

PENGGUNAAN TEKNIK PENGKODEAN LOW DENSITY PARITY CHECK PADA DATA SATELIT PENGINDERAAN JAUH

Oleh :
Ali Syahputra Nasution*
Budhi Gustiandi*
Ayom Widipaminto**

Abstrak

Telah dilakukan kajian penggunaan teknik pengkodean Low Density Parity Check (LDPC) pada data satelit penginderaan jauh. Teknik pengkodean LDPC merupakan salah satu kelas dari teknik pengkodean linear block dengan karakteristik matriks parity checknya yang bersifat sparse yang hanya berisi sedikit bit '1' jika dibandingkan dengan jumlah bit '0'. Teknik pengkodean LDPC akan diterapkan pertama kalinya pada satelit penginderaan jauh sumber daya alam Landsat Data Continuity Mission (LDCM) yang direncanakan akan diluncurkan pada tahun 2012. Terdapat beberapa algoritma dalam dekoding LDPC yaitu dekoding One-Step Majority-Logic (ML), dekoding Bit-Flipping (BF), dekoding Weighted Majority-Logic, dekoding Weighted Bit-flipping, dan dekoding Sum-Product Algorithm (SPA). Makalah ini membahas mengenai cara kerja dan perbandingan kinerja dari masing-masing algoritma dekoding LDPC. Dari hasil kajian, diketahui bahwa algoritma dekoding Majority-logic memiliki kinerja perbaikan error yang cukup baik namun kompleksitas dekodingnya sangat sederhana bila dibandingkan dengan beberapa algoritma dekoding lainnya. Sedangkan algoritma dekoding Sum-product Algorithm (SPA) memiliki kinerja perbaikan error yang paling baik namun kompleksitas dekodingnya sangat tinggi bila dibandingkan dengan beberapa algoritma dekoding lainnya. Diharapkan hasil kajian dapat berguna dalam mengekstraksi rawdata satelit LDCM.

Kata Kunci: LDPC, One-Step Majority-Logic, Bit-Flipping, Weighted Majority-Logic, Weighted Bit-flipping, Sum-Product Algorithm.

Abstract

Study about utilizing Low Density Parity Check (LDPC) coding technique on remote sensing satellite data has been conducted. LDPC coding technique is one of a class of linear block coding technique with sparse parity check matrix characteristic and only contain a little bits of "1" compared with number of bits "0". LDPC coding technique will be implemented for the first time on natural resource remote sensing satellite Landsat Data Continuity Mission (LDCM) that planned will be launched in 2012. There are several algorithm in LDPC decoding, such as One-step Majority-Logic (MLG) decoding, Bit-Flipping (BF) decoding, Weighted Majority-Logic decoding, weighted Bit-Flipping decoding, and Sum-Product Algorithm (SPA) decoding. This paper discuss about how each LDPC decoding algorithm work and their performance comparison. From the study, there has been known that MLG decoding algorithm have error correcting performance good enough but with very simple decoding complexity when compared to other's decoding algorithm. While SPA decoding algorithm have the best error correcting performance with very high decoding complexity when compared with other's decoding algorithm. Study results are expected to be useful in extracting the LDCM satellite rawdata.

Key Words: LDPC, One-Step Majority-Logic, Bit-Flipping, Weighted Majority-Logic, Weighted Bit-flipping, Sum-Product Algorithm.

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 dalam penyeleksian sehingga data yang dikirim dapat sampai di penerima dengan baik dengan suatu proses deteksi dan koreksi error atau *error control coding*.

* Perekraya di Bidang Teknologi Akusisi dan Stasiun Bumi Penginderaan Jauh

**Kepala Bidang Teknologi Akuisis dan Stasiun Bumi Penginderaan Jauh

Secara umum terdapat dua strategi perbaikan error yaitu dengan menggunakan metode *Automatic Repeat Request* (ARQ) dan *Forward Error Correction* (FEC). ARQ hanya dapat digunakan pada kanal transmisi dua arah (ada kanal feedback) sedangkan untuk kanal satu arah, hanya dapat menggunakan FEC. FEC adalah metode yang paling umum digunakan untuk mengurangi *delay* proses transmisi atau komunikasi. Terdapat dua jenis teknik FEC yang digunakan dalam sistem komunikasi digital yaitu *Block code* dan *Convolutional code*. Kedua jenis pengkodean tersebut mempunyai karakteristik kelebihan dan kekurangan masing-masing. *Block code* sangat baik dalam mengatasi *burst error* sedangkan *convolutional code* sangat baik saat terjadi *random error*.

