Jurnal Teknovasi
Volume 03, Nomor 2, 2016, 01-09
ISSN : 2355-701X

DETEKSI DAN KOREKSI MULTI BIT ERROR DENGAN PARTITION HAMMING CODE

Fajar Muhajir[1], Syahril Efendi[2] & Sutarman[3]
[1,2,3]Program Studi Pasca Sarjana, Teknik Informatika, Universitas Sumatera Utara
Jl. Universitas No. 9 Kampus USU Padang Bulan, Medan 20115 Sumatera Utara
*E-mail : fajar.muhajir@yahoo.co.id

ABSTRAK

Mengirimkan data dari pengirim ke penerima merupakan salah satu bentuk komunikasi. Data dapat dikirim menggunakan saluran *nirkabel* ataupun saluran kabel, pada saat proses data dikirimkan dapat terjadi *error* yang disebabkan oleh saluran yang bising. *Error* yang terjadi dapat merusak data yang dikirim, *error* tersebut dapat berupa *single bit error* atau *multi bit error*. *Error* dapat diperbaiki dengan menerapkan *error control coding*. *Hamming code* merupakan salah satu contoh teknik *error control coding* yang dapat mendeteksi dan mengkoreksi *error*. Pada penelitian ini dianalisa metode untuk mendeteksi dan mengkoreksi *multi bit error* pada pesan yang dikirimkan menggunakan *partition hamming code*. Sebelum data dikirim ke penerima, pengirim membuat pola *partition bit* pesan yang lebih kecil dari *hamming code* (7,4) dan menambahkan *bit parity* dalam semua blok pesan yang dipecah sehingga menjadi sebuah *codeword* baru. *Partition* digunakan agar penerima mudah dalam mendeteksi dan mengkoreksi *multi bit error*.

Kata Kunci : Deteksi, Koreksi*, Multi Bit Error, Partition, Hamming Code.*

PENDAHULUAN

Komunikasi memiliki peranan yang besar bagi kehidupan dalam berbagai hal, banyak kegiatan sehari-hari yang menggunakan sistem komunikasi. Pada saat pengiriman data atau pesan yang ditransmisikan melalui sistem komunikasi dapat terpengaruh oleh kebisingan (*noise*) yang menyebabkan terjadinya *error*. Banyak faktor penyebab data yang dikirimkan mengalami *error*, diantaranya gangguan cuaca, tegangan listrik tidak stabil, radiasi dan *cross talk* (Fu, et al., 2009).

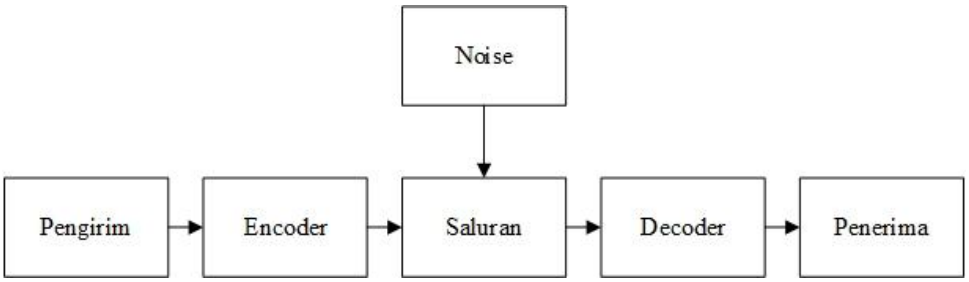
Pada awal lahirnya sistem komunikasi, permasalahan menarik yang selalu dibahas adalah teknik mengirim informasi dalam konsep yang mudah dimengerti dan berbiaya murah. Konsep yang digunakan untuk mengirim data diperkenalkan dalam bentuk kode binary, sehingga memberikan kemudahan dan informasi tentang keadaan data pada saat pengiriman. Salah satu upaya yang dapat dilakukan untuk memperbaiki data yang mengalami *error* adalah dengan menerapkan *error control coding*. Menurut Shannon, komunikasi melalui saluran yang bising (*noise*) dengan kesalahan sekecil mungkin diperlukan sebuah *encoding* dan *decoding* yang tepat untuk mendeteksi dan mengkoreksi bit yang *error* (Shannon, 1948). *Error detection* dan *error correcting* merupakan *encoding* dan *decoding* yang dapat diterapkan dalam *error control coding* (ECC). ECC dapat digunakan dalam pengamanan data untuk menghindari kerusakan pada data yang dikirimkan. *Hamming code* merupakan salah satu jenis *error detection* dan *error correction* yang banyak digunakan pada berbagai sistem pengiriman data (Wismal, et al., 2014). *Hamming code* memperkenalkan teknik untuk mendeteksi dan mengkoreksi *error* yang berada pada sisi penerima. *Hamming code* bekerja dengan menambahkan beberapa bit ekstra pada beberapa posisi bit pesan yang disebut cek bit (*bit parity*), dengan menggunakan logika Ex-OR untuk mencari nilai cek bit berdasarkan nilai bit pesan (Abuelyaman, et al., 2008). Jika kebisingan (*noise*) terjadi cukup tinggi pada saat pengiriman, bit pesan yang diterima bisa mengalami *error* (Demir, et al., 2006). Permasalahan yang terjadi disaat pengiriman data yang mengalami gangguan karena saluran bising (*noise*), dapat menyebabkan bit data yang dikirim mengalami *error*. Dalam 1 blok pesan (*codeword*) yang dikirim berisi 7 bit data mengalami *multi bit error,* sehingga membuat *hamming code* tidak

