PRISMA 8 (2025): 385-391
PRISMA, Prosiding Seminar Nasional Matematika
https://journal.unnes.ac.id/sju/index.php/prisma/
ISSN 2613-9189

Enkoding dan Dekoding Kode Siklik Menggunakan Sirkuit

Zam Zam Yusuf Wibowo[a,*], Putranto Hadi Utomo[a]

[a]*Universitas Sebelas Maret, Jebres, Surakarta, 57126, Indonesia*

* *Alamat Surel: zamzam.yusuf@student.uns.ac.id*

Abstrak

Salah satu topik utama dalam bidang komunikasi adalah kode linear yang merupakan kode paling dasar dalam teori pengkodean dan sangat berguna untuk aplikasi praktis. Dalam pengiriman sebuah informasi diperlukan adanya proses yang dinamakan enkoding dan dekoding dimana pengirim mentransmisikan pesan teks kepada penerima. Dalam pengoperasiannya, ada kemungkinan terdapat *error* sehingga diperlukan sebuah *error-correcting code*. Dari sekian banyak kode *error-correcting code*, salah satu jenisnya adalah kode siklik. Kode siklik banyak digunakan dalam berbagai sistem karena kemudahan dalam implementasinya dan memiliki struktur aljabar yang baik di mata banyak peneliti. Proses enkoding dan dekoding kode siklik ini dapat dikonstruksi ulang ke dalam sebuah bentuk sirkuit yang merupakan salah satu implementasi dasar dalam *engineering*. Tujuan artikel yaitu menjelaskan bagaimana proses enkoding dan dekoding menggunakan sirkuit. Kode Hamming mempunyai beberapa elemen yang identik dengan kode siklik sehingga dapat digunakan untuk memudahkan pembentukan sirkuit. Dari proses konstruksi sirkuit, dapat dibuktikan kelebihan struktur kode siklik dan diketahui perbandingan antara penerapan sebuah sistem melalui sirkuit dan analisis secara matematis melalui aljabar matriks.

Kata kunci:

Kode siklik, Enkoding, Dekoding, Sirkuit

1. Pendahuluan

Dewasa ini internet adalah suatu media komunikasi yang tidak dapat dipisahkan dari masyarakat. Banyak bidang dalam kehidupan yang menjadikan internet sebagai fondasi dengan memanfaatkan teknologi seperti *cloud computing*, *artificial intelligence*, dan lain-lain. Salah satu elemen utama dalam bidang komunikasi adalah *linear code* atau kode linear. Kode linear adalah beberapa kode paling dasar dalam teori pengkodean dan sangat berguna untuk aplikasi praktis (Pless, 1998). Proses pengubahan pesan menjadi *codeword* disebut enkoding (Lin & Xing, 2004). Kode linear digunakan untuk mengenkode pesan yang ada kemudian dikirim melalui saluran komunikasi. Kode biner biasa digunakan dalam hal ini, yaitu digit dalam kode dengan input 0 atau 1. Namun sifat dan teorema kode linear masih berlaku untuk basis bilangan lainnya.

Pada pengiriman pesan dalam bidang komunikasi ada kemungkinan terjadinya gangguan atau *noise*. *Noise* dapat menyebabkan informasi yang diterima berbeda disebabkan perubahan representasi biner pada data saat dilakukannya transmisi data atau dekoding (Pless, 1998). Untuk mencegahnya, diperlukan sebuah *error-correcting code*. Dekoding merupakan proses mengubah barisan simbol yang diterima menjadi *codeword* (Lin & Xing, 2004). Perangkatnya disebut sebagai *dekoder*. Dari sekian banyak *error-correcting code*, salah satu jenisnya adalah kode siklik.

