KONSTRUKSI *LEXICOGRAPHIC* UNTUK MEMBANGUN KODE HAMMING (7, 4, 3)

Aurora Nur Aini, Bambang Irawanto
Jurusan Matematika FMIPA UNDIP
Jl. Prof. Soedarto, S. H, Semarang 50275

Abstract. Hamming code can correct single error in massages transmission. Hamming codes can be constructed by Lexicographic codes. Lexicographic construction is a greedy algorithm that produces error correcting codes known as lexicographic codes. There are two ways to construct lexicographic codes. They are greedy construction and lexicographic constructions. Given codes with minimum distance d and length n. To construct the greedy algorithm, the codeword with length n are processed in some fixed order, and the next codeword is inserted in the code when its distance from all codewords previously selected is $\geq d$. The Lexicographic Construction is a different approach with a goal to speed up the process of generating lexicodes by storing the reusable information in the memory.

***Keyword** :* linear codes, lexicographic codes, coset, generator matrix

1. PENDAHULUAN

Kode Hamming merupakan salah satu kode yang sering digunakan dalam pengkodean pesan digital. Kode Hamming tersebut dapat dibentuk dengan menggunakan kode *lexicographic.* Algoritma Greedy tersebut terbagi menjadi dua, yaitu konstruksi Greedy dan konstruksi *lexicographic*. Konstruksi *lexicographic* dapat menghasilkan kode lebih cepat dari pada konstruksi Greedy karena pada konstruksi ini menggunakan kembali informasi-informasi pada iterasi sebelumnya dengan menyimpannya dalam memori. Pada konstruksi *lexicographic* terdapat beberapa parameter yang digunakan yaitu bobot koset, koset *leader*, dan sindrom. Kode *lexicographic* yang berkerja di lapangan berhingga GF_2 atau kode *lexicographic* biner, merupakan kode linier. Pada kode linier, kombinasi linier dari kodekata-kodekata yang ada juga merupakan kodekata. Untuk membentuk kode Hamming (7, 4, 3) dengan menggunakan konstruksi *lexicographic* dibutukan 4 iterasi.

2. PEMBAHASAN

Definisi 1

Diberikan kode blok linier $\mathbb{C}$ dan vektor $a \in V_n(F)$, koset dari C adalah

$$a + C = \{a + x \mid x \in \mathbb{C}\}$$

Vektor a dan b dikatakan berada pada koset yang sama jika $a - b \in$ C.

Definisi 2

Sindrom s atas vektor $y \in V_n(Z_2)$, yang bersesuaian dengan kode (n,k,d) adalah vektor pada subruang $V_{n-k}(Z_2)$ yang didefinisikan sebagai berikut :

$s = \mathbf{H} \cdot y^T$

Teorema 1

Pada masing-masing koset terdapat vektor tunggal m, yang disebut *lexicographically earliest vector* atau vektor paling awal *lexicographic*, yang terdiri dari bit informasi yang semuanya nol dan bit sisa (*redundant bit*) yang membentuk sindrom untuk koset tersebut. Jika diasumsikan matriks generator berada pada bentuk standar, *lexicographic earliest vector* merupakan bentuk :

$m = (00\ldots00|s_1 \ldots s_{n-k})$

Bukti:

Tanpa mengurangi sifat umum, diasumsikan kode blok berada pada bentuk standar, yang berarti, k bit pertama merupakan bit informasi. Hal ini untuk menunjukkan bahwa vektor m ada pada masing-masing koset. Asumsikan vektor y, dengan beberapa bit informasi tak nol, kodekata dari matriks generator selalu dapat ditambahkan ke *y*, sehingga akan membuat semua bit informasi dari *y* menjadi nol, oleh karena itu diperoleh *m*. Sesuai dengan definisi koset, *y* dan *m* berada pada koset yang sama, jadi dapat disimpulkan bahwa vektor *m* ada untuk masing-masing koset. Sindrom untuk koset ini adalah :

$s = H.\, y^T = H\,.\, m^T = [A^T \mid I_{n-k}]\,.\,[0\ldots0\ \ p_1\ldots p_{n-k}]^T$

Dimana $p_1, p_2, \ldots, p_{n-k}$ adalah bit sisa atas vektor *m*. Karena *k* bit pertama dari vektor *m* semua nol, *k* bit pertama matriks cek paritas tidak mempengaruhi perhitungan, jadi diperoleh

$s = [A^T \mid I_{n-k}]\,.\,[0\ldots0\ \ p_1\ldots p_{n-k}]^T = [I_{n-k}]\,.\,[p_1\ldots p_{n-k}]^T = [p_1\ldots p_{n-k}]^T$

Atau sindrom adalah semua bit sisa dari vektor *m*. Ketunggalan sindrom mempengaruhi ketunggalan vektor *m*.

Konstruksi *lexicographic* terdiri atas iterasi-iterasi yang mengulang penghitungan parameter dari koset yang terpartisi oleh lexicode L_{k+1}^d, dengan menggunakan parameter koset *lexicode* L_k^d.

Notasi L_k^d menyatakan kode ke-*k* yang memiliki jarak minimum *d* yang dihasilkan oleh konstruksi *lexicographic*.