dapat mendeteksi dan mengkoreksi *multi bit error*. Maka dilakukan penelitian untuk memecah (*partition*) bit pesan tersebut menjadi pesan yang lebih kecil dan menambahkan *bit parity* pada setiap bit pesan.

Sistem Komunikasi

Dalam komunikasi data digital, tingkat akurasi data yang dikirim pada saluran transmisi sangat diperlukan. Ada beberapa jenis komunikasi pada umumnya yaitu: satelit, transmisi data, penyimpanan data, *mobile*, transfer file, dan transmisi audio atau video digital (Aflakian, et al., 2011). Data atau pesan dapat ditransmisikan melalui saluran *nirkabel* atau kabel. Dalam penggunaan saluran *nirkabel* atau kabel tidak selamanya informasi tersebut sampai dengan benar, bisa saja terjadi *error* yang menyebabkan pesan yang dikirim berbeda dengan pesan yang diterima.

Saluran transmisi adalah media fisik yang dilalui informasi yang dikirimkan, seperti saluran telepon, atau atmosfer dalam kasus komunikasi *nirkabel*. Gangguan yang tidak diinginkan (*noise*) dapat terjadi di saluran komunikasi dan menyebabkan informasi yang diterima menjadi berbeda dari informasi asli dikirim (Shannon, 1948). Kesalahan (error) dapat mengubah atau merusak isi data pesan yang mengalami *noise*. Data biner yang rusak dapat berubah dari 1 menjadi 0 atau sebaliknya 0 menjadi 1 (Bogart Jr, 1992). Ada beberapa jenis kesalahan yang bisa terjadi selama saluran mengalami gangguan yaitu single bit *error*, multi bit *error* dan burst *error* (Forouzan, 2007). Kinerja sistem komunikasi dapat diukur dengan output daya rasio signal to noise atau disebut *error* probabilitas (Forouzan, 2007). Dasar-dasar komunikasi *nirkabel* secara resmi didirikan oleh Claude E. Shannon pada tahun 1948. Shannon menegaskan bahwa dengan pengkodean yang tepat dari informasi atau pesan, kesalahan yang disebabkan oleh saluran bising (*noise*) dapat dikurangi ke tingkat yang diinginkan ketika data rate dibatasi oleh kapasitas saluran (Shannon, 1948).



