

PERFORMANSI TEKNIK PENGKODEAN LOW DENSITY PARITY CHECK PADA SISTEM TRANSMISI DATA SATELIT PENGINDERAAN JAUH LDCM

Ali Syahputra Nasution¹⁾, Dinari Nikken Sulastrie Sirin²⁾, Hidayat Gunawan³⁾

^{1,2,3} Perekayasa di Pusat Teknologi dan Data Penginderaan Jauh – LAPAN

Jl. LAPAN No 70, Pekayon, Pasar Rebo, Jakarta Timur 13710

Telp: (021)8710786, Fax: (021)8717715

E-mail: ali.syahputra@lapan.go.id; alisyahputra2003@yahoo.com

Abstrak

Telah dilakukan kajian penggunaan teknik pengkodean Low Density Parity check (LDPC) pada data satelit penginderaan jauh Landsat Data Continuity Mission (LDCM). Teknik pengkodean LDPC diterapkan pertama kalinya pada data satelit penginderaan jauh LDCM yang telah diluncurkan pada 11 Februari 2013. Secara umum terdapat dua algoritma/metode dalam decoding LDPC yaitu hard decision (bit flip) dan soft decision (log domain sum product). Makalah ini membahas mengenai performansi teknik pengkodean LDPC pada sistem transmisi data satelit LDCM yang dilakukan melalui simulasi dan uji coba berbasis perangkat lunak Matlab. Adapun parameter yang dianalisis adalah perbandingan Bit Error Rate (BER) terhadap Eb/No dengan mensimulasikan beberapa ukuran matriks parity check dan jumlah iterasi maksimum pada kedua metoda decoding LDPC. Hasil simulasi dan uji coba menunjukkan bahwa dengan metode decoding bit flip dan log domain sum product, perbedaan ukuran matriks dan jumlah iterasi maksimum mempengaruhi performansi LDPC.

Kata Kunci: LDPC, Bit Flip, Log Domain Sum Product, LDCM, BER, Eb/No

1. PENDAHULUAN

Pada sistem transmisi data satelit penginderaan jauh, semakin besar data yang dikirim maka semakin besar pula resiko kesalahan dalam pengiriman data dikarenakan banyaknya gangguan-gangguan yang ada pada kanal transmisi, seperti *noise*, interferensi, dan *fading*. Oleh karena itu diperlukan proses proses deteksi dan koreksi *error* atau *error control coding (ECC)* sehingga data yang dikirim dapat sampai di penerima dengan baik.

Banyak teknik pengkodean yang digunakan untuk mengoreksi dan mendeteksi kesalahan yang terjadi antara lain *Convolutional Codes (CC)*, *Reed Solomon (RS) Codes*, dan *Turbo Codes (TC)*. Terdapat pula macam-macam karakteristik kelebihan dan kekurangan dari setiap teknik pengkodean tersebut. CC tidak efisien dari segi *bandwidth*, walaupun tersedia pilihan *puncturing* untuk mengurangi masalah ini, namun hal ini membuat performansi BER menjadi buruk. RS Codes merupakan pengkodean yang sangat baik, namun untuk mencapai performansi yang baik dibutuhkan sejumlah *Signal to Noise Ratio (SNR)* tertentu. Penggabungan RS dan CC dapat mengurangi masalah ini, namun membutuhkan *bandwidth* yang lebih tinggi. TC memiliki karakteristik antara lain memiliki performansi mendekati kapasitas namun membutuhkan sejumlah *bandwidth* yang cukup besar, proses *decoding* yang lambat, dan dapat mencapai BER 10^{-6} . TC tidak cocok diterapkan pada aplikasi yang membutuhkan *bandwidth* kecepatan tinggi. Tren teknologi saat ini mengarah ke data rate yang tinggi dan *bandwidth* yang terbatas, namun memiliki performansi yang baik. Salah satu metode pengkodean yang dapat menunjang hal tersebut adalah *Low Density Parity check (LDPC) code* (Wai Fong, 2002).

