BAB II

KAJIAN TEORI

A. Lapangan Berhingga

Himpunan merupakan suatu kumpulan obyek-obyek yang didefinisikan dengan jelas pada suatu batasan-batasan tertentu. Contoh himpunan hewan berkaki empat H4 ={sapi, kambing, kuda, badak, harimau, unta}. Contoh lain himpunan bilangan prima kurang dari 12 yaitu A = {2,3,5,7,11}.

Definisi 2.1 (Sukirman, 2010: 68).

Himpunan tak kosong G yang memiliki operasi biner • disebut suatu grup apabila memenuhi aksioma-aksioma berikut ini:

i. Operasi • pada G bersifat asosiatif

$\forall a, b, c \in G$ *berlaku* $(a \bullet b) \bullet c = a \bullet (b \bullet c)$.

ii. G memuat elemen identitas, yaitu e.

$\exists\, e \in G, \forall\, a \in G$ *berlaku* $a \bullet e = e \bullet a = a$.

iii. Setiap unsur G memiliki invers di dalam G juga.

$\forall a \in G, \forall a^{-1} \in G$ *sedemikian sehingga* $a \bullet a^{-1} = a^{-1} \bullet a = e$. *Maka* a^{-1} *disebut invers dari a.*

Suatu grup G dengan operasi biner • ditulis (G, •). Jika (G, •) suatu grup yang bersifat komutatif yaitu $\forall a, b \in G$ berlaku $a \bullet \mathrm{b} = \mathrm{b} \bullet \mathrm{a}$ maka (G, •) disebut grup komutatif atau grup abelian. Apabila terdapat $H \subset G$ dan $H \neq \emptyset$, maka H disebut kompleks dari G (Sukirman, 2010:97). Misalkan terdapat $(G, °)$ suatu grup dan H kompleks dari G, dan apabila $(H, °)$ suatu grup, maka dikatakan H adalah subgrup dari G. Operasi pada grup G dan H harus sama (Sukirman, 2010:99).

Definisi 2.2 (Sukirman, 2010:174).

Diberikan H suatu subgrup dari grup G, sehingga H disebut subgrup normal dari G (diberi simbol $H \lhd G$) jika dan hanya jika $\forall g \in G$ berlaku $gH = Hg$.

Contoh 2.1.

i. (B,+) dengan B adalah himpunan bilangan bulat, merupakan suatu grup dengan elemen identitas 0 dan setiap elemen mempunyai *invers* terhadap penjumlahan.

ii. Diberikan B9 = {0,1,2,3,4,5,6,7,8}, yaitu himpunan bilangan bulat tak negatif kurang dari 9. (B9,+), operasi $+$ pada B9 bersifat asosiatif, memiliki elemen identitas 0 tetapi *invers*nya tidak termuat di B9 sehngga B9 tidak tertutup terhadap operasi $+$. Dengan demikian (B9,+) bukan merupakan grup. Sedangkan, (B9, $\bullet$) dengan B9 merupakan himpunan bilangan bulat tak negatif kurang dari 9 dan operasi $\bullet$ adalah operasi perkalian yang bersifat asosiatif tetapi hasil perkalian elemen-elemennya tidak termuat di B9 sehingga B9 tidak tertutup. Jadi, (B9, $\bullet$) juga bukan suatu grup.

Definisi 2.3 (Fraleigh, 1989: 167).

Suatu ring $(R, +, \bullet)$ *adalah himpunan tak kosong R yang dilengkapi dengan dua operasi biner yang ditunjukkan dengan tanda* $("+")$ *untuk penjumlahan dan* $("\bullet")$ *untuk perkalian serta memenuhi aksioma-aksioma sebagai berikut:*

i. $(R, +)$ *merupakan grup komutatif atau grup abelian*

ii. $(R, \bullet)$ *memenuhi sifat asosiatif*

iii. Memenuhi sifat distributif kiri dan distributif kanan, untuk setiap $a, b, c \in R$ *berlaku* $a(b + c) = ab + ac$ *dan* $(a + b)c = ac + bc$.

Suatu *ring* $(R, \bullet)$ dikatakan *ring* komutatif apabila operasi perkalian ("•") pada R bersifat komutatif yaitu untuk setiap $a, b \in R$ berlaku $a \bullet b = b \bullet a$.

Contoh 2.2.

Perhatikan himpunan-himpunan berikut.

B9 = {0,1,2,3,4,5,6,7,8}, yaitu himpunan bilangan bulat tak negatif kurang dari 9.

Z_{10} = {[0], [1], [2], [3], [4], [5], [6], [7], [8], [9]}, yaitu himpunan semua kelas bilangan bulat modulo 10.

Z_{11} = {[0], [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]}, yaitu himpunan semua kelas bilangan bulat modulo 11.

Pada himpunan-himpunan tersebut didefinisikan operasi + dan • sehingga dapat dijelaskan sebagai berikut.

i. $(Z_{10}, +, \bullet)$ merupakan suatu *ring* karena himpunan Z_{10} memenuhi semua syarat aksioma suatu *ring* yaitu $(Z_{10}, +)$ merupakan grup abelian, $(Z_{10}, \bullet)$ bersifat asosiatif, dan memenuhi sifat distribusi kanan dan kiri.

ii. $(Z_{11}, +, \bullet)$ merupakan suatu *ring* karena himpunan Z_{11} memenuhi semua syarat aksioma suatu *ring* yaitu $(Z_{11}, +)$ merupakan grup abelian, $(Z_{11}, \bullet)$ bersifat asosiatif, dan memenuhi sifat distribusi kanan dan kiri.

iii. $(B9, +, \bullet)$ bukan suatu *ring* karena (B9,+) maupun (B9,•) bukan suatu grup seperti yang telah dijelaskan pada Contoh 2.1.

Definisi 2.4 (Sukirman, 2006: 35)

Diberikan $(R, +, \bullet)$ *suatu ring. Apabila* $S \neq \phi$, $S \subset R$ *dan* $(S, +, \bullet)$ *adalah suatu ring, maka dikatakan bahwa S adalah subring dari* R.

Contoh 2.3.

Misalkan B adalah *ring* bilangan bulat terhadap penjumlahan dan perkalian aritmetik. B_2 adalah himpunan semua bilangan genap dan B_2 terhadap penjumlahan dan perkalian aritmetik adalah *ring* dan karena B_2 adalah himpunan bagian dari B, maka B_2 adalah subring dari B.

Suatu sub*ring* yang memiliki sifat khusus disebut dengan ideal. Berikut diberikan definisi ideal.

Definisi 2.5 (Sukirman, 2006: 50).

Diberikan R suatu ring dan S subring dari R, maka:

i. *S disebut ideal kanan dari R, jika $\forall a \in S, \forall r \in R$ berlaku $ar \in S$.*
ii. *S disebut ideal kiri dari R, jika $\forall a \in S, \forall r \in R$ berlaku $ra \in S$.*
iii. *S disebut ideal dua sisi dari R, jika $\forall a \in S, \forall r \in R$ berlaku $ar \in S$ dan $ra \in S$.*

Teorema 2.1 (Sukirman, 2006: 51).

Diberikan R adalah ring, $S \neq \phi$, $S \subset R$. S adalah ideal dari R jika dan hanya jika:

i. $\forall a, b \in S, a - b \in S;$
ii. $\forall a \in S, \forall r \in R, ar \in S$ dan $ra \in S$.

