

ABSTRAK

Sinyal OFDM terdiri atas penjumlahan sejumlah subcarrier yang dimodulasi secara independen. Apabila fasa setiap *subcarrier* sama maka akan dihasilkan daya sinyal maksimum sebesar N kali daya rata-ratanya. Perbandingan antara daya sinyal maksimum dengan daya rata-ratanya biasa disebut sebagai *Peak to Average Power Ratio* (PAPR). Nilai PAPR yang besar pada OFDM membutuhkan amplifier dengan *dynamic range* yang lebar untuk mengakomodasi amplitudo sinyal. Jika hal ini tidak terpenuhi maka akan terjadi distorsi linear yang menyebabkan subcarrier menjadi tidak lagi orthogonal dan pada akhirnya menurunkan kinerja OFDM.

Untuk mengatasi PAPR yang tinggi ini adalah dengan meminimalkan nilai PAPR pada sinyal OFDM yang akan ditransmisikan. Dengan kata lain, diperlukan skema pereduksi PAPR pada bagian transmitter sinyal OFDM. Ada beberapa teknik untuk menurunkan nilai PAPR yaitu, *Partial Transmit Sequence* (PTS), *Selective Mapping* (SLM), *Clipping*, *Interleaving*, *Sub Block Coding*, *Peak Windowing*.

Pada tugas akhir ini, dibuat implementasi reduksi PAPR dengan metode *clipping* menggunakan DSK TMS320C6416T. TMS320C6416 merupakan mikroprosesor berkecepatan tinggi dengan tipe arsitektur yang sesuai digunakan untuk mengolah sinyal secara real time. Setelah dilakukan simulasi, implementasi dan analisa, dapat disimpulkan bahwa metode *clipping* dapat menurunkan nilai PAPR sebesar 0.3 % sampai 50%. Nilai penurunan PAPR tersebut tergantung pada level *clipping* dan jumlah *subcarrier* yang digunakan. Semakin besar reduksi PAPR, nilai BER yang dihasilkan semakin bagus. Nilai BER yang dihasilkan dari reduksi PAPR menggunakan metode *clipping* ini bisa mencapai 10^{-5} .

Kata kunci : *OFDM, PAPR, Clipping, DSK TMS*

[Halaman ini sengaja dikosongkan]

ABSTRACT

OFDM signal consists the sum of a number of independently modulated subcarriers. If the phase of each subcarrier are equals, it will produce the maximum signal power amounts N times of average power. Comparison between the maximum signal power and the average power commonly referred as Peak to Average Power Ratio (PAPR). The huge PAPR value on OFDM requires an amplifier with a wide dynamic range to accommodate the signal amplitude. If this is not complete then linear distortion will happen which causes the subcarrier is no longer orthogonal and will reduce the performance of OFDM in the end.

To solve the high PAPR value is minimizing the PAPR value of the OFDM signal which will be transmitted. It requires the scheme PAPR reducer on OFDM signal transmitter. There are several techniques to reduce the PAPR value, Partial Transmit Sequence (PTS), Selective Mapping (SLM), Clipping, Interleaving, Sub Block Coding, Peak Windowing.

In this final project, the implementation of the PAPR reduction using clipping method on DSK TMS320C6416T. TMS320C6416 is a high-speed microprocessor with suitable architecture type is used to process signals in real time. After the simulation, implementation and analysis can be concluded that the method of clipping can reduce the PAPR value by 0.3% to 50%. The value of PAPR reduction depends on the clipping level and amounts of used subcarrier. The more PAPR reduction, the result of BER value is better. Value BER is got from PAPR reduction using this clipping method until 10^{-5} .

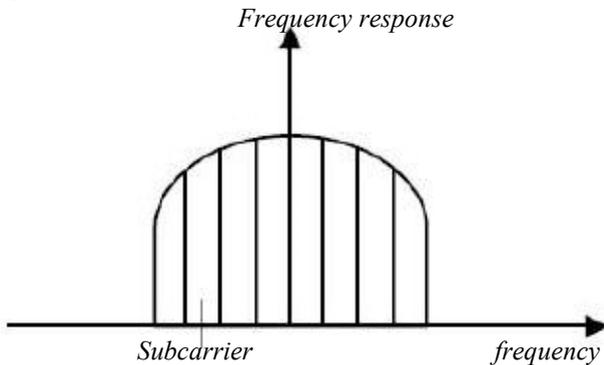
Key word : *OFDM, PAPR, Clipping, DSK TMS*

[Halaman ini sengaja dikosongkan]

BAB II TINJAUAN PUSTAKA

1.1 *Orthogonal Frequency Division Multiplexing (OFDM)*

Orthogonal Frequency Division Multiplexing (OFDM) adalah satu jenis skema *multicarrier modulation* yang antara *subcarrier* yang satu dengan yang *subcarrier* lainnya saling *overlap*. Spektrum frekuensi OFDM dapat dilihat pada Gambar 2.1. Dalam OFDM, data masukan dialirkan ke beberapa *subcarrier* paralel ortogonal yang memiliki laju data yang lebih rendah.

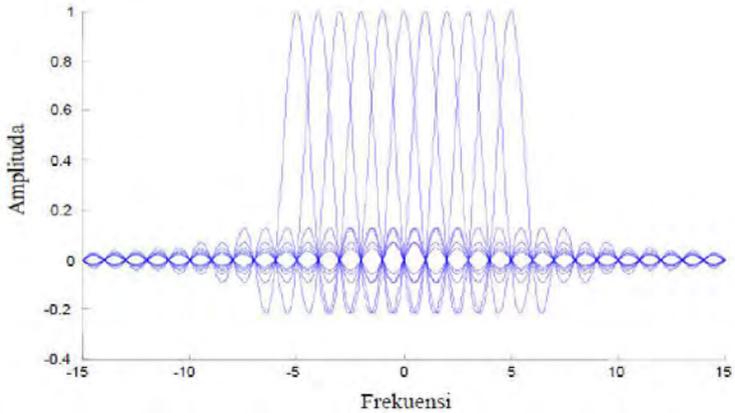


Gambar 2. 1 Carrier dibagi menjadi beberapa *subcarrier* [1]

OFDM sebagai salah satu aplikasi skema transmisi data paralel, yang mengurangi pengaruh dari *multipath fading* dan menghilangkan penggunaan ekualiser kompleks. Dalam sistem *multicarrier*, *fading*, atau *interferensi* akan mempengaruhi sedikit dari *subcarrier* dan *Error Correction Coding (ECC)* dapat digunakan untuk memperbaiki error dari *subcarrier* tersebut.

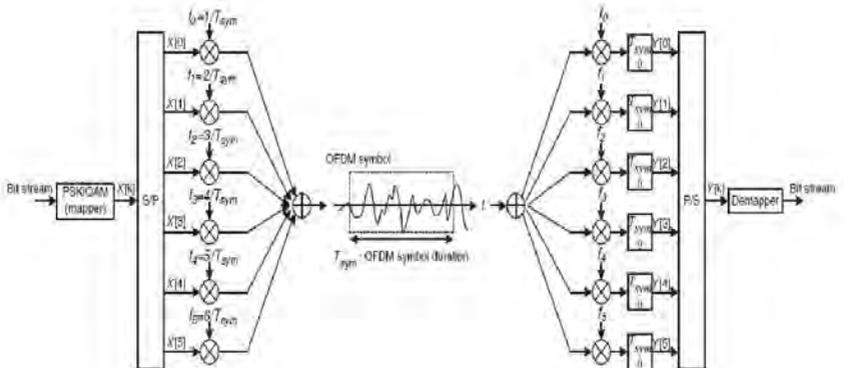
Secara matematis, ke-*orthogonal*-an OFDM dapat dijelaskan sebagai berikut. Misalkan fungsi $\varphi_m(t)$ adalah *subcarrier* ke- m dan $\varphi_n(t)$ adalah *subcarrier* ke- n . Kedua *subcarrier* ini dikatakan saling ortogonal satu sama lain pada interval $a < t < b$ jika memenuhi kondisi :

$$\int_a^b \varphi_m(t) \times \varphi_n(t) dt = 0 \quad (2.1)$$



Gambar 2. 2 Spektrum Frekuensi OFDM [1]

OFDM dapat juga dikatakan variasi dari FDM (*Frekuensi Division Multiplexing*), yang menempatkan sinyal-sinyal *subcarrier* sedekat mungkin, sehingga dapat menghemat pemakaian lebar bandwidth, namun tetap mempertahankan sifat ortogonal antar sinyal. Gambar 2.3 adalah blok diagram OFDM secara garis besar.



Gambar 2. 3 Blok Diagram OFDM [1]

Kelebihan skema transmisi OFDM adalah :

- ✓ OFDM adalah cara efektif untuk menangani *frequency selective fading*, penyebaran tundaan (*delay*), kompleksitas implementasi lebih rendah dibandingkan sistem *single carrier* yang menggunakan ekualiser.
- ✓ Pada kanal yang relatif bervariasi terhadap waktu, dimungkinkan untuk meningkatkan kapasitas secara signifikan dengan mengubah *data rate* per *subcarrier* berdasarkan SNR (*signal to noise ratio*) dari *subcarrier* tersebut.
- ✓ OFDM tahan terhadap interferensi *narrowband* karena efek interferensi hanya beberapa persen dari *subcarrier*.
- ✓ OFDM membuat jaringan frekuensi tunggal memungkinkan, yang menjadi menarik untuk aplikasi *broadcasting*.
- ✓ OFDM tahan terhadap ISI (*Intersymbol Interference*), dapat meningkatkan modulasi dan coding untuk tiap *subcarrier* dan ekualiser yang sederhana.
- ✓ Modulasi dengan kompleksitas rendah.
- ✓ OFDM mendapatkan efisiensi spectral yang lebih baik dibandingkan teknik *Frequency Division Multiplexing* (FDM) konvensional.
- ✓ Implementasi diital sederhana dengan menggunakan (*Discrete Fourier Transform*) DFT/*Inverse-DFT*.

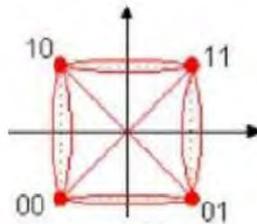
Kelemahan OFDM adalah :

- ✓ OFDM lebih sensitif terhadap frekuensi *offset* dan gangguan fase. Sehingga membutuhkan spesifikasi ketat untuk osilator local.
- ✓ OFDM memiliki (*Peak to Average Power Ratio*) PAPR yang tinggi, memberi karakteristik gelombang tertransmisi yang cenderung mengurangi efisiensi daya dari amplifier RF.
- ✓ OFDM membutuhkan *power amplifier* yang bersifat linear karena PAPR yang tinggi memberi karakteristik sinyal yang ditransmisikan.
- ✓ Terdapat rugi – rugi efisiensi transmisi karena penggunaan *cyclic prefix*.

1.2 *Quadrature Phase Shift Keying (QPSK)*

1.2.1 QPSK Secara Umum

Secara umum modulasi PSK disebut sebagai modulasi MPSK (*M-ary PSK*), dimana M merupakan bilangan yang menunjukkan jumlah level digital. MPSK adalah jenis variasi modulasi fasa yang merupakan peningkatan dari modulasi BPSK pada sisi jumlah perubahan fasa sinyal pembawa. Bila $M = 4$ level, maka modulasi pulsa yang diterapkan adalah QPSK (*quaternary* atau *quadrature PSK*). Bila $M = 2$, maka modulasi pulsa yang dihasilkan adalah BPSK (*binary PSK*). Nilai M mengikuti bilangan 2^v dengan $v = 1, 2, 3$, dst. Dengan nilai v yang demikian itu, maka kemungkinan nilai M adalah 2, 4, 8, 16, dst. Pada sistem modulasi QPSK terdapat empat fasa berbeda yang dihasilkan untuk masing-masing bit. Kombinasi bit *00* akan memodulasi sinyal pembawa yang mempunyai fasa 45 derajat, sedang dibit *01* memodulasi sinyal pembawa yang mempunyai fasa 135 derajat. Sementara bit *10* dan *11*, masing-masing dengan fasa 315 derajat dan 225 derajat. Konstelasi atau penggambaran penyebaran fasa isyarat QPSK ditunjukkan pada Gambar 2.4.

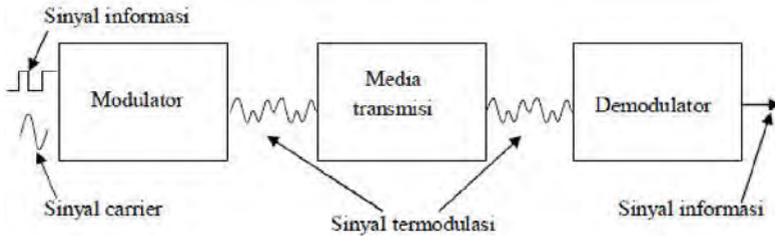


Gambar 2. 4 Diagram Konstelasi QPSK [1]

Setiap harga fasa yang mungkin berkorespondensi dengan pasangan bit yang unik disebut dibit. Sebagai contoh, kita dapat memilih set harga fasa untuk merepresentasikan *set gray coded* dibit : 00, 01, 11, 10.

Gambar 2.5 menunjukkan Sistem komunikasi PSK/QPSK. Modulator memodulasi sinyal pembawa dengan informasi input dan memproduksi sinyal termodulasi PSK atau QPSK. Sinyal termodulasi

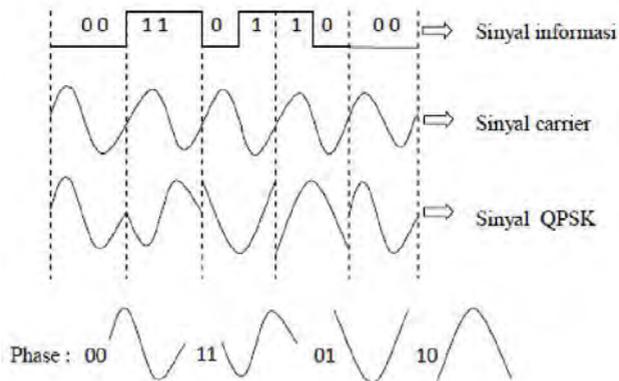
ditransmisikan melewati medium seperti udara, kabel, seratoptik, ke input demodulator. Demodulator menerima sinyal transmisi kemudian merekonstruksi data informasi aslinya.



