Prosiding Pendidikan Matematika, Matematika, dan Statistika Volume 7 (2023)
e-ISSN No. 2721-6802

Error Detection dan *Error Correction* pada Komunikasi Digital Menggunakan Hamming *Code*

Hafizhah[1], Putranto Hadi Utomo[2]
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret[1, 2]
hafizhahfifi110@gmail.com[1]

***Abstrak*—**Peran komunikasi digital menjadi aspek penting dalam kehidupan manusia untuk tetap terkoneksi. Dalam mengirimkan suatu pesan digital berisi informasi berupa teks, gambar, audio, maupun video melalui *noisy channel*, terdapat kemungkinan pesan akan mengalami *error* sehingga pesan menjadi tidak *reliable*. Oleh karena itu, *coding theory* untuk menjawab fenomena tersebut. Dalam *coding theory*, dibahas dua proses penting, yaitu *encoding* dan *decoding*. *Sender* melakukan *encoding* sebelum mentransmisikan pesan kepada *receiver*. Kemudian, *receiver* melakukan *decoding* sehingga pesan dapat diterima dengan baik. Pada proses *decoding* terdapat dua aspek penting, yaitu *error detection* dan *error correction*. Dalam artikel ini, akan dibahas Hamming *code* untuk melakukan *error detection* dan *error correction* dengan menggunakan *parity-check matrix*. Tujuan penelitian ini, yaitu untuk mengetahui cara kerja *error detection* dan *error correction* pada komunikasi digital menggunakan Hamming *code*, membentuk contoh perhitungan, serta membentuk program sederhana menggunakan *SageMath* 9.3. Subjek penelitian ini merupakan pesan digital berupa teks 'GOLDEN' yang ditransmisikan *sender* ke *receiver*. Metode penelitian ini menggunakan studi kepustakaan. Proses *encoding* dilakukan dengan mengalikan pesan *binary digits* dengan *generator matrix* G untuk memperoleh *codeword*. E*rror detection* dilakukan dengan mengalikan *codeword* dengan *transposed parity-check matrix* H^T untuk memperoleh *syndrome*. Sedangkan *error correction*, dilakukan dengan melakukan operasi XOR *received codeword* dengan *error* untuk memperoleh *corrected codeword*. Dalam program sederhana menggunakan *SageMath* 9.3, diperoleh hasil bahwa Hamming *code* dapat melakukan *error detection* maksimum dua *errors* dan *error correction* maksimum satu *error* (dilihat dari hasil pesan teks setelah proses *encoding*).

Kata kunci: Error Detection, Error Correction, Komunikasi Digital, Hamming Code

I. PENDAHULUAN

Tak dapat dielakkan, komunikasi menjadi aspek yang tak dapat terlepaskan dalam berkehidupan. Peran komunikasi tidak hanya sebagai kebutuhan sekunder ataupun tersier, melainkan kebutuhan primer bahkan suatu keharusan yang harus tersedia. Komunikasi merupakan suatu interaksi agar saling terkoneksi antara pihak satu dengan pihak lain untuk mengemukakan dan bertukar informasi berupa data, teks, video, maupun suara melalui media tertentu sebagai perantara. Peradaban membawa komunikasi dari era konvensional ke modern sampai digital. Komunikasi digital mencakup terjadinya transmisi pesan yang berisi informasi melalui suatu *channel* dari pihak satu ke pihak lain. Dalam suatu komunikasi digital, melibatkan dua pihak yang saling berinteraksi, yaitu pihak *sender* dan *receiver*. Menurut Utomo [9], fokus perhatian dalam transmisi pesan berisi informasi adalah integritas pesan, hal ini menjadi penting bagi pihak *sender* untuk berkomunikasi dengan pihak *receiver* supaya pesan yang ditransmisikan sama seperti pesan yang diterima. Pesan yang ditransmisikan oleh pihak *sender* memiliki kemungkinan terjadi *error* karena melalui *noisy channel* sehingga menyebabkan disinformasi (pesan yang ditransmisikan tidak sama dengan pesan yang diterima). Dalam hal ini, informasi yang ditransmisikan dalam bentuk *binary* berisi suatu urutan yang terdiri dari 0 dan 1. *Error* menyebabkan 0 berubah menjadi 1 atau sebaliknya. Oleh karena itu, untuk mencapai transmisi pesan yang *reliable*, diperlukan *coding theory* guna melakukan tujuan utama yaitu *error detection* dan *error correction* pada pesan. Terdapat dua proses penting dalam *coding theory*, yaitu *encoding* dan *decoding*. Pada proses *encoding*, pesan yang akan ditransmisikan akan dikonversikan dahulu ke dalam *code* tertentu oleh pihak *sender*. Kemudian, *code* yang akan diterima oleh pihak *receiver* dikembalikan ke dalam bentuk asli pesan pada proses *decoding*. Pada tahun 1950, Richard Wesley Hamming menemukan *code* yang disebut Hamming *code* untuk melakukan *error detection* dan *error correction* dengan menggunakan *parity-check matrix* [12]. Menurut Lin dan Junior [14], Hamming *code* merupakan *linear code* untuk *error correction* yang sudah digunakan sangat luas untuk *error control* pada komunikasi digital dan sistem penyimpanan data. Pada penelitian ini akan dibahas terkait cara kerja *error detection* dan *error correction* pada komunikasi digital menggunakan Hamming *code*, contoh perhitungan, serta program sederhana

menggunakan *SageMath* versi 9.3. Tujuan penelitian ini, yaitu untuk mengetahui cara kerja *error detection* dan *error correction* pada komunikasi digital menggunakan Hamming *code*, membentuk contoh perhitungan, serta membentuk program sederhana menggunakan *SageMath* versi 9.3. Dengan penelitian ini, diharapkan dapat menambah pengetahuan dan berperan dalam *reliability* pentransmisian pesan digital melalui *noisy channel* serta dapat menjadi ekspansi ilmu matematika pada bidang *coding*.

