

IMPLEMENTASI ENCODER REED-SOLOMON PADA FPGA BERBASIS CCSDS (REED-SOLOMON ENCODER IMPLEMENTATION ON FPGA BASED ON CCSDS)

Patria Rachman Hakim¹, Abdul Rahman, Deddy El Amin, Widya Roza dan Elvira Rahim
Peneliti Pusat Teknologi Satelit, LAPAN
¹e-mail: patriarachmanhakim@yahoo.com

ABSTRACT

One of the function of Payload Data Handling (PDH) system on a satellite is to do channel coding on satellite image data. CCSDS (Consultative Committee for Space Data Systems) recommends the use of Reed-Solomon (RS) encoder for this channel coding purpose. To be able to realize high data rate transmission, RS encoder algorithm implementation on satellite PDH system needs FPGA (Field Programmable Gate Array). This research aims to design RS(255,223) encoder based on CCSDS and then to implement the designed encoder on FPGA with more optimal circuit design compared to commercial RS encoder (IP-core). Based on experiments done, the designed encoder has some advantages in terms of efficiency of logic gate used and the performances of the resulted output data. Furthermore, this research also develops parallel encoding method which will be implemented on satellite PDH system. Experiment results show that by using this parallel method, the resulted output data has a higher data rate and does not need dummy data to be appended to the output data. It is hoped that these two results can support current satellite PDH system development in Satellite Technology Center.

Keywords: *Channel coding, Reed-Solomon encoder, PDH, FPGA, CCSDS*

ABSTRAK

Salah satu fungsi sistem *Payload Data Handling* (PDH) pada sebuah satelit adalah melakukan *channel coding* untuk data citra satelit. *Consultative Committee for Space Data Systems* (CCSDS) telah merekomendasikan penggunaan *encoder Reed-Solomon* (RS) untuk keperluan *channel coding* tersebut. Untuk dapat merealisasikan transmisi dengan laju data yang tinggi, maka implementasi algoritma *encoder* RS pada sistem PDH satelit membutuhkan *Field Programmable Gate Array* (FPGA). Penelitian ini bertujuan untuk merancang modul *encoder* RS(255,223) berbasis CCSDS dan mengimplementasikan *encoder* tersebut pada FPGA dengan desain rangkaian yang lebih optimal dibandingkan dengan *encoder* RS komersial (*IP-core*). Berdasarkan hasil pengujian yang telah dilakukan, *encoder* yang dirancang memiliki beberapa kelebihan dalam hal efisiensi gerbang logika yang digunakan dan tingkat kinerja data keluaran yang dihasilkan. Selain itu, pada penelitian ini juga dikembangkan metode *encoding* paralel yang akan diterapkan pada sistem PDH satelit. Hasil pengujian menunjukkan bahwa dengan menggunakan metode tersebut, data keluaran yang dihasilkan *encoder* memiliki laju data yang lebih tinggi dan tidak membutuhkan data *dummy* untuk melengkapi data keluaran. Kedua hasil tersebut diharapkan dapat mendukung pengembangan sistem PDH satelit yang dilakukan di Pusat Teknologi Satelit saat ini.

Kata kunci: *Channel coding, Encoder Reed-Solomon, PDH, FPGA, CCSDS*

1 PENDAHULUAN

Penyimpanan dan pengiriman data digital umumnya diasosiasikan dengan beberapa teknik pengolahan data diantaranya yaitu teknik kompresi dan enkripsi data serta teknik koreksi error. Teknik kompresi data adalah suatu metode untuk mengurangi volume sebuah data dengan tetap menjaga kualitas data tersebut. Teknik enkripsi data merupakan suatu metode untuk mengacak susunan sebuah data dengan tujuan agar data tersebut sulit dipahami oleh pihak lain. Sementara itu, teknik koreksi error adalah suatu metode untuk menjaga kebenaran sebuah data selama proses penyimpanan atau pengiriman. Ketiga teknik pengolahan data tersebut dapat diaplikasikan pada sisi pengirim data (*encoder*) dan pada sisi penerima data (*decoder*). Pada sistem satelit, *Payload Data Handling* (PDH) melakukan fungsi *encoder* yaitu melakukan proses kompresi, enkripsi dan *encoding* untuk data yang akan ditransmisikan, sedangkan sisi ruas bumi akan melakukan proses *decoding*, dekripsi dan dekompresi untuk data yang diterima.

Consultative Committee for Space Data Systems (CCSDS) telah menyusun beberapa rekomendasi untuk melakukan ketiga proses pengolahan data di atas. CCSDS merekomendasikan penggunaan metode *Discrete Wavelet-Transform* (DWT) dan *Bit-Plane Encoder* (BPE) untuk keperluan kompresi data citra [The Consultative Committee for Space Data Systems, 2005], sedangkan untuk kebutuhan enkripsi data satelit, CCSDS merekomendasikan metode *Advanced Encryption Standard* (AES) [The Consultative Committee for Space Data Systems, 2012]. Sementara itu untuk teknik koreksi error atau yang umum dikenal sebagai *channel coding*, untuk selanjutnya akan disebut *encoding/decoding*, CCSDS merekomendasikan penggunaan metode *Reed-Solomon* dan kode konvolusional [The Consultative

Committee for Space Data Systems, 2011]. Penelitian ini akan membahas mengenai proses perancangan *encoder* pada sistem PDH satelit, khususnya perancangan algoritma koreksi *Reed-Solomon* (RS), yang diaplikasikan pada *Field Programmable Gate Array* (FPGA). FPGA sangat diperlukan dalam sistem PDH satelit karena sistem PDH satelit membutuhkan akurasi dan kecepatan pengolahan data yang cukup tinggi.

