IMPLEMENTASI KRIPTOSISTEM *MCELIECE* MENGGUNAKAN KODE *HAMMING* DAN KODE REPETISI

SKRIPSI

OLEH
MOCH YUSLIH ABIDIN
NIM. 19610069

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

IMPLEMENTASI KRIPTOSISTEM *MCELIECE* MENGGUNAKAN KODE *HAMMING* DAN KODE REPETISI

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
Moch Yuslih Abidin
NIM. 19610069

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

IMPLEMENTASI KRIPTOSISTEM *MCELIECE* MENGGUNAKAN KODE *HAMMING* DAN KODE REPETISI

SKRIPSI

Oleh
Moch Yuslih Abidin
NIM. 19610069

Telah Disetujui Untuk Diuji

Malang, 12 Desember 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. Fachrur Rozi, M.Si.
NIP. 19800527 200801 1 012

IMPLEMENTASI KRIPTOSISTEM *MCELIECE* MENGGUNAKAN KODE *HAMMING* DAN KODE REPETISI

SKRIPSI

Oleh
Moch Yuslih Abidin
NIM. 19610069

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

Tanggal 23 Desember 2025

Ketua Penguji	: Hisyam Fahmi, M.Kom.	.....
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. Fachrur Rozi, M.Si.
NIP. 19800527 200801 1 012

PERSYARATAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	:	Moch Yuslih Abidin
NIM	:	19610069
Program Studi	:	Matematika
Fakultas	:	Sains dan Teknologi
Judul Skripsi	:	Implementasi Kriptosistem Mc Eliece Menggunakan Kode Hamming dan Kode Repetisi

Menyatakan dengan ini sebenarnya 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, 23 Desember 2025
Yang membuat pernyataan,

Moch Yuslih Abidin
NIM. 19600169

MOTTO

"The scariest moment is always just before you start."

PERSEMBAHAN

Dengan penuh rasa syukur ke hadirat Allah SWT atas segala rahmat, pertolongan dan kemudahan-Nya, sehingga skripsi ini dapat terselesaikan dengan baik, Karya ini penulis persembahkan kepada:

Bapak, Ibu, dan Adik, yang kasih sayangnya tak pernah berkurang, yang doanya menjadi kekuatan terbesar dalam setiap setiap langkah, dan yang pengorbanannya takkan pernah mampu penulis balas dengan apa pun di dunia ini.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Penulis menyampaikan ucapan terima kasih kepada semua pihak yang membantu dan mempermudah dalam penyususnan skripsi ini, baik secara langsung maupun tidak langsung. Ucapan terima kasih disampaikan tekhusus kepada:

1. Prof. Dr. Hj. Ilfi Nurdiana, M.Si., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Si., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Fachrur Rozi, M.Si., selaku Ketua Program Studi Matematika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Muhammad Khudzaifah, M.Si., selaku Dosen Pembimbing I yang telah memberi bimbingan dan arahan dengan penuh kesabaran dan ketelitian kepada penulis.
5. Erna Herawati, M.Pd., selaku Dosen Pembimbing II yang telah memberi bimbingan dan arahan dengan penuh kesabaran dan ketelitian.
6. Hisyam Fahmi, M.Kom., selaku Ketua Penguji yang telah memberikan kritik serta saran kepada penulis.
7. Mohammad Nafie Jauhari, M.Si., selaku Anggota Penguji 1 yang telah memberikan kritik serta saran kepada penulis.
8. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Tekonologi Universitas Islam Maulana Malik Ibrahim Malang.
9. Kedua orang tua dan seluruh keluarga yang selalu memberikan dukungan baik dalam bentuk material, moral maupun batin sehingga penulis dapat menyelesaikan hingga akhir.
10. Seluruh teman-teman mahasiswa Angkatan 2019 khususnya Program Studi Matematika yang telah memberikan bantuan dan dukungan dalam keadaan apapun.

Semoga Allah SWT senantiasa melimpahkan pahala yang terbaik atas segala kebaikan yang telah diberikan kepada penulis. Harapan penulis agar penelitian ini dapat bermanfaat bagi pembaca maupun penulis sendiri untuk

meningkatkan ilmu pengetahuan serta pemahaman ilmu matematika, utamanya pada bidang statistika.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 23 Desember 2025

Moch Yuslih Abidin

DAFTAR ISI

DAFTAR TABEL

DAFTAR LAMPIRAN

ABSTRAK

Abidin, Moch Yuslih. 2025. **Implementasi Kriptosistem *McEliece* Menggunakan Kode Hamming dan Kode Repetisi**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim. Pembimbing: (I) Muhammad Khudzaifah, M.Si. (II) Erna Herawati, M.Pd.

Kata Kunci: Kriptosistem *McEliece*, Kode Hamming, Kode Repetisi, Post-Quantum Kriptografi

Keamanan data merupakan isu krusial dalam era digital karena meningkatnya kerugian akibat kejahatan siber. Penelitian ini menerapkan kriptosistem *McEliece* yang diintegrasikan dengan kode Hamming dan kode Repetisi untuk meningkatkan keamanan dan keandalan komunikasi data. Algoritma *McEliece* dipilih karena efisiensi dan ketahanannya terhadap serangan komputer kuantum, menjadikannya kandidat kuat dalam kriptografi pasca-kuantum. Data yang digunakan dalam simulasi ini adalah pesan teks yang dikonversi menjadi biner berdasarkan tabel ASCII. Proses pembentukan kunci melibatkan matriks generator G, matriks non-singular S, dan matriks permutasi P untuk menghasilkan kunci publik $G' = S \cdot G \cdot P$. Proses enkripsi dilakukan dengan merepetisi bit pesan sebanyak tiga kali sebelum dikalikan dengan matriks publik dan ditambahkan vektor *error* acak pada tahap dekripsi, penerima menggunakan kunci privat untuk melakukan koreksi kesalahan melalui perhitungan nilai *syndrome* menggunakan matriks *parity check*. Hasil analisis simulasi menunjukkan bahwa sistem berhasil mengembalikan pesan asli secara benar. Penelitian ini menyimpulkan bahwa penggunaan kode repetisi dengan jumlah repetisi yang lebih banyak akan meningkatkan kemampuan koreksi kesalahan, serta menunjukkan bahwa implementasi kriptografi ini selaras dengan nilai Islami dalam menjaga ''amanah'' dan kerahasiaan informasi.

ABSTRACT

Abidin, Moch Yuslih. 2025. **Implementation of McEliece Cryptosystem Using Hamming Code and Repetition Code.** Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Muhammad Khudzaifah, M.Si. (II) Erna Herawati, M.Pd.

Keywords: McEliece Cryptosystem, Hamming Code, Repetition Code, Post-Quantum Cryptography.

Data security is a crucial issue in the digital era due to increasing losses caused by cybercrime. This research implements the McEliece cryptosystem integrated with Hamming code and repetition code to enhance the security and reliability of data communication. The McEliece algorithm was chosen for its efficiency and resistance to quantum computer attacks, making it a strong candidate in post-quantum cryptography. The data used in this simulation is text messages converted into binary based on the ASCII table. The key generation process involves a generator matrix G, a non-singular matrix S, and a permutation matrix P to produce the public key $G' = S \cdot G \cdot P$. The encryption process is carried out by repeating the message bits three times before multiplying them by the public matrix and adding a random error vector. In the decryption stage, the recipient uses the private key to perform error correction through syndrome value calculations using a parity check matrix. Simulation results show that the system successfully restores the original message correctly. This research concludes that using a higher number of repetitions will increase the error correction capability and demonstrates that the implementation of this cryptography aligns with Islamic values in maintaining ”amanah” (trust) and information confidentiality.

مستخلص البحث

عابدين، محمد يصلح .٢٠٢٥ .**تطبيق نظام التشفير ماك إليس (*McEliece*) باستخدام كود هامينج (*Hamming*)وكود التكرار** .(***Repetition***) بحث جامعي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (١) محمد حذيفة، الماجستير. (٢) إيرنا هيراواتي، الماجستير.

الكلمات الرائيسية :نظام التشفير ماك إليس؛ كود هامينج؛ كود التكرار؛ التشفير ما بعد الكم.

تعد أمن البيانات قضية ملحة في العصر الرقمي بسبب تزايد الخسائر الناجمة عن الجرائم السيبرانية .يطبق هذا البحث نظام التشفير ماك إليس المتكامل مع كود هامينج وكود التكرار لتعزيز أمن وموثوقية اتصالات البيانات .تم اختيار خوارزمية ماك إليس لكفاءتها ومقاومتها لهجمات الكمبيوتر الكمي، مما يجعلها مرشحًا قويًا في مجال التشفير ما بعد الكم .البيانات المستخدمة في هذه المحاكاة هي رسائل نصية تم تحويلها إلى ثنائي (*binary*) بناءً على جدول *ASCII*. تتضمن عملية تكوين المفتاح مصفوفة المولد G ، والمصفوفة غير المفردة S ، ومصفوف التبديل P لإنتاج المفتاح العام $G' = S \cdot G \cdot P$.تتم عملية التشفير عن طريق تكرار بتات الرسالة ثلاث مرات قبل ضربها في المصفوفة العامة وإضافة ناقل خطأ عشوائي . في مرحلة فك التشفير، يستخدم المتلقي المفتاح الخاص لتصحيح الأخطاء من خلال حسابات قيمة المتلازمة (*syndrome*)باستخدام مصفوفة التحقق من التكافؤ .أظهرت نتائج المحاكاة أن النظام نجح في استعادة الرسالة الأصلية بدقة .يخلص هذا البحث إلى أن استخدام عدد أكبر من التكرارات سيزيد من قدرة تصحيح الخطأ، كما يوضح أن تطبيق هذا التشفير يتماشى مع القيم الإسلامية في الحفاظ على الأمانة وسرية المعلومات.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini keamanan data menjadi isu yang sangat penting, terutama dengan meningkatnya jumlah transaksi *online* dan pertukaran informasi sensitif. Menurut laporan dari Cybersecurity Ventures, kerugian global akibat kejahatan siber diperkirakan mencapai 6 triliun dollar pada tahun 2021 dan diprediksi akan meningkat setiap tahunnya (Prayoga dkk., 2022). Dengan demikian, solusi untuk mengamankan data dan informasi menjadi sangat mendesak. Kriptografi, sebagai salah satu metode untuk melindungi data, menawarkan berbagai teknik untuk memastikan kerahasiaan, integritas, dan keaslian informasi yang dikirimkan melalui jaringan.

Kriptografi adalah ilmu yang mempelajari teknik untuk mengamankan komunikasi dan informasi dengan cara mengubah data menjadi format yang tidak dapat dibaca oleh pihak yang tidak berwenang. Secara umum, kriptografi dibagi menjadi dua kategori utama: kriptografi simetris dan asimetris. Kriptografi simetris menggunakan kunci yang sama untuk enkripsi dan dekripsi, sedangkan kriptografi asimetris menggunakan pasangan kunci publik dan privat (Arif & Nurokhman, 2023). Dengan kemajuan teknologi, berbagai algoritma kriptografi telah dikembangkan untuk meningkatkan keamanan data, termasuk *Advanced Encryption Standard* (AES) dan Rivest-Shamir-Adleman (RSA) (Hutagalung dkk., 2023).

Dalam konteks kriptografi, terdapat berbagai kriptosistem yang dirancang untuk memenuhi kebutuhan spesifik dalam pengamanan data. Salah satu kriptosistem yang menarik perhatian adalah *McEliece*, yang menggunakan kode-kode *error-correcting* untuk menyediakan keamanan yang tinggi. *McEliece* dikenal karena kemampuannya untuk memberikan keamanan yang kuat meskipun dengan kunci yang relatif besar, menjadikannya pilihan yang menarik dalam aplikasi yang memerlukan tingkat keamanan tinggi (Maulyanda dkk., 2021). Selain itu, penelitian mengenai *McEliece* menunjukkan potensi untuk diintegrasikan dengan berbagai teknik kriptografi lainnya, seperti kode *Hamming* dan kode repetisi, untuk meningkatkan efisiensi dan efektivitas dalam pengamanan data.

Integrasi antara teknologi kriptografi dan nilai-nilai spiritual atau kemanusiaan juga patut dicermati. Dalam konteks ini, beberapa penelitian menunjukkan bahwa penerapan teknologi, termasuk kriptografi, dapat mendukung prinsip-prinsip etika dan keadilan dalam masyarakat (Andalan, 2019). Misalnya, penggunaan tanda tangan digital yang aman dapat membantu dalam memastikan keaslian dokumen dan transaksi, yang sejalan dengan nilai-nilai kejujuran dan transparansi yang diajarkan dalam banyak tradisi agama, termasuk Islam. Hal ini juga termasuk sifat amanah yang dianjurkan oleh Allah SWT dalam firman-Nya dalam surat Al-Mu'minun ayat 8 yang artinya (Kemenag, 2024):

"(Sungguh beruntung pula) orang-orang yang memelihara amanat dan janji mereka"

Kode repetisi adalah salah satu teknik pengkodean yang digunakan dalam kriptografi untuk meningkatkan keandalan komunikasi data. Metode ini bekerja dengan cara mengulangi pesan yang sama beberapa kali sebelum dikirimkan. Misalnya, jika pesan asli adalah "A", maka dengan menggunakan kode repetisi,

pesan yang dikirimkan bisa menjadi "AAA". Dengan cara ini, jika terjadi kesalahan dalam transmisi, penerima masih memiliki peluang untuk mendapatkan pesan yang benar, asalkan sebagian besar dari pesan yang diterima adalah benar. Keuntungan utama dari kode repetisi adalah kesederhanaannya dan kemampuannya untuk memberikan perlindungan terhadap kesalahan yang terjadi selama transmisi. Namun, kelemahan dari metode ini adalah efisiensi *bandwidth* yang rendah, karena pengulangan pesan dapat menyebabkan penggunaan *bandwidth* yang lebih besar. Meskipun demikian, kode repetisi tetap relevan dalam konteks kriptografi, terutama dalam situasi di mana keandalan komunikasi sangat penting. Penelitian lebih lanjut tentang penerapan kode repetisi dalam kriptosistem yang lebih kompleks dapat membuka jalan untuk solusi yang lebih efisien dan aman dalam pengiriman data (Stallings, 2017).

