

# Penggunaan Logika Even Parity pada Beberapa Error Correction Code Terutama pada Hamming Code

**Kevin Tirtawinata – NIM 13507097**  
*Program Studi Teknik Informatika*  
*Sekolah Teknik Elektro dan Informatika*  
*Institute Teknologi Bandung*  
*Jl. Ganesha 10, Bandung*  
Email : [cybernetic\\_4ever@hotmail.com](mailto:cybernetic_4ever@hotmail.com)

**Abstrak** – Makalah ini merupakan makalah yang menjelaskan hal-hal yang berkaitan dengan Hamming Code terutama pada bagian Perbaikan Kesalahan. Hamming Code merupakan kode yang mengandung beberapa kode yang dapat mengidentifikasi kesalahan pada keseluruhan kode hingga 2 digit kesalahan. Beberapa perangkat elektronik yang memerlukan ketepatan pada kode-kode yang bersangkutan menggunakan linear error-correcting code yang salah satunya adalah Hamming Code.

**Kata Kunci** – Hamming Code, Perbaikan Kesalahan, kode biner, Even Parity, Parity Bit

## 1. PENDAHULUAN

Pada berbagai alat telekomunikasi dan informasi terdapat berbagai kesalahan pada transmisi data. Untuk menghindari hal tersebut, berbagai perangkat yang digunakan dalam pengiriman data memiliki sistem yang dapat memberikan data tambahan yang mampu digunakan untuk mengoreksi data yang dikirimkan. Sistem koreksi tersebut dikenal luas oleh berbagai macam orang dengan sebutan error correcting code atau juga forward error correction. Keuntungan dari adanya error correcting code adalah tidak diperlukannya pengiriman kembali data dalam suatu pengiriman. Error correcting code juga digunakan dalam pengamanan data dalam jumlah besar untuk menghindari kerusakan pada data yang bersangkutan.

Hamming code merupakan salah satu jenis linear error correcting code yang sederhana, Hamming code banyak digunakan pada berbagai peralatan elektronik.

Hamming Code diperkenalkan oleh Richard W. Hamming pada tahun 1950. Pada tahun 1940-an Hamming bekerja sebagai pengamat pada perusahaan Bell Telephone Laboratories. Ia bertugas di bagian Laboratorium dan menangani mesin Bell Model V. Bell Model V merupakan alat pengirim pesan menggunakan gelombang elektromagnetik. Akibat banyaknya gangguan yang ada, banyak kesalahan yang harus diperbaikinya. Hal itu membuatnya harus memperbaiki banyak kesalahan. Dan pada waktu itu pula ia menemukan Hamming Code untuk

mempermudah memperbaiki kode-kode yang ia tangani

## 2. PEMBAHASAN

Hamming code merupakan sistem yang dikembangkan dari error correction code yang menggunakan parity bit, selain Hamming Code banyak juga sistem lain yang lebih efisien dalam error correction code pada data yang terdiri dari banyak bit. Karena pengecekan secara parity ini juga maka kita dapat mengecek kode-kode yang ada.

Linear error-correcting code memiliki berbagai keterbatasan kesalahan. Pada Hamming Code, kesalahan yang dapat diketahui hanya 1 (satu) buah sedangkan yang dapat dideteksi adalah 2 (dua) buah.

### 2.1 Parity Bit

Parity Bit adalah sistem pengecekan data yang pertama kali ada. Parity bit code menunjukkan ganjil atau genapnya jumlah bit yang bernilai 1. Parity bit dibagi menjadi 2 jenis, yaitu even dan odd. Even parity akan menunjukkan nilai 1 bila jumlah bit 1 genap, sebaliknya odd parity akan memberikan nilai 1 jika jumlah bit satu ganjil.

Parity bit code ini bebas diletakkan dimanapun sesuai kesepakatan penerima dan juga pengirim.

7 bits of data	Byte with parity bit code	
	even	odd
0101010	00101010	10101010
1001000	11001000	01001000
1010010	01010010	11010010
0101100	00101100	10101100
0000000	10000000	00000000
1111111	01111111	11111111

Tabel 1. Contoh byte with parity bit code pada awal byte

Parity bit memiliki kelemahan yang tidak dapat mendeteksi bila terjadi dua buah kesalahan sekaligus. Contohnya bila 1111111 terkirim sebagai 11011110 pada odd parity code yang seharusnya adalah 11111111.

Hingga saat ini parity code masih digunakan pada perangkat SCSI dan PCI buses pada komputer. Penggunaan dari parity code ini dikarenakan minimnya kesalahan yang dilakukan pada SCSI dan PCI buses saat mengcopy data dari memory.

