Perbandingan Kinerja LDPC (*Low Density Parity Check*) Dengan Metode Decoding *Bit Flip* Pada Kanal AWGN dan Kanal *Multipath Fading*

Beny Nur Prasetyo[1], Ir.Yoedy Moegiharto MT[2]
[1]Mahasiswa Politeknik Elektronika Negeri Surabaya, Jurusan Teknik Telekomunikasi
[2]Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember
Kampus ITS, Surabaya 60111
e-mail : ajib_coy@yahoo.com e-mail : yoedy@eepis-its.edu

ABSTRAK

Sebagai efek dari perkembangan jaman, teknologi telekomunikasi khusunya dalam sistem komunikasi *wireless* dituntut untuk dapat menyediakan layanan data yang berkecepatan tinggi (*high data rate*) dengan *Quality of Service* (QOS) yang *reliable* dengan kata lain memiliki *Bit Error Rate* (BER) yang kecil dengan daya sekecil mungkin. Masalah yang timbul dalam komunikasi bergerak adalah adanya AWGN dan *multipath fading* yang mengakibatkan adanya kesalahan data yang diterima.

Tugas akhir ini membandingkan performansi teknik pengkodean *Low Density Parity Check* (LDPC) pada kanal AWGN dan *multipath fading*. Metode *encoding* yang digunakan pada sisi *transmitter* dengan menggunakan metode *sparse matriks* berbasis *Approximation Lower Triangular* dengan *coderate* ½, sedangkan pada sisi *receiver* menggunakan metode decoding *bit flip*.

Dengan perbedaan ukuran matriks, hasil yang dicapai tidak terlalu ada perbedaan dalam kinerja LDPC. Pada perbedaan nilai iterasi, semakin banyak iterasi, mempengaruhi kinerja LDPC sehingga hasil yang dicapai semakin baik. Target BER 10^{-5} pada kanal AWGN tercapai pada rata-rata nilai 2,058 dB, sedangkan pada kanal *multipath fading* pada nilai 6,2 dB sehingga kinerja *Low Density Parity Check Code* melalui kanal AWGN lebih bagus dibandingkan saat melewati *Multipath fading* dengan metode *decoding bit flip*.

Kata Kunci : *Low Density Parity Check Code, sparse matriks berbasis Approximation Lower Triangular, Kanal AWGN, Multipath fading, decoding bit flip, BER (Bit Error Rate*)

1. PENDAHULUAN

Pada zaman sekarang semakin banyak kebutuhan yang diperlukan dalam dunia teknologi. Baik dari bentuk komunikasi, system yang handal, integrasi *wireless* dengan berbagai layanan yang bermacam-macam. Pada proses pengiriman maupun penerimaan data semakin besar ukuran data yang dibutuhkan,yang mana semakin besar pula resiko kesalahan dalam transmisi data tersebut. Sehingga diperlukan proses dalam penyeleksian sehingga data yang dikirim dapat sampai pada penerima dengan baik. Apalagi saat seperti ini banyak terdapat gangguan atau *noise* yang ditimbulkan oleh lingkungan sekitar terhadap terjalinnya sebuah komunikasi.

Sehingga di perlukan suatu proses untuk mengoreksi dan mendeteksi atau disebut error control coding agar data atau informasi pada saat dikirim maupun di terima dapat menghasilkan dengan baik. Banyak metode yang di gunakan untuk mengoreksi dan mendeteksi kesalahan yang tejadi. Terdapat pula macam-macam karakteristik kelebihan dan kekurangan dari setiap metode tersebut.

Salah satu metode tersebut adalah *Low Density Parity Check (LDPC) Code*. *Low Density Parity Check Code* adalah salah satu kelas dari linear block kode yang memiliki kepadatan rendah,nama itu berasal dari karakteristik *parity check*-nya yang hanya berisi sedikit bit "1" jika dibandingkan dengan jumlah bit "0". Keuntungan dari penggunaan LDPC adalah dapat menyediakan performa yang sangat mendekati nilai kapasitas dari berbagai macam kanal dan mempunyai proses decoding yang linier dan diharapkan dengan pengiriman *high-bit-rate* dapat membantu untuk menghasilkan probabilitas kesalahan bit yang rendah.