Gambar 1. Rincian proses transmisi data

Gambar 1 menunjukkan proses pengiriman data dari pengirim ke penerima dan dapat kapan saja terjadi *error* yang disebabkan saluran yang bising (*noise*) pada saat pengiriman. Untuk dapat mendeteksi dan mengkoreksi kesalahan pada saat pengiriman, lapisan data link sangat berperan penting pada sistem komunikasi. Lapisan data link bertanggung jawab untuk mengatur *framing*, *addresing*, *flow control*, *error control* dan *access control* (Forouzan, 2007). Penggunaan saluran *wireless* pada komunikasi data digital sangat mudah mengalami gangguan, solusi yang paling tepat untuk memperbaiki kesalahan (*error*) yang disebabkan oleh saluran yang bising (*noise*) dengan menerapkan *error detection* dan *error correction*, dikareakan *error detection* dan *error correction* dapat menghemat penggunaaan *bandwith* tanpa harus mengirim kembali (Tanenbaum, et al., 2011).

Ada banyak teknik pengkodean yang dikembangkan untuk sistem komunikasi agar mencapai kapasitas saluran yang baik pada saat pengiriman dalam saluran yang bising (*noise*). Pada tahun 1950, Richard W. Hamming menemukan kode blok linear untuk mendeteksi dan mengkoreksi kesalahan (*error*). *Hamming code* adalah salah satu kode untuk mendeteksi dan mengoreksi kesalahan tunggal (*single bit error*) dalam setiap blok kode pesan (*codeword*) (Hamming, 1950). Hamming code merupakan sebuah kode yang sempurna dan dapat diterjemahkan dengan mudah. Hamming kode telah banyak diterapkan dalam komunikasi digital dan sistem penyimpanan data, karena memilki keberhasilan yang tinggi dan sistem *encoding* dan *decoding* yang cepat dan akurat (Lin, et al., 2004) .

Error Control Coding

Error control coding (ECC) selalu dikaitkan dengan konsistensi komunikasi melalui saluran yang bising (*noise*). *Error control coding* selalu dipakai pada komunikasi satelit, transmisi data, penyimpanan data, komunikasi *mobile*, transfer file, dan transmisi audio atau video digital. *Error control coding* mempunyai tugas untuk mengatur jumlah data yang dikirimkan kepada penerima dengan menggunakan sebuah algoritma. Dengan adanya pengaturan jumlah data ini, maka *error control coding* dapat menjamin tidak terjadi penumpukan data pada sisi penerima (Ullah, et al., 2011) . *Error control coding* berkaitan dengan deteksi dan koreksi kesalahan transmisi yang disebabkan oleh kebisingan (*noise*) di saluran. Dalam penerapannya, *error control coding* mengedepankan beberapa aspek penting diantaranya adalah *encoding* pesan yang cepat, pesan yang dikodekan mudah ditransmisikan, pesan yang diterima cepat untuk *decoding*, transfer maksimum informasi dalam satu waktu, kemampuan maksimal deteksi atau koreksi (Hill, 1986). *Error control coding* banyak digunakan dalam kehidupan sehari-hari seperti dalam membaca *compact disc* (CD), menerima transmisi dari satelit, atau sistem *mobile* (Huffman, et al., 2003). *Error control coding* telah digunakan secara luas dalam sistem komunikasi digital dan sistem penyimpanan digital karena efektivitas dalam mencapai hasil yang efisien dan tingkat perbaikan kesalahan yang tinggi.

Penggunaan teknik *error control coding* menjadi keharusan bagi pengguna sistem komunikasi digital atau sistem penyimpanan digital, untuk mendeteksi kesalahan yang terjadi dapat dilakukan dengan menambahkan bit cek (*bit parity*) ke dalam data pesan asli. Pemilihan *error control coding* harus sesuai untuk jenis kesalahan yang terjadi pada komunikasi digital atau sistem penyimpanan data (Imai, 1990) .

Hamming Code

Pada tahun 1947 Richard Wesley Hamming membuat penelitian tentang *hamming code*, Richard menyatakan bahwa jika m adalah bit pesan dan p adalah cek bit maka dapat menghasilkan sebuah *codeword* untuk bisa mendeteksi dan mengkoreksi kesalahan pada saat pesan ditransmisikan (Hamming, 1950).