Kode siklik banyak digunakan dalam berbagai sistem karena implementasinya yang mudah. Kode ini juga dikenal memiliki struktur aljabar yang baik di mata banyak peneliti di bidang *coding theory*. Kode siklik pertama kali dipelajari oleh Eugene Prange, seorang peneliti dari Amerika Serikat pada tahun 1957 (Prange, 1957). Sejak itu studi tentang kode siklik yang dilakukan oleh peneliti teori koding dan aljabar terus berkembang. Banyak kode yang digunakan saat ini merupakan kode siklik, beberapa diantaranya

To cite this article:
Wibowo, Z. Y. & Utomo, P. H. (2025). Enkoding dan Dekoding Kode Siklik Menggunakan Sirkuit. *PRISMA, Prosiding Seminar Nasional Matematika 8*, 385-391

adalah kode BCH, kode Reed Solomon, dan kode Golay. Maka tidak salah apabila dikatakan kode siklik adalah kode yang penting dengan berbagai alasan.

Proses enkoding dan dekoding kode siklik ini dapat dikonstruksi ulang ke dalam sebuah bentuk sirkuit yang merupakan salah satu implementasi dasar dalam *engineering*. *Engineering* atau perekayasaan adalah aplikasi ilmu pengetahuan yang merancang, mengembangkan, membangun, atau memelihara suatu sistem, program, dan struktur (Mano & Ciletti, 2013). Tujuannya adalah menyelesaikan masalah dan membuat hidup lebih baik. Sirkuit ini dapat dimengerti sebagai jembatan antara analisis matematis dan *engineering*. Kemudian kode Hamming mempunyai beberapa elemen yang identik dengan kode siklik seperti panjang, dimensi, serta basis sehingga dapat digunakan sebagai pijakan untuk memudahkan pembentukan sirkuit. Berdasarkan latar belakang tersebut, maka artikel ini bertujuan memaparkan sekilas tentang proses enkoding dan dekoding pada kode siklik menggunakan sirkuit yang kemudian akan dibandingkan bagaimana kepraktisan analisis secara matematis dan proses melalui sirkuit yang merupakan salah satu penerapan di dalam bidang *engineering*.

2. Metode

Penelitian artikel ini menggunakan metode tinjauan pustaka dimana peneliti mengandalkan berbagai literatur dari buku dan jurnal-jurnal yang berhubungan untuk mengetahui lebih lanjut mengenai kode siklik. Setelah itu akan direkonstruksi ulang sirkuit untuk proses enkoding dan dekoding kode tersebut dan dibandingkan menggunakan metode aljabar matriks. Rekonstruksi ulang bertujuan untuk memudahkan penelitian ke depannya dan menambahkan apabila terdapat kekurangan dari penelitian-penelitian sebelumnya.

3. Hasil dan Pembahasan

3.1. Dasar Proses Enkoding dan Dekoding Kode Siklik

Dalam proses enkoding, pesan dikalikan dengan matriks generator untuk memperoleh *codeword* yang terdiri dari pesan dan *parity bits*. Setelah itu pesan melewati *noisy channel* dimana ini dapat mengakibatkan adanya *error* sehingga *codeword* yang diterima tidak sama dengan *codeword* awal. Maka perlu dilakukan proses dekoding dimana dekoder bertugas mendeteksi *error* dan mengoreksinya. Deteksi *error* dilakukan dengan mengalikan *codeword* dan matriks cek paritas transpose untuk memperoleh *syndrome*. Setelah *error* telah dikoreksi, dekoder memulihkan *codeword* sehingga diperoleh pesan (Kashkoulli, 2018).

Kode Hamming adalah kode linear untuk *error correction* yang banyak digunakan sebagai *error control* pada komunikasi digital dan sistem penyimpanan data. Kode ini memungkinkan deteksi dan perbaikan *error* dalam data yang dikirim melalui saluran yang mungkin mengalami gangguan selama proses transmisi. Kode Hamming dapat melakukan deteksi error maksimum dua kesalahan dan koreksi error maksimum satu kesalahan (Hafizhah & Utomo, 2023). Variabel n adalah panjang sebuah kode dimana pada kode Hamming, nilai $n = 2^p - 1$ dengan p adalah digit paritas dan $k = n - p$ adalah digit informasi. Setiap n kolom dari matriks cek paritas memuat pasangan *p-tuple* tidak nol berbeda, yang kemudian disebut sebagai lokasi angka dari digit tersebut.