Bukti:

($\Rightarrow$) Diketahui S suatu ideal dari R maka menurut definisi, S adalah dubring dari R yang memenuhi $\forall a, b \in S$, $a - b \in S$ dan $\forall a \in S$, $\forall r \in R$ berlaku $ar \in S$ dan $ra \in S$.

(⇐) Apabila $\forall a, b \in S$ dan menurut ketentuan $a - b \in S$ dan $ab \in S$, maka S adalah subring dari R. Selanjutnya, karena $\forall a \in S$, $\forall r \in R$, berlaku$ar \in S$ dan $ra \in S$ maka S adalah ideal dari R. ■

Contoh 2.4.

Misalkan $Z_{12} = \{[0], [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]\}$ adalah himpunan semua kelas bilangan bulat modulo 12. $(Z_{12}, +, \bullet)$ merupakan ring dengan operasi penjumlahan dan perkalian modulo 12 maka $N = \{[0], [4], [8]\}$ adalah ideal dari Z_{12} yang tidak memuat pembagi nol.

Definisi 2.6 (Hartley & Howkes, 1970: 59).

Suatu ideal S atas daerah integral R disebut ideal utama atas R jika S dibangun oleh elemen tunggal $a \in R$, sedemikian sehingga $S = aR$.

Contoh 2.5.

Jika R adalah daerah integral, maka ideal $\{0\}$ dan R adalah ideal utama yang dibangun oleh masing-masing 0 dan 1.

Definisi 2.7 (Herstein, 1996:148).

Suatu ideal I atas ring R adalah ideal maksimal dari R jika dan hanya jika ideal dari R memuat I yaitu I itu sendiri dan R.

Definisi 2.8 (Vanstone dan Oorschot, 1989: 21).

Sebuah lapangan F merupakan himpunan elemen-elemen tertutup yang memuat dua operasi biner yaitu penjumlahan dan perkalian dinotasikan dengan "+" dan "•" sehingga aksioma-aksioma di bawah ini terpenuhi untuk semua $a, b, c \in F$.

i. $a + (b + c) = (a + b) + c$

ii. $a + b = b + a$

iii. *ada elemen* $0 \in F$ *sedemikian sehingga* $a + 0 = a$

iv. *ada elemen* $-a \in F$ *sedemikian sehingga* $a + (-a) = 0$

v. $a \bullet (b \bullet c) = (a \bullet b) \bullet c$

vi. $a \bullet b = b \bullet a$

vii. *ada elemen* $1 \in F$ *sedemikian sehingga* $a \bullet 1 = a$

viii. *untuk setiap* $a \neq 0$, *ada elemen* $a^{-1} \in F$ *sedemikian sehingga* $a \bullet a^{-1} = 1$

ix. $a \bullet (b + c) = a \bullet b + a \bullet c$.

Berikut diberikan contoh lapangan.

Contoh 2.6.

Diberikan himpunan $Z_7 = \{[0], [1], [2], [3], [4], [5], [6]\}$, yang merupakan himpunan semua kelas bilangan bulat modulo 7 dengan penjumlahan modulo 7 dan perkalian modulo 7 yaitu $(Z_7, +, \bullet)$, maka $(Z_7, +, \bullet)$ yang merupakan suatu lapangan. Berikut diberikan bukti dengan menggunakan Tabel *Cayley* Z_7 :

Tabel 2.1. Tabel *Cayley* Z_7 Hasil Penjumlahan Modulo 7

$+_7$	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[1]	[1]	[2]	[3]	[4]	[5]	[6]	[0]
[2]	[2]	[3]	[4]	[5]	[6]	[0]	[1]
[3]	[3]	[4]	[5]	[6]	[0]	[1]	[2]
[4]	[4]	[5]	[6]	[0]	[1]	[2]	[3]
[5]	[5]	[6]	[0]	[1]	[2]	[3]	[4]
[6]	[6]	[0]	[1]	[2]	[3]	[4]	[5]

diagonal utama

Tabel 2.2. Tabel *Cayley* Z_7 Hasil Perkalian Modulo 7

$*_7$	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[2]	[0]	[2]	[4]	[6]	[1]	[3]	[5]
[3]	[0]	[3]	[6]	[2]	[5]	[1]	[4]
[4]	[0]	[4]	[1]	[5]	[2]	[6]	[3]
[5]	[0]	[5]	[3]	[1]	[6]	[4]	[2]
[6]	[0]	[6]	[5]	[4]	[3]	[2]	[1]

diagonal utama

Memperhatikan tabel *Cayley* Z_7 untuk penjumlahan modulo 7 memenuhi sifat tertutup, elemen nolnya adalah [0], *invers* terhadap penjumlahan modulo 7, yaitu $-[0] = [0]$, $-[1] = [6]$, $-[2] = [5]$, $-[3] = [4]$, $-[4] = [3]$, $-[5] = [2]$, $-[6] = [1]$. Tabel simetris terhadap diagonal utama, sehingga penjumlahan modulo 7 maupun perkalian modulo 7 bersifat komutatif. Himpunan Z_7 terhadap perkalian modulo 7 bersifat tertutup, memiliki elemen kesatuan yaitu [1], *invers* terhadap perkalian modulo 7, yaitu $[1]^{-1} = [1]$, $[2]^{-1} = [4]$, $[3]^{-1} = [5]$, $[4]^{-1} = [2]$, $[5]^{-1} = [3]$, $[6]^{-1} = [6]$. Jadi setiap elemen Z_7 terhadap operasi perkalian dan penjumlahan memiliki *invers*. Terbukti bahwa $(Z_7, +, \bullet)$ merupakan suatu lapangan.

Suatu lapangan yang memuat sebanyak elemen berhingga disebut dengan lapangan berhingga. Himpunan kelas bilangan bulat modulo n yang dinotasikan dengan Z_n atas operasi standar penjumlahan dan perkalian modulo n disebut lapangan.

Teorema 2.2 (Vanstone & Oorschot , 1989 : 22).

Himpunan Z_n merupakan lapangan berhingga jika dan hanya jika n bilangan prima.

Bukti:

(⇒) Akan ditunjukkan himpunan Z_n merupakan lapangan berhingga jika n bilangan prima.

Diketahui n bukan bilangan prima, maka $n = ab$ dengan $a, b \in$ bilangan prima, $a > 1,\ b < n - 1$. Karena Z_n merupakan lapangan, maka setiap elemen tak nolnya pasti memiliki *invers*. Misalkan c adalah *invers* dari b, berarti $bc \equiv 1(mod\ n)$ dan $abc \equiv a(mod\ n)$. Karena $n = ab$ maka $ab \equiv 0(mod\ n)$, akibatnya $0 \equiv a(mod\ n)$ yang berarti $n = a$ sehingga n prima. Hal ini kontradiksi dengan pengandaian bahwa n bukan bilangan prima. Jadi n bilangan prima.

(⇐)Akan ditunjukkan jika n bilangan prima maka himpunan Z_n merupakan lapangan berhingga.