Pada penelitian (Sinaga dkk., 2024) dijelaskan bahwa kriptosistem *McEliece* merupakan algoritma yang dikenal karena efisiensi dan ketahanannya terhadap serangan komputer kuantum. Mekanismenya bekerja dengan membentuk kunci publik dari kombinasi matriks generator dan matrik permutasi, sedangkan kunci privat terdiri dari informasi yang diperlukan untuk melakukan dekripsi terhadap pesan yang telah terenkripsi. Keunggulan utama dari kriptosistem *McEliece* terletak pada ketahanannya terhadap algoritma XOR, yang mampu memecahkan sistem kriptografi berbasis faktorisasi bilangan prima dengan efisiensi tinggi. Oleh karena itu, *McEliece* dipandang sebagai salah satu kandidat paling menjanjikan dalam pengembangan kriptografi pasca kuantum.

Penelitian sebelumnya di Indonesia telah menunjukkan berbagai pendekatan dalam penerapan kriptografi, namun sering kali masih terdapat kekurangan dalam

hal metode dan populasi yang diteliti. Sebagian besar penelitian berfokus pada algoritma kriptografi tertentu tanpa mempertimbangkan integrasi dengan sistem lain atau aplikasi praktis di dunia nyata (Hidayatuloh dkk., 2021). Selain itu, banyak penelitian yang belum mengeksplorasi potensi penggunaan kode *Hamming* dan kode repetisi dalam konteks kriptosistem *McEliece*, yang dapat menjadi celah penelitian yang signifikan.

Kode repetisi, sebagai salah satu metode pengkodean, menawarkan cara untuk meningkatkan keandalan komunikasi dengan mengulangi pesan yang sama beberapa kali. Ini dapat memberikan perlindungan tambahan terhadap kesalahan yang mungkin terjadi selama transmisi data (Maulyanda dkk., 2021). Penelitian lebih lanjut dalam penggunaan kode ini dalam konteks kriptosistem *McEliece* dapat memberikan wawasan baru tentang bagaimana mengoptimalkan keamanan dan efisiensi dalam pengiriman data.

Berdasarkan latar belakang di atas, maka penulis melakukan penelitian dengan judul "*Implementasi Kriptosistem McEliece Menggunakan Kode Hamming dan Kode Repetisi*". Pengintegrasian *McEliece* dengan kode *Hamming* dan kode repetisi menjadi fokus utama pada penelitian ini yang bertujuan untuk mengembangkan metode baru yang tidak hanya meningkatkan keamanan data tetapi juga efisiensi dalam proses enkripsi dan dekripsi. Penelitian ini diharapkan dapat memberikan kontribusi signifikan bagi pengembangan ilmu pengetahuan, serta memberikan manfaat praktis bagi sektor teknologi informasi dan diharapkan dapat menghasilkan wawasan baru tentang *Post-Quantum* Kriptografi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah pada skripsi ini dapat dirumuskan sebagai berikut :

1. Bagaimana proses simulasi enkripsi pada algoritma *McEliece* dengan kode *Hamming* dan kode Repetisi ?
2. Bagaimana proses simulasi dekripsi pada algoritma *McEliece* dengan kode *Hamming* dan kode Repetisi ?
3. Bagaimana pengaruh banyak repetisi dalam koreksi pada algoritma *McEliece* dengan kode *Hamming* ?

1.3 Tujuan Penelitian

Berdasarkan latar belakang yang telah dijelaskan, tujuan peneliti adalah sebagai berikut

1. Untuk menjelaskan hasil simulasi enkripsi algortima *McEliece* dengan kode *Hamming* dan kode Repetisi.
2. Untuk menjelaskan hasil simulasi dekripsi algoritma *McEliece* degan kode *Hamming* dan kode Repetisi.
3. Untuk menganalisis hasil simulasi pengaruh banyak repetisi dalam koreksi pada algoritma *McEliece* dengan kode *Hamming*.

1.4 Manfaat Penelitian

Berdasar pada tujuan penelitian maka diharapkan memiliki beberapa manfaat sebagai berikut :

1. Bagi Penulis

   Penelitian ini diharapkan dapat memberikan kapahaman tentang cara mengamankan pesan menggunakan algoritma *McEliece* dengan kode *Hamming* dan kode Repetisi.

2. Bagi Pembaca

   Penelitian ini diharapkan dapat menjadi bahan referensi atau wawasan baru terkait kriptografi *post-quantum* , terutama dalam penerapan algoritma kriptosistem *McEliece* dengan kode *Hamming* dan kode Repetisi dan tingkat keamanannya.

1.5 Batasan Masalah

Batasan masalah yang terdapat pada penelitian ini adalah konversi karakter biner dengan tabel ASCII.

1.6 Definisi Istilah

Penggunaan istilah dalam penulisan ini :

1. ***Ciphertext*** adalah teks hasil enkripsi yang tidak dapat dibaca tanpa proses dekripsi menggunakan kunci yang sesuai.
2. ***Plainntext*** adalah teks asli atau pesan sebelum mengalami proses enkripsi.
3. **Enkripsi** adalah proses mengubah plainteks menjadi cipherteks menggunakan algortima dan kunci tertentu untuk menjaga kerahasiaan informasi.
4. **Dekripsi** adalah proses mengubah cipherteks kembali menjadi plainteks menggunakan kunci yang sesuai.

5. **Kriptosistem *McEliece*** adalah sistem kriptografi berbasis kode linear yang menggunakan kode koreksi kesalahan sebagai dasar keamanannya.
6. **Kode *Hamming*** adalah kode koreksi kesalahan yang dapat mendeteksi dan memperbaiki satu kesalahan bit dalam blok data tertentu.
7. **Kode Repetisi** adalah teknik pengkodean yang mengulangi setiap bit pesan beberapa kali untuk meningkatkan keandalan dalam mendeteksi dan memperbaiki kesalahan transmisi.
8. **Kunci Publik** adalah kunci yang digunakan dalam skema kriptografi asimetris untuk enkripsi atau verifikasi tanda tangan digital dan dapat dibagikan secar terbuka
9. **Kunci Privat** adalah kunci rahasia dalam kriptografi asimetris yang digunakan untuk dekripsi atau pembuatan tanda tangan digitak dan hasrus dijaga kerahasiaannya.
10. ***Post-quantum*** adalah bidang kriptografi yang meneliti dan mengembangkan algoritman yang dapat bertahan terhadap kemampuan komputasi kuantum.

BAB II

KAJIAN TEORI

2.1 Keterbagian

Definisi 2.1

Misalkan a dan b adalah dua buah bilangan bulat dengan syarat $a \neq 0$. Kita menyatakan bahwa a habis membagi b (a *divides* b) jika terdapat bilangan bulat c sedemikian sehingga $b = ac$ (Munir, 2019).

Contoh 2.1

1. $9|72$ karena $72 \div 9 = 8$ (bilangan bulat) atau $72 = 9 \times 8$.
2. $4 \nmid 10$ karena $10 \div 4 = 2.5$ (bukan bilangan bulat). Jadi, 10 tidak habis dibagi 4.

Notasi : $a|b$ jika $b = ac, c \in Z$ dan $a \neq 0$.

2.2 Bilangan Prima

Definisi 2.2

Bilangan prima adalah bilangan bulat positif yang lebih besar dari 1 dan yang tidak dapat dibagi oleh bilangan bulat positif lain selain 1 dan dirinya sendiri (Rosen, 1987).

Contoh 2.1

1. Bilangan 13 adalah bilangan prima karena hanya dapat dibagi oleh 1 dan 13
2. Bilangan 103 adalah bilangan prima karena hanya dapat dibagi oleh 1 dan 103

3. Bilangan 1033 adalah bilangan prima karena hanya dapat dibagi oleh 1 dan 1033

2.3 Faktor Perseketuan Tersebar (FPB)

Definisi 2.3

Jika $a, b \in \mathbb{Z}$, yang keduanya bukan 0, maka faktor persekutuan terbesar dari a dan b adalah p yang merupakan bilangan bulat positif terbesar yang habis membagi a dan b (Handayani & Yulina, 2020).

Contoh 2.3

1. FPB dari 12 dan 15

   Faktor-faktor positif dari 12 adalah $\{1, 2, 3, 4, 6, 12\}$

   Faktor-faktor positif dari 15 adalah $\{1, 3, 5, 15\}$

   Faktor persekutuannya adalah $\{1, 3\}$

   Jadi FPB $(12,15) = 3$

2. FPB dari 45 dan 60

   Faktor-faktor positif dari 45 adalah $\{1, 3, 5, 9, 15, 45\}$

   Faktor-faktor positif dari 60 adalah $\{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$

   Faktor persekutuannya adalah $\{1, 3, 5, 15\}$

   Jadi FPB $(45,60) = 15$

2.4 Relatif Prima

Definisi 2.4

Dua bilangan bulat a dan b yang keduanya tidak 0, disebut relatif prima jika *GCD* (*Greatest Common Divisor*) atau FPB dari $(a, b) = 1$ (Burton, 2011).

Contoh 2.4

Faktor-faktor positif dari 8 adalah $\{1, 2, 4, 8\}$

Faktor-faktor positif dari 15 adalah $\{1, 3, 5, 15\}$

Jadi FPB $(8,15) = 1$

2.5 Aritmatika Modulo

Misalkan a adalah sebuah bilangan bulat dan m merupakan bilangan bulat positif. Operasi $a \ mod \ m$, didefinisikan sebagai sisa hasil pembagian bilangan a dengan m. Hasil operasi modulo selalu berada pada himpunan $\{0, 1, 2, \ \ldots, m - 1\}$(Munir, 2019).

Dengan kata lain, jika $a \ mod \ m = r$, maka a dapat dituliskan dalam bentuk:

$$a = mq + r, \qquad 0 \leq r < m$$

Contoh 2.5

1. $35 \ mod \ 6 = 5$ karena $35 = 6.5 + 5$
2. $50 \ mod \ 7 = 1$ karena $50 = 7.7 + 1$
3. $19 \ mod \ 4 = 3$ karena $19 = 4.4 + 3$

2.6 Kongruen

Definisi 2.5

Jika a dan b adalah bilangan bulat, kita katakan bahwa a kongruen dengan b modulo m jika $m|(a - b)$. Ditulis :

$$a \equiv b \ (\text{mod} \ m)$$

Jika tidak, maka ditulis $a \not\equiv b \ (\text{mod} \ m)$ (Rosen, 1987).

Contoh 2.6

1. $38 \equiv 14 \pmod{12}$, karena $38 - 14 = 24$ habis dibagi 12
2. $9 \not\equiv 4 \pmod{6}$, karena $9 - 4 = 5$ tidak habis dibagi 6

Teorema 2.1

Jika a, b, c dan m adalah bilangan bulat dengan $m > 0$ sehingga $c \equiv d \pmod{m}$, maka berlaku :

$a + c \equiv b + d \pmod{m}$

$a - c \equiv b - d \pmod{m}$

$a.c \equiv b.d \pmod{m}$(Rosen, 1987)

2.7 Fungsi Totient Euler

Definisi 2.6

Fungsi *totient* Euler dilambangkan dengan $\phi(n)$, didefinisikan untuk setiap bilangan bulat $n \geq 1$ yang nilainya menunjukkan banyaknya bilangan bulat positif yang lebih kecil dari n dan relatif prima terhadap n (Munir, 2019).

Contoh 2.7

1. $\phi(12) = 4$, karena bilangan yang relatif prima dan bilangan bulat lebih kecil dari 12 yaitu $1, 5, 7, 11$
2. $\phi(15) = 8$, karena bilangan yang relatif prima dan bilangan bulat lebih kecil dari 15 yaitu $1, 2, 4, 7, 8, 11, 13, 14$

2.8 Lapangan

2.8.1 Lapangan (Field)

Lapangan f, yang biasa dinotasikan $\{F, +, \times\}$, adalah sekumpulan elemen dengan dua operasi biner, yang disebut penjumlahan dan perkalian, sedimikian hingga $\forall\, a, b, c$ dalam F memenuhi aksioma tersebut (Stallings, 2017).

Secara sederhana lapangan adalah tempat dimana kita dapat melakukan operasi seperti penjumlahan dan perkalian. Pengurangan $a - b$ dipandang sebagai $a + (-b)$ yang dalam hal ini $(-b)$ adalah elemen balikan penjumlahan dari b, sedangkan pembagian $\frac{a}{b}$ dipandang sebagai $a \times b^{-1}$ yang dalam hal ini b^{-1} adalah elemn balikan perkaloan dari b (Munir, 2019).

2.8.2 Lapangan Hingga $GF(p)$

Jika p adalah bilangan prima, maka lapangan hingga berorde p, dilambangkan $GF(p)$, didefiisikan sebagai himpunan bilangan bulat $\{0, 1, 2, \ldots, p-1\}$ dengan operasi aritmetika dilakukan modulo p (Stallings, 2017). Jadi, penjumlahan dan perkalian dalam $GF(p)$ mengikuti aturan aritmatika modular.

Contoh 2.8

1. Lapangan $GF(2)$:

   Himpunannya adalah $\{0, 1\}$

   Penjumlahan $(mod\ 2)$:

   a. $0 + 0 = 0$

   b. $0 + 1 = 1$

   c. $1 + 0 = 1$

d. $1 + 1 = 0$

Perkalian $(mod\ 2)$:

e. $0 \,.\, 0 = 0$

f. $0 \,.\, 1 = 0$

g. $1 \,.\, 0 = 0$

h. $1 \,.\, 1 = 1$

2.8.3 Ruang Vektor atas Lapangan Hingga

Misalkan F_q adalah suatu lapangan hingga dengan orde q. Suatu himpunan tak kosong V dengan operasi penjumlahan vektor dan perkalian skalar oleh elemen-elemen dari F_q, disebut ruang vektor atas F_q apabila memenuhi sifat-sifat berikut (Ling & Xing, 2004). Untuk semua $u, v, w \in V$ dan untuk semua $\lambda, \mu \in F_q$:

1. $u + v \in V$;
2. $(u + v) + w = u + (v + w)$;
3. Terdapat elemen $0 \in V$ yang memenuhi $0 + v = v = v + 0$ untuk semua $v \in V$;
4. Untuk setiap $u \in V$ terdapat elemen dari V, disebut $-u$, sehingga $u + (-u) = 0 = (-u) + u$;
5. $u + v = v + u$;
6. $\lambda v \in V$;
7. $\lambda(u + v) = \lambda u + \lambda v$, $(\lambda + \mu)u = \lambda u + \mu u$;
8. $(\lambda\mu)u = \lambda(\mu u)$;
9. Jika 1 adalah identitas perkalian dari F_q, maka $1u = u$.