Gambar 2. 5 Blok Diagram Sistem QPSK [1]

Keterangan gambar :

Pada modulator terjadi proses pencampuran antara sinyal informasi dan sinyal pembawa sehingga menjadi sinyal termodulasi. Dan sinyal inilah yang ditransmisikan sebagai input pada demodulator. Proses yang terjadi pada demodulator adalah proses pengembalian kembali menjadi sinyal awal atau sinyal informasi. Hal ini dilakukan oleh demodulator dengan memisahkan sinyal informasi dengan sinyal pembawa. Bentuk sinyal modulasi QPSK ditunjukkan pada Gambar 2.6.



Gambar 2. 6 Sinyal Modulasi QPSK [1]

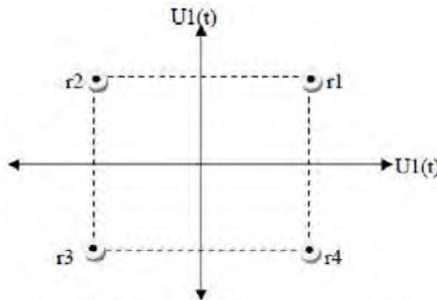
1.2.2 Probability Error

Secara umum dalam sistem transmisi tidak dapat dihindarkan adanya derau. Dengan adanya hal ini akan mengganggu transmisi sinyal sehingga dapat mengakibatkan kesalahan penerimaan. Hal yang demikian itu, sudah pasti terjadi juga pada sinyal QPSK. Untuk mengetahui pengaruh derau terhadap sinyal QPSK, maka berikut ini akan dibahas mengenai peluang kesalahan (error probability) deteksi sinyal QPSK akibat adanya derau. Diagram ruang (space diagram) sinyal QPSK seperti ditunjukkan pada Gambar 2.7 vektor satuan untuk sistem koordinat ini adalah [1] :

$$\mu_1(t) = \sqrt{\frac{2}{T_s}} \cos \omega_c t \quad (2.2)$$

$$\mu_2(t) = \sqrt{\frac{2}{T_s}} \sin \omega_c t \quad (2.3)$$

dengan $T_s = 2 T_b$ (T_b : durasi bit)



Gambar 2. 7 Diagram Ruang Sinyal QPSK [1]

Pada sistem komunikasi, biasanya derau diasumsikan sebagai derau putih aditif dengan distribusi gaussian (*Additive White Gaussian Noise*). Di sini, derau putih aditif gaussian yang menyertainya dapat dinyatakan sebagai berikut [6] :

$$n(t) = n_1 \mu_1(t) + n_2 \mu_2(t) \quad (2.4)$$

$$P_e = 1 - P_c \quad (2.5)$$

Dari Gambar 2.7, terlihat bahwa jika ditransmisikan S_1 , maka deteksi akan benar selama derau tidak memindahkan r_1 keluar dari kuadran pertama. Pada keadaan demikian itu disyaratkan bahwa besar n_1 dan n_2 harus berada pada selang antara $-\sqrt{PSTb}$ sampai tak terhingga. Karena jarak antara sinyal yang ditransmisikan $d = 2\sqrt{PSTb}$, maka peluang deteksi benar P_c jika yang ditransmisikan S_1 dapat dihitung sebagai berikut [1] :

$$P_c = \left(1 - \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{d^2}{4\eta}}\right)\right)^2 \quad (2.6)$$

Dengan demikian peluang kesalahan P_e jika yang ditransmisikan S_1 adalah [6] :

$$P_e = 1 - \left(1 - \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{d^2}{4\eta}}\right)\right)^2 \quad (2.7)$$

Karena setiap sinyal yang ditransmisikan baik itu S_1 , S_2 , S_3 maupun S_4 serupa dan sama (equally likelly) maka peluang kesalahan pun akan sama.

Selain itu juga untuk menyatakan kinerja ini sering digunakan bit error rate (BER). Pada dasarnya pemakaian peluang kesalahan atau BER dapat dipertukarkan, meskipun sebenarnya ada sedikit perbedaan arti antara keduanya. Peluang kesalahan lebih cenderung merupakan ukuran kemungkinan secara teoritis (matematis) mengenai harapan terjadinya kesalahan untuk satu sistem, sedangkan BER pada dasarnya merupakan ukuran secara empiris (historis) mengenai record kesalahan sistem yang aktual. Sebagai contoh, jika sistem mempunyai peluang kesalahan 10^{-6} , ini berarti secara matematis (teoritis) diharapkan ini terjadi kesalahan satu bit dari setiap 10^6 bit yang ditransmisikan, jika sistem mempunyai BER 10^{-6} , ini berarti secara empiris (historis) pada waktu yang telah berlalu secara nyata sistem ini mempunyai kesalahan satu bit untuk setiap 10^6 bit yang ditransmisikan. Pada dasarnya peluang kesalahan merupakan fungsi dari *carrier to noise ratio* (C/N) pada penerima.

1.3 *Peak to Average Power Ratio (PAPR)*

Dalam sistem OFDM, sinyal OFDM terdiri atas penjumlahan sejumlah subcarrier yang dimodulasi secara independen. Apabila fasa setiap *subcarrier* sama maka akan dihasilkan daya sinyal maksimum sebesar N kali daya rata-ratanya. Perbandingan antara daya sinyal maksimum dengan daya rata-ratanya biasa disebut sebagai *Peak to Average Power Ratio (PAPR)*. Pada suatu waktu, penjumlahan ini sangat besar dan pada waktu yang lain sangat kecil, artinya nilai daya maksimum dari sinyal pasti selalu lebih besar dari daya rata-ratanya. Nilai PAPR yang besar merupakan tantangan dalam mengimplementasikan OFDM karena dapat mengurangi efisiensi dan meningkatkan cost dari RF power amplifier. Nilai PAPR yang besar pada OFDM membutuhkan amplifier dengan dynamic range yang lebar untuk mengakomodasi amplitudo sinyal. Jika hal ini tidak terpenuhi maka akan terjadi distorsi linear yang menyebabkan subcarrier menjadi tidak lagi orthogonal dan pada akhirnya menurunkan kinerja OFDM.

Sinyal OFDM terdiri dari sejumlah *subcarrier* yang dimodulasi sendiri-sendiri sehingga dapat menghasilkan perbandingan antara daya puncak dan daya rata-rata (*Peak to Average Power Ratio*) yang cukup besar ketika dijumlahkan (superposisi). Ketika N sinyal dengan fasa yang sama dijumlahkan akan menghasilkan daya puncak sebesar N kali dari daya rata-rata. Daya puncak didefinisikan sebagai daya dari gelombang sinus dengan amplitudo sama dengan nilai *envelope* maksimum. Secara matematis, PAPR suatu sinyal S(t) dideskripsikan sebagai berikut [2] :

$$\text{PAPR} = \frac{P_{max}}{P_{av}} = \frac{\max_{t_0 \leq t \leq T_s} |s(t)|^2}{\frac{1}{T_s} \int_0^{T_s} |s(t)|^2 dt} \quad (2.8)$$

Untuk mengatasi PAPR yang tinggi ini adalah dengan meminimalkan nilai PAPR pada sinyal OFDM yang akan ditransmisikan. Dengan kata lain, diperlukan skema reduksi PAPR pada bagian transmitter sinyal OFDM. Ada beberapa teknik untuk menurunkan nilai PAPR yaitu :

1. Metode distorsi atau pencacatan sinyal

Yang termasuk dalam metode ini adalah : *clipping*, *active constellation extension*, *peak windowing*, *peak cancellation*, *companding*, *envelope scaling*.

2. Metode probabilistik atau pengacakan sinyal (*signal scrambling*)

Yang termasuk dalam metode ini adalah : *multiple signal representation* (MSR), *selective mapping* (SLM), *interleaving*, *partial transmit sequence* (PTS).

3. Metode koding

Yang termasuk dalam metode ini adalah : skema kode blok, skema kode sub-blok, kode blok dengan koreksi error.

1.4 *Clipping*

Dari beberapa metode reduksi PAPR yang telah disebutkan sebelumnya, metode yang digunakan pada sistem ini adalah metode *clipping*. Metode ini digunakan karena *clipping* merupakan cara paling sederhana untuk mengurangi PAPR, yaitu dengan memotong sinyal amplitude puncak menjadi terbatas ke level maksimum yang dibutuhkan. Meskipun menjadi solusi tersederhana, *clipping* menyebabkan distorsi *in-band*, yang menyebabkan penurunan *Bit Error Rate* (BER), kedua distorsi non-linear pada sinyal OFDM secara signifikan meningkatkan level emisi *Out-Of-Band* (OOB).

Ada beberapa teknik *clipping*, yaitu *classical clipping*, *deep clipping*, dan *smooth clipping*.

1.4.1 *Classical Clipping*

Classical Clipping adalah salah satu teknik *clipping* yang populer untuk mereduksi PAPR dan sering disebut juga *hard clipping* atau *soft clipping*. Penggunaan *classical clipping* berpengaruh pada performansi sistem OFDM, termasuk *Power Spektral Density* (PSD), *Peak to Average Power Ratio* (PAPR), dan perhitungan *Bit Error Rate* (BER). Fungsi dasar *classical clipping* ditunjukkan pada persamaan 2.8 [2].

$$f(r) = \begin{cases} r, & \text{bila } r \leq A \\ A, & \text{bila } r > A \end{cases} \quad (2.8)$$

$f(r)$ merupakan simbol OFDM, A ialah level *clipping*. Dari persamaan 2.8 dihasilkan sinyal yang amplitudonya tidak melebihi A . Nilai A ditentukan oleh parameter *Clipping Ratio* (CR) yang dinyatakan persamaan berikut [2]:

$$CR = \frac{A}{\sigma} \quad (2.9)$$

$$\sigma = \sqrt{N} \quad (2.10)$$

Dengan CR adalah Clipping Ratio, σ ialah nilai pembanding, dan N merupakan jumlah subcarrier OFDM.

1.4.2 Deep Clipping

Telah diketahui bahwa *deep clipping* adalah jenis *clipping* yang bertujuan untuk memecahkan masalah *peaks regrowth* yang disebabkan oleh penyaringan (*filtering*) OOB. Pada teknik *deep clipping*, fungsi *clipping* telah dimodifikasi untuk pemotongan lebih “dalam” pada puncak amplitude yang tinggi. Parameter ke dalam *clipping* tersebut. Fungsi dasar *classical clipping* ditunjukkan pada persamaan 2.11 [2].

$$f(r) = \begin{cases} r, & \text{bila } r \leq A \\ A - p(r - A), & \text{bila } A < r < \frac{1+p}{A}A \\ 0, & \text{bila } r > \frac{1+p}{A}A \end{cases} \quad (2.11)$$

Dimana $f(r)$ adalah sinyal OFDM, A adalah level *clipping*, dan p adalah (*depth factor*) *clipping*.

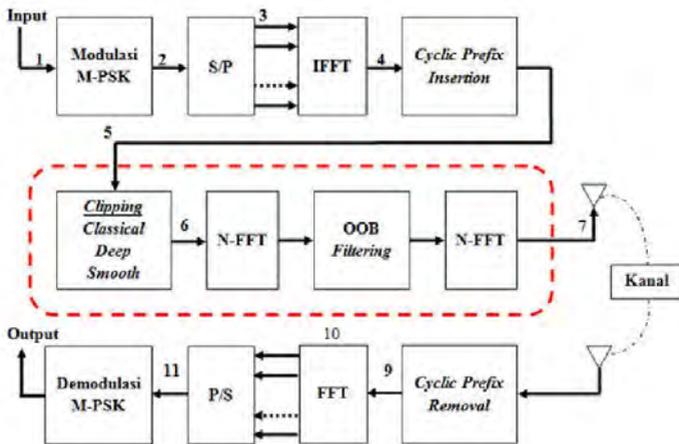
1.4.3 Smooth Clipping

Teknik pemotongan sinyal pada *smooth clipping* ini lebih halus lagi, yaitu dengan persamaan 2.12 [2].

$$f(r) = \begin{cases} r - \frac{1}{b}r^3, & \text{bila } r \leq \frac{3}{2}A \\ A, & \text{bila } r > \frac{3}{2}A \end{cases} \quad (2.12)$$

Dimana $b = \frac{27}{4}A^3$

Diagram blok transmitter OFDM dengan disertai *clipping* secara umum seperti diperlihatkan Gambar 2.8.



Gambar 2. 8 OFDM dengan *Clipping*

Namun dalam tugas akhir ini, tipe *clipping* yang digunakan adalah *classical clipping*, dimana yang diamati adalah pengaruh *classical clipping* pada performansi sistem OFDM, termasuk *Peak to Average Power Ratio* (PAPR) dan perhitungan *Bit Error Rate* (BER).

1.5 **High Power Amplifier (HPA)**

Salah satu komponen yang dapat menyebabkan terjadinya distorsi nonlinier pada sistem adalah *power amplifier* (PA). *Power amplifier* sendiri diwakili oleh fungsi amplitudo (AM/AM) dan fungsi fase (AM/PM). *Power amplifier* akan menyebabkan distorsi nonlinier

sehingga muncul interferensi antar *carrier* yang juga mengakibatkan interferensi antar simbol dan pelebaran spektrum, namun hal ini dapat dikurangi dengan cara menaikkan IBO (*Input Back-off*) yang merupakan perbandingan antara daya masukan saturasi dengan daya input rata-rata.

$$\text{IBO} = 10 \log \frac{P_{\text{input saturasi}}}{P_{\text{input average}}} \quad (2.13)$$

Sedangkan sinyal keluaran dari *power amplifier* akan memiliki persamaan berikut [3]:

$$U(t) = F_A[p(t)]e^{jF_p[p(t)]} \cdot e^{j\varphi(t)} \quad (2.14)$$

FA[.] adalah fungsi AM/AM dan FP[.] adalah fungsi AM/PM [3].

