Prosiding Seminar Nasional Sains dan Teknologi Seri 02
Fakultas Sains dan Teknologi, Universitas Terbuka
Vol. 1 No. 2 (2024)
e-ISSN: 3047-6569

PENERAPAN KODE HAMMING PADA KRIPTOSISTEM MCELIECE

Kusuma Widyawati*, Putranto Hadi Utomo

Program Studi Matematika, Universitas Sebelas Maret, Surakarta

**Penulis Korespondensi*: kusuma_widya21@student.uns.ac.id

ABSTRAK

Komunikasi di era modern ini menjadi lebih mudah dengan adanya teknologi modern berupa *smartphone*, laptop maupun komputer. Seiring dengan berkembangnya teknologi, seringkali terjadi penyalahgunaan. Salah satunya adalah penyadapan dan kebocoran data penting yang dilakukan oleh pihak ketiga. Oleh karena itu, digunakan kriptosistem McEliece yang dapat mengamankan pesan dari serangan komputer kuantum. Kriptosistem McEliece memanfaatkan *error correcting codes* (kode koreksi kesalahan) dalam penerapannya. Penelitian ini bertujuan untuk mengkaji penerapan kode Hamming pada kriptosistem McEliece. Penelitian ini dilakukan dengan tiga tahapan yaitu pembangkitan kunci, proses enkripsi pesan, dan proses dekripsi sekaligus pengkoreksian *error*. Pada penelitian ini ditetapkan suatu kode Hamming dengan $n = 7$ dan $k = 4$, matriks non-singular yang dinotasikan dengan S berukuran 4×4, matriks permutasi yang dinotasikan dengan P berukuran 7×7 dan pesan yang dinotasikan dengan m dengan panjang 8-bit. Penelitian ini menghasilkan *chipertext* dan *plaintext* yang diperoleh dari proses proses enkripsi dan dekripsi pesan menggunakan kode Hamming pada kriptosistem McEliece. Dengan menerapkan kode Hamming pada algoritma McEliece dapat meningkatkan keamanan pertukaran informasi, serta menghadapi potensi ancaman dari komputer kuantum.

Kata kunci: komunikasi, kriptosistem McEliece, kode Hamming, enkripsi, dekripsi.

1 PENDAHULUAN

Salah satu aspek penting dalam kehidupan sehari-hari adalah komunikasi. Komunikasi merupakan proses penyampaian pesan atau interaksi dari pengirim kepada penerima. Oleh karena itu, komunikasi harus ada timbal balik (*feedback*) antara komunikator dengan komunikan. Komunikasi dapat dilakukan secara langsung maupun tidak langsung. Di era modern ini, komunikasi dapat dilakukan dengan mengirimkan pesan melalui *smartphone*, laptop maupun komputer dengan bantuan internet. Dengan bantuan internet, segala informasi dapat diakses dengan mudah. Akan tetapi, seringkali terjadi penyalahgunaan teknologi, salah satunya adalah penyadapan dan kebocoran informasi sehingga data-data penting dapat diakses orang lain dengan mudah. Oleh karena itu, perlu dipelajari ilmu untuk mengamankan data, yaitu kriptografi.

Kriptografi merupakan sebuah studi yang mempelajari tentang pengiriman pesan (yang memiliki kemungkinan diretas oleh lawan) secara aman. Gagasan utama dibalik kriptografi yaitu pengirim pesan memilih pesan yang akan dikirimkan, mengaplikasikan serangkaian proses enkripsi (mengubah teks menjadi rangkaian *ciphpertext*), dan mengirimkan pesan terenkripsi ini melalui saluran (*channel*). Penerima kemudian memperoleh pesan terenkripsi tersebut (yang juga disebut sebagai *ciphpertext*), dan menggunakan proses dekripsi yang diketahuinya untuk memulihkan menjadi pesan asli dari pengirim.

Jika lawan mencuri pesan yang terenkripsi, dia tidak akan bisa memulihkan pesan asli tanpa mengetahui proses dekripsi yang rahasia. Proses enkripsi dan dekripsi biasanya melibatkan penggunaan potongan informasi rahasia, dikenal sebagai kunci privat (*private key*), yang dapat digunakan untuk melakukan tugas ini. Sebagian besar orang beranggapan bahwa enkripsi dan dekripsi merupakan operasi yang saling berkebalikan.