Jika $\Lambda(k+1,d)$ menyatakan matriks generator untuk *lexicodes* L_{k+1}^d, dan $\Lambda(k,d)$ menyatakan matriks generator untuk *lexicodes* L_k^d, maka pencarian atas matriks generator untuk kode ke-(*k*+1) pada konstruksi *lexicographic* adalah :

$$\Lambda(k+1,d) = \Lambda(k,d) \cup \langle 1^{d-\rho} | f_{lexi} \rangle$$

f_{lexi} adalah vektor yang digunakan sebagai generator dalam konstruksi *lexicographic* dan ρ adalah bobot maksimum dari koset *leader* kode L_k^d. Untuk dapat menemukan f_{lexi} dan ρ dilakukan dengan cara mencari di antara semua koset *leader* kode L_k^d hingga menemukan bobot maksimum koset *leader*, dan kemudian mencari di antaranya koset yang berada pada urutan paling awal dalam barisan *lexicographic* untuk menjadi f_{lexi}.

Definisi 3

Diberikan kode linier $\mathbb{C}$ dan vektor $x \in V_n(Z_2)$. Jarak d_x dari kode $\mathbb{C}$ ke vektor *x* adalah jarak Hamming minimum antara *x* dan vektor dari $\mathbb{C}$,

$d_x = \min(d(x,c))\forall c \in \mathbb{C}$.

Jari-jari persekitaran (*covering radius*) ρ adalah jarak d_x dari vektor $V_n(Z_2)$ yang paling jauh dari kode $\mathbb{C}$.

$$\rho = \max_{\forall x \in V_n(Z_2)} \min_{\forall c \in \mathbb{C}} d(x,c)$$

Dengan kata lain, jari-jari persekitaran ρ merupakan bobot dari koset leader dengan bobot terbesar. Jari – jari persekitaran tersebut terbatas ke atas dan terbatas ke bawah oleh pertidak samaan berikut:

$$\left\lfloor \frac{d}{2} \right\rfloor \leq \rho \leq (d-1)$$

Teorema 2

Atas lapangan berhingga $V_n(Z_2)$, kode C non trivial (*n*, *k*, *d*) merupakan *lexicodes* jika dan hanya jika kode tersebut dibentuk oleh konstruksi *lexicographic* C = L_k^d.

Bukti :

Mula-mula akan dibuktikan bahwa L_k^d selalu merupakan *lexicodes*, dengan induksi untuk *k* sebarang dan *d* tetap. Untuk *k* = 1, maka L_1^d = $\{0^d, 1^d)$ yang merupakan lexicode (*d*, 1, *d*). Dari induksi tersebut, dapat diasumsikan bahwa L_k^d merupakan lexicode (*n*, *k*, *d*) untuk kode C_k. Dari definisi konstruksi *lexicographic*, L_{k+1}^d memiliki parameter ($n+\rho$, *k+1*, *d*), dimana ρ adalah jari-jari persekitaran atas L_k^d. Menurut lexicode ($n+\rho$, *k'*, *d*), kode $C_{k'}$ dibentuk dengan mengulangi memilih vektor paling awal *lexicographic* (*lexicographic earliest vector*) di ruang $V_{n+\rho}$ (Z_2). C_k dibangun untuk proses pembangunan ini, jadi $C_k \subseteq C_{k'}$ dan dengan hipotesis induksi, $L_k^d \subseteq C_{k'}$. Jadi, vektor $v = \langle 1^{d-\rho} | f_{lexi}(\mathbb{C}_k) \rangle$ merupakan vektor paling awal *lexicographic* dengan jarak ρ dari L_k^d . Oleh karena itu haruslah $v \in C_{k'}$, dan karena $C_{k'}$ linier, $L_{k+1}^d \subseteq C_{k'}$

000…0	L_k^d
111…1	w

Tabel : struktur untuk lexicode ke- (k+1) atau dapat ditulis L_{k+1}^d .
Terdapat $d-\rho$ bit satu pada ruas kiri. Karena jari-jari persekitaran atas L_k^d adalah ρ, maka n bit (misal vektor w) harus berada pada jarak $\leq \rho$ dari bit paling kanan n atas L_{k+1}^d.
Pada penjumlahan, vektor $v \in V_{n+\rho}$ (GF_q) yang berada pada $C_{k'}$ tetapi tidak berada pada L_{k+1}^d memiliki n bit paling kanan yang berada pada jarak $\leq \rho$ dari L_{k+1}^d, dan bit-bit yang lain berada pada jarak $\leq \left\lceil \frac{(d-\rho)}{2} \right\rceil$ dari L_{k+1}^d. Seperti terlihat dari tabel di atas. Asumsi bahwa variabel yang ada non-trivial, menyebabkan $\rho < d$, dapat digunakan pertidaksamaan segitiga untuk melihat bahwa jarak dari v ke L_{k+1}^d (dan berpengaruh ke $C_{k'}$) harus berada pada

$$= \rho + \left\lceil \frac{(d-\rho)}{2} \right\rceil = \left\lceil \frac{(d+\rho)}{2} \right\rceil < d$$

Hal ini kontradiksi dengan definisi semula untuk konstruksi $C_{k'}$. Jadi seharusnya

$L_{k+1}^d = C_{k'}$.

■

Definisi 4
Sekawan (*companion*) atas koset *leader* l, atas vektor $x \in V_n(Z_2)$ adalah koset *leader* dari koset yang berisi vektor $l + x$ sebagai anggotanya. Dituliskan $k_x(l)$ untuk menotasikan sekawan dari l dalam hubungannya dengan x.