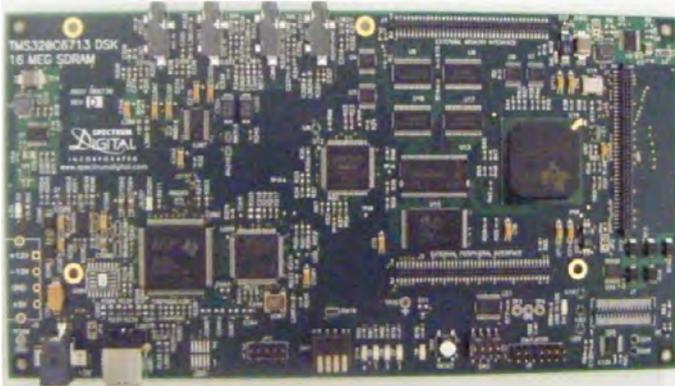
$$F_A[.] = \text{Asaturation, jika } \leq \text{Asaturation}$$

$$F_p[.] = 0, \text{ jika } > \text{Asaturation} \quad (2.15)$$

Dengan F adalah amplitudo linear dan *Asaturation* adalah amplitudo saturasi. *Clipping distortion* digunakan untuk membatasi distorsi yang dialami sinyal sehingga pengaruh nonlinearitas dapat dikurangi. Sinyal *clipping* didapatkan dari hasil pengurangan sinyal nonlinear dengan sinyal sebelum masuk ke PA yang telah dikali *clipping ratio*. Kemudian sinyal yang diterima oleh penerima OFDM merupakan hasil pengurangan sinyal yang telah mengalami distorsi nonlinear dan AWGN dengan sinyal clipping dibagi dengan *clipping ratio*.

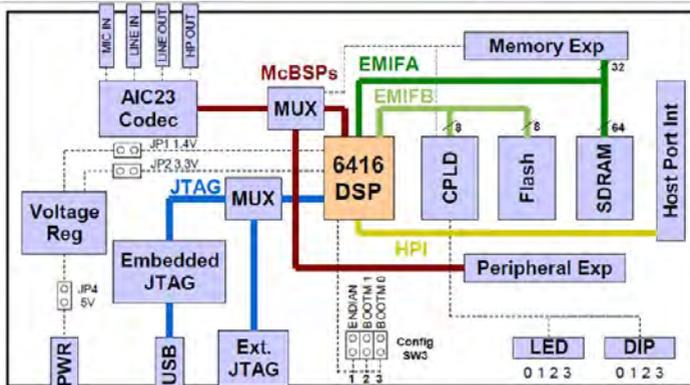
1.6 Digital Signal Processor Starter Kit TMS320C6416T

Digital Signal Processor TMS320C6416 merupakan mikroprosesor berkecepatan tinggi dengan tipe arsitektur yang cocok digunakan untuk mengolah sinyal. Notasi C6x merupakan kode dari produk DSP keluaran Texas Instruments TMS320C6000. Dengan menggunakan arsitektur *very long instruction word* (VLIW), DSP C6x menjadi prosesor tercepat keluaran Texas Instruments. Arsitektur VLIW pada DSP C6x sangat cocok untuk proses perhitungan yang intensif. Secara fisik DSK TMS320C6416T dapat dilihat pada Gambar 2.9.



Gambar 2. 9 DSK TMS320C6416T [4]

Untuk memudahkan dalam pemahaman, sebuah diagram skematik fungsional nya diberikan pada Gambar 2.10.



Gambar 2. 10 Blok Diagram Skematik DSK TMS320C6416T [4]

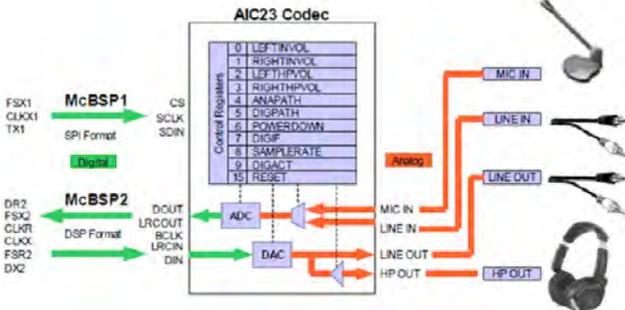
Untuk memudahkan dalam pemahaman, sebuah diagram skematik fungsionalnya diberikan pada Gambar 2.10, disini ada beberapa fitur kunci seperti:

- ✓ Texas Instruments TMS320C6416T DSP beroperasi pada 225 MHz
- ✓ Kode stereo AIC23

- ✓ 8 MB DRAM sinkron
- ✓ Memori Flash 512 KB non-volatile
- ✓ 4 user dapat diakses LED dan DIP switch
- ✓ Opsi boot dapat dikonfigurasi
- ✓ Menggunakan konektor standar ekspansi untuk daughter card.
- ✓ JTAG emulasi melalui on-board JTAG emulator dengan antarmuka USB atau eksternal emulator
- ✓ Single tegangan catu daya (+ 5 v)

DSK menggunakan suatu stereo codec dari Texas Instrument bertipe AIC23 untuk interface sinyal audio input dan output. Codec menyampel sinyal-sinyal analog pada microphone atau line input dan mengkonversikannya ke dalam bentuk data digital sehingga dapat diproses dengan DSP. Ketika DSP telah menyelesaikan pengolahan data (pemfilteran, dsb), maka codec ini juga bisa berfungsi dalam mengkonversi sinyal digital menjadi sinyal output analog melalui line out atau headphone.

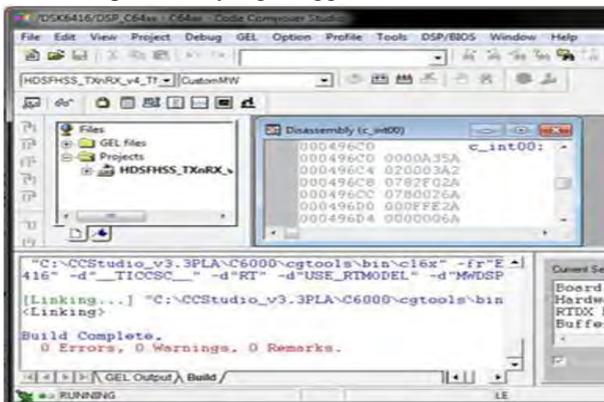
Codec berkomunikasi menggunakan dua channel serial, satu untuk mengontrol register konfigurasi line codec, dan satunya untuk mengirim dan menerima sampel-sampel audio digital. McBSP0 digunakan sebagai unidirectional control channel. Ini seharusnya deprogram untuk mengirim suatu 16-bit control word ke AIC23 dalam format SPI. Ada 7 bit atas pada control word yang seharusnya menspesifikasi register untuk dimodifikasi dan ada 9 bit bawah yang berisi nilai-nilai register. Control channel hanya digunakan ketika mengkonfigurasi codec, pada umumnya dalam kondisi idle ketika data audio sedang ditransmisi. McBSP1 digunakan sebagai bi-directional data channel. Semua audio data mengalir melalui data channel. Beberapa format data telah bisa disupport didasarkan pada tiga variable pada sample width, clock signal source, dan serial data format. Contoh-contoh DSK secara umum menggunakan sebuah lebar sampel 16-bit dengan codec dalam master mode sehingga mampu membangkitkan frame sync dan bil clocks pada sampel rate yang tepat tanpa ada usaha ekstra pada sisi DSP. Pilihan format serial adalah mode DSP yang dirancang secara khusus untuk beroperasi dengan McBSP port pada TI DSP.



Gambar 2. 11 TMS320C6416T DSK Codec Interface [4]

1.7 Code Composer Studio

Code Composer Studio (CCS) merupakan sebuah *Integrated Development Environment (IDE)* untuk Texas Instruments (TI) *embedded processor* yang merupakan gabungan dari *C compiler*, *assembler*, *linker*, dan *debugger*. CCS merupakan software *easy to-use* untuk *build* dan *debug* program pada *DSP processor*, dan mampu melakukan analisis secara real time. IDE menyediakan *single user interface* untuk membawa setiap step dari aliran pengembangan aplikasi. *Tools* dan *interfaces* memungkinkan pengguna untuk memulai lebih cepat dan menambah fungsionalitas ke dalam aplikasi mereka berkat produktivitas dari peralatan yang canggih tersebut.

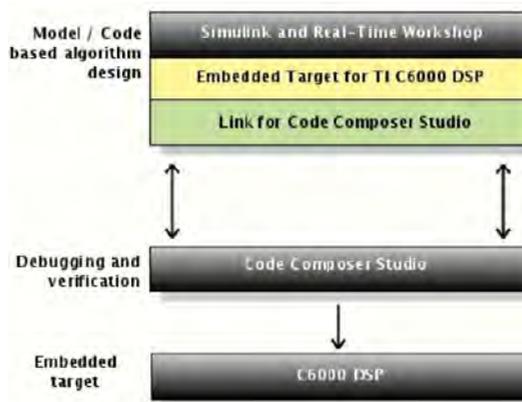


Gambar 2. 12 Tampilan Code Composer Studio v3.3

CCS menyediakan sebuah *integrated development environment* (IDE) yang dapat membangkitkan kode program untuk dijalankan pada DSK. Tool untuk pembangkitan kode yang terdapat di dalam CCS yaitu *C compiler*, *assembler* dan *linker*. CCS memiliki kemampuan grafis dan mendukung *real-time debugging*. *C compiler* mengkompilasi sebuah program dalam bahasa C dengan ekstensi **.c*, untuk menghasilkan file *assembly* menggunakan ekstensi **.asm*. *Assembler* memproses file **.asm* untuk menghasilkan file bahasa mesin dengan ekstensi **.obj*. Kemudian *linker* menggabungkan file–file tersebut menjadi *executable file* dengan ekstensi **.out*. DSK harus dihubungkan ke PC melalui port USB agar bisa diprogram dengan bantuan CCS. Untuk analisis *real-time* dapat menggunakan fasilitas *real-time data exchange* (RTDX) yang memungkinkan pertukaran data antara PC dan DSK tanpa melepas DSK.

CCS juga mendukung kegiatan *debugging* antara lain, setting *breakpoint*, secara otomatis meng-*update* jendela saat *breakpoint*, mengamati nilai variabel, melihat dan mengamati memori dan register, menggunakan *probe point* untuk mengalirkan data dari dan ke target untuk mengumpulkan *snapshot memory*, menggambarkan sinyal yang ada pada target, melakukan *profiling* terhadap statistik eksekusi, memeriksa instruksi C dan instruksi yang di-*disassembly* pada target.

Software matlab telah menyediakan sebuah fungsi untuk berkomunikasi dengan DSK TMS320C6x dengan bantuan CCS. Kemudian CCS mengintegrasikan simulasi yang sudah dibentuk dari Simulink Matlab kemudian mengkonversikan ke dalam bahasa C maupun *assembly*. Semua *software* ini membangun dan memvalidasi hasil perancangan pengolahan sinyal digital mulai dari konsep sampai dengan menghasilkan kode yang akan dijalankan oleh DSK TMS320C6x. Lalu file keluaran dari proses ini kemudian diimplementasikan ke dalam DSK TMS320C6416. Proses *running* dapat diakses dari *CCS debugging tools* atau melihat langsung hasil simulasi *Simulink* Matlab.



Gambar 2. 13 Diagram alir antara Simulink, CCS, dan C6000 [5]

Rekomendasi ini digunakan untuk prediksi propagasi gelombang langit (*skywave*) pada frekuensi antara 2-30 MHz. Rekomendasi ini berlaku untuk jarak lintasan hingga 7000 Km, jarak lintasan lebih dari 9000 Km, dan lintasan yang memiliki jarak antara 7000-9000 Km.

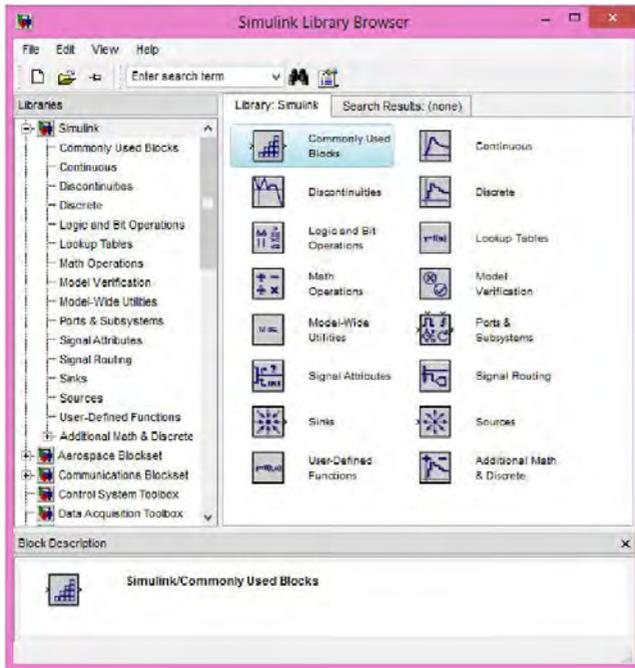
1.8 Simulink

Simulink merupakan *software add-on* yang terdapat di Matlab dengan perangkat matematis yang dikembangkan oleh *The Mathworks*. Simulink yang berupa *extension* grafis dari Matlab digunakan untuk pemodelan maupun simulasi dari sebuah sistem. Dengan menggunakan Simulink, sistem dapat dirancang melalui blok-blok diagram. Banyak blok diagram yang disediakan pada *Simulink Library* seperti *Fuzzy Logic*, *Neural Network*, *DSP*, *Statistik*, dan lain sebagainya baik sebagai perangkat input (seperti *function generator*) serta perangkat output (seperti *oscilloscopes*). Simulink terintegrasi dengan Matlab dan data dapat ditransfer dengan mudah antar kedua program tersebut. Dengan adanya *templates/building blocks* menjadi salah satu keuntungan utama dari Simulink sehingga tidak perlu menulis kode program untuk proses matematisnya. Untuk mulai menjalankan *simulink*, klik *icon simulink* di bagian atas *toolbar* matlab.