Hamming code adalah salah satu kode yang dapat mendeteksi dan mengkoreksi *single bit error* dengan menambahkan beberapa *bit parity* berdasarkan bit pesan, *hamming code* juga bisa mendeteksi dan mengkoreksi *burst error*. Hamming code menggunakan operasi logika Ex-OR (Exclusive–OR) dalam proses pendeteksian maupun proses pengoreksian *error*, sedangkan input dan output data dari algoritma *hamming code* berupa bilangan binary. *Hamming code* (7,4) dengan 7 bit pesan dan 4 bit parity, *bit parity* dicari menggunakan modulo-2 atau Ex-OR berdasarkan letak bit pesan (Tanenbaum, et al., 2011).

Bit parity yang telah dicari kemudian disisipkan ke dalam bit pesan dan ditempatkan pada posisi yang dihitung berdasarkan rumus perhitungan posisi check bit berikut :

$Ci = 2i - 1$

Sehingga didapat tabel 1 menunjukan posisi check bit seperti berikut:

Tabel 1. Posisi Check Bit

Check bit	r1	r2	r3	r4	r5	r6	r7	r8	r9
Posisi check bit	1	2	4	8	16	32	64	128	256

Untuk mencari nilai dari *bit parity* harus berdasarkan dari nilai dan letak bit pesan, jika bit pesan 2n bit, maka jumlah *bit parity* yang disisipkan ada sebanyak c = (n + 1) bit. Sehingga didapat tabel 2. data bit dan check bit.

Tabel 2. Tabel Jumlah Check Bit

Bit Pesan	1	2	4	8	16	32	64	128	256
Check bit	2	3	3	4	5	6	7	8	9

Jurnal Teknovasi
Volume 03, Nomor 2, 2016, 1-10
ISSN : 2355-701X

Jumlah keseluruhan bit pesan dan *bit parity* dalam *hamming code* disebut *codeword* dan diberi nomor secara berurutan dimulai dengan bit 1 pada ujung kiri. Bit yang merupakan pangkat 2 (1, 2, 4, 8, 16, seterusnya) adalah *bit parity* disebut r. Sisanya (3, 5, 6, 7, 9) adalah bit pesan disebut b.

Tabel 3. Posisi Check Bit Dan Bit Pesan

Posisi bit	1	2	3	4	5	6	7	8	9	10	11
Kode bit	r1	r2	b1	r3	b2	b3	b4	r4	b5	b6	b7

Dari tabel 3 dapat dilihat posisi dari bit pesan dan *bit parity* dalam *hamming code*. Setelah posisi *bit parity* diketahui, selanjutnya dicari nilai *bit parity* berdasarkan nilai dan posisi bit pesan yang ditunjukkan dibawah ini :

r1 = XOR dari bit 1, 3, 5, 7, 9, 11
r2 = XOR dari bit 2, 3, 6, 7, 10, 11
r3 = XOR dari bit 4, 5, 6, 7
r4 = XOR dari 8, 9, 10, 11

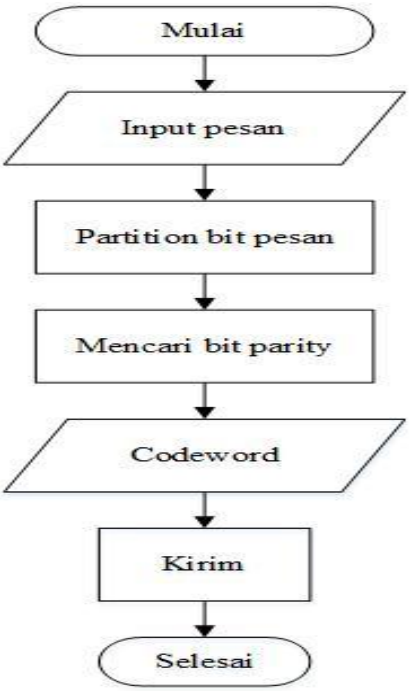
METODE PENELITIAN

Data yang digunakan

Data yang digunakan dalam penelitian ini adalah pesan teks, yang dapat dikenali dengan karakter ASCII dan dirubah menjadi kode binary untuk memudahkan dalam mendeteksi dan mengkoreksi *error* yang terjadi.