Pada iterasi ke-k konstruksi *lexicographic*, digunakan parameter koset dari $\Lambda(k-1,d)$ sebagai input, dan outputnya adalah parameter koset untuk $\Lambda(k,d)$. Teorema berikut menyangkut tentang perubahan koset *leader* di antara dua iterasi :

Teorema 3
Diberikan kode linier C (n, k, d) dengan S merupakan himpunan koset *leader* atas C, dan kode linier C' yang direntang oleh C $\cup \langle 1^{d-\rho} | \lambda \rangle$ untuk $\lambda = f_{lexi} \in V_n(Z_2)$ dan $d-\rho \in$ Z. Maka S' yang merupakan himpunan koset *leader* untuk C' dapat diperoleh dari S menurut korespondensi bijektif berikut:

$$\emptyset : a, l \in S \longrightarrow l' \in S'$$

Dimana $a \in V_{d-\rho-1}(Z_2)$ dan l' didefinisikan sebagai berikut :

$$l' = \begin{cases} \langle 0|a|l \rangle \; jika \; wt(\langle 0|a|l \rangle) \leq wt(\langle 1|\bar{a}|k_\lambda(l) \rangle) \\ \langle 1|\bar{a}|k_\lambda(l) \rangle \; jika \; wt(\langle 0|a|l \rangle) > wt(\langle 1|\bar{a}|k_\lambda(l) \rangle) \end{cases}$$

Dengan $k_\lambda(l)$ menyatakan sekawan dari koset *leader* l atas λ.

Dari teorema tersebut, dapat diketahui bahwa ada dua pilihan untuk menjadi koset *leader*, dan kita harus memilih di antaranya yang memiliki bobot paling kecil.

Pada teorema ini mula-mula harus ditemukan $\lambda = f_{lexi}$, koset di $\Lambda(k+1,d)$ merupakan penggabungan dua buah koset dari $\Lambda(k,d)$. Koset *leader* yang baru merupakan salah satu dari koset *leader* pada koset yang telah digabungkan.

Dari teorema tersebut dapat dilihat bahwa informasi atas bobot koset *leader* sangat penting dalam perhitungan untuk membentuk *array* standar iterasi selanjutnya sehingga *corollary* berikut perlu diperhatikan.

Corollary 1
Jika $wt(l)$ menyatakan bobot koset terpartisi oleh $\Lambda(k-1,d)$ dan $wt(a)$ adalah bobot vektor $a \in V_{d-\rho-1}(Z_2)$, maka bobot $wt(l')$ dari koset yang termasuk $\Lambda(k,d)$ adalah :

$$wt(l') = \min\left(wt(a) + wt(l); d - \rho - wt(a) + wt(k_\lambda(l))\right)$$

Kostruksi *lexicographic* menghendaki untuk menemukan anggota paling awal *lexicographic* (*lexicographic earliest member*) di dalam koset. Teorema berikut menyangkut perubahan anggota paling awal *lexicographic* di antara dua iterasi.

Teorema 4
Diberikan vektor paling awal *lexicographic*, m, untuk semua koset yang termasuk ke dalam $\Lambda(k-1,d)$, vektor paling awal lexicographic m' untuk semua

koset yang termasuk dalam $\Lambda(k,d)$ dibentuk menggunakan pemetaan sebagai berikut :
$m' = \langle 0|a|m\rangle$
Dimana $a \in V_{d-\rho-1}(Z_2)$.

Pada konstruksi *lexicographic* dibutuhkan anggota awal *lexicographic m* atas koset. Berdasarkan teorema 3.2.3 terdapat korespondensi satu-satu antara koset dengan vektor awal *lexicographic*, karena dimensi dari koset harus lebih kecil daripada dimensi vektor *m*, sehingga :

Corollary 2
Diberikan sindrom S_l yang termasuk dalam koset $\Lambda(k-1,d)$, sindrom $S_{l'}$ dari koset yang termasuk kode $\Lambda(k,d)$ adalah :
$S' = \langle a|S_l\rangle$

Teorema 5
Diberikan S_λ dan koset dengan parameter S_l, *l*, dan $wt(l)$, maka :
$S_{k_\lambda(l)} = S_\lambda + S_l$
Bukti :
$S_{k_\lambda(l)} = H\cdot k_\lambda(l) = H\cdot(\lambda+l) = H\cdot\lambda + H\cdot l = S_\lambda + S_l$

Membentuk kode Hamming dengan menggunakan konstruksi *Lexicographic*
Untuk membentuk kode Hamming digunakan algoritma sebagai berikut :
1. Iterasi diawali dengan membentuk $\Lambda(1,d)$ yang merupakan kode pengulangan dengan *n* = *d* dan selanjutnya mencari parameter koset dari $\Lambda(1,d)$ tersebut.
2. Menemukan ρ dan S_λ dan mencetak vektor basis selanjutnya.
3. Menggunakan *corollary* 1 dan 2 dan teorema 5 untuk meng-update *array* standar
4. Mengulangi iterasi dari point 2.