Error correction code yang hanya dapat mengecek sebuah kesalahan akan menjadi masalah untuk data yang panjang, sehingga penggunaan error correction code biasanya digunakan dengan cara membagi-bagi data ke dalam ukuran bit yang lebih kecil. Penggunaan parity bit code biasanya digunakan untuk setiap 7 bit memiliki satu parity code. 7 bit juga dapat disebut sebagai jarak kode.

Semakin kecil jarak kode, maka kesalahan pun dapat dikurangi, sebaliknya untuk jumlah data yang sama, jumlah parity bit code menjadi lebih banyak, dan ukuran data yang diterima pun lebih besar.

Pengulangan jarak selalu sama untuk setiap data yang dimasukkan sesuai dengan sistem yang sudah diperjanjikan sebelumnya.

Kebanyakan dari error correcting code bertipe SECDED ( single error correction and double error detecting ). Untuk tiga buah kesalahan atau lebih dapat tidak terdeteksi.

## 2.2 Hamming Code

Hamming Code merupakan error correcting code seperti even parity code, namun kode-kode hamming ini tidak hanya terdapat 1 tiap beberapa kode, melainkan memiliki posisi yang beraturan.

Banyaknya kode hamming yang dibutuhkan bertamhab sesuai dengan banyaknya data dan posisi yang harus diisi dengan data tersebut. Persamaan jumlah parity code yang dibutuhkan dengan banyaknya data ialah :

$$2^n - (1 + n) \geq (data)$$

misal untuk data 4, maka kita membutuhkan 3 buah hamming code dibutuhkan code.

Banyaknya kode agar dapat SECDED minimal harus sesuai dengan persamaan di atas. Karena setiap hamming code mengecek jumlah data yang spesifik dengan posisi yang spesifik sesuai dengan yang dapat dilihat di table. Hal ini menyebabkan jumlah Hamming code yang diperlukan adalah spesifik sesuai dengan besarnya data yang ada.

Banyaknya bit data yang akan dikirimkan	Jumlah Hamming Code yang dibutuhkan	Jumlah data yang dikirimkan
1	2	3
4	3	7
11	4	15
26	5	31
57	6	63

Tabel diatas menunjukkan hubungan banyaknya data yang memenuhi SECDED Hamming Code. Dapat kita perhatikan bahwa untuk mengkoreksi suatu kesalahan dari data yang besar Hamming Code sangat efektif, karena banyaknya data bertambah sebesar 2 pangkat n. Pemosisian Hamming Code pun memiliki aturan tersendiri agar dapat mengecek kesalahan.

Aturan yang digunakan dalam menyusun data dengan menggunakan Hamming Code alaha sebagai berikut :

- Tiap posisi yang merupakan 'power of 2' ( bilangan yang merupakan 2 pangkat n untuk semua n bilangan asli ) dikosongkan untuk diisi dengan Hamming Code. ( posisi 1, 2, 4, 8, 16, 32 dikosongkan)
- Tiap bilangan diurutkan ke dalam posisi yang tidak digunakan untuk mengisi Hamming Code ( posisi yang diisi adalah 3,5,6,7,9,10,dst..)

Contoh : untuk kode 1100 diubah menjadi xx1x100

Posisi	1	2	3	4	5	6	7
bit	x	x	1	x	1	0	0

- Pengisian untuk kode dengan posisi ke n adalah dengan parity bit untuk bilangan yang sesuai dengan aturan :

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11	p5	d12	d13	d14	d15	
Parity bit coverage	p1	X		X		X		X		X		X		X		X		X		X	
	p2		X	X			X	X			X	X			X	X			X	X	
	p3				X	X	X	X					X	X	X	X					X
	p4								X	X	X	X	X	X	X	X					
	p5																X	X	X	X	X

Gambar 1. Tabel Kode untuk membuat Hamming Code

- o Semua bilangan sebelum n-1 tidak diperhatikan
- o Memparity semua bilangan sebanyak n ke depan, dan meloncat bilangan sebanyak n
- o Contoh : untuk mengisi posisi kedua, cek posisi ke 2,3 loncat posisi ke 4, 5 cek posisi ke 6 dan 7. Sehingga untuk contoh di atas  $p2 = 2 \oplus 3 \oplus 6 \oplus 7 \oplus$ , dan p2 adalah 1

Untuk contoh diatas maka table lengkapnya adalah

Posisi	1	2	3	4	5	6	7
bit	0	1	1	1	1	0	0

P1 adalah 1, p2 adalah 0 dan p3 adalah 0, sehingga untuk kode 1100 maka akan dikirimkan bit sebagai 1010100 dengan hamming code.