2.9 Teori Pengkodean

2.9.1 Kode Linier

Definisi 2.6

Kode linier dengan panjang n atas $\mathbb{F}_q$ adalah subruang dari ruang vektor $\mathbb{F}_q^n$. Oleh karena itu himpunan C dari $\mathbb{F}_q^n$ adalah kode linier jika dan hanya jika

1. $u + v \in C$ untuk setiap u, v dalam C, dan
2. $au \in C$ untuk setiap $u \in C$ dan $a \in \mathbb{F}_q$

Kode linier banyak digunakan dalam praktik karena sejumlah alasan. Salah satu alasannya adalah karena mereka mudah ditemukan. Alasan lainnya adalah kode linier sangat cepat dan mudah. Dekoding juga sering dipermudah oleh linearitas kode.

2.9.2 Matriks Generator dan Matriks Paritas

Karena kode linier adalah ruang vektor, sebuah kode linier $[n, k]$ dapat ditentukan dengan memberikan basis dari k kata *codeword*.

Definisi 2.7

Sebuah matriks G dengan ukuran $k \times n$ yang baris barisnya membentuk basis untuk kode linier $[n, k]$ disebut matrik generator dari kode tersebut.(Birken, 2021)

Yang sebenarnya, matriks G adalah matriks generator untuk beberapa kode linier C jika dan hanya jika baris-baris G adalah linier independen.

Definisi 2.8

Misalkan C adalah kode siklik $[n, k]$. Matriks paritas untuk C adalah matriks $(n - k) \times n$ matriks H sedemikian rupa sehingga $c \in C$ jika dan hanya jika $cH^T =$

0. Dengan kata lain, matriks paritas untuk C merupakan matriks generator untuk $C^{\perp}$.(Birken, 2021)

2.10 Kode *Hamming*

Kode *Hamming* adalah salah satu jenis kode deteksi dan koreksi kesalahan yang diperkenalkan oleh Richard W. Hamming pada tahun 1950. Kode ini bekerja dengan menambahkan bit-bit paritas tertentu pada data sehingga sistem dapat mendeteksi bahkan mengoreksi kesalahan yang terjadi pada saat transmisi data.

Definisi 2.9

Misalkan $r \geq 2$. Sebuah kode linier biner dengan panjang $n = 2^r - 1$, dengan matriks paritas H yang kolom-kolomnya terdiri dari semua vektor tidak nol dari $\mathbb{F}_2^r$ disebut sebagai kode *hamming* biner dengan panjang $2^r - 1$ dilambangkan dengan Ham$(r, 2)$(Ling & Xing, 2004).

Contoh

Ham$(3,2)$: kode hammming panjang 7 dengan matriks *parity check*

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Definition 2.10

Misalkan x dan y adalah kata (string) dengan panjang n yang tersusun dari suatu alfabet A. Jarak *Hamming* antara x ke y yang dilambangkan dengan $d(x, y)$ didefinisikan sebagai jumlah posisi di mana x dan y berbeda. Jika $x = x_1 \cdots x_n$ dan $y = y_1 \cdots y_n$, maka

$$d(x, y) = d(x_1, y_1) + \cdots + d(x_n, y_n)$$

Di mana x_i dan y_1 dianggap sebagai kata dengan panjang 1 dan

$$d(x_i, y_i) = \begin{cases} 1 \text{ jika } x_i \neq y_i \\ 0 \text{jika } x_i = y_i \end{cases}$$

Contoh

Misalkan A {0,1} dan misalkan $x = 00100$, $y = 00011$, dan $z = 10000$

Maka,

$$d(x, y) = 3$$

$$d(x, z) = 2$$

Definisi 2.11

Misalkan x adalah sebuah kata dalam $\mathbb{F}_q^n$. Bobot *hamming* dari x, yang dinyatakan dengan $wt(x)$, didefinisikan sebagai jumlah titik koordinat yang tidak nol dalam x yaitu

$$wt(x) = d(x, 0)$$

Untuk setiap x dari F_q, bobot *hamming* dari x, adalah sebagai berikut :

$$wt(x) = d(x, 0) = \begin{cases} 1 \text{ jika } x \neq 0 \\ 0 \text{ jika } x = 0 \end{cases}$$

2.11 Kode Repetisi

Kode pengulangan (kode repetisi) adalah salah satu kode linier paling sederhana dalam teori pengkodean. Prinsipinya, setiap bit pesan diulang beberapa kali agar menambah redundansi pada bit pesan untuk mempersulit celah para pembobol data yang dapat merugikan (Ling & Xing, 2004).

2.12 Kriptografi

Kriptografi berasal dari bahasa Yunani, yaitu *cryptos* yang berarti “rahasia” dan *graphein* yang berarti “menulis”. Secara harfiah, kriptografi dapat dimaknai

sebagai “tulisan rahasia”. Dalam literatur, terdapat berbagai definisi kriptografi yang berkembang dari masa ke masa. Salah satu definisi klasik menyebutkan bahwa kriptografi merupakan ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikan pesan tersebut ke dalam bentuk yang sulit untuk dipahami maknanya.

Pada awalnya, kriptografi terutama digunakan untuk menjaga keamanan komunikasi, misalnya di bidang militer, diplomasi, maupun kegiatan intelejen. Akan tetapi, dalam perkembangannya kriptografi tidak hanya digunakan untuk menjaga kerahasiaan (*privacy*), melainkan juga memiliki tujuan lain, seperti menjaga integritas data (*data integrity*), memastikan keaslian informasi (*authentication*), serta mencegah penyangkalan pihak tertentu terhadap pesan yang telah dikirim (anti-penyangkalan/*non-repudiation*) (Munir, 2019).

2.13 Kriptosistem *McEliece*

Kriptosistem *McEliece* diusulkan oleh *McEliece* pada tahun 1978 dan versi aslinya yaitu kode Goppa, tetap tidak dapat dipecahkan. Komputer kuantum tampaknya tidak memberikan peningkatan signifikan dalam menyerang sistem berbasis kode, melebihi peneingkatan generik yang mungkin dengan algoritma Grover, sehingga skema enkripsi *McEliece* adalah salah satu kandidat menarik untuk kriptografi pasca-kuantum. Kekurangan dari sistem ini adalah ukuran kunci yang cukup besar untuk menyembunyikan kode Goppa yang terstruktur dengan baik dan dapat di decode dengan efisien dalam kunci publik, matriks generator penuh dari kode bolak-balik perlu diterbitkan. Berbagai upaya untuk mengurangi ukuran kunci telah menggunakan kode lain, terutama kode di atas bidang yang lebih besar alih alih kode subfield.

Secara umum kriptosistem *McEliece* dibangun berdasarkan teori pengkodean dan memiliki tiga komponen utama, yaitu matrik generator G, matriks non-singular S, dan matriks permutasi P. Matriks generator G berasal dari kode linier yang memiliki kemampuan *error correction*, sedangkan matriks S digunakan untuk mengacak basis yang membuat hubungan antara pesan dan *codeword* tidak langung terlihat, dan matriks permutasi P berfungsi sebagai pengubah posisi bit hasil enkripsi agar pola kodenya sulit dikenali. Ketiga matriks ini dikombinasikan menjadi matriks publik G' melalui persamaan :

$$G' = \mathrm{S} \times \mathrm{G} \times \mathrm{P}$$

Dalam proses enkripsi, pesan awal (*plaintext*) terlebih dahulu dikonversi ke bentuk binerdan dibagi menjadi blok-blok dengan panjang tertentu sesuai parameter kode $[n, k]$. Setiap blok pesan m_i dikalikan dengan matriks publik G' untuk menghasilkan *codeword* C_i. Selanjutnya, ditambahkan vektor error acak e dengan panjang n dan bobot t untuk memperoleh *ciphertext* $C_i{}' = C_i + e$. Penambahan vektor error ini bertujuan untuk meningkatkan keamanan sistem, karena tanpa mengetahui kunci privat, pihak lain akan kesulitan mengidentifikasi pesan asli di antara berbagai kemungkinan hasil encoding.

Proses dekripsi dilakukan oleh penerima pesan dengan menggunakan kunci privat. Pertama, *chipertext* dikalikan dengan invers dari matriks permutasi, yaitu $Y = C_i{}' \times P^{-1}$ untuk mengembalikan urutan bit ke posisi semula. Setelah itu, dilakukan proses decoding untuk memperbaiki error menggunakan algoritma koreksi kesalahan dari kode yang digunakan, misalnya *Hamming code*. Tahap terakhir adalah mengalikan hasil decoding dengan invers dari matriks S untuk

mendapatkan kembali pesan asli yang kemudian dikonversi dari biner ke karakter semula.

Kelebihan utama dari kriptosistem *McEliece* adalah ketahanannya terhadap serangan kuantum serta efisiensinya dalam proses decoding. Mekanisme berbasis *error-correcting* membuat proses pemulihan pesan berlangsung cepat, sementara kompleksitas matematisnya sangat sulit dipecahkan tanpa kunci privat. Namun terdapat kelemahan utama sistem ini yang terletak pada ukuran kunci publik yang relatif besar, karena diperlukan publikasi seluruh matriks generator yang telah dipermutasi. Upaya untuk mengatasi kelemahan ini telah dilakukan melalui pengembangan varian *McEliece* yang menggunakan jenis kode berbeda seperti Reed-Solomon, BCH, Golay, LDPC, hingga *Hamming Code*.

2.14 Kajian Integrasi Topik dengan Al-Qur'an & Hadist

Al-Qur'an adalah wahyu Allah SWT yang diturunkan kepada Rasulullah SAW sebagai pedoman hidup bagi kaum muslimin. Kandungannya tidak terbatas pada ajaran akidah dan syariat saja, melainkan juga mencakup aspek-aspek yang berkaitan dengan perilaku manusia, baik berupa perintah maupun larangan. Dengan demikian, Al-Qur'an embahas persoalan muamalah hingga tata krama kehidupan sehari-hari. Salah satu diantaranya adalah tuntutan untuk bersikap amanah dan menegakkan kebenaran, baik dalam hubungan dengan Allah SWT maupun dalam interaksi dengan sesama manusia (Andika dkk., 2020).

Secara etimologis, amanah berarti jujur dan dapat dipercaya, baik dalam ucapan maupun perbuatan. Kata ini berasal dari bahasa Arab *amina amanatan* yang bermakna pesan atau perintah yang wajib disampaikan secara benar. Kejujuran

dalam hal ini mencakup kejujuran terhadap diri sendiri maupun terhadap orang lain, karena amanah merupakan bagian dari fitrah manusia. Dalam kaitannya dengan pesan, amanah berarti menyampaikan informasi sesuai dengan isi yang sebenarnya tanpa ditambah atau dikurangi. Secara terminologis, amanah dipahami sebagai segala sesuatu yang menjadi tanggung jawab seseorang, baik terkait hak Allah SWT maupun hak sesama manusia, meliputi benda, ucapan, perbuatan, maupun kepercayaan. Misalnya, dalam harta dan penghasilan seseorang terdapat hak fakir miskin, sehingga pemiliknya berkewajiban menunaikannya melalui zakat, infak, dan sedekah.

Menurut tafsir Al-Qurtubi, amanah dipahami sebagai tanggung jawab yang diemban oleh manusia, baik dalam urusan agama maupun dunia. Amanah meliputi ucapan maupun tindakan, yang harus dijaga serta direalisasikan dalam bentuk pelaksanaan yang benar. Pandangan ini menegaskan bahwa amanh bukan sekedar titipan yang bersifat material, tetapi juga mencakup dimensi spiritual dan sosial yang melekat dalam kehidupan manusia sehari-hari. Dengan demikian, amanah menuntut setiap individu untuk menjalankan kewajibannya dengan penuh kesadaran dan integritas.

Dalam tafsir *Al-Misbah*, Quraish shihab menjelaskan bahwa amanah berarti pemberian atau titipan dari orang lain yang wajib dijaga dengan baik dan dikembalikan tepat waktu ketika diminta kembali oleh pemiliknya. Definisi ni menyoroti aspek muamalah dalam kehidupan sehari-hari, di mana kejujuran dan kepercayaan menjadi pilar utama dalam menjalin hubungan sosial. Amanah dalam perspektif ini identik dengan keharusan menjaga hak orang lain agar tidak diselewengkan, baik dalam bentuk barang, informasi, maupun tanggung jawab

yang dititipkan. Hal ini menunjukkan bahwa amanah memiliki nilai praktis yang tinggi dalam menjaga harmoni sosial dan menumbuhkan rasa saling percaya di tengah masyarakat.

Rasyid Rida menambahkan bahwa amanah adalah bentuk kepercayaan yang diberikan kepeda seseorang, yang dengannya menimbulkan rasa tenang bagi pemberi amanah karena yakin titipannya akan dijaga. Pemahaman ini menekankan dimensi psikologis dari amanah, yakni menghadirkan rasa aman dan menghilangkan rasa takut maupun cemas. Dengan kata lain, amanah bukan hanya berkaitan dengan pemenuhan tanggung jawab secara lahiriah, tetapi juga dengan penciptaan rasa tentram dan keyakinan dalam hati. Jika amanah ditegakkan, maka tercipta lingkungan yang penuh dengan rasa saling percaya, sebaliknya jika amanah dikhianati, maka lahirlah kegelisahan, kecurigaan, dan rusaknya tatanan sosial.

Al-Qur'an menegaskan pentingnya sikap amanah sebagai landasan moral dalam kehidupan manusia. Amanah tidak hanya berkaitan dengan titipan harta atau tanggung jawab sosial, tetapi juga dengan kewajiban menyampaikan kebenaran secara utuh dan jujur. Hal ini ditegaskan melalui kisah apra rasul yang diutus kepada kaumnya. Mereka diamanahi untuk menyampaikan risalah (pesan) tanpa mengurangi ataupun menambah isinya. Salah satunya terekam dalam firman Allah SWT pada QS. Al-A'raf[7]:68 (Kemenag, 2024).