Rancangan Sistem

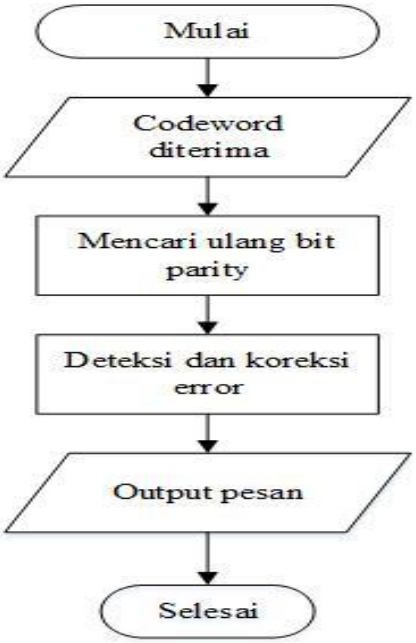
Berikut ini adalah rancangan dari sistem yang akan penulis kerjakan untuk menyelesaikan permasalahan. Berdasarkan Gambar 2. proses awal dimulai dengan input sebuah pesan yang ingin dikirim, pesan tersebut menggunakan karakter yang kemudian dirubah menjadi desimal dan dirubah menjadi binary menggunakan tabel ASCII. Selanjutnya pilih pola pesan yang ingin dipakai untuk pengiriman dan mencari nilai *bit parity* berdasarkan pencarian bit pesan untuk ditambahkan pada bit pesan. Pesan yang telah ditambahkan *bit parity* menjadi sebuah *codeword* baru dan dikirim ke penerima.



Gambar 2. Flowchart Proses Pegiriman Pesan

Berdasarkan Gambar 3. setelah *codeword* yang dikirim sampai kepada penerima, penerima kemudian mencari kembali nilai *bit parity* berdasarkan pesan yang diterima. Setelah *bit parity* diketahui dan selanjutnya penerima mencocokkan nilai *bit parity* yang diterima dengan *bit parity* yang dicari oleh penerima untuk mengetahui *error* yang terjadi. Saat proses pengiriman bit pesan bisa saja dalam saluran transmisi terjadi kesalahan (*error*) yang menyebabkan beberapa bit data dalam *codeword* yang dikirim berubah dari 1 manjadi 0 atau dari 0 menjadi 1. *Error* yang terjadi dapat diartikan sebagai terjadinya gangguan (*noise*) dalam pengiriman *codeword*.

Setelah *bit parity* dari sisi penerima telah dicari kemudian dicocokkan dengan *bit parity* dari pengirim. Jika semua *bit parity* pengirim sesuai dengan *bit parity* penerima, artinya tidak ada bit pesan yang *error*. Tetapi jika ada 1 atau lebih *bit parity* dari penerima yang tidak sama dengan *bit parity* pengirim maka ada terdeteksi bit pesan yang *error*. Setelah terdeteksi *error* pada pesan, maka selanjutya penerima mengkoreksi dimana letak *error* yang terjadi. Untuk mencari letak *error* yang terjadi dengan cara menghitung jumlah dan letak *bit parity* yang tidak sama dan menjumlahkannya, maka akan didapat dimana letak bit *error* yang terjadi.