3.2. Konstruksi Sirkuit

Diberikan contoh matriks cek paritas H untuk sebuah kode pengoreksi satu kesalahan dengan panjang 7 sebagai berikut.

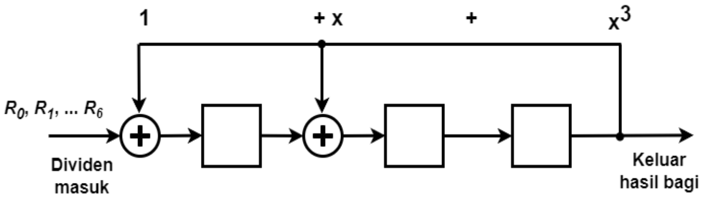
$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Kemudian dibuat sebuah sirkuit berdasarkan suatu polinomial biner. Dalam algoritma pembagian, $c(x) = m(x)g(x) + q(x)$, dimana derajat $r(x)$ kurang dari derajat $m(x)$ (Berlekamp, 2015). Dimisalkan $c(x)$ pada kasus ini adalah $x^6 + x^5 + x^3 + x^2 + 1$ dan $g(x)$ adalah $x^3 + x + 1$. Proses pembagian dilakukan dengan operasi *Exclusive-OR* (*XOR*) sebagai pengganti pengurangan pada pembagian polinomial biasa.

Hasil bagi antara $c(x)$ dan $m(x)$ diwakili oleh digit pertama dari g, dimana $g = (00001111)$, yang mana merupakan koefisien dari hasil bagi sehingga $g(x) = x^3 + x^2 + x + 1$. Sementara sisa dari pembagian adalah $q = (100)$ atau $q(x) = x^2$.

```
        00001111
1011 | 00001101101
       0000
        0001
        0000
         0011
         0000
          0110
          0000
           1101
           1011
            1101
            1011
             1100
             1011
              1111
              1011
               100
```

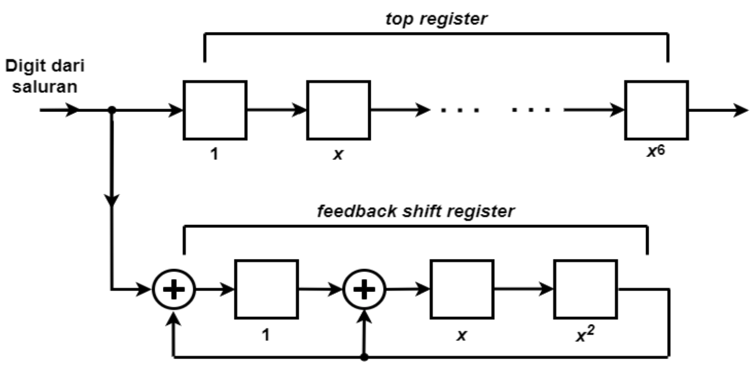
Pembagian sebelumnya dapat diimplementasikan ke dalam sebuah sirkuit seperti pada Gambar 1. Ketiga *flip-flop* disini menyimpan hasil bagi secara berurutan.



Gambar 1. Sirkuit pembagian $c(x) = x^6 + x^5 + x^3 + x^2 + 1$ dan $g(x) = x^3 + x + 1$

Dari contoh sebelumnya, status *flip-flop* secara berurutan adalah $000, 001, 011\ 110, 110, 110, 111, 100$. Ketika pembagian selesai, koefisien dari sisa pembagian muncul pada *flip-flop*.