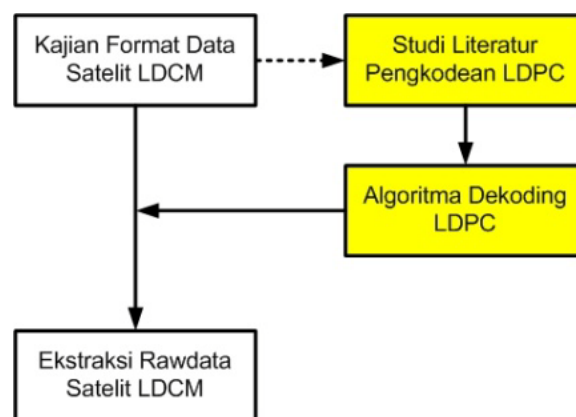
Salah satu metode error control coding yang digunakan pada data satelit penginderaan jauh adalah Low Density Parity Check (LDPC) code. Teknik Teknik pengkodean *Low Density Parity Check* (LDPC) termasuk salah satu kelas dari *linear block code*. LDPC code didesain menggunakan matriks *parity check* yang bersifat *sparse*, dimana jumlah bit '0' lebih banyak dibanding bit '1' atau rasio jumlah bit 1 terhadap jumlah elemen dalam matrik *parity check* adalah $\ll 0,5$ [1]. LDPC code pertama kali diperkenalkan oleh Gallager [2] pada awal tahun 1960-an dalam disertasinya. Dikarenakan keterbatasan teknologi dalam penerapan encoder dan decoder pengkodean LDPC, LDPC diabaikan hingga hampir 30 tahun. Selama periode tersebut, pada tahun 1981, Tanner menyediakan interpretasi baru dari kode LDPC yang disebut *bipartite graph* atau *Tanner graph*. Namun juga tidak ditanggapi selama 14 tahun. Kemudian sejak tahun 1993, penelitian mengenai LDPC code dimulai kembali antara lain oleh David J. C. MacKay et.al [3] T. Richardson [4]. Saat ini, LDPC telah diterapkan pada berbagai aplikasi modern seperti 10Gbase-T Ethernet, WiFi, WiMAX, *Digital Video Broadcasting* (DVB). Keuntungan dari penggunaan LDPC adalah dapat menyediakan kinerja yang sangat mendekati kapasitas kanal dan mempunyai proses dekoding yang linear dan diharapkan dengan pengiriman *bit rate* yang tinggi dapat membantu untuk menghasilkan probabilitas kesalahan bit yang rendah [5].

Teknik pengkodean LDPC akan diterapkan pertama kalinya sebagai *forward error control* pada data satelit penginderaan jauh sumber daya alam *Landsat Data Continuity Mission* (LDCM). Satelit LDCM / Landsat 8 merupakan generasi satelit Landsat masa depan yang rencananya akan diluncurkan pada tahun 2012 [6]. Satelit ini akan membawa dua sensor utama yaitu *Operational Land Imager* (OLI) dan *Thermal Infrared Sensor* (TIRS) yang dipaketkan dengan format data sesuai rekomendasi *Consultative Committee For Space Data Systems* (CCSDS) [7].

Pada konfigurasi transmisi dari satelit ke stasiun bumi LDCM [8], ada pilihan bahwa demodulator telah dilengkapi modul ekstraksi. Penulisan paper ini dimaksudkan untuk mengkaji teknik pengkodean LDPC khususnya metode / algoritma dekoding LDPC yang berguna untuk mengekstraksi rawdata satelit LDCM.

2. METODOLOGI

Kajian ini merupakan hasil studi literature beberapa makalah, artikel, buku, dan sumber lainnya yang berisi mengenai teknik pengkodean *low density parity check* (LDPC). Dari hasil studi literatur dianalisa algoritma-algoritma dekoding LDPC yang berguna dalam mengestraksi rawdata satelit penginderaan jauh LDCM.



Gambar 2.1. Diagram alur penelitian

3. HASIL DAN PEMBAHASAN

Pengkodean LDPC didefinisikan sebagai ruang nol dari sebuah matriks *parity check* (H) yang memiliki sifat struktural [9] yaitu:

- (1) setiap baris terdiri atas bobot bit '1' ρ ,
- (2) setiap kolom terdiri atas bobot bit '1' Υ ,
- (3) jumlah dari kesamaan bit '1' diantara dua kolom, yang dinotasikan dengan λ , tidak lebih besar dari 1; jadi $\lambda = 0$ atau 1;
- (4) ρ dan Υ keduanya kecil dibandingkan dengan panjang dari kode dan jumlah baris dalam H.

Karena keduanya ρ dan Υ lebih kecil dibandingkan dengan panjang kode (n) dan jumlah dari baris pada matriks (J), oleh karena itu H memiliki kepadatan kurang dari 1. Karena alasan ini, H dikatakan sebuah matriks *low density parity check* dan kodenya ditentukan oleh H karenanya disebut sebuah kode LDPC. Density r dari matriks parity check H didefinisikan sebagai perbandingan atas jumlah total bit 1 pada H dengan jumlah total entri pada H. Jadi,

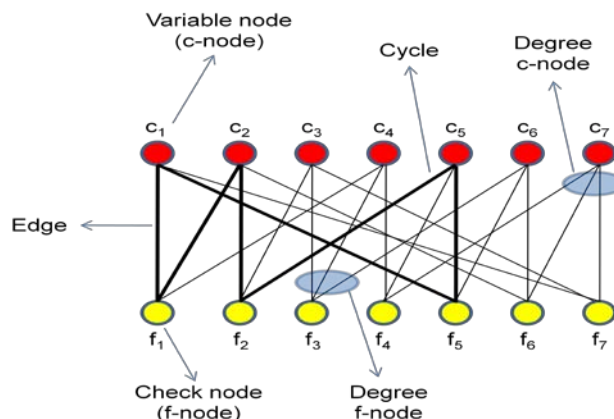
$$r = \rho / n = \Upsilon / J$$

LDPC code dikelompokkan dalam dua kelas yaitu regular dan irregular LDPC codes. Regular codes mempunyai jumlah bit '1' (bobot) dalam setiap baris dan setiap kolom konstan (oleh karena itu Gallager codes diklasifikasikan sebagai *regular codes*), sedangkan *irregular codes* memiliki jumlah bit '1' (bobot) dalam setiap baris dan setiap kolom yang bervariasi.

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

Gambar 3.1. Contoh representasi matriks parity check LDPC

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, dan 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**.



