

Pendekodean Kanal *Reed Solomon* Berbasis FPGA Untuk Transmisi Citra pada Sistem Komunikasi Satelit Nano

Andi Nurul Utami Husain, Gamantyo Hendrantoro, dan Suwadi

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: suwadi@ee.its.ac.id

Abstrak—Sistem komunikasi satelit saat ini sedang dikembangkan oleh mahasiswa-mahasiswi ITS, salah satunya yaitu Sistem komunikasi satelit nano 2,4 GHz untuk pengiriman citra. Untuk mengatasi kerusakan informasi yang diterima akibat noise selama transmisi, maka di stasiun bumi dibutuhkan pendekodean kanal *Reed Solomon* (255,239) untuk mendeteksi dan mengoreksi informasi jika terjadi kesalahan pada informasinya. Dalam makalah ini bertujuan untuk merancang pendekodean kanal *Reed Solomon* (255,239) untuk diimplementasikan pada stasiun bumi. Pendekodean ini secara teori mampu mendeteksi dan mengoreksi maksimum 8 simbol *error* yang terjadi pada informasi. Pendekodean *Reed Solomon* (255,239) dirancang menggunakan bahasa pemrograman VHDL (*VHSIC Hardware Description Language*) dan dimasukkan kedalam board FPGA (*Field-Programmable Gate Array*). Pengujiannya dilakukan hanya secara simulasi dengan menginjeksikan *error* pada data informasi yang diterima oleh *Reed Solomon* (255,239) karena relatif sulit diuji dengan menambahkan *error* saat tahap implementasi. Secara simulasi pendekodean ini telah mampu mengoreksi sebanyak 8 simbol *error* sesuai dengan teori. Pada tahap implementasinya juga telah berhasil karena telah mampu menerima informasi yang sesuai dengan informasi yang dikirim dengan asumsi tidak terdapat *error* pada informasi.

Kata Kunci— *Reed Solomon* (255,239), FPGA, citra, VHDL

I. PENDAHULUAN

TRANSMISI satelit merupakan metode transmisi yang sedang dikembangkan oleh mahasiswa-mahasiswi perguruan tinggi di Indonesia termasuk ITS (Institut Teknologi Sepuluh Nopember). Saat ini mahasiswa-mahasiswi ITS sedang dalam penelitian mengembangkan sistem komunikasi satelit nano dalam pengiriman citra/gambar, dimana kamera pada *payload* satelit akan mengambil gambar/citra selama satelit mengorbit kemudian ditransmisikan ke stasiun bumi. Sistem komunikasinya dibatasi pada arah downlink saja dan beroperasi pada frekuensi 2,4 GHz.

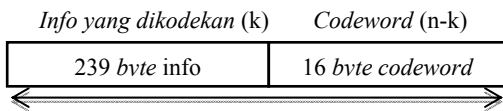
Citra/gambar merupakan komponen multimedia dengan bentuk informasi visual. Selama penransmisian, citra sering mengalami penurunan mutu (*degradasi*) akibat *noise* [1]. Akibatnya citra yang diterima oleh stasiun bumi lebih sulit untuk diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah untuk diinterpretasi, maka perlu untuk meningkatkan performansi kanal. Untuk meningkatkan performansi kanal diperlukan pengkodean kanal (*channel*

coding) di sisi satelit dan pendekodean kanal (*channel decoding*) di stasiun bumi.

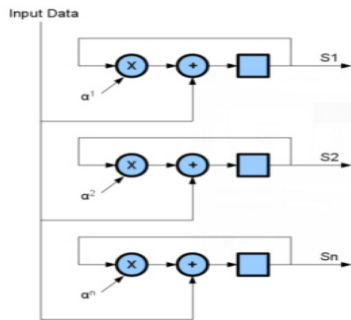
Pengkodean kanal (*channel coding*) berfungsi untuk menjaga informasi atau data digital dari *error* yang mungkin terjadi selama proses transmisi dengan cara menambahkan bit tambahan (*redundancy*) ke dalam data yang akan dikirimkan. Kemudian pada sisi penerima, data tersebut didekodekan (*channel decoding*) untuk mendeteksi dan mengoreksi *error* sehingga menghasilkan output yang mendekati citra asal [2]. *Channel code* yang digunakan yaitu *Reed Solomon*(255,239), dimana pendekodean tersebut mampu mendeteksi dan mengoreksi 8 simbol *error* yang bisa saja muncul secara acak dan tak terduga. Karena hal tersebut maka *Reed Solomon code* merupakan salahsatu *channel code* yang direkomendasikan oleh CCSDS (*Consultative Committee for Space Data Systems*) [3].