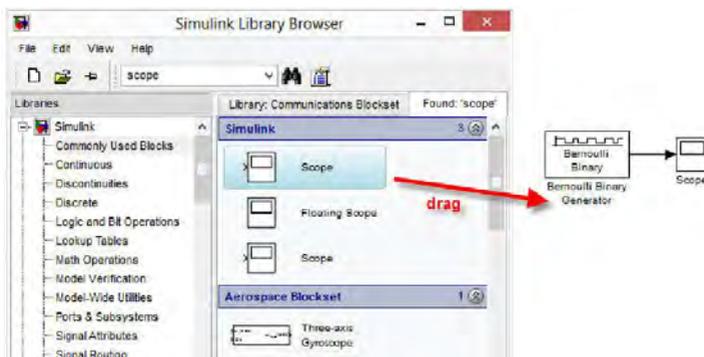


Gambar 2. 14 Menjalankan Matlab Simulink

Selanjutnya muncul *window* “*Simulink Library Browser*” yang berisi berbagai macam blockset untuk membuat pemodelan dasar suatu sistem. Dalam membuat pemodelan sistem baru hanya dengan klik *icon* “*Create A New Model*”, kemudian muncul *window* dengan lembar kosong. Lembar ini yang nantinya digunakan untuk meletakkan objek dari blok-blok sistem yang akan dimodelkan. Caranya, memilih blok yang diinginkan dari *Simulink Library* kemudian di-*drag* ke *window* kosong tersebut. Selanjutnya, hubungkan blok-blok tersebut dengan menarik output dari suatu blok ke input dari blok lain hingga muncul anak panah. Setelah pemodelan sistem sudah dibentuk, dapat dilakukan simulasinya dengan menentukan waktu simulasi pada kotak *toolbar* lalu klik *icon run*. Hasil dari simulasi tersebut dapat dilihat pada sebuah *scope*, *display*, *vector scope*, *workspace* dan sebagainya yang terdapat pada blockset “*sink*”, tergantung dengan output yang dihasilkan blok tersebut, terlihat pada Gambar 2.15 dan 2.16.



Gambar 2. 15 Simulink Library Browser



Gambar 2. 16 Membuat blok sistem pada Simulink

[Halaman ini sengaja dikosongkan]

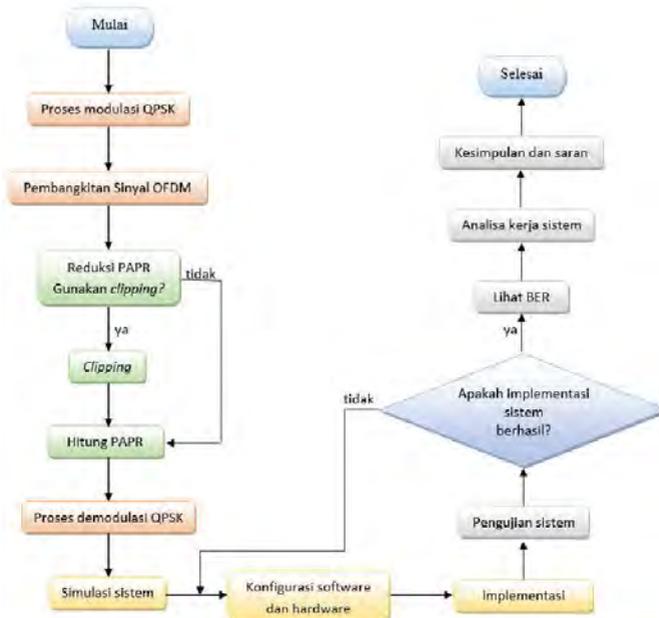
BAB III

PERANCANGAN, SIMULASI DAN IMPLEMENTASI SISTEM

Pada bab ini akan dijelaskan mengenai pemodelan blok sistem “reduksi PAPR sinyal OFDM dengan metode *clipping*” sehingga dapat disimulasikan dan selanjutnya diimplementasikan pada DSK TMS320C6416T.

1.1 Alur Pengerjaan Tugas Akhir

Dalam melakukan perancangan sistem perlu dilakukan beberapa tahapan diantaranya tahap pemodelan sistem, analisa, simulasi serta implementasi sistem.



Gambar 3.1 Flowchart Pengerjaan Tugas Akhir

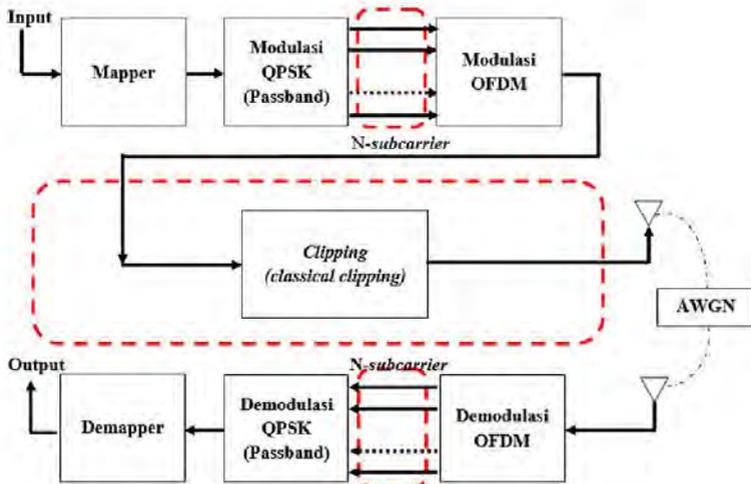
Gambar 3.1 merupakan alur dalam mengerjakan tugas akhir. Pertama adalah pemodelan sistem pada matlab Simulink, yaitu membangkitkan sinyal yang menggunakan modulasi QPSK kemudian

membangkitkan sinyal OFDM dengan N -subcarrier. Setelah itu apabila sinyal OFDM tersebut akan direduksi PAPR-nya menggunakan metode *clipping*, maka sinyal OFDM masuk pada blok metode *clipping* lalu dihitung nilai PAPR-nya. Namun apabila tidak direduksi, maka sinyal OFDM langsung dihitung PAPR-nya. Pada proses pengembalian sinyal dilakukan proses demodulasi sehingga sinyal yang diterima sama dengan sinyal yang dikirim.

Tahap berikutnya adalah simulasi dan implementasi sistem. Oleh karena itu perlu dilakukan konfigurasi software dan hardware agar implementasi sistem berjalan lancar. Apabila implementasi sistem berhasil, maka diuji kehandalan sisten dengan melihat nilai BER yang dihasilkan. Tapi apabila implementasi sistem gagal, maka perlu dicek atau dilakukan konfigurasi ulang antara software dan hardware. Setelah didapat nilai BER, selanjutnya dianalisa kerja sistem tersebut dan ditarik kesimpulan.

1.2 Tahap Pemodelan Sistem

Pemodelan sistem reduksi PAPR sinyal OFDM dengan metode *clipping* menggunakan *matlab simulink*. Alur pemodelan reduksi PAPR sinyal OFDM dengan metode *clipping* dapat dilihat pada Gambar 3.2.



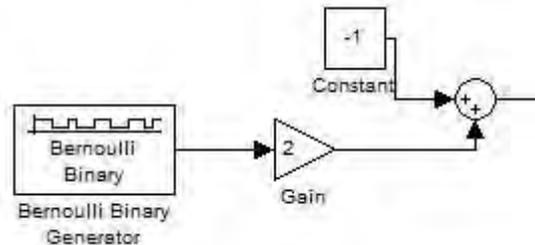
Gambar 3. 2 Blok Diagram PAPR Clipping

1.3 Tahap Simulasi Sistem

Pada tahap ini simulasi sistem dilakukan dengan membuat blok sistem menggunakan Matlab Simulink. Ada 7 tahap pembuatan blok, yaitu Pembangkitan Sinyal & Mapper, Modulasi & OFDM, Metode Reduksi PAPR (*Clipping*), Perhitungan PAPR, Demodulasi, Demapper & Pengembalian Sinyal, dan Perhitungan BER. Penjelasan dari masing – masing blok adalah :

1.3.1 Pembangkitan Sinyal & Mapper

Sinyal input berupa bit data biner [0 1] dibangkitkan menggunakan *Bernoulli Binary Generator*. Gambar 3.3 merupakan blok simulink pembangkitan sinyal dari *Bernoulli Binary Generator*.



Gambar 3. 3 Blok *Bernoulli Binary Generator* dan Mapper

Parameter yang diatur pada *Bernoulli Binary Generator* adalah :

- ✓ *Probability of a zero* : 0.5
(Probabilitas munculnya nilai 0 dan 1 sama)
- ✓ *Initial seed* : 61
(Nilai acak yang dihasilkan pada generator)
- ✓ *Sample time* : 1/100
(Laju bit yang digunakan. Dalam 1 detik ada 100 bit)
- ✓ *Output data type* : *double*

Misal bit sinyal informasi yang dibangkitkan dari Bernoulli sebanyak 100bit, yaitu seperti Tabel 3.1.

Tabel 3. 1 Bit Data dari *Bernoulli Generator* (Sebelum Mapping)

Sinyal Informasi Serial															
1	0	1	1	0	1	1	1	0	0	0	1	1	0	1	1
0	0	0	0	1	1	0	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	0	0	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0	1	0	0	0	1	1	0	1
1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1
1	1	1	0	0	0	0	0	0	1	0	1	1	1	1	0
1	0	1	1	1											

Kemudian bit data tersebut dikonversi (*mapper*) menjadi data $[-1 \ 1]$ dengan cara memberikan “2x” penguatan pada sinyal informasi kemudian ditambah konstanta “-1”. Hasil *mapper* seperti Tabel 3.2.

Tabel 3. 2 Bit Data Setelah Mapping

Sinyal Informasi Serial															
1	-1	1	1	-1	1	1	1	-1	-1	-1	1	1	-1	1	1
-1	-1	-1	-1	1	1	-1	1	-1	1	-1	-1	-1	1	1	1
-1	1	-1	-1	-1	1	-1	-1	-1	-1	1	-1	1	1	-1	-1
1	-1	1	1	-1	-1	1	-1	1	-1	-1	-1	1	1	-1	1
1	1	-1	1	1	-1	1	1	-1	1	1	-1	-1	-1	-1	1
1	1	1	-1	-1	-1	-1	-1	-1	1	-1	1	1	1	1	-1
1	-1	1	1	1											

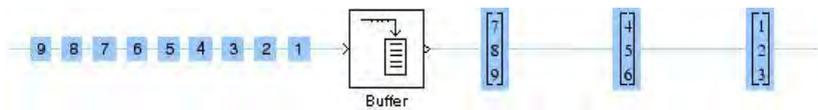
1.3.2 Modulasi dan OFDM

Setelah bit data biner dikonversi ke dalam bentuk $[-1 \ 1]$, maka data tersebut diubah dari serial ke paralel, yaitu dengan menambahkan blok *buffer*. Parameter yang harus diubah pada *buffer*:

- ✓ *Output buffer size (per channel)* : 16
(Jumlah *frame* yang dibuat)

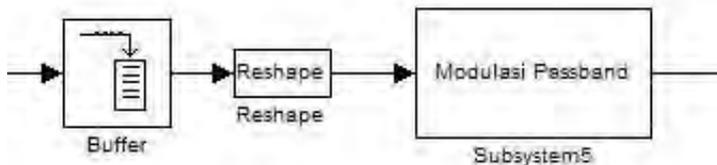
- ✓ *Buffer overlap* : 0
(Jumlah *frame* yang saling overlap)
- ✓ *Initial condition* : 0
(*frame* yang diabaikan)

Gambar 3.4 menunjukkan cara kerja blok *buffer*. Dimana data serial diubah menjadi data paralel dengan jumlah *buffer* / paralel data sebanyak 3. Dalam sistem ini, *buffer* yang diberikan sebanyak 16.



Gambar 3. 4 Cara Kerja Blok *Buffer*

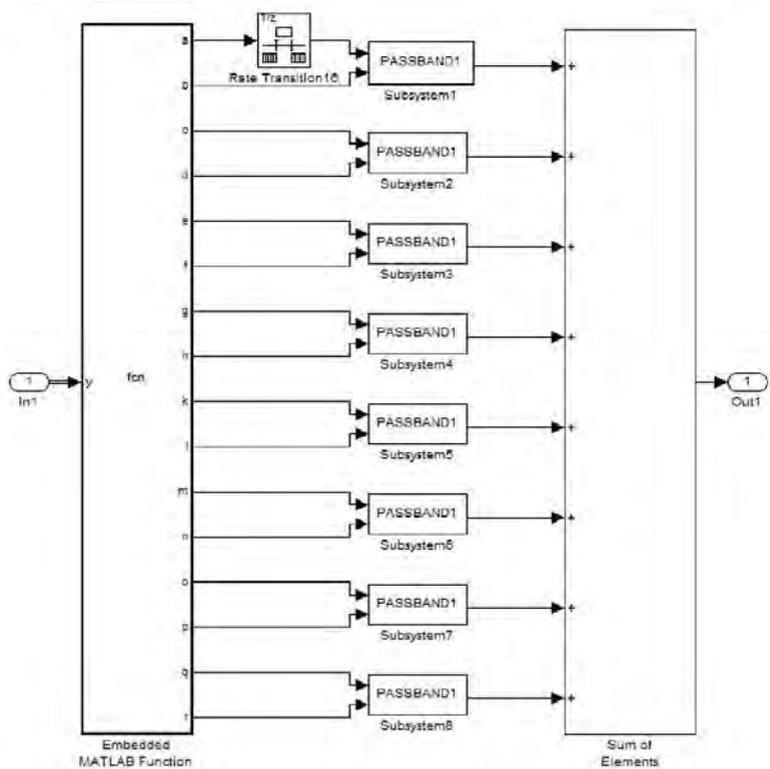
Kemudian ditambah blok *reshape* untuk mengubah dimensi matriks dari 1x1 menjadi 16x1.



Gambar 3. 5 Blok Serial ke Paralel dan *Subsystem* Modulasi

Kemudian data yang sudah dibuffer dilewatkan pada blok *Embedded Matlab Function* sebanyak 16 bit ada. Ini artinya, sistem ini akan membangkitkan sinyal OFDM dimana dalam 1 simbol OFDM terdiri dari 16 bit data. Gambar 3.6 menampilkan blok OFDM.

Dalam blok *Embedded Matlab Function* terdapat *syntax* untuk melewatkan bit hasil *buffer* ke 16 input paralel secara bergantian. Gambar 3.7 adalah *syntax* dari *Embedded Matlab Function*.