Beberapa kegiatan penelitian mengenai *channel coding* telah dilakukan di Pusat Teknologi Satelit, diantaranya mengenai analisis kinerja algoritma *Reed-Solomon* [Dwiyanto, dan Sugihartono, 2011], simulasi *IP-core encoder* RS [Pratomo, B., 2012], dan perancangan sistem PDH satelit [Rosa, W., Amin, D.E., dan Nasser, E.N., 2014]. Walaupun demikian, ketiga penelitian tersebut lebih berfokus pada analisis dan aplikasi penggunaan algoritma *Reed-Solomon*, dimana dalam beberapa penelitian tersebut digunakan *IP-core encoder* RS komersial. Salah satu tujuan penelitian ini adalah menghasilkan modul *encoder* RS(255,223) berbasis FPGA Altera sebagai pengganti *IP-core encoder* komersial, dan selanjutnya dapat digunakan untuk berbagai aplikasi. Modul *encoder* RS yang dihasilkan diharapkan memiliki kinerja yang sama atau bahkan lebih baik dibandingkan dengan *encoder* komersial tersebut. Karena *encoder* yang akan dirancang ini bersifat *special-purpose* untuk keperluan PDH satelit LAPAN, maka terdapat beberapa bagian pada *encoder* komersial yang dapat disederhanakan sesuai dengan kebutuhan.

Selain menghasilkan modul *encoder* di atas, penelitian ini juga bertujuan untuk menghasilkan sistem PDH satelit yang lebih efisien. Pada sistem PDH yang telah dikembangkan di Pusat Teknologi Satelit [Rosa, W., Amin, D.E., dan Nasser, E.N., 2014], terdapat beberapa aspek yang dapat ditingkatkan, diantaranya yaitu mengenai kebutuhan

data *dummy* dan kecepatan data keluaran yang dihasilkan. Pada penelitian ini akan dikembangkan sistem *encoding* paralel dengan menggunakan modul *encoder* RS yang telah dirancang untuk meningkatkan kinerja sistem PDH. Konsep utama dari teknik *encoding* paralel adalah dengan membagi beban proses perhitungan data *parity* kepada beberapa modul *encoder* sehingga secara keseluruhan akan mempercepat proses *encoding*. Format data dan skema *encoding* yang digunakan dalam penelitian ini sama seperti pada sistem PDH sebelumnya yaitu berdasarkan rekomendasi CCSDS.

Makalah ini terdiri dari 5 bagian penulisan. Bagian pertama berisi mengenai latar belakang dan tujuan dilakukannya penelitian ini. Bagian kedua dan ketiga berisi mengenai teori dasar dan proses perancangan *encoder* RS(255,223) berdasarkan CCSDS yang diimplementasikan pada FPGA, serta metode *encoding* paralel pada sistem PDH satelit. Sementara itu, bagian keempat memberikan beberapa hasil simulasi dan pengujian menggunakan simulator ModelSim dan modul FPGA, berserta dengan analisis kinerja dari modul *encoder* yang telah dirancang. Akhirnya beberapa kesimpulan dan saran disajikan pada bagian kelima.

2 TEORI DASAR

2.1 Encoder Reed-Solomon (RS)

Algoritma *Reed-Solomon* (RS) merupakan sebuah algoritma koreksi *channel-coding* yang berfungsi untuk mendeteksi dan memperbaiki sebuah paket data. Secara umum, algoritma RS (n,k) menunjukkan bahwa algoritma ini memiliki k buah data masukan dan $(n-k)$ buah data *parity*, sehingga menghasilkan paket data yang terdiri dari n buah data keluaran yang akan ditransmisikan, atau disimpan pada proses penyimpanan data. Algoritma RS(n,k) tersebut dapat memperbaiki $(n-k)/2$ *byte error* yang terdapat dalam paket data tersebut.

CCSDS telah merekomendasikan penggunaan *encoder* RS(255,223) untuk proses transmisi data satelit. Pada RS(255,223) tersebut terdapat 223 *byte* data informasi dan 32 *byte* data *parity*, sehingga algoritma tersebut mampu mengoreksi 16 *byte error*. Pada keadaan terbaik, RS(255,223) dapat mengoreksi kesalahan 128 bit pada sebuah paket data jika 128 bit tersebut tepat membentuk 16 *byte* data, tetapi pada keadaan terburuk, RS(255,223) tidak dapat mengoreksi kesalahan 17 bit jika seluruh bit tersebut terdapat pada *byte* data yang berbeda.

Hal penting yang harus diperhatikan adalah lokasi *byte error* pada paket data tersebut. Jika suatu data mengalami *burst error* (*pike noise* pada transmisi data atau goresan panjang pada sebuah disk) sebanyak 17 *byte* berurutan atau lebih maka data tersebut tidak dapat diperbaiki. Untuk mengatasi hal tersebut, CCSDS telah merekomendasikan penggunaan teknik *interleave-5*. Pada teknik *interleave-5*, proses *encoding* dilakukan sebanyak lima kali secara serial dimana data masukan untuk setiap proses *encoding* ini merupakan hasil pencampuran dari lima buah paket data asli. Hal tersebut menyebabkan *burst error* yang terjadi pada salah satu paket data asli dapat diperbaiki karena *burst error* tersebut akan tersebar pada lima paket data baru setelah proses *interleaving*, sehingga algoritma koreksi ini dapat memperbaiki *burst error* tersebut, selama setiap paket baru tidak memiliki *error* sebanyak 17 *byte* atau lebih.

