kolaborasi*, 216–226.

Indriani, H. (2021). *Konsep Wahyu Menurut Al-Qur'an*. Universitas Islam Negeri Sultan Maulana Hasanuddin Banten.

Jamaludin, Sulaiman., O.K., Tandungan, S., Putra, L. M., Yuswardi, Yulianti, N., Sidabutar, J., Aisa, S., Tantriawan, H., Arizal, M., & Pakpahan, A. F. (2022). *Kriptografi: Teknik Keamanan Data*. Yayasan Kita Menulis.

Jochemsz, E. (2002). *Goppa Codes and The McEliece Cryptosystem*. University of Amsterdam at Netherland.

Kemenag. (2023). *Qur'an Kemenag*. https://quran.kemenag.go.id/quran/per-ayat/surah/18?from=94&to=96

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

Lee. (2010). Abstract Algebra An Introductory Course. In *The Mathematical Gazette* (Vol. 24, Issue 258). https://doi.org/10.2307/3607096

Ling, S., & Xing, C. (2004). *Coding Theory*. Cambridge University Press.

Manullang, A. F., Candiwan, C., & Harsono, L. D. (2017). Asesmen Keamanan Informasi Menggunakan Indeks Keamanan Informasi (KAMI) pada Institusi XYZ. *Journal of Information Engineering and Educational Technology*, *1*(2), 73. https://doi.org/10.26740/jieet.v1n2.p73-82

Menezes, A. J., Oorschoot, P. C., & Vanstone, S. A. (1996). *Handbook of Applied Crypthography*. CRC Press.

Muhsetyo, G. (2014). *Teori Bilangan*. Universitas Terbuka.Tangerang Selatan.

Munawarah, H., & Yusuf, M. (2022). *Bank Digital Syariah Analisis Cyber Security Menurut Hukum Positif Di Indonesia Dan Hukum Ekonomi Syariah Hasanatul*.

Munir, R. (2019). *Kriptografi (2nd ed.)*. Informatika Bandung.

Oktavia, R.E, Utomo, P.H. (2023). *Penerapan kode reed solomon pada kriptosistem mceliece*. *9*, 79–88.

Sadikin, R. (2012). *Kriptografi untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java* (T. A. Prabawati (ed.)). CV. Andi Offset.

Şatir, E., & Kendirli, O. (2022). A symmetric DNA encryption process with a biotechnical hardware. *Journal of King Saud University - Science*, *34*(3). https://doi.org/10.1016/j.jksus.2022.101838

Setyawan, I., Utomo, P.H. (2023). *Penerapan Kode Golay Diperpanjang pada Kriptosistem McEliece. 6, 20-23.*

Siim, S. (2015). *Study Of McEliece Cryptosystem*. MTAT.07.022.

Sukirman. (2006). *Pengantar Teori Bilangan*.Hanggar Kreator.Yogyakarta.

Trappe, W. (2006). *Introduction to Cryptography with Coding Theory* (2nd ed.). Pearson Education.

LAMPIRAN

Lampiran 1. Tabel ASCII

Decimal	Octal	Hex	Binary	Value	Description
000	000	000	0000 0000	NUL	"null" character
001	001	001	0000 0001	SOH	start of header
002	002	002	0000 0010	STX	start of text
003	003	003	0000 0011	ETX	end of text
004	004	004	0000 0100	EOT	end of transmission
005	005	005	0000 0101	ENQ	enquiry
006	006	006	0000 0110	ACK	acknowledgment
007	007	007	0000 0111	BEL	bell
008	010	008	0000 1000	BS	backspace
009	011	009	0000 1001	HT	horizontal tab
010	012	00A	0000 1010	LF	line feed
011	013	00B	0000 1011	VT	vertical tab
012	014	00C	0000 1100	FF	form feed
013	015	00D	0000 1101	CR	carriage return
014	016	00E	0000 1110	SO	shift out
015	017	00F	0000 1111	SI	shift in
016	020	010	0001 0000	DLE	data link escape
017	021	011	0001 0001	DC1	device control 1 (XON)
018	022	012	0001 0010	DC2	device control 2
019	023	013	0001 0011	DC3	device control 3 (XOFF)
020	024	014	0001 0100	DC4	device control 4