Untuk memproses pengkodean kanal diperlukan perangkat keras (*hardware*) sebagai medianya. Perangkat keras yang digunakan yaitu FPGA (*Field Programmable Gate Array*). Pendekodean *Reed Solomon* ini akan diimplementasikan pada sebuah FPGA (*Field Programmable Gate Array*) dengan menggunakan bahasa pemrograman VHDL (*VHSIC Hardware Description Language*) yang memanfaatkan *tools* dari *Quartus II*. Pemilihan FPGA disebabkan FPGA mampu beroperasi pada frekuensi yang semakin tinggi, memiliki memori yang cukup besar sebagai penyimpanan citra. Untuk merepresentasikan fungsi *hardware* pada FPGA maka digunakan bahasa VHDL dalam merencanakan suatu desain atau sistem tertentu. Makalah ini membahas mengenai perancangan pendekodean *Reed Solomon* (255,239) untuk transmisi citra pada satelit nano 2,4 Ghz sedangkan pembahasan mengenai pengkodean *Reed Solomon* (255,239) terdapat pada makalah lain [4].

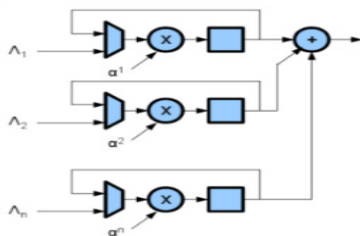
Sistematika penulisan pada nakalah ini terdiri dari pendahuluan yang menguraikan hal-hal yang melatarbelakangi judul makalah. Selanjutnya pada bagian II menguraikan teori dasar yang menunjang makalah ini yaitu, karakteristik citra, prinsip kerja dari algoritma pendekodean *Reed Solomon*, dan karakteristik FPGA. Pada bagian III menjelaskan tahap-tahap perancangan yang dilakukan, dan implementasi ke *hardware*. Pada bagian IV berisi hasil pengujian dan analisis kinerja program Pendekodean *Reed Solomon* (255,239) dengan melakukan simulasi dan secara implementasi dengan memberikan informasi berupa data teks dan data citra. Pada bagian V berisi pembahasan hasil pengujian dan analisis yang telah dilakukan serta menarik kesimpulan dari pembahasan tersebut..



Gambar. 1. Elemen kode Reed Solomon



Gambar. 2. Skema sindrom error



Gambar. 3. Skema Chien Search

II. TINJAUAN PUSTAKA

A. Citra Digital

Citra Digital memiliki beberapa format yang memiliki karakteristik tersendiri. Format pada citra digital ini umumnya berdasarkan tipe dan cara kompresi yang digunakan pada citra digital tersebut. Salah satu format citra digital yaitu format *Joint Photographic Expert Group* (JPEG/JPG) merupakan format gambar yang berukuran kecil karena sudah terkompresi dengan dimensi yang sama dengan format BMP.

Waktu pengiriman gambar dapat dihitung dengan menggunakan persamaan (1).

$$t \text{ (detik)} = \frac{\text{ukuran file (byte)} \times 8 \text{ bit}}{\text{baudrate (bps)}} \tag{1}$$

B. Reed Solomon

Kode *Reed Solomon* adalah sebuah kode yang dirumuskan oleh *Irving Reed* dan *Gus Solomon* pada tahun 1960. Kode *Reed Solomon* mendeskripsikan sebuah cara sistematis untuk membentuk sebuah kode yang mampu mengkoreksi *error* yang muncul secara acak dan tak terduga (*bursty*) pada paket data yang diterima [2].

Sebuah kode *Reed Solomon* ditulis dalam bentuk $RS(n,k)$ dimana n adalah panjang blok atau panjang kode yang terdiri dari susunan beberapa simbol, sedangkan k adalah panjang informasi atau jumlah simbol data yang akan dikodekan. Panjang *block code* ini dinyatakan oleh $n = 2^m - 1$ dimana m adalah jumlah bit per simbol sedangkan jumlah simbol *parity* yang harus ditambahkan untuk mengkoreksi sejumlah *error* sebanyak $n - k = 2t$ dimana t adalah jumlah simbol *error* yang mampu dikoreksi.

C. Pendekodean Reed Solomon

Sebuah RS (n,k) mampu mendeteksi *error* sebesar t , jika terdapat RS $(255,239)$ maka jumlah *error* yang mampu dikoreksi yaitu 8 simbol. Pendekodean *Reed Solomon* mengganti simbol yang salah dengan simbol yang sebenarnya tanpa memperdulikan apakah *error* yang terjadi pada simbol tersebut disebabkan oleh satu *bit* yang rusak atau semua *bit* pada simbol tersebut mengalami kerusakan. Alasan inilah yang menyebabkan kode *Reed Solomon* dianggap lebih baik dibanding *binary codes* [2]. Proses pendekodean *Reed Solomon* dijalankan dengan mencari sindrom *error* pada data informasi.

1. Mencari sindrom *error*
2. Mencari polinomial *error*, yaitu $\Omega(x)$ dan $A(x)$ untuk digunakan pada perhitungan besaran *error* (*error magnitude*) dengan algoritma *Berlekamp-Massey*.
3. Mencari lokasi *error* dengan algoritma *Chien*.
4. Mencari nilai/besaran *error* pada data pada lokasi yang telah diperoleh sebelumnya dengan menggunakan algoritma *Forney*
5. Melakukan proses penjumlahan XOR antara nilai *error* yang diperoleh pada prosedur 4 dengan data pada lokasi *error*. Hal ini akan membuat data kembali seperti semula.