Selain telah merekomendasikan penggunaan *encoder* RS(255,223) dan teknik *interleave-5* di atas, CCSDS juga mengatur pemilihan polinomial primitif serta beberapa parameter *encoder* untuk lebih menyederhanakan rangkaian logika yang dibutuhkan pada implementasi *hardware*. Polinomial primitif yang disarankan adalah $Po(x)=x^8+x^7+x^2+x+1$, sedangkan parameter *encoder* yang

disarankan adalah $a=112$ dan $b=11$ [The Consultative Committee for Space Data Systems, 2011]. Dengan menggunakan ketiga parameter tersebut maka *encoder* RS(255,223) hasil rekomendasi CCSDS akan menggunakan polinomial generator $G(x)$ sebagai berikut:

$$G(x) = \sum_{i=0}^{32} G_i X^i$$

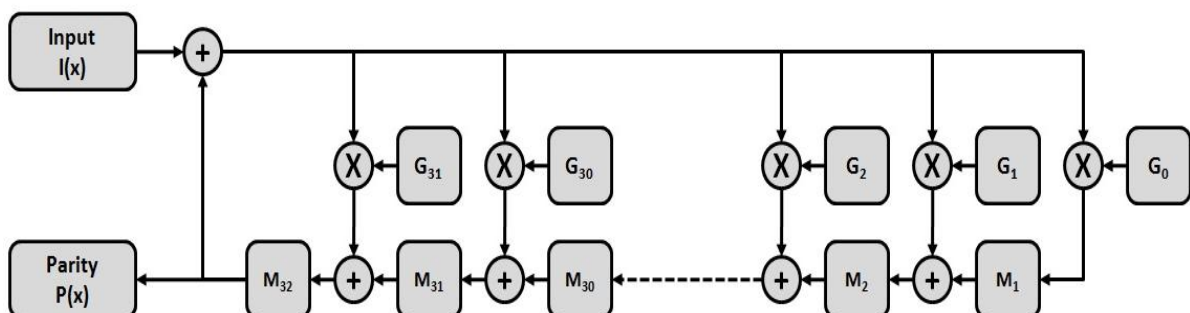
Simbol G_i merupakan konstanta koefisien generator $G(x)$, dimana pada encoder RS(255,223) terdapat 33 buah koefisien yang nilainya dapat dilihat pada Tabel 2-1 [The Consultative Committee for Space Data Systems, 2011].

Secara formal, modul *encoder* RS (255,223) akan menghasilkan 32 *byte* data *parity* $P(x)$ berdasarkan informasi 223 *byte* data masukan $I(x)$ yang akan ditransmisikan. Proses pembentukan data *parity* tersebut relatif sederhana, dimana data *parity* $P(x)$ tersebut

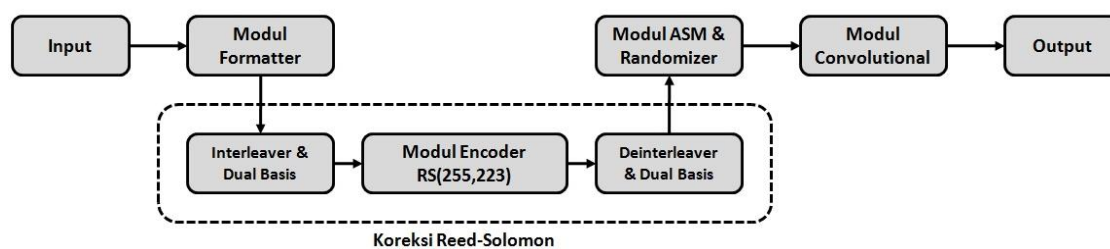
merupakan sisa pembagian antara 223 *byte* data masukan $I(x)$ terhadap 33 *byte* generator $G(x)$. Pada aritmatika dasar, sisa pembagian 100 terhadap 16 dapat ditentukan dengan mudah yaitu 4 karena $(6 \times 16) + 4 = 100$. Tetapi sayangnya perhitungan pada algoritma *channel-coding* membutuhkan operasi dalam *Finite Field* atau *Galois Field* [Sklar, B., 2001]. Walaupun seluruh operasi harus dilakukan pada lingkup *Galois Field* ini, tetapi secara umum teknik pembagian yang digunakan sama seperti teknik pembagian pada operasi aritmatika dasar. Operasi pembagian dalam *Galois Field* ini lebih mudah divisualisasikan dengan menggunakan skema rangkaian *Linear Feedback Shift Register (LFSR)* yang terdiri dari operasi perkalian dan penjumlahan serta operasi penundaan (memori). Skematik operasi pembagian dalam *Galois Field* dapat dilihat pada Gambar 2-1.

Tabel 2-1: NILAI KOEFISIEN GENERATOR POLINOMIAL $G(x)$

Koefisien $G(x)$	G_{32}	G_{31}	G_{30}	G_{29}	G_{28}	G_{27}	G_{26}	G_{25}	G_{24}	G_{23}	G_{22}	G_{21}	G_{20}	G_{19}	G_{18}	G_{17}	G_{16}
		1	91	127	86	16	30	13	235	97	165	8	42	54	86	171	32
	G_0	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	



Gambar 2-1: Skematik operasi pembagian dalam *Galois Field* [Sklar, B., 2001]



Gambar 2-2: Skematik sistem PDH satelit di Pusat Teknologi Satelit [Rosa, W., Amin, D.E., dan Nasser, E.N., 2014]

2.2 Payload Data Handling (PDH)

Pada sistem PDH satelit yang telah dikembangkan di Pusat Teknologi Satelit [Rosa, W., Amin, D.E., dan Nasser, E.N., 2014], data muatan yang akan ditransmisikan akan mengalami berbagai proses pengolahan data berdasarkan rekomendasi CCSDS, diantaranya yaitu proses *formatting*, proses *channel coding* menggunakan algoritma *Reed-Solomon* dan kode konvolusional serta proses randomisasi dan penambahan data ASM. Secara umum, kinerja sistem PDH yang telah dirancang sudah cukup baik. Walaupun demikian, khusus untuk modul *channel coding Reed-Solomon* beserta beberapa modul penunjangnya seperti modul *interleaver*, modul dual basis serta modul randomisasi dan ASM, masih terdapat beberapa aspek kinerja yang dapat ditingkatkan. Beberapa aspek tersebut antara lain dalam hal kecepatan data keluaran dan kebutuhan akan data *dummy* pada data keluaran. Pada sistem PDH tersebut, dibutuhkan 89 buah data *dummy* untuk melengkapi data keluaran sehingga kecepatan data keluaran hanya setengah dari kecepatan data masukan. Kedua hal tersebut dapat mempersulit proses *decoding* jika tidak menggunakan *decoder* yang sesuai dengan karakteristik *encoder* tersebut. Secara umum skematik sistem PDH yang telah dikembangkan di Pusat Teknologi Satelit yang berkaitan dengan koreksi *Reed-Solomon* dapat dilihat pada Gambar 2-2.