Gambar 3.2. Representasi grafik Tanner

Dalam sistem komunikasi, kehadiran berbagai jenis noise dapat merusak sinyal. Channel coding bermaksud memperoleh kembali data asli dari bit codeword yang ditransmisikan dan melakukan proses deteksi dan koreksi bila terjadi kesalahan pada bit codeword. Dalam teknik

pengkodean *linear block*, proses deteksi kesalahan dapat dilakukan dengan mengecek persamaan *parity check (syndrome)*. Misalkan LDPC code (n,k) dengan matrik *parity check* H , c merupakan codeword yang dikirimkan pada kanal yang *noisy*, y merupakan codeword yang diterima pada kanal output, dan e merupakan pola error yang disebabkan noise kanal.

$$y = c + e \rightarrow e = c + y$$

Saat vektor y diterima, decoder menghitung *syndrome* dari y dengan mengkalikan vektor y yang diterima dengan *transpose* dari matrik *parity check*nya.

$$s = y \cdot H^T \quad (1)$$

Jika persamaan *parity check (syndrome)* $s = 0$, maka tidak ada error pada codeword yang diterima, namun jika $s \neq 0$, maka ada error pada codeword yang diterima. Sebuah komponen tidak nol pada s menandakan sebuah kegagalan *parity*. Jumlah total kegagalan *parity* sama dengan jumlah komponen-komponen *syndrome* tidak nol dalam s .

$$s = e \cdot H^T \quad (2)$$

Terdapat beberapa metode / algoritma dekoding LDPC untuk mengoreksi kesalahan pada data dari kompleksitas dekoding rendah hingga ke tinggi dan dari kinerja perbaikan error yang cukup baik hingga paling baik. Diantaranya yaitu Dekoding *One-Step Majority-Logic* (MLG), Dekoding *Bit-Flipping* (BF), Dekoding *Weighted Majority-Logic*, Dekoding *Weighted Bit-Flipping*, dan Dekoding *Sum-Product Algorithm* (SPA) [9][10].

Berikut penjelasan cara kerja dari masing-masing metode / algoritma dekoding LDPC.

a) Dekoding One-Step Majority-Logic (MLG)

Misalkan y merupakan vektor yang diterima setelah dilakukan hard decision. Telah diketahui bahwa jika error terjadi ($s \neq 0$), berarti bahwa beberapa persamaan *parity check* tidak terpenuhi. Ide dibalik dekoding *one-step majority-logic* adalah bahwa setiap bit codeword ditentukan nilainya agar persamaan *parity check*nya. Jadi, jika kebanyakan dari *parity check*nya tidak terpenuhi, kita harus membalikkan (*flip*) nilainya, jika tidak simpan nilai aslinya. Jika dinotasikan kumpulan *parity check* yang berhubungan dengan bit i adalah P_i , sehingga aturan update untuk setiap bit codeword adalah:

$$y_i' = y_i + \text{majority}(P_i) \quad (3)$$

dimana $+$ merupakan penambahan modulo 2 dan y_i' merupakan nilai terbaharui untuk bit codeword i . Agar prosedur ini bekerja dengan benar, P_i harus ortogonal dengan bit codeword i .

Cara lain untuk menjelaskan dekoding MLG melalui grafik Tanner. Seluruh *variabel nodes* mengirimkan nilai yang diterima ke *check nodes* yang bersesuaian. Saat mendekoda variabel node i , dinotasikan dengan c_i , *check node* $P_i^{(j)}$, $j \in \{0,1, \dots, \text{deg}(c_i)\}$, mengirimkan penjumlahan modulo-2 dari seluruh nilai yang datang ke dia kembali ke c_i , kecuali yang berasal dari c_i . c_i kemudian didekoda sebagai majority dari message yang datang ke dia. Ortogonalitas dapat dijamin dengan membuang 4-cycles dari grafik Tanner. Berdasarkan penjelasan di atas dapat dirangkum langkah-langkah algoritma dekoding *one-step majority-logic* sebagai berikut:

Untuk $i = 0$ hingga $n - 1$ lakukan

- Langkah 1. Hitung nilai dari majority P_i .
- Langkah 2. Berikan nilai $y_i + \text{majority}(P_i)$ kepada codeword bit i (y_i')
- Langkah 3. Hitung ulang persamaan-persamaan *parity check* berdasarkan nilai baru dari y_i' .

selesai

b) Dekoding Bit-Flipping (BF)

Dekoding bit-flipping telah diperkenalkan oleh Gallager selama penemuan pengkodean LDPC. Telah diketahui bahwa jika error terjadi ($s \neq 0$), berarti bahwa beberapa persamaan parity check tidak terpenuhi. Ide utama dibalik algoritma BF adalah pemilihan seluruh variabel yang dimuat dalam lebih dari beberapa jumlah δ yang tetap dari persamaan *parity check* yang tidak terpenuhi dan membalikkan (*flip*) nilainya. Dengan menggunakan nilai ini, *parity check* dihitung kembali dan prosedur diulang. Secara sederhana kita memilih untuk membalikkan variabel yang termasuk dalam jumlah terbesar dari persamaan *parity check* yang tidak terpenuhi. Ketika *parity check* terpenuhi, atau jumlah iterasi maksimum tercapai, dekoding berhenti yang menyatakan dekoding berhasil atau gagal. Berdasarkan penjelasan di atas dapat dirangkum langkah-langkah algoritma dekoding *bit-flipping* sebagai berikut:

Saat $s \neq 0$ dan jumlah iterasi maksimum belum tercapai lakukan

- Langkah 1. Temukan jumlah persamaan-persamaan *parity-check* yang gagal untuk masing-masing bit, ditandai dengan $f_i, i = 0, 1, \dots, n - 1$.
- Langkah 2. Identifikasi kumpulan S dari bit-bit dimana f_i -nya paling tinggi (bit yang paling salah).
- Langkah 3. Balikkan (*flip*) bit-bit pada kumpulan S .
- Langkah 4. Hitung ulang persamaan-persamaan *parity check* berdasarkan nilai baru dari bit yang dibalik.
- Langkah 5. Naikkan jumlah iterasi dengan 1