أُبَلِّغُكُمْ رِسٰلٰتِ رَبِّيْ وَاَنَا۠ لَكُمْ نَاصِحٌ اَمِيْنٌ ﴿٦٨﴾

"*Aku sampaikan kepadamu risalah-risalah (amanat) Tuhanku dan aku terhadap kamu adalah penasihat yang tepercaya"*.

Ayat di atas menegaskan pentingnya amanah dalam menyampaikan pesan. Makna amanah dalam ayat tersebut tidak hanya dipahami secara tekstual dan

lahiriah saja, tetapi juga dijelaskan secara lebih rinci dalam tafsir. Untuk memahami lebih dalam kandungan maknanya, para ulama tafsir memberikan penjelasan mengenai maksud amanah dalam tersebut. Ada dalam tafsir berikut:

"Dia, Hud, menjawab sekaligus memberikan penjelasan tentang kesalahan anggapan dan dugaan kaumnya, 'Wahai kaumku! Aku tidak seperti sangkaanmu. Bukan aku kurang waras karena aku sadar dengan ucapan dan tindakanku. Aku juga bukan pendusta, tetapi yang aku lakukan adalah berdasarkan tuntunan dari Tuhanku karena aku ini adalah seorang Rasulullah SAW yang diutus dari Tuhan seluruh alam kepada kamu semua. Aku menyampaikan kepadamu amanat, pesan, dan tuntunan dari Tuhanku dan pemberi nasihat yang menghendaki kebaikan dan kebahagiaanmu dunia dan akhirat, dan aku adalah orang yang tepercaya, jujur, bukan pembohong, yang diutus kepada kamu'." (Kemenag, 2016).

Tafsir tersebut menekankan bahwa Nabi Hud diutus dengan sifat jujur, amanah, dan dapat dipercaya, sehingga risalah Allah sampai kepada umatnya tanpa perubahan atau pengkhianat. Pesan ini memberi teladan bahwa menjaga keaslian informasi adalah bagian penting dari amanah. Dalam konteks kehidupan modern seperti ini, prinsip serupa juga berlaku dalam kriptografi, di mana pesan harus dijaga kerahasiaan dan keutuhannya agar tidak disalahgunakan atau diubah oleh pihak yang tidak berwenang. Dengan demikian, nilai amanah yang diemban para rasul memiliki relevansi kuat dengan upaya manusia menjaga keamanan informasi pada era digital.

Dalam Al-Qur'an dan hadist, amanah memiliki kedudukan yang sangat mulia karena menjadi salah satu tanda keimanan seseorang. Sebaliknya, mengkhianati amanah termasuk sifat kemunafikan.

عَنْ أَبِي هُرَيْرَةَ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ مِنْ عَلَامَاتِ الْمُنَافِقِ ثَلَاثَةٌ وَإِنْ صَامَ وَصَلَّى وَزَعَمَ أَنَّهُ مُسْلِمٌ إِذَا حَدَّثَ كَذَبَ وَإِذَا وَعَدَ أَخْلَفَ وَإِذَا اؤْتُمِنَ خَانَ

Rasulullah SAW bersabda, "Tanda orang munafik ada tiga: (1) apabila berbicara, ia berdusta; (2) apabila berjanji, ia mengingkari; (3) apabila diberi amanah, ia berkhianat" (HR.Bukhari dan Muslim).

Hadist ini menunjukkan bahwa pelanggaran terhadap amanah bukan hanya bentuk penyimpangan moral, tetapi juga mencerminkan penyakit hati yang dapat menghapus nilai keimanan. Orang yang berkhianat terhadap amanah kehilangan kepercayaan dari manusia, bahkan menjauh dari rahmat Allah SWT. Maka, menjaga amanah menjadi barometer kejujuran dan kesetiaan seseorang terhadap janji dan tanggung jawab yang telah diemban.

Kemunafikan dalam konteks amanah tidak selalu berupa pengkhianatan terhadap harta, tetapi juga mencakup pengkhianatan terhadap ilmu, jabatan, dan pesan yang diemban. Orang yang menyembunyikan kebenaran, memanipulasi informasi, atau menyampaikan pesan yang menyesatkan termasuk dalam bentuk pengkhianatan amanah. Dalam konteks modern, hal ini dapat dianalogikan dengan penyalahgunaan data, pelanggaran privasi, atau manipulasi informasi yang dapat merugikan orang lain. Dengan demikian, konsep amanah tidak hanya bersifat spiritual, tetapi juga menjadi prinsip etika dalam pengelolaan informasi dan komumikasi di era digital.

Selain itu, amanah memiliki hubungan yang erat dengan tabligh, yakni kewajiban menyampaikan kebenaran dengan jujur dan tepat sasaran. Seorang yang amanah tidak hanya menjaga kepercayaan, tetapi juga memastikan bahwa pesan yang ia sampaikan benar dan bermanfaat. Rasulullah SAW dikenal sebagai "Al-Amin" (yang terpercaya) sekaligus "Al-Muballgh" (penyampai risalah). Kombinasi

antara manah dan tabligh menjadikan beliau teladan utama dalam menyampaikan wahyu tanpa menambah atau mengurangi sedikitpun. Nilai inilah yang harus diteladani oleh setiap muslim dalam kehidupan sehari-hari yaitu menyampaikan kebenaran, menjaga keutuhan pesan, serta menjauhkan diri dari segala bentuk maniuplasi dan kebohongan.

2.15 Kajian Topik dengan Teori Pendukung

Kemajuan teknologi informasi yang semakin pesat membawa dampak signifikan bagi kehidupan manusia. Selain memeberikan kemudahan dalam proses komunikasi dan penyebaran informasi, perkembangan ini juga menimbulkan tantangan serius berupa meningkatnya ancaman keamanan data dan maraknya kejahatan siber. Kondisi tersebut menyebabkan pengguna internet menghadapi resiko kebocoran inforamasi saat melakukan pengiriman pesan. Untuk mengatasi permasalahan ini, penerapan ilmu kriptografi menjadi sangat penting sebagai lapisan keamanan pesan. Kriptografi berperan dalam menjaga kerahasiaan, keutuhan, serta keaslian informasi, sehingga rresiko akses oleh pihak tidak berwenang dapat diminimalkan.

Penelitan ini memfokuskan pada implementasi kriptosistem *McEliece* dengan memanfaatkan *Hamming* code dan kode repetisi sebagai bagian dari skema keamanannya. Kriptosistem *McEliece* merupakan salah satu skema enkripsi kunci publik yang tahan terhadap serangan *post-quantum*, karena didasarkan pada kesulitan decoding kode linier. *Hamming* code digunakankarena memiliki kemampuan mendeteksi dan mengoreksi kesalahan bit, sementara kode repetisi menambah tingkat redudansi untuk meningkatkan keandalan dalam transmisi data.

Dengan memadukan konsep teori bilangan dan teori pengkodean, penelitian ini bertujuan mengoptimalkan pemilihan parameter kunci serta menunjukkan bahwa kombinasi *McEliece* dengan *Hamming* dan kode repetisi mampu memberikan tingkat keamanan yang lebih baik dalam menjaga pesan digital dari kebocoran pesan tersebut.

BAB III

METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini merupakan jenis penelitian eksperimental dengan metode studi literatur dan simulasi. Studi literatur ini dilakukan untuk mengumpulkan berbagai teori, penelitian terdahulu, serta referensi yang relevan mengenai ilmu terkait dalam bentuk berupa jurnal, buku, skripsi, dan sumber lain yang mendukung. Simulasi dilakukan untuk menguji proses enkripsi dan dekripsi pesan dengan mengimplementasikan kriptosistem *McEliece* menggunakan kode *Hamming* dan kode repetisi.

3.2 Pra Penelitian

Pra penelitian dimulai dengan memepelajari teori dasar kriptografi, konsep kriptosistem *McEliece*, serta karakteristik *Hamming* code dan kode repetisi. Dari studi literatur dipilih kombinasi *Hamming* code dan kode repetisi yang disimulasikan untuk memberikan gambaran pada awal penelitian.

3.3 Tahapan Penelitian

1. Proses Pembentukan Kunci
   a. Membuat matriks generator G dari *Hamming* code sesuai dengan parameter $[n, k]$
   b. Membangkitkan matriks generator G berukuran $k \times n$
   c. Membentuk matriks non-singular S berukuran $k \times k$

d. Membentuk matriks permutasi P berukuran $n \times n$

e. Menghitung matriks G' dengan ukuran $k \times n$ dengan rumus sebagai berikut $G' = S \times G \times P$

2. Proses Enkripsi Pesan

a. Input *plaintext* dan mengubahnya ke bentuk biner berdasarkan tabel ASCII

b. Untuk kode repetisi, hasil biner dari pesan direpetisi dengan kode repetisi sebanyak tiga kali.

c. Deretan biner kemudian dibagi blok-blok dengan panjang masing masing 4 bit

d. Setiap blok m_i dikalikan dengan matriks generator publik G' untuk menghasilkan vektor kode C_i

e. Pengirim secara acak membangkitkan sebuah vektor error e dengan panjang n bit yang memiliki bobot t, yaitu elemen bernilai nol-nol dalam vektor tersebut.

f. Pengirim menghitung ciphertext $C_i' = C_i + e$

3. Proses dekripsi pesan dan koreksi kesalahan

a. Penerima terlebih dahulu menghitung invers dari martriks permutasi P yang digunakan pada proses pembangkitan kunci.

b. Penerima memperoleh vektor hasil antara dengan menghitung $Y_i = C_i' \times P^{-1}$, di mana C_i' merupakan ciphertext yang diterima.

c. Penerima menghitung matriks pemeriksa paritas H

d. Dengan menggunakan matriks H, penerima melakukan pemeriksaaan kesalahan dengan menggunakan *syndrome* $(Y_i(S_i) = Y_i \times H^T)$.

Kemudian matriks diacak kolomnya. Kemudian Jika $s_i = 0$ maka tidak ditemukan kesalahan, jika $s_i \neq 0$ maka hal tersebut menunjukkan adanya kesalahan.

e. Berdasarkan nilai *syndrome* s_i, penerima dapat menentukan posisi bit mana yang mengalami kesalahan dengan mencocokkannya terhadap kolom pada matriks H. Jika $s(Y_i)$ sama dengan kolom ke-i pada matriks H. Maka kesalahan terjadi pada bit ke-i dari biner Y_i
f. Setelah posisi kesalahan diketahui, penerima melakukan proses koreksi dengan menambahkan vektor kesalahan e' yang memiliki bobot 1 pada posisi yang salah, sehingga diperoleh $Y_i{}' = Y_i + e'$
g. Setelah vektor dikoreksi menjadi $Y_i{}'$, penerima mengambil empat bit pertama dari sebelah kiri pada blok biner $Y_i{}'$ sebagai bagian dari pesan yang telah diperbaiki.
h. Langkah berikutnnya, penerima menghitung invers dari matriks S yang digunakan dalam pembangkitan kunci.
i. Penerima kemudian memperoleh pesan dengan mengalikan $Y_i{}'$ dengan invers dari matriks S, yaitu $m_i{}' = Y_i{}' \times S^{-1}$
j. Untuk kode repetisi, Setiap 3 bit yang diulang dari pesan yang diperoleh diambil 1 bit pesan asli.
k. Penerima menggabungkan seluruh blok secara berurutan hingga membentuk keseluruhan pesan biner, kemudian mengonversinya kembali menjadi teks menggunakan tabel ASCII.

4. Pengaruh Repetisi
    a. Menguji dengan beberapa repetisi $r = 1, 3, 5, 10, 15, 30$ dengan tiga puluh kali percobaan untuk menjamin keakuratan data.
    b. Setiap data (waktu) yang diperoleh dimasukkan kedalam tabel.
    c. Menggunakan *tools* dari *Microsoft Excel* untuk mendapatkan hasil uji *Anova Single Factor* dan menganalisisnya.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembentukam Kunci

Proses pembentukan kunci pada algortima *McEliece* merupakan tahap yang sangat penting untuk memastikan keamanan dan efektivitas sistem kriptografi. Tahapan ini dimulai dengan penentuan parameter dasar seperti nilai n, k, dan d jika diperlukan, yang kemudian digunakan untuk membentuk matriks generator G sebagai dsar kunci publik. Selanjutya, algoritma *McEliece* memanfaatkan matriks non-singular S dan matriks permutasi acak P yang dibangkitkan secara acak untuk menghasilkan matriks kunci publik tersamarkan $G' = SGP$. Pengacakan terstruktur melalui kombinasi S dan P inilah yang menciptakan kerumitan tambahan, sehingga kunci publik sulit dianalisis secara langsung. Dengan demikian, mekanisme pembentukan kunci ini menjadikan *McEliece* sebagai metode kriptografi yang kuat dalam melindungi pesan dari berbagai serangan kriptanalis.

Dalam tahap awal sistem kriptografi *McEliece*, pengirim memulai dengan membangkitkan kunci, sebuah proses penting yang akan membentuk dasar dari keamanan seluruh sistem. Pertama-tama, pengguna menggunakan algoritma decoding, seperti metode kode Hamming, untuk menentukan parameter kunci n dan k, yang pada contoh ini diperoleh nilai $n = 7$ dan $k = 4$. Selanjutnya, matriks generator G ditentukan sebagai berikut :

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

Setelahnya, dilakukan pembangkitan matriks S secara acak dengan dimensi $k \times k$ dan matriks P dengan dimensi $n \times n$.

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

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

Selanjutnya, matriks generator G' diperoleh dari hasil perkalian matriks S, G, dan P dengan rumus sebagai berikut:

$$G' = S \times G \times P$$

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

Dengan demikian, matriks generator G' dengan ukuran 4×7 berhasil dibentuk dan siap digunakan sebagai kunci publik pada tahap enkripsi dalam algoritma *McEliece*.

4.2 Proses Enkripsi

Proses enkripsi pada algoritma *McEliece* pesan teks terlebih dahulu dikonversi menjadi bentuk biner sepanjang 8 bit. Setelah itu, matriks generator G' yang berperan sebagai kunci publik digunakan untuk membentuk blok-blok ciphertext, sehingga pesan asli berubah menjadi bentuk yang tidak dapat dipahami oleh pihak yang tidak memiliki wewenang. Penambahan kesalahan (*error*) secara acak pada setiap blok ciphertext memberikan tingkat perlindungan ekstra,

menjadikan tahapan enkripsi ini komponen vital dalam upaya menjaga kerahasiaan informasi pada kriptosistem *McEliece*.