Diketahui n bilangan prima. Untuk dapat membuktikan Z_n merupakan lapangan, maka akan dibuktikan bahwa setiap elemen tak nol mempunyai *invers*. Karena n adalah bilangan prima, maka $\text{FPB}(n, a) = 1$, untuk $0 < a < n$. Akibatnya terdapat bilangan bulat x dan y sedemikian sehingga $x.n + y.a = 1$ yang berarti $y.a \equiv 1(mod\ n)$, diperoleh $y \equiv a^{-1}(mod\ n)$. Jadi terbukti bahwa setiap elemen tak nolnya mempunyai *invers*. Jadi Z_n adalah lapangan jika n prima. Karena Z_n memiliki elemen bilangan berhingga maka Z_n adalah lapangan berhingga. ■

Berikut diberikan contoh lapangan berhingga.

Contoh 2.7.

$Z_2, Z_3, Z_5, Z_7, Z_{11}$ dan Z_{13} merupakan lapangan berhingga, tetapi Z_9 dan Z_{15} bukan lapangan berhingga karena pada operasi perkalian untuk elemen [3]

di Z_9 serta elemen [3] dan [5] di Z_{15} tidak mempunyai *invers*. Selain itu Z_{10} juga bukan lapangan berhingga karena pada operasi perkalian untuk elemen [2] dan [5] tidak mempunyai *invers*.

Diberikan *ring* R dan ideal I atas *ring* R. Karena R adalah grup atas penjumlahan dan I adalah subgrup normal dari R, dapat dibentuk suatu grup faktor $R/I = \{r + I \mid r \in R\}$. Dengan menganalogikan grup atas koset-koset, didefinisikan hasil kali dua koset $(s + I).(t + I) = st + I$.

Teorema 2.3 (Gallian, 2006:269)

Suatu ring komutatif R dengan elemen kesatuan dan I adalah ideal dari R. Suatu R/I adalah lapangan jika dan hanya jika I adalah ideal maksimal.

Bukti:

($\Rightarrow$) Akan ditunjukkan suatu R/I adalah lapangan jika I adalah ideal maksimal. Diketahui R/I lapangan dan J ideal dari R yang memuat I. Diketahui R suatu ring komutatif dengan elemen kesatuan dan I adalah ideal dari R. Diberikan $j \in J$ tetapi $j \notin I$. Suatu $j + I$ adalah elemen tak nol dari R/I, jika terdapat elemen $k + I$ sedemikian hingga $(j + I).(k + I) = jk + I = 1 + I$, yaitu identitas perkalian dari R/I karena $j \in J$ maka terdapat $jk \in J$ dan $1 - jk \in I \subset J$. Sehingga, $1 = (1 - jk) + jk \in J$. Oleh karena itu $J = R$, terbukti I ideal maksimal.

($\Leftarrow$) Akan ditunjukkan jika I adalah ideal maksimal.maka R/I adalah lapangan Diketahui R/I lapangan dan J ideal dari R yang memuat I.Diketahui I ideal maksimal dan $j \in R$ tetapi $j \notin I$. Hal ini menunjukkan bahwa $j + I$ memiliki invers perkalian. Memandang $J = \{jr + i | r \in R, i \in I\}$. Ini adalah ideal dari R

yang memuat I. Karena I adalah ideal maksimal maka $J = R$. Sehingga, $1 \in J$, $1 = jk + i'$, dengan $i' \in I$, maka

$$1 + I = jk + i' + I = jk + I = (j + I)(k + I). \qquad \blacksquare$$

Teorema 2.4 (Gallian, 2006: 311).

Diberikan lapangan F dan $f(x) \in F[x]$, ideal $\langle f(x) \rangle$ adalah ideal maksimal di $F[x]$ jika dan hanya jika $f(x)$ polinomial tak tereduksi atas lapangan F.

Bukti:

($\Rightarrow$) Diketahui $\langle f(x) \rangle$ ideal maksimal di $F[x]$. Polinomial $f(x)$ juga bukan polinomial nol maupun elemen identitas di $F(x)$, karena $\{0\}$ maupun $F[x]$ adalah ideal maksimal di $F[x]$. Jika $f(x) = g(x)h(x)$ adalah faktor dari $f(x)$ atas lapangan F, maka $\langle f(x) \rangle \subseteq \langle g(x) \rangle \subseteq F[x]$. Sedemikian, $\langle f(x) \rangle = \langle g(x) \rangle$ atau $F[x] = \langle g(x) \rangle$.

($\Leftarrow$) Diketahui $f(x)$ polinomial tak tereduksi atas F. Diberikan I suatu ideal di $F[x]$ sedemikian sehingga $\langle f(x) \rangle \subseteq I \subseteq F[x]$. Karena $F[x]$ adalah daerah asal ideal utama , maka $I = \langle g(x) \rangle$ untuk suatu $g(x)$ di $F[x]$. Jadi, untuk $\langle f(x) \rangle \subseteq \langle g(x) \rangle$ maka $f(x) = g(x)h(x)$, dengan $h(x) \in F[x]$. Karena $f(x)$ polinomial tak tereduksi atas F, maka $g(x)$ adalah konstan atau $h(x)$ konstan. $\blacksquare$

Polinomial tak tereduksi adalah polinomial yang tidak dapat dinyatakan sebagai hasil perkalian dari dua polinomial dengan derajat yag lebih kecil dari derajat $f(x)$ dalam $Z_p[x]$. Himpunan $Z_p[x]$ adalah semua polinomial dalam x atas lapangan Z_p (p prima) dengan setiap polinomial berderajat hingga.

Akibat 2.1 (Gallian, 2006: 311).

Jika F adalah lapangan dan $f(x)$ polinomial tak tereduksi atas F maka $F[x]/\langle f(x)\rangle$ adalah lapangan.

Contoh 2.8.

Diberikan $\langle x^2+x+1\rangle$ adalah ideal maksimal di $Z_2(x)$ sehingga $Z_2(x)/\langle x^2+x+1\rangle$ adalah lapangan.

Lapangan berhingga F yang memuat q elemen disebut *Galois field* (lapangan Galois) yang dinotasikan dengan $GF(q)$.

Definisi 2.9 (Vanstone & Oorschot, 1989 : 28).

Jika F suatu lapangan berhingga dengan q elemen, dan $q=p^n$ dengan p bilangan prima dan n bilangan asli, maka F dilambangkan dengan $GF(q)$.

Perhatikan bahwa q mempunyai bentuk p^n , yaitu q merupakan suatu bilangan prima p atau hasil pemangkatan dari p. Notasi GF (p^n) adalah suatu lapangan dengan karakteristik p. Lapangan Z_p dapat dinotasikan dengan $GF(p)$.

Definisi 2.10.

Untuk suatu polinomial $f(x)\in F(x)$, kelas ekuivalensi yang memuat $g(x)\in F(x)$ adalah

$$[g(x)]=\{h(x)\in F(x): h(x)\equiv g(x)(mod f(x))\},$$

yaitu himpunan semua polinomial yang apabila dibagi dengan $f(x)$ menghasilkan sisa yang sama dengan $g(x)$.

Operasi penjumlahan dan perkalian dalam kelas-kelas ekuivalensi tersebut didefinisikan sebagai berikut. Untuk $g(x), h(x)\in Z_p[x]$,

$$[g(x)]+[f(x)]=[g(x)+f(x)] \text{ dan } [g(x)].[f(x)]=[g(x).f(x)].$$

Diberikan $Z_p[x]/f(x)$ yaitu himpunan semua kelas-kelas ekuivalensi dalam $Z_p[x]$ yang kongruen modulo $f(x)$, dengan $f(x)$ adalah polinomial tak tereduksi.