Sindrom Error

Pencarian sindrom *error* untuk menentukan apakah paket data yang diterima oleh *decoder* mengalami kerusakan atau tidak. Hasil pencarian sindrom akan menentukan Polinomial sindrom. Nilai sindrom ditentukan dengan menggunakan persamaan (2). Polinomial sindrom *error* secara umum direpresentasikan sebagai [5]:

$$S(n) = \sum_{i=0}^{2t-1} S_i \tag{2}$$

$$S(x) = S_0 + S_1x + \dots + S_{2t-1}x^{2t-1} \tag{3}$$

Dimana jika $S(x) = 0$, maka pada data yang diterima tidak terdapat *error*. Jika data mengalami *error* maka elemen sindrom *error* akan menghasilkan nilai tertentu. Pada RS (n,k) akan terdapat sejumlah $2t$ sindrom, maka sebuah RS(255,239) memiliki 16 elemen sindrom dari $S(0), S(1), \dots, S(15)$.

Polinomial Error

Polinomial *error* digunakan untuk mencari polinomial lokasi *error* dan *error magnitude* pada data yang diterima. Untuk mencari Polinomial *error* digunakan algoritma *Berlekamp-Massey* atau *Euclidean*. Polinomial *error* (lokasi *error* dan *error magnitude*) ditentukan dengan menggunakan Polinomial sindrom yang didapatkan sebelumnya. Polinomial *error* direpresentasikan dengan [5]

$$A(x)S(x) = \Omega(x) \text{ mod } x^{2t} \tag{4}$$

Algoritma yang digunakan pada RS $(255,239)$ ini yaitu *Berlekamp-Massey*. Algoritma ini dipilih karena menggunakan teknik iterasi yang lebih efisien digunakan jika *encoder*-nya menggunakan teknik LFSR [6].

Lokasi Error

Lokasi *error* dapat di peroleh menggunakan Polinomial $\Lambda(x)$ yang telah diperoleh pada tahap sebelumnya. Pencarian lokasi *error* ini menggunakan metode *chien search* dimana skemanya ditunjukkan pada Gambar 3.

Untuk menentukan lokasi *error* diperlukan nilai *inverse* dari *galois field* (2^m), maka untuk RS (255,239) dibutuhkan nilai *inverse* dari *galois field* (256), setelah itu mensubstitusikan satu persatu variabel x pada $A(x)$ dengan nilai *inverse* yang didapatkan sebelumnya. Jika hasilnya memberikan nilai nol (0), maka menunjukkan bahwa data posisi yang menghasilkan nilai nol (0) memiliki *error*. Jika hasilnya memberikan nilai bukan nol ($\neq 0$) maka data pada lokasi tersebut tidak terdapat *error* [6].

Besaran Error

Setelah lokasi *error* diperoleh, tahap selanjutnya adalah mencari besar *error* (*error magnitude*) pada data yang diterima dengan algoritma *Forney*. Untuk mencari *error magnitude* digunakan persamaan (5).

$$Y_j = X_j^{j-1} \frac{b^{\Omega(X_j^{-1})}}{\Lambda(X_j^{-1})} \quad (5)$$

Polinomial *error* yang didapatkan pada prosedur 2 akan digunakan pada rumus mencari besaran *error* yang telah diperlihatkan diatas. Nilai turunan diperoleh dilakukan dengan membuat pangkat genap menjadi nol, kemudian substitusikan x menjadi X_j^{-1} dan membagi persamaan dengan X_j^{-1} . X_j^{-1} nantinya akan disubstitusikan dengan bentuk index dari lokasi *error* yang telah didapatkan sebelumnya [6].

Koreksi Error

Koreksi *error* dapat dilakukan dengan melakukan penjumlahan XOR antara data pada lokasi *error* dengan besar *error* yang diperoleh menggunakan algoritma *Forney* [6]. Hasil dari penjumlahan tersebut akan menghasilkan data yang sebenarnya. Agar *Reed Solomon* (255,239) decoder dapat mendeteksi *error* dengan sempurna diperlukan FIFO buffer sebagai penyimpanan data informasi yang diterima secara sementara [7].

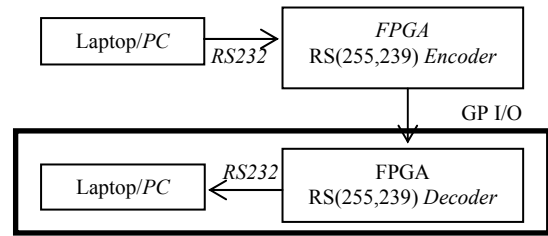
D. Field Programmable Gate Array (FPGA)

FPGA merupakan rangkaian *combinational* dan *sequential* yang dikombinasikan sedemikian rupa untuk bisa diimplementasikan langsung pada *silicon*. FPGA ini memiliki *performance* tinggi dan memiliki fungsi yang spesifik. Fungsi yang spesifik berarti hasil rangkaian yang dibuat hanya melakukan satu proses khusus, contoh : jika *user* membuat rangkaian *Full Adder*, maka rangkaian ini hanya dapat melakukan penjumlahan saja. Karakteristik FPGA yaitu memiliki gerbang logika yang berjumlah hingga jutaan gerbang logika. Karena hal tersebutlah maka FPGA dinyatakan sangat cocok untuk aplikasi sistem digital. Selain itu FPGA juga memiliki memori yang cukup besar. Bahasa program *default* dari FPGA yaitu bahasa pemrograman VHDL (*VHSIC Hardware Description Language*) [8].