II. METODE PENELITIAN

Metode penelitian ini terbagi menjadi dua bagian, yaitu: subjek penelitian dan langkah penelitian.

A. *Subjek Penelitian*

Subjek penelitian ini merupakan pesan digital berupa teks yang ditransmisikan melalui *noisy channel*. Dalam hal ini, digunakan 16 karakter huruf dari A sampai P $(A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P)$ dan indeks berbentuk bilangan *decimal* dari 0 sampai 15 $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$. Setiap bilangan akan merepresentasikan suatu huruf, yaitu: A direpresentasikan oleh 0, B direpresentasikan oleh 1, C direpresentasikan oleh 2, dan seterusnya sampai dengan P direpresentasikan oleh 15. Kemudian, dibentuk kata 'GOLDEN', nantinya akan menjadi pesan yang akan ditransmisikan oleh *sender* ke *receiver*.

B. *Langkah Penelitian*

Berikut merupakan langkah penelitian yang dilakukan.

1. Mempelajari dan memahami konsep cara kerja tentang *error detection* dan *error correction* pada komunikasi digital menggunakan Hamming *code* dengan studi kepustakaan dari berbagai *reference* berupa buku, artikel, *lecture note*, skripsi, dan tesis.
2. Membentuk algoritma yang digunakan untuk *error detection* dan *error correction* pada komunikasi digital menggunakan Hamming *code*.
3. Membentuk contoh perhitungan *error detection* dan *error correction* pada komunikasi digital menggunakan Hamming *code* berdasarkan algoritma yang dibentuk.
4. Membentuk program sederhana *error detection* dan *error correction* menggunakan Hamming *code* pada *SageMath* versi 9.3.

III. HASIL DAN PEMBAHASAN

A. *Algoritma Error Detection dan Error Correction pada Komunikasi Digital Menggunakan Hamming Code*

Menurut Mshelia et al. [2], sistem komunikasi digital memegang peranan yang sangat penting dalam meningkatnya permintaan untuk komunikasi data. Pada sistem komunikasi, ketika data ditransmisikan, *error* dapat terjadi karena *noise* yang tidak diinginkan dan interferensi dari *channel* komunikasi. Sistem komunikasi digital berurusan dengan informasi dalam bentuk data, video, atau suara untuk pola transmisi satu titik ke yang lainnya. Selama transmisi, sinyal digital sangat terdistorsi akibat dari *error* yang disebabkan oleh berbagai *noise*. Menurut Jibril [6], komunikasi digital melibatkan transmisi sinyal pesan dalam bentuk digital. Menurut Aydin [8], data digital ditransmisikan melalui suatu *noisy channel* (bisa melalui suatu kabel, *network*, ruang, udara, dsb) dan sering adanya *noise* pada *channel*. *Noise* tersebut dapat mendistorsi pesan yang akan ditransmisikan. Oleh karena itu, apa yang diterima oleh *receiver* mungkin tidak sama dengan yang ditransmisikan oleh *sender*.

Menurut Jibril [6], bidang *coding theory* lahir dari karya penting Claude E. Shannon (1948) tentang komunikasi digital. Karya Shannon menjelaskan tentang seberapa cepat, efisien, dan andal sistem komunikasi digital dapat melalui channel transmisi yang paling signifikan. Karya Shannon secara khusus berfokus pada bagaimana untuk mentransmisikan data dengan andal melalui *channel* dimana terdapat *noise* yang dapat mengubah sinyal. Fokus pengamatan adalah bahwa dimungkinkan untuk meningkatkan keandalan sistem komunikasi digital secara signifikan dengan menggunakan skema *error correction* yang efisien. Sinyal digital yang akan ditransmisikan akan melalui proses *encoding* dengan *code* yang telah ditentukan pada transmisi dan kemudian melalui proses decoding untuk diterjemahkan oleh *receiver*. Menurut Singh [3], *coding theory* adalah bidang studi yang berkaitan dengan transmisi data melalui *noisy channel* dan pemulihan pesan yang rusak. Dengan istilah lain, *coding theory* berkaitan dengan pencapaian

Prosiding Pendidikan Matematika, Matematika, dan Statistika Volume 7 (2023)
e-ISSN No. 2721-6802

transmisi informasi yang andal dan efisien melalui *noisy channel*. Tujuan dari *coding theory* adalah untuk *encoding* informasi sehingga meskipun terdapat *error* dalam transmisi data, *receiver* dapat melakukan *error correction* dan memperoleh informasi asli yang ditransmisikan. Objek dari *coding theory* adalah transmisi dari pesan melalui *noisy channel*. Menurut Roering [1], bidang *coding theory* yaitu meliputi studi tentang pelestarian informasi, kompresi informasi, dsb. Pada bagian ini yang menjadi fokus pembahasan adalah pelestarian informasi yang terdiri dari *error detection* dan *error correction* dari aspek *coding theory*.
Diberikan beberapa definisi terkait *coding theory* oleh Udomkavanich [11], Hirschfeld [5], dan [16].