Contoh 2.9.

Mengikuti contoh 2.6, $Z_2(x)/\langle x^2+x+1\rangle$ adalah suatu lapangan yang $GF(2^2)$ dengan koefisien elemen-elemennya atas $Z_2 = \{[0],[1]\}$ dan memiliki 4 elemen. Elemen-elemen dari $GF(2^2) = Z_2(x)/\langle x^2+x+1\rangle = \{[0],[1],[x],[1+x]\}$.

Definisi 2.11 (Vlcek, 2004).

Polinomial tak tereduksi $f(x)$ dengan derajat m atas $GF(q)$ dikatakan primitif jika n adalah bilangan bulat positif terkecil, untuk $f(x)$ faktor dari $x^n - 1$, berlaku $n = q^m - 1$.

Berikut diberikan contoh lapangan Galois yang dikonstruksi dari polinomial primitif.

Contoh 2.10.

Pada $GF(2^3)$, polinomial primitif berderajat 3 yang membangun semua elemen lapangan adalah $P(x) = x^3 + x + 1$. Polinomial primitif $P(x)$ atas $GF(2)$ dengan koefisien variabel adalah elemen $GF(2) = \{0,1\}$.

Diberikan a adalah akar primitif atas $GF(2^3)$ maka $a^3 + a + 1 = 0$ dan

$$a^3 = a^3 + 0 = a^3 + (a^3 + a + 1) = a + 1$$

dihasilkan elemen dari $GF(2^3)$ seperti pada Tabel 2.3.

Tabel 2.3. Elemen-Elemen $GF(2^3)$

Bentuk Pangkat	Bentuk Polinomial	Bentuk Biner
0	0	000
a^0	1	001

a^1	a	010
a^2	a^2	100
a^3	$a+1$	011
a^4	a^2+a	110
a^5	a^2+a+1	111
a^6	a^2+1	101
$a^{7\ mod\ 7}=a^0$	1	001

Berdasarkan Tabel 2.3 dapat dinyatakan elemen GF(2^3) = {000,001,010,011,100,101,110,111}.

B. Kode Linear

1. Ruang Vektor atas Lapangan Berhingga

Definisi 2.12 (Ling dan Xing, 2004: 17).

Diberikan F_q suatu lapangan berhingga dengan elemen sebanyak q. Himpunan tak kosong V, dengan operasi penjumlahan vektor dan perkalian skalar terhadap elemen-elemen F_q adalah ruang vektor atas F_q untuk setiap $u, v, w \in V$ dan untuk setiap $\lambda, \mu \in F_q$ berlaku aksioma berikut:

i. $u+v \in V$

ii. $(u+v)+w = u+(v+w)$

iii. ada elemen $0 \in V$ *yang berlaku* $0+v = v+0 = v$ *untuk setiap* $v \in V$

iv. ada elemen $-u \in V$ *yang berlaku* $u+(-u) = (-u)+u = 0$ *untuk setiap* $u \in V$

v. $u+v = v+u$

vi. $\lambda v \in V$

vii. $\lambda(u+v) = \lambda u + \lambda v$

viii. $(\lambda + \mu)u = \lambda u + \mu u$

ix. $(\lambda\mu)u = \lambda(\mu u)$

x. apabila 1 adalah elemen identitas terhadap perkalian dari F_q *maka memenuhi* $1.u = u$

Himpunan semua vektor dengan panjang n atas lapangan F_q ditulis F_q^n.

Contoh 2.11.

Ruang vektor atas F_2^4 yaitu $\{0000,1110,1101,0011\}$.

C. *Codeword*

Definisi 2.13 (Ling & Xing, 2004 : 5).

Diberikan $X = \{x_1, x_2, \dots, x_q\}$ *adalah suatu himpunan yang berukuran* q*, yang dapat disebut alfabet kode dan elemen-elemennya disebut simbol kode.*

i. Suatu word $\boldsymbol{q}-$er *panjang* n *yaitu barisan* $w = w_1 w_2 \dots w_n$ *dengan* $w_i \in X$*untuk setiap i.*

ii. Kode blok $\boldsymbol{q}-$er *dengan panjang* n *atas* X *merupakan himpunan tak kosong C pada word* $q-er$ *mempunyai panjang* n *yang sama.*

iii. Elemen dari C disebut dengan codeword.

iv. Kode dengan panjang n *dan berukuran* M *disebut dengan kode-*(n, M).

Berikut diberikan contoh kode.

Contoh 2.12.

Suatu himpunan yang beranggotakan warna yaitu B={merah, kuning, hijau, biru, ungu} akan dikodekan menjadi suatu pesan rahasia yang terdiri dari angka biner 0 dan 1 dengan panjang 3 yaitu :

Tabel 2.4. Pesan Warna dalam Bentuk Kode

Warna	Kode
Merah	001
Kuning	010
Hijau	011
Biru	100
Ungu	101

Jadi himpunan B dapat ditulis dalam bentuk kode yaitu B={001, 010, 011, 100,101}.

Contoh 2.13.

(i) $C_1 = \{00,01,10,11\}$ adalah kode-$(2,4)$ yang artinya kode dengan panjang 2 berukuran 4.

(ii) $C_2 = \{000,011,101,110\}$ adalah kode-$(3,4)$ yang artinya kode dengan panjang 3 berukuran 4.

(iii) $C_3 = \{0011,0101,1010,1100,1001,0110\}$ adalah kode-$(4,6)$ yang artinya kode dengan panjang 4 berukuran 6.

Kode atas alfabet kode $F_2 = \{0,1\}$ disebut dengan kode biner. Kode atas alfabet kode $F_3 = \{0,1,2\}$ disebut dengan kode terner. Sedangkan, kode atas alfabet kode $F_4 = \{0,1,2,3\}$ disebut dengan kode quarterner.

D. Kode Blok

Definisi 2.14 (Vanstone dan Oorschot, 1989).

Kode Blok C dengan panjang n berisi M elemen atas lapangan A adalah himpunan $n-$tupel yang berjumlah M dengan masing-masing koordinat dari $n-$tupel yang diambil dari simbol pada lapangan A dan dinotasikan dengan kode $C[n, M]$ atas lapangan A.

Elemen-elemen dari kode $C[n, M]$ disebut dengan *codeword*. Sedangkan elemen-elemen n −tupel (blok dengan panjang *n*) yang tidak berada dalam blok $C[n, M]$ disebut dengan *word*.

Berikut diberikan contoh kode blok.

Contoh 2.14.

C[2,4] = {00,01,10,11} atas Z_2.

C[3,8] = {000,001,010,011,100,101,110,111} atas Z_2.

C[3,27] = {000,001,002,010,020,011,012,021,022,100,200,101,102,201,

202,110,210,220,120,111,112,121,122,211,212,221,222} atas Z_3.

C[5,12] = {00001,00010,00100,01000,00011,00101,00110,01001,01010,01100,

00111,01111} atas Z_2.