Kriptosistem McEliece merupakan kriptosistem berbasis kode yang diusulkan oleh Robert Mceliece pada tahun 1978. Kriptosistem McEliece adalah kriptosistem kunci publik (*public key*) yang memanfaatkan *error-correcting codes* sebagai metode enkripsi. Kriptosistem ini dianggap aman digunakan era komputer kuantum (McEliece. 1978). Komputer kuantum adalah komputer yang dijalankan berdasarkan hukum mekanika kuantum dan mampu menyelesaikan perhitungan besar dengan cepat (Bernstein, Buchmann, dan Dahmen, 2009). Terdapat beberapa kode yang dapat digunakan pada kriptosistem McEliece salah satunya yaitu menggunakan kode Hamming. Kode Hamming adalah sebuah sistem kode kesalahan yang dapat mendeteksi dan memperbaiki kesalahan saat data disimpan atau diterima. Kode ini dikembangkan oleh Richard W. Hamming dan digunakan dalam berbagai aplikasi, termasuk telekomunikasi, jaringan komputer, dan sistem penyimpanan data (Ples, 1998).

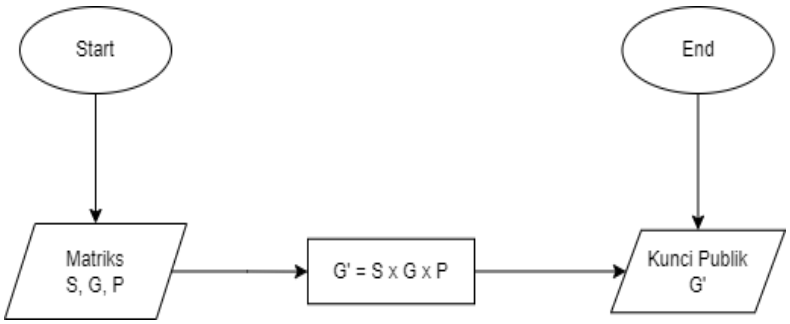
2 METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini yaitu kajian pustaka dengan menggunakan beberapa referensi berupa buku, tesis, artikel, jurnal serta skripsi mengenai kode Hamming, kriptosistem McEliece. Terdapat 3 langkah yang dilakukan dalam penelitian ini, yaitu pembangkitan kunci, proses enkripsi pesan, dan proses dekripsi dan koreksi *error*.

2.1 Pembangkitan Kunci

Langkah-langkah dalam pembangkitan kunci sebagai berikut.

a. Pengirim memilih sebuah kode linear $[n, k]$ yang dapat memperbaiki t *error*, dalam penelitian ini akan menggunakan kode Hamming.
b. Bangkitkan matriks generator berukuran $k \times n$.
c. Pengirim secara acak memilih matriks invertibel S biner dengan ukuran $k \times k$.
d. Pengirim secara acak memilih matriks permutasi P dengan ukuran $n \times n$.
e. Pengirim menghitung matriks G' dengan ukuran $k \times n$. $G' = S \times G \times P$.