Pada bagian ini pesan dikonversi karakter ke table ASCII menjadi bentuk biner dengan panjang delapan *bit* selanjutnya kode repetisi digunakan untuk mengulang *bit-n*. Misalnya, digunakan pesan "iya" berdasarkan tabel ASCII konversi kode binernya adalah 01101111 01111001 01101001

"i" = 01101001

"y" = 01111001

"a" = 01000001

"iya" = 01101001 01111001 01000001

Proses selanjutnya adalah enkripsi repetisi sebanyak r. Sebagai contoh, jika digunakan nilai $r = 3$, maka diperoleh:

000 111 111 000 111 000 000 111 000 111 111 111 111 000 000 111 000 111 000 000 000 000 000 111

Karena $k = 4$, maka kode biner yang telah direpetisi dikelompokkan menjadi blok dengan panjang empat bit tiap blok.

$m_1 = [0\ 0\ 0\ 1]$ $m_2 = [1\ 1\ 1\ 1]$ $m_3 = [1\ 0\ 0\ 0]$ $m_4 = [1\ 1\ 1\ 0]$

$m_5 = [0\ 0\ 0\ 0]$ $m_6 = [0\ 1\ 1\ 1]$ $m_7 = [0\ 0\ 0\ 1]$ $m_8 = [1\ 1\ 1\ 1]$

$m_9 = [1\ 1\ 1\ 1]$ $m_{10} = [1\ 1\ 1\ 0]$ $m_{11} = [0\ 0\ 0\ 0]$ $m_{12} = [0\ 1\ 1\ 1]$

$m_{13} = [0\ 0\ 0\ 1]$ $m_{14} = [1\ 1\ 0\ 0]$ $m_{15} = [0\ 0\ 0\ 0]$ $m_{16} = [0\ 0\ 0\ 0]$

$m_{17} = [0\ 0\ 0\ 0]$ $m_{18} = [0\ 1\ 1\ 1]$

Berdasarkan proses diatas, diperoleh blok pesan m_1 sampai dengan m_{18}. Selanjutnya, setiap blok tersebut dikalikan dengan matriks G' untuk mencari nilai C_n $(n = 1, 2, 3, \dots, 18)$ melalui persamaan berikut:

$$C_n = m_n \cdot G'$$

$$C_1 = m_1 \cdot G' = [0\ 0\ 0\ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 1\ 0\ 1\ 1\ 0\ 1]$$

$$C_2 = m_2 \cdot G' = [1\ 1\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [1\ 0\ 1\ 0\ 0\ 1\ 0]$$

$$C_3 = m_3 \cdot G' = [1\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 0\ 1\ 1\ 1\ 0\ 0]$$

$$C_4 = m_4 \cdot G' = [1\ 1\ 1\ 0] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [1\ 1\ 1\ 1\ 1\ 1\ 1]$$

$$C_5 = m_5 \cdot G' = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$C_6 = m_6 \cdot G' = [0\ 1\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [1\ 0\ 0\ 1\ 1\ 1\ 0]$$

$$C_7 = m_7 \cdot G' = [0\ 0\ 0\ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 1\ 0\ 1\ 1\ 0\ 1]$$

$$C_8 = m_8 \cdot G' = [1\ 1\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [1\ 0\ 1\ 0\ 0\ 1\ 0]$$

$$C_9 = m_9 \cdot G' = [1\ 1\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [1\ 0\ 1\ 0\ 0\ 1\ 0]$$

$$C_{10} = m_{10} \cdot G' = [1\ 1\ 1\ 0] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [1\ 1\ 1\ 1\ 1\ 1\ 1]$$

$$C_{11} = m_{11} \cdot G' = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$C_{12} = m_{12} \cdot G' = [0\ 1\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [1\ 0\ 0\ 1\ 1\ 1\ 0\ ]$$

$$C_{13} = m_{13} \cdot G' = [0\ 0\ 0\ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 1\ 0\ 1\ 1\ 0\ 1]$$

$$C_{14} = m_{14} \cdot G' = [1\ 1\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 1\ 0\ 1\ 0\ 1\ 0]$$

$$C_{15} = m_{15} \cdot G' = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$C_{16} = m_{16} \cdot G' = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$C_{17} = m_{17} \cdot G' = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$C_{18} = m_{18} \cdot G' = [0\ 1\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [1\ 0\ 0\ 1\ 1\ 1\ 0]$$

Diperoleh C_i sebagai berikut:

$C_1 = [0\ 1\ 0\ 1\ 1\ 0\ 1]$ $C_2 = [1\ 0\ 1\ 0\ 0\ 1\ 0]$ $C_3 = [0\ 0\ 1\ 1\ 1\ 0\ 0]$

$C_4 = [1\ 1\ 1\ 1\ 1\ 1\ 1]$ $C_5 = [0\ 0\ 0\ 0\ 0\ 0\ 0]$ $C_6 = [1\ 0\ 0\ 1\ 1\ 1\ 0]$

$C_7 = [0\ 1\ 0\ 1\ 1\ 0\ 1]$ $C_8 = [1\ 0\ 1\ 0\ 0\ 1\ 0]$ $C_9 = [1\ 0\ 1\ 0\ 0\ 1\ 0]$

$C_{10} = [1\ 1\ 1\ 1\ 1\ 1\ 1]$ $C_{11} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$ $C_{12} = [1\ 0\ 0\ 1\ 1\ 1\ 0\ ]$

$C_{13} = [0\ 1\ 0\ 1\ 1\ 0\ 1]$ $C_{14} = [0\ 1\ 0\ 1\ 0\ 1\ 0]$ $C_{15} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$

$C_{16} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$ $C_{17} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$ $C_{18} = [1\ 0\ 0\ 1\ 1\ 1\ 0]$

Selanjutnya, *ciphertext* C_i ditambahkan dengan vektor *error* acak e sepanjang 7 bit, yaitu $[0\ 0\ 1\ 0\ 0\ 0\ 0]$. Hal ini dilakukan untuk mensimulasikan gangguan saluran (*noise*) berdasarkan persamaan tersebut:

$$C_i \oplus e = C'$$

$C_1' = [0\ 1\ 0\ 1\ 1\ 0\ 1] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 1\ 1\ 1\ 1\ 0\ 1]$

$C_2' = [1\ 0\ 1\ 0\ 0\ 1\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [1\ 0\ 0\ 0\ 0\ 1\ 0]$

$C_3' = [0\ 0\ 1\ 1\ 1\ 0\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 0\ 0\ 1\ 1\ 0\ 0]$

$C_4' = [1\ 1\ 1\ 1\ 1\ 1\ 1] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [1\ 1\ 0\ 1\ 1\ 1\ 1]$

$C_5' = [0\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 0\ 1\ 0\ 0\ 0\ 0]$

$C_6' = [1\ 0\ 0\ 1\ 1\ 1\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [1\ 0\ 1\ 1\ 1\ 1\ 0]$

$C_7' = [0\ 1\ 0\ 1\ 1\ 0\ 1] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 1\ 1\ 1\ 1\ 0\ 1]$

$C_8' = [1\ 0\ 1\ 0\ 0\ 1\ 1] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [1\ 0\ 0\ 0\ 0\ 1\ 0]$

$C_9' = [1\ 0\ 1\ 0\ 0\ 1\ 1] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [1\ 0\ 0\ 0\ 0\ 1\ 0]$

$C_{10}' = [1\ 1\ 1\ 1\ 1\ 1\ 1] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [1\ 1\ 0\ 1\ 1\ 1\ 1]$

$C_{11}' = [0\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 0\ 1\ 0\ 0\ 0\ 0]$

$C_{12}' = [1\ 0\ 0\ 1\ 1\ 1\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [1\ 0\ 1\ 1\ 1\ 1\ 0]$

$C_{13}' = [0\ 1\ 0\ 1\ 1\ 0\ 1] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 1\ 1\ 1\ 1\ 0\ 1]$

$C_{14}' = [0\ 1\ 0\ 1\ 0\ 1\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 1\ 1\ 1\ 0\ 1\ 0]$

$C_{15}' = [0\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 0\ 1\ 0\ 0\ 0\ 0]$

$C_{16}' = [0\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 0\ 1\ 0\ 0\ 0\ 0]$

$C_{17}' = [0\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 0\ 1\ 0\ 0\ 0\ 0]$

$C_{18}' = [1\ 0\ 0\ 1\ 1\ 1\ 0] \oplus [0\ 0\ 1\ 0\ 0\ 0\ 0] = [1\ 0\ 1\ 1\ 1\ 1\ 0]$

Dengan demikian, diperoleh *cipherteks* sebagai berikut:

0111101 1000010 0001100 1101111 0010000

1011110 0111101 1000011 1000011 1101111

0010000 1011110 0111101 0111010 0010000

0010000 0010000 1011110

atau ditampilkan secara keseluruhan sebagai berikut:

011110110000100001100110111100100001011110011110110000111000011110 1

111001000010111100111101011101000100000010000001000001011110

4.3 Proses Dekripsi

Proses dekripsi terdiri dari beberapa tahapan penting untuk memngembalikan *ciphertext* menjadi pesan aslinya. Setelah *ciphertext* diterima, langkah pertama adalah menggunakan invers dari matriks P dan S yang sebelumnya dipakai dalam pembangkitan kunci. Invers matriks P diterapkan terlebih dahulu pada setiap blok *ciphertext* untuk menghilangkan efek permutasi. Tahap selanjutnya adalah proses koreksi *error* yang dilakukan dengan memanfaatkan matriks *parity check* guna mendeteksi dan memperbaiki kesalahan yang muncul pada *ciphertext*. Setelah *error* diperbaiki, hasilnya dikalikan dengan invers matriks S untuk mendapatkan kembali representasi dari pesan yang sudah direpetisi lalu dengan dekripsi kode repetisi

dapat menghasilkan bilangan biner yang akan dikonversi kembali ke teks atau karakter menggunakan tabel ASCII sehingga pesan dapat dibaca seperti semula.

Setelah penerima menerima teks, proses selanjutnya yaitu menghitung dan menentukan invers matriks P atau P^{-1}

$$P^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Setelah P^{-1} diketahui, menggunakan rumus sebagai berikut $C_i{}' \cdot P^{-1}$ untuk menghasilkan matriks y_i.

$$y_i = C_i{}' \times P^{-1}$$

$$y_1 = [0\ 1\ 1\ 1\ 1\ 0\ 1] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 1\ 1\ 0\ 0\ 1\ 1]$$

$$y_2 = [1\ 0\ 0\ 0\ 0\ 1\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [0\ 0\ 0\ 1\ 1\ 0\ 0]$$

$$y_3 = [1\ 0\ 1\ 1\ 1\ 1\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [0\ 0\ 0\ 0\ 0\ 1\ 1]$$

$$y_4 = [1\ 1\ 0\ 1\ 1\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [0\ 1\ 1\ 1\ 1\ 1\ 1]$$

$$y_5 = [0\ 0\ 1\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$y_6 = [1\ 0\ 1\ 1\ 1\ 1\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 0\ 0\ 1\ 1\ 1\ 1]$$

$$y_7 = [0\ 1\ 1\ 1\ 1\ 0\ 1] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 1\ 1\ 0\ 0\ 1\ 1]$$

$$y_8 = [1\ 0\ 0\ 0\ 0\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [0\ 0\ 0\ 1\ 1\ 0\ 0]$$

$$y_9 = [1\ 0\ 0\ 0\ 0\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [0\ 0\ 0\ 1\ 1\ 0\ 0]$$

$$y_{10} = [1\ 1\ 0\ 1\ 1\ 1\ 1] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [0\ 1\ 1\ 1\ 1\ 1\ 1]$$

$$y_{11} = [0\ 0\ 1\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$y_{12} = [1\ 0\ 1\ 1\ 1\ 1\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 0\ 0\ 1\ 1\ 1\ 1]$$

$$y_{13} = [0\ 1\ 1\ 1\ 1\ 0\ 1] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 1\ 1\ 0\ 0\ 1\ 1]$$

$$y_{14} = [0\ 1\ 1\ 1\ 0\ 1\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 1\ 0\ 0\ 1\ 0\ 1]$$

$$y_{15} = [0\ 0\ 1\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$y_{16} = [0\ 0\ 1\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$y_{17} = [0\ 0\ 1\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$y_{18} = [1\ 0\ 1\ 1\ 1\ 1\ 0] \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = [1\ 0\ 0\ 1\ 1\ 1\ 1]$$

Melakukan proyeksi *error* dengan mencari nilai syndrome $S(y_i)$. Dengan rumus sebagai berikut:

$$S(y_i) = H \times {y_i}^T$$

Matriks *parity check* H didapatkan dari $G = [I_4|X]$ di mana I_4 merupakan matriks identitas berukuran 4×4 dan X suatu matriks berukuran 4×3

$$\left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)$$

$$[I_4|X]$$

Selanjutnya, matriks *parity check* $H = [Y|I_3]$ ditentukan, di mana Y diperoleh dari hasil transpose dari X (X^T). Dengan demikian, diperoleh hasil sebagai berikut:

$$H = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right)$$

$$[I_3|Y]$$

Selanjutnya, dilakukan pengacakan kolom untuk memperoleh matriks H sesuai dengan *sagemath* sebagai berikut:

$$H = \left(\begin{array}{cccc|ccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)$$

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

Perkalian antara matriks H dengan transpose dari vektor y_i akan menghasilkan nilai *syndrome* $S(y_i)$ berdasarkan persamaan berikut:

$$S(y_i) = y_i \cdot H^T$$

$$S(y_1) = [1\ 1\ 1\ 0\ 0\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_2) = [0\ 0\ 0\ 1\ 1\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_3) = [0\ 0\ 0\ 0\ 0\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_4) = [0\ 1\ 1\ 1\ 1\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_5) = [1\ 0\ 0\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_6) = [1\ 0\ 0\ 1\ 1\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_7) = [1\ 1\ 1\ 0\ 0\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_8) = [0\ 0\ 0\ 1\ 1\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_9) = [0\ 0\ 0\ 1\ 1\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_{10}) = [0\ 1\ 1\ 1\ 1\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_{11}) = [1\ 0\ 0\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_{12}) = [1\ 0\ 0\ 1\ 1\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_{13}) = [1\ 1\ 1\ 0\ 0\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_{14}) = [1\ 1\ 0\ 0\ 1\ 0\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_{15}) = [1\ 0\ 0\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_{16}) = [1\ 0\ 0\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_{17}) = [1\ 0\ 0\ 0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

$$S(y_{18}) = [1\ 0\ 0\ 1\ 1\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0]$$