Gambar 3. 6 Blok OFDM 8 Subcarrier

Dari *syntax* pada Gambar 3.7, berarti blok tersebut memiliki 1 input (y) dan 16 output (a, b, c, d, e, f, g, h, k, l, m, n, o, p, q, r), dimana input baris “1” dilewatkan output “a”, baris “2” dilewatkan output “b”, dan seterusnya hingga input baris “16” dilewatkan output “r” sedangkan input baris “17” dilewatkan kembali output “a”, dan seterusnya secara berulang. Tabel 3.3 merupakan output data setelah diparalel.

```

1  function [a,b,c,d,e,f,g,h,k,l,m,n,o,p,q,r]= fcn(y)
2
3 - a = y(1,1);
4 - b = y(2,1);
5 - c = y(3,1);
6 - d = y(4,1);
7 - e = y(5,1);
8 - f = y(6,1);
9 - g = y(7,1);
10 - h = y(8,1);
11 - k = y(9,1);
12 - l = y(10,1);
13 - m = y(11,1);
14 - n = y(12,1);
15 - o = y(13,1);
16 - p = y(14,1);
17 - q = y(15,1);
18 - r = y(16,1);

```

Gambar 3. 7 *Syntax Embedded Matlab Function Data Paralel*

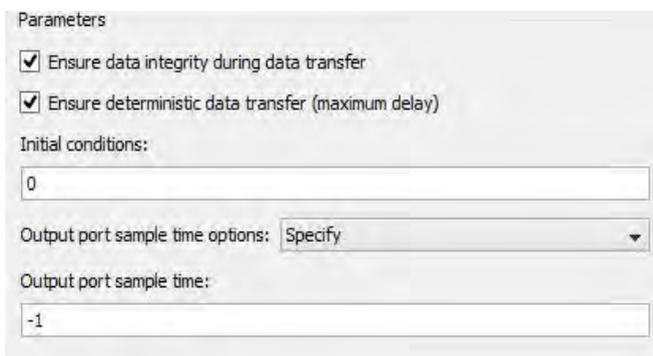
Tabel 3. 3 Bit Data Setelah Proses Paralel

	Sinyal Informasi Paralel						
a	0	1	-1	-1	1	1	1
b	0	-1	-1	1	-1	1	1
c	0	1	-1	-1	1	-1	1
d	0	1	-1	-1	1	1	-1
e	0	-1	1	-1	-1	1	-1
f	0	1	1	1	-1	-1	-1
g	0	1	-1	-1	1	1	-1
h	0	1	1	-1	-1	1	-1
k	0	-1	-1	-1	1	-1	-1
l	0	-1	1	-1	-1	1	1
m	0	-1	-1	1	-1	1	-1
n	0	1	-1	-1	-1	-1	1
o	0	1	-1	1	1	-1	1
p	0	-1	1	1	1	-1	1
q	0	1	1	-1	-1	-1	1
r	0	1	1	-1	1	1	-1

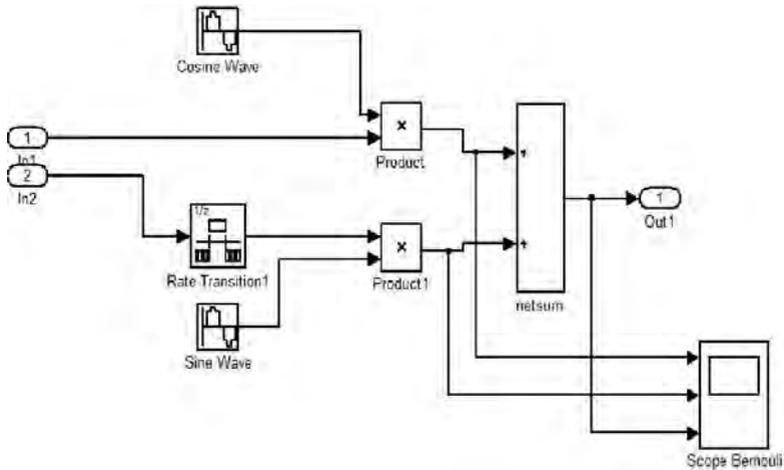
Berdasarkan Tabel 3.3 diketahui bahwa setelah melewati buffer, data ter-*delay* sebanyak 16 bit (1 bit per output paralel / 1 simbol OFDM). *Delay* tersebut dilihat dari jumlah data yang bernilai “0”. Selanjutnya masuk pada blok modulasi *passband*.

Pada blok modulasi terdapat proses penumpangan sinyal informasi ke dalam sinyal *carrier*. Disini sinyal akan dipecah menjadi dua yaitu menjadi komponen *Inphase* dan *Quadrature*. Setelah itu, sinyal asli dikalikan sinyal *carrier* yaitu berupa sinyal cosinus (untuk *inphase*) dan $-\sin$ (untuk *quadrature*) lalu masing-masing sinyal hasil perkalian dengan sinyal *carrier* dijumlahkan. Hasil penjumlahan 2 sinyal tersebut merupakan hasil dari modulasi QPSK. Output 1 modulasi QPSK ini merupakan output dari 1 *subcarrier*.

Sebelum masuk ke perkalian sinyal (blok *product*), ternyata data informasi harus ditambahkan blok *rate transition* untuk menyamakan laju bit dengan sinyal *carrier*-nya. Dan blok *rate transition* ini bisa menambah *delay* data informasi. Hal ini karena dalam blok *rate transition* terdapat parameter “*Ensure deterministic data transfer (maximum delay)*” yang harus dicentang. Apabila parameter tersebut tidak dicentang, maka tidak akan menambah *delay*. Parameter pada blok *rate transition* akan ditunjukkan pada Gambar 3.8.



Gambar 3. 8 Parameter pada Blok *Rate Transition*

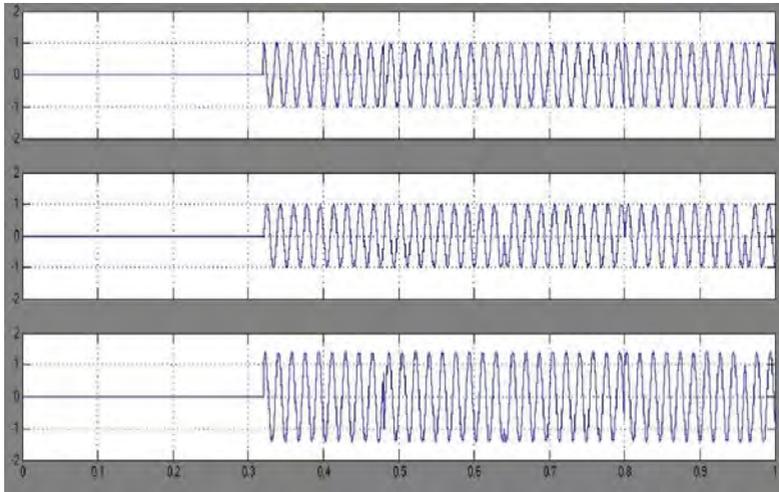


Gambar 3. 9 Blok Modulasi QPSK

Parameter yang diatur pada *sine/cosine wave* adalah :

- ✓ *Amplitude* : 1
(Amplitudo sinyal)
- ✓ *Bias* : 0
(Nilai konstanta yang ditambahkan dalam sinyal)
- ✓ *Frequency (rad/sec)* : $2 \cdot \pi \cdot 900 / 16$
(frekuensi *carrier* yang digunakan)
- ✓ *Phase (rad)* : $\pi / 2$
(Perbedaan fase yang diberikan. Untuk sinyal *cosinus*, fase yang diberikan adalah “ $\pi/2$ ” sedangkan untuk sinyal *sinus*, fase yang diberikan adalah “0”)
- ✓ *Sample time* : 1/1600
(Dalam 1 detik terdapat 1600 sampel)

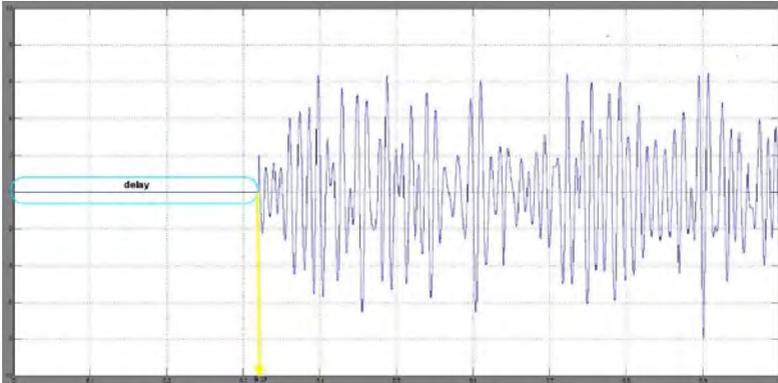
Output dari sinyal hasil modulasi QPSK ditampilkan pada *scope* seperti Gambar 3.10.



Gambar 3. 10 Output Sinyal Modulasi QPSK

Sinyal pertama adalah hasil perkalian dari data informasi dengan *carrier inphase*, sinyal kedua adalah hasil perkalian dari data informasi dengan *carrier quadrature*, dan sinyal ketiga adalah hasil penjumlahan dari sinyal pertama dan kedua.

Sinyal OFDM merupakan penjumlahan dari berbagai sinyal QPSK yang memiliki nilai frekuensi *carrier* yang berbeda - beda. Sistem ini menggunakan 8 *subcarrier* dengan 8 frekuensi *carrier* yang berbeda, yaitu mulai “ $2 \cdot \pi \cdot 900/16$ rad” hingga “ $2 \cdot \pi \cdot 1600/16$ rad”. Selisih antara frekuensi *subcarrier* 1 dengan frekuensi *subcarrier* 2 adalah “ $1/T_{ofdm}$ ”, dimana $T_{ofdm} = 16/100$. Artinya, $T_s = 1/100$ sedangkan dalam 1 simbol OFDM terdapat 16 bit. Dan pada *subcarrier* pertama adalah $f_{c1} = 9/T_{ofdm}$ ($f_{c1} = 9/(16/100) = 900/16$). Karena selisih antara tiap *subcarrier* adalah “ $1/T_{ofdm}$ ”, maka $f_{c2} = 9/(16/100) + 1/(16/100) = 10/(16/100) = 1000/16$. Begitu juga dengan frekuensi *subcarrier* berikutnya hingga diperoleh frekuensi *subcarrier* terakhir, $f_{c8} = 1600/16$. Gambar 3.7 adalah hasil sinyal OFDM yang dilihat dari *scope*.

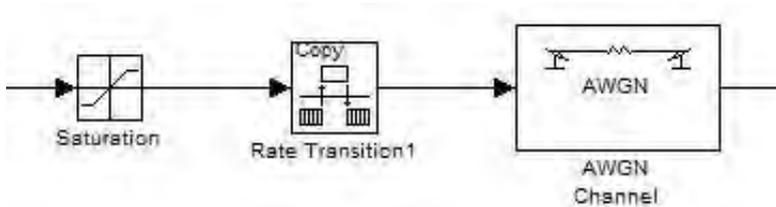


Gambar 3. 11 Sinyal OFDM dari *Scope*

Dalam sinyal tersebut terdapat *delay* sebesar 0.32 detik. Artinya, karena sampel time sinyal yang dibangkitkan 1/100, maka sinyal tersebut terkena *delay* sebesar $0.32 \text{ detik} * 100 \text{ bit/detik} = 32 \text{ bit}$ atau 2 simbol OFDM. *Delay* tersebut diperoleh dari *delay buffer* dan *delay rate transition* yang telah dibahas sebelumnya.

1.3.3 Metode Reduksi PAPR (*Clipping*)

Metode yang digunakan untuk mereduksi PAPR pada sistem ini adalah metode *clipping*, yaitu pencacatan sinyal dengan cara memotong puncak sinyal. Level *clipping* ditentukan sendiri. Blok simulink yang digunakan untuk memotong sinyal adalah blok *saturation*. Gambar 3.12 merupakan blok reduksi PAPR.



Gambar 3. 12 Blok Reduksi PAPR (*Clipping*) dan AWGN

Parameter yang diatur pada *saturation* adalah :

- ✓ *Upper limit* : 7
(Amplitudo maksimum sinyal)
- ✓ *Lower limit* : -7
(Amplitudo minimum sinyal)

Artinya, amplitudo maksimum sinyal yang masuk kanal AWGN sebesar 7 volt sedangkan amplitudo minimumnya sebesar -7 volt. Sinyal yang memiliki amplitudo di atas 7 volt dan di bawah -7 volt akan terpotong secara otomatis. Reduksi PAPR dengan metode memotong sinyal ini akan menambah nilai *bit error rate* (BER) yang diperoleh.

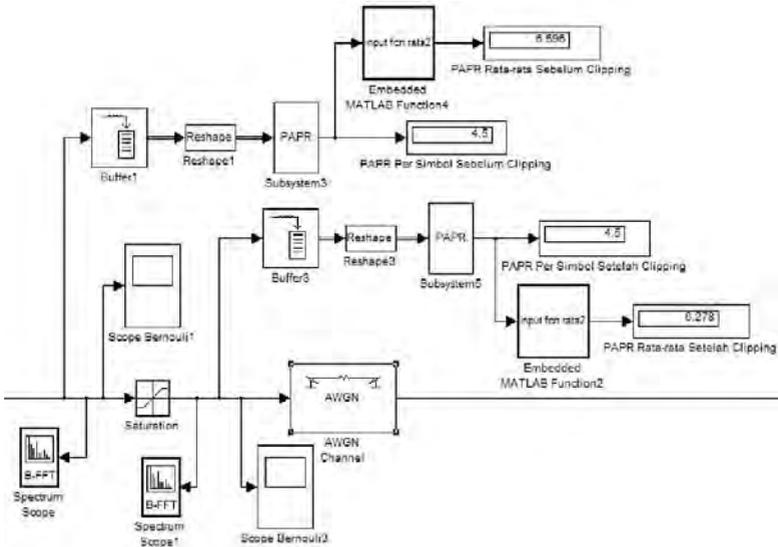
Setelah itu, sinyal OFDM tersebut ditransmisikan melewati kanal AWGN. Blok "*AWGN Channel*" adalah sebuah blok yang berfungsi untuk mensimulasikan kanal agar sesuai dengan kanal sebenarnya dimana terdapat noise yang berupa AWGN (*additive white gaussian noise*). Parameter yang perlu diperhatikan dalam pengaturan blok ini adalah penentuan *signal to noise ratio* (S/N) dengan E_b/N_0 yang akan menjadi variable saat pengujian.