Definisi 1. Misalkan $A = \{a_1, a_2, \cdots, a_q\}$ merupakan suatu himpunan dengan *size* q, maka A disebut *code alphabet* dan elemen di dalamnya disebut *code symbol*.

(i) Suatu himpunan A^n didefinisikan sebagai $A^n = \{w_1 w_2 \cdots w_n | w_i \in A\}$, suatu elemen $w = w_1 w_2 \cdots w_n \in A^n$ disebut q-*ary word* dengan *length* n atas A. Secara ekuivalen, $w = w_1 w_2 \cdots w_n$ dapat dianggap sebagai *vector* $w_1 w_2 \cdots w_n$.
(ii) q-*ary block code* dengan *length* n atas A merupakan suatu himpunan tak kosong $\mathcal{C}$ dari A^n .
(iii) Elemen dari $\mathcal{C}$ disebut *codeword* dalam $\mathcal{C}$.
(iv) Banyaknya *codewords* dalam $\mathcal{C}$, dinyatakan sebagai $|\mathcal{C}|$, disebut sebagai *size* dari $\mathcal{C}$.
(v) Banyaknya *symbols* dalam *word* disebut *length*.
(vi) *Information rate* dari *code* $\mathcal{C}$ dengan *length* n didefinisikan sebagai $(log_q)|\mathcal{C}|/n$.
(vii) *Code* dengan *length* n dan *size* k disebut (n, k)–*code.*

Definisi 2. *Binary code* (2-*ary*) merupakan suatu himpunan yang berisikan urutan dari 0 dan 1, setiap urutan merupakan *codeword*.

Definisi 3. *Binary code* diperoleh dari mengambil 2^k *words* dengan *length* k dan menambahkan $(n - k)$ c*heck bits* pada setiap *word* sehingga memberikan *codeword* dengan *length* n. Suatu *code* dengan *length* n dan 2^k*codewords* disebut (n, k) *code*. Banyaknya k disebut d*imension* dari *code*, dikatakan bahwa suatu *code* memiliki k *message bits*.

Generator matrix dan *parity-check matrix* merupakan *matrix* yang berperan penting dalam *coding theory*. Berikut diberikan definisi *dual code* oleh Kosek [10] dan definisi *generator matrix, parity-check matrix*, serta *standard form* dari kedua *matrix* tersebut oleh Lindell [15].
Definisi 4. Misalkan L merupakan (n, k)–*code*. Himpunan

$$L^{\perp} = \{x \in \mathbb{F}_q^n \,|\, x \cdot c = 0, c \in L\}$$

disebut *dual code* dari L.

Definisi 5. Definisi *generator matrix* dan *parity-check matrix* adalah sebagai berikut.

(i) *Generator matrix* $\mathcal{G}$ untuk *linear code* $\mathcal{C}$ merupakan matrix yang mana barisnya berupa *basis* untuk $\mathcal{C}$.
(ii) *Parity-check matrix* H untuk $\mathcal{C}$ merupakan *generator matrix* untuk *dual code* $\mathcal{C}^{\perp}$.

Definisi 6. *Standard form generator matrix* dan *parity-check matrix* adalah sebagai berikut.

(i) *Generator matrix* disebut sebagai *standard form* jika memiliki *form* $[I_k|X]$ dimana I_k merupakan $k \times k$ *identity matrix*.
(ii) *Parity-check matrix* disebut sebagai *standard form* jika memiliki *form* $[Y\,|I_{n-k}]$ dimana $Y = -X^T$.

Menurut Kashkoulli [7], Hamming *code* merupakan suatu metode sederhana dari *error detection* dan *error correction* yang sering digunakan. Hamming *code* melakukan pengecekan *error* untuk semua *bits* yang berada di *codeword*. Hamming *code* disebut *linear code* untuk *error detection* dan *error correction* yang dapat melakukan detection maksimal dua *bit errors* dan dapat melakukan correction maksimal satu *bit error*. Diberikan definisi Hamming *code* oleh Bowman [4] dan contoh Hamming *code* oleh Kosek [10].