Teknik pengkodean LDPC termasuk salah satu kelas dari *linear block code*. LDPC code didesain menggunakan matriks *parity check* yang bersifat *sparse*, dimana jumlah bit '1' lebih sedikit dibandingkan dengan jumlah bit '0', atau rasio jumlah bit '1' terhadap jumlah elemen dalam matrik *parity check* adalah $\ll 0,5$ (CCSDS 131.1-O-2, 1997). Teknik pengkodean LDPC pertama kali diperkenalkan oleh Gallager pada awal tahun 1960-an dalam disertasinya (Gallager, 1962). Dikarenakan keterbatasan teknologi dalam penerapan *encoder* dan *decoder*, LDPC diabaikan hingga hampir 30 tahun. Selama periode tersebut, pada tahun 1981, Tanner menyediakan interpretasi baru dari LDPC code yang disebut *bipartite graph* atau grafik Tanner (Tanner, 1981). Namun juga tidak ditanggapi selama 14 tahun. Kemudian sejak tahun 1993, penelitian mengenai LDPC dimulai kembali antara lain oleh David J. C. MacKay et.al, T. Richardson. Saat ini, LDPC telah diterapkan pada berbagai aplikasi modern seperti 10Gbase-T Ethernet, WiFi, WiMAX, *Digital Video Broadcasting (DVB)* (Tuan Ta), Keuntungan dari penggunaan LDPC code

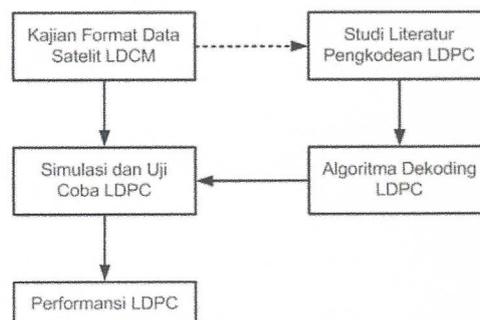
adalah dapat menyediakan kinerja yang sangat mendekati kapasitas kanal (David J. Mackay, 1996) dan mempunyai proses dekoding yang linear dan diharapkan dengan pengiriman *bit rate* yang tinggi dapat membantu untuk menghasilkan probabilitas kesalahan bit yang rendah.

Teknik pengkodean LDPC telah diterapkan untuk pertama kalinya sebagai *error control coding* pada data satelit penginderaan jauh *Landsat Data Continuity Mission* (LDCM/Landsat-8) (H. Garon et al. 2010). Satelit LDCM yang telah diluncurkan pada tanggal 11 Februari 2013 membawa dua sensor utama yaitu *Operational Land Imager* (OLI) dan *Thermal Infrared Sensor* (TIRS). LDPC yang diimplementasikan pada LDCM menggunakan versi *code rate* 7/8, dan berdimensi (8170, 7136). Matriks *parity check* LDPC yang digunakan pada data satelit LDCM memiliki ukuran 1022 x 8176 dan bersifat regular. Paket data LDCM ditransmisikan ke stasiun bumi melalui X-band (8200.5 MHz) dengan menggunakan modulasi OQPSK. Pada konfigurasi transmisi dari satelit ke stasiun bumi LDCM (Doug Daniels, 2010), ada pilihan bahwa demodulator telah dilengkapi modul ekstraksi.

Penulisan makalah ini bertujuan untuk menganalisis performansi teknik pengkodean LDPC pada sistem transmisi data satelit penginderaan jauh LDCM yang dilakukan melalui suatu simulasi dan uji coba beberapa ukuran matriks *parity check* dan jumlah iterasi maksimum pada kedua metoda dekoding LDPC (*bit flip* dan *log domain sum product*). Kegiatan ini merupakan salah satu tahap penguasaan teknik ekstraksi rawdata satelit LDCM, fokus pada dekoding LDPC. Kegiatan ini diharapkan dapat menambah pemahaman dalam mengatasi atau meminimalisir kesalahan pada data saat ditransmisikan pada kanal yang *noise*.

2. METODOLOGI

Dalam menyelesaikan penelitian ini dilakukan langkah-langkah seperti tampak pada gambar 1 di bawah ini.