Seluruh *error* terdeteksi pada baris pertama, yaitu $[1\ 0\ 0]$. Selanjutnya, vektor *error* e' ditambahkan dengan memberikan nilai bit 1 pada posisi bit yang salah melalui persamaan berikut:

$$y_i' = y_i + e'$$

$y_1' = [1\ 1\ 1\ 0\ 0\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 1\ 1\ 0\ 0\ 1\ 1]$

$y_2' = [0\ 0\ 0\ 1\ 1\ 0\ 0] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [1\ 0\ 0\ 1\ 1\ 0\ 0]$

$y_3' = [0\ 0\ 0\ 0\ 0\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [1\ 0\ 0\ 0\ 0\ 1\ 1]$

$y_4' = [0\ 1\ 1\ 1\ 1\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [1\ 1\ 1\ 1\ 1\ 1\ 1]$

$y_5' = [1\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 0\ 0\ 0\ 0\ 0\ 0]$

$y_6' = [1\ 0\ 0\ 1\ 1\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 0\ 0\ 1\ 1\ 1\ 1]$

$y_7' = [1\ 1\ 1\ 0\ 0\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 1\ 1\ 0\ 0\ 1\ 1]$

$y_8' = [0\ 0\ 0\ 1\ 1\ 0\ 0] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [1\ 0\ 0\ 1\ 1\ 0\ 0]$

$y_9' = [0\ 0\ 0\ 1\ 1\ 0\ 0] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [1\ 0\ 0\ 1\ 1\ 0\ 0]$

$y_{10}' = [0\ 1\ 1\ 1\ 1\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [1\ 1\ 1\ 1\ 1\ 1\ 1]$

$y_{11}' = [1\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 0\ 0\ 0\ 0\ 0\ 0]$

$y_{12}' = [1\ 0\ 0\ 1\ 1\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 0\ 0\ 1\ 1\ 1\ 1]$

$y_{13}' = [1\ 1\ 1\ 0\ 0\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 1\ 1\ 0\ 0\ 1\ 1]$

$y_{14}' = [1\ 1\ 0\ 0\ 1\ 0\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 1\ 0\ 0\ 1\ 0\ 1]$

$y_{15}' = [1\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 0\ 0\ 0\ 0\ 0\ 0]$

$y_{16}' = [1\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 0\ 0\ 0\ 0\ 0\ 0]$

$y_{17}' = [1\ 0\ 0\ 0\ 0\ 0\ 0] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 0\ 0\ 0\ 0\ 0\ 0]$

$y_{18}' = [1\ 0\ 0\ 1\ 1\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0] = [0\ 0\ 0\ 1\ 1\ 1\ 1]$

Setelah perbaikan *error* dilakukan, langkah selanjutnya adalah mengambil 4 bit biner pertama dari sisi kiri vektor y_i' untuk dikalikan dengan matriks S^{-1} sebagai berikut:

$y_1' = [0\ 1\ 1\ 0]$	$y_2' = [1\ 0\ 0\ 1]$	$y_3' = [1\ 0\ 0\ 0]$
$y_4' = [1\ 1\ 1\ 1]$	$y_5' = [0\ 0\ 0\ 0]$	$y_6' = [0\ 0\ 0\ 1]$
$y_7' = [0\ 1\ 1\ 0]$	$y_8' = [1\ 0\ 0\ 1]$	$y_9' = [1\ 0\ 0\ 1]$
$y_{10}' = [1\ 1\ 1\ 1]$	$y_{11}' = [0\ 0\ 0\ 0]$	$y_{12}' = [0\ 0\ 0\ 1]$
$y_{13}' = [0\ 1\ 1\ 0]$	$y_{14}' = [0\ 1\ 0\ 0]$	$y_{15}' = [0\ 0\ 0\ 0]$
$y_{16}' = [0\ 0\ 0\ 0]$	$y_{17}' = [0\ 0\ 0\ 0]$	$y_{18}' = [0\ 0\ 0\ 1]$

Setelah tiga bit terakhir pada partisi dibuang, empat bit awal yang diperoleh kemudian dikalikan dengan S^{-1} untuk mencari nilai m_i'. Adapun matriks S^{-1} adalah sebagai berikut:

$$S^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$m_1' = [0\ 1\ 1\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 0\ 0\ 1]$$

$$m_2' = [1\ 0\ 0\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [1\ 1\ 1\ 1]$$

$$m_3' = [1\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [1\ 0\ 0\ 0]$$

$$m_4' = [1\ 1\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [1\ 1\ 1\ 0]$$

$$m_5' = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 0\ 0\ 0]$$

$$m_6' = [0\ 0\ 0\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 1\ 1\ 1]$$

$$m_7' = [0\ 1\ 1\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 0\ 0\ 1]$$

$$m_8' = [1\ 0\ 0\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [1\ 1\ 1\ 1]$$

$$m_9' = [1\ 0\ 0\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [1\ 1\ 1\ 1]$$

$$m_{10}' = [1\ 1\ 1\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [1\ 1\ 1\ 0]$$

$$m'_{11} = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 0\ 0\ 0]$$

$$m'_{12} = [0\ 0\ 0\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 1\ 1\ 1]$$

$$m'_{13} = [0\ 1\ 1\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 0\ 0\ 1]$$

$$m'_{14} = [0\ 1\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [1\ 1\ 0\ 0]$$

$$m'_{15} = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 0\ 0\ 0]$$

$$m'_{16} = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 0\ 0\ 0]$$

$$m'_{17} = [0\ 0\ 0\ 0] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 0\ 0\ 0]$$

$$m'_{18} = [0\ 0\ 0\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = [0\ 1\ 1\ 1]$$

Setelah diperoleh 18 blok yang masing-masing berisi 4 bit, data tersebut dikonversi menjadi 24 blok dengan representasi biner 3 bit sebagai berikut:

000	111	111	000	111	000	000	111
000	111	111	111	111	000	000	111
000	111	000	000	000	000	000	111

Dengan mengembalikan repetisi tiga, maka didaptkan kembali *plainteks* semula yaitu :

01101001 yaitu “i”

01111001 yaitu “y”

01000001 yaitu “a”

Dengan demikian, keseluruhan pesan adalah “iya”. Proses ini telah dilakukan dengan perhitungan waktu proses enkripsi dan dekripsi pada pesan “iya” yaitu 24 bit menggunakan variasi nilai repetisi r sebesar 1, 5, 10, 15, 30 yang dapat dilihat dalam lampiran 3 dan 4 hasil pengukuran menunjukkan bahwa meskipun nilai r meningkat, waktu eksekusi yang diperoleh berada dalam rentang mikrodetik dengan perbedaan yang tidak signifikan, mengindikasi bahwa untuk pesan pendek, peningkatan repetisi belum memberikan pengaruh besar terhadap waktu proses. Data ini menjadi dasar dalam pembahasan mengenai efisiensi waktu dan *overhead* pada algoritma kode repetisi yang dijalankan sekaligus melengkapi analisis kinerja pada proses ini.

4.4 Pengaruh Repetisi

Pengujian pada tahap ini dilakukan untuk mengetahui sejauh mana penambahan jumlah repetisi r memengaruhi efisiensi waktu pemrosesan dalam sistem yang dirancang. Nilai repetisi yang diuji pada uji coba ini divariasikan menjadi enam kelompok, yaitu $r = 1, 3, 5, 10, 15, 30$ dengan masing-masing kelompok dilakukan sebanyak 30 kali percobaan untuk mendapatkan data yang valid secara statistik. Tabel 4.1 adalah hasil data yang diperoleh dari hasil uji coba.

Tabel 4.1 Hasil Data Waktu Eksekusi Berdasarkan Variasi Repetisi

	$r = 1$	$r = 3$	$r = 5$	$r = 10$	$r = 15$	$r = 30$
1	0,214884	0,236075	0,251334	0,359933	0,354884	0,485672
2	0,211169	0,242438	0,252813	0,312661	0,339495	0,501665
3	0,206604	0,227524	0,265492	0,315453	0,352143	0,575515
4	0,210274	0,229218	0,270335	0,351535	0,353212	0,497281
5	0,215681	0,239664	0,309137	0,404303	0,346803	0,501014
6	0,222982	0,239383	0,284218	0,808650	0,355846	0,496342
7	0,211442	0,243476	0,268096	0,834526	0,348346	0,499427
8	0,210001	0,234900	0,265034	0,807003	0,345400	0,498909
9	0,211154	0,231047	0,263833	0,365527	0,357808	0,505627
10	0,208560	0,247329	0,263397	0,319800	0,353319	0,492049
11	0,217343	0,235300	0,277462	0,340518	0,354953	0,492084
12	0,209897	0,240704	0,306812	0,667745	0,352815	0,545451
13	0,207297	0,245992	0,255221	0,455289	0,353136	0,492387
14	0,205199	0,245338	0,256453	0,349680	0,346866	0,489099
15	0,199770	0,231389	0,258999	0,311646	0,353397	0,518076
16	0,205084	0,245283	0,257360	0,301285	0,344177	0,601711
17	0,201036	0,237107	0,262240	0,295836	0,346787	0,496569
18	0,202458	0,235294	0,253186	0,299325	0,356184	0,499291
19	0,207520	0,233435	0,252692	0,312498	0,348394	0,753980
20	0,222402	0,238696	0,254850	0,311963	0,365310	0,652664
21	0,209580	0,239881	0,264441	0,291716	0,346764	0,716974
22	0,212977	0,236884	0,261549	0,314612	0,351432	0,796750
23	0,232223	0,235835	0,266579	0,293533	0,352769	0,674853
24	0,209751	0,237272	0,258288	0,293786	0,382764	0,501577
25	0,218786	0,234590	0,279174	0,297276	0,483277	0,502745
26	0,214570	0,233375	0,254763	0,296052	0,366270	0,493002
27	0,216393	0,232895	0,255476	0,307875	0,351693	0,496902
28	0,218093	0,235200	0,259064	0,288250	0,372211	0,541165
29	0,233797	0,235040	0,257564	0,297324	0,352045	0,503481
30	0,214574	0,237416	0,260571	0,310927	0,396699	0,518327

Tabel 4.2 Ringkasan Statistik

Groups	*Count*	*Sum*	*Average*	*Variance*
1	30	6,381502	0,212717	0,0000629874
3	30	7,117979	0,237266	0,0000247681
5	30	7,946434	0,264881	0,000201003
10	30	11,51652	0,383884	0,026801312
15	30	10,7852	0,359507	0,000681502
30	30	16,34059	0,544686	0,007405089

Berdasarkan hasil pengujian yang dirangkum pada Tabel 4.1, terlihat adanya tren peningkatan waktu eksekusi yang berbanding lurus dengan jumlah repetisi yang digunakan. Rata-rata waktu eksekusi tercepat dicatatkan pada repetisi $r = 1$ sebesar $0{,}21272$ detik, sementara waktu eksekusi terlama berada pada repetisi $r = 30$ dengan rata-rata $0{,}54469$ detik. Data menunjukkan bahwa meskipun beban repetisi meningkat hingga tiga puluh kali lipat, selisih waktu absolut yang dihasilkan masih berada pada rentang yang sangat singkat, yaitu kurang dari satu detik.

Tabel 4.3 Hasil Analisis Anova Satu Arah

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	2,291163	5	0,458233	78,15965	1,17863E-42	2,266062
Within Groups	1,020123	174	0,005863			
Total	3,311286	179				

Untuk membuktikan apakah perbedaan waktu tersebut memiliki pengaruh yang signifikan secara ilmiah, dilakukan analisis statistik menggunakan metode *Anova: Single Factor*. Berdasarkan perhitungan pada Tabel 4.2, diperoleh nilai *F* hitung sebesar $78{,}1596$ yang jauh lebih besar dibandingkan nilai $F\ crit$ sebesar $2{,}26606$. Hal ini diperkuat dengan perolehan nilai *P-value* sebesar $1{,}2 \times 10^{-42}$ yang berada jauh di bawah ambang batas signifikansi $0{,}05$. Hasil ini menginformasi bahwa secara statistik, jumlah repetisi memberikan pengaruh yang signifikan terhadap durasi waktu eksekusi sistem. Semakin besar nilai r, maka waktu yang dibutuhkan semakin lama.

4.5 Kajian Islami terhadap Implementasi Kriptosistem *McEliece* Menggunakan Kode *Haminng* dan Kode Repetisi

Proses enkripsi pada kriptosistem *McEliece* bekerja dengan mengubah pesan asli menjadi bentuk sandi yang tidak dapat dipahami tanpa kunci privat. Dalam perspektif Islam, tindakan ini dapat dipahami sebagai wujud nyata menjaga amanah dan menjaga kerahasiaan pesan, sebagaimana dianjurkan dalam syariat. Nabi Muhammad SAW menegaskan bahwa kerahasiaan adalah bentuk amanah. Hal ini dijelaskan dalam hadist:

إِذَا حَدَّثَ رَجُلٌ رَجُلاً بِحَدِيْثٍ ثُمَّ الْتَفَتَ فَهُوَ أَمَانَةٌ

"*Apabila seseorang membicarakan suatu pembicaraan kemudian ia menoleh (ke kanan dan ke kiri), maka pembicaraan itu adalah amanah***."** (HR. Abu Daud).

Hadist ini menjelaskan bahwa menjaga rahasia orang lain adalah kewajiban, meskipun bentuk rahasia tersebut kini lebih banyak berada dalam ranah digital. Enkripsi merupakan cara untuk mencegah kebocoran pesan kepada pihak yang tidak berhak, sehingga sejalan dengan prinsip menjaga amanah tersebut. Sebagaimana algoritma *McEliece* memberikan perlindungan kuat terhadap serangan kuantum maupun penyadapan, Islam juga mengajarkan bahwa setiap amanah harus dijaga dari berbagai bentuk pengkhianatan.