Jika digit $R_6, R_5, \dots, R_0$ ditransmisi dengan urutan indeks yang paling besar, maka akan terlihat seperti pada Gambar 2. Dekoder melakukan pembagian saat digit diterima dari saluran.



Gambar 2. Sirkuit untuk membagi *word* diterima dengan polinomial tetap $g(x) = x^3 + x + 1$

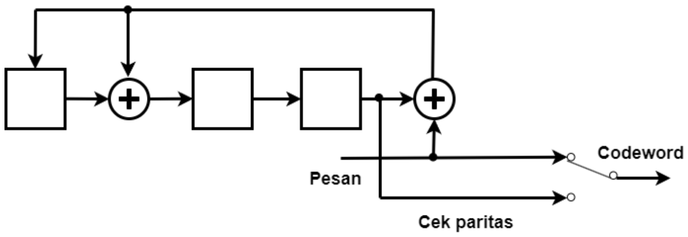
Setelah seluruh n digit telah diterima, polinomial $c(x)$ muncul pada *top register*, dan sisa $q(x)$ muncul pada *feedback shift register.*

3.3. Proses Enkoding dan Dekoding dengan Sirkuit

Selanjutnya dibuat sebuah sirkuit dalam proses enkoding kode. Diberikan suatu kode siklik $(7, 4)$ dengan $g(x) = 1 + x + x^3$. Sirkuit enkoding berdasarkan $g(x)$ diperlihatkan pada Gambar 3. Misal pesan $u = (1011)$ dienkode. Saat digit pesan digeser menuju *register*, maka informasi dalam *register* dituliskan pada Tabel 1. Didapatkan untuk *shift register* yaitu (100) setelah empat pergeseran dan dituliskan sebelum pesan u untuk mendapatkan *codeword*. Maka *codeword* lengkap yang dienkode adalah (1001011) atau $1 + x^3 + x^5 + x^6$.

Tabel 1. Pergeseran register pada $g(x) = 1 + x + x^3$

Input	Register
	000 (kondisi awal)
1	110 (pergeseran pertama)
1	101 (pergeseran kedua)
0	100 (pergeseran ketiga)
1	100 (pergeseran keempat)



Gambar 3. Enkoder kode Hamming (7,4) dengan $g(x) = 1 + x + x^3$

Dalam tahap awal, *shift register* bernilai (000). Sumber pesan mengirimkan digit $C_{n-1}, C_{n-2}, \dots, C_m$ menuju *feedback shift register*. Kemudian konten dari *shift register* dikirim ke saluran dan input menuju *feedback shift register* diatur ke nol untuk memulai transmisi blok selanjutnya.

Sementara itu untuk proses dekoding terdiri dari tiga langkah yaitu: menghitung sindrom, asosiasi sindrom dengan pola error, dan koreksi error.

Dimisalkan sebuah kode $g(x)$ berkonfigurasi $(7, 4)$ dibentuk dengan $g(x) = 1 + x + x^3$. Kode ini mempunyai *minimum distance* 3 dan mampu mengoreksi satu kesalahan dalam sebuah blok berisi tujuh digit. Karena memuat tujuh digit dalam satu blok, maka terdapat tujuh pola satu kesalahan. Misalkan dalam sebuah proses dekoding, polinomial yang diterima adalah $r(x) = r_0 + r_1x + r_2x^2 + r_3x^3 + r_4x^4 + r_5x^5 + r_6x^6$ bergeser dalam *syndrome register* dari sebelah kiri. *Error pattern* dan sistem terkorespondensinya dituliskan pada Tabel 2.