Gambar 1. Metodologi Penelitian

2.1 Representasi LDPC code

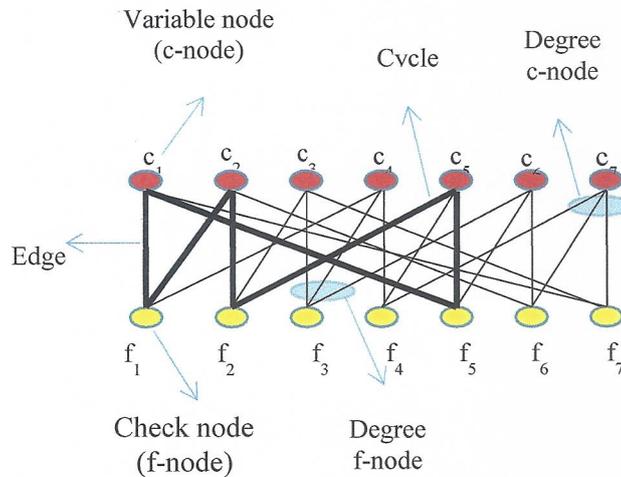
LDPC adalah salah satu kelas dari *linear block code* dimana matriks *parity check* $m \times n$ H mempunyai kerapatan rendah, dimana jumlah bit '1' pada baris (ω_r), dan jumlah bit 1 pada kolom sangat sedikit daripada ukuran matriks H ($\omega_r \ll m$, $\omega_c \ll n$). LDPC code dikelompokkan dalam dua kelas yaitu *regular* dan *irregular*. LDPC *code regular* mempunyai jumlah bit '1' dalam setiap baris dan setiap kolom konstan ($\omega_r = \omega_c (n/m)$), sedangkan LDPC *code irregular* memiliki jumlah bit '1' dalam setiap baris dan setiap kolom yang bervariasi. Gambar 2 menunjukkan contoh matriks *parity check* H LDPC *code regular*.

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

Gambar 2. Contoh Representasi Matriks Parity check LDPC Code Regular

Tanner memperkenalkan sebuah representasi grafik untuk LDPC code. Grafik Tanner adalah *bipartite graph* yaitu suatu grafik yang tidak langsung berhubungan tetapi memisahkan antara dua kelas. Ada dua tipe *node* dalam Grafik Tanner, yaitu *variabel nodes* (**c-nodes**) dan *check nodes* (**f-nodes**). Kemudian yang menyambungkan antara *c-nodes* dan *f-nodes* disebut *edge*. Jumlah *edge* yang terhubung ke *node* disebut *degree* dari *node* tersebut. Urutan dari *edge* yang terhubung yang bermula dan berakhir pada

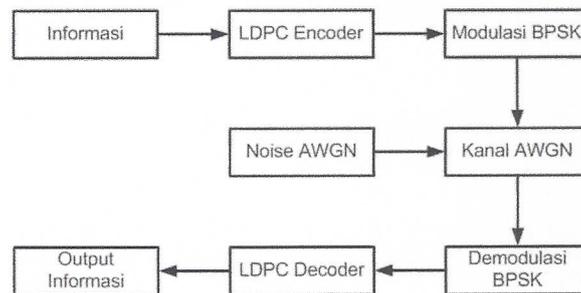
vertex yang sama, dan memenuhi kondisi dimana tidak boleh ada *vertex* yang muncul lebih dari satu kali (kecuali *vertex* inialisasi dan akhir) disebut dengan *cycle*. Matriks *parity check* $m \times n$ H di atas dapat dibuat grafik Tannernya seperti terlihat pada gambar 3 di bawah ini.



Gambar 3. Representasi Grafik Tanner

Pada gambar 2 di atas, dapat diamati c_1 , c_2 , dan c_4 dihubungkan dengan *check node* pertama (f_1) pada h_{11} , h_{12} , $h_{14} = 1$ (sedangkan yang lain adalah nol). Kemudian langkah selanjutnya yaitu *check node* f_1, f_2, \dots, f_7 , mengikuti baris 1, 2, ..., 7 pada H . Dengan catatan dengan mengikuti persamaan $cH^T = 0$, nilai bit yang dihubungkan pada *f-node* yang sama harus berjumlah 0.

Rancangan simulasi dan uji coba performansi LDPC dapat dilihat pada gambar 4 di bawah ini.