Data biasa	Hamming Code
1	111
1111	1111111
1110	0010110
1101	1010101

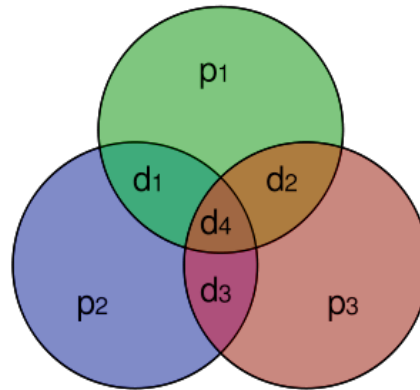
Pengecekan kode dengan hamming seperti itu belum dapat memenuhi SECDED, yang terjadi ialah single error correction or double error detecting yang tidak dapat diketahui yang mana yang menjadi masalah. Yang dimaksud adalah adanya kemungkinan kesalahan 1 atau 2 digit. Untuk menjadikanya SECDED, maka harus ditambahkan 1 digit terakhir yang memeriksa 7 digit lainnya dengan parity bit.

Hamming code yang lengkap adalah Hamming Code dengan tambahan 1 digit parity code di paling belakang data.

Data biasa	Hamming Code	Hamming Code dengan tambahan digit terakhir
1	111	1111
1111	1111111	11111111
1110	0010110	00101101
1101	1010101	10101010

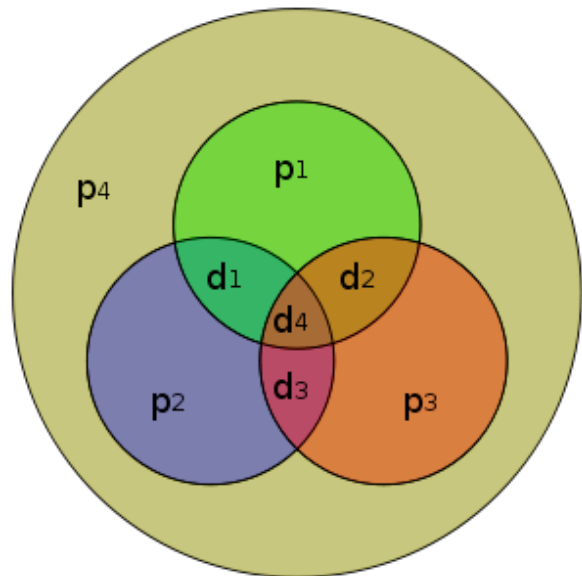
Digit terakhir berfungsi sebagai pembanding, bila pada kesalahan kode, bila digit terakhir salah, maka berarti ada 1 kesalahan kode pada digit yang lain, sebaliknya jika digit terakhir benar dan terjadi kesalahan pada 7 digit yang lain berarti terjadi kesalahan adalah 2 digit pada 7 digit yang lain.

Untuk mempermudah pengerjaan, Hamming Code juga dapat direpresentasikan dengan diagram Venn dengan posisi



Gambar 2. Diagram Venn untuk Hamming ( 7,4 )

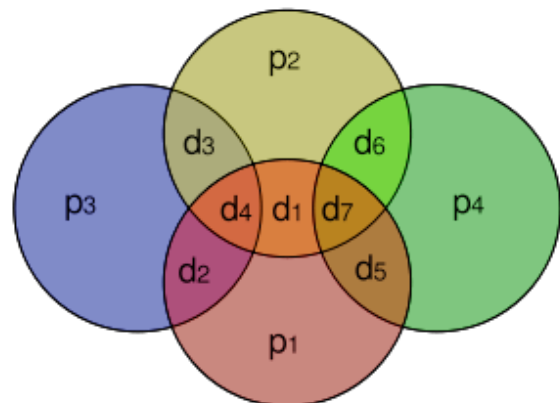
Dan bila Hamming berbentuk Hamming ( 8 ,4 )



Gambar 3. Diagram Venn untuk Hamming ( 8 ,4 )

Pengunaan diagram venn untuk mengecek Hamming Code sangat mudah dilakukan.

Diagram Venn untuk Hamming ( 11 , 7 )



Gambar 4. Untuk Diagram Venn ( 11 , 7 )

Penggunaan diagram Venn ini juga dapat digunakan sebagai penyusun dari digram venn.