Definisi 7. Diberikan bilangan integer $r \geq 2$, dengan $n = (q^r - 1)/(q - 1)$. Hamming *code* $Ham(r, q)$ merupakan suatu *linear* $[n, n - r]$ *code* dalam $\mathbb{F}_q^n$ dimana kolom dari $r \times n$ *parity-check matrix* merupakan n *non-zero vectors* yang berbeda dari $\mathbb{F}_q^n$ dengan entri *non-zero* pertama sama dengan 1.

e-ISSN No. 2721-6802

Contoh 1. Mendefinisikan Hamming *code* $\mathcal{C}_{Ham} = [7, 4, 3]_2$ menggunakan *generator matrix*

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

Jika mendefinisikan *matrix*

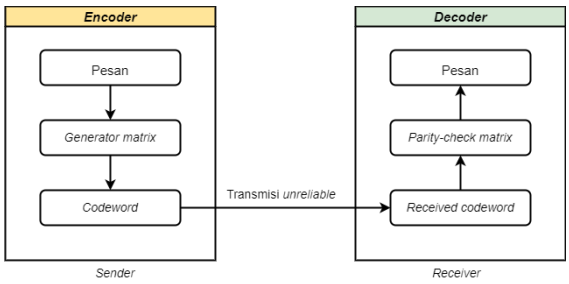
$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

lalu $HG = 0$ dan H merupakan *parity-check matrix* untuk $\mathcal{C}_{Ham}$.
Diberikan tabel *dimension* dari semua kemungkinan Hamming *code* oleh Kashkoulli [7] pada Tabel 1.

TABEL 1. *DIMENSION* DARI SEMUA KEMUNGKINAN HAMMING *CODE*

Parity bits	*Total bits*	*Data bits*	Notasi
3	7	4	$H(7,4)$
4	15	11	$H(15,11)$
5	31	26	$H(31,26)$
...			
r	$n = 2^r - 1$	$k = 2^r - r - 1$	$H(n,k)$ $= (2^r - 1, 2^r - r - 1)$

Dalam mentransmisikan pesan, *sender* sebagai *encoder* melakukan proses *encoding* dengan menggunakan *generator matrix* untuk memperoleh *codeword* yang terdiri dari pesan dan *redundant bits* atau *parity bits*. Kemudian, pesan ini melalui *noisy channel* yang dapat mengakibatkan terjadinya *error* sehingga *codeword* yang diterima oleh *sender* tidak sama dengan *codeword* pada *sender*. Oleh karena itu, *receiver* sebagai *decoder* melakukan proses *decoding* dengan menggunakan *parity-check matrix* untuk memulihkan *codeword* sehingga dapat diperoleh pesan. Menurut Kashkoulli [7], skema proses *encoding* dan *decoding* menggunakan Hamming *code* dalam transmisi pesan dapat dilihat pada Gambar 1 berikut.



GAMBAR 1. KONSEP *CODING* PADA TRANSMISI PESAN DIGITAL

Diberikan definisi *syndrome* oleh Kosek [10] dan Biswas [13].

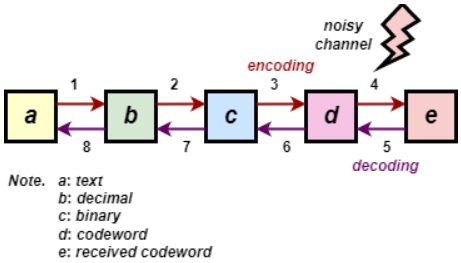
Definisi 8. Misalkan L merupakan (n, k)*−code* dengan *parity-check matrix* H. Untuk setiap $y \in \mathbb{F}_q^n$, *word* $y \cdot H^T$ disebut *syndrome* dari y. *Syndrome* mengidentifikasi adanya *error* dalam *received codeword*. Nilai *syndrome* merupakan posisi dari *code* dimana *error* berada.

Berikut merupakan algoritma *error detection* dan *error correction* pada komunikasi digital menggunakan Hamming *code*.

1. Pihak *sender* melakukan proses *encoding* terlebih dahulu sebelum mentransmisikan pesan berisi informasi dengan cara mengalikan *code* c dengan *generator matrix* $\mathcal{G}$ untuk memperoleh *codeword* d ($d = c \,.\, \mathcal{G}$).
2. Pihak *receiver* melakukan proses *decoding* yang terdiri dari dua hal penting, yaitu *error detection* dan *error correction*. *Error detection* dilakukan dengan menghitung *syndrome* s dengan cara mengalikan pesan yang diterima oleh *receiver* e dengan *parity-check matrix* H^T ($s = e \,.\, H^T$). Jika hasil yang diperoleh $s = 0$, maka tidak terjadi *error* dalam e, artinya e dapat diterima dengan baik sebagai *codeword*. Tetapi, jika $s \neq 0$, maka terjadi *error* dalam e, artinya e tidak dapat diterima dengan baik sebagai *codeword* dan harus ditemukan letak atau posisi dimana *error* itu berada. Posisi *error* dapat ditemukan dengan mencari tahu pada kolom di H^T yang berisikan s. Jika pada posisi kolom di H^T ke-i, maka *error* terjadi pada posisi ke-i.
3. Setelah ditemukan posisi *error* (*error detection* selesai dilakukan), maka dilakukan *error detection* $d$dengan melakukan operasi *Exclusive* OR (XOR) pesan yang diterima e dengan *error* r untuk mengembalikan atau memulihkan atau memperoleh kembali *codeword* d.

B. Contoh Perhitungan Error Detection dan Error Correction pada Komunikasi Digital Menggunakan Hamming Code

Secara sederhana, alur atau skema proses transmisi pesan teks pada komunikasi digital dapat dilihat pada Gambar 1.