Gambar 4. Blok Rancangan Simulasi dan Uji Coba Performansi LDPC

Blok rancangan simulasi dan uji coba performansi LDPC menggunakan bahasa pemrograman berupa *software* matlab. Dalam simulasi ini, bit informasi yang dimasukkan ke LDPC *encoder* bukan merupakan data LDPC sebenarnya, namun merupakan serangkaian bit-bit (data biner) acak yang dihasilkan melalui perintah pada matlab. Demikian pula halnya dengan sistem modulasinya, dimana output LDPC *encoder* yang berupa *codeword* akan dimodulasi dengan menggunakan modulasi BPSK yang kemudian dikirim melalui kanal AWGN, disini kanal AWGN akan dtambahkan dengan *noise white Gaussian*, keluaran kanal AWGN bercampur *noise* akan diterima oleh demodulator BPSK kemudian masuk ke LDPC *decoder* di sisi penerima. Pada sisi penerima dilakukan dekoding LDPC dengan metode *hard decision* (bit flip) dan *soft decision* (*log domain sum product*). Dari output *decoder* LDPC akan dihasilkan output bit informasi yang telah dikirimkan.

Pengkodean LDPC yang diterapkan pada simulasi ini adalah LDPC *codes* regular, namun belum sesuai dengan yang terdapat pada LDPC, dimana ukuran matriks $m \times n$ H yang digunakan berukuran 100×200 dan 200×400 (rate $\frac{1}{2}$). Analisis performansi LDPC dilakukan berdasarkan parameter BER terhadap E_b/N_0 dengan mensimulasikan beberapa ukuran matriks *parity check* dan jumlah iterasi maksimum pada kedua metoda *decoding* LDPC (*bit flip* dan *log domain sum product*).

2.2 Dekoding LDPC code

Secara umum ada 2 metode dekoding LDPC *code* yaitu *hard decision* dan *soft decision*.

a. Dekoding hard decision (Bit flip)

Dekoding *bit flip* didasarkan pada penandaan *hard decision* ('0' atau '1') untuk setiap bit yang diterima. Bagian yang penting dari Dekoding *bit flip* adalah pelewatan pesan (*message passing*) antar *node* pada grafik Tanner.

Algoritma *hard decision decoding (bit flip)* (Sarah J. Johnson) sebagai berikut:

- Langkah 1: inialisasi: setiap *variable node* menandai nilai bit yang diterima kanal dan mengirimkan pesan ke *check node* yang terhubung pada grafik Tanner yang mengindikasikan nilainya.
- Langkah 2: *Parity update*: dengan menggunakan pesan dari *variable node*, setiap *check node* mengecek apakah persamaan check paritas terpenuhi. Jika seluruh persamaan cek paritas terpenuhi algoritma berhenti. Jika tidak, setiap *check node* mengirimkan pesan ke *variable node* yang terhubung mengindikasikan apakah persamaan cek paritas terpenuhi atau tidak.
- Langkah 3 : *Variable update*; jika mayoritas pesan yang diterima setiap *message node* tidak memenuhi, *variable node* merubah (*flip*) nilai saat ini. Kemudian kembali ke langkah 2, jika jumlah iterasi maksimum terlampaui dan *codeword* belum *valid*, maka algoritma berhenti dan pesan *failure to converge* dilaporkan.

b. Dekoding soft decision (Log domain sum product)

Algoritma dekoding *sum product* mirip dengan algoritma *bit flip*, hanya pesan yang dipertukarkan antara *variable node* dan *check node* adalah nilai probabilitas yang direpresentasikan dengan (LLR).

Algoritma dekoding *soft decision (log domain sum product)* (Sarah J. Johnson) sebagai berikut:

- Hitung nilai probabilitas *prior log-likelihood ratios* dari bit-bit yang diterima.

$$r_i = \begin{cases} \log \frac{p}{1-p}, & \text{if } y_i = 1, \\ \log \frac{1-p}{p}, & \text{if } y_i = 0. \end{cases} \quad (1)$$

- Tentukan jumlah iterasi maksimum; hitung nilai inialisasi $M_{j,i} = r_i$.
- Proses dari *check node* menuju *variable node*
Hitung nilai probabilitas ekstrinsik dari suatu *check node* ke- j menuju *variable node* ke- i . Nilai probabilitas ekstrinsik tersebut tergantung pada probabilitas *variable node* ke- i lainnya yang terhubung ke *check node* ke- j selain *variable node* ke- i ($M_{j,i'}$).