Parameter yang diatur pada *AWGN Channel* adalah :

- ✓ *Initial seed* : 45
(nilai acak yang dihasilkan generator)
- ✓ *Mode* : *Signal to noise ratio* (E_b/N_0)
(Ada beberapa mode, namun yang digunakan adalah E_b/N_0)
- ✓ *E_b/N_0 (dB)* : -10 sampai 0
(Rasio antara energy bit dengan *power spectral density noise*)
- ✓ *Number of bits per symbol* : 2
(QPSK jadi menggunakan 2 bit/symbol)
- ✓ *Input signal power (watts)* : 1
(Power yang sebenarnya dari simbol pada masukan blok)
- ✓ *Symbol period (s)* : 1/100
(periode simbol pada kanal AWGN)

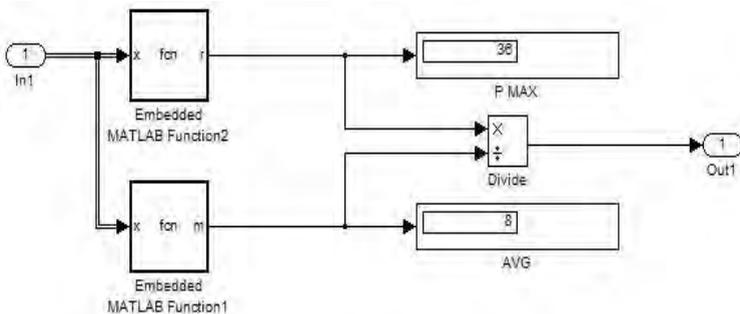
1.3.4 Perhitungan PAPR

Peak to Average Power Ratio (PAPR) dihitung per simbol OFDM, dimana dalam satu simbol OFDM terdiri dari 16 bit data. Karena itu sinyal informasi yang dikirim harus *buffer* terlebih dahulu tiap 16 bit lalu *direshape* untuk menyamakan matriksnya. Kemudian dihitung PAPR menggunakan blok *Embedded Matlab Function*. Dalam blok *Embedded Matlab Function* bisa memasukkan rumus dari PAPR sesuai persamaan 2.8. PAPR dihitung saat sinyal sebelum dan sesudah *clipping*. Kemudian kedua nilai PAPR tersebut dibandingkan dan dianalisa.



Gambar 3. 13 Blok PAPR Sebelum dan Sesudah *Clipping*

Perhitungan PAPR per simbol OFDM diperoleh dari kuadrat daya maksimal dibagi kuadrat daya rata-rata. Gambar 3.14 merupakan blok untuk menghitung daya maksimal, daya rata-rata, PAPR sedangkan Gambar 3.15 adalah syntax kuadrat daya maksimal dan daya rata-rata pada *Embedded Matlab Function*.



Gambar 3. 14 *Subsystem* Blok Perhitungan PAPR per Simbol OFDM

```

1 function r = fcn(x)
2
3 - r = max(power(x,2));
4
1 function m = fcn(x)
2
3 - m = mean(power(x,2));
4

```

Gambar 3. 15 *Syntax* Daya Maksimal dan Daya Rata-rata

Berdasarkan *syntax* pada Gambar 3.15 maka akan diperoleh nilai kuadrat daya maksimal sebesar 36 watt dan kuadrat daya rata-rata sebesar 8 watt, seperti yang terlihat pada display Gambar 3.14. $\text{PAPR 1 simbol OFDM} = 36/8 = 4.5$.

Nilai PAPR per simbol OFDM dapat dilihat pada display Gambar 3.13. Setelah diperoleh nilai PAPR per simbol OFDM, maka dihitung nilai PAPR rata-rata dari semua sinyal OFDM yang dibangkitkan. Perhitungan PAPR rata-rata juga menggunakan *syntax* pada *Embedded Matlab Function*, seperti pada Gambar 3.16.

```

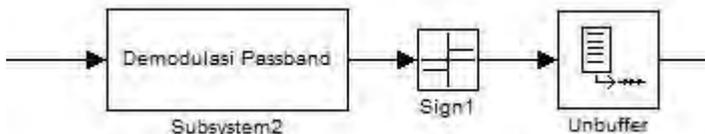
1  function rata2 = fcn(input)
2  - yy=0;
3  persistent x;
4  persistent y;
5
6  - if isempty(x)
7  -     x = 0; % initialize the persistent variable
8  - end
9  - if isnan(input)
10 -     input=0;
11 -     x=0;
12 - end
13 - if isempty(y)
14 -     y = input;% initialize the persistent variable
15 - end
16 - if input~=0
17 -     y =y+input;
18 -     x=x+1;
19 - end
20
21 - yy=y;
22 - rata2=yy/x;
23

```

Gambar 3. 16 *Syntax* PAPR Rata-rata

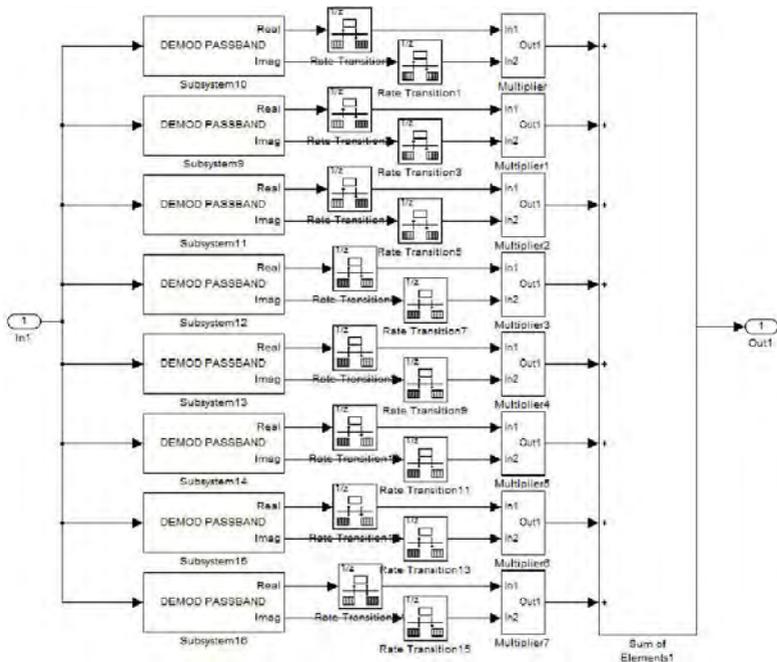
1.3.5 Demodulasi

Demodulasi merupakan step untuk mengembalikan data sinyal informasi. Di dalam *subsystem* “Demodulasi Passband” terdapat blok demodulasi QPSK dan pemecahan kembali sinyal OFDM berdasarkan masing-masing frekuensi *carrier* yang telah diberikan pada proses modulasi QPSK sinyal OFDM.



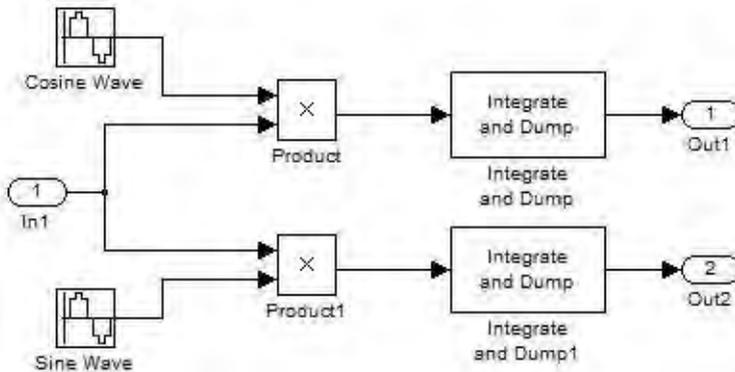
Gambar 3. 17 Blok Demodulasi dan Sign

Blok *subsystem* “Demodulasi *Passband*” bisa dilihat lebih rinci pada Gambar 3.18.



Gambar 3. 18 Subsystem Blok Demodulasi Passband

Untuk mengembalikan sinyal OFDM, maka sinyal informasi yang dikirim diparalel kembali menjadi 8, sesuai jumlah *subcarrier*. Lalu setiap *subcarrier* dikalikan kembali dengan sinyal *carrier*-nya, yaitu *cosinus* (untuk *inphase*) dan $-sinus$ (untuk *quadrature*) sesuai *frekuensi carrier* yang telah diberikan pada bagian modulasi. Misal jika pada proses modulasi, *frekuensi carrier* yang digunakan adalah $2\pi \cdot 900/16 \text{ rad}$, maka pada proses demodulasi ini juga menggunakan *frekuensi* $2\pi \cdot 900/16 \text{ rad}$. Pengaturan lainnya pada blok *cosine/sine wave* pada bagian demodulasi sama persis dengan bagian modulasi. Kemudian hasil perkalian sinyal informasi dan sinyal *cosinus/sinus* dihitung nilai integralnya menggunakan blok *Integrate and Dump*. Hal ini bertujuan untuk memisahkan sinyal informasi dengan *carrier*-nya. Blok demodulasi QPSK bisa dilihat pada Gambar 3.19.



Gambar 3. 19 Blok Demodulasi QPSK

Parameter yang diatur pada *Integrate and Dump* adalah :

- ✓ *Integration Period (number of samples)* : 256
(jumlah sampel dalam satu simbol sinyal OFDM)
- ✓ *Offset (number of samples)* : 512
(jumlah sampel simbol OFDM yang diabaikan perhitungannya akibat delay)

Perhitungan *integration period* berdasarkan jumlah sampel yang digunakan dalam 1 simbol OFDM. Sample time yang diberikan adalah 1/100, sehingga sample time pada 1 simbol OFDM adalah 16/100. Sementara jumlah titik sampel yang diberikan adalah 1/1600, artinya dalam 1 detik ada 1600 titik sampel. Lalu untuk menghitung jumlah sampel dalam 1 simbol OFDM adalah $16/100 * 1600 = 256$ sampel. Sedangkan untuk perhitungan *offset* berdasarkan berapa banyak simbol OFDM yang terkena *delay* sebelum sinyal ditransmisikan. Seperti yang telah dijelaskan sebelumnya, *delay* yang dihasilkan sebanyak 2 simbol OFDM sehingga jumlah sampel *offset*-nya adalah $2 * 256 = 512$ sampel.

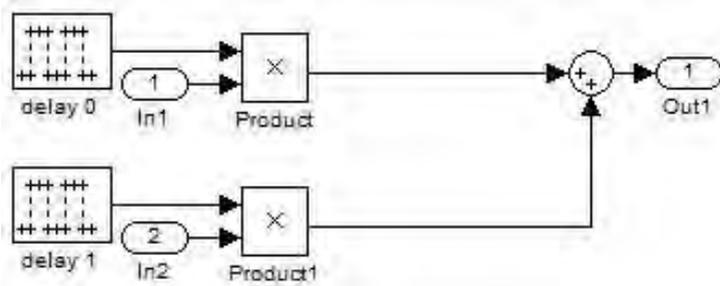
Tabel 3. 4 Output *Integrate and Dump*

	Sinyal Informasi Paralel						
aa	0	0	0	120.3729	-139.001	-104.991	99.83404
bb	0	0	0	-151.318	-167.458	112.9872	-149.187
cc	0	0	0	171.749	-85.4759	-131.839	162.2517
dd	0	0	0	125.8216	-85.4486	-63.2216	130.4868
ee	0	0	0	-169.329	136.6795	-108.83	-155.567
ff	0	0	0	117.4263	140.3523	122.2463	-159.175
gg	0	0	0	145.5994	-140.272	-28.8318	127.6695
hh	0	0	0	142.9364	149.8566	-124.519	-155.272
kk	0	0	0	-137.312	-102.825	-94.8273	146.0625
ll	0	0	0	-126.048	124.1866	-121.7	-150.751
mm	0	0	0	-120.299	-63.8923	177.8564	-129.882
nn	0	0	0	87.73154	-167.851	-118.616	-122.407
oo	0	0	0	98.29577	-125.565	75.80703	97.5228
pp	0	0	0	-142.029	68.95244	156.5472	114.5199
qq	0	0	0	119.7968	104.4133	-126.979	-147.185
rr	0	0	0	118.5767	105.3134	-127.152	149.9322

Berdasarkan data pada Tabel 3.4, blok *integrate and dump* menimbulkan *delay* 1 simbol OFDM sehingga total *delay* yang dihasilkan menjadi 3 simbol OFDM. Pola data yang dihasilkan pada Tabel 3.4 ini sama dengan Tabel 3.3, dengan asumsi bahwa pada output *integrate and dump* yang bernilai di atas “0” diubah menjadi “-1” dan di bawah “0” menjadi “1”. Misal, nilai [a = 0 1 -1 -1 1 1 1] sedangkan nilai [aa = 0 0 0 120.3729 -139.001 -104.991 99.83404] atau [aa = 0 0 0 1 -1 -1 1]. Begitu juga dengan data lainnya.

Setelah itu, output *integrate and dump* yang masih paralel diubah kembali menjadi data serial menggunakan blok *multiplier* yang dibangkitkan menggunakan *pulse generator*. Blok *multiplier* berisi blok untuk mengalikan sinyal hasil demodulasi yang bernilai *integer* dengan logika ”1” sehingga nantinya akan didapat kembali

bit-bit data informasi serta mengembalikan *sample rate* yang tereduksi akibat adanya blok *buffer* menjadi *sample rate* yang sesuai dengan inputnya. Gambar 3.20 merupakan isi dari *subsystem multiplier*.



Gambar 3. 20 Blok *Multiplier*

Parameter yang diatur pada blok *pulse generator* :

- ✓ *Pulse Type* : pilih *Sample based*
- ✓ *Time (t)* : *Use simulation time*
- ✓ *Amplitude* : 1
(Nilai amplitud sinyal yang diberikan)
- ✓ *Period (number of sample)* : 16
(1 sinyal OFDM 16 bit)
- ✓ *Pulse width (number of sample)* : 1
(Lebar pulsa yang dibangkitkan)
- ✓ *Phase delay (number of sample)* : 0 sampai 15
(Delay pulsa yang diinginkan, mulai dari 0 sampai 15 karena ada 16 bit data yang akan dijumlahkan secara bergantian)
- ✓ *Sample time (s)* : 1/100
(Dalam 1 detik ada 100 bit)

16 output dari *integrate and dump* dikalikan dengan 16 line input *pulse generator* dengan *phase delay* pulsa yang dibangkitkan dari 0 sampai 15, untuk menyusun kembali data paralel menjadi data serial. Antara *integrate and dump* dan *product* membutuhkan blok

rate transition untuk menyamakan laju bitnya. Penambahan blok *rate transition* pada tiap bit data informasi ini akan menambah *delay* dari simbol OFDM di sisi penerima sehingga delaynya menjadi 4 simbol OFDM (64 bit). Lalu semua hasil perkalian antara output *integrate and dump* dan *pulse generator* dijumlahkan.