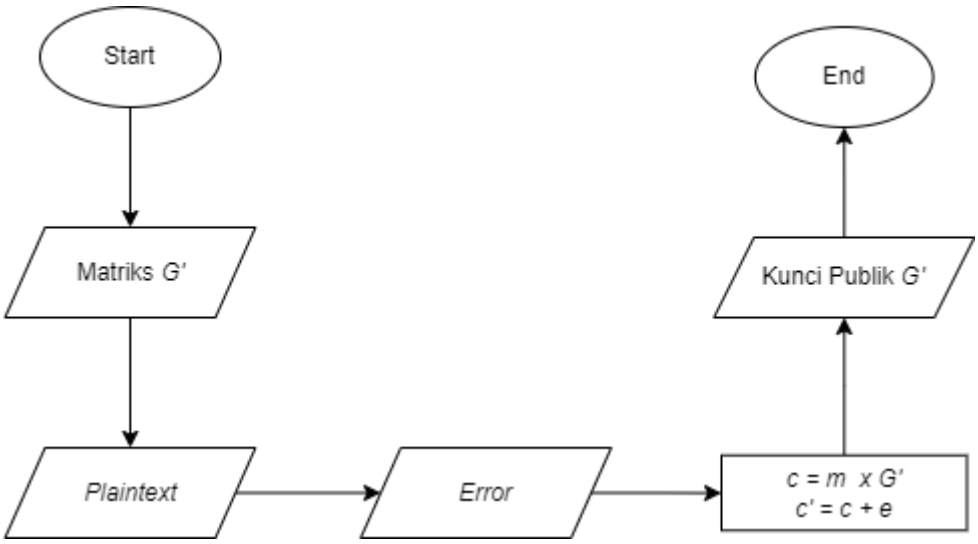


Gambar 1. *Flowchart* pembangkitan kunci

2.2 Proses Enkripsi Pesan

Langkah-langkah dalam proses enkripsi pesan sebagai berikut.

a. Pengirim mengubah *plain text* ke bentuk *binary* berdasarkan tabel ASCII.
b. Pengirim membagi *binary* ke dalam blok blok dengan panjang 4-bit.
c. Pengirim menghitung vektor $c_i = m_i \times G'$.
d. Pengirim secara acak menghasilkan vektor e berukuran n-bit yang memiliki t elemen non-nol (vektor dengan panjang n dan bobot t).
e. Pengirim menghitung *chipertext* $c_i' = c_i + e$

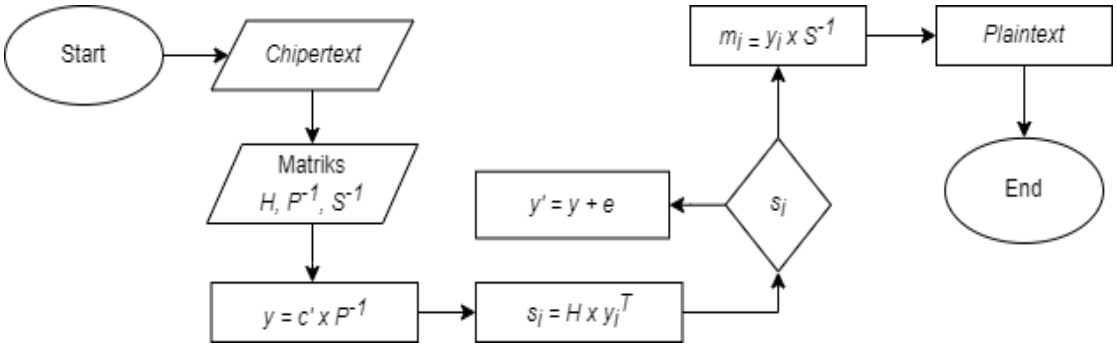


Gambar 2. *Flowchart* proses enkripsi pesan

2.3 Proses Deskripsi dan Koreksi *Error*

Langkah - langkah dalam pendeskripsian pesan dan koreksi *error* sebagai berikut.

a. Penerima menghitung invers dari matriks P.
b. Penerima menghitung $y_i = c_i'.P^{-1}$.
c. Penerima menghitung matriks *parity check* H.
d. Kemudian penerima memeriksa *error* dengan mencari *syndrome* dari $y_i(s_i) = H.y_i^T$). jika $s_i = 0$ maka tidak terjadi *error* pada y_i, jika $s_i \neq 0$ ini menujukkan terjadi *error* pada y_i.
e. Selanjutnya penerima mencari letak bit yang *error* dengan melihat matriks H. Jika s_i sama dengan kolom ke-i pada matriks H, maka *error* terjadi pada bit ke-i dari biner y_i.
f. Tahap berikutnya adalah mengoreksi pesan y_i dengan menambahkan *error* e dengan bobot 1 pada bit yang salah. $y_i' = y_i + e'$.
g. Setelah y_i dikoreksi dan menghasilkan y_i' kemudian mengambil 4-bit dari kiri pada *binary* y_i'.
h. Penerima menghitung invers S.
i. Penerima mengoperasikan $y_i'.S^{-1}$. Didapatkan pesan $m_i' = y_i'.S^{-1}$.
j. Penerima menggabungkan blok-blok sesuai urutan sehingga memiliki panjang 8-bit. Pesan diubah ke bentuk teks berdasarkan tabel ASCII