$$E_{j,i} = 2 \tanh^{-1} \left(\prod_{i' \in B_j, i' \neq i} \tanh(M_{j,i'}/2) \right) \quad (2)$$

- Gabungkan nilai probabilitas intrinsik dan ekstrinsik dari setiap *variablenode*, kemudian hitung nilai total LLR dari *variablenode* ke- i . Jika total LLR dari kanal menunjukkan negatif, maka bit bernilai '1', dan sebaliknya Jika total LLR dari kanal menunjukkan positif, maka bit bernilai '0'.

$$L_i = LLR(P_i^{int}) = r_i + \sum_{j \in A_i} E_{j,i} \quad (3)$$

- Dengan menggunakan nilai dari LLR total dari setiap *variablenode*, setiap *check node* mengecek apakah persamaan cek paritas terpenuhi ($s=cH^T$). Jika seluruh persamaan cek paritas terpenuhi algoritma berhenti. Jika tidak, proses berlanjut untuk iterasi berikutnya.
- Untuk menghindari pengiriman kembali nilai $M_{j,i}$ ke setiap informasi *check node* yang telah dimiliki, pesan dari *variable node* ke- i ke *check node* ke- j merupakan penjumlahan (persamaan 3) tanpa komponen $E_{j,i}$ yang telah diterima dari *check node* ke- j .

$$M_{j,i} = \sum_{j' \in A_i, j' \neq j} E_{j',i} + r_i \quad (4)$$

2.3 Modulasi BPSK

Modulasi BPSK adalah salah satu modulasi digital dengan membedakan fase, dimana perbedaan antara bit '1' dan bit '0' ditandai dengan adanya perbedaan fase pada sinyal output dari modulator BPSK tersebut. Arus data logika '1' ditransmisikan sinyal pembawa dengan fase 0° , sedangkan arus data logika '0' ditransmisikan sinyal pembawa dengan fase 180° .

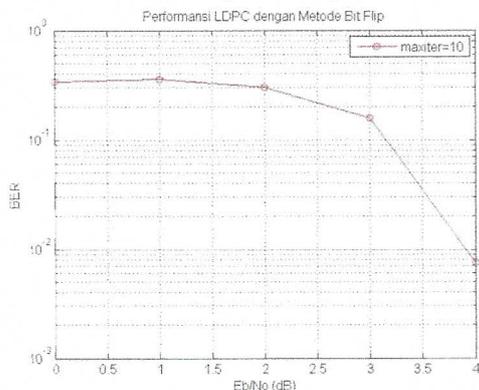
2.4 Kanal transmisi AWGN

Pada umumnya sinyal yang diterima di penerima adalah sinyal langsung ditambah dengan sinyal yang telah terdistorsi baik karena pantulan, hamburan maupun pembelokan pada saat transmisi. Penambahan *noise* juga menyebabkan terjadinya perubahan karakteristik sinyal yang dikirim (amplitudo maupun pergeseran frekuensi *carrier*). Dalam hal ini jenis kanal yang akan dibahas adalah kanal *Additive White Gaussian Noise* (AWGN). Kanal AWGN merupakan salah satu contoh dari kanal DMC (*Discrete*

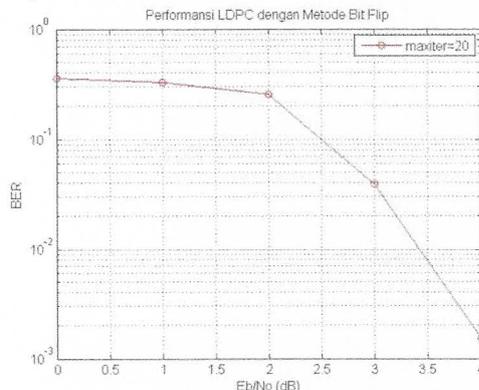
Memoryless Channel). AWGN juga merupakan tipe kanal komunikasi digital yang paling mudah dianalisa. Dalam kanal ini diasumsikan tidak ada distorsi atau pengaruh lainnya selain penambahan *noisewhite Gaussian*.