Selesai

c) Dekoding Weighted Majority-Logic

Dekoding weighted MLG digunakan untuk mengatasi kehilangan informasi saat menggunakan dekoding *hard decision* pada algoritma regular MLG. Algoritma ini secara sederhana menghitung nilai dari setiap persamaan *parity check* dengan pertama sekali menghitung *weighted check sum*nya. Misalkan pada kanal AWGN, *weighted check sum* dihitung dengan persamaan:

$$E_i \triangleq \sum_{s_j^{(l)}} (2s_j^{(l)} - 1) |y_j|_{min}^{(l)} \quad (3)$$

dimana E_i adalah *weighted check sum* yang orthogonal pada posisi bit codeword i , dan

$$|y_j|_{min}^{(l)} \triangleq \{ \min\{|y_i|\} : 0 \leq i \leq n - 1, h_{j,i} = 1 \} \quad (4)$$

Kemudian, dengan menggunakan kita dapat menggunakan aturan yang sederhana untuk memutuskan nilai dari persamaan parity check j

$$P_i = \begin{cases} 1, & \text{untuk } E_i > 0 \\ 0, & \text{untuk } E_i \leq 0 \end{cases}$$

Prosedur dekoding kemudian berlanjut dengan algoritma regular MLG, maka hanya pada posisi inilah informasi *soft* digunakan. Berdasarkan penjelasan di atas dapat dirangkum langkah-langkah algoritma dekoding *weighted majority-logic* sebagai berikut:

Untuk $i = 0$ hingga $n - 1$ lakukan

- Langkah 1. Hitung nilai dari *majority* P_i .
- Langkah 2. Berikan nilai $y_i + \text{majority}(P_i)$ kepada codeword bit i (y_i')
- Langkah 3. Hitung ulang persamaan-persamaan parity check berdasarkan nilai baru dari y_i' .

Selesai

d) Dekoding Weighted Bit-Flipping

Kerugian performansi yang signifikan diketahui dengan *hard decision* yang dibutuhkan agar algoritma dapat bekerja, karena informasi hilang dalam proses. Untuk mengatasi informasi yang hilang ini, ukuran reliabilitas dari *hard decision* diperkenalkan.

Pada kanal AWGN, ukuran keandalan adalah *magnitude* dari setiap symbol y_i yang diterima yang dinotasikan dengan $|y_i|$, $i \in \{0, 1, \dots, n-1\}$. Tentukan $|y|_{\min}$ berdasarkan **persamaan (4)**. Dengan kata lain $|y_j|_{\min}$ adalah reliabilitas minimum dari seluruh bit codeword yang berpartisipasi dalam persamaan *parity check* j . Kemudian tentukan *weighted check sum* E_i untuk setiap bit codeword ke i berdasarkan **persamaan (3)**.

Dimana s_j adalah kumpulan persamaan *parity check* yang ortogonal pada bit codeword i . Berdasarkan penjelasan di atas dapat dirangkum langkah-langkah algoritma *weighted bit-flipping* sebagai berikut:

Saat $s \neq 0$ dan jumlah iterasi maksimum belum tercapai lakukan

- Langkah 1. Hitung nilai E_i , $i = 0, 1, \dots, n-1$.
- Langkah 2. Identifikasi kumpulan S dari bit-bit dimana E_i -nya paling besar (bit yang paling salah).
- Langkah 3. Balikkan (*flip*) bit-bit pada kumpulan S .
- Langkah 4. Hitung ulang persamaan-persamaan *parity check* berdasarkan nilai baru dari bit yang dibalik.
- Langkah 5. Naikkan jumlah iterasi dengan 1

Selesai

e) Dekoding Sum-Product Algorithm (SPA)

SPA merupakan suatu algoritma dekoding *iterative based on belief propagation* (IDBP) yang sangat efisien untuk mendekoda kode LDPC. Seperti halnya algoritma dekoding MAP dan variasi-variasinya, SPA merupakan sebuah algoritma dekoding *soft-in soft-out* simbol ke simbol. SPA memproses simbol-simbol yang diterima secara iteratif untuk meningkatkan kehandalan dari masing-masing simbol kode terdekoda berdasarkan pada penjumlahan *parity check* yang dihitung dari *hard decision* dari simbol-simbol yang diterima dan matriks *parity check sparse* H dari sebuah kode LDPC. Kehandalan dari sebuah simbol terdekoda dapat diukur dengan probabilitas posteriori marjinalnya, *log-likelihood ratio* (LLR)-nya, atau nilai dari simbol yang diterima yang berkaitannya. Pengukuran-pengukuran kehandalan dari simbol-simbol kode yang terhitung pada bagian akhir dari masing-masing iterasi dekoding digunakan sebagai masukan untuk iterasi berikutnya. Proses iterasi dekoding tersebut berlanjut sampai sebuah kondisi penghentian (atau kriteria) terpenuhi. Kemudian, berdasarkan pada pengukuran-pengukuran kehandalan simbol-simbol kode terhitung tersebut, dilakukan *hard decision*.

Misalkan sebuah kode LDPC C dengan panjang n yang ditentukan oleh sebuah matriks *parity check* H dengan J baris, h_1, h_2, \dots, h_j , dimana

$$h_j = (h_{j,0}, h_{j,1}, \dots, h_{j,n-1})$$

Untuk $1 \leq j \leq J$, tentukan kumpulan indeks sebagai berikut untuk h_j :

$$B(h_j) = \{l: h_{j,l} = 1, 0 \leq l < n\} \quad (5)$$

yang dinamakan pendukung dari h_j .