Langkah–langkah untuk membangun matriks generator kode *lexicographic* untuk kode Hamming (7, 4, 3) adalah sebagai berikut :

1. Iterasi dimulai dengan L_1^3

L_k^d = (*n*, *k*, *d*) , untuk k = 1, maka L_1^d = (*d*, *1*, *d*) sehingga L_1^3 = (3, 1, 3), dengan matriks generator *lexicographic*-nya $\Lambda(1,3)$.
$\Lambda(1,3)$ = [111] yang merupakan kode pengulangan dengan *n* = 3.
Dari anggota pertama *lexicographic* tersebut dapat diketahui :
Jumlah anggota koset = $2^n = 2^3 = 8$
Jumlah koset = $2^{n-k} = 2^2 = 4$

Matriks cek paritas untuk *lexicodes* L_1^d dicari cara :

a. Merubah bentuk matriks generator *lexicodes* L_1^d, G, menjadi bentuk standar G’ = [I_k A] dengan menggunakan matriks permutasi P sesuai definisi 3.2.1.
b. Mencari matriks cek paritas untuk G’ dengan menggunakan H’ = [A^T I_{n-k}].
c. Untuk mencari H, yaitu matriks cek paritas untuk G, dilakukan dengan menggunakan matriks permutasi P^T, dimana P x P^T = I

Untuk *lexicodes* L_1^3 telah diperoleh bentuk matriks generator standar, jadi

G’= G = [1 11]. Dengan 2 bit terakhir sebagai matriks A. Jadi, H’ = H = $\begin{bmatrix}1 & 1 & 0\\ 1 & 0 & 1\end{bmatrix}$

Array standar untuk $\Lambda(1,3)$ adalah sebagai berikut, dengan kolom pertama berisi koset *leader*, dan baris pertama adalah adalah kode C, yaitu semua kodekata yang mungkin dari $\Lambda(1,3)$. Kolom paling kanan adalah sindrom s^T, dimana s = H. x^T, $\forall x \in C$ untuk masing-masing koset, dan bukan termasuk dalam *array* standar :

Array Standar		sindrom
000	111	00
001	110	01
010	101	10
100	011	11

Tabel 1

Dari array standar tersebut, diketahui bobot maksimum koset *leader* ρ = 1.
Dipilih $f_{lexi} = \lambda = [001]$ dengan cara :
1. Memilih koset *leader* yang mempunyai bobot paling besar diantara semua koset *leader*.

2. Memilih diantara koset *leader* tersebut yang memiliki urutan *lexicographic* paling kecil.
3. Jika koset *leader* bukan merupakan kodekata dengan urutan paling kecil dari kodekata-kodekata pada koset yang sama, dipilih diantara koset tersebut yang memiliki urutan *lexicographic* paling kecil..

Dari iterasi ini, diperoleh beberapa informasi, yaitu :

- $\rho = 1$
- $f_{lexi}(L_1^3) = \lambda = [001]$,
- $a \in V_{d-\rho-1}(Z_2)$, jadi $a = \{0,1\}$, maka $wt(a_1) = 0$, dan $wt(a_2) = 1$

Dengan menggunakan informasi-informasi di atas, dapat ditentukan hal-hal sebagai berikut :

a. Sekawan (*companion*) dari koset *leader* l pada tabel 1 atas λ =[001] adalah sebagai berikut :

Tabel 2

l	λ	$l+\lambda$	$k_\lambda(l)$
000	001	001	001
001	001	000	000
010	001	011	100
100	001	101	010

b. Bobot koset *leader* untuk lexicodes L_2^3 ditentukan menurut *corollary* 1 sebagai berikut :

$$wt(l') = \min\left(wt(a) + wt(l); d - \rho - wt(a) + wt(k_\lambda(l))\right).$$

Dengan $wt(a)$ adalah bobot dari $a \in V_{d-\rho-1}(Z_2)$, dan $wt(k_\lambda(l))$ adalah bobot dari sekawan koset *leader* l atas λ yang diperoleh dari tabel 2, maka

Tabel 3

$wt(a)+wt(l)$	$d-\rho-wt(a)+wt(k_\lambda(l))$	$wt(l')$
0 + 0 = 0	3 – 1 – 0 + 1 = 3	0
0 + 1 = 1	3 – 1 – 0 + 0 = 2	1
1	3	1
1	3	1
1 + 0 = 1	3 – 1 – 1 + 1 = 2	1
1 + 1 = 2	3 – 1 – 1 + 0 = 1	1
2	2	2

2	2	2

Pada tabel di atas, baris 1 sampai 4 menggunakan a = 0, dan baris 5 sampai 8 menggunakan a = 1.

c. Jika S_l merupakan sindrom dari lexicodes L_1^3, sindrom untuk *lexicodes* L_2^3 (ditulis $S_{l'}$) dapat dicari menggunakan S_l menurut *corollary* 2, yaitu $S_{l'} = \langle a|S_l\rangle$.

Dengan menggunakan $S_l = H \,.\, l^T$, maka

Tabel 4

a	S_l	$S_{l'} = \langle a\|S_l\rangle$
0	00	000
0	01	001
0	10	010
0	11	011
1	00	100
1	01	101
1	10	110
1	11	111

d. Sindrom dari sekawan koset *leader* l atas λ atau $S_{k_\lambda(l)}$ menurut teorema 5, $S_{k_\lambda(l)} = S_\lambda + S_l$. Maka dengan λ = 001 dan $H = \begin{bmatrix} 110 \\ 101 \end{bmatrix}$ diperoleh :

Tabel 5

S_λ	S_l	$S_{k_\lambda(l)}$
$H.[001]^T = [01]^T$	$H.\ [000]^T = [00]^T$	$[01]^T + [00]^T = [01]^T$
$H.[001]^T = [01]^T$	$H.\ [001]^T = [01]^T$	$[01]^T + [01]^T = [00]^T$
$[01]^T$	$[10]^T$	$[11]^T$
$[01]^T$	$[11]^T$	$[10]^T$

e. Koset *leader* untuk *lexicodes* L_2^3 dapat diturunkan dari koset *leader* *lexicodes* L_1^3 menurut teorema 3,

$$l' = \begin{cases} \langle 0|a|l\rangle \; jika \; wt(\langle 0|a|l\rangle) \le wt(\langle 1|\bar{a}|k_\lambda(l)\rangle) \\ \langle 1|\bar{a}|k_\lambda(l)\rangle \; jika \; wt(\langle 0|a|l\rangle) > wt(\langle 1|\bar{a}|k_\lambda(l)\rangle) \end{cases}$$