Blok “sign” digunakan untuk mengubah semua nilai positif (+) menjadi “1” dan semua nilai negatif (-) menjadi “-1” sehingga data akan kembali dalam bentuk $[-1 \ 1]$. Kemudian data paralel dikonversi kembali menjadi data serial menggunakan blok “*Unbuffer*”. Cara kerja blok *unbuffer* ditunjukkan pada Gambar 3.21, dimana data yang telah diparalel sebanyak 16 bit dijadikan data serial kembali dengan urutan bit yang sama seperti sinyal informasi yang dibangkitkan. Sedangkan output data serial dari *unbuffer* ditunjukkan pada tabel 3.5.



Gambar 3. 21 Cara Kerja Blok *Unbuffer*

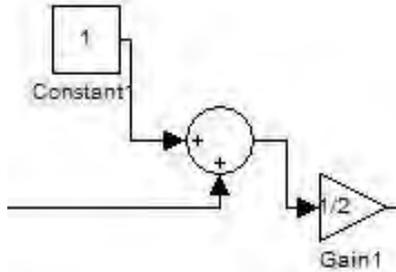
Tabel 3. 5 Bit Data Setelah Sign

Sinyal Informasi Serial														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-1	1	1	-1	1	1	1	-1	-1	-1	1	1	-1	1
-1	-1	-1	-1	1	1	-1	1	-1	1	-1	-1	-1	1	1
-1	1	-1	-1	-1										

1.3.6 Demapper

Demapper ini merupakan pengembalian sinyal dari $[-1 \ 1]$ menjadi data biner $[0 \ 1]$ kembali seperti inputnya. Untuk mengubah

kembali ke data biner, maka sinyal output dari *unbuffer* ditambah dengan konstanta “1”, kemudian dikuatkan dengan nilai gain “ $\frac{1}{2}$ ”. Sehingga akan diperoleh nilai sinyal output sistem. Sinyal output inilah yang akan dianalisa perhitungan *Bit Error Rate* (BER) nya.



Gambar 3. 22 Blok *Demapper*

Bit data output yang akan dianalisa BER-nya ditunjukkan pada tabel 3.6.

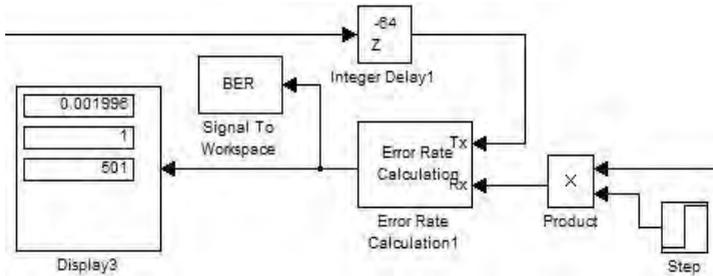
Tabel 3. 6 Bit Data Output

Sinyal Informasi Serial															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	1	1	0	1	1	1	0	0	0	1	1	0	1	1
0	0	0	0	1	1	0	1	0	1	0	0	0	1	1	1
0	1	0	0	0											

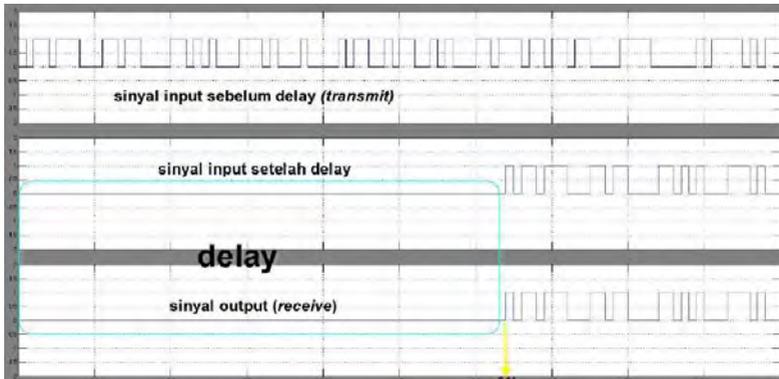
1.3.7 Perhitungan BER

Apabila nilai bit sinyal output tidak sama dengan nilai bit sinyal input yang dibangkitkan *Bernoulli* berarti terjadi kesalahan bit. Untuk membandingkan antara bit data input dan output digunakan blok *Error Rate Calculation*. Nilai BER, jumlah error, dan jumlah bit yang dikirim akan ditampilkan pada blok *display*.

Pada sistem ini delay yang dihasilkan oleh *receiver* sebanyak 64 bit sehingga di bagian *transmitter* harus ditambahkan blok *Integer Delay* sebesar 64. Selain itu di sisi *receiver* harus ditambahkan blok perkalian sinyal output dengan *step* yang dibangkitkan agar perhitungan bit data antara *transmitter* dan *receiver* sesuai.



Gambar 3. 23 Blok Perhitungan BER



Gambar 3. 24 Perbandingan Sinyal Input dan Output

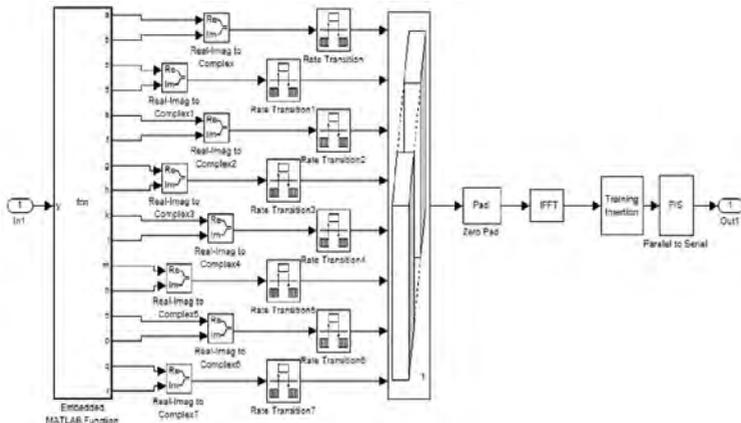
Parameter yang diatur pada blok *step* :

- ✓ *Step time* : 64/100
(Jumlah bit yang akan dikalikan dengan “0” adalah 64, lainnya dikalikan dengan “1”)
- ✓ *Initial value* : 0
(64 bit pertama bernilai “0”)

- ✓ *Final value* : 1
(Bit lainnya bernilai “1”)
- ✓ *Sample time (s)* : 1/100
(Dalam 1 detik ada 100 bit)

1.3.8 High Power Amplifier (HPA)

Blok HPA ini terletak sebelum melewati kanal AWGN. Namun ada kendala dalam implementasi HPA *nonlinear* pada sistem yang telah dibuat. Karena sistem yang telah dibuat adalah OFDM *passband* sedangkan simulasi menggunakan HPA dalam penelitian – penelitian sebelumnya menggunakan OFDM *baseband*. Sementara untuk persamaan HPA *nonlinear* OFDM *passband* belum ada yang sesuai untuk sistem yang telah dibuat. Oleh karena itu, dibuat kembali sistem OFDM *baseband* agar HPA *nonlinear* bisa diimplementasikan pada DSK TMS320C6416T. Untuk blok HPA OFDM *baseband* sudah ada di dalam Simulink.

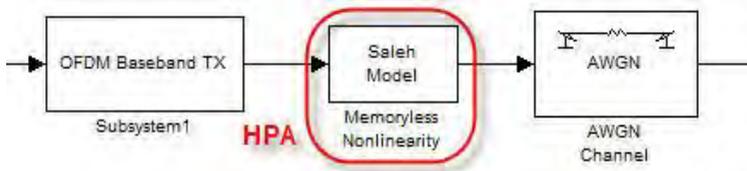


Gambar 3. 25 *Subsystem OFDM Baseband*

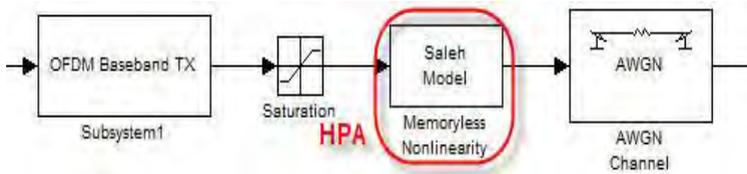
Dalam membuat sistem OFDM *baseband* hampir sama dengan OFDM *passband*. Pada OFDM *passband* terdapat proses perkalian sinyal yang masih kompleks (real - imajiner) dengan sinyal carrier \cos/\sin dengan frekuensi carrier tertentu. Sementara OFDM *baseband* outputnya masih berupa bilangan kompleks (real dan

imajier). Kemudian bilangan kompleks tersebut ditambahkan IFFT sebanyak jumlah subcarrier yang digunakan.

Gambar 3.26 merupakan blok HPA tanpa metode, dimana output dari sinyal OFDM *baseband* langsung masuk pada blok HPA *nonlinear* kemudian output dari HPA melewati kanal AWGN. Sedangkan Gambar 3.27 merupakan blok HPA menggunakan metode *clipping*, dimana ouput dari OFDM *baseband* harus masuk pada blok *saturation* terlebih dahulu untuk dilakukan proses *clipping* / pemotongan puncak sinyal. Setelah itu baru masuk pada blok HPA *nonlinear* lalu dilewatkan pada kanal AWGN.



Gambar 3. 26 Blok HPA Tanpa Metode



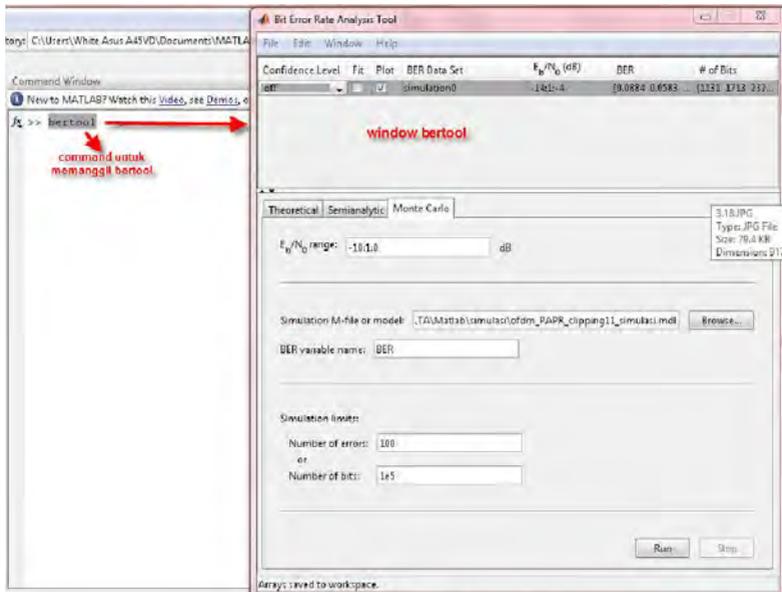
Gambar 3. 27 Blok HPA dengan Metode *Clipping*

1.3.9 Simulasi Sistem Menggunakan *Monte Carlo*

Pada tahap simulasi *monte carlo* ini, parameter blok kanal AGWN harus diubah menjadi “ $E_b/N_0(dB)$: E_bN_0 ” serta parameter yang diubah pada blok *Error Rate Calculation* adalah “*Target number of errors* : $maxNumErrs$ ” dan “*Maximum number of symbols* :

maxNumBits”. Selain itu, perlu ditambahkan blok *Signal To Workspace* untuk melihat nilai BER pada setiap nilai *EbNo* yang diberikan.

Untuk menjalankan simulasi, buka *bertool* melalui perintah pada *command window*. Gambar 3.28 akan menjelaskan lebih rinci parameter yang harus di-*setting* pada *monte carlo*.



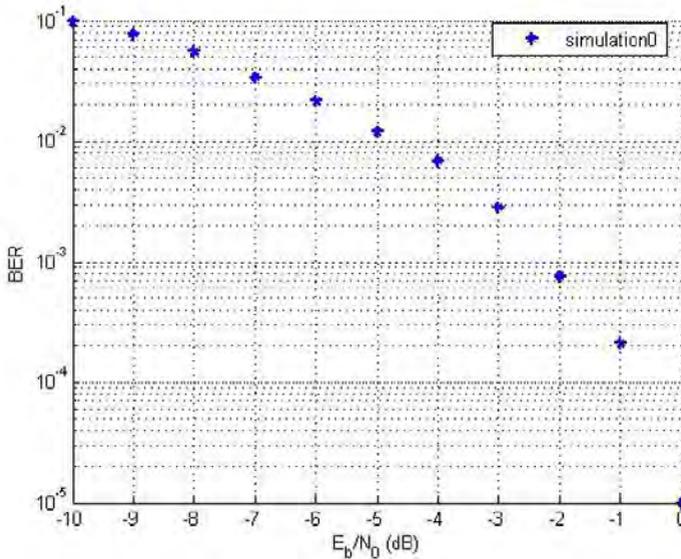
Gambar 3. 28 Bertool Pada Monte Carlo

Parameter yang diatur pada *Monte Carlo* adalah :

- ✓ *Eb/No range* : -10:1:0 dB
(nilai *Eb/N* di-*set* mulai -10 sampai 0 dB dengan step 1 dB)
- ✓ *Simulation M-file or model*
(pilih blok simulink yang akan disimulasikan)
- ✓ *BER variable name* : BER
(Sesuai dengan nama variabel yang diberikan pada blok *Signal To Workspace*)
- ✓ *Number of errors* : 100
(Jumlah error maksimal yang ingin dicapai)

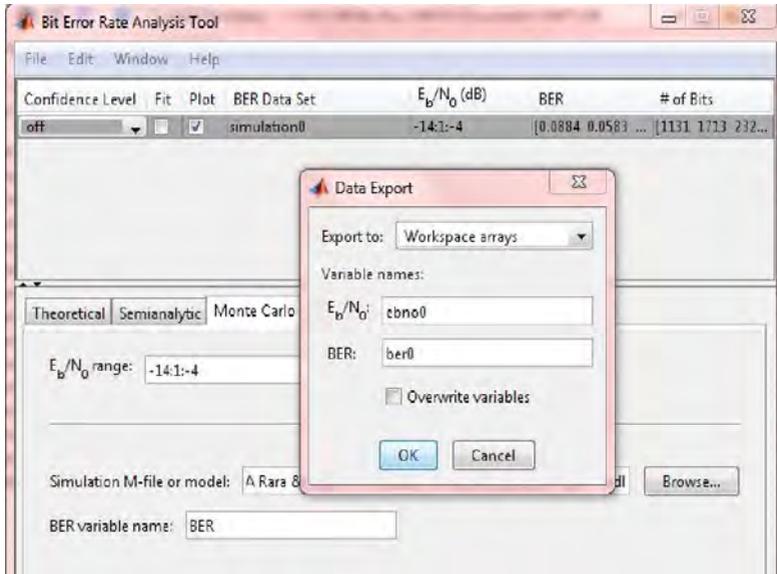
- ✓ *Number of bits* : 1e5
(Jumlah bit maksimal yang ingin dikirim)

Setelah semua parameter diisi, maka klik “run” untuk menjalankan simulasi. Simulasi akan terus berjalan menghitung BER sesuai dengan nilai E_b/N_0 yang telah ditentukan. Apabila salah satu kondisi pada “*Number of errors* atau *Number of bits*” terpenuhi, maka nilai BER akan ditampilkan pada *figure BER simulation*. Simulasi akan berhenti apabila nilai E_b/N_0 yang diberikan sudah dijalankan semua.