Implementasi dari dekoding SPA adalah berdasarkan pada perhitungan dari probabilitas *posteriori* marjinal,

$$P(v_l|y)$$

untuk $0 \leq l < n$, dimana y merupakan sekuens *soft decision* yang diterima. Kemudian, LLR untuk masing-masing bit kode dinyatakan dengan

$$L(v_l) = \log \frac{P(v_l=1|y)}{P(v_l=0|y)} \quad (6)$$

Misalkan $p_l^0 = P(v_l = 0)$ dan $p_l^1 = P(v_l = 1)$ sebagai probabilitas prior dari $v_l = 0$ dan $v_l = 1$.

Untuk $0 \leq l < n$, $1 \leq j \leq J$, dan masing-masing $h_j \in A_l$, misalkan $q_{j,l}^{x,(i)}$ sebagai probabilitas kondisional dimana bit kode yang ditransmisikan v_l memiliki nilai x , diberikan penjumlahan check yang dihitung berdasarkan pada vektor-vektor check pada $A_l \setminus h_j$ pada iterasi dekoding ke- i . Untuk $0 \leq l < n$, $1 \leq j \leq J$, dan masing-masing $h_j \in A_l$, misalkan $\sigma_{j,l}^{x,(i)}$ sebagai probabilitas kondisional dimana jumlah-pemeriksa s_j dipenuhi (misal, $s_j = 0$), diberikan $v_l = x$ (0 atau 1) dan bit-bit kode lain dalam $B(h_j)$ memiliki distribusi terpisah $\{q_{j,t}^{v_l,(i)} : t \in B(h_j) \setminus l\}$; yaitu,

$$\sigma_{j,l}^{x,(i)} = \sum_{\{v_t : t \in B(h_j) \setminus l\}} P(s_j = 0 | v_l = x, \{v_t : t \in B(h_j) \setminus l\}) \cdot \prod_{t \in B(h_j) \setminus l} q_{j,t}^{v_t,(i)} \quad (7)$$

Nilai-nilai $\sigma_{j,l}^{x,(i)}$ yang terhitung tersebut kemudian digunakan untuk memperbaharui nilai-nilai dari $q_{j,l}^{x,(i+1)}$ sebagai berikut :

$$q_{j,l}^{x,(i+1)} = \alpha_{j,l}^{(i+1)} p_l^x \prod_{t \in B(h_j) \setminus l} \sigma_{j,t}^{x,(i)} \quad (8)$$

dimana $\alpha_{j,l}^{(i+1)}$ dipilih sehingga

$$q_{j,l}^{0,(i+1)} + q_{j,l}^{1,(i+1)} = 1$$

Nilai-nilai $q_{j,l}^{x,(i+1)}$ yang terhitung tersebut kemudian digunakan untuk memperbaharui nilai-nilai dari $\sigma_{j,l}^{x,(i+1)}$ berdasarkan pada persamaan (7). Pembaharuan di antara $q_{j,l}^{x,(i)}$ dan $\sigma_{j,l}^{x,(i)}$ dilakukan secara iteratif selama proses dekoding tersebut.

Pada langkah iterasi ke- i , probabilitas-probabilitas pseudo-posteriori diberikan oleh

$$P^{(i)}(v_l = x | y) = \alpha_l^{(i)} p_l^x \prod_{h_j \in A_l} \sigma_{j,l}^{x,(i-1)} \quad (9)$$

dimana $\alpha_l^{(i)}$ dipilih sehingga bahwa $P^{(i)}(v_l = 0 | y) + P^{(i)}(v_l = 1 | y) = 1$. Berdasarkan pada probabilitas-probabilitas tersebut, dapat dibentuk vektor sebagai berikut sebagai kandidat terdekoda:

$$z^{(i)} = (z_0^{(i)}, z_1^{(i)}, \dots, z_{n-1}^{(i)})$$

dengan

$$z^{(i)} = \begin{cases} 1, & \text{untuk } P^{(i)}(v_l = 1 | y) > 0,5 \\ 0, & \text{lainnya} \end{cases} \quad (10)$$

Kemudian, hitung $z^{(i)} \cdot H^T$. Apabila $z^{(i)} \cdot H^T = 0$, hentikan proses iterasi dekoding, dan keluaran $z^{(i)}$ sebagai *codeword* terdekoda. Berdasarkan penjelasan di atas dapat dirangkum langkah-langkah algoritma dekoding *sum-product algorithm* sebagai berikut:

Inisialisasi : Atur $i = 0$ dan jumlah maksimum iterasi ke I_{max} . Untuk setiap pasangan (j, l) sehingga bahwa $h_{j,l} = 1$ dengan $1 \leq j \leq J$ dan $0 \leq l < n$, atur $q_{j,l}^{0,(0)} = p_l^0$ dan $q_{j,l}^{1,(0)} = p_l^1$.

Langkah 1. Untuk $0 \leq l < n$, $1 \leq j \leq J$, dan masing-masing $h_j \in A_l$, hitung probabilitas-probabilitas $\sigma_{j,l}^{0,(i)}$ dan $\sigma_{j,l}^{1,(i)}$. Lanjutkan ke langkah 2.

Langkah 2. Untuk $0 \leq l < n$, $1 \leq j \leq J$, dan masing-masing $h_j \in A_l$, hitung nilai-nilai dari $q_{j,l}^{0,(i+1)}$ dan $q_{j,l}^{1,(i+1)}$ serta nilai-nilai dari $P^{(i+1)}(v_l = 0 | y)$ dan $P^{(i+1)}(v_l = 1 | y)$.

Bentuk $z^{(i+1)}$ dan uji $z^{(i+1)} \cdot H^T$. Apabila $z^{(i+1)} \cdot H^T = 0$ atau jumlah iterasi maksimum I_{max} dicapai, lanjutkan ke langkah 3. Selain itu, atur $i := i + 1$ dan kembali ke langkah 1.