Pengurangan kecepatan data keluaran tersebut, dan juga kebutuhan akan data *dummy*, disebabkan karena proses *interleaving* dan proses *encoding*

dilakukan secara serial (berurutan) sehingga sebuah proses akan dilakukan setelah proses yang sebelumnya telah selesai. Salah satu contoh yang paling sederhana adalah pada skematik PDH tersebut, lima buah proses *encoding* terhadap lima paket data hasil *interleaving* dilakukan secara serial sehingga proses pembentukan data *parity* harus menunggu proses *encoding* kelima selesai. Padahal pembentukan data *parity* tersebut dapat dilakukan lebih awal, lebih tepatnya yaitu setelah proses *encoding* pertama menghasilkan data *parity*.

3 PERANCANGAN ENCODER

3.1 Perancangan Modul Encoder RS (255,223)

Seluruh operasi perhitungan pada skematik pembagian dalam *Galois Field* pada Gambar 2-1 dapat dengan mudah diimplementasikan menggunakan FPGA. Operasi penjumlahan dalam *Galois Field* misalnya dapat diimplementasikan dengan hanya menggunakan gerbang logika XOR. Sementara itu, operasi penundaan atau penyimpanan juga dapat diimplementasikan menggunakan gerbang *D-flipflop*, yang hanya berisi gerbang primitif seperti gerbang logika AND, OR dan NOT. Satu-satunya kendala dalam implementasi *hardware* pada skematik tersebut adalah operasi perkalian dalam *Galois Field*, dimana pada skematik tersebut, diperlukan beberapa buah gerbang perkalian terhadap sebuah konstanta yang berasal dari polinomial generator $G(x)$. Pada FPGA, operasi perkalian tersebut dapat dengan mudah direalisasikan dengan menggunakan modul LUT (*look-up*

table), karena modul LUT telah tersedia pada *library* FPGA. Walaupun demikian, metode LUT ini tidak efisien karena membutuhkan jumlah memori bit yang cukup banyak dan membutuhkan waktu pemrosesan dalam 2 siklus *clock*. Untuk menghasilkan *encoder* dengan kinerja yang optimal maka dibutuhkan sebuah modul perkalian dalam *Galois Field* yang hanya membutuhkan gerbang logika primitif yaitu gerbang AND, OR, NOT atau XOR.

Untuk merancang modul *encoder* RS(255,223) pada FPGA dibutuhkan perhitungan hasil perkalian dua variabel dalam *Galois Field*, dimana salah satu variabel perkalian tersebut adalah sebuah konstanta koefisien. Secara umum perkalian dalam *Galois Field* antara sebuah variabel A ($a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0$)

dengan sebuah konstanta B ($b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$) akan menghasilkan variabel baru C ($c_7, c_6, c_5, c_4, c_3, c_2, c_1, c_0$), dimana nilai bit c_i dapat ditentukan menggunakan operasi penjumlahan dalam *Galois Field* [Gambles, J.W., dan Winkert, T., 1993; Clarke, C.K.P., 2002]. Pada *encoder* RS(255,223) ini terdapat 32 buah gerbang perkalian, dan berkat rekomendasi CCSDS, *encoder* tersebut dapat lebih disederhanakan dan hanya membutuhkan 15 gerbang perkalian saja karena koefisien generator $G(x)$ bersifat simetris. Berdasarkan hasil perhitungan yang telah dilakukan, maka modul *encoder* RS(255,223) yang dirancang membutuhkan 15 gerbang perkalian dengan persamaan yang dapat dilihat pada Tabel 3-1.

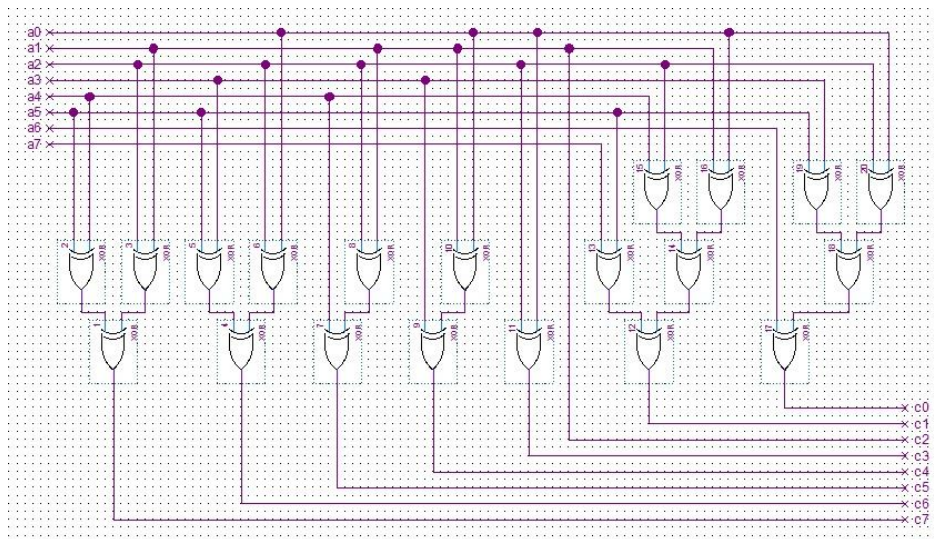
Tabel 3-1: GERBANG PERKALIAN PADA ENCODER RS(255,223)