Dari informasi yang diperoleh sebelumnya diketahui bahwa nilai a = {0, 1}

Lexicodes L_1^3 memiliki koset *leader* $l \in$ S = {0,1,10,100}, Jika *l'* menyatakan koset *leader* dari *lexicodes* L_2^3, maka $l' \in S'$ dapat diperoleh dari *l*. Dengan menggunakan koset *leader* l dari tabel 1 dan $K_\lambda(l)$ dari tabel 2, diperoleh :

Tabel 6

0| *a* | *l*	$1 \mid \bar{a} \mid K_\lambda(l)$	*l'*∈ *S'*
0 0 000	1 1 001	00000
0 0 001	1 1 000	00001
0 0 010	1 1 100	00010
0 0 100	1 1 010	00100
0 1 000	1 0 001	01000
0 1 001	1 0 000	10000
0 1 010	1 0 100	01010
0 1 100	1 0 010	01100

Dapat dituliskan, *l'* ∈ *S'* = {0, 1, 10, 100, 1000, 10000, 01010, 01100}

2. *Iterasi kedua, membentuk* L_2^3

L_2^3 memiliki parameter ($n + \rho$, k +1, d) = (3 + 2, 2, 3) = (5, 2, 3), yaitu kode dengan matriks generator *lexicographic*nya $\Lambda(2,3)$. Dari kode tersebut, dapat diketahui:

Jumlah anggota koset = $2^n = 2^5 = 32$

Jumlah koset = $2^{n-k} = 2^3 = 8$

$\Lambda(2,3)$ dihasilkan dari

$\Lambda(1,3) \cup \langle 1^{d-\rho} \mid f_{lexi}(C)\rangle$, sehingga

$$\Lambda(2,3) = \Lambda(1,3) \cup \langle 1^{d-\rho} \mid f_{lexi}(C)\rangle$$
$$= [111] \cup \langle 1^{3-1} \mid 001\rangle$$
$$= [111] \cup \langle 11 \mid 001\rangle$$

Keterangan, karena dimensi kedua vektor tersebut tidak sama, maka untuk dapat menggabungkan kedua vektor tersebut, kita harus menyamakan dimensinya, yaitu dengan menambahkan bit 0 di depan vektor [111], sejumlah selisih dimensi kedua vektor tersebut sehingga diperoleh dimensi yang sama.

Maka diperoleh:

$$\Lambda(2,3) = \begin{bmatrix} 00111 \\ 11001 \end{bmatrix}$$

Matriks cek paritas untuk *lexicodes* L_2^3 dicari menggunakan cara yang sama dengan iterasi pertama di atas, jadi :

Matriks generator *lexicodes* L_2^3 = G = $\begin{bmatrix} 00111 \\ 11001 \end{bmatrix}$. G' = G x P

Jika diambil P = $\begin{bmatrix} 01000 \\ 00100 \\ 10000 \\ 00010 \\ 00001 \end{bmatrix}$,

diperoleh G' = $\begin{bmatrix} 10011 \\ 01101 \end{bmatrix}$

H' = $\begin{bmatrix} 01100 \\ 10010 \\ 11001 \end{bmatrix}$. Dicari matriks P^T sedemikian hingga P x P^T = I, dan ditemukan

$$P^T = \begin{bmatrix} 00100 \\ 10000 \\ 01000 \\ 00010 \\ 00001 \end{bmatrix}$$

Maka, H = H' x P^T = $\begin{bmatrix} 11000 \\ 00110 \\ 10101 \end{bmatrix}$

Dari iterasi sebelumnya dapat dibuat *array* standar untuk $\Lambda(2,3)$ berikut, dengan kolom paling kanan adalah sindrom untuk masing-masing koset:

Tabel 7

Array Standar				Sindrom
00000	00111	11001	11110	000
00001	00110	11000	11111	001
00010	00101	11011	11100	010
00100	00011	11101	11010	011
01000	01111	10001	10110	100
10000	10111	01001	01110	101
01010	01101	10011	10100	110
01100	01011	10101	10010	111

Dari *array* standar di atas, dapat dilihat bahwa bobot maksimum koset *leader* ρ = 2. $f_{lexi}(L_2^3)$ dicari dengan cara yang serupa pada iterasi pertama, dan diperoleh $f_{lexi}(L_2^3) = \lambda =$ [01010].

Dari iterasi ini, diperoleh beberapa informasi, yaitu:

- ρ = 2
- $f_{lexi}(L_2^3) = \lambda$ = [01010]

- $a \in V_{d-\rho-1}(Z_2)$, jadi tidak ada nilai *a* yang mungkin dan $\omega_a = 0$.