III. PERANCANGAN DAN IMPLEMENTASI

A. Penentuan Perangkat Keras dan Lunak

Jenis FPGA yang digunakan adalah *Altera Cyclone II* (EP2C20F484C7) atau biasanya disebut *Altera DE1* yang merupakan keluarga FPGA keluarga *Altera* yang dikembangkan oleh perusahaan dengan nama yang sama. Pemilihan FPGA *DE1* ini karena memenuhi spesifikasi-spesifikasi yang dibutuhkan untuk implementasinya.



Gambar. 4. Diagram blok pengkodean kanal

FPGA jenis ini dilengkapi *interface* yang dibutuhkan sesuai dengan implementasinya yaitu *interface RS232*, memiliki banyak memori yaitu SDRAM (8 MB), SRAM (512 KB), Flash (4 MB) dengan kapasitas yang cukup besar untuk ukuran citra ± 500 Kbyte. Selain itu, harga untuk FPGA jenis ini juga relatif lebih murah dibanding FPGA jenis lainnya. Software yang digunakan oleh FPGA keluarga *Altera* untuk membuat program yaitu *software Quartus*. Untuk proses simulasi digunakan *software simulator* yang juga khusus digunakan oleh FPGA keluarga *Altera* yaitu *Qsim* dan *Modelsim Altera* [9].

B. Perancangan Sistem

Rancangan sistem untuk bagian pengkodean-pendekodean kanal *Reed Solomon* (255,239) sendiri dapat dilihat pada gambar 4. Pada makalah ini dibatasi pada perancangan Pendekodean (*decoder*) *Reed Solomon* (255,239) saja dan data informasi citra diambil tidak melalui kamera melainkan dari laptop/PC.

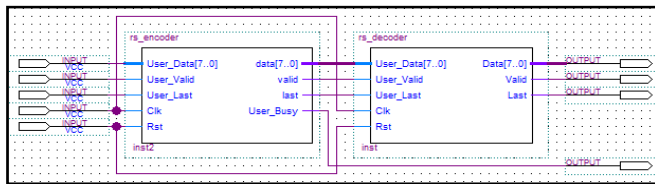
Berdasarkan Gambar 4, maka tiap-tiap blok dapat dijelaskan sebagai berikut:

FPGA (*decoder*), blok ini berfungsi untuk mendeteksi dan mengkoreksi apabila terjadi *error* pada data informasi yang diterima. Oleh karena itu blok ini berisi program pendekodean (*decoder*) RS (255,239). Berdasarkan dari teori dasar *Reed Solomon* bahwa *Input* dan *outputnya* adalah data paralel 8 bit, sedangkan *input RS232* merupakan data serial maka *output decoder* harus diubah terlebih dahulu menjadi data serial sehingga data informasi dapat dikirim oleh RS232 ke penerima.

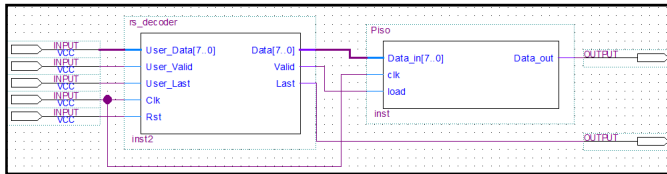
Laptop/PC, blok ini berfungsi untuk menerima data informasi baik berupa data teks ataupun citra melalui *interface* USB-RS232 dari *board* FPGA (*decoder*). Proses penerimaan data informasi pada laptop/PC dilakukan melalui terminal *emulator*, dimana terminal ini akan menampilkan dan menyimpan data informasi yang diterima pada hardisk laptop/PC.

Dari penjelasan blok diatas, maka perancangan RS (255,239) *decoder* secara keseluruhan membutuhkan 4 tahap. Tahap pertama yaitu memprogram pendekodean RS (255,239) sesuai dengan tahap-tahapnya. Setelah itu disimulasikan dengan menggunakan simulator *Qsim* untuk memastikan keluarannya sesuai yang diharapkan dengan asumsi tidak terjadi *error* pada data informasi yang diterima oleh *decoder*. Dalam tahap ini simulasi *decoder* dilakukan terpisah dengan *encoder* sehingga keberhasilan sistem pengkodean kanal RS (255,239) belum terlihat.