Bila ingin membentuk Hamming (12, 7) cukup dengan menambahkan lingkaran terluar seperti Hamming (8,4)

### 2.3 Pengecekan Data yang Dikirim dengan Parity Bit

Data yang dikirim dengan parity code hanya dapat mendeteksi kesalahan sebanyak 1 buah kesalahan tanpa mengetahui letak kesalahan. Pengecekan dilakukan dengan membuat kode yang disampaikan dalam bentuk parity dan dibandingkan, bila hasil dari pembuatan ulang dengan data yang dikirimkan berbeda maka data yang sampai tidak valid. Namun letak dari kesalahan tidak dapat diprediksi, letak kesalahan hanya dapat diantara 2 kode saja.

Dalam aplikasi elektronika, bila data yang dikirim tidak sesuai maka akan dikirim ulang data bagian data yang salah.

Kelemahan utama dari parity code adalah akan memberakan bila 3 dari keseluruhan data yang salah, dan bila kesalahan terjadi 3 kali maka akan terlihat seperti sebuah kesalahan.

Contoh bila data yang dikirimkan adalah 1001000 dengan odd parity code pada awal kode maka seharusnya dikirim mejadi 01001000 dan bila terkirim 01001001 akan memberikan hasil yang salah, dan tidak dapat terdeteksi digit yang salah, namun bila kode yang sampai adalah 01101100 maka akan dibenarkan. Hal ini membuat terkadang parity bit harus digunakan dengna jarka antar kode tida terlalu besar.

### 2.4 Pengecekan Data yang Dikirim dengan Hamming Code

Pada setiap data yang ditambahkan dengan Hamming Code, bila terjadi kesalahan makan akan dapat diperbaiki malihat dari Hamming Code dan juga data-data lain. Kesalahan juga bukan hanya terdapat pada data, pada pengiriman data barupa bit berHamming Code, kesalahan pad Hamming Code juga dimungkinkan, dan hal itu juga dapat diperiksa kebenarannya.

Pengecekan pada setiap bit dari data dapat dilakukan karena setiap data memiliki berbagai hubungan yang berbeda dengan hamming code yang ada.

Position	1	2	3	4	5	6	7
Data	P1	P2	D1	P3	D2	D3	D4
Parity bit coverage	P1	x		x		x	X
	P2		x	x		x	X
	P3				x	x	x

Tabel hubungan antara posisi dengan kode hamming

	3	5	6	7
P1	X	X		X
P2	X	X	X	
P3		X	X	X

Dapat kita lihat bahwa tiap data memiliki relasi yang berbeda-beda, sehingga bila terjadi kesalahan dapat menjadi sebuah kesalahan yang terdeteksi, maupun 2 buah kesalahan yang tidak dapat diketahui kebenarannya.

Pada Hamming Code yang lengkap, digit terakhir dibelakang berguna untuk mengecek apakah yang terjadi merupakan sebuah kesalahan atau dua buah kesalahan sekaligus. Bila terjadi sebuah kesalahan dapat diketahui letaknya dan bila terjadi 2 kesalahan pun dapat diketahui.

Terlebih lagi dengan adanya parity bit pada akhir maka dapat mengetahui kesalahan bila terjadi 3 buah kesalahan sekaligus.

Urutan pengecekan pada data erupa hamming code lengkap lebih baik dmula dari data terakhir karena mempengaruhi secara keseluruhan data.

Pada alat-alat elektronik seperti RAM yang harus memberikan data yang tepat setiap waktunya Hamming Code lengkap digunakan secara efisien.

### 2.5 Penerapan dengan Matriks

Matriks dapat digunakan untuk membuat kode dengan cara melakkan perkalian Matriks. Selain ada matriks untuk membuat kode, terdapat juga matriks untuk mengecek kebenaran dari Hamming Code dan juga decoder untuk Hamming Code.

Hammming (8,4) adalah Hamming dengan tambahan parity bit.

Matriks penyusun untuk Hamming (8,4) ( harus di transpose terlebih dahulu untuk dikalikan dengan 4 digit bilangan biner )

$$G := \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}_{8,4}$$

Matriks penyusun untuk Hamming (7,4)  
( sudah di transpose )

$$G := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Langkah-langkah yang harus dilakukan adalah mengalikan 4 digit bilangan biner dan hasilnya akan di mod dengan 2, dan hasilnya adalah data dengan Hamming Code, dan siap di transmisi.