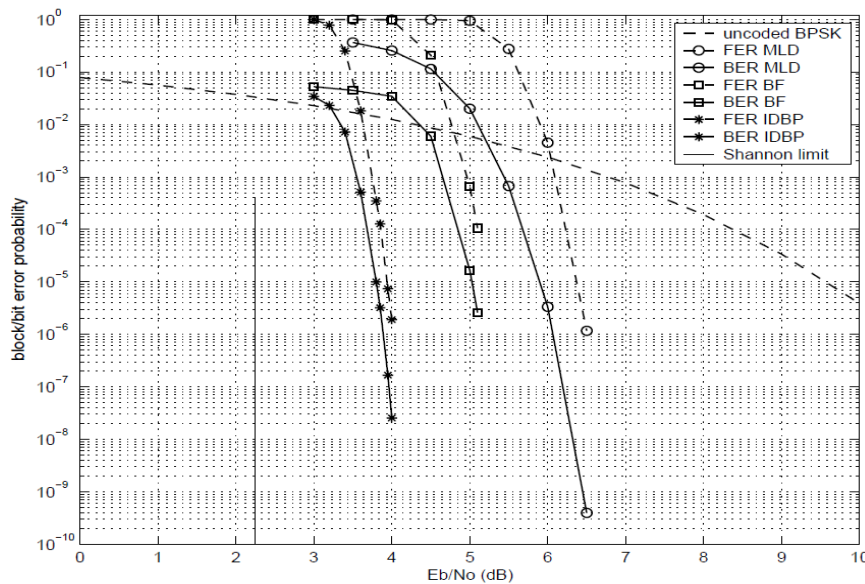
Langkah 3. Keluarkan $z^{(i+1)}$ sebagai codeword terdekod dan hentikan proses dekoding tersebut. Selesai

Dekoding MLG dan BF merupakan skema dekoding *hard decision* dan sangat mudah diterapkan. Dekoding MLG memiliki *delay* dekoding yang kecil dan dapat mencapai kecepatan dekoding yang sangat tinggi. Dekoding SPA merupakan skema dekoding *soft decision* yang membutuhkan kompleksitas perhitungan yang paling tinggi tetapi sangat baik kinerja perbaikannya. Weighted MLG dan Weighted BF merupakan skema dekoding yang berada di antaranya yaitu *hard decision* dan *soft decision* yang menawarkan kompleksitas perhitungan yang tidak terlalu tinggi dan baik dalam kinerja perbaikannya.

Tabel 1. Perbandingan kinerja perbaikan error dan kompleksitas perhitungan dari beberapa metode/ algoritma dekoding LDPC

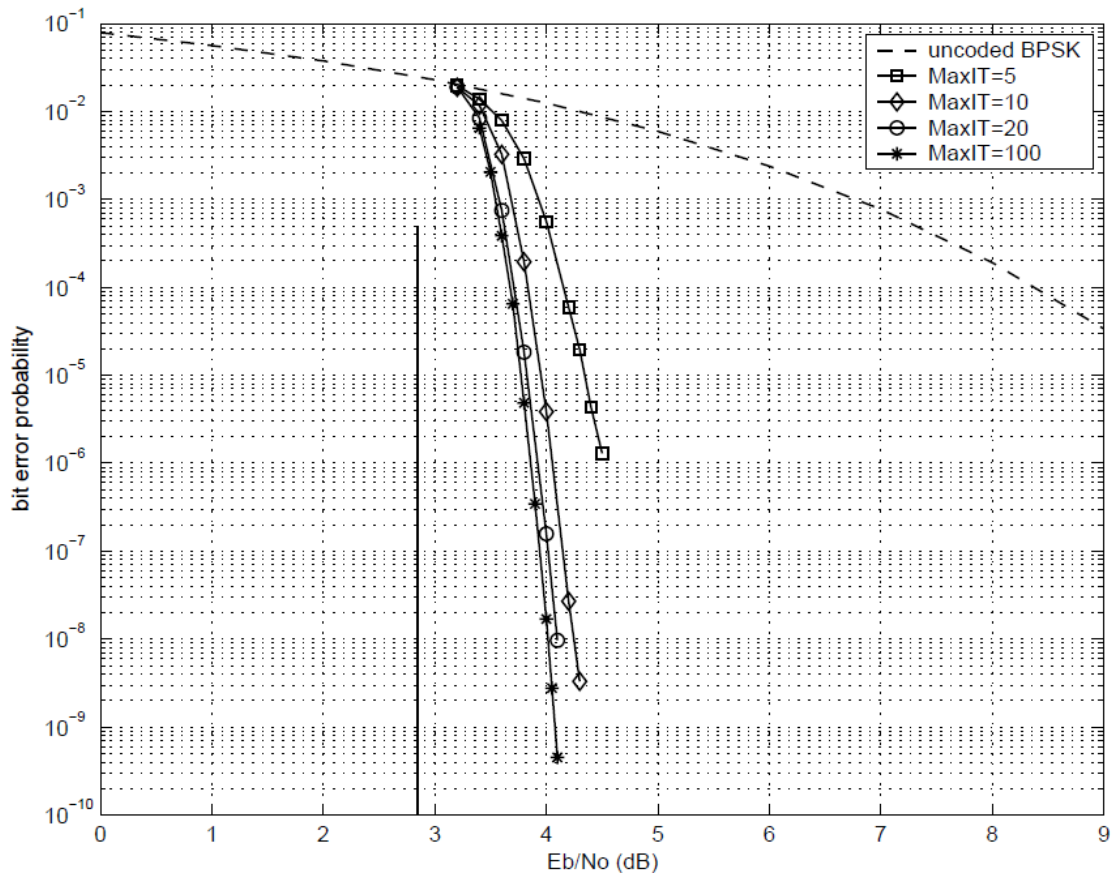
No.	Metode Dekoding	Jenis	Kinerja Perbaikan Error	Kompleksitas Perhitungan
1	One-step Majority-Logic	Hard Decision	■	■
2	Bit-Flipping	Hard Decision	■ ■	■ ■
3	Weighted Majority-Logic	Hard-Soft Decision	■ ■ ■	■ ■ ■
4	Weighted Bit-Flipping	Hard-Soft Decision	■ ■ ■ ■	■ ■ ■ ■
5	Sum-Product Algorithm	Soft Decision	■ ■ ■ ■ ■	■ ■ ■ ■ ■