GAMBAR 2. PROSES TRANSMISI PESAN TEKS PADA KOMUNIKASI DIGITAL

Pesan asli berupa teks dari pihak *sender* akan ditransmisikan sampai kepada pihak *receiver*. Pesan yang terdapat pada pihak *sender* harus sama isinya dengan pesan yang terdapat pada pihak *receiver*, dalam hal ini pesan berupa informasi teks. Pesan berupa teks (a) akan dikonversikan terlebih dahulu ke dalam bentuk bilangan *decimal* (b). Setelah itu, diperlukan konversi lagi dari bilangan *decimal* ke dalam bentuk bilangan *binary* (c) sehingga diperoleh suatu *code* yang berisi urutan 0 dan 1. Untuk memastikan atau menjamin keutuhan isi pesan (*reliability*), maka pihak *sender* melakukan proses yang disebut dengan *encoding* yang menghasilkan *codeword* (d). *Codeword* merupakan gabungan atau kombinasi antara informasi (*data bits*) dengan *redundancy bits* atau *parity bits*. Pada proses perjalanannya, terdapat kemungkinan bahwa pesan mengalami serangan *error* karena melewati *noisy channel*. Hal itu menyebabkan isi pesan berubah ketika diterima oleh pihak *receiver* (e). Pihak *receiver* harus mengembalikan *codeword* (d) yang tanpa error, yaitu melalui proses *decoding*. Dalam *decoding*, dilakukan *error detection* dan *error correction* sehingga diperoleh *codeword* yang kemudian akan dikonversikan ke dalam bentuk bilangan *decimal* untuk mengetahui pesan berupa teks.

Andaikan pihak *sender* ingin mentransmisikan pesan berisi informasi berupa teks yaitu '$GOLDEN$' ke pihak *receiver* melalui suatu *noisy channel*. Berikut diberikan langkah transmisi pesan teks digital.

1) Menetapkan dimensi Hamming *code* untuk *encoding* dan *decoding* sehingga mengetahui berapa banyaknya *data bits* dan *parity bits* yang akan dipakai.
   Terdapat banyak *dimension* Hamming *code*, yaitu $H(7,4)$, $H(15,11)$, $H(31,26)$, . . ., $H(n,k)$. Dalam hal ini, digunakan $H(7,4)$ Hamming *code*.

2) Menetapkan aturan atau sistem tertentu untuk mengkonversikan pesan berupa teks menjadi *code* (urutan bilangan) yang disesuaikan dengan keberadaan dimensi Hamming *code*. Penetapan aturan konversi teks menjadi bilangan ini dapat berdasarkan *American Standard Code for Information Interchange* (ASCII), urutan huruf dan angka, ataupun aturan lainnya yang ditetapkan.
Dalam hal ini digunakan 16 karakter huruf dari A sampai P $(A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P)$ dan bilangan *decimal* dari 0 sampai 15 $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$. Representasi huruf dalam bilangan *decimal* secara berurutan dari A direpresentasikan oleh 0, B direpresentasikan oleh 1, C direpresentasikan oleh 2, dan seterusnya sampai dengan P direpresentasikan oleh 15. Untuk pesan teks '$GOLDEN$', maka representasi huruf oleh bilangannya menjadi $G = 6$, $O = 14$, $L = 11$, $D = 3$, $E = 4$, dan $N = 13$.

3) Mengkonversikan setiap bilangan *decimal* ke dalam bentuk sistem bilangan *binary*.
Dalam hal ini, sistem bilangan *binary* terdiri dari 4 digit untuk setiap hurufnya yang merupakan *data bits* karena akan digunakan $(7, 4)$ Hamming *code*. Konversi bilangan *decimal* $6, 14, 11, 3, 4,$ dan 13 ke bilangan *binary* yaitu dengan cara membagi bilangan *decimal* dengan 2 kemudian mengambil sisa pembagiannya dan menulis sisa pembagian dari urutan bawah ke atas. Jika digit bilangan *binary*-nya berjumlah 3, maka ditambahkan 0 di depan (sisi kiri) bilangan *binary* agar jumlah digit menjadi 4. Jika bilangan *binary*-nya berjumlah 4, maka tidak perlu ditambahkan 0 di depan (sisi kiri) bilangan *binary*.
Untuk konversi 6, maka $6:2 = 3$ sisa 0, $3:2 = 1$ sisa 1, $1:2 = 0$ sisa 1. Penulisan bilangan *binary* dari bawah ke atas sehingga 011. Di depan (sisi kiri) 011 ditambahkan 0 sehingga konversi 6 yaitu 0110. Melalui cara tersebut, diperoleh konversi bilangan *decimal* ke bilangan *binary* menjadi $6 = 0110, 14 = 1110, 11 = 1011, 3 = 0011, 4 = 0100,$ dan $13 = 1101$.