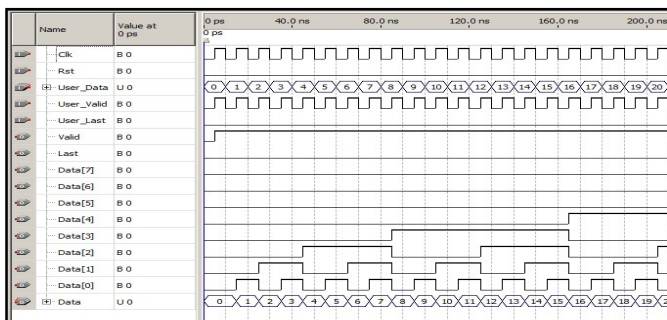
Tahap kedua yaitu program RS (255,239) *encoder* dan *decoder* digabung dalam satu *project* untuk melihat hasil kinerjanya telah sesuai atau tidak. *Encoder* dan *decoder* sebelumnya dibuatkan skematiknya atau biasa disebut chip. Kemudian skematik tersebut dihubungkan satu sma lain. Blok diagram skematiknya dapat dilihat pada Gambar 5.



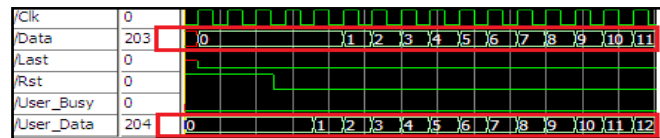
Gambar. 5. Blok diagram skematik Reed Solomon (255,239)



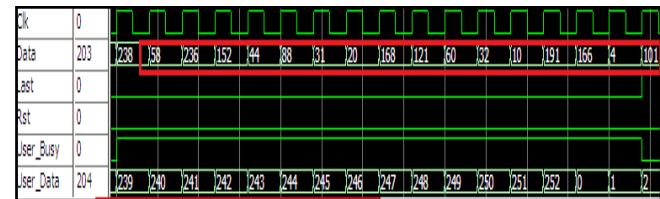
Gambar. 6. Blok diagram skematik program Decoder-PISO



Gambar. 7. Simulasi tanpa error



Gambar. 8. Simulasi keluaran Reed Solomon (255,239) encoder



Gambar. 9. 16 byte tambahan pada simulasi RS (255,239) encoder

Tabel 1. Deskripsi hasil simulasi tanpa error

Nama	Polaritas	Deskripsi
Clk		Clock dengan periode 10 ns
Rst	Low	Terjadi reset jika diberi input 1
Data		Data informasi yang dikeluarkan oleh decoder sesuai dengan data informasi yang dikirim.
Valid	High	Semua informasi akan diterima oleh decoder
Last	Low	Menghentikan pengiriman informasi jika diberi bit 1, jika bit 0 maka informasi akan terus dikirim.

Setelah menggabungkan dan disimulasikan dengan menggunakan simulator Modelsim Altera, maka encoder dan decoder dipisahkan kembali karena nantinya akan diletakkan pada board FPGA yang berbeda. Penggabungan project ini hanya untuk kebutuhan pengujian secara simulasi yang dapat

membuktikan bahwa RS (255,239) decoder ini mampu mendeteksi dan mengoreksi error.

Tahap ketiga yaitu membuat program pengubah data paralel 8 bit menjadi data serial agar bisa dilewatkan oleh RS232 kemudian mensimulasikannya. Setelah tahap-tahap sebelumnya berhasil maka tahap selanjutnya yaitu menggabungkan program Pendekodean RS (255,239) dengan program Paralel 8bit to serial, kemudian mensimulasikannya untuk memastikan keluaran sistem sesuai dengan yang diharapkan. Setelah proses penggabungan sukses, maka untuk kebutuhan implementasi pada FPGA DE1 diperlukan penetapan pin I/O. Tiap-tiap jenis FPGA memiliki komponen I/O yang berbeda-beda dan nama-nama pinnya pun berbeda-beda tergantung jenis FPGAny.

C. Implementasi Sistem

Implementasinya dilakukan bersamaan dengan Pengkodean kanal RS (255,239). Implementasi dilakukan dengan mendownload project program pendekodean kanal RS (255,239) kedalam board FPGA DE1 melalui USB-Blaster. Pada sisi pengkodeannya juga diberikan perlakuan yang sama. Implementasi ini mengacu pada gambar 4. Setelah mengatur sesuai dengan Gambar 4, maka dilakukan implementasi dengan memanfaatkan terminal emulator untuk mengirim dan menerima informasi.

IV. PENGUJIAN DAN ANALISIS DATA

A. Simulasi Pendekodean RS (255,239) Tanpa Error

Simulasi Pendekodean Reed Solomon (255,239) tanpa error ini dijalankan oleh simulator Qsim dimana semua inputnya dimasukkan secara manual dan tidak berpengaruh terhadap waktu. Gambar 7 memperlihatkan hasil simulasinya dengan penjelasan pada Tabel 1.