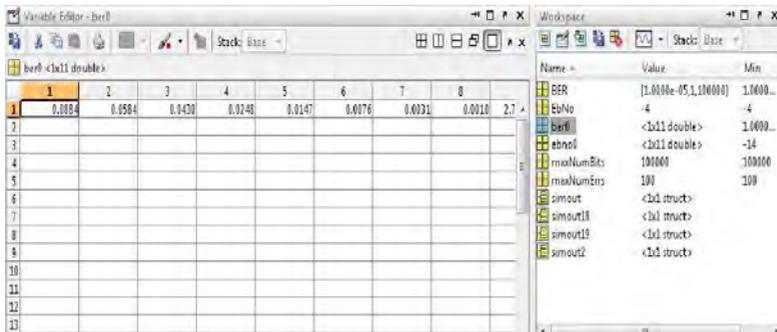


Gambar 3. 29 Grafik E_b/N_0 vs BER yang Dihasilkan dari Simulasi Bertool *Monte Carlo*

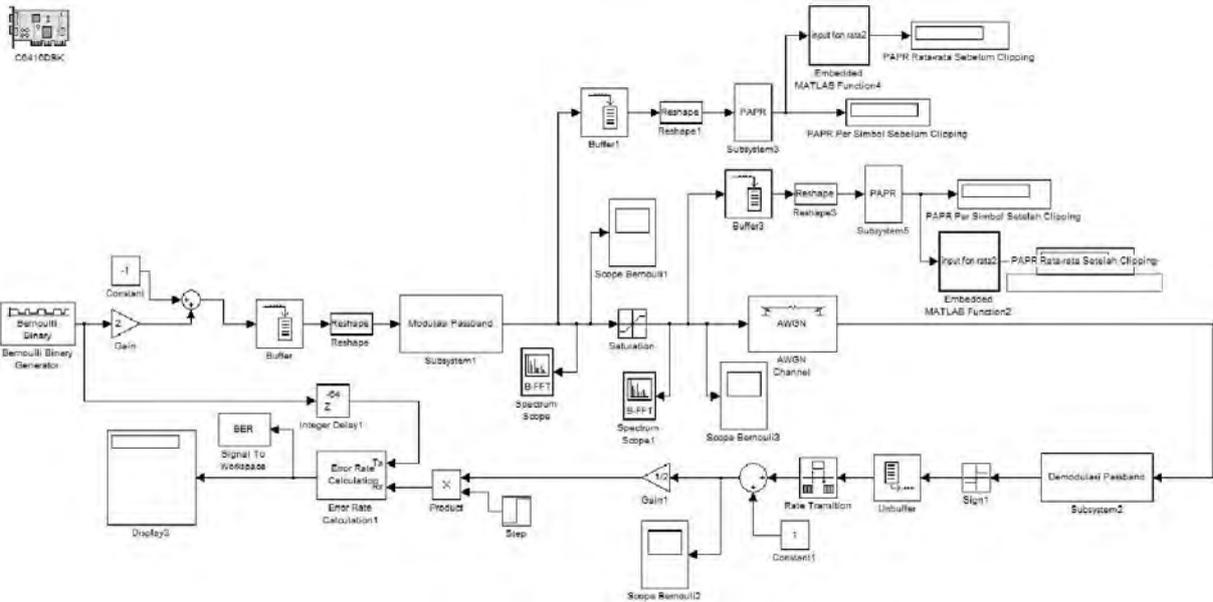
Selain itu, nilai BER tersebut bisa dilihat secara rinci, yaitu dengan cara *export* hasil simulasi dan melihat datanya pada *workspace* BER. Seperti yang ditampilkan pada Gambar 3.30 dan 3.31.



Gambar 3. 30 Export BER



Gambar 3. 31 Nilai BER yang telah di-export



Gambar 3. 32 Blok Simulasi Reduksi PAPR Menggunakan Metode *Clipping*

1.4 Implementasi Sistem Pada DSK TMS320C6416T

Adapun perangkat lunak yang digunakan dalam Tugas Akhir ini diantaranya MATLAB R2009a dan *Code Composer Studio* (CCS) v3.3 yang berjalan diatas sistem operasi Windows 7. Pada bagian ini hanya akan dijelaskan mengenai instalasi CCS v3.3 dan cara integrasi antara Matlab Simulink dengan CCS v3.3.

1.4.1 Matlab 2009a

MATLAB Simulink adalah suatu produk bawaan dari *software* MATLAB yang dikhususkan untuk tujuan pemodelan blok dan simulasi suatu sistem. *Software* MATLAB yang digunakan adalah MATLAB versi R2009a.

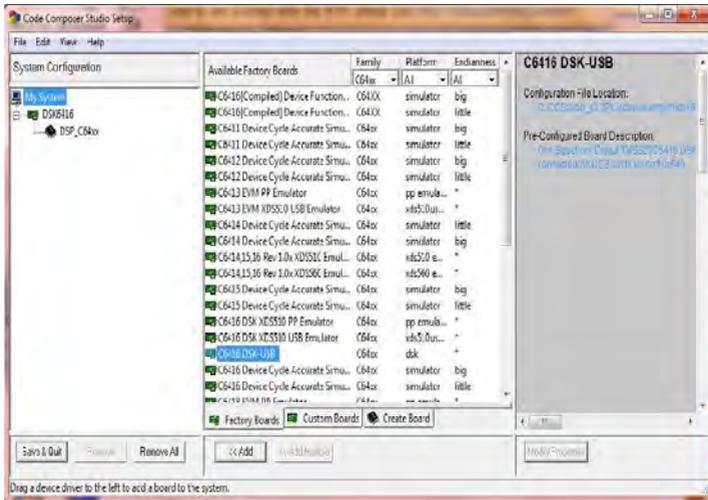
1.4.2 Instalasi *Code Composer Studio* (CCS) v3.3

Perangkat lunak atau *software code composer studio* berfungsi untuk menterjemahkan blok model sistem pada Simulink MATLAB sehingga dapat dijalankan pada perangkat DSK TMS320C6416T. *Software code composer studio* (CCS) yang dipakai adalah versi 3.3.



Gambar 3. 33 Tampilan awal Instalasi *Code Composer Studio*

Langkah selanjutnya adalah melakukan instalasi C6000 Code Composer Studio v3.3. Langkah yang kedua yaitu menginstal FlashBurn v2.80 dan yang terakhir melakukan instalasi DSK6416 Drivers and Target Content dimana instalasi yang terakhir ini dibutuhkan sesuai dengan prosesor yang akan digunakan.



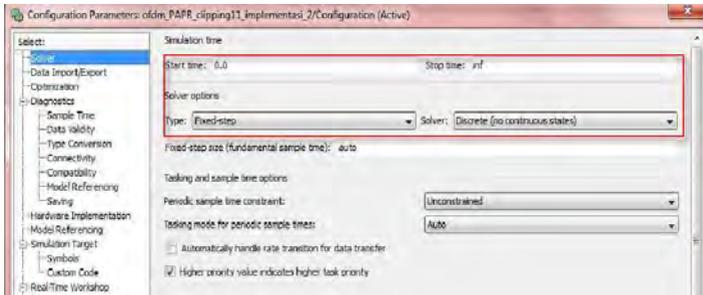
Gambar 3. 34 Tampilan Pilihan Prosesor yang Digunakan

1.4.3 Integrasi antara *Matlab Simulink* dengan *CCS*

Tahap selanjutnya yaitu mengintegrasikan Simulink dengan CCS v3.3 blok sistem telah dibuat di Simulink dapat pada DSK TMS320C6416T. Untuk dapat *build* sistem ke dalam *DSK board* diperlukan beberapa pengaturan konfigurasi yang dapat diakses pada menu bar, pilih “*Simulation*” kemudian pilih “*Configuration Parameter*”. Beberapa parameter yang harus diatur sebagai berikut :

1. *Solver*

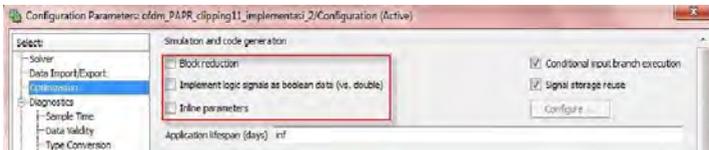
Yang diubah adalah bagian *Solver option*, yaitu “Type : Fixed-step” dan “Solver : Discrete (no continuous states)”.



Gambar 3. 35 Pengaturan *Solver*

2. *Optimization*

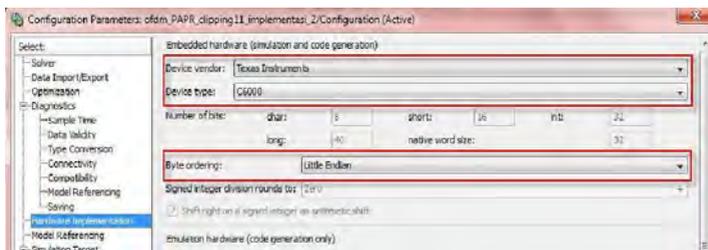
Pada *Simulation and code generation*, tanda centang pada *block reduction* dan *implement logic signal as boolean data* dihilangkan. Tujuannya agar Simulink tidak menterjemahkan blok-blok yang tidak terpakai untuk diubah dalam bahasa C.



Gambar 3. 36 Pengaturan *Optimization*

3. *Hardware Implementation*

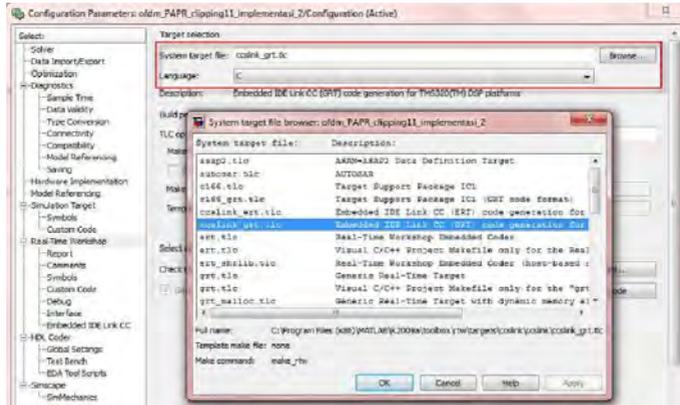
Parameter yang diatur adalah “*Device vendor : Texas Instrument*”, “*Device type : C6000*”, dan “*Byte ordering : Little Endian*”.



Gambar 3. 37 Pengaturan *Hardware Implementation*

4. Real-Time Workshop

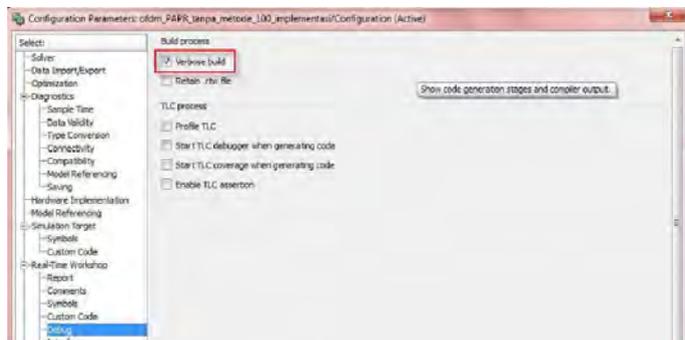
Bagian yang diubah adalah “System target file : pilih *ccslink_grt.tlc*”. Hal ini bertujuan untuk menghubungkan antara Simulink dengan CCS.



Gambar 3. 38 Pengaturan Real-Time Workshop

5. Debug

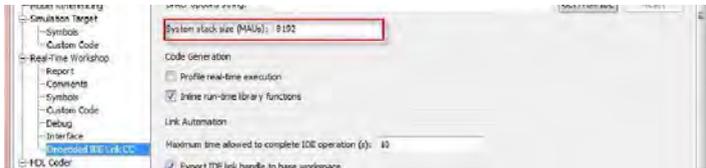
Tandai pilihan *Verbose Build Mode* untuk menampilkan proses pembentukan kode-kode dari file simulink untuk memprogram DSK yang ditampilkan pada *Matlab Command Windows*.



Gambar 3. 39 Pengaturan Debug

6. *Embedded IDE Link CC*

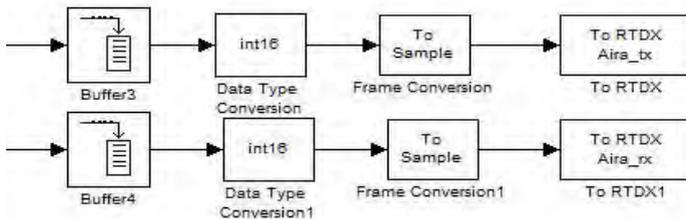
Untuk bagian *Project options*, ubah *System stack size* menjadi 8192 bytes. Nilai ini disesuaikan dengan maksimum *stack size* pada DSK board.