3. HASIL DAN PEMBAHASAN

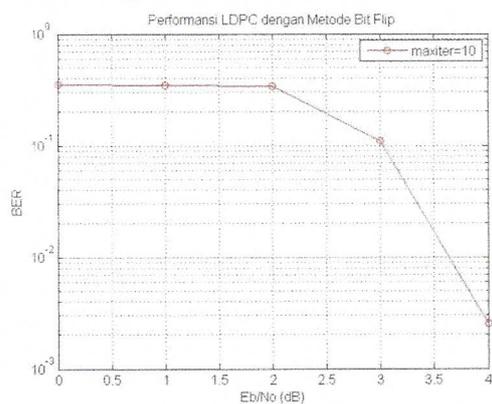
Melalui simulasi dan uji coba diperoleh data-data sebagai berikut:



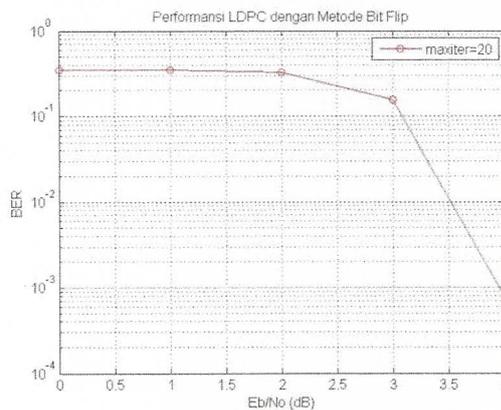
Gambar 5a. Metode Decoding Hard Decision (Bit Flip) Ukuran Matriks 100 x 200, Iterasi=10



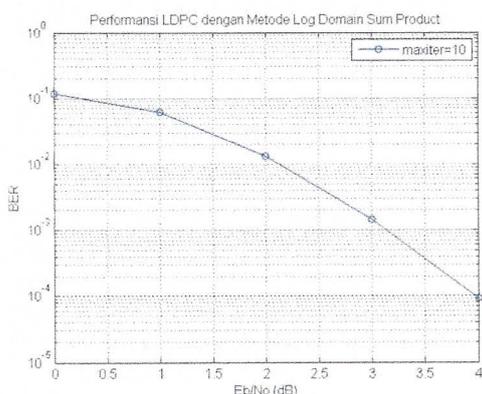
Gambar 5b. Metode Decoding Hard Decision (Bit Flip) Ukuran Matriks 100 x 200, Iterasi=20



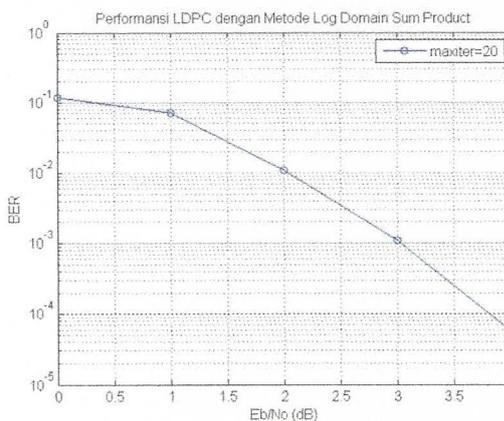
Gambar 6c. Metode Decoding Hard Decision (Bit Flip) Ukuran Matriks 200 x 400, Iterasi=10



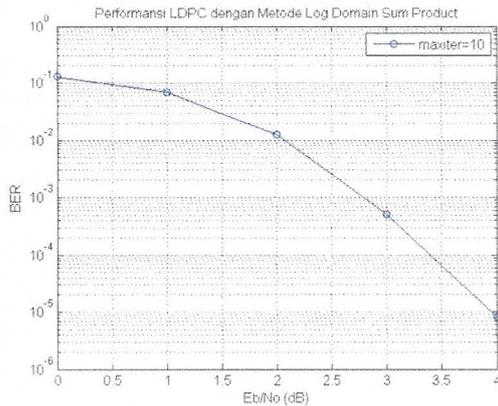
Gambar 6d. Metode Decoding Hard Decision (Bit Flip) Ukuran Matriks 200 x 400, Iterasi=20