Gambar 3. *Flowchart* proses deskripsi dan koreksi *error*

3 HASIL DAN PEMBAHASAN

Pada bab ini diberikan hasil dan pembahasan penerapan kode Hamming pada kriptosistem McEliece.

3.1 Pembangkitan Kunci

Pada proses pembangkitan kunci, pertama-tama, pengguna menggunakan metode kode Hamming, untuk menentukan parameter kunci (n) dan (k), yang pada contoh ini diberikan nilai $(n = 7)$ dan $(k = 4)$. Selanjutnya, matriks generator (G) ditentukan sebagai berikut.

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

Setelah itu, dilakukan pembangkitan matriks (S) secara acak dengan dimensi $(k \times k)$ dan matriks (P) dengan dimensi $(n \times n)$.

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

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

Kemudian lakukan perkalian matriks SGP untuk memperoleh nilai matriks G'.

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

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

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

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

Dengan demikian diperoleh matriks $G' = \begin{bmatrix}0&0&0&1&1&1&1\\0&0&1&1&0&1&0\\1&0&0&0&0&1&1\\0&1&0&1&0&0&1\end{bmatrix}$.

3.2 Proses Enkripsi Pesan

Pada bagian ini, pengirim mengubah pesan teks ke bentuk biner berdasarkan tabel ASCII dengan panjang 8 bit. Sebagai contoh pengirim akan mengirimkan pesan "CIA", berdasarkan tabel ASCII didapatkan pesan CIA $= 010000110100100101000001$.

$$C = [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1]$$

$$I = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1]$$

$$A = [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$$

Kemudian bagi kode biner tersebut menjadi beberapa blok m_i dengan panjang 4 –bit.

$$m_1 = [0 \quad 1 \quad 0 \quad 0]$$
$$m_2 = [0 \quad 0 \quad 1 \quad 1]$$
$$m_3 = [0 \quad 1 \quad 0 \quad 0]$$
$$m_4 = [1 \quad 0 \quad 0 \quad 1]$$
$$m_5 = [0 \quad 1 \quad 0 \quad 0]$$
$$m_6 = [0 \quad 0 \quad 0 \quad 1]$$

Selanjutnya, pesan yang telah dibagi menjadi beberapa blok m_i dikalikan dengan matriks generator G' untuk memperoleh matriks c_i.

$$c_1 = m_1 \times G' = [0 \quad 1 \quad 0 \quad 0] \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0]$$

$$c_2 = m_2 \times G' = [0 \quad 0 \quad 1 \quad 1] \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$= [1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0]$$

$$c_3 = m_3 \times G' = [0 \quad 1 \quad 0 \quad 0] \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0]$$

$$c_4 = m_4 \times G' = [1 \quad 0 \quad 0 \quad 1] \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0]$$

$$c_5 = m_5 \times G' = [0 \quad 1 \quad 0 \quad 0] \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0]$$

$$c_6 = m_6 \times G' = [0 \quad 0 \quad 0 \quad 1] \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1]$$

Kemudian c_i ditambahkan dengan *error* e dengan panjang 7 bit, *error* e dipilih secara acak. Pada contoh ini diberikan $e = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$. Dengan rumus $c_i' = c_i + e$ maka diperoleh nilai c_i' sebagai berikut.

$$c_1' = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0] + [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0]$$

$$c_2' = [1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0] + [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0]$$

$$c_3' = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0] + [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0]$$

$$c_4' = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0] + [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] = [1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0]$$

$$c_5' = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0] + [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0]$$

$$c_6' = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1] + [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] = [1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1]$$

Sehingga *chipertext* CIA = 101101001010101011010110011010110101101001.