Dekripsi dalam kriptosistem *McEliece* hanya dapat dilakukan oleh pihak yang memiliki kunci privat, yaitu penerima yang berhak. Hal ini mencerminkan prinsip penting dalam Islam mengenai menyampaikan amanah hanya kepada pihak yang berhak menerimanya. Surat an-nisa ayat 58 (Kemenag, 2025).

﴿ ۞ اِنَّ اللّٰهَ يَأْمُرُكُمْ اَنْ تُؤَدُّوا الْاَمٰنٰتِ اِلٰٓى اَهْلِهَاۙ وَاِذَا حَكَمْتُمْ بَيْنَ النَّاسِ اَنْ تَحْكُمُوْا بِالْعَدْلِۗ اِنَّ اللّٰهَ نِعِمَّا يَعِظُكُمْ بِهٖۗ اِنَّ اللّٰهَ كَانَ سَمِيْعًاۢ بَصِيْرًا ٥٨ ﴾

"Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya".

Ayat tersebut menegaskan bahwa setiap bentuk amanah termasuk informasi wajib disampaikan dengan benar kepada pihak yang berhak menerimanya. Dalam kriptosistem *McEliece*, prinsip ini tercermin melalui mekanisme yang memastikan bahwa hanya penerima sah yang dapat memulihkan pesan, sementara pihak yang tidak memiliki otoritas tidak dapat mengaksesnya. Dengan demikian, proses deskripsi dalam kriptografi tidak hanya bersifat teknis, tetapi juga mencerminkan nilai kejujuran dan tanggung jawab dalam penyampaian informasi sebagaimana diajarkan dalam Islam.

Selain itu, konsep amanah dalam Islam juga menekankan pentingnya menjaga kerahasiaan dan keutuhan pesan. Dalam konteks komunikasi digital, informasi yang dikirimkan seseorang kepada penerima merupakan titipan yang harus dijaga dari kebocoran maupun manipulasi. *McEliece* dengan struktur matematis yang kuat, memberikan perlindungan terhadap informasi tersebut sehingga pihak yang tidak berhak tidak dapat membacanya. Hal ini selaras dengan prinsip syariat untuk menjaga sesuatu yang telah dipercayakan, baik berupa data, pesan, maupun pengetahuan.

Mekanisme kunci publik dan kunci privat pada *McEliece* juga menggambarkan bahwa tidak semua orang berhak mengakses informasi tertentu. Islam mengajarkan batasan yang jelas terkait hak akses, di mana suatu amanah hanya boleh dibuka oleh pemilik izin atau orang yang telah diberi kepercayaan. Dengan demikian, penggunaan algoritma kriptografi bukan hanya memenuhi teknis

keamanan informasi, tetapi juga merupakan bentuk implementasi nilai moral dalam memelihara kerahasiaan, amanah, dan integritas komunikasi.

Proses ini telah dilakukan peneliti dengan perhitungan waktu proses enkripsi dan dekripsi pada pesan "iya" yaitu 24 bit menggunakan variasi nilai repetisi r sebesar 1, 5, 10, 15, 30 hasil pengukuran menunjukkan bahwa meskipun nilai r meningkat, waktu eksekusi yang diperoleh berada dalam rentang mikrodetik dengan perbedaan yang tidak signifikan, mengindikasi bahwa untuk pesan pendek, peningkatan repetisi belum memberikan pengaruh besar terhadap waktu proses. Data ini menjadi dasar dalam pembahasan mengenai efisiensi waktu dan *overhead* pada algoritma kode repetisi yang dijalankan sekaligus melengkapi analisis kinerja pada proses ini.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pembahasan diatas dapat didapatkan :

1. Proses enkripsi dimulai dengan mengubah pesan plainteks menjadi biner dengan 8 bit berdasarkan tabel ASCII. Setelah itu biner direpetisi sesuai dengan r yang mempengaruhi panjang dan redudansi pesan. Kemudian bibagi ke dalam blok-blok 4 bit. Selanjutnya, dikalikan dengan matriks generator G' dan hasil tersebut ditambahkan error dengan bobot 1 untuk menghasilkan *cipherteks*
2. Proses dekripsi diawali dengan mengalikan *cipherteks* dengan invers matriks permutasi P. Kemudian kode Hamming mulai mengoreksi n memperbaiki data bit dengan mencari *syndrome* dari y_i kemudian dikalikan dengan matriks *parity check*. Hasil setelahnya y_i dikalikan dengan H transpose yang bisa mengoreksi kesalahan. Setelah *error* dikoreksi dikalikan dengan invers dari matriks S. Kemudian blok-blok digabungkan dan di hilangkan repetisinya sesuai r. Setelah itu tahap terakhir yaitu memkonversi biner ke dalam tabel ASCII untuk memperoleh pesan asli.
3. Parameter r pada kode repetisi berfungsi menentukan jumlah pengulangan setiap bit pesan, dimana nilai r menambah redundansi Dalam pengujian dengan variasi $r = 1, 3, 5, 10, 15, 30$. Secara statistik, penambahan jumlah repetisi r memberikan pengaruh yang signifikan

secara statistik terhadap waktu eksekusi sitem. Hal ini dibuktikan melalui uji *Anova*: *Single Factor* yang menghasilkan nilai F hitung $78{,}1596$ jauh lebih besar daripada F kritis $2{,}2660$ serta nilai *P-value* $1{,}2 \times 10^{-42}$ yang berada dibawah taraf signifikansi $0{,}05$. Begitupun terdapat korelasi positif antara jumlah repetisi dengan durasi pemrosesan, di mana rata-rata waktu eksekusi meningkat seiring bertambahnya nilai r. Meskipun secara statistik perbedaannya cukup signifikan, secara praktis selisih waktu antara repetisi terendah dan tertinggi hanya sebesar $0{,}33197$ detik. Dengan demikian, penambahan proses repetisi hingga 30 kali (tertinggi) tetap layak untuk diimplementasikan karena peningkatan keamanan yang diberikan tidak membebani performa waktu sistem secara berlebihan karena masih di bawah satu detik.

5.2 Saran

Saran untuk penelitian selanjutnya, diharapkan dapat menggunakan gabungan dari kode lainnya. Pada penelitian ini sebatas pesan berupa kata sebagai objek, diharapkan peneliti selanjutnya dapat mengekplorasi dalam bentuk audio, video, gambar serta menganalisi keamannanya.

DAFTAR PUSTAKA

Andika, T., Taquyuddin, M., & Admizal, I. (2020). Amanah dan Khianat dalam Al-Qur'an Menurut Quraish Shihab. *Al Tadabbur: Jurnal Ilmu Alquran Dan Tafsir*, *5*(2), 177–206.

Arif, Z., & Nurokhman, A. (2023). Analisis Perbandingan Algoritma Kriptografi Simetris Dan Asimetris Dalam Meningkatkan Keamanan Sistem Informasi. *Jurnal Teknologi Sistem Informasi*, *4*(2), 394–405. https://doi.org/10.35957/jtsi.v4i2.6077

Birken, P. (2021). Introduction to Algebraic Coding Theory. In *Student Solutions Manual*. https://doi.org/10.1201/9781003182306-30

Burton, D. M. (2011). Elementary Number Theory. In *Mathematical Miniatures*. https://doi.org/10.5948/upo9780883859575.049

Handayani, R., & Yulina. (2020). Teori Bilangan. In *Paper Knowledge . Toward a Media History of Documents* (Vol. 3, Nomor April).

Hidayatuloh, K., Yustantina, Y., & Kusmadi, K. (2021). Perbandingan Metode Stream Cipher Dan Hill Cipher Dalam Keamanan Data. *Infotronik : Jurnal Teknologi Informasi dan Elektronika*, *6*(1), 27. https://doi.org/10.32897/infotronik.2021.6.1.647

Hutagalung, J., Ramadhan, P. S., & Sihombing, S. J. (2023). Keamanan Data Menggunakan Secure Hashing Algorithm (SHA)-256 dan Rivest Shamir Adleman (RSA) pada Digital Signature. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *10*(6), 1213–1222. https://doi.org/10.25126/jtiik.1067319

Kemenag. (2024). *Qur'an Kemenag*. https://quran.kemenag.go.id/

Kemenag, R. (2016). *Tafsir Wajiz Jilid I*. Lajnah Pentashihan Mushaf Al-Qur'an.

Kemenag, R. (2025). *Qur'an Kemenag*. https://quran.kemenag.co.id

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

Maulyanda, Pane, S. F., & Awangga, R. M. (2021). Cryptography: Perancangan Middleware Web Service Encryptor menggunakan Triple Key MD5. Base64, dan AES. *Jurnal Tekno Insentif*, *15*(2), 65–75. https://doi.org/10.36787/jti.v15i1.497

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

Prayoga, D., Hayati, F., Putra, H. A. Y., Rizki, I. N., & Fitroh, F. (2022). Risiko Keamanan Data Pribadi Pelanggan Dalam Penggunaan Big Data. *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, *5*(3), 459–463. https://doi.org/10.32672/jnkti.v5i3.4381

Rosen, K. H. (1987). Elementary Number Theory and Its Applications. In *Mathematics of Computation* (Vol. 48, Nomor 177). https://doi.org/10.2307/2007902

Sinaga, C. S., Gultom, A. F., Nababan, D. R., Simanjuntak, A. R., & Ginting, E. (2024). Cryptography With Mceliece Algorithm (Code Based Cryptography). *Jurnal Teknik Indonesia*, *3*(02), 49–53. https://doi.org/10.58471/juti.v3i02.665

Stallings, W. (2017). *Cryptography and Network Security Seventh Edition*.

LAMPIRAN

Lampiran 1. *Script Code*

```
from sage.all import *
import time

# kode lain ...

def rep_encode_blocks(msg, n=4, F=GF(2)):
    """
    Encode seluruh bit pesan, setiap bit direpetisi n kali.
    Output dipisah menjadi blok-blok berukuran n.
    """
    blocks = []
    for bit in msg:
        b = int(bit)
        block = vector(F, [b]*n)
        blocks.append(block)
    return blocks  # list of blocks

def rep_decode_blocks(blocks, n=4, F=GF(2)):
    """
    Decode blok-blok ukuran n dengan majority vote setiap blok.
    Input berupa list of vectors.
    """
    decoded = []
    threshold = (n + 1) // 2
    for block in blocks:
        s = sum(int(x) for x in block)
        bit = 1 if s >= threshold else 0
        decoded.append(F(bit))
    return vector(F, decoded)
from sage.all import GF, vector

def _is_sequence(x):
    # Anggap string bukan sequence di sini
    return hasattr(x, '__len__') and not isinstance(x, (str, bytes))

def flatten_one_level(seq):
    """
    Jika seq berisi elemen yang semuanya skalar -> kembalikan list(seq)
    Jika seq berisi elemen yang sebagian adalah iterable
(tuple/list/vector),
      maka flatten satu level: gabungkan elemen-elemennya menjadi satu
daftar skalar.
    """
    # jika kosong, return []
    if len(seq) == 0:
        return []
    # periksa apakah ada elemen yang iterable (bukan string/bytes)
    has_iterable = any(_is_sequence(el) for el in seq)
    if not has_iterable:
        return list(seq)
    # flatten satu level
    out = []
    for el in seq:
        if _is_sequence(el):
            out.extend(list(el))
        else:
            out.append(el)
    return out
```

```
def split_into_blocks_safe(v, n=4, F=GF(2)):
    """
    Memecah input v menjadi blok ukuran n (Sage vectors over F).
    - Dapat menerima: vector/list of ints, list of tuples, vector of
vectors, dll.
    - Flatten satu level jika diperlukan.
    - Menghasilkan list of vectors (masing-masing elemen adalah vector
over F).
    """
    # Jika v adalah Sage vector, ambil listnya
    try:
        seq = list(v)
    except TypeError:
        # fallback
        seq = v

    # Flatten satu level bila diperlukan
    flat = flatten_one_level(seq)

    # Pastikan semua elemen sekarang skalar (int/0/1 atau GF element)
    # Konversi ke F dengan aman: gunakan int(x) untuk extraksi jika
bisa.
    scalar_list = []
    for el in flat:
        # jika el adalah GF element -> langsung pakai
        if hasattr(el, 'parent') and getattr(el, 'parent') is F:
            scalar_list.append(el)
            continue
        # jika el bisa dikonversi ke int (mis. '0','1', True/False) ->
gunakan int
        try:
            val = int(el)
        except Exception:
            raise TypeError("Elemen tidak dapat dikonversi ke skalar
GF: {!r}".format(el))
        scalar_list.append(F(val))

    L = len(scalar_list)
    if L % n != 0:
        raise ValueError("Panjang data ({}) bukan kelipatan n =
{}".format(L, n))

    blocks = []
    for i in range(0, L, n):
        blk_list = scalar_list[i:i+n]
        blocks.append(vector(F, blk_list))
    return blocks


# ------------------------
# Contoh penggunaan
# ------------------------
start = time.time()
F = GF(2)
n = 4
r = 3
C=codes.HammingCode(F,3)
G=C.generator_matrix();print('Matriks G = \n',G)
H=C.parity_check_matrix();print('Matriks H = \n',H)
Ht=C.parity_check_matrix().transpose();print('Matriks H Transpose =
\n',Ht)
j=7
Per=matrix(Permutation(choice(Permutations(list(range(1,j+1)),j).list()
)));
```

```
P=matrix(GF(2),Per);
S=random_matrix(GF(2),4, algorithm='unimodular');
Gp=S*G*P;print(Gp)
print('Matriks P = \n',P)
print('Matriks P invers = \n',P.inverse())
print('Matriks S = \n',S)
print('Matriks S invers = \n',S.inverse())
Gp=S*G*P;print('Generator Publik =\n',Gp)
m = vector(F, [0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1,
0, 1, 0, 0, 1])
print('Num Repetition : ', r)
# Encode jadi blok-blok 4 bit
encoded_blocks = rep_encode_blocks(m, n=r, F=F)
blocks = split_into_blocks_safe(encoded_blocks, n=4, F=F)
print(encoded_blocks)
print(blocks)

received_blocks=[]
corection_blocks=[]
print("\nEncoded per blok 4 bit:")
for i, blk in enumerate(blocks):
    print(f"Blok {i}: {blk}")
    m1=blocks[i]
    m=m1*Gp;print(m)
    e=vector(GF(2),[0,0,1,0,0,0,0]);C1=m+e
    received_blocks.append(C1)

for i, blk in enumerate(received_blocks):
    print('Received Blocks : ',f"Blok {i}: {blk}")
    m1SG=received_blocks[i]*(P.inverse());print('Hasil kali P invers =
',m1SG)
    syndrome=H*m1SG;print('Syndrome=',syndrome)
    m1=C.decode_to_message(m1SG);print('Hasil koreksi :',m1)
    m1a=m1*S.inverse();print('Hasil kali S invers = ',m1a)
    corection_blocks.append(m1a)
    print('Decoding Result : ',m1a)
dblocks = split_into_blocks_safe(corection_blocks, n=r, F=F)
# Decode
decoded = rep_decode_blocks(dblocks, n=r, F=F)
print("\nDecoded message:", decoded)

end = time.time()

# pemanggilan
waktu=end-start
```