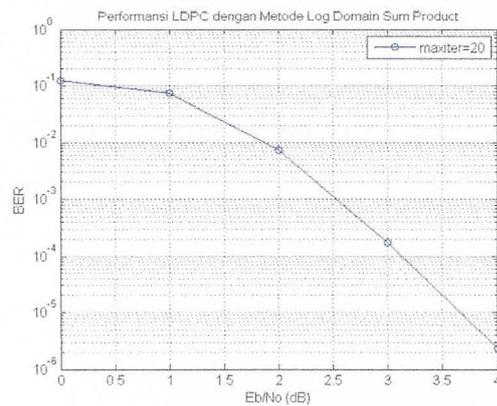
Gambar 6a. Metode Decoding Soft Decision (Log Domain Sum Product) Ukuran Matriks 100 x 200, Iterasi=10



Gambar 6b. Metode Decoding Soft Decision (Log Domain Sum Product) Ukuran Matriks 100 x 200, Iterasi=20



Gambar 6c. Metode Decoding Soft Decision (Log Domain Sum Product) Ukuran Matriks 200 x 400, Iterasi=10



Gambar 6d. Metode Decoding Soft Decision (Log Domain Sum Product) Ukuran Matriks 200 x 400, Iterasi=20

Berdasarkan hasil simulasi di atas dapat dilihat bahwa seiring dengan perubahan ukuran matriks, maka performansi LDPC baik dengan metode *decoding hard decision (bit flip)* dan metode *decoding soft decision (log domain sum product)* mengalami perubahan yang cukup signifikan. Hal ini dapat dilihat pada sistem LDPC dengan metode *bit flip* ukuran matriks 100 X 200 dengan iterasi maksimum = 20 (lihat gambar 5b), pada Eb/No = 4 dB target BER hampir mencapai 10^{-3} , sedangkan untuk ukuran matriks 200 x 400 dengan iterasi maksimum = 20 (lihat gambar 5d), pada Eb/No = 4 dB target BER mencapai lebih dari 10^{-3} .

Demikian juga halnya pada sistem LDPC dengan metode *log domain sum product*, pada ukuran matriks 100 X 200 dengan iterasi maksimum = 20 (lihat gambar 6b), pada Eb/No = 4 dB target BER hampir mencapai 10^{-5} , sedangkan untuk ukuran matriks 200 x 400 dengan iterasi maksimum = 20 (lihat gambar 6d), pada Eb/No = 4 dB target BER telah mencapai lebih dari 10^{-5} .

Kemudian berdasarkan pengamatan dapat dilihat bahwa seiring dengan kenaikan jumlah iterasi maksimum, maka performansi LDPC baik dengan metode *decoding hard decision (bit flip)* dan metode *decoding soft decision (log domain sum product)* juga mengalami perubahan yang cukup signifikan. Hal ini dapat dilihat pada sistem LDPC dengan metode *bit flip* ukuran matriks 100 X 200 dengan iterasi maksimum = 10 (lihat gambar 5a), pada Eb/No = 4 dB target BER mencapai sekitar 10^{-2} , sedangkan dengan iterasi maksimum = 20 (lihat gambar 5b), pada Eb/No = 4 dB target BER hampir mencapai 10^{-3} . Pada ukuran matriks 200 x 400 dengan iterasi maksimum = 10 (lihat gambar 5c), pada Eb/No = 4 dB target BER hampir mencapai 10^{-3} , sedangkan dengan iterasi maksimum = 20 (lihat gambar 5d), pada Eb/No = 4 dB target BER mencapai lebih dari 10^{-3} .

Demikian juga halnya pada sistem LDPC dengan metode *log domain sum product*, pada ukuran matriks 100 X 200 dengan iterasi maksimum = 10 (lihat gambar 6a), pada Eb/No = 4 dB target BER mencapai sekitar 10^{-4} , sedangkan dengan iterasi maksimum = 20 (lihat gambar 6b), pada Eb/No = 4 dB target BER mencapai lebih dari 10^{-4} . Pada ukuran matriks 200 x 400 dengan iterasi maksimum = 10 (lihat gambar 6c), pada Eb/No = 4 dB target BER mencapai 10^{-5} , sedangkan dengan iterasi maksimum = 20 (lihat gambar 6d), pada Eb/No = 4 dB target BER hampir mencapai 10^{-6} .