4) Melakukan *encoding* untuk memperoleh *codeword* (d) dengan mengalikan pesan berupa *binary* (c) dengan *generator matrix* $(\mathcal{G})$. Setelah memperoleh *codeword*, pesan ditransmisikan.
Untuk $(7, 4)$ Hamming *code*, didefinisikan *generator matrix* $\mathcal{G}$ berikut.

$$\mathcal{G} = (I_k|P)$$

$$\mathcal{G} = (I_4|P)$$

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

Code ini memiliki Hamming *distance* $d_{Ham} = 3$ yang mampu melakukan *error detection* maksimum dua *errors* dan *error detection* maksimum satu *error* dengan *information rate* $4/7$. *Code* d memuat 64 *codewords*.
Untuk $c_1 = (0 \quad 1 \quad 1 \quad 0)$, maka

$$d_1 = c_1 \,.\, \mathcal{G}$$

$$= (0 \quad 1 \quad 1 \quad 0) \,.\, \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$d_1 = (0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1).$$

Dengan cara yang sama seperti pada $c_1 = (0 \quad 1 \quad 1 \quad 0)$, maka

1. Untuk $c_2 = (1 \quad 1 \quad 1 \quad 0)$, diperoleh $d_2 = (1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0)$.
2. Untuk $c_3 = (1 \quad 0 \quad 1 \quad 1)$, diperoleh $d_3 = (1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)$.
3. Untuk $c_4 = (0 \quad 0 \quad 1 \quad 1)$, diperoleh $d_4 = (0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0)$.
4. Untuk $c_5 = (0 \quad 1 \quad 0 \quad 0)$, diperoleh $d_5 = (0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0)$.
5. Untuk $c_6 = (1 \quad 1 \quad 0 \quad 1)$, diperoleh $d_6 = (1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0)$.

Dari proses *encoding*, diperoleh *codeword* $d_1 = (0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 1)$, $d_2 = (1\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0)$, $d_3 = (1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1)$, $d_4 = (0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0)$, $d_5 = (0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0)$, dan $d_6 = (1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0)$.

5) Selama proses transmisi melalui *noisy channel*, terdapat kemungkinan *codeword* mengalami serangan *error* yang menyebabkan perubahan digit dari 0 ke 1 atau sebaliknya. Oleh sebab itu, dilakukan *error detection* dengan mencari *syndrome* (s), diperoleh dari mengalikan pesan yang diterima oleh pihak *receiver* (e) dengan *transposed parity-check matrix* (H^T). Jika $s = 0$, berarti tidak terjadi *error* tetapi jika $s = 1$, maka terjadi *error*. Kemudian, menemukan posisi kolom pada H^T yang berisi s. Andaikan pada posisi ke-i, maka *error* terjadi pada posisi ke-i.

Melalui *noisy* channel, diperoleh *received codeword* $e_1 = (0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 1)$, $e_2 = (0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0)$, $e_3 = (1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1)$, $e_4 = (0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 0)$, $e_5 = (0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0)$, dan $e_6 = (1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0)$. Akan dilakukan *error detection* dengan mencari *syndrome* untuk mngetahui apakah terjadi *error* atau tidak. Dari *generator matrix* $\mathcal{G}$, dapat diperoleh *parity-check matrix* H.

$$H = (P^T | I_{n-k})$$

$$H = (P^T | I_3)$$

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

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

Untuk $e_1 = (0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 1)$, maka

$$s_1 = e_1 . H^T$$

$$= (0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 1) . \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$s_1 = (0\ \ 0\ \ 0)$$

Dengan cara yang sama seperti pada $e_1 = (0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 1)$, maka

1. Untuk $e_2 = (0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0)$, diperoleh $s_2 = (1\ \ 0\ \ 1)$.
2. Untuk $e_3 = (1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1)$, diperoleh $s_3 = (1\ \ 1\ \ 0)$.
3. Untuk $e_4 = (0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 0)$, diperoleh $s_4 = (1\ \ 1\ \ 1)$.
4. Untuk $e_5 = (0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0)$, diperoleh $s_5 = (0\ \ 0\ \ 0)$.
5. Untuk $e_6 = (1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0)$, diperoleh $s_6 = (0\ \ 0\ \ 0)$.

Terlihat bahwa pada e_1, e_5, e_6 tidak terjadi *error*. *Error* terjadi pada e_2, e_3, dan e_4. Pada H^T, e_2 terletak pada baris pertama, e_3 terletak pada baris kedua, dan e_4 terletak pada baris ketiga. Hal ini berarti, *error* di e_2 terletak pada posisi *bit* pertama, *error* di e_3 terletak pada posisi *bit* kedua, dan *error* di e_4 terletak pada posisi *bit* ketiga.

Prosiding Pendidikan Matematika, Matematika, dan Statistika Volume 7 (2023)
e-ISSN No. 2721-6802

6) Setelah posisi *error* ditemukan, maka dilakukan *error correction* untuk memperoleh *codeword* atau pesan yang *corrected* (d) dengan menambahkan (operasi XOR) *codeword* yang diterima (e) dengan *error* (r).
Melakukan *error correction* pada e_2, e_3, dan e_4.
$$d = e + r$$
Untuk $e_2 = (0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0)$, maka
$$d_2 = e_2 + r_2$$
$$= (0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0) + (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)$$
$$d_2 = (1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0).$$
Dengan cara yang sama seperti pada $e_2 = (0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0)$, maka
   1. Untuk e_3, diperoleh $d_3 = (0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)$.
   2. Untuk e_4, diperoleh $d_4 = (1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0)$.