Low density Parity Check Code juga dapat digunakan untuk pengiriman dan penerimaan data yang berukuran besar. Dengan metode ini akan di implementasikan dengan melewati sebuah kanal *AWGN* dan kanal *multipath fading*. Untuk dapat mengetahui bagaimana pengaruh yang didapat ketika sebuah

pendeteksian dan pengkoreksian kesalahan dilewatkan sebuah kanal.

2. DASAR TEORI

Latar Belakang LDPC

LDPC pertama kali ditemukan oleh Galagher pada 1960 pada doctoral dissertation dan hampir tidak dianggap lagi sekitar 35 tahun kemudian. Kemudian muncul Tanner tahun 1982 yaitu mengembangkan LDPC code dan memperkenalkan representasi grafik LDPC code, atau dikenal dengan *Tanner Graph*. Pembelajaran tentang LDPC code dihidupkan kembali sekitar 1990, yang dikerjakan oleh MacKay, Luby, dan lain-lain. Ketika pertama kali ditemukan, karena batas dari daya computer saat itu yang belum tinggi, kode ini dianggap tidak praktis, dan terlupakan. Saat ini dengan kehebatan daya computer yang telah hadir, kode LDPC telah digunakan kembali dan termasuk kedalam kode dengan performa yang terbaik pada situasi sinyal rendah ke-noise. Ternyata dengan system Gallagher merupakan keuntungan dari linear block kode yang menggunakan parity check matrik dengan *sparse* (kerapatan rendah atau jumlah elemen *non-zero* yang sedikit).

Representasi Code LDPC

a. Representasi matriks

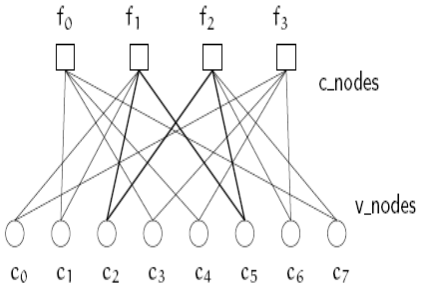
LDPC adalah *linear block kode* dimana *parity check matriks* H mempunyai kerapatan rendah yang mempunyai nilai bit 1, regular LDPC adalah linear block kode yang memiliki parity *check matriks* H yang berisi ωc ,jumlah bit 1 pada tiap kolom dan $\omega r = \omega c(n/m)$ adalah jumlah bit 1 pada tiap baris. Dimana $\omega c << m$ (*equivalent*, $\omega c << m$). Kode rate $R=k/n$ dihubungkan dengan parameter melalui $R=1-\omega c/\omega r$ (diasumsikan H adalah full rank). Jika H adalah *sparse* atau kerapatanny rendah ,tapi jumlah dari bit 1 pada tiap baris atau kolom tidak konstan disebut dengan irregular LDPC.

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

Gambar 1. Representasi matriks

b. Representasi grafik

Tanner memperkenalkan sebuah representasi grafik untuk code-code LDPC. Grafik *Tanner* adalah *bipartite graph* yaitu suatu grafik yang yang tidak langsung berhubungan tetapi memisahkan antara dua kelas, dimana *edge* hanya terhubung dengan dua node tetapi tidak bertempat pada kelas yang sama. Ada dua type nodes dalam Graph Tanner, yaitu *variable nodes (v-nodes)* dan *check nodes (c-nodes).*dan yang menyambungkan antara *v-node* dan *c-node* disebut *edge.*