4. KESIMPULAN

Berdasarkan simulasi telah diperoleh hasil bahwa pada Eb/No 4 dB, performansi LDPC dengan metode *decoding hard decision (bit flip)* dan metode *decoding soft decision (log domain sum product)* mengalami perbaikan BER seiring dengan perubahan ukuran matriks *parity check* maupun jumlah iterasi maksimum. Oleh karena itu dapat disimpulkan bahwa perbedaan ukuran matriks dan jumlah iterasi maksimum mempengaruhi performansi sistem LDPC. Semakin besar ukuran matriks *parity check* dan jumlah iterasi maksimum, semakin kecil error (BER) yang dihasilkan. Namun semakin besar ukuran matriks dan jumlah iterasi maksimum membutuhkan waktu proses yang lebih lama. Program simulasi performansi LDPC ini masih terdapat kekurangan sehingga perlu diperbaiki dan dikembangkan dengan menggunakan parameter-parameter sesuai yang diterapkan pada LDPC.

5. DAFTAR RUJUKAN

- [1]. Barry, John. R., 2001. *Low-Density Parity check Codes*. Georgia Institute of Technology.
- [2]. CCSDS. 2007. *Low Density Parity check Codes for Use in Near-Earth and Deep Space Application* (Issue 2). Research and Development for Space Data System Standards, Orange Book. Washington DC.
- [3]. Daniels, Doug., 2010. *LDCM Science Data Reception*. USA: NASA and USGS.
- [4]. Eroz, Mustafa, dkk., *DVB-S2 Low Density Parity check Codes with Near Shannon Limit Performance*. USA: Hughes Network System.
- [5]. Fong, Wai, dkk., *Low Density Parity check Codes: Bandwidth Efficient Channel Coding*. NAS/GSFC.
- [6]. Fong, Wai. (2002). *White Paper for Low Density Parity check (LDPC) Codes for CCSDS Channel Coding Blue Book*. NASA GSFC.
- [7]. Gallager, R.G., 1962. Low Density Parity check Codes. *IRE Transactions on Information Theory*, pp. 21-28.
- [8]. Garon, H, dkk., 2010. *Report on Flight Application: Successful Use of a High Rate LDPC Code with High Data Rate in a Restricted Band*. NASA GSFC.
- [9]. Garon, H, dkk., 2011. *Landsat Data Continuity Mission (LDCM) CCSDS File Extraction Protocol (CFDP)-A Working Guideline In Pseudo Code*. CCSDS Next Gen Uplink WG Meeting, NASA GSFC.
- [10]. Johnson, Sarah J., *Introducing Low-Density Parity check Codes*. Australia: School of Electrical Engineering and Computer Science, The University of Newcastle.
- [11]. Mackay, David J.C., M. Neal, Radford., 1996. *Near Shannon Limit Performance of Low Density Parity check Codes*.
- [12]. NASA & USGS., 2010. *LDCM Space to International Cooperator (IC) Interface Specification Document (ISD)*. USA: Department of Interior US. Geological Survey.
- [13]. Nasution, Ali Syahputra, dkk., 2011. *Laporan Akhir Kajian Format Data CCSDS Satelit Penginderaan Jauh Sumber Daya Alam LDCM*. Riset Insentif PKPP, Jakarta.
- [14]. Nugroho, Bagawan S., 2007. *LDPC Codes BER Simulation*. Diunduh dari <http://www.mathworks.com/matlabcentral/fileexchange/14869-ldpc-codes-ber-simulation>. [Diakses 19 April 2013].
- [15]. Ricahrdson, T., Urbanke, R., 2001. Design of Capacity-Approaching Low Density Parity check Codes. *IEEE Transf. Inform. Theory* 47: 619-637.
- [16]. S. Lin, D.J. Costello, Jr., 2004. *Error Control Coding Book*. 2nd ed. New Jersey: Pearson Prentice Hall.
- [17]. Ta, Tuan. (n.d.). *A Tutorial on Low Density Parity-Check Codes*. Austin: The University of Texas.
- [18]. Tanner, R. M., 1981. A Recursive Approach to Low Complexity Codes. *IEEE Trans. Information Theory*, pp 533-547.
- [19]. Zaheer, Saikh Faisal, 2005. *LDPC Code Simulation*. Diunduh dari <http://www.mathworks.com/matlabcentral/fileexchange/8977-ldpc-code-simulation>. [Diakses 23 April 2013].