3.3 Proses Dekripsi dan koreksi *Error*

Setelah memperoleh nilai c_i' selanjutnya pada proses dekripsi dan koreksi *error* menghitung matriks P^{-1}.

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

Setelah diperoleh matriks P^{-1} kemudian mengalikan nilai c_i' dengan matriks P^{-1} untuk memperoleh matriks y_i.

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

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

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

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

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

$$= [0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1]$$

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

$$= [1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0]$$

Selanjutnya, memasuki tahap koreksi *error* dengan mencari nilai s_i. Untuk mencari nilai s_i dilakukan dengan rumus berikut.

$$s_i = H \times y_i^T$$

Matriks *parity check* diperoleh dari matriks $G = [I_4 X]$, di mana I_4 adalah matriks identitas berukuran 4×4 dan X adalah matriks berukuran 4×3. Dan matriks $H = [Y \; I_3]$ dimana $Y = X^T$. Sehingga didapatkan matriks H berukuran 3×7.

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

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

Setelah diperoleh matriks *parity check* H, selanjutnya akan dicari s_i dengan mengalikan mtariks *parity check* H dengan *transpose* dari y_i.

$$s_1 = H \times y_1^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s_2 = H \times y_2^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s_3 = H \times y_3^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s_4 = H \times y_4^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s_5 = H \times y_5^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s_6 = H \times y_6^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Jika nilai $s_i \neq 0$ maka terjadi *error* pada y_i , sehingga untuk mencari letak bit yang salah dengan melihat matriks H. Sedangkan, jika s_i sama dengan kolom matriks H, maka *error* terjadi pada bit ke-*i* dari *binary* y.

$S(y_1) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ = nilai yang sama dengan kolom ke-3 pada matriks H, maka *error* pada y_1 terletak di bit ke-3.

$S(y_2) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ = nilai yang sama dengan kolom ke-3 pada matriks H, maka *error* pada y_2 terletak di bit ke-3.

$S(y_3) = \begin{bmatrix}0\\1\\1\end{bmatrix}$ = nilai yang sama dengan kolom ke-3 pada matriks H, maka *error* pada y_3 terletak di bit ke-3.

$S(y_4) = \begin{bmatrix}0\\1\\1\end{bmatrix}$ = nilai yang sama dengan kolom ke-3 pada matriks H, maka *error* pada y_4 terletak di bit ke-3.

$S(y_5) = \begin{bmatrix}0\\1\\1\end{bmatrix}$ = nilai yang sama dengan kolom ke-3 pada matriks H, maka *error* pada y_5 terletak di bit ke-3.

$S(y_6) = \begin{bmatrix}0\\1\\1\end{bmatrix}$ = nilai yang sama dengan kolom ke-3 pada matriks H, maka *error* pada y_6 terletak di bit ke-3.

Setelah letak *error* ditemukan, selanjutnya mengkoreksi *error* dengan menambahkan *error* e' dengan bobot 1 pada bit yang salah.

$$y_1' = y_1 + e' = [0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0] + [0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0]$$
$$= [0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1]$$

$$y_2' = y_2 + e' = [1\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1] + [0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0]$$
$$= [1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1]$$

$$y_3' = y_3 + e' = [0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1] + [0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0]$$
$$= [0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1]$$

$$y_4' = y_4 + e' = [1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1] + [0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0]$$
$$= [1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 1]$$

$$y_5' = y_5 + e' = [0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1] + [0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0]$$
$$= [0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1]$$

$$y_6' = y_6 + e' = [1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 0] + [0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0]$$
$$= [1\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0]$$

Setelah proses koreksi diperoleh y_i', selanjutnya mengambil 4 bit *binary* y_i' dari kiri untuk dikalikan dengan matriks S^{-1} untuk memperoleh m_i'.