Dengan menggunakan informasi-informasi di atas, dapat ditentukan hal-hal sebagai berikut :

a. Sekawan (*companion*) dari koset *leader l* pada tabel 7 atas $\lambda = [01010]$ adalah sebagai berikut :

Tabel 8

l	λ	$l + \lambda$	$k_\lambda(l)$
00000	01010	01010	01010
00001	01010	01011	01100
00010	01010	01000	01000
00100	01010	01110	100[illegible]
01000	01010	00010	000[illegible]
10000	01010	11010	00100
01010	01010	00000	00000
01100	01010	00110	00001

b. Bobot koset *leader* untuk *lexicodes* L_2^3 ditentukan menurut *corollary* 1 sebagai berikut :

Tabel 9

$wt(a) + wt($	$d - \rho - wt(a) + wt(k$	$wt(l')$
0 + 0 = 0	3 – 2 – 0 + 2 = 3	0
0 + 1 = 1	3 – 2 – 0 + 2 = 3	1
1	2	1
1	2	1
1	2	1
1	2	1
2	1	1
2	2	2

c. Jika S_l merupakan sindrom dari *lexicodes* L_2^3, sindrom untuk *lexicodes* L_3^3 (ditulis $S_{l'}$) dapat dicari menggunakan S_l menurut *corollary* 2, yaitu $S_{l'} = \langle a|S_l \rangle$.

Dengan koset *leader l* dari tabel 7 dan S_l = H . l^T, maka

a	S_l	$S_{l'}$
-	000	000
-	001	001
-	010	010
-	011	011
-	100	100
-	101	101
-	110	110
-	111	111

Tabel 10

d. Sindrom dari sekawan koset *leader l* atas λ atau $S_{k_\lambda(l)}$ menurut teorema 5, $S_{k_\lambda(l)} = S_\lambda + S_l$. Maka dengan λ = [01010] dan H = $\begin{bmatrix} 11000 \\ 00110 \\ 10101 \end{bmatrix}$ diperoleh :

Tabel 11

S_λ	S_l	$S_{k_\lambda(l)}$
H.[01010]^T = [110]^T	H. [00000]^T= [000]^T	[110]^T + [000]^T = [110]^T
H.[01010]^T = [110]^T	H. [00001]^T= [001]^T	[110]^T + [001]^T =[111]^T
[110]^T	[010]^T	[100]^T
[110]^T	[011]^T	[101]^T
[110]^T	[100]^T	[110]^T
[110]^T	[101]^T	[011]^T
[110]^T	[110]^T	[000]^T
[110]^T	[111]^T	[001]^T

e. Koset *leader* untuk *lexicodes* L_3^3 dapat diturunkan dari koset *leader lexicodes* L_2^3 menurut teorema 3,

$$l' = \begin{cases} \langle 0|a|l \rangle \text{ jika } wt(\langle 0|a|l \rangle) \le wt(\langle 1|\bar{a}|k_\lambda(l) \rangle) \\ \langle 1|\bar{a}|k_\lambda(l) \rangle \text{ jika } wt(\langle 0|a|l \rangle) > wt(\langle 1|\bar{a}|k_\lambda(l) \rangle) \end{cases}$$

Dari perhitungan sebelumnya diketahui bahwa tidak ada nilai *a* yang memenuhi. Jadi perhitungan *l'* diubah menjadi

$$l' = \begin{cases} \langle 0|l \rangle \text{ jika } wt(\langle 0|l \rangle) \le wt(\langle 1|k_\lambda(l) \rangle) \\ \langle 1|k_\lambda(l) \rangle \text{ jika } wt(\langle 0|l \rangle) > wt(\langle 1|k_\lambda(l) \rangle) \end{cases}$$

Lexicodes L_2^3 memiliki koset leader sebagai berikut :

$l \in$ S = {0,0,10,100, 1000, 10000, 01010, 01100},

Jika l' menyatakan koset leader dari lexicodes L_3^3, maka $l' \in$ S' dapat diperoleh dari l . Dengan menggunakan koset leader l dari tabel 7 dan $K_\lambda(l)$ dari tabel 8, diperoleh :

Tabel 12

$0 \mid l$	$1 \mid K_\lambda(l)$	Korespondensi $l' \in S'$
0 00000	1 01010	0 00000
0 00001	1 01100	0 00001
0 00010	1 01000	0 00010
0 00100	1 10000	0 00100
0 01000	1 00010	0 01000
0 10000	1 00100	0 10000
0 01010	1 00000	1 00000
0 01100	1 00001	001100

Dapat dituliskan, $l' \in S'$ = {0, 1, 10, 100, 1000, 10000, 100000, 001100}.

3. *Iterasi ketiga, membentuk lexicodes* L_3^3

L_3^3 memiliki parameter $(n + \rho, k + 1, d) = (5 + 1, 2 + 1, 3) = (6, 3, 3)$

Maka dapat diketahui bahwa :

Jumlah anggota koset= $2^n = 2^6 = 64$

Jumlah koset = $2^{n-k} = 2^3 = 8$

Matriks Generator L_3^3 adalah $\Lambda(3,3)$ yang dihasilkan dari

$\Lambda(2,3) \cup \langle 1^{d-\rho} \mid f_{lexi}(C) \rangle$, sehingga

$\Lambda(3,3) = \Lambda(2,3) \cup \langle 1^{d-\rho} \mid f_{lexi}(C) \rangle$
$= \langle 111 \rangle \cup \langle 1 \mid 01010 \rangle$

Diperoleh:

$$\Lambda(3,3) = \begin{bmatrix} 000111 \\ 011001 \\ 101010 \end{bmatrix}$$

Matriks cek paritas untuk *lexicodes* L_2^3 dicari menggunakan cara yang sama dengan iterasi sebelumnya, dan diperoleh

$$H = \begin{bmatrix} 111000 \\ 100110 \\ 010101 \end{bmatrix}$$

Dari iterasi sebelumnya dapat dibuat *array* standar untuk *lexicodes* L_3^3:

Tabel 13

Array Standar								s
00 00 00	00 01 11	01 10 01	10 10 10	01 11 10	10 11 01	11 00 11	11 01 00	0 0 0
00 00 01	00 01 10	01 10 00	10 10 11	01 11 11	10 11 00	11 00 10	11 01 01	0 0 1
00 00 10	00 01 01	01 10 11	10 10 00	01 11 00	10 11 11	11 00 01	11 01 10	0 1 0
00 01 00	00 00 11	01 11 01	10 11 10	01 10 10	10 10 01	11 01 11	11 00 00	0 1 1
00 10 00	00 11 11	01 00 01	10 00 10	01 01 10	10 01 01	11 10 11	11 11 00	1 0 0
01 00 00	01 01 11	00 10 01	11 10 10	00 11 10	11 11 01	10 00 11	10 01 00	1 0 1
10 00 00	10 01 11	11 10 01	00 10 10	11 11 10	00 11 01	01 00 11	01 01 00	1 1 0
00 11 00	**00 10 11**	01 01 01	10 01 10	01 00 10	10 00 01	11 11 11	11 10 00	1 1 1

Dari *Array* standar di atas, dapat dilihat bahwa bobot maksimum koset *leader* ρ = 2. Koset *leader* dengan ρ = 2 tersebut adalah tunggal. Dari koset tersebut dicari kode dengan urutan *lexicographic* paling kecil untuk diambil sebagai f_{lexi}, dan diperoleh $f_{lexi} = \lambda$ = [001011].

Dari iterasi ini, diperoleh beberapa informasi, yaitu :

- $\rho = 2$
- $f_{lexi}(L_2^3) = \lambda$ = [001011]
- $a \in V_{d-\rho-1}(Z_2)$, jadi tidak ada nilai *a* yang mungkin dan $\omega_a = 0$

Dengan menggunakan informasi-informasi di atas, dapat ditentukan hal-hal sebagai berikut :

a. Sekawan (*companion*) dari koset *leader* *l* pada tabel 13 atas $\lambda = [001011]$ adalah sebagai berikut :

Tabel 14

l	λ	$l + \lambda$	$k_\lambda(l)$
000000	001011	001011	001100
000001	001011	001010	100000
000010	001011	001001	010000
000100	001011	001111	001000
001000	001011	000011	000100
010000	001011	011011	000010
100000	001011	101011	000001
001100	001011	000111	000000

b. bobot koset *leader* untuk *lexicodes* L_4^3 ditentukan menurut *corollary* 1 sebagai berikut:

Tabel 15

$wt(a)+wt($	$d-\rho-wt(a)+wt(k$	$wt(l')$
0 + 0 = 0	3 – 2 – 0 + 2 = 3	0
0 + 1 = 1	3 – 2 – 0 + 1 = 2	1
1	2	1
1	2	1
1	2	1
1	2	1
1	2	1
2	1	1

c. Jika S_l merupakan sindrom dari *lexicodes* L_3^3, sindrom untuk *lexicodes* L_4^3 (ditulis $S_{l'}$) dapat dicari menggunakan S_l menurut *corollary* 2, yaitu $S_{l'} = \langle a|S_l\rangle$. Dengan menggunakan koset *leader* l dari tabel 13, dan $S_l = H \,.\, l^T$, maka

Tabel 16

a	S_l	$S_{l'}$
-	000	000
-	001	001
-	010	010
-	011	011
-	100	100
-	101	101
-	110	110
-	111	111

d. Sindrom dari sekawan koset *leader* l atas λ atau $S_{k_\lambda(l)}$ menurut teorema 5, $S_{k_\lambda(l)} = S_\lambda + S_l$. Maka dengan λ = [001011] dan H = $\begin{bmatrix}111000\\100110\\010101\end{bmatrix}$ diperoleh:

Tabel 17

S_λ	S_l	$S_{k_\lambda(l)}$
.$[001011]^T$ = $[111]^T$	H. $[000000]^T$= $[000]^T$	$[111]^T + [000]^T$ = [1[illegible]1]^T
.$(001011)^T$ = $[111]^T$	H. $[000001]^T$= $[001]^T$	$[111]^T + [001]^T$ = [1[illegible]0]^T
$[111]^T$	$[010]^T$	$[101]^T$
$[111]^T$	$[011]^T$	$[100]^T$
$[111]^T$	$[100]^T$	$[011]^T$
$[111]^T$	$[101]^T$	$[010]^T$
$[111]^T$	$[110]^T$	$[001]^T$
$[111]^T$	$[111]^T$	$[000]^T$

e. Koset *leader* untuk *lexicodes* L_4^3 dapat diturunkan dari koset *leader* *lexicodes* L_3^3 menurut teorema 3,

$$l' = \begin{cases} \langle 0|a|l\rangle \; jika \; wt(\langle 0|a|l\rangle) \le wt(\langle 1|\bar{a}|k_\lambda(l)\rangle) \\ \langle 1|\bar{a}|k_\lambda(l)\rangle \; jika \; wt(\langle 0|a|l\rangle) > wt(\langle 1|\bar{a}|k_\lambda(l)\rangle) \end{cases}$$

Lexicodes L_3^3 memiliki koset *leader* sebagai berikut :

$l \in$ S = {0,1,10,100, 1000, 10000, 01010, 01100},

Jika l' menyatakan koset *leader* dari *lexicodes* L_4^3, maka $l' \in$ S' dapat diperoleh dari l . Dengan menggunakan l dari tabel 13 dan $K_\lambda(l)$ dari tabel 14, diperoleh:

Tabel 18

0 | l	1 | $K_\lambda(l)$	$l' \in S'$
0 00000	1 001100	0 000000
0 000001	1 100000	0 000001
0 000010	1 010000	0 000010
0 000100	1 001000	0 000100
0 001000	1 000100	0 001000
0 010000	1 000010	0 010000
0 100000	1 000001	0 100000
0 001100	1 000000	1 000000

Dapat dituliskan, $l' \in S'$ = {0, 1, 10, 100, 1000, 10000, 100000, 1000000}

4. Iterasi keempat, membentuk lexicodes L_4^3

L_4^3 memiliki parameter (n + ρ, k +1+, d) = (6 + 1, 3 + 1, 3) = (7, 4, 3)

Maka dapat diketahui bahwa :

Jumlah anggota koset = 2^n = 2^7 = 128

Jumlah koset = 2^{n-k} = 2^3 = 8

Matriks generator *lexicodes* L_4^3 adalah $\Lambda(4,3)$ yang dihasilkan dari $\Lambda(3,3) \cup \langle 1^{d-\rho} \,|\, f_{lexi}(C)\rangle$,

Maka,

$$\Lambda(4,3) = \Lambda(3,3) \cup \langle 1^{d-\rho} |\, f_{lexi}(C)\rangle$$

$$= \begin{bmatrix}000111\\011001\\101010\end{bmatrix} \cup \langle 1|\, 001011\rangle$$

Sehingga diperoleh

$$G = \Lambda(4,3) = \begin{bmatrix}0000111\\0011001\\0101010\\1001011\end{bmatrix}$$ sebagai matriks generator kode (7, 4, 3)

Matriks cek paritas untuk G di atas adalah

$$H = \begin{bmatrix} 1111000 \\ 1100110 \\ 1010101 \end{bmatrix}$$

Matriks generator G tersebut dapat diubah menjadi matriks generator bentuk baku G'.

Jika diambil P matriks permutasi ukuran 7 x 7,

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

Maka G' = GP

$$G' = \begin{bmatrix} 0000111 \\ 0011001 \\ 0101010 \\ 1001011 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1000011 \\ 0100101 \\ 0010110 \\ 0001111 \end{bmatrix}$$

3. KESIMPULAN

Kode *Lexicographic* dapat digunakan untuk membentuk kode-kode lain, seperti kode Hamming (7, 4, 3). Untuk dapat menghasilkan kode *lexicographic* dilakukan dengan menggunakan algoritma Greedy. Algoritma Greedy terdiri dari dua macam, yaitu konstruksi *lexicographic* dan konstruksi Greedy. Konstruksi *lexicographic* terdiri atas iterasi-iterasi yang mengulang perhitungan pada iterasi sebelumnya. Jumlah iterasi untuk konstruksi *lexicographic* sama dengan jumlah dimensi kode yang akan dibentuk Untuk membentuk kode Hamming (7, 4, 3) yaitu kode dengan panjang 7, dimensi 4, dan jarak minimum 3, maka iterasi yang diperlukan berjumlah 4 iterasi. Dari konstruksi *Lexicographic* dapat langsung diperoleh *array* standar yang berguna pada proses pendekodean yaitu untuk mencari kesalahan (*error*) pada pesan.

4. DAFTAR PUSTAKA

[1] Acosta, Kishion, dkk.2001. *Lexicographic and non-Lexicographic Greedy Codes*. University of California, Los angeles. http://www.wyki.org/Lexicographic_code, diakses bulan November 2008

[2] *Anton*, Howard.1984. *Aljabar Linier Elementer*. Erlangga : Jakarta.

[3] *Conway*, J.H & N.J.A. Sloane. *Lexicographic Codes : Error - Correcting codes from Game Theory*. IEEE Transaction on Information Theory, May 1986. http://www.research.att.com/~njas/doc/lex.pdf, diakses bulan November 2008

[4] Gilbert, Jimmie & Linda Gilbert. *Element of Modern Algebra*.

[5] *Harjito*, Drs,dkk.2006. *Buku Ajar Aljabar 1*. Universitas Diponegoro.

[6] *Hillman*, Abraham.P & Gerald L. Alexanderson. 1993. *Abstract Algebra, a First Undergraduate Course*, *fifth edition*. PWS Publishing Company : Boston MA.

[7] *Kanemasu*, Melisa. *Golay Codes*. http://www.math.mit.edu/phase2/UJM/vol1/MKANEM~1.pdf.

[8] Rizki. K, Ikhsan.2008. *Membangun Kode Golay (24, 12, 8) dengan Matriks Generator*. Jurusan Matematika, FMIPA. Universitas Diponegoro.

[9] Spasov, Dejan. 2006. *Implementing The Lexicografik Construction*. Boston University. http://nislab.bu.edu/nislab/projects/lexicode/project%20paper.pdf, diakses bulan November 2008

[10] Suryadi, H.S & S. Harini Machmudi. 1984. *Teori dan Soal Pendahuluan Aljabar Linier*. Ghalia Indonesia : Jakarta.

[11] Trachtenberg, Ari. *Designing* Lexicographic *Codes with Given Trellis Complexity*. IEEE Transaction on Information Theory, January 2002. http://people.bu.edu/trachten/ diakses bulan November 2008

[12] Widyaningsih, Santi. 2008. *Deteksi dan Koreksi Error pada pesan digital dengan kode Hamming*. Jurusan Matematika, FMIPA. Universitas Diponegoro.

[13] Vanstone, S.A & Oorschot, P.C. 1989. *An Introduction to Error Correcting Codes with Applications*. London : Kluwer Academic Publisher.