Gambar 2. Representasi grafik

Pada gambar 2 grafik tanner menghubungkan H. Dengan mengamati c_1, c_3, c_4dan c_7 dihubungkan dengan *c-node* yang disebut f_0 pada , $h_{01}, h_{03}, h_{04}, h_{07} =1$ (sedangkan yang lain adalah nol). Kemudian langkah selanjutnya yaitu c-node f_1, f_2, f_3, mengikuti baris 1, 2, 3 pada H. Dengan catatan dengan mengikuti persamaan $cH^T=0$. Nilai bit yang dihubungkan pada *c-node* yang sama harus berjumlah 0. Kita dapat memproses terus kolom untuk membangun grafik *tanner*

Regular dan irregular code LDPC

Code LDPC disebut dengan *regular* jika ω_c adalah bernilai konstan Pada untuk setiap kolom dan ω_r pada setiap barisnya. Pada contoh adalah *regular* : tiap *v node* memiliki dua koneksi *edge* dan tiap *c-node* memiliki 4 koneksi *edge* (pada tiap v-node adalah 2 dan tiap *c-node* ada 4). Jika H adalah *low density* tetapi jumlah 1 pada tiap baris ataupun kolom tidak konstan dinamakan code *irregular* LDPC.

Dimana
ω_c adalah jumlah bit 1 pada tiap kolom
$\omega_r = \omega_c (n/m)$ adalah jumlah bit 1 pada tiap baris
$R = 1- \omega_c/\omega_r$ adalah kode rate

Konstruksi code LDPC

Ada banyak cara untuk menkonstruksi code LDPC antara lain

1. Code Gallagher
2. Code Mackay

Decoding LDPC

1. *Bit flipping* (BF*) decoding*
2. *Sum product decoding*

Proses Decoding LDPC

a. Hard Decision decoding (bit flip)

Ilustrasi proses decoding *iterative* menggunakan *bit flip*, didasarkan pada penandaan *hard decision* (0 atau 1) untuk setiap bit yang diterima. Bagian yang penting dari *decoding iterative* adalah pelewatan pesan (*message passing*) antar node pada grafik tanner. Pada decoding *bit flip* pesannya sederhana: sebuah message node mengirimkan pesan pada setiap *check node* yang terhubung mendeklarasikan dirinya 0 atau 1. Dan setiap *check node* mengirimkan pesan ke setiap *message node* yang terhubung pada grafik tanner, apakah cek paritas terpenuhi atau tidak, decoding *sum product* LDPC memiliki prosedur yang sama, hanya saja pesan yang di pertukarkan lebih kompleks.

decoding *bit flip* :

1) Langkah 1: inisialisasi: setiap *bit node* menandai nilai bit yang diterima kanal dan mengirimkan pesan ke *check node* yang terhubung pada grafik *tanner* mengindikasikan nilainya.
2) Langkah 2: *Parity update*: dengan menggunakan pesan dari *bit 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 *bit node* yang terhubung mengindikasikan apakah persamaan cek paritas terpenuhi atau tidak.
3) Langkah 3 : *Bit update* ; jika mayoritas pesan yang diterima setiap *message node* tidak memenuhi, *message 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. Log domain Sum Product decoding

Proses log domain decoding algoritm sebagai berikut:

1. checks to bit

Setiap check node m mengumpulkan seluruh informasi yang masuk, $L(q_{\iota-m})$'s dan memperbaharui kekuatan bit 1 berdasarkan pada informasi seluruh bit yang dihubungkan pada *check node* m

$$L(\acute{r}_{m-\iota}) = 2 \tanh^{-1} \prod_{\iota \in L(m)\backslash \iota} tanh\left(\frac{L(q\iota - m)}{2}\right)$$

2. Bits to *checks*

$$L(q\iota\text{-}_m) = L(p\iota) + \sum_{m' \; EM(\iota)\backslash m} L(\acute{r}m - \iota)$$

3.*Checks stop criterion*

Proses decoder menghasilkan peluang a posteriori untuk bit l dengan menjumlahkan informasi dari seluruh check nodes yang menghubungkan bit l