B. Simulasi Pendekodean Reed Solomo (255,239) dengan 8 Error

Simulasi ini dijalankan dengan metode “testbench” yang khusus digunakan untuk project FPGA. Testbench yaitu suatu metode simulasi yang dituliskan dalam bahasa pemrograman VHDL. Pada testbench inilah diinjeksikan 8 error pada input decoder sehingga kemampuan mendeteksi dan mengoreksi error dapat terlihat pada hasil simulasinya. Untuk melihat sinyal-sinyal dari simulasi tetbench tersebut, maka dibutuhkan simulator Modelsim-Altera. Simulasi ini diberi sinyal clock dengan periode 20ns, informasi yang diberikan sebanyak 1249 byte, dan dijalankan oleh simulator dengan periode waktu selama 33×10^{-5} detik.

Sebelum menunjukkan hasil simulasi tahap-tahap pada pendekodean RS (255,239), akan diperlihatkan terlebih dahulu hasil simulasi keluaran dari Pengkodean RS (255,239) yang merupakan data informasi sebenarnya. Hasil simulasi menunjukkan bahwa data informasi yang dikeluarkan oleh encoder berupa bilangan desimal 0-238 secara berurut kemudian ditambahkan 16 byte yang berfungsi sebagai codeword, sehingga jumlah keseluruhannya yaitu 255 byte. Codeword yang ditambahkan pada data informasi yaitu 58, 236, 152, 44, 88, 31, 20, 168, 121, 60, 32, 10, 191, 166, 4, 101.

Locations	{25...}	{254}	{253}	{252}	{99}	{29}	{19}	{9}	{0}
-----------	---------	-------	-------	-------	------	------	------	-----	-----

Gambar. 10. Simulasi pencarian *error*

Errors	{39...}	{49}	{17}	{85}	{152}	{69}	{255}	{44}	{77}
Locations	{25...}	{254}	{253}	{252}	{99}	{29}	{19}	{9}	{0}

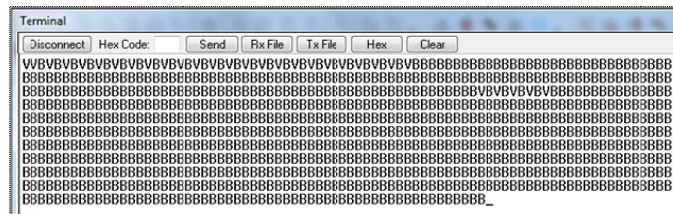
Gambar. 11. Simulasi *magnitude error*

D	203	49	16	87	3	
Data_O	203	0	1	2	3	
/correct_data	197	0	49	16	87	3

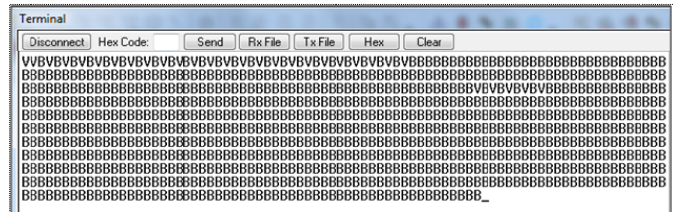
Gambar 12. Simulasi koreksi *error*

ne/Sy...	{76}	{185}	{63}	{130}	{109}	{169}	{202}	{3}	{90}	{80}	{8}	{60}	{238}	{94}	{157}	{169}	{235}
Errors	{255...}	{150}	{255}	{255}	{255}	{255}	{255}	{255}	{255}	{255}							
Locat...	{255...}	{167}	{255}	{255}	{255}	{255}	{255}	{255}	{255}								

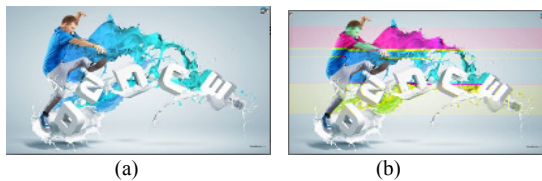
Gambar. 13. Simulasi lokasi *error* dan besaran *error*



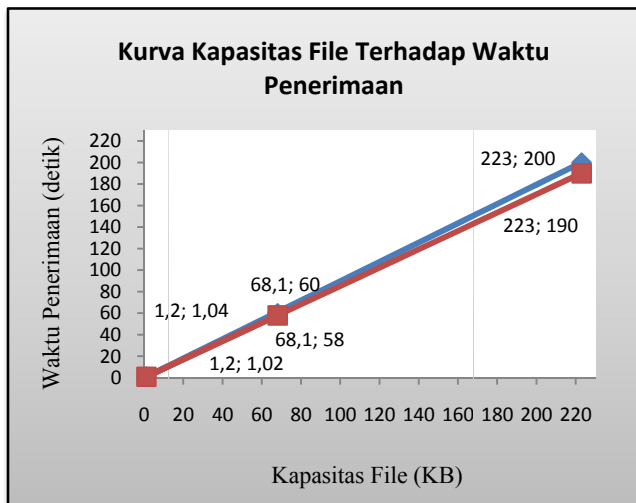
Gambar. 14. Karakter ASCII yang dikirim



Gambar .15. Karakter ASCII yang diterima



Gambar. 16. Citra berkapasitas 68,1 KB, (a) yang dikirim ; (b) yang diterima



Gambar. 17. Kurva kapasitas file terhadap waktu penerimaan

Tabel 2.
Kinerja Pendekodean *Reed solomon* (255,239)