Berikut beberapa kinerja penggunaan pengkodean LDPC dengan menggunakan beberapa metode / algoritma dekoding LDPC yang telah dijelaskan di atas.



Gambar 3.3. Probabilitas error bit dan blok dari LDPC code (4095,3367) yang didasarkan pada beragam algoritma decoding [11][12]

Gambar di atas menunjukkan kinerja pengkodean LDPC dengan beragam algoritma dekoding pada di atas BER 10^{-10} . Pada BER 10^{-8} , dekoding MLG mencapai *coding gain* 5,5 dB dari *uncoded* BPSK, sedangkan dengan IDBP (SPA) hampir mencapai *coding gain* 8 dB dari *uncoded* BPSK. *Error floor* tidak ada di atas BER 10^{-8} untuk *iterative decoding*.



Gambar 3.4. Probabilitas error bit dari LDPC code (8176,7156) dengan algoritma IDBP (SPA) melalui beberapa iterasi [11][12]

Terlihat pada gambar di atas, pada BER 10^{-8} , *gap* kinerja antara 20 iterasi dan 100 iterasi hanya 0,2 dB, sedangkan *gap* kinerja antara 20 iterasi dan 100 iterasi sekitar 0,05 dB. Untuk memperpendek *delay* dekoding dan mengurangi kompleksitas perhitungan, maka jumlah iterasi maksimum diatur menjadi 10 dan hanya kehilangan 0,2 dB.

4. KESIMPULAN

Teknik pengkodean *low density parity check* (LDPC) akan diterapkan pertama kalinya pada data satelit *Landsat Data Continuity Mission* (LDCM) yang akan diluncurkan pada tahun 2012. LDPC didesain berdasarkan matriks *parity check*nya yang bersifat *sparse*. Ada beberapa metode / algoritma yang digunakan dalam mendekoda LDPC yaitu *one-stop majority-logic* (MLG), *bit-flipping* (BF), *weighted majority-logic*, *weighted bit-flipping*, dan *sum-product algorithm* (SPA). Dari hasil kajian berdasarkan beberapa hasil penelitian algoritma SPA memberikan kinerja perbaikan error yang paling baik namun kompleksitas perhitungannya paling tinggi. Paper ini masih bersifat kajian, oleh karena itu perlu pengembangan lebih lanjut terkait kinerja teknik pengkodean LDPC yang dapat dilakukan dengan melakukan pengujian secara simulasi baik secara *software* maupun *hardware* dengan menggunakan *sample* data misi LDCM (jika telah tersedia). Diharapkan hasil kajian dapat berguna dalam proses ekstraksi rawdata satelit LDCM.

DAFTAR PUSTAKA

- CCSDS, *Low Density Parity Check Codes for Use in Near-Earth and Deep Space Application*. Research and Development for Space Data System Standards, Orange Book. Issue 2. Washington DC. September 2007
- Gallager, R.G. *Low Density Parity Check Codes*. IRE Transactions on Information Theory. 1962.

- Mackay, David J.C., M. Neal., *Near Shannon Limit Performance of Low Density Parity Check Codes*, Radford, 12 July 1996.
- Ricahrdsen, T., Urbanke, R. *Design of Capacity-Approaching Low Density Parity Check Codes*. IEEE Transf. Inform. Theory 47: 619-637, Februari 2001.
- Sun, Jian, *An Introduction to Low Density Parity Check (LDPC) Codes*. WCRL Seminar Series. West Virginia University. 3 Juni 2003.
- USGS website, dari <http://landsat.usgs.gov>.
- NASA & USGS, *LDCM Space to International Cooperator (IC) Interface Specification Document (ISD)*. Department of Interior US. Geological Survey. USA. September 2010.
- Daniels, Doug, *LDCM Science Data Reception*, NASA and USGS, USA, 23 Maret 2010.
- S. Lin, D.J. Costello, Jr, *Error Control Coding Book. 2nd ed.* New Jersey: Pearson Prentice Hall 2004.
- Balatsoukas-Stimming, Alexios. *Analysis and Design of LDPC Codes for the Relay Channel*. Technical university of Crete, 26 Februari 2010.
- Fong, Wai, dkk., *Low Density Parity Check Codes: Bandwidth Efficient Channel Coding*. NAS/GSFC.
- Fong, Wai. *White Paper for Low Density Parity Check (LDPC) Codes for CCSDS Channel Coding Blue Book*. NASA/GSFC, 2002

DAFTAR RIWAYAT HIDUP PEMAKALAH

Nama Penulis : Ali Syahputra Nasution
Tempat & Tanggal Lahir : Batang Kuis, 16 April 1983
Pendidikan Terakhir : S1 Teknik Telekomunikasi STT Telkom Bandung
Instansi : LAPAN
Alamat Kantor : Gedung LAPAN Satca Jl. LAPAN No.70
Pekayon, Pasar Rebo Jakarta Timur 13710, Indonesia
Telp : 021-8710786 Fax: 021-8717715
Alamat Rumah : Jl. Pondok Pesantren RT 06 RW 09,
Kelurahan Pasir Gunung Selatan, Kec. Cimanggis, Kota Depok
Jawa Barat
HP: 081315732372
Alamat E-mail : ali.syahputra@lapan.go.id
alisyahputra2003@yahoo.com
Seminar Terakhir Yang Diikuti : Seminar Ilmu Pengetahuan dan Teknologi Dirgantara XV
Motto Hidup : Tetap optimis, selagi kita masih hidup tidak ada yang tidak mungkin