Lampiran 2. *Output*

```
Matriks G =
 [1 0 0 0 0 1 1]
[0 1 0 0 1 0 1]
[0 0 1 0 1 1 0]
[0 0 0 1 1 1 1]
Matriks H =
 [1 0 1 0 1 0 1]
[0 1 1 0 0 1 1]
[0 0 0 1 1 1 1]
Matriks H Transpose =
 [1 0 0]
[0 1 0]
[1 1 0]
[0 0 1]
[1 0 1]
[0 1 1]
[1 1 1]
[1 1 1 0 0 0 1]
[1 0 0 0 1 0 1]
[1 1 0 1 1 0 0]
[0 0 1 0 0 1 1]
Matriks P =
 [0 1 0 0 0 0 0]
[0 0 0 0 0 0 1]
[0 0 0 1 0 0 0]
[0 0 0 0 1 0 0]
[0 0 1 0 0 0 0]
[1 0 0 0 0 0 0]
[0 0 0 0 0 1 0]
Matriks P invers =
 [0 0 0 0 0 1 0]
[1 0 0 0 0 0 0]
[0 0 0 0 1 0 0]
[0 0 1 0 0 0 0]
[0 0 0 1 0 0 0]
[0 0 0 0 0 0 1]
[0 1 0 0 0 0 0]
Matriks S =
 [1 1 0 0]
[0 1 0 1]
[1 0 1 1]
[0 1 0 0]
Matriks S invers =
 [1 0 0 1]
[0 0 0 1]
[1 1 1 0]
[0 1 0 1]
Generator Publik =
 [1 1 1 0 0 0 1]
[1 0 0 0 1 0 1]
[1 1 0 1 1 0 0]
[0 0 1 0 0 1 1]
Num Repetition :  3
[(0, 0, 0), (1, 1, 1), (1, 1, 1), (0, 0, 0), (1, 1, 1), (0, 0, 0), (0,
0, 0), (1, 1, 1), (0, 0, 0), (1, 1, 1), (1, 1, 1), (0, 0, 0), (1, 1,
1), (0, 0, 0), (0, 0, 0), (1, 1, 1), (0, 0, 0), (1, 1, 1), (1, 1, 1),
(0, 0, 0), (1, 1, 1), (0, 0, 0), (0, 0, 0), (1, 1, 1)]
[(0, 0, 0, 1), (1, 1, 1, 1), (1, 0, 0, 0), (1, 1, 1, 0), (0, 0, 0, 0),
(0, 1, 1, 1), (0, 0, 0, 1), (1, 1, 1, 1), (1, 0, 0, 0), (1, 1, 1, 0),
(0, 0, 0, 0), (0, 1, 1, 1), (0, 0, 0, 1), (1, 1, 1, 1), (1, 0, 0, 0),
(1, 1, 1, 0), (0, 0, 0, 0), (0, 1, 1, 1)]

Encoded per blok 4 bit:
```

```
Blok 0: (0, 0, 0, 1)
(0, 0, 1, 0, 0, 1, 1)
Blok 1: (1, 1, 1, 1)
(1, 0, 0, 1, 0, 1, 1)
Blok 2: (1, 0, 0, 0)
(1, 1, 1, 0, 0, 0, 1)
Blok 3: (1, 1, 1, 0)
(1, 0, 1, 1, 0, 0, 0)
Blok 4: (0, 0, 0, 0)
(0, 0, 0, 0, 0, 0, 0)
Blok 5: (0, 1, 1, 1)
(0, 1, 1, 1, 0, 1, 0)
Blok 6: (0, 0, 0, 1)
(0, 0, 1, 0, 0, 1, 1)
Blok 7: (1, 1, 1, 1)
(1, 0, 0, 1, 0, 1, 1)
Blok 8: (1, 0, 0, 0)
(1, 1, 1, 0, 0, 0, 1)
Blok 9: (1, 1, 1, 0)
(1, 0, 1, 1, 0, 0, 0)
Blok 10: (0, 0, 0, 0)
(0, 0, 0, 0, 0, 0, 0)
Blok 11: (0, 1, 1, 1)
(0, 1, 1, 1, 0, 1, 0)
Blok 12: (0, 0, 0, 1)
(0, 0, 1, 0, 0, 1, 1)
Blok 13: (1, 1, 1, 1)
(1, 0, 0, 1, 0, 1, 1)
Blok 14: (1, 0, 0, 0)
(1, 1, 1, 0, 0, 0, 1)
Blok 15: (1, 1, 1, 0)
(1, 0, 1, 1, 0, 0, 0)
Blok 16: (0, 0, 0, 0)
(0, 0, 0, 0, 0, 0, 0)
Blok 17: (0, 1, 1, 1)
(0, 1, 1, 1, 0, 1, 0)
Received Blocks :  Blok 0: (0, 0, 0, 0, 0, 1, 1)
Hasil kali P invers =  (0, 1, 0, 0, 0, 0, 1)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 1, 0, 0)
Hasil kali S invers =  (0, 0, 0, 1)
Decoding Result :  (0, 0, 0, 1)
Received Blocks :  Blok 1: (1, 0, 1, 1, 0, 1, 1)
Hasil kali P invers =  (0, 1, 1, 0, 1, 1, 1)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 1, 1, 0)
Hasil kali S invers =  (1, 1, 1, 1)
Decoding Result :  (1, 1, 1, 1)
Received Blocks :  Blok 2: (1, 1, 0, 0, 0, 0, 1)
Hasil kali P invers =  (1, 1, 0, 0, 0, 1, 0)
Syndrome= (1, 0, 1)
Hasil koreksi : (1, 1, 0, 0)
Hasil kali S invers =  (1, 0, 0, 0)
Decoding Result :  (1, 0, 0, 0)
Received Blocks :  Blok 3: (1, 0, 0, 1, 0, 0, 0)
Hasil kali P invers =  (0, 0, 1, 0, 0, 1, 0)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 0, 1, 0)
Hasil kali S invers =  (1, 1, 1, 0)
Decoding Result :  (1, 1, 1, 0)
Received Blocks :  Blok 4: (0, 0, 1, 0, 0, 0, 0)
Hasil kali P invers =  (0, 0, 0, 0, 1, 0, 0)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 0, 0, 0)
Hasil kali S invers =  (0, 0, 0, 0)
```

```
Decoding Result :  (0, 0, 0, 0)
Received Blocks :  Blok 5: (0, 1, 0, 1, 0, 1, 0)
Hasil kali P invers =  (1, 0, 1, 0, 0, 0, 1)
Syndrome= (1, 0, 1)
Hasil koreksi : (1, 0, 1, 0)
Hasil kali S invers =  (0, 1, 1, 1)
Decoding Result :  (0, 1, 1, 1)
Received Blocks :  Blok 6: (0, 0, 0, 0, 0, 1, 1)
Hasil kali P invers =  (0, 1, 0, 0, 0, 0, 1)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 1, 0, 0)
Hasil kali S invers =  (0, 0, 0, 1)
Decoding Result :  (0, 0, 0, 1)
Received Blocks :  Blok 7: (1, 0, 1, 1, 0, 1, 1)
Hasil kali P invers =  (0, 1, 1, 0, 1, 1, 1)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 1, 1, 0)
Hasil kali S invers =  (1, 1, 1, 1)
Decoding Result :  (1, 1, 1, 1)
Received Blocks :  Blok 8: (1, 1, 0, 0, 0, 0, 1)
Hasil kali P invers =  (1, 1, 0, 0, 0, 1, 0)
Syndrome= (1, 0, 1)
Hasil koreksi : (1, 1, 0, 0)
Hasil kali S invers =  (1, 0, 0, 0)
Decoding Result :  (1, 0, 0, 0)
Received Blocks :  Blok 9: (1, 0, 0, 1, 0, 0, 0)
Hasil kali P invers =  (0, 0, 1, 0, 0, 1, 0)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 0, 1, 0)
Hasil kali S invers =  (1, 1, 1, 0)
Decoding Result :  (1, 1, 1, 0)
Received Blocks :  Blok 10: (0, 0, 1, 0, 0, 0, 0)
Hasil kali P invers =  (0, 0, 0, 0, 1, 0, 0)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 0, 0, 0)
Hasil kali S invers =  (0, 0, 0, 0)
Decoding Result :  (0, 0, 0, 0)
Received Blocks :  Blok 11: (0, 1, 0, 1, 0, 1, 0)
Hasil kali P invers =  (1, 0, 1, 0, 0, 0, 1)
Syndrome= (1, 0, 1)
Hasil koreksi : (1, 0, 1, 0)
Hasil kali S invers =  (0, 1, 1, 1)
Decoding Result :  (0, 1, 1, 1)
Received Blocks :  Blok 12: (0, 0, 0, 0, 0, 1, 1)
Hasil kali P invers =  (0, 1, 0, 0, 0, 0, 1)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 1, 0, 0)
Hasil kali S invers =  (0, 0, 0, 1)
Decoding Result :  (0, 0, 0, 1)
Received Blocks :  Blok 13: (1, 0, 1, 1, 0, 1, 1)
Hasil kali P invers =  (0, 1, 1, 0, 1, 1, 1)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 1, 1, 0)
Hasil kali S invers =  (1, 1, 1, 1)
Decoding Result :  (1, 1, 1, 1)
Received Blocks :  Blok 14: (1, 1, 0, 0, 0, 0, 1)
Hasil kali P invers =  (1, 1, 0, 0, 0, 1, 0)
Syndrome= (1, 0, 1)
Hasil koreksi : (1, 1, 0, 0)
Hasil kali S invers =  (1, 0, 0, 0)
Decoding Result :  (1, 0, 0, 0)
Received Blocks :  Blok 15: (1, 0, 0, 1, 0, 0, 0)
Hasil kali P invers =  (0, 0, 1, 0, 0, 1, 0)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 0, 1, 0)
```

```
Hasil kali S invers =  (1, 1, 1, 0)
Decoding Result :  (1, 1, 1, 0)
Received Blocks :  Blok 16: (0, 0, 1, 0, 0, 0, 0)
Hasil kali P invers =  (0, 0, 0, 0, 1, 0, 0)
Syndrome= (1, 0, 1)
Hasil koreksi : (0, 0, 0, 0)
Hasil kali S invers =  (0, 0, 0, 0)
Decoding Result :  (0, 0, 0, 0)
Received Blocks :  Blok 17: (0, 1, 0, 1, 0, 1, 0)
Hasil kali P invers =  (1, 0, 1, 0, 0, 0, 1)
Syndrome= (1, 0, 1)
Hasil koreksi : (1, 0, 1, 0)
Hasil kali S invers =  (0, 1, 1, 1)
Decoding Result :  (0, 1, 1, 1)

Decoded message: (0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1,
1, 0, 1, 0, 0, 1)
0.4342000484466553
```

RIWAYAT HIDUP

Moch Yuslih Abidin, lahir di Kabupaten Malang pada 6 September. Penulis merupakan anak pertama dari tiga bersaudara dari pasangan Bapak Yuliadi dan Ibu Anik Bidayati. Penulis telah menempuh pendidikan mulai TK Ahmad Yani, selanjutnya penulis melanjutkan ke SDN Genengan 1 pada tahun 2007 dan lulus pada tahun 2013. Selanjutnya penulis melanjutkan pendidikan ke SMPN 4 Kepanjen hingga tahun 2016 dan melanjutkan ke SMAN 6 Malang dan lulus pada tahun 2019. Selanjutnya penulis melanjutkan studi ke jenjang perguruan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang, Fakultas Sains dan Teknologi, Program Studi Matematika

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 : Moch Yuslih Abidin
NIM : 19610069
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Kriptositem McEliece Menggunakan Kode Hamming dan Kode Repetisi
Pembimbing I : Muhammad Khudzaifah, M.Si.
Pembimbing II : Erna Herawati, M.Pd.

No	Tanggal	Hal	Tanda Tangan
1.	29 Januari 2025	Konsultasi Bab I, II, dan III	1.
2.	21 Februari 2025	Konsultasi Bab I, II, dan III	2.
3.	14 Mei 2025	Konsultasi Bab I, II, dan III	3.
4.	18 Juni 2025	ACC Bab I, II, dan III	4.
5.	13 Agustus 2025	Konsultasi Kajian Agama Bab I dan II	5.
6.	20 Agustus 2025	Konsultasi Kajian Agama Bab I dan II	6.
7.	27 Agustus 2025	Konsultasi Kajian Agama Bab I dan II	7.
8.	26 September 2025	ACC Kajian Agama Bab I dan II	8.
9.	7 November 2025	ACC Seminar Proposal	9.
10.	12 November 2025	Konsultasi Revisi Seminar Proposal	10.
11.	14 November 2025	Konsultasi Bab IV	11.
12.	18 November 2025	Konsultasi Bab IV	12.
13.	20 November 2025	Konsultasi Bab IV dan V	13.
14.	21 November 2025	ACC Bab IV dan V	14.
15.	24 November 2025	Konsultasi Kajian Agama Bab IV	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.	26 November 2025	Konsultasi Kajian Agama Bab IV	16.
17.	27 November 2025	ACC Kajian Agama Bab IV	17.
18.	28 November 2025	ACC Seminar Hasil	18.
19.	9 Desember 2025	Konsultasi Revisi Seminar Hasil	19.
20.	16 Desember 2025	Sidang Skripsi	20.
21	23 Desember 2025	ACC Keseluruhan	21.

Malang, 23 Desember 2025
Mengetahui,
Ketua Program Studi Matematika

Dr. Fachrur Rozi, M.Si.
NIP. 19800527 200801 1 012