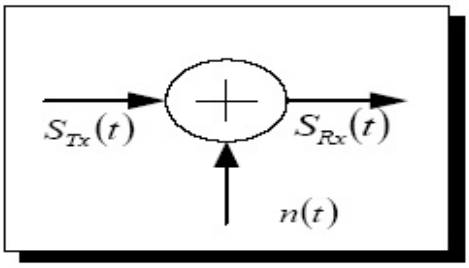
$$L(q\iota\text{-}_m) = L(p\iota) + \sum_{m \quad EM(\iota)\backslash m} L(\acute{r}m - \iota)$$

Hard decision dibuat berdasarkan L(L(qι) dan menghasilkan pendekodean masukan x yang dapat diperiksa terhadap matriks parity check H.Jika Hx = 0 atau iterasi maksimum maka proses decoder berhenti dan keluarannya adalah x, Jika sebaliknya maka proses decoder akan kembali pada step 1-3

Kanal *AWGN (Additive White Gaussian Noise)*

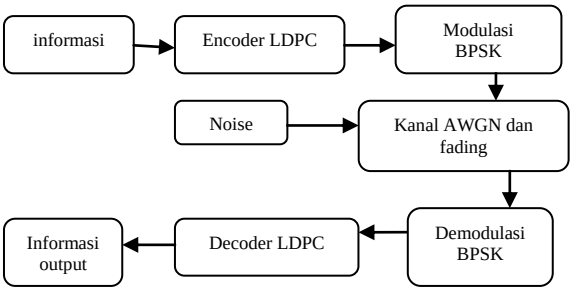
Kanal *AWGN* merupakan model kanal yang sering digunakan dalam berbagai aplikasi komunikasi. Noise ini bersifat white karena rapat spektral

dayanya konstan untuk semua frekuensi. Noise ini juga bersifat menjumlahkan dan merupakan suatu proses acak dengan distribusi gaussian. Jadi *AWGN* merupakan noise yang bersifat menjumlahkan, yang mempunyai fungsi rapat spektral daya yang konstan untuk semua frekuensi dan probabilitas berdistribusi gaussian. Kanal *AWGN* merupakan model kanal sederhana dan umum dalam suatu sistem komunikasi. Model kanal ini dapat di lihat seperti gambar dibawah :



Gambar 3. Kanal AWGN

3. Perancangan sistem

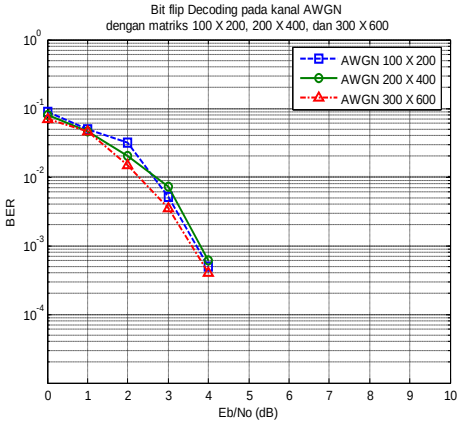


Gambar 4. Blok diagram LDPC

Dari diagram blok diatas dapat dilihat bahwa informasi yang berupa deretan bit-bit akan dimasukkan ke encoder LDPC, kemudian *output encoder* akan dimodulasi dengan menggunakan modulasi BPSK yang kemudian dikirim melalui kanal *AWGN*, disini kanal *AWGN* akan ditambahkan dengan noise. Keluaran kanal *AWGN* bercampur noise akan diterima oleh *decoder* di sisi terima yang sebelumnya didemodulasi terlebih dahulu, dari *output decoder* LDPC akan dihasilkan output. Bahasa pemrograman yang digunakan pada perancangan sistem ini berupa software mathlab. Jadi semua blok pada perancangan system ini berupa software under matlab.