Decimal	Octal	Hex	Binary	Value	Description
021	025	015	0001 0101	NAK	negative acknowledgement
022	026	016	0001 0110	SYN	synchronous idle
023	027	017	0001 0111	ETB	end of transmission block
024	030	018	0001 1000	CAN	cancel
025	031	019	0001 1001	EM	end of medium
026	032	01A	0001 1010	SUB	substitute
027	033	01B	0001 1011	ESC	escape
028	034	01C	0001 1100	FS	file separator
029	035	01D	0001 1101	GS	group separator
030	036	01E	0001 1110	RS	request to send/record separator
031	037	01F	0001 1111	US	unit separator
032	040	020	0010 0000	SP	space
033	041	021	0010 0001	!	exclamation mark
034	042	022	0010 0010	"	double quote
035	043	023	0010 0011	#	number sign
036	044	024	0010 0100	$	dollar sign
037	045	025	0010 0101	%	percent
038	046	026	0010 0110	&	ampersand
039	047	027	0010 0111	'	single quote
040	050	028	0010 1000	(	left/opening parenthesis
041	051	029	0010 1001	)	right/closing parenthesis
042	052	02A	0010 1010	*	asterisk
043	053	02B	0010 1011	+	plus

Decimal	Octal	Hex	Binary	Value	Description
044	054	02C	0010 1100	,	comma
045	055	02D	0010 1101	-	minus or dash
046	056	02E	0010 1110	.	dot
047	057	02F	0010 1111	/	forward slash
048	060	030	0011 0000	0	
049	061	031	0011 0001	1	
050	062	032	0011 0010	2	
051	063	033	0011 0011	3	
052	064	034	0011 0100	4	
053	065	035	0011 0101	5	
054	066	036	0011 0110	6	
055	067	037	0011 0111	7	
056	070	038	0011 1000	8	
057	071	039	0011 1001	9	
058	072	03A	0011 1010	:	colon
059	073	03B	0011 1011	;	semi-colon
060	074	03C	0011 1100	<	less than
061	075	03D	0011 1101	=	equal sign
062	076	03E	0011 1110	>	greater than
063	077	03F	0011 1111	?	question mark
064	100	040	0100 0000	@	"at" symbol
065	101	041	0100 0001	A	
066	102	042	0100 0010	B	

Decimal	Octal	Hex	Binary	Value	Description
067	103	043	0100 0011	C	
068	104	044	0100 0100	D	
069	105	045	0100 0101	E	
070	106	046	0100 0110	F	
071	107	047	0100 0111	G	
072	110	048	0100 1000	H	
073	111	049	0100 1001	I	
074	112	04A	0100 1010	J	
075	113	04B	0100 1011	K	
076	114	04C	0100 1100	L	
077	115	04D	0100 1101	M	
078	116	04E	0100 1110	N	
079	117	04F	0100 1111	O	
080	120	050	0101 0000	P	
081	121	051	0101 0001	Q	
082	122	052	0101 0010	R	
083	123	053	0101 0011	S	
084	124	054	0101 0100	T	
085	125	055	0101 0101	U	
086	126	056	0101 0110	V	
087	127	057	0101 0111	W	
088	130	058	0101 1000	X	
089	131	059	0101 1001	Y	

Decimal	Octal	Hex	Binary	Value	Description
090	132	05A	0101 1010	Z	
091	133	05B	0101 1011	[	left/opening bracket
092	134	05C	0101 1100	\	back slash
093	135	05D	0101 1101	]	right/closing bracket
094	136	05E	0101 1110	^	caret/circumflex
095	137	05F	0101 1111	_	underscore
096	140	060	0110 0000	`	
097	141	061	0110 0001	a	
098	142	062	0110 0010	b	
099	143	063	0110 0011	c	
100	144	064	0110 0100	d	
101	145	065	0110 0101	e	
102	146	066	0110 0110	f	
103	147	067	0110 0111	g	
104	150	068	0110 1000	h	
105	151	069	0110 1001	i	
106	152	06A	0110 1010	j	
107	153	06B	0110 1011	k	
108	154	06C	0110 1100	l	
109	155	06D	0110 1101	m	
110	156	06E	0110 1110	n	
111	157	06F	0110 1111	o	
112	160	070	0111 0000	p	