Rangkaian Gerbang Perkalian XOR 8-bit untuk A($a_7a_6a_5a_4a_3a_2a_1a_0$)				
*Bit hasil perkalian $c_i=00110110$ melambangkan $c_i=XOR(a_5, a_4, a_2, a_1)$				
Gerbang 1/31	Gerbang 2/30	Gerbang 3/13/19/29	Gerbang 4/28	Gerbang 5/27
Konstanta 91	Konstanta 127	Konstanta 86	Konstanta 16	Konstanta 30
$c_7=00110110$	00101010	10100110	11111000	00101000
$c_6=00101101$	10111111	01110101	00000100	00111100
$c_5=00010110$	01011111	00111010	10000010	10011110
$c_4=00001011$	10101111	10011101	01000001	11001111
$c_3=00000101$	01010111	11001110	10100000	01100111
$c_2=00000010$	10101011	11100111	11010000	10110011
$c_1=10110111$	11111111	11010101	00010000	11110001
$c_0=01101101$	01010101	01001100	11110000	01010000
Gerbang 6/26	Gerbang 7/25	Gerbang 8/24	Gerbang 9/23	Gerbang 10/22
Konstanta 13	Konstanta 235	Konstanta 97	Konstanta 165	Konstanta 8
10010000	00001101	00000010	10100011	11110000
01011000	00001011	10000011	01110010	00001000
00101100	00000101	11000001	10111001	00000100
10010110	00000010	01100000	01011100	10000010
11001011	10000001	00110000	10101110	01000001
11100101	01000000	00011000	11010111	10100000
01100010	00101101	00001110	11001000	00100000
00100001	00011011	00000101	01000111	11100000
Gerbang 11/21	Gerbang 12/20	Gerbang 14/18	Gerbang 15/17	Gerbang 16
Konstanta 42	Konstanta 54	Konstanta 171	Konstanta 32	Konstanta 113
11001100	00100100	01110011	11111100	11111010
10101010	10110110	01001010	10000010	10000111
01010101	11011011	10100101	01000001	01000011
00101010	11101101	11010010	10100000	00100001
10010101	11110110	01101001	11010000	10010000
11001010	11111011	10110100	11101000	11001000
10101001	11011001	00101001	00001000	00011110
10011000	01001000	11100111	11111000	11110101

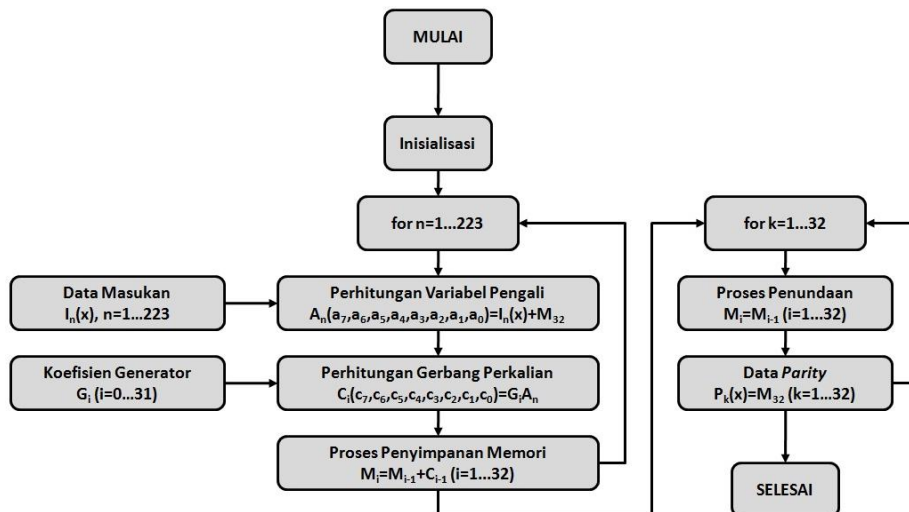
Sebagai contoh pada Tabel 3-1 tersebut, gerbang perkalian pertama yang merupakan perkalian variabel $A(a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$ dengan salah satu koefisien generator yang bernilai $G_1=91$, akan menghasilkan variabel hasil perkalian $C(c_7, c_6, c_5, c_4, c_3, c_2, c_1, c_0)$, dimana notasi $c_7=00110110$ merepresentasikan operasi $c_7=XOR(a_5, a_4, a_2, a_1)$, notasi $c_6=00101101$ merepresentasikan $c_6=XOR(a_5, a_3, a_2, a_0)$, dan demikian seterusnya. Skematik rangkaian untuk gerbang perkalian pertama tersebut ($G_1=91$) dapat dilihat pada Gambar 3-1, dengan catatan bahwa rangkaian logika tersebut belum mengalami optimasi rangkaian.

Dari Gambar 3-1, tampak bahwa gerbang perkalian tersebut membutuhkan 20 gerbang logika XOR 1-bit, sangat

efisien dibandingkan dengan penggunaan sebuah modul LUT (*Look-Up Table*) 256 byte. Jumlah gerbang tersebut masih dapat lebih disederhanakan dengan memanfaatkan kesamaan sebagian operasi yang terdapat pada beberapa gerbang perkalian lainnya. Walaupun demikian, perancangan FPGA pada penelitian ini tidak menggunakan diagram skematik rangkaian melainkan menggunakan VHDL, sehingga proses penyederhanaan rangkaian logika sulit dilakukan. Penggunaan VHDL tersebut dimaksudkan agar *encoder* dapat menghasilkan data keluaran dengan kecepatan yang lebih tinggi. Gambar 3-2 menunjukkan algoritma yang digunakan pada perancangan modul *encoder* RS(255,223) ini menggunakan VHDL.