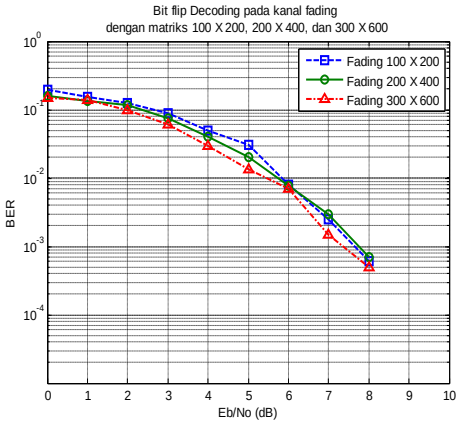
Informasi yang dikirimkan berupa data biner untuk *encoder* LDPC data informasi yang berupa data biner dikodekan kembali menjadi bit-bit *codeword*, *codeword* inilah yang kemudian akan dikirim atau diproses melalu kanal *AWGN* dengan ditambah *noise*, *codeword* ditambah *noise* tersebut akan diterima oleh *decoder* LDPC, disini data *codeword* bercampur *noise* tadi dikodekan kembali dan menghasilkan output, output ini akan berupa bit-bit info

4.Analisa Hasil



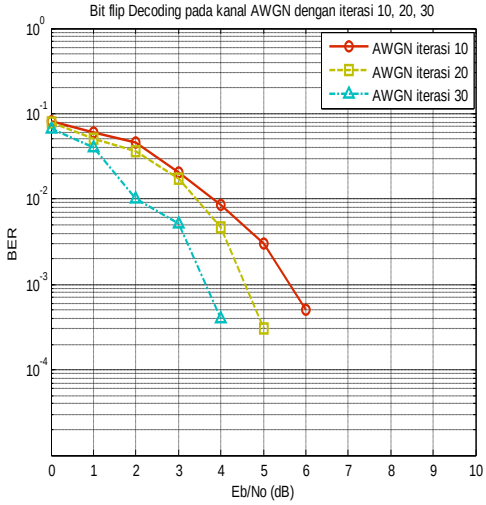
Gambar 5 Grafik Sistem LDPC dengan matriks yang berbeda menggunakan metode *decoding bit flip* pada kanal AWGN

Gambar 5 merupakan hasil dari simulasi yang dilakukan dengan menggunakan *bit flip* sebagai dekoding pada simulasi kinerja pengkodean LDPC, ukuran matriks 100 X 200 pada kanal AWGN dengan target BER 10^{-3} dapat dicapai pada EbNo = 3,70 dB, untuk ukuran matriks 200 X 400 target BER tercapai pada EbNo = 3,79 dB, pada ukuran matriks 300 X 600 target BER tercapai pada EbNo = 3,57 dB.



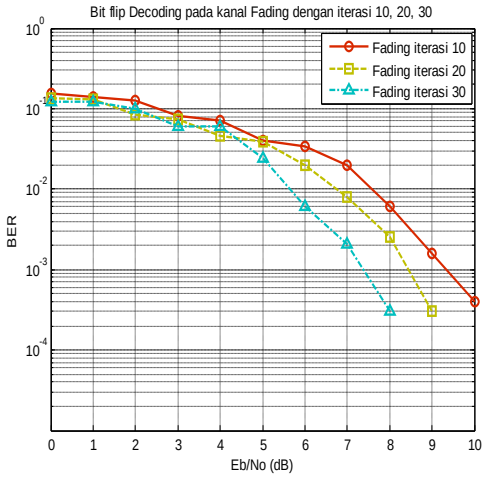
Gambar 6 Grafik Sistem LDPC dengan matriks yang berbeda menggunakan metode *decoding bit flip* pada kanal *fading*

Gambar 6 merupakan hasil dari simulasi yang dilakukan dengan menggunakan *bit flip* sebagai dekoding pada simulasi kinerja pengkodean LDPC, ukuran matriks 100 X 200 pada kanal *fading* dengan target BER 10^{-3} dapat dicapai pada EbNo = 7,64 dB, untuk ukuran matriks 200 X 400 target BER tercapai pada EbNo = 7,75 dB, pada ukuran matriks 300 X 600 target BER tercapai pada EbNo = 7,37 dB.