Data O	D	Besaran error	Correct data	Rcnt
0	49	49	0	254
1	16	17	1	253
2	87	85	2	252
155	3	152	155	99
225	164	69	225	29
235	20	255	235	19
20	56	44	20	9
101	40	77	101	0

Sindrom Error

Hasil simulasi perhitungan sindrom yang ditunjukkan pada gambar diatas mengindikasikan terdapat *error* pada data informasi yang diterima. Perhitungan sindrom yang didapatkan yaitu :

$$\begin{aligned}
 S_0 &= 71 & S_8 &= 128 \\
 S_1 &= 160 & S_9 &= 171 \\
 S_2 &= 78 & S_{10} &= 41 \\
 S_3 &= 2 & S_{11} &= 84 \\
 S_4 &= 137 & S_{12} &= 37 \\
 S_5 &= 162 & S_{13} &= 66 \\
 S_6 &= 226 & S_{14} &= 237 \\
 S_7 &= 38 & S_{15} &= 33
 \end{aligned}$$

Polinomial error

Pencarian polinomial ini menggunakan algoritma Berlekamp-Massey. Algoritma ini akan menghasilkan polinomial lokasi *error* dimana polinomial yang didapatkan dari hasil simulasi yaitu: $A(x) = 85x + 17x^2 + 125x^3 + 74x^4 + 124x^5 + 77x^6 + 240x^7 + 179x^8 + x^9$

Lokasi Error

Pencarian lokasi *error*, seperti ditunjukkan dalam Gambar 10, membutuhkan polinomial lokasinya untuk mengetahui pada lokasi mana *error* tersebut terjadi. Hasil simulasi menunjukkan bahwa lokasi *error* terdapat pada rcnt (rx counter) ke 254, 253, 252, 99, 29, 19, 9, dan 0.

Magnitude Error

Pencarian *magnitude error*, Gambar 11, yaitu mencari besaran *error* yang terjadi pada lokasi *error* yang telah didapatkan sebelumnya. Hasil simulasi menunjukkan bahwa besaran *error* yang terjadi yaitu: 49,17,85,152,69,255,44, dan 77.

Koreksi Error

Hasil simulasi data yang telah dikoreksi telah sesuai dengan teorinya dimana kinerjanya dapat dilihat pada Tabel 2.

Hasil simulasi ini juga memperlihatkan bahwa selama periode waktu 3×10^7 ps (33×10^{-5} detik) informasi yang diterima hanya 1243 byte sementara data yang dikirim dengan periode waktu yang sama sebanyak 1249 byte. Hal ini disebabkan karena pada program koreksi *error* memiliki FIFO buffer sebagai penyimpanan data sementara sehingga akan terjadi *delay* pada penerimaan data.

C. Simulasi Pendekodean *Reed Solomo* (255,239) dengan 9 *Error*

Simulasi ini dilakukan untuk menguji apakah benar *Reed Solomon* (255,239) hanya mampu mengoreksi maksimum 8

simbol *error*. Untuk membuktikan hal tersebut maka diinjeksikan 9 *error* kedalam program *testbench*nya.

Gambar 16 adalah hasil simulasi program dimana simulasi tersebut membuktikan bahwa RS (255,239) hanya mampu mengoreksi maksimum 8 simbol sesuai dengan teorinya. Hasil simulasi ini juga menegaskan bahwa pendkodean RS(255,239) akan gagal mendeteksi dan mengoreksi semua data informasi yang *error* jika *error* yang terjadi lebih dari 8 simbol (Lihat Gambar 12 dan 13).

D. Hasil Implementasi Reed Solomon (255,239)

Metode implementasi sistem ditunjukkan oleh gambar 4, dimana pengujiannya dilakukan dengan memberikan 2 jenis data informasi yaitu informasi berupa data teks dan informasi berupa data citra. Semua informasi dikirim dan diterima dengan baudrate 9600 bps.

Pengujian untuk data teks dilakukan dengan dua cara. Pengujian pertama yaitu mengirim karakter ASCII sebanyak 880 karakter. Hasil pengujian ini ditunjukkan pada gambar 13, dimana hasilnya menunjukkan bahwa karakter ASCII yang diterima sama dengan karakter ASCII yang dikirim. ASCII dikirim karakter demi karakter dimana tiap karakter terdiri dari 8 bit, maka FPGA (*encoder*) mengirim tiap 8 bit ke FPGA (*encoder*) sehingga waktu pengirimannya *real-time*.

Pengujian kedua yaitu mengirim file teks (.txt) dengan kapasitas 223 KB. Hasil pengujiannya membuktikan bahwa FPGA (*encoder*) mampu menerima file teks (.txt) tanpa terjadi kesalahan informasi. Waktu penerimaan file tersebut selama 200 detik sedangkan waktu penerimaan berdasarkan persamaan (1) selama 190 detik. Karakter ASCII yang dikirim dan diterima diberikan pada Gambar 14 dan 15.