A. Jarak Hamming dan Bobot Hamming

Jarak hamming digunakan untuk menghitung banyaknya perbedaan posisi dua *codeword*. Misalkan, jarak hamming antara dua *codeword* x dan y artinya banyaknya posisi yang berbeda antara dua *codeword* x dan y. Jarak hamming dinotasikan $d(x, y)$. Berikut diberikan definisi jarak hamming.

Definisi 2.15 (Ling & Xing, 2004: 9).

Diberikan x dan y adalah word dengan panjang n atas lapangan A, jarak hamming dari x dan y dinotasikan $d(x, y)$ *dengan x dan y berbeda. Jika* $x = x_1 \dots x_n$ *dan* $y = y_1 \dots y_n$*, maka*

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

Berikut diberikan contoh menghitung jarak hamming.

Contoh 2.15.

Diberikan *codeword* x, y, z sebagai berikut:

$x = 2101001 \qquad y = 2200001 \qquad z = 2110010$

Jarak hamming $d(x, y) = 2$, $d(x, z) = 4$, $d(y, z) = 4$.

Definisi 2.116 (Vanstone dan Oorschot, 1989 : 7).

Diberikan C suatu kode $-[n, M]$ *maka jarak hamming d dari kode C adalah sebagai berikut:*

$$d = min\,\{d(x, y):\ x, y \in C\ , x \neq y\}.$$

Artinya, jarak hamming dari suatu kode adalah jarak minimum antara dua *codeword* yang berbeda, atas semua pasang *codeword*.

Berikut diberikan contoh menghitung jarak hamming suatu kode blok.

Contoh 2.16.

Diberikan kode [7,4] $C = \{c_0, c_1, c_2, c_3\}$:

$c_0 = (0000000)$ $\qquad$ $c_2 = (1001101)$

$c_1 = (0010010)$ $\qquad$ $c_3 = (1011111)$

dihasilkan jarak hamming dari kode C sebagai berikut:

$d(c_0, c_1) = 2 \qquad d(c_0, c_3) = 6 \qquad d(c_1, c_3) = 4$

$d(c_0, c_2) = 4 \qquad d(c_1, c_2) = 6 \qquad d(c_2, c_3) = 2$

sehingga disimpulkan bahwa kode C mempunyai jarak $d = 2$.

Definisi 2.17 (Ling & Xing, 2004 : 48).

Jika C suatu kode maka bobot hamming minimum dari C dilambangkan dengan $w(C)$ *yang berarti bobot terkecil dari codeword yang bernilai tak nol dari C.*

Berikut diberikan contoh menghitung bobot hamming.

Contoh 2.17.

Diketahui *codeword* $a = 1110010$, $b = 12201001$, dan $c = 21022130$ maka bobot hamming untuk *codeword* a, b, c adalah :

$w(1110010) = 4$

$w(12201001) = 5$

$w(21022130) = 6.$

Definisi 2.18 (Ling & Xing, 2004 : 46).

Suatu kode C (C tidak harus linear), bobot hamming minimum dari C dinotasikan dengan w(C) yang merupakan bobot terkecil dari codeword tak nol dari C, didefinisikan

$$w(C) = \min_{c \in C, c \neq 0} \{w(c)\}.$$

Berikut diberikan contoh bobot hamming suatu kode blok.

Contoh 2.18.

Diberikan kode $C[7,4] = \{c_0, c_1, c_2, c_3\}$:

$c_0 = (0000000)$ $\quad$ $c_2 = (1001101)$

$c_1 = (0010010)$ $\quad$ $c_3 = (1011111)$

dihasilkan bobot hamming dari kode C sebagai berikut:

$w(c_1) = 2$ $\quad$ $w(c_2) = 4$ $\quad$ $w(c_3) = 6$

sehingga disimpulkan bahwa bobot hamming kode C adalah $w(C) = 2$.

B. Pengertian Kode Linear

Pengkodean yang baik adalah proses *encoding* dan decoding apabila timbul kesalahan dapat dideteksi dan dapat diperbaiki. Alfabet pada kode dalam kode

linear merupakan elemen lapangan berhingga F_q. Kode yang terbentuk oleh ruang vektor atas lapangan berhingga disebut kode linear.

Berikut diberikan definisi kode linear.

Definisi 2.19 (Ling & Xing, 2004 : 45).

Kode linear C dengan panjang n atas F_q *merupakan subruang vektor atas* F_q^n *dengan* F_q^n *adalah himpunan semua vektor dengan panjang n yang entri-entrinya adalah elemen* F_q. *Bentuk* F_q^n *adalah sebagai berikut :* $F_q^n = \{(v_1, v_2, \ldots, v_n) : v_i \in F_q\}$.

Kode $C(n, k)$ adalah kode linear C dengan panjang n dan berdimensi k atas lapangan F_q (Ling & Xing, 2004: 46).

Berikut diberikan contoh kode linear.

Contoh 2.19.

S_1={0000, 1000, 0100, 1100}

S_2={0000, 1100, 0011, 1111}

$S_3 = \{000000, 012010, 011201, 020211\}$.

S_1 dan S_2 merupakan kode linear atas F_2^4, sedangkan S_3 merupakan kode linear atas F_3^6.

C. Kode Siklik

Definisi 2.20 (Ling & Xing, 2004:133).

Himpunan S atas F_q^n *adalah kode siklik apabila terdapat* $(a_0, a_1, \ldots, a_{n-1}) \in S$ *maka juga terdapat* $(a_{n-1}, a_0, a_1, \ldots, a_{n-2}) \in S$.

Berikut diberikan contoh kode siklik.

Contoh 2.20.

(i) Kode C-(3,2) yaitu kode C = {000,101,110,011} merupakan kode siklik karena perputaran dari setiap *codeword* pada C merupakan *codeword* pada C juga.

(ii) Diberikan kode C-(7,4) yaitu $C = \{1101000, 0110100, 0011010, 0001101\}$ merupakan kode siklik karena perputaran dari setiap *codeword* pada C merupakan *codeword* pada C juga.

(iii) Diberikan kode $S = \{0112, 2011, 1201, 1120\}$ merupakan kode siklik karena perputaran dari setiap *codeword* pada S merupakan *codeword* pada S juga.

D. Polinomial Generator

Polinomial generator merupakan kode siklik C $\neq \{0\}$ dengan panjang n atas F_q, yang terdapat polinomial monik unik $g(x) \in C$ berderajat minimal (Stichtenoth, 2009 : 317). Polinomial monik adalah polinomial $a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$ dengan koefisien tak nol pada pangkat tertinggi dari x yaitu $a_n = 1$. Polinomial monik $g(x)$ membagi $x^n - 1$ dan membangun C sehingga polinomial generator juga membagi $x^n - 1$. Bentuk umum dari polinomial generator $g(x)$ adalah sebagai berikut.

$$g(x) = \prod_{i=1}^{2t} (x - \beta^i)$$

dengan β adalah unsur primitif dalam $GF(q)$ dan t adalah banyaknya t-*error* yang dapat dikoreksi.

Berikut diberikan contoh polinomial generator.

Contoh 2.21 (Vanstone & Oorschot, 1989:29).