Gambar 7 Grafik Sistem LDPC dengan jumlah iterasi yang berbeda menggunakan metode *decoding bit flip* pada kanal AWGN.

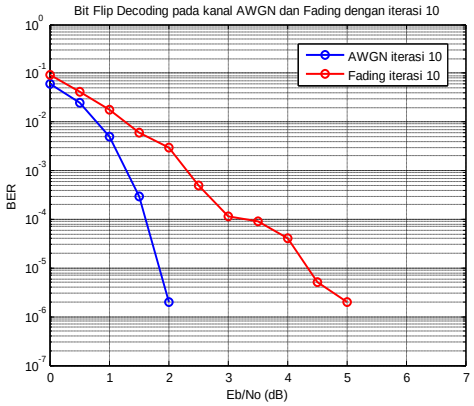
Gambar 7 merupakan hasil dari simulasi yang dilakukan dengan menggunakan *bit flip* sebagai dekoding pada simulasi kinerja pengkodean LDPC, ukuran matriks 100 X 200 pada kanal AWGN dengan pengamatan perbandingan nilai iterasi yang digunakan. Pada BER 10^{-3} dengan iterasi 10 di EbNo = 5,60 dB, pada iterasi 20 di EbN0 = 4,57 dB, pada iterasi 30 di EbNo = 3,64 dB



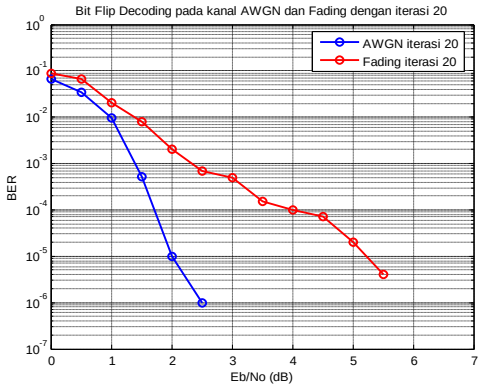
Gambar 8 Grafik Sistem LDPC dengan jumlah iterasi yang berbeda menggunakan metode *decoding bit flip* pada kanal *fading.*

Gambar 8 merupakan hasil dari simulasi yang dilakukan dengan menggunakan *bit flip* sebagai dekoding pada simulasi kinerja pengkodean LDPC, ukuran matriks 100 X 200 pada kanal *fading* dengan pengamatan perbandingan nilai iterasi yang digunakan. Pada BER 10^{-3} dengan iterasi 10 di EbNo = 9,34 dB, pada iterasi 20 di EbN0 = 8,43 dB, pada iterasi 30 di EbNo = 7,38 dB

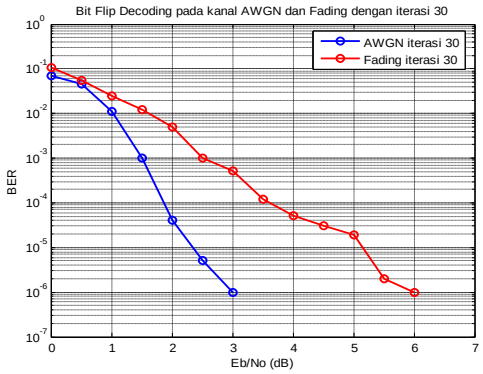
Jika diperhatikan pada setiap kenaikan jumlah iterasi *decoding*, maka dapat dilihat terjadi perbaikan performansi berupa *coding gain*. Hasil yang didapatkan berdasarkan grafik tersebut yaitu nilai optimal untuk iterasi *decoding* adalah 30. Karena untuk iterasi 10 dan 20 terjadi peningkatan *coding gain* yang tidak terlalu besar. Meskipun dari hal ketiga iterasi yang digunakan terjadi peningkatan *coding gain* sebesar 1 dB.