Pengujian untuk data citra dilakukan dengan mengirim *file* citra (.jpg) berkapasitas 1,2 KB dengan dimensi 46 x 35 *pixel* dan berkapasitas 68,1 KB dengan dimensi 201 x 114 *pixel*. Kedua *file* tersebut berhasil diterima oleh penerima dengan kapasitas *file* dan dimensi yang sama dengan *file* yang dikirim, tetapi terjadi beberapa perubahan warna pada *file* yang diterima. Perubahan ini disebabkan karena data yang dikeluarkan oleh pendkodean RS(255,239) merupakan data keluaran pengkodean (*encoder*) RS(255,239) dimana pada data tersebut terdapat 16 *byte* tambahan tiap 255 *byte* (satu *block*). Data citra terdiri dari ribuan bit-bit sehingga banyak *codeword* tambahan (16 *byte*) yang terjadi, hal inilah yang menyebabkan citra tersebut mengalami perubahan warna yang cukup signifikan, lihat Gambar 16.

Waktu penerimaan *file* citra berkapasitas 1,2 KB yaitu 1,04 detik sedangkan waktu penerimaan berdasarkan persamaan (2.1) selama 1,02 detik. Waktu penerimaan *file* citra berkapasitas 68,1 KB yaitu 60 detik sedangkan waktu penerimaan berdasarkan persamaan (2.1) selama 58 detik. Kurva perbandingan kapasitas *file* dengan waktu penerimaan dapat dilihat pada Gambar 17.

Seperti halnya dengan hasil simulasi, *delay* yang terjadi disebabkan karena program ini menggunakan FIFO *buffer*, hanya saja saat implementasi kecepatan informasi diolah oleh program lebih lama karena pada FPGA program tersebut diubah terlebih dahulu menjadi rangkaian-rangkaian logika yang akan terhubung satusama lain.

V. KESIMPULAN/RINGKASAN

Pendkodean kanal *Reed Solomon* (255,239) untuk transmisi citra pada sistem komunikasi satelit nano telah diuji kemampuan mendeteksi dan mengoreksi *error*nya secara simulasi dan kemampuan menerima informasi yang sesuai dengan data yang dikirim oleh bagian pengkodean secara implementasi. Pendkodean ini diberi informasi sebanyak 1249 byte dengan menginjeksikan sebanyak 8 simbol *error* tiap 255 byte. Hasil simulasinya membuktikan pendkodean ini mampu mendeteksi dan mengoreksi 8 simbol *error* tersebut sehingga informasi yang diterima sama dengan yang dikirim dan berhasil mendeteksi tetapi gagal mengoreksi jika *error* yang terjadi tiap 255 *byte* lebih dari 8 simbol. Pendkodean ini mampu menerima informasi berupa karakter ASCII, file teks (.txt) dan file citra (.jpg) tanpa terjadi *error* dengan waktu penerimaannya hampir sama dengan waktu penerimaan berdasarkan persamaan secara teori. Pendkodean ini hanya untuk mendeteksi dan mengoreksi, apabila terjadi *error* tetapi tidak mengembalikan data informasi yang sebenarnya, dalam hal ini *codeword* (16 *byte*) tambahan tidak dihilangkan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada tim penelitian strategis nasional 2012 Kemdikbud “Pengembangan stasiun bumi untuk komunikasi data, citra dan video dengan satelit LEO VHF/UHF/S-band menuju kemandirian teknologi satelit” yang telah memberikan dukungan finansial.

DAFTAR PUSTAKA

- [1] Ze-Nian Li dan Mark S. Drew, “*Fundamentals of Multimedia*”, Prentice Hall, 2004.
- [2] B. Sklar, “*Digital Communications : Fundamentals and Applications*”. Prentice Hall, 2001
- [3] CCSDS Recommendation For Space Data System Standar, “*Telemetry Channel Coding*”, USA, 1999
- [4] Jariyah, Ainun, “Pengkodean Reed Solomon(255,239) Berbasis FPGA Untuk Transmisi Citra Pada Satelit Nano”, Laporan Tugas Akhir, Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember, Surabaya, 2013.
- [5] K. C. C. Wai dan S. J. Yang, “Field Programmable Gate Array Implementation of Reed-Solomon Code, RS (255, 239)”, New York 2006.
- [6] C.K.P. Clarke, “Reed-Solomon error correction, BBC R&D White Paper, WHP 03”1, July 2002.
- [7] Al Azad, Aqib dan Minhazul. Huq, Iqbalur. Rahman Rokon “Efficient Hardware Implementation of Reed Solomon Encoder and Decoder in FPGA using Verilog”, 2011.
- [8] Stephen Brown dan Jonathan Rose, “*Architecture of FPGAs and CPLDs: A Tutorial*”, Department of Electrical and Computer Engineering, University of Toronto.
- [9] Corporation, Altera, “DE1 Development and Education Board User Manual”, versi 1.1, 2006.