Polinomial generator atas GF(9) yang terbentuk atas fungsi ireduksibel $f(x) = x^2 + 1$ dari $Z_3[x]$. Untuk mencari elemen primitif perlu dicoba dan jika dimisalkan $\alpha = x$ ternyata bukan elemen primitif. Tetapi, jika diambil $\alpha = 1 + x$ maka hasilnya sebagai berikut:

$(1+x)^0 = 1$ $\qquad$ $(1+x)^4 = 2$

$(1+x)^1 = 1 + x$ $\qquad$ $(1+x)^5 = 2 + 2x$

$(1+x)^2 = 2x$ $\qquad$ $(1+x)^6 = x$

$(1+x)^3 = 1 + 2x$ $\qquad$ $(1+x)^7 = 2 + x$

Jadi $\alpha = 1 + x$ merupakan polinomial generator dari GF(9)*. Notasi GF(9)* menerangkan polinomial generator $\alpha = 1 + x$ yang tidak menghasilkan polinomial 0.

E. Matriks Generator dan Matriks *Parity*-Cek

Matriks generator digunakan untuk membentuk suatu kode linear. Setelah matriks generator terbentuk, kode akan dikirimkan yang nantinya akan diterima oleh pengguna. Apabila kode yang diterima tidak sesuai dengan yang dikirimkan maka perlu penambahan redudansi ke dalam informasi yang mengalami *error* agar dapat mendeteksi maupun mengoreksi informasi yang salah seperti aslinya. Redundansi yang dimaksud adalah bit-bit untuk mengecek terjadinya *error* yang dikenal dengan nama matriks *parity*-cek.

Definisi 2.21 (Ling & Xing, 2004 : 52).

i. Matriks generator untuk kode linear C adalah matriks G dimana baris-baris membentuk basis untuk C. Bentuk standar matriks generator adalah $(I_k|X)$.

*ii. Matriks parity-*cek *H dari kode linear C adalah matriks generator untuk kode dual* $C^{\perp}$*. Bentuk standar matriks parity-cek adalah* $(-X^T|I_{n-k})$*.*

Jika C merupakan kode linear-(n, k) maka matriks generator untuk C adalah matriks berukuran $k \times n$ dan matrisk *parity*-cek untuk C berukuran $(n-k) \times n$.

Berikut diberikan contoh bentuk matriks generator dan matriks *parity*-cek.

Contoh 2.22.

Diberikan matriks $\tilde{G}$ yang membangun kode-$(6,4)$ atas $\mathrm{GF}(3)$, yaitu

$$\tilde{G} = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 1 & 1 & 1 \\ 0 & 1 & 0 & 2 & 1 & 0 \end{bmatrix}.$$

Operasi baris elementer yang sesuai terhadap $\tilde{G}$ untuk menghasilkan matriks G yang membangun kode yang sama sebagai berikut:

$$\begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 1 & 1 & 1 \\ 0 & 1 & 0 & 2 & 1 & 0 \end{bmatrix} \begin{matrix} \\ \\ R_3 - R_1 \\ R_4 - R_2 \end{matrix} \quad \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix} R_3 - R_2$$

$$\begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 2 & 1 & 2 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix} \begin{matrix} R_1 - 2R_2 \\ R_2 - R_4 \\ R_3 - 2R_4 \\ \\ \end{matrix} \quad \begin{bmatrix} 1 & 0 & 1 & 2 & 0 & 2 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix} R_1 - R_3$$

$$\begin{bmatrix} 1 & 0 & 0 & 2 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix} R_1 - 2R_4 \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix}$$

sehingga dihasilkan matriks G adalah

$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix}$ yang mempunya bentuk $G = [I_4 A]$, sehingga diperoleh matriks $A = \begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 1 & 1 \\ 0 & 2 \end{bmatrix}$.

Matriks generator untuk kode dual $C^{\perp}$ atau disebut dengan matriks *parity*-cek H berbentuk $H = [-A^T I_{n-k}]$, yaitu

$$A^T = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & 2 & 1 & 2 \end{bmatrix} \qquad -A^T = \begin{bmatrix} -2 & -1 & -1 & -0 \\ -0 & -2 & -1 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix}$$

$$H = [-A^T I_{n-k}] = [-A^T I_2] \begin{bmatrix} 1 & 2 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 1 \end{bmatrix}.$$

Contoh 2.23.

Diketahui kode C-(6,3) biner atas Z_2 dengan matriks generator H untuk C sebagai berikut:

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

Matriks H di atas berbentuk $H = [A I_3]$ sehingga didapatkan matriks

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

Matriks generator G untuk kode $C^{\perp}$ berbentuk $G = [I_3 \,/\, -A^T]$ sehingga dihasilkan matriks generator G yaitu

$$A^T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \qquad -A^T = \begin{bmatrix} -1 & -0 & -1 \\ -1 & -1 & -0 \\ -0 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$G = [I_3 \,/\, -A^T] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

F. Syndrome

Definisi 2.22 (Ling S. & Xing C, 2004 : 59).

Diberikan suatu kode linear C dengan panjang n atas F_q dan $u = (u_1, u_2, \dots, u_n) \in F_q^n$. Coset dari C yang ditentukan oleh u adalah himpunan

$$C + u = \{v + u : v \in C\} = \{u + v : v \in C\} = u + C.$$

Definisi 2.23 (Ling S. & Xing C, 2004 : 60).

Suatu word dengan bobot hamming terkecil dalam suatu koset disebut coset leader.

Contoh 2.24.

Diberikan suatu kode - (5,3) C dengan matriks generator dan matriks *parity*-cek yaitu

$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$ serta kode C yaitu

$C = \{00000, 10011, 01001, 00110, 11010, 10101, 01111, 11100\}$.

Berikut diberikan tabel standar *array* untuk kode-*(5,3) C.*

Tabel 2.5 Standar *Array* kode-*(5,3) C*

Coset Leader								
00000	00000	10011	01001	00110	11010	10101	01111	11100
00001	00001	10010	01000	00111	11011	10100	01110	11101
00010	00010	10001	01011	00100	11000	10111	01101	11110
10000	10000	00011	11001	10110	01010	00101	11111	01100

Berdasarkan Definisi 2.21, *coset leader* dipilih dari *word* yang berbobot terkecil yaitu 1. vektor-vektor kolom pertama pada Tabel 2.5 merupakan *coset*

leader dari koset-koset pada kolom selanjutnya. Selain vektor 00010, koset 00010+C juga memiliki *coset leader* lain yaitu 00100. Selain itu, vektor 00001 pada koset 00001+C juga memiliki *coset leader* lain yaitu 01000.

Berikut diberikan definisi yang berkaitan dengan *syndrome*.

Definisi 2.24 (Ling & Xing, 2004 : 62).

Diberikan C adalah $[n, k, d]$ *−kode linear atas* F_q *dan diberikan H adalah matriks parity*-cek *untuk C. Untuk setiap* $u \in F_q^n$*, maka syndrome oleh u adalah word* $S(u) = uH^T \in F_q^{n-k}$.

Berikut diberikan contoh menghitung *syndrome*.

Contoh 2.25.