Gambar 3. Flowchart Proses Penerima Pesan

Rancangan Partition Hamming Code

Sebelum pesan dikirim ke penerima, pesan terlebih dahulu dipecah (*partition*) menjadi beberapa pola pesan yang lebih kecil. Kemudian dicari nilai *bit parity* pada setiap pola bit pesan yang telah dipecah dan kemudian *bit parity* beserta bit pesan dikirim kepada penerima. Setelah pesan sampai kepada penerima, penerima juga mencari kembali nilai *bit parity* pada pesan yang telah sampai dan membandingkan *bit parity* yang dicari dengan *bit parity* yang dikirim untuk mengetahui *error* yang terjadi. Dalam penggunaan *partition*, *hamming code* hanya dapat mendeteksi dan mengkoreksi 1 atau 2 bit pesan yang *error* dalam satu blok pesan (*codeword*), jika *error* yang terjadi lebih banyak maka *hamming code* hanya dapat mendeteksi dan tidak bisa mengkoreksi pesan yang *error*. Berikut contoh hamming code dalam penggunaan pemecah (*partition*) pesan:

Jurnal Teknovasi
Volume 03, Nomor 2, 2016, 1-10
ISSN : 2355-701X

Tabel 4. Partition 1 Bit Pesan

Bit Pesan	Bit + Parity	Codeword	Kirim	Diterima	Bit + Parity	Check Error	Letak Error
1	xx1	111	111	110	xx0	000	1+2 = 3
0	xx0	000	000	001	xx1	111	1+2 = 3
1	xx1	111	111	111	xx1	111	1+2 = 3
0	xx0	000	000	000	xx0	000	1+2 = 3
1	xx1	111	111	110	xx0	000	1+2 = 3
0	xx0	000	000	001	xx1	111	1+2 = 3
1	xx1	111	111	110	xx0	000	1+2 = 3

Tabel 5. Partition 2 Bit Pesan

Bit Pesan	Bit + Parity	Codeword	Kirim	Diterima	Bit + Parity	Check Error	Letak Error
10	xx1x0	11100	11100	11000	xx0x0	00000	1+2 = 3
10	xx1x0	11100	11100	11000	xx0x0	00000	1+2 = 3
10	xx1x0	11100	11100	11101	xx1x1	01111	1+4 = 5
1	xx1	111	111	111	xx1	111	-

Tabel 6. Partition 3 Bit Pesan

Bit Pesan	Bit + Parity	Codeword	Kirim	Diterima	Bit + Parity	Check Error	Letak Error
101	xx1x01	101101	101101	100101	xx0x01	010101	1+2 = 3
010	xx0x10	100110	100110	100111	xx0x11	110011	2+4 = 6
1	xx1	111	111	111	xx1	111	-

Tabel 7. Partition 4 Bit Pesan

Bit Pesan	Bit + Parity	Codeword	Kirim
1010	xx1x010	1011010	1011010
101	xx1x01	101101	101101
Diterima	**Bit + Parity**	**Check Error**	**Letak Error**
1001010	xx0x010	0111010	1+2=3
101100	xx1x00	11100	2+4=6

Tabel 8. Partition 5 Bit Pesan

Bit Pesan	Bit + Parity	Codeword	Kirim
10101	xx1x010x1	001101011	001101011
01	xx0x1	00011	00011
Diterima	**Bit + Parity**	**Check Error**	**Letak Error**
001101111	xx1x011x1	111001111	1+2+4 = 7
00010	xx0x0	00000	1+4 = 5

Tabel 9. Partition 6 Bit Pesan

Bit Pesan	Bit + Parity	Codeword	Kirim
101010	xx1x010x10	0011010110	0011010110
1	xx1	111	111
Diterima	**Bit + Parity**	**Check Error**	**Letak Error**
0011010111	xx1x010x11	0111010011	2+8 = 10
110	xx0	000	1+2 = 3

Tabel 10. Partition 7 Bit Pesan

Bit Pesan	Bit + Parity	Codeword	Kirim
1010101	xx1x010x101	11110100101	11110100101
Diterima	**Bit + Parity**	**Check Error**	**Letak Error**
11010100101	xx0x010x101	00010100101	1+2 = 3

Analisis algoritma

Pengujian deteksi dan koreksi menggunakan *partition hamming code* yang dilakukan untuk mengetahui apakah dengan menggunakan *hamming code* dapat mendeteksi dan mengkoreksi *multi bit error*, adapun bentuk pengujiannya sebagai berikut :