Gambar 3-1: Skematik rangkaian gerbang perkalian $G_1=91$



Gambar 3-2: Algoritma perancangan modul *encoder* RS(255,223) menggunakan VHDL

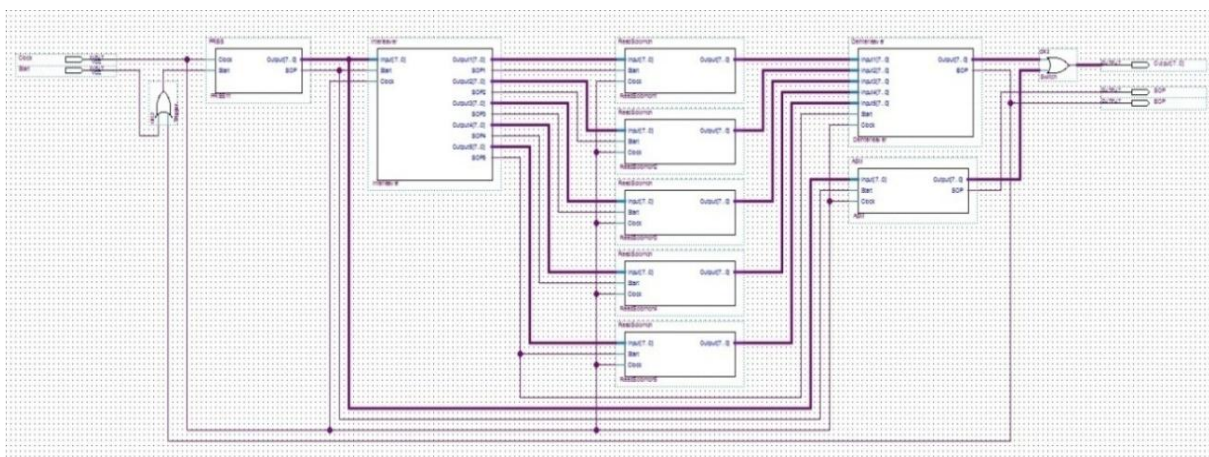
3.2 Perancangan Modul Encoder Paralel

Untuk mempercepat proses pembentukan data *parity* pada sistem PDH dengan teknik *interleave-5*, maka lima proses *encoding* sebaiknya dilakukan menggunakan lima buah *encoder* RS(255,223) secara terpisah. Dengan demikian, proses pembentukan data *parity* dapat dilakukan lebih awal sehingga tidak menimbulkan kondisi *idle* dalam proses pembentukan data keluaran. Secara umum, skematik rangkaian modul *encoder* paralel yang dikembangkan dapat dilihat pada Gambar 3-3.

Sesuai rekomendasi CCSDS, modul ini menggunakan *encoder* RS(255,223) dan modul *interleaver-5*. Karena proses *interleaving* akan membagi lima paket data menjadi lima paket baru, maka setiap modul *encoder* akan memproses setiap paket baru tersebut secara hampir bersamaan, pada penelitian digunakan perbedaan waktu sebesar 1 *clock*. Sebuah modifikasi terhadap modul *encoder* RS(255,223) yang telah dirancang adalah data *parity* akan dihasilkan setiap 5 *clock*. Hal tersebut dilakukan untuk mempermudah pembentuk data *parity* pada saat proses *de-interleaving*. Dengan pemrosesan secara paralel tersebut, sesaat setelah *encoder* pertama menghasilkan data

parity, data keluaran akan mencetak data tersebut dan dilanjutkan dengan mencetak data *parity* dari *encoder* kedua dan begitu seterusnya. Setelah menghasilkan lima buah data *parity*, selanjutnya data keluaran akan mencetak kembali data *parity* dari *encoder* pertama. Karena itulah data *parity* setiap *encoder* diatur sebesar 5 *clock*. Dengan demikian sistem keluaran dapat mencetak data keluaran secara berkesinambungan untuk setiap *clock*, mulai data informasi pertama $I_1(x)$ hingga data *parity* terakhir $P_{160}(x)$.

Optimasi rangkaian lainnya yang dilakukan yaitu mengenai operasi *dual* basis pada modul *interleaver* dan *deinterleaver*. Pada sistem PDH sebelumnya, operasi *dual* basis ini dilakukan dengan menggunakan modul LUT, sementara pada penelitian ini digunakan metode yang sama seperti pada skematik rangkaian gerbang perkalian pada Gambar 3-1. Penentuan gerbang logika yang dibutuhkan dalam operasi *dual* basis tersebut dapat diperoleh dari matriks *dual* basis yang telah direkomendasikan CCSDS [The Consultative Committee for Space Data Systems, 2011]. Tabel 3-2 menunjukkan hubungan matriks *dual* basis dan persamaan gerbang logika yang dibutuhkan.



Gambar 3-3: Diagram skematik rangkaian *encoder* menggunakan metode *encoding* paralel

Tabel 3-2: MATRIKS DUAL BASIS DAN RANGKAIAN LOGIKA YANG DIBUTUHKAN

Matriks Dual Basis	Gerbang Logika
<p>Modul interleaver C(c7,c6,c5,c4,c3,c2,c1,c0) A(a0,a1,a2,a3,a4,a5,a6,a7) C=AT⁻¹</p> $T_{al}^{-1} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$	<p>c7=XOR(a7,a6,a4,a3,a0) c6=XOR(a7,a5,a4,a3,a1,a0) c5=XOR(a6,a5,a4,a3,a2) c4=XOR(a5,a4,a3) c3=XOR(a7,a6,a5,a3,a2) c2=XOR(a7,a6,a3,a2,a0) c1=XOR(a2,a1) c0=XOR(a5,a3,a0)</p>
<p>Modul deinterleaver C(c0,c1,c2,c3,c4,c5,c6,c7) A(a7,a6,a5,a4,a3,a2,a1,a0) C=AT</p> $T_{al} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$	<p>c7=XOR(a7,a6,a2,a1,a0) c6=XOR(a6,a4,a3,a1,a0) c5=XOR(a7,a6,a5,a4,a1) c4=XOR(a7,a6,a5,a3,a2,a1,a0) c3=XOR(a3,a2,a0) c2=XOR(a6,a5,a3,a1,a0) c1=XOR(a6,a5,a3,a0) c0=XOR(a7,a6,a5,a4,a3,a2,a1)</p>