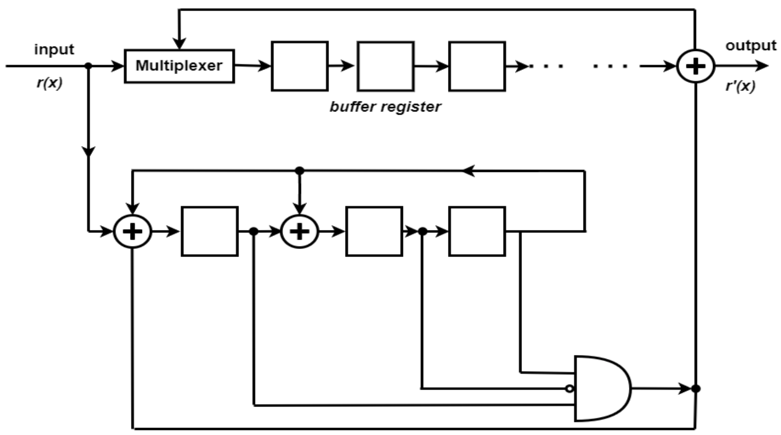
Tabel 2. *Error pattern* $e(x)$, sindrom $s(x)$, dan vektor sindrom dari $r(x)$ pada $g(x) = 1 + x + x^3$

Error pattern $\boldsymbol{e(x)}$	**Sindrom** $\boldsymbol{s(x)}$	**Vektor sindrom** $(\boldsymbol{s_0, s_1, s_2})$
$e_6(x) = \alpha^6$	$s(x) = 1 + x^2$	(101)
$e_5(x) = \alpha^5$	$s(x) = 1 + x + x^2$	(111)
$e_4(x) = \alpha^4$	$s(x) = x + x^2$	(011)
$e_3(x) = \alpha^3$	$s(x) = 1 + x$	(110)
$e_2(x) = \alpha^2$	$s(x) = x^2$	(001)
$e_1(x) = \alpha^1$	$s(x) = x$	(010)
$e_0(x) = \alpha^0$	$s(x) = 1$	(100)

Dapat diamati bahwa $e_6(x) = \alpha^6$ adalah satu-satunya *error pattern* dengan *error* terletak di α^6. Ketika pola ini terjadi, sindrom pada *syndrom register* menjadi (101) setelah seluruh polinomial diterima $r(x)$ memasuki *syndrom register*. Deteksi sindrom ini mengindikasikan bahwa r^6 adalah digit yang *error* dan perlu dikoreksi. Misalkan *error* terjadi pada lokasi x^i $[i.e., e_i(x) = x^i]$ untuk $0 \leq i < 6$. Setelah seluruh polinomial diterima telah bergeser dalam *syndrome register*, *syndrome register* belum menjadi (101); namun setelah $6 - i$ lain bergeser, *syndrom register* menjadi (101), dan digit diterima selanjutnya yang keluar dari *buffer* selanjutnya adalah digit *error*. Maka dari itu, hanya sindrom (101) yang perlu dideteksi, yang hal ini dapat diperoleh melalui sebuah gerbang AND dengan tiga input.

Misalkan terdapat *codeword* $v = (1001011)$ atau $v(x) = 1 + x^3 + x^5 + x^6$ ditransmisikan dan diterima $r = (1011011)$ atau $r(x) = 1 + x^2 + x^3 + x^5 + x^6$. *Error* terjadi pada lokasi x^2. Ketika seluruh polinomial diterima telah bergeser menuju sindrom dan *buffer register*, *syndrome register* memuat (001) sesuai Tabel 2.