1. Pengujian deteksi dan koreksi dilakukan dengan menggunakan pola *partition hamming code*, dimana pesan yang akan dikirim dipecah (*partition*) menjadi pola bit pesan yang lebih kecil dan ditambahkan dengan *bit parity* pada setiap blok pesan yang dipecah sehingga menjadi sebuah *codeword*.
2. Pemecahan pola bit pesan bertujuan untuk memudahkan dalam mendeteksi dan mengkoreksi *single bit error* dan *multi bit error*.
3. Dalam penggunaan *partition hamming code* banyak memakai *bit parity* dibandingkan dengan *hamming code* (7,4), dan bertujuan untuk dapat mendeteksi dan mengkoreksi *multi bit error*.
4. *Codeword* yang dipakai dalam pengiriman menggunakan *partition hamming code* lebih banyak dibanding dengan *hamming code* (7,4).

Hasil pada penelitian deteksi dan koreksi *multi bit error* menggunakan *partition hamming code*, dapat memudahkan pengirim dan penerima dalam mendeteksi dan mengkoreksi pesan yang terdapat *multi bit error* dalam pengiriman.

HASIL DAN PEMBAHASAN

Hasil Partition Hamming Code

Analisis total bit pesan yang dipecah dan ditambahkan *bit parity* pada setiap blok pesan yang telah dipecah sangat mempengaruhi tingkat efisiensi dalam proses pengirim pesan. Dari hasil analisis tersebut dapat dihasilkan perbandingan diantara metode *partition* bit pesan dengan *hamming code* (7,4) untuk mendeteksi dan mengkoreksi *multi bit error*. Untuk perbandingan *partition hamming code* dapat dilakukan dengan membandingkan *bit overhead*, *code rate* dan jumlah *error* yang dapat dideteksi, penulis memakai contoh pesan sebanyak 21 bit pesan yang ditambahkan *bit parity* pada setiap *codeword* dan didapat hasil pembanding.

Jurnal Teknovasi
Volume 03, Nomor 2, 2016, 1-10
ISSN : 2355-701X

Tabel 11. Analisis perbandingan *multi bit error*

No	Jumlah pemecah bit	Data bit (k)	Total *parity* (r)	*Codeword* (n=k+r)	Bit Overhead (BO=r/k)	Code rate (R=k/n)	Jumlah koreksi error bit
1	(1,2)	21	42	64	200%	32,8%	21
2	(2,3)	21	32	54	152%	38,8%	11
3	(3,3)	21	21	42	100%	50%	7
4	(4,3)	21	17	38	80,9%	55,2%	6
5	(5,4)	21	18	39	85%	53,8%	5
6	(6,4)	21	14	35	66,6%	60%	4
7	(7,4)	21	12	33	57%	63,6%	3
8	Burst	21	21	42	100%	50%	7

Dari tabel diatas dapat dilihat perbandingan dari setiap pemecah bit pesan dalam mendeteksi dan mengkoreksi *multi bit error* menggunakan *partition hamming code*, didapat perbadingan dalam penggunaan *bit parity* dan jumlah codeword dan jumlah *error* yang dapat dideteksi dan dikoreksi. Dalam penggunaan pemecah (1,2) bit pesan ditambahkan 2 *bit parity*, dapat mendeteksi dan mengkoreksi 21 bit pesan. Akan tetapi sangat banyak dalam penggunaan *bit parity* yang mengakibatkan *bit overhead* meningkat dan *codeword* menjadi lebih banyak. Berbeda dengan *hamming code* (7,4) yang hanya dapat mendeteksi 3 bit *error* akan tetapi sedikit dalam pemakaian *bit parity*.

Pembahasan

Dari analisa metode *hamming code* dengan menggunakan *partition* bit, dapat disimpulkan bahwa *partition hamming code* dapat mendeteksi dan mengkoreksi banyak *error* yang terjadi pada pesan yang ditransmisikan dibandingkan dengan mengirim 7 bit pesan sekaligus tanpa dipecah terlebih dahulu. Kelemahan yang didapat dari penggunaan *partition hamming code* dalam mentransmisikan pesan adalah menyebabkan banyak penggunaan *bit parity* dan membuat *codeword* menjadi lebih banyak dari biasanya, dikarenakan dalam semua blok pesan yang telah dipecah menggunakan *bit parity* untuk mendeteksi dan mengkoreksi *error* yang terjadi pada saat pengiriman.