Diketahui suatu matriks generator dan matriks *parity*-cek serta kode C seperti pada Contoh 2.24. Berdasarkan Tabel 2.5 diketahui *coset leader* u yang digunakan untuk menghitung *syndrome* dengan rumus $S(u) = uH^T$ sebagai berikut.

$$S(00000) = [00000]\begin{bmatrix}1 & 1\\ 0 & 1\\ 1 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix} = [00] \qquad S(00010) = [00010]\begin{bmatrix}1 & 1\\ 0 & 1\\ 1 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix} = [10]$$

$$S(00001) = [00001]\begin{bmatrix}1 & 1\\ 0 & 1\\ 1 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix} = [01] \qquad S(10000) = [10000]\begin{bmatrix}1 & 1\\ 0 & 1\\ 1 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix} = [11]$$

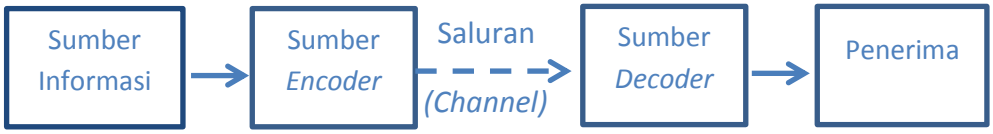
Hasil perhitungan pada Contoh 2.24 dan Contoh 2.25 dinyatakan pada Tabel 2.6 berikut.

Tabel 2.6 *Coset Leader* dan *Syndrome* kode-(5,3) C

Coset Leader	*Syndrome*
00000	00
00001	01
00010	10
10000	11

G. *Encoding* dan *Decoding* (Vanstone & Oorschot, 1989:1)

Teori kode pengoreksi *error* merupakan salah satu cabang matematika yang bergerak dibidang transmisi dan penyimpanan data. Media informasi tidak selalu memberikan keakuratan dalam menerima informasi, adakalanya terjadi suatu gangguan saat pengiriman pesan/informasi. Apabila terjadi suatu *error* pada saat pengiriman pesan/informasi, kesalahan tetap dapat terdeteksi bahkan diperbaiki dengan menambahkan suatu redundansi ke dalam pesan/informasi yang telah diubah dalam bentuk kode.



Gambar 2.1. Diagram Proses Pengiriman Pesan/Informasi

1. Proses *Encoding*

Suatu pesan/informasi diubah ke dalam bentuk kode untuk memudahkan proses pengiriman data (pesan/informasi). Proses mengubah pesan/informasi ke dalam bentuk kode disebut dengan proses *encoding*. Misalkan suatu himpunan

simbol {A,B,C,D} akan ditransmisikan ke dalam bentuk kode biner dengan panjang 2, yaitu:

A = 00

B = 10

C = 01

D = 11

Apabila pengirim mengirimkan sebuah kode 00, maka penerima akan membaca kode 00 sebagai A.

Contoh 2.26.

Diberikan matriks generator $G = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$ dan pesan yang akan dikirim adalah $u = \{00,10,01,11\}$, maka u dikodekan menjadi

$$v_1 = u_1 G = 00 \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = 0000$$

$$v_2 = u_2 G = 10 \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = 1110$$

$$v_3 = u_3 G = 01 \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = 0101$$

$$v_4 = u_4 G = 11 \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = 1011.$$

Jadi pesan yang dikirimkan adalah $v = \{0000,1110,0101,1011\}$.

Sumber informasi mengirimkan simbol dari himpunan informasi {A,B,C,D}. Kemudian, sumber *encoder* memasangkan atau memetakan setiap informasi dengan urutan biner (0,1) dan ditransmisikan. Selanjutnya sumber *decoder* menerima urutan biner dari *channel* atau saluran, lalu dikonversi kembali ke huruf alfabet untuk dapat diterima oleh pengguna (penerima).

2. Proses *Decoding*

Proses mengembalikan kode menjadi suatu pesan/informasi seperti yang dikirimikan disebut dengan proses *decoding*. Pada saat proses *decoding* akan terjadi suatu proses deteksi *error* dan pengoreksian *error* jika terjadi *error*. Artinya, jika sumber *decoder* membaca informasi seperti yang dikirimkan oleh sumber *encoder* maka tidak perlu ada proses pengoreksian *error*. Sebaliknya, jika sumber *decoder* membaca pesan atau informasi yang salah maka perlu dilakukan suatu proses pengoreksian *error*.

Pada Contoh 2.26, jika sumber *encoder* mengirimkan 00 dan sumber *decoder* menerima 01, maka telah terjadi satu kesalahan. Sumber *decoder* tidak bisa mengetahui terjadinya *error* karena pesan 01 juga merupakan informasi valid dari C

Jika penerima membaca 1100 akan diketahui bahwa telah terjadi kesalahan karena urutan ini bukan salah satu input dari sumber encoder. Apabila kesalahan acak maka *decoder* akan mentransmisikan pesan ke 1110 karena urutan kode yang diterima hanya terjadi satu *error*.

Definisi 2.25.

Suatu kode C dapat mendeteksi sebanyak u *kesalahan* $(u \geq 1)$ *jika untuk setiap codeword* $c_1 \in C$ *yang dikirim kemudian diterima codeword* c_2 *dengan* $d(c_1, c_2) = u$ *sehingga* $c_2 \neq C$.

Selanjutnya diberikan teorema mengenai kemampuan deteksi kesalahan kode.

Teorema 2.5.

Suatu kode C dikatakan mendeteksi u kesalahan jika dan hanya jika $d(C) \geq (u + 1)$. *Dengan kata lain, kode dengan jarak d dapat mendeteksi dengan tepat (d-1) kesalahan.*

Bukti:

($\Rightarrow$) Diketahui $d(C) \leq u$, maka terdapat $c_1, c_2 \in C$ sedemikian sehingga $1 \leq d(c_1, c_2) = d(C) \leq u$. Jika *codeword* c_1 yang dikirim dan diterima *codeword* c_2 maka terjadi kesalahan sebanyak $d(C)$ dengan $1 \leq d(C) \leq u$.Akan tetapi kesalahan tersebut tidak terdeteksi karena $c_2 \in C$. Jadi, C bukan mendeteksi u kesalahan.

($\Leftarrow$) Diketahui $d(C) \geq u + 1$. Menurut definisi jarak dari suatu kode, hal ini berakibat jika $c \in C$ dan x sedemikian sehingga $1 \leq d(c, x) \leq u < d(C)$, maka $x \notin C$. Artinya jika terjadi kesalahan hingga sebanyak u, maka kesalahan tersebut selalu terdeteksi. ■

Setelah kode yang mengalami kesalahan terdeteksi maka langkah selanjutnya adalah pengoreksian. Ada beberapa macam kode pengoreksi *error*. Diantaranya yaitu kode *Reed Solomon* yang merupakan subkelas dari kode BCH.

3. Kode BCH

Kode BCH adalah suatu kelas kode yang ditemukan oleh R.C. Bose dan D. Ray Chaudhuri pada tahun 1960 dan juga ditemukan secara independen oleh A. Hocquenghem pada tahun 1959. Nama BCH diambil dari penemu-penemu tersebut Bose Chaudhuri Hocquenghem.

Definisi 2.26 (Vanstone & Oorschot, 1989 : 205).