4 PENGUJIAN DAN ANALISIS

4.1 Pengujian Encoder RS (255,223)

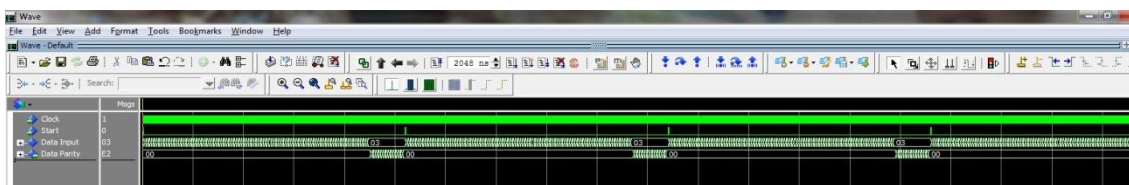
Pengujian untuk modul *encoder* RS(255,223) yang dirancang dilakukan dalam beberapa tahapan. Pengujian tahap pertama dilakukan dengan menggunakan *software* Matlab untuk memvalidasi algoritma koreksi *encoder* RS(255,223) yang dirancang. Hasil pengujian menunjukkan bahwa data keluaran yang dihasilkan menggunakan algoritma yang dirancang sama dengan hasil perhitungan Matlab. Setelah memvalidasi algoritma tersebut, maka algoritma tersebut diimplementasikan dalam rangkaian logika FPGA. Pada pengujian tahap kedua, terdapat dua buah pengujian yaitu pengujian menggunakan simulator FPGA ModelSim dan pengujian fisik modul FPGA menggunakan *signal tap* Quartus. Setelah pengujian pada simulator FPGA ModelSim berhasil, baru dilakukan pengujian fisik modul FPGA. Pengujian

simulator ini dilakukan untuk meminimalisasi jumlah pengujian fisik agar mengurangi penggunaan modul FPGA. Kedua hasil pengujian yang dilakukan tersebut, baik menggunakan simulator maupun pengujian fisik, menghasilkan data keluaran yang akurat, sesuai dengan hasil perhitungan pada Matlab. Jadi akurasi data keluaran yang dihasilkan *encoder* RS(255,223) yang dirancang dalam penelitian ini sangat baik, sesuai dengan *encoder* RS(255,223) yang direkomendasikan CCSDS.

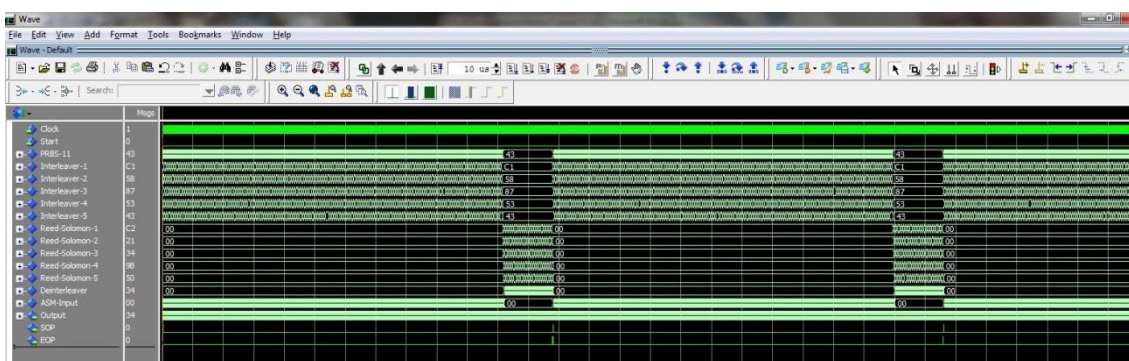
Selanjutnya akan dibandingkan efisiensi dari penggunaan gerbang logika FPGA, antara rangkaian *encoder* yang dirancang dengan rangkaian modul *IP-core encoder* RS(255,223) dari Altera. Beberapa aspek yang dibandingkan adalah jumlah gerbang logika dasar dan jumlah memori yang digunakan. Tabel 4-1 menunjukkan hasil perbandingan tersebut.

Tabel 4-1: PERBANDINGAN EFISIENSI GERBANG LOGIKA RS(255,223)

Rangkaian	Hasil Perancangan	Modul <i>IP-core</i>
Logic Combinatorial	303	599
Logic Register	273	473
Memori Bit	0	0



Gambar 4-1: Timing diagram encoder RS(255,223) yang dirancang



Gambar 4-2: Timing diagram encoder dengan metode encoding paralel

Tampak bahwa *encoder* yang dirancang memiliki jumlah gerbang logika yang lebih sedikit, walaupun tidak terlalu berbeda jauh. Hal tersebut dikarenakan *encoder* ini bersifat *special-purpose*, dimana *encoder* ini khusus dirancang berdasarkan parameter CCSDS, sementara *IP-core encoder* RS(255,223) Altera bersifat umum dan fleksibel untuk bermacam aplikasi, sehingga penggunaan gerbang logika pada *encoder* yang dirancang ini dapat diminimalisasi.

Setelah memastikan akurasi data keluaran yang dihasilkan dan gerbang logika yang digunakan cukup efisien, selanjutnya dilakukan analisis kinerja untuk data keluaran *encoder* yang telah dirancang, yang dapat dilihat pada Gambar 4-1. Data masukan yang digunakan pada pengujian ini dihasilkan oleh modul PRBS-11, sebuah algoritma yang dapat menghasilkan data *pseudo-random*.

Walaupun tidak dapat terlihat dengan jelas, *encoder* yang dirancang

hanya membutuhkan waktu tunda (*delay*) selama 2 siklus *clock* antara data masukan dengan data keluaran (*parity*), sementara itu *IP-core encoder* RS Altera membutuhkan *delay* sebesar 4 siklus *clock* [Rosa, W., Amin, D.E., dan Nasser, E.N., 2014]. Waktu tunda ini memang tidak terlalu signifikan mengingat satu siklus *clock* berada pada orde milidetik, tetapi hasil tersebut menunjukkan bahwa *encoder* yang dirancang dapat menghasilkan kinerja yang lebih optimal.