Untuk dapat mengurangi penggunaan *bit parity* yang terlalu banyak pada setiap blok pesan yang dipecah menjadi lebih kecil, dapat diterapkan metode *auto repetition request* (ARQ) pada saat pengiriman blok pesan dengan mengirim *acknowledgement* (ACK). Apabila ada *error* yang tidak dapat dikoreksi oleh *hamming code*, penerima dapat mengirim *acknowledgement* (ACK) kepada pengirim yang memberitahukan *error* yang terjadi tidak dapat dikoreksi. Setelah pengirim menerima ACK dari penerima, pengirim dapat merubah pola *partition* yang lain agar penerima dapat mendeteksi dan mengkoreksi kesalahan yang terjadi.

KESIMPULAN

Berdasarkan analisis dari pengujian sistem secara menyeluruh yang dilakukan, maka ada beberapa hal yang dapat dijadikan kesimpulan pada penelitian ini, antara lain Pemakain pola *partition* bit pesan menjadi beberapa blok bit pesan yang lebih kecil, *hamming code* dapat mendeteksi dan mengkoreksi multi bit *error* yang terjadi. Dengan memakai pola *partition* bit pesan menjadi 1 bit pesan dalam 1 blok pesan, *hamming code* dapat mendeteksi dan mengkoreksi *error* dalam semua blok pesan. Dengan pemakaian pola *partition* bit pesan, *hamming code* bekerja lebih lambat dikarenakan banyak menggunakan *bit parity* dalam semua *codeword*.

Jurnal Teknovasi
Volume 03, Nomor 2, 2016, 01-09
ISSN : 2355-701X

DAFTAR PUSTAKA

Abuelyaman, E. S., & Al-Sehibani, A.-A. s. (2008). Optimization of the Hamming Code for Error Prone Media. *International Journal of Computer Science and Network Security, 8*, 278-285.

Aflakian, D., Siddiqui, T., Khan, N. A., & Aflakian, D. (2011). Error Detection and Correction over Two-Dimensional and Two-Diagonal Model and Five-Dimensional Model. *International Journal of Advanced Computer Science and Applications, 2*(07), 16-19.

Bogart Jr, T. F. (1992). *Introduction to Digital Circuits.* International Edition.

Demir, U., & Akta, O. (2006). Raptor versus Reed Solomon Forward Error Correction Codes. *International Symposium on Computer Networks, 6*, 264-269.

Forouzan, B. (2007). *Data Communications and Networking.* United States: McGraw-Hill.

Fu, B., & Ampadu, p. (2009). On Hamming Product Codes With Type-II Hybrid ARQ for On-Chip Interconnects. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, 56*, 2042-2054.

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes. *Bell System Technical Journal, 29*, 147-160.

Hill, R. (1986). *A first course in coding theory.* Oxford: Clarendon Press.

Huffman, W. C., & Pless, V. (2003). *Fundamentals of Error Correcting Codes.* Cambridge University Press.

Imai, H. (1990). *Essentials of Error-Control Coding Techniques.* Kanagawa: Academic Press, Inc.

Lin, S., & Costello, D. J. (2004). *Error Control Coding: Fundamentals and Application.* London: Prentice-Hall, Inc.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal, 27*, 379-423.

Tanenbaum, A., David, J., & Wetherall. (2011). *Computer Network.* United States: Pearson Education. Inc.

Ullah, R., Khan, J., Latif, S., & Ullah, I. (2011). Indication of Efficient Technique for Detection of Check Bits in Hamming Code. *Spriger, 8*, 344-356.

Wismal, A., Suwadi, & Suryani, T. (2014). Implementasi Encoder dan Decoder Hamming pada DSK TMS320C6416T. *Jurnal Teknik POMITS Vol. 3, No. 1*, A 40 - 45.