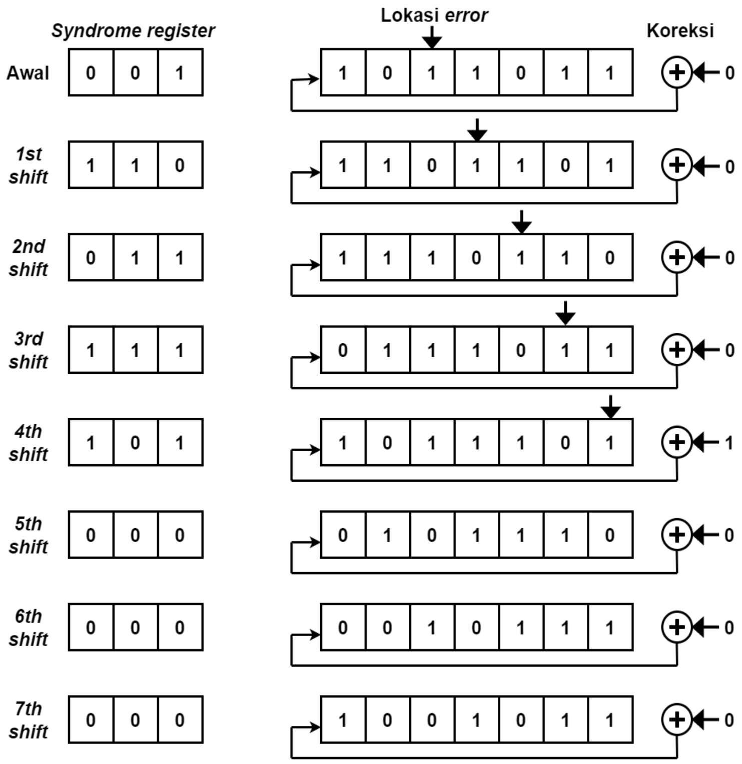
Dari contoh tersebut, dibuat sebuah sirkuit dekoder seperti pada Gambar 4. Polinomial diterima $r(x) = r_0 + r_1 x + \cdots + r_{n-1}x^{n-1}$ dideskripsikan dari digit dengan derajat tertinggi r^{n-1} ke derajat terendah r_0. Setelah dekoding digit r_i, *buffer* dan *syndrom register* bergeser sekali ke kanan. Digit diterima selanjutnya yang didekoding adalah r_{i-1}, r_{i-2}, dst.



Gambar 4. Sirkuit dekoder untuk kode $(7, 4)$ dengan $g(x) = 1 + x + x^3$

Untuk mendekode sebuah kode siklik, polinomial diterima $r(x)$ perlu bergeser menuju *syndrome register* dari akhir sebelah kanan untuk menghitung sindrom. Ketika $r(x)$ telah bergeser menuju *syndrome register*, maka pada register memuat $s^{n-k}(x)$, yaitu sindrom dari $r^{n-k}(x)$, perputaran siklik ke-$(n - k)$ dari $r(x)$. Jika $s^{n-k}(x)$ berkorespondensi dengan *error pattern* $e(x)$ dan $e_{n-1} = 1$, digit derajat tertinggi yaitu $r_n - 1$ dari $r(x)$ adalah salah dan perlu dikoreksi. Saat r_{n-1} terkoreksi, *error effect* perlu dihilangkan dari $s^{n-k}(x)$. Sindrom baru, dilambangkan $s_1^{n-k}(x)$, adalah jumlah dari $s^{n-k}(x)$ dan sisa $\rho(x)$ hasil dari membagi x^{n-k-1} dengan polinomial generator $g(x)$. Karena derajat x^{n-k-1} kurang dari $g(x)$, maka $\rho(x) = x^{n-k-1}$. Sehingga $s_1^{n-k}(x) = s^{n-k}(x) + x^{n-k-1}$, yang berarti efek dari *error* di x^{n-1} pada sindrom dapat dihilangkan dengan memasukkan *error* digit ke *syndrome register* dari akhir sebelah kanan melewati sebuah gerbang *Exclusive-OR* (*XOR*).

Pada Gambar 5, digit pada *syndrome register* dan *buffer register* dituliskan setelah setiap pergeseran. Terdapat sebuah *pointer* untuk memperlihatkan lokasi *error* setelah pergeseran. Setelah empat pergeseran, *syndrome register* menjadi (101), dan digit *error* r_2 adalah digit yang keluar dari *buffer register*.