Matriks Pengecekan Hamming (7,4)

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

perkalian matriks ini dengan data dengan Hamming Code 7 bit akan menghasilkan Null bila Data dengan Hamming Code ini memenuhi standar Hamming Code

Matriks Pengecek un(8,4)

$$H := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}_{4,8}$$

Matriks ini memiliki cara kerja sama seperti matriks untuk menguji Hamming (7,4)

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil

Daftar Lengkap Hamming Code untuk bilangan dengan 4 Digit

Data	Hamming (8,4)
0000	00000000
0100	11100001

1100	10011001
0010	01111000
1010	01010101
0110	10110100
1110	11001100
0001	11010010
1001	00110011
0101	01001011
1101	10101010
0011	10000111
1011	01100110
0111	00011110
1111	11111111

Dengan Hamming (8,4) kita dapat mengetahui letak kesalahan seperti yang akan dibahas pada bagian pembahasan.

#### 3.2 Pembahasan

Contoh kesalahan pada Hamming (7,4)

011110

dengan mengecek digit-digit yang tidak termasuk dalam 'power of 2' kita dapat memperoleh kode yang dimaksud yaitu '1110'

dengan menggunakan perkalian matriks sebagai berikut

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Kode yang didapat adalah '0010110' berbeda kode yang diberikan '011110'

Dengan cara membandingkan dengan kode-kode yang lain kita dapat mengetahui 2 kemungkinan kebenaran

1. 2 buah kesalahan pada data yaitu pada digit ke 2 dan ke 4, atau digit ke 3 dan 5 '0101010'
2. atau kesalahan bahwa bilangan yang seharusnya adalah '0111100' yang memenuhi kaedah Hamming Code

Jika hanya digunakan Hamming Code (7,4) maka banyak sekali data seperti ini yang tidak dapat diketahui kebenarannya, berbeda dengan menggunakan Hamming Code (8,4).

Pengecekan dengan menggunakan Matriks (7,4)

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \text{mod } 2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Hamming code yang benar akan menghasilkan matriks berisi dengan nilai 0.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 3 \\ 3 \end{pmatrix} \text{mod } 2 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Hamming Code yang salah tidak akan memberikan nilai nol semua untuk hasil perhitungan dengan menggunakan matriks pengecekan.

Pencarian Kesalahan pada Hamming (8,4)  
Misal diberikan kode masukan sebagai '00101100'

Langkah 1.

$$P4 = 1 \oplus 2 \oplus 3 \oplus 4 \oplus 5 \oplus 6 \oplus 7$$

$$P4 = 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0$$

$$P4 = 1$$

Seharusnya digit terakhir adalah 1 bukan 0,  
Lalu pengecekan akan dilakukan pada 7 digit yang lain, dan bila 7 digit yang lain memenuhi, maka digit terakhirlah yang salah. Dan bila terdapat kesalahan, maka kesalahan yang terjadi adalah kesalahan 1 digit.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \text{mod } 2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

karena ketujuh digit yang lain benar, maka kesalahan dapat kita simpulkan ada pada digit terakhir dari data yang mengecek parity dari 7 data yang lain.

Pengecekan yang dilakukan pada Hamming Code 8 digit haruslah dilakkan sebanyak 2 kali, karena kemungkinan terjadi kesalahan bila salah satu bernilai benar.

#### 4. KESIMPULAN

Error Correction code digunakan dalam berbagai alat elektronik untuk memvalidasi data yang diterima. Pengecekan pada linear error correction code yang memeriksa data dengan tipe bit dapat dilakukan dengan mudah dengan membandingkan jumlah kemuncuan bit bernilai 1.

Hamming Code merupakan salah satu error correction code yang sering digunakan dalam berbagai perangkat elektronik dan mampu memenuhi SECDED.

Hamming Code efektif sebagai SECDED untuk berapapun jumlah data asalkan dengan penambahan parity bit code pada digit terakhir, dan memenuhi semua digit yang lain.

Penggunaan Matriks untuk membentuk dan mengecek Hamming Code sangat efektif digunakan untuk menghemat waktu, Penggunaan Matriks ini biasanya digunakan pada alat-alat elektronik.

Penerapan Logika pada bilangan biner sangat berguna seperti penggunaan aljabar boolean dalam matriks untuk mengecek Hamming Code.

#### DAFTAR REFERENSI

[http://en.wikipedia.org/Hamming\\_code.htm](http://en.wikipedia.org/Hamming_code.htm)

[http://en.wikipedia.org/Hamming\(7,4\).htm](http://en.wikipedia.org/Hamming(7,4).htm)

<http://en.wikipedia.org/error-correction-code.htm>