4.2 Pengujian Modul *Encoder* Paralel

Pada dasarnya, *encoder* yang dirancang bertujuan untuk mempercepat kecepatan data keluaran dan menghilangkan data *dummy* dari data keluaran yang akan disimpan atau ditransmisikan. Hasil pengujian yang dilakukan menunjukkan bahwa kedua hal tersebut dapat tercapai. *Timing diagram* untuk *encoder* dengan menggunakan metode *encoding* paralel tersebut dapat dilihat pada Gambar 4-2.

Tabel 4-2: PERBANDINGAN KEBUTUHAN GERBANG LOGIKA SISTEM PDH

Rangkaian	Hasil Perancangan	Sistem PDH Awal
Logic Combinatorial	3229	1275
Logic Register	1603	800
Memori Bit	0	92160

Dari hasil pengujian tersebut tampak bahwa *encoder* yang dirancang tidak membutuhkan data *dummy* pada data keluaran yang dihasilkan, sedangkan pada sistem PDH sebelumnya [Rosa, W., Amin, D.E., dan Nasser, E.N., 2014] dibutuhkan 89 buah data *dummy* untuk melengkapi data keluaran. Selain itu *encoder* dengan metode *encoding* paralel ini dapat menghasilkan data keluaran untuk setiap *clock*, yang menunjukkan bahwa *encoder* yang dirancang dapat mengikuti kecepatan data masukan. Jadi seluruh 1279 *byte* data keluaran, yang terdiri dari 4 *byte* data ASM, 1115 *byte* data masukan dan 160 *byte* data *parity*, dapat dihasilkan dalam 1279 siklus *clock*. Kedua hal tersebut, yaitu kecepatan data keluaran dan tidak dibutuhkannya data *dummy* pada data keluaran, sangat penting pada saat proses *decoding* jika *decoder* yang digunakan adalah *decoder* generik dan bukan *decoder* yang khusus didedikasikan untuk *encoder* ini.

Gerbang logika kombinatorial yang digunakan pada *encoder* ini lebih banyak dibandingkan *encoder* pada sistem PDH sebelumnya, tetapi *encoder* ini tidak membutuhkan penggunaan memori bit. Jumlah gerbang logika yang lebih banyak tersebut disebabkan karena *encoder* dengan metode *encoding* paralel ini membutuhkan lima buah *encoder* RS. Perbandingan gerbang logika yang dibutuhkan dapat dilihat pada Tabel 4-2. Sebagai catatan, perbandingan *encoder* paralel tersebut belum menggunakan modul *randomizer*.

5 KESIMPULAN DAN SARAN

Kegiatan penelitian ini berhasil merancang sebuah modul *encoder* RS(255,223) untuk keperluan PDH

(*Payload Data Handling*) satelit dengan kinerja yang lebih baik dibandingkan dengan *IP-core encoder* RS komersial. Gerbang logika yang digunakan juga lebih efisien karena tidak membutuhkan fitur-fitur umum yang dimiliki oleh *encoder* komersial tersebut. Selain itu, *encoder* yang dirancang berdasarkan konsep *encoding* paralel juga dapat menghasilkan kinerja yang lebih baik dibandingkan *encoder* pada sistem PDH saat ini, terutama dalam hal kecepatan data keluaran dan tidak membutuhkan data *dummy* untuk melengkapi data keluaran. Walaupun demikian, sistem PDH dengan konsep *encoding* paralel membutuhkan gerbang logika yang lebih banyak dibandingkan sistem PDH sebelumnya, karena menggunakan lima *encoder* yang bekerja secara paralel.

Selanjutnya diperlukan penelitian untuk dapat lebih meminimalisasi penggunaan gerbang logika pada *encoder* RS(255,223), karena masih terdapat beberapa penyederhanaan yang dapat dilakukan. Walaupun demikian, semakin sedikit jumlah gerbang logika yang dibutuhkan biasanya akan semakin menambah waktu pemrosesan, sehingga faktor *trade-off* tersebut perlu diperhatikan dengan seksama.

DAFTAR RUJUKAN

- Clarke, C.K.P., 2002. *R&D White Paper: Reed-Solomon Error Correction*, British Broadcasting Corporation (BBC).
- Dwiyanto, dan Sugihartono, 2011. *Simulasi Kinerja Forward Error Control Coding untuk Satelit Mikro Penginderaan Jarak Jauh*, Jurnal Teknologi Dirgantara Vol. 9 No. 2, LAPAN, hal. 82-92.

- Gambles, J.W., dan Winkert, T., 1993. *Space Qualified High Speed Reed-Solomon Encoder*, 5th NASA Symposium on VLSI Design, hal. 3.1.1-3.1.9.
- Pollara, F., 1985. *A Software Simulation Study of a (255,223) Reed-Solomon Encoder/Decoder*, Jet Propulsion Laboratory, California.
- Pratomo, B., 2012. *Pengkodean Reed-Solomon Menggunakan FPGA Secara Simulasi*, Prosiding SIPTEKGAN XVI-2012, LAPAN, hal. 76-83.
- Rosa, W., Amin, D.E., dan Nasser, E.N., 2014. *Design and Implementation of Payload Data Handling Based on Field Programmable Gate Array*, Prosiding ICARES, Yogyakarta, hal. 48-54.
- Sklar, B., 2001. *Digital Communications: Fundamentals and Applications 2nd Edition*, Prentice Hall, ISBN-13: 978-0-13-084788-1.
- The Consultative Committee for Space Data Systems, 2005. *Image Data Compression*, Blue Book, CCSDS 122.0-B-1.
- The Consultative Committee for Space Data Systems, 2011. *TM Synchronization and Channel Coding*, Blue Book, CCSDS 131.0-B-2.
- The Consultative Committee for Space Data Systems, 2012. *CCSDS Cryptographic Algorithms*, Blue Book, CCSDS 352.0-B-1.