Diperoleh *codeword* atau pesan yang *corrected* $d_2 = (1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0)$, $d_3 = (0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)$, dan $d_4 = (1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0)$.

7) Dari d, dapat diperoleh pesan berupa *binary* (c) karena d dihasilkan dari *standard form generator matrix* yang mana 4 digit dari kiri merupakan pesan informasi atau *data bits* dan 3 digit dari kanan merupakan *parity bits*. Setelah memperoleh c, maka dikonversikan ke dalam bentuk *decimal* (b) yang setiap bilangan *decimal* merepresentasikan teks huruf (a) sesuai aturan pada langkah 2).
Dari d, maka dapat diperoleh pesan *binary* $c_1 = (0 \quad 1 \quad 1 \quad 0)$, $c_2 = (1 \quad 1 \quad 1 \quad 0)$, $c_3 = (1 \quad 0 \quad 1 \quad 1)$, $c_4 = (0 \quad 0 \quad 1 \quad 1)$, $c_5 = (0 \quad 1 \quad 0 \quad 0)$, dan $c_6 = (1 \quad 1 \quad 0 \quad 1)$.
Setelah diperoleh c_1, c_2, c_3, c_4, c_5, c_6, maka dikonversikan ke dalam bentuk *decimal* yaitu dengan cara mengalikan masing-masing *bits* dalam bilangan dengan nilai posisinya sehingga diperoleh pesan *decimal* $b_1 = 6$, $b_2 = 14$, $b_3 = 13$, $b_4 = 3$, $b_5 = 4$, dan $b_6 = 13$. Kemudian, dari bentuk *decimal* dikonversikan ke dalam huruf sehingga diperoleh pesan berupa teks $a_1 = G$, $a_2 = O$, $a_3 = L$, $a_4 = D$, $a_5 = E$, $a_6 = N$ yang membentuk kata ‘GOLDEN’.

C. Program Sederhana Error Detection dan Error Correction Menggunakan Hamming Code pada SageMath Versi 9.3

Untuk membentuk program sederhana *error detection* dan *error correction* menggunakan Hamming code, digunakan *software SageMath* versi 9.3. Mendefinisikan *generator matrix* dan *parity-check matrix* terlebih dahulu dengan *length* 7, *dimension* 4, dan *minimum distance* 3 melalui *syntax* berikut.

```
G = matrix(GF(2),[[1,0,0,0,1,0,1],
 [0,1,0,0,1,1,0],
 [0,0,1,0,1,1,1],
 [0,0,0,1,0,1,1]])
H = matrix(GF(2),[[1,1,1,0,1,0,0],
 [0,1,1,1,0,1,0],
 [1,0,1,1,0,0,1]])
C = codes.HammingCode(GF(2),3)
C
```

Diperoleh *output* `[7, 4] Hamming Code over GF(2).` Dilanjutkan mendefinisikan karakter huruf yang akan dipakai, yaitu dalam hal ini menggunakan huruf A sampai dengan huruf P (16 karakter). Huruf A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P disimpan dalam variabel `key_dict1`. Sedangkan representasi masing-masing huruf dengan *bits* (*binary digits*) disimpan dalam variabel `val_dict1`. Konversi dari huruf menjadi *bits* disimpan dalam `dict1` dan konversi dari *bits* menjadi huruf disimpan dalam `dict2`. Dapat dituliskan *syntax* berikut.

```
strings = 'ABCDEFGHIJKLMNOP'
dict1 = {}
for i in range(len(strings)):
    dict1[str(strings[i])] = f'{i:04b}'

key_dict1 = list(dict1.keys())
val_dict1 = list(dict1.values())
```

```
dict2 = {}
for j in range(len(strings)):
    dict2[str(val_dict1[j])] = key_dict1[j]
```

Setelah itu melakukan proses transmisi pesan, *encoding*, dan *decoding* dengan menuliskan *syntax* berikut.

```
def str2bin(string):
    return dict1[string]

def bin2vec(biner):
    return vector(GF(2),[int(k) for k in biner])

def vec2bin(vector):
    return ''.join(str(k) for k in vector)

def bin2str(biner):
    return dict2[biner]

def text2bin(t):
    return ''.join(str2bin(j) for j in t)

def entradec_str(string):
    result = ['','','','','']
    for s in string:
        v = bin2vec(str2bin(s))
        ve = C.encode(v, encoder_name="Systematic")
        ves = vec2bin(ve)
        vt = channel.transmit(ve)
        vts = vec2bin(vt)
        vd = C.decode_to_code(vt, decoder_name="Syndrome")
        vds = vec2bin(vd)
        vm = C.decode_to_message(vt, decoder_name="Syndrome")
        vms = vec2bin(vm)
        m = bin2str(vms)
        result[0] += ves
        result[1] += vts
        result[2] += vds
        result[3] += vms
        result[4] += m
       # result[5] += bin2str(vts)
    return result
```

Dalam hal ini, *sender* ingin mentransmisikan pesan berupa informasi teks "GOLDEN" dan menampilkan hasil dari *error detection* pada proses *decoding* dengan mencoba jika *error rate*-nya berupa $0, 1, 2, 3, 4, 5, 6, 7$ yang *syntax*-nya dapat dituliskan berikut.

```
text = "GOLDEN"
for i in [0,1,2,3,4,5,6,7]:
    channel = channels.StaticErrorRateChannel(C.ambient_space(),i)
    #print(entradec_str(text)[5])
    print(entradec_str(text)[4])
```