Kode BCH atas $F = GF(q)$ *dengan panjang blok* n *dan jarak yang ditentukan (designed distance)* δ *adalah suatu kode yang dibentuk oleh suatu polinomial* $g(x) = KPK\{f_i(x) : a \leq i \leq a + \delta - 2\} \in F[x]$ *dengan KPK adalah kelipatan persekutuan terkecil. Himpunan akar dari polinomial* $g(x)$ *memuat* $d - 1$ *elemen yang berbeda* $a^a, a^{a+1}, \dots, a^{a+d-2}$*, dengan* a *adalah suatu akar primitif ke-n dari elemen kesatuan dan* a *adalah suatu bilangan bulat.*

Definisi 2.27 (Vanstone & Oorschot, 1989 : 205).

Jika $n = q^m - q$ *untuk bilangan bulat positif untuk bilangan bulat positif m, maka kode tersebut adalah suatu kode primitif, dan jika* $a = 1$ *maka kode tersebut disebut narrow-sense.*

Elemen a adalah suatu akar ke-n dari elemen kesatuan, sehingga a^j juga merupakan suatu akar ke-n dari elemen kesatuan untuk semua j sehingga $(x - a^j)$ membagi $(x^n - 1)$ dan $g(x)$ juga membagi $(x^n - 1)$.

Contoh 2.27.

Misal a suatu elemen primitif dari $GF(2^4)$ sedemikian sehingga $1 + a + a^4 = 0$. Polinomial minimal dari $a, a^3,$ dan a^5 adalah berturut-turut

$\phi_1(x) = 1 + x + x^4$,

$\phi_3(x) = 1 + x + x^2 + x^3 + x^4$,

$\phi_5(x) = 1 + x + x^2$,

Kode BCH yang mengoreksi dua *error* dengan panjang $n = 2^4 - 1 = 15$ dibentuk oleh $(x) = KPK\{\phi_1(x), \phi_3(x)\} = (1 + x + x^4)(1 + x + x^2 + x^3 + x^4) = 1 + x^4 + x^6 + x^7 + x^8$. Didapatkan $n - k = 8$ sehingga kode tersebut

adalah suatu kode $(15,7,\geq 5)$. Bobot dari polinomial generatornya adalah 5, sehingga kode tersebut adalah suatu kode $(15,7,5)$.

H. Pengantar Kode *Reed Solomon* pada Aplikasi *Steganography*

1. Sejarah Kode *Reed Solomon*

Gustave Solomon lahir di Brooklyn, New York pada tanggal 27 Oktober 1930, dan meninggal pada tanggal 31 Januari 1996 di Beverly Hills, CA. Dia adalah seorang ahli matematika dan insinyur yang merupakan salah satu penemu dari teori aljabar tentang *error-correction*.

Solomon bersama-sama dengan Irving S. Reed dikenal telah mengembangkan sebuah teori aljabar tentang *error-detecting* dan *error-correcting codes* yang dikenal sebagai *Reed Solomon error correction*. Kode ini digunakan untuk melindungi kerahasiaan informasi digital, dan telah digunakan secara luas di bidang komunikasi dan media penyimpanan digital modern.

Kasus yang paling umum misalnya digunakan kode RS-(255, 223) dimana pesan 223 simbol (masing-masing delapan bit) di *encode* menjadi 255 simbol. Standar RS-(255, 223) kode *Reed Solomon* mampu memperbaiki hingga 16 simbol kesalahan dalam setiap *codeword*. Kode *Reed Solomon* ditemukan oleh Irving S. Reed dan Gustave Solomon pada tahun 1960. Mereka membawakan jurnal yang berjudul "*Polynomial Codes over Certain Finite Fields*" dalam seminarnya kala itu.

Pada saat artikel tersebut ditulis, teknologi digital saat itu tidak cukup maju untuk menerapkan konsep tersebut. Aplikasi dari kode tersebut baru

bisa digunakan pertama kali pada tahun 1982, yang diproduksi secara masal yaitu berupa cakram padat / *compact disc*, di mana dua kode *interleaver Reed-Solomon* digunakan. Algoritma *decoding* yang efisien untuk kode *Reed Solomon* dengan jarak yang besar telah dikembangkan Elwyn Berlekamp dan James Massey pada 1969.

Sekarang ini kode *Reed Solomon* telah dimanfaatkan untuk banyak aplikasi antara lain aplikasi komputer dan media panyimpanan seperti sistem RAID (*Redundant Array of Independent Disk*s) pada *hard disk drive*, CD *(compact disk)*, DVD *(Digital Versatile Disk)*, dan *blue-ray disk*, telekomunikasi dan data teknologi seperti DSL (*Digital Subscribe Line*) & WiMAX (*Worldwide Interoperability for Microwave Acces*), dalam siaran televisi digital seperti sistem ATSC (*Advanced Television Systems Committee*) di Amerika, dan sistem DVB (*Digital Video Broadcasting*) di Eropa. Selain itu, aplikasi kode *Reed Solomon* untuk sarana bertukar informasi/pesan yang mengutamakan tingkat keamanan adalah *steganography*.

2. Sejarah *Steganography*

Steganography merupakan suatu cara penyembunyian pesan ke dalam pean lainnya sedemikian rupa sehingga orang lain tidak menyadari adanya perubahan di dalam pesan yang terkirim. Kata *steganography* (*steganography*) berasal dari bahasa Yunani yaitu steganos yang artinya tersembunyi atau terselubung dan graphein yang artinya menulis sehingga *steganography* berarti “menulis tulisan yang tersembunyi atau terselubung”

(Sellars, 1996). Teknik ini biasa digunakan untuk menyembunyikan pesan rahasia pada media komunikasi.

Catatan pertama tentang *steganography* ditulis oleh sejarawan Yunani Herodotus yaitu ketika Histaeus seorang raja kejam Yunani dipenjarakan oleh Raja Darius di Susa pada abad 5 Sebelum Masehi. Histaeus harus mengirim pesan rahasia kepada anak laki-lakinya Aristagoras di Militus. Histaeus menulis pesan dengan cara mentato pesan pada kulit kepala seorang budak dan ketika rambut budak itu mulai tumbuh, Histaeus mengutus budak itu pergi ke Militus untuk mengirim pesan di kulit kepalanya tersebut kepada Aristagoras.

Teknik *steganography* yang lain adalah tinta yang tak terlihat. Teknik ini pertama kali digunakan pada zaman Romawi kuno dengan menggunakan air sari buah jeruk, urine, atau susu sebagai tinta untuk menulis pean. Cara membacanya adalah dengan dipanaskan di atas nyala lilin, tinta yang sebelumnya tidak terlihat setelah terkena panas akan berangsur-angsur menjadi gelap dan pesan dapat dibaca.

Dari contoh *steganography* konvensional tersebut pada intinya adalah teknik *steganography* konvensional berusaha merahasiakan pesan komunikasi dengan cara menyembunyikan pesan atau mengkamuflase pesan. Maka, prinsip dasar dalam *steganography* dikonsentrasikan kerahasiaan komunikasinya bukan pada datanya (Johnson, 1998). Pada abad 20, *steganography* telah mengalami perkembangan. *Steganography* telah merambah juga ke media digital.

Sebagai contohnya, misalkan akan mengirimkan pesan tersembunyi yang berisi MERAH. Kalimat yang akan dikirimkan adalah Memang Enak Rasa Asam Harumanis.

cover text/*cover media* : emang nak asa sam arumanis

pesan tersembunyi : MERAH

stego text (*stegogramme*): Memang Enak Rasa Asam Harumanis.