Decimal	Octal	Hex	Binary	Value	Description
113	161	071	0111 0001	q	
114	162	072	0111 0010	r	
115	163	073	0111 0011	s	
116	164	074	0111 0100	t	
117	165	075	0111 0101	u	
118	166	076	0111 0110	v	
119	167	077	0111 0111	w	
120	170	078	0111 1000	x	
121	171	079	0111 1001	y	
122	172	07A	0111 1010	z	
123	173	07B	0111 1011	{	left/opening brace
124	174	07C	0111 1100	|	vertical bar
125	175	07D	0111 1101	}	right/closing brace
126	176	07E	0111 1110	~	tilde
127	177	07F	0111 1111	DEL	delete

RIWAYAT HIDUP

Soviana. Dilahirkan di Probolinggo, 23 April 2002. Pendidikan dasar diperoleh melalui TK Taman Indria II dan dilanjutkan di SDN Jati 2 yang lulus di tahun 2014. Pada tahun yang sama, peneliti melanjutkan di SMPN 3 Probolinggo hingga tahun 2017. Selanjutnya peneliti melanjutkan di SMAN 4 Probolinggo hingga tahun 2020. Setelah lulus dari SMAN, peneliti melanjutkan pendidikan di Program Studi Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Selama menempuh Pendidikan di Kampus Ulul Albab UIN Maulana Malik Ibrahim Malang, selain menyelesaikan tugasnya sebagai mahasiswa, peneliti juga memiliki aktivitas yakni menjadi panitia yang diselenggarakan di kampus. Menurutnya dengan kegiatan tersebut, dapat menambah pengalaman dan relasi. Baginya, Matematika adalah salah satu ilmu yang selalu dibutuhkan dalam beberapa kasus pada kehidupan sehari-hari.

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 : Soviana
NIM : 200601110111
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Kode Golay Menggunakan Kriptosistem *McEliece* dalam Mengamankan Pesan
Pembimbing I : Prof. Dr. H. Turmudi, M.Si., Ph.D.
Pembimbing II : Dr. Abdussakir, M.Pd.

No	Tanggal	Hal	Tanda Tangan
1.	11 Oktober 2023	Konsultasi Topik dan Data	1.
2.	18 Oktober 2023	Konsultasi Bab I, II, dan III	2.
3.	17 November 2023	Konsultasi Kajian Agama Bab I dan II	3.
4.	23 November 2023	Konsultasi Revisi Kajian Agama Bab I dan II	4.
5.	18 Desember 2023	ACC Bab I, II, dan III	5.
6.	21 Desember 2023	ACC Kajian Agama Bab I dan II	6.
7.	19 Januari 2024	ACC Seminar Proposal	7.
8.	25 Maret 2024	Konsultasi Revisi Seminar Proposal	8.
9.	22 April 2024	Konsultasi Bab IV dan V	9.
10.	21 Mei 2024	ACC Bab IV dan V	10.
11.	27 Mei 2024	Konsultasi Kajian Agama Bab IV	11.
12.	28 Mei 2024	ACC Kajian Agama Bab IV	12.
13.	7 Juni 2024	ACC Seminar Hasil	13.
14.	19 Juni 2024	Konsultasi Revisi Seminar Hasil	14.
15.	20 Juni 2024	ACC Matriks Revisi 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.	20 Juni 2024	ACC Sidang Skripsi	16.
17.	26 Juni 2024	ACC Akhir Keseluruhan	17.

Malang, 26 Juni 2024
Mengetahui,
Ketua Program Studi Matematika

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