Gambar 9 Perbandingan kanal AWGN dan *Fading* dengan *Iterasi* 10.



Gambar 10 Perbandingan kanal AWGN dan *Fading* dengan *Iterasi* 20.



Gambar 11 Perbandingan kanal AWGN dan *Fading* dengan *Iterasi* 30.

5. KESIMPULAN

Berdasarkan pada hasil simulasi dan analisa yang telah dibahas pada bab 4, dapat diketahui kinerja dari LDPC yang dihasilkan adalah dalam bentuk ukuran matriks saat melewati kanal AWGN dan *Rayleigh fading*, dan perbandingan nilai iterasi pada kanal AWGN dan *Rayleigh fading*. Maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Perbedaan ukuran matriks tidak terlalu mempengaruhi hasil error yang dihasilkan karena sifat dari LDPC yaitu semakin besar ukuran matriks yang digunakan maka jumlah bit '1' semakin rendah. Hasil simulasi pada sistem LDPC dengan ukuran matriks 100 X 200 pada kanal AWGN dengan target BER 10^{-5} dapat dicapai pada Eb/No = 1,96 dB, sedangkan untuk ukuran matriks 200 X 400 target BER tercapai pada Eb/No = 1,92 dB, pada ukuran matriks 300 X 600 target BER tercapai pada Eb/No = 1,896 dB.
2. Pada saat iterasi 30 memiliki performa yang paling bagus dalam menghasilkan nilai BER karena nilai iterasi 30 membutuhkan proses delay yang lebih lama daripada saat iterasi 10 dan iterasi 20.
   Hasil simulasi pada sistem LDPC dengan nilai iterasi 10 pada kanal AWGN dengan target BER 10^{-5} dapat dicapai pada Eb/No = 1,84 dB, sedangkan untuk iterasi 20 target BER tercapai pada EbNo = 2 dB, pada nilai iterasi 30 target BER tercapai pada Eb/No = 2,333 dB.

6. SARAN

Dalam pemodelan simulasi LDPC saat melewati kanal AWGN dan *Rayleigh Fading* banyak terdapat pengembangan yang dapat dilakukan.

1. Bit informasi yang dikirimkan harus lebih banyak lagi tetapi proses yang dibutuhkan untuk mendapatkan probabilitas kesalahan bit akan membutuhkan waktu yang cukup lama.
2. Menggunakan jenis metode *decoding* yang lain agar memberikan perbedaan hasil BER.

7. DAFTAR PUSTAKA :

[1] Sarah J.Johnson, Steven R. Weller, Low Density parity-check codes: Design and decoding. School of Electrical Engineering and Computer Science University of Newcastle. 2002

[2] G.Gallager, Low-Density Parity-Check Codes, Cambridge, MA:M.I.T, Press. 1963

[3] Radford M Neal. Sparse Matrix Methods and Probabilistic Interference Algoritma,Dept. of Statistics and Dept. of Computer Science University of Toronto. 1999

[4] Feras A.K Al-Zuraiqi, " Analysis, Simulation and Modelling Of Mobile and Fixed Fading Channel", June. 2004

[5] Sun Jian," An Introduction to Low Density Parity Check (LDPC) Codes", June 2003

[6] Bernhard M.J. Leiner, LDPC Codes– a brief Tutorial, April. 2005

[7] William E. Ryan, " An Introduction to LDPC codes", August. 2003

[8] Marius Pop, "Statistical Analysis of Sum-of-Sinusoids Fading Channel Simulators", February, 1999.

[9] R. H. Clarke, **"A** Statistical Theory of Mobile-Radio Reception", Bell ***Syst.*** *Tech. J.*, pp. 957- 1000, Ju1.-Aug. 1968.

[10] L. W. Couch, "Digital and Analog Communication Systems", Prentice Hall, 1997