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

$$m_1' = y_1' \times S^{-1} = [0\ \ 1\ \ 0\ \ 0] \times \begin{bmatrix}0 & 0 & 0 & 1\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\1 & 0 & 0 & 0\end{bmatrix} = [0\ \ 1\ \ 0\ \ 0]$$

$$m_2' = y_2' \times S^{-1} = [1 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 1 \quad 1]$$

$$m_3' = y_3' \times S^{-1} = [0 \quad 1 \quad 0 \quad 0] \times \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = [0 \quad 1 \quad 0 \quad 0]$$

$$m_4' = y_4' \times S^{-1} = [1 \quad 0 \quad 0 \quad 1] \times \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = [1 \quad 0 \quad 0 \quad 1]$$

$$m_5' = y_5' \times S^{-1} = [0 \quad 1 \quad 0 \quad 0] \times \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = [0 \quad 1 \quad 0 \quad 0]$$

$$m_6' = y_6' \times S^{-1} = [1 \quad 0 \quad 0 \quad 0] \times \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = [0 \quad 0 \quad 0 \quad 1]$$

Diperoleh hasil dari proses dekripsi, yaitu 010000110100100101000001. Selanjutnya pesan biner dibagi ke dalam blok-blok dengan panjang 8-bit untuk mengubah pesan biner ke dalam bentuk pesan teks. Berdasarkan tabel ASCII diperoleh $01000011 - 01001001 - 01000001 = \text{CIA}$.

4 KESIMPULAN

Dalam proses pembentukan matriks generator G' diperlukan perkalian dari matriks non-singular S, matriks generator G dan matriks permutasi P. Kemudian, pesan yang berupa teks CIA diubah menjadi bentuk biner dengan panjang 8-bit lalu dibagi menjadi 6 dengan panjang 4-bit (m_i) . Selanjutnya, dicari *chipertext* dan diperoleh pesan CIA $= 101101001010101011010110011010110101101001$. Kemudian dengan proses dekripsi diperoleh *plaintext* $01000011 - 01001001 - 01000001 = \text{CIA}$. Penerapan kode Hamming pada algoritma McEliece dapat meningkatkan keamanan dalam melakukan komunikasi serta melindungi dari ancaman komputer kuantum. Saran untuk penelitian selanjutnya, diharap dapat melakukan penerapan pada *software* python.

UCAPAN TERIMA KASIH

Saya mengucapkan terima kasih kepada Pak Putranto Hadi Utomo selaku dosen pembimbing saya dan juga teman-teman saya yang membantu dalam proses pengerjaan artikel ini. Tak lupa saya mengucapkan terima kasih kepada orangtua saya yang selalu memberi dukungan kepada saya.

DAFTAR PUSTAKA

Falakh, M. F. (2023). Implementasi Kode Hamming pada Algoritma McEliece untuk Mengamankan Pesan. Skripsi. Universitas Islam Negeri Maulana Malik Ibrahim.

Bernstein, D. J., Buchmann, J., & Dahmen, E. (2009). *Post-Quantum Cryptography*. Berlin: Springer.

McEliece, R. J. (1978). A Public Key Cryptosystem Based on Algebraic Coding Theory. DSN Progress Report,42-44.

Menezes, A., Oorschot, P. dan Vanstone, S. (1997). *Handbook of Applied Cryptography*. Florida: CRC Press.

Utami, R. P. (2023). Penerapan Kode Hammard pada Kriptosistem McEliece. Skripsi. Universitas Sebelas Maret.

Oktavia, R. E. (2023). Kriptografi Berbasis Teori Koding: Kriptosistem McEliece. Skripsi. Universitas Sebelas Maret.

Pless, V. (1998). Introduction to the Theory of Error-Correcting Code, Third Edition. Chicago: A Wiley Interscience Publication.

Irawanto, B. and Widyaningsih, S. (2009). Deteksi dan Koreksi *Error* pada Pesan Digital dengan Kode Hamming. *Jurnal Sains dan Matematika*, 17(3), 127-130.

Ilmiyah N. F. (2018). Kajian tentang Kriptosistem McEliece dalam Menghadapi Tantangan Komputer Kuantum di Era Revolusi Industri 4.0. Prosiding Seminar Nasional MIPA 2018. Institut Agama Islam Negeri Kediri, Kediri.

Roering, C. (2013). Coding Theory-Based Cryptography: McEliece Cryptosystems in Sage. Tesis. Saint John's University.