Nama Penulis : Budhi Gustiandi
Tempat & Tanggal Lahir : Bandung, 7 Agustus 1982
Pendidikan Terakhir : S1 Teknik Elektro
Institut Teknologi Bandung
Instansi : LAPAN
Alamat Kantor : Gedung LAPAN Satca Jl. LAPAN No.70
Pekayon, Pasar Rebo Jakarta Timur 13710, Indonesia
Telp : 021-8710786 Fax: 021-8717715
Alamat Rumah : Kompleks Lapan Satca, Jl. Kalisari No 92c RT 11 RW 01
Kelurahan Pekayon, Kecamatan Pasar Rebo, Jakarta Timur
Jawa Barat
HP: 085697323796
Alamat E-mail : budhigustiandi@gmail.com
Seminar Terakhir Yang Diikuti : Seminar Ilmu Pengetahuan dan Teknologi Dirgantara XV
Motto Hidup : Knowledge is power, but character is more

Nama Penulis : Ayom Widipaminto
Tempat & Tanggal Lahir : Bandung, 2 Nopember 1974
Pendidikan Terakhir : S1 Teknik Elektro Institut Teknologi Bandung
S2 Teknik Elektro Universitas Indonesia
Instansi : LAPAN
Alamat Kantor : Jl. LAPAN No.70, Pekayon, Pasar Rebo Jakarta Timur 13710
Telp : 021-8710786
Alamat Rumah : Jl. Tapir IV No.10 Perumahan Cikarang Baru Jababeka
Kab. Bekasi Jakarta Timur 13710
HP: 08157109699
Alamat E-mail : ayomwidi@yahoo.com
Seminar Terakhir Yang Diikuti : Seminar Ilmu Pengetahuan dan Teknologi Dirgantara XV
Motto Hidup : Kerjakanlah pekerjaan yang membawa berkah bagimu dan orang yang kamu cintai

HASIL DISKUSI DALAM PELAKSANAAN SEMINAR

Pertanyaan :

1. Satelit penginderaan jauh mana saja yang telah menerapkan pengkodean dengan menggunakan *Low Density Parity Check* (LDPC) tersebut ?
2. Apakah pada saat penerapan nantinya di satelit *Landsat Continuity Data Mission* (LDCM), pengkodean *Low Density Parity Check* (LDPC) akan digabungkan dengan pengkodean *Forward Error Correction* (FEC) lain seperti pengkodean *Reed Solomon* atau hanya berdiri sendiri ?
3. Menurut informasi yang saya peroleh dari poster, disebutkan bahwa pengujian pengkodean LDPC tersebut akan dilakukan setelah tersedia data simulasi, kapan kira-kira hal tersebut dilakukan ?
4. Akankah LDPC ini diterapkan pada satelit-satelit penginderaan jauh generasi mendatang lainnya selain LDCM ?
5. Hal apa yang menjadi konsekuensi ketika *coding gain* mencapai 9 dB (yang merupakan hal yang sangat baik) ?
6. Skema dekoding mana yang disarankan oleh dokumen yang dikeluarkan oleh pihak USGS ?
7. Apa hubungan penggunaan algoritma dekoding LDPC dengan spesifikasi kebutuhan demodulator ?

Jawaban :

1. Pada saat ini belum ada satelit penginderaan jauh yang menerapkan pengkodean LDPC tersebut, rencananya pengkodean tersebut akan pertama kali diterapkan pada satelit LDCM yang rencananya akan diluncurkan pada bulan Desember 2012.
2. Pada saat penerapan nanti di LDCM, hanya akan menggunakan FEC LDPC dan tidak digabungkan dengan FEC lain.
3. Kami telah berusaha untuk selalu mendapatkan informasi terbaru mengenai LDCM, terutama terkait dengan format datanya, namun menurut informasi terakhir yang kami peroleh belum ada kepastian ketersediaan data simulasi dari pihak NASA atau USGS. Kami perkirakan data simulasi tersebut akan tersedia pada pertengahan tahun 2012 sebelum LDCM diluncurkan di akhir tahun yang sama.
4. Di konsorsium *Consultative Committee for Space Data System* (CCSDS), LDPC telah diterima sebagai salah satu standar pengkodean data satelit, namun mengenai penerapannya akan diserahkan kembali kepada para pihak pembuat satelit yang bersangkutan. Namun, apabila LDPC tersebut telah sukses diterapkan pada LDCM yang akan diluncurkan pada akhir tahun 2012, kami yakin bahwa satelit-satelit penginderaan jauh generasi mendatang akan mengadopsi teknik pengkodean serupa karena kinerjanya diperkirakan lebih baik daripada teknik-teknik pengkodean yang ada sekarang ini.

5. Panjang bit *parity* yang digunakan akan semakin panjang sehingga akan mengorbankan *bandwidth* yang dapat digunakan.
6. USGS belum mengeluarkan pernyataan secara resmi mengenai skema dekoding mana yang akan mereka gunakan untuk struktur data LDCM.
7. Semakin rumit algoritma yang digunakan, maka semakin tinggi pula kebutuhan spesifikasi teknis *hardware* yang akan digunakan, karena berkaitan dengan kapasitas gerbang logika yang tersedia pada masing-masing IC yang digunakan pada demodulator yang berbeda.