Gambar 5. Proses koreksi *error* pada sirkuit dekoder untuk $g(x) = 1 + x + x^3$

3.4. Proses Dekoding dengan Matriks

Sirkuit yang didapatkan untuk proses enkoding dan dekoding kode sebelumnya merupakan salah satu pendekatan secara *engineering* yang mana melalui berbagai tahap untuk mendapatkan hasil yang diinginkan. Proses ini dapat dipahami secara matematis untuk mendapatkan perhitungan yang lebih cepat. Telah disebutkan pada subbab sebelumnya bahwa untuk mendeteksi *error*, sebelumnya *codeword* dikali dengan matriks cek paritas transpose untuk memperoleh sindrom. Sindrom yang telah didapat lalu digunakan untuk mencari lokasi *error* pada *codeword*. Setelah *error* terdeteksi, maka *error* dikoreksi yang kemudian dekoder memulihkan *codeword* sehingga dapat diperoleh pesan.

Dengan $v = (1001011)$, $r = (1011011)$, dan $H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$, maka

$$s = rH^T = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

Berdasarkan perhitungan diperoleh $s = (001)$. Bila diperhatikan dalam Tabel 2, pada vektor sindrom (101), pola *error* yang didapatkan adalah $e_2(x) = \alpha^2$. Ini berarti *error* terdapat pada r_2 dan perlu dikoreksi. Maka pesan semestinya adalah $r(x) = 1 + x^3 + x^5 + x^6$, yang sesuai dengan pesan awal (v).

4. Simpulan

Kode siklik adalah salah satu kelas kode yang banyak digunakan dalam berbagai sistem karena kemudahan pada implementasinya dan memiliki struktur aljabar yang baik di mata banyak peneliti. Salah satu bentuk kode siklik adalah kode dengan parameter $n = 7,\ k = 4$ atau $(7, 4)$. Dalam pengiriman sebuah informasi diperlukan adanya proses yang dinamakan enkoding dan dekoding dimana pengirim mentransmisikan pesan teks kepada penerima. Proses enkoding dan dekoding pada kode siklik dapat dikonstruksi ulang ke dalam sebuah bentuk sirkuit yang merupakan salah satu implementasi dasar dalam *engineering*. Kode Hamming mempunyai beberapa elemen yang identik dengan kode siklik sehingga dapat digunakan untuk memudahkan pembentukan sirkuit. Dari proses konstruksi sirkuit, diketahui bahwa analisis secara matematis mampu mempersingkat proses perhitungan. Diketahui juga bahwa kode siklik dapat direpresentasikan dengan polinomial generator sehingga penulisannya menjadi lebih singkat.

Daftar Pustaka

Berlekamp, E. R. (2015). *Algebraic Coding Theory*. Singapore: World Scientific Publishing Company.

Fraleigh, J. B. (1994). *First Course in Abstract Algebra*. New York: Addison-Wesley Publisihing Company.

Gallian, J. (2012). *Contemporary Abstract Algebra*. Minnesota: University of Minnesota Duluth.

Hafizhah & Utomo, P. H. (2023). Error Detection dan Error Correction pada Komunikasi Digital Menggunakan Hamming Code. *Prosiding Seminar Pendidikan Matematika dan Matematika 7*.

Kashkoulli, M. (2018). The Implementation and Verification of Hamming Code. (Master's Thesis). Politecnico di Torino. Torino.

Lin, S. & Xing, C. (2004). *Coding Theory: A First Course*. New York: Cambridge University Press.

Mano, M. M. & Ciletti, M. (2013). *Digital Design : With an Introduction to the Verilog HDL*. New Jersey: Pearson Education, Inc.

Pless, V. (1998). *Introduction to the Theory of Error-Correcting Codes*. New York: John Wiley and Sons Inc.

Prange, E. (1957). Cyclic Error-Correcting Codes in Two Symbols. *AFCRC- TN-57*, 103.