Diperoleh hasil GOLDEN, GOLDEN, CACAHF, KDCEPO, ENPANO, LENICA, JBEMLC, JBEMLC. Dari perolehan hasil, terlihat bahwa jika *error rate* sebesar 0 dan 1, maka program dapat melakukan *error correction* sehingga pesan teks dapat diterima dengan baik. Tetapi, jika *error rate* sebesar > 1, maka program tidak mampu untuk melakukan *error correction* sehingga pesan teks tidak dapat diterima dengan baik. Program sederhana *error detection* dan *error correction* pada Hamming *code* menggunakan *SageMath* 9.3 dapat dilihat melalui *repository* pada https://github.com/hafizhahfifi/Hamming.

IV. SIMPULAN DAN SARAN

Pada komunikasi digital, pesan berisi informasi yang ditransmisikan *sender* kepada *receiver* melalui *noisy channel* memungkinkan untuk terjadi *error* sehingga pesan menjadi tidak *reliable*. Maka dari itu, *sender* perlu melakukan proses *encoding* dan *receiver* melakukan proses *decoding* (*error detection* dan *error correction*). *Sender* mentransmisikan pesan teks kepada *receiver*. Pesan teks dikonversi dalam bilangan

decimal kemudian bilangan *binary*. Selanjutnya, *sender* melakukan proses *encoding* dengan mengalikan pesan *binary digits* dengan *generator matrix* $\mathcal{G}$ untuk memperoleh *codeword*. *Codeword* yang melalui *noisy channel* dapat mengakibatkan substansi *received codeword* tidak sama dengan *codeword*. Untuk memulihkannya, *receiver* melakukan proses *decoding*, yaitu *error detection* dilakukan dengan mengalikan *codeword* dengan *transposed parity-check matrix* H^T untuk memperoleh *syndrome* dan *error correction* dilakukan dengan melakukan operasi XOR *received codeword* dengan *error* untuk memperoleh *corrected codeword*. Dari program sederhana menggunakan *SageMath* 9.3, diperoleh hasil bahwa Hamming *code* dapat melakukan *error detection* maksimum dua *errors* dan *error correction* maksimum satu *error* (dilihat dari hasil pesan teks setelah proses *encoding*). Saran untuk penelitian selanjutnya, diharapkan dapat melakukan transmisi pesan digital berupa gambar, audio, ataupun video menggunakan Hamming *code*.

UCAPAN TERIMA KASIH

Terima kasih kepada orang tua, Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret, dosen konsultan Dr. Putranto Hadi Utomo, S.Si., M.Si., teman-teman, dan seluruh pihak yang membantu penelitian ini sehingga dapat berjalan dengan lancar.

DAFTAR PUSTAKA

[1] C. Roering, "Coding Theory-Based Cryptography: McEliece Cryptosystems in Sage", Honours Thesis, Departments of Mathematics and Computer Science, College of Saint Benedict/Saint John's University, 2013.

[2] D. E. Mshelia., P. E. Dibal, and S. Isuwa, "Reducing the Bit Error Rate of a Digital Communication System using an Error-control Coding Technique", International Journal of Scientific Engineering Research, vol. 8, pp. 625-628, April 2017.

[3] H. Singh, "Code based Cryptography: Classic McEliece", Scientific Analysis Group Defence R&D Organisation, 2020.

[4] J. C. Bowman. 2015. "Math 422 Coding Theory & Cryptography", University of Alberta.

[5] J. W. P. Hirschfeld. 2014. "Coding Theory", Mustansiriyah University.

[6] M. Jibril, "Algebraic Codes for Error Correction in Digital Communication Systems", Doctor Degree Thesis, Faculty of Technology, University of Plymouth, England, 2011.

[7] M. Kashkoulli. 2018. "The Implementation and Verification of Hamming Code", Master Degree Thesis, Department of Electronics and Telecommunication, Politecnico di Torino.

[8] N. Aydin, "An Introduction to Coding Theory via Hamming Codes", A Computational Science Module, Department of Mathematics, Kenyon College, 2007.

[9] P. H. Utomo, "Constrained Arrays and Erasure Decoding", Doctor Degree Thesis, Eindhoven University of Technology, 2018.

[10] P. M. Kosek. 2014. "Error Correcting Codes", Master Degree Thesis, The Ohio State University.

[11] P. Udomkavanich. 2006. "2301532: Coding Theory", Chulalongkorn University.

[12] R. W. Hamming, "Error Detecting and Error Correcting Codes", The Bell System Technical Journal, vol. 29, no. 2, pp. 147–160, April 1950.

[13] S. Biswas. "Introduction to Coding Theory: Basic Codes and Shannon's Theorem".

[14] S. Lin and D. J. S. Junior, "Error Control Coding: Fundamentals and Applications. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1983.

[15] Y. Lindell. 2010. Introduction to Coding Theory Lecture Notes*, Bar-Ilan University.

[16] 2010. "MATH 3302 Coding and Cryptography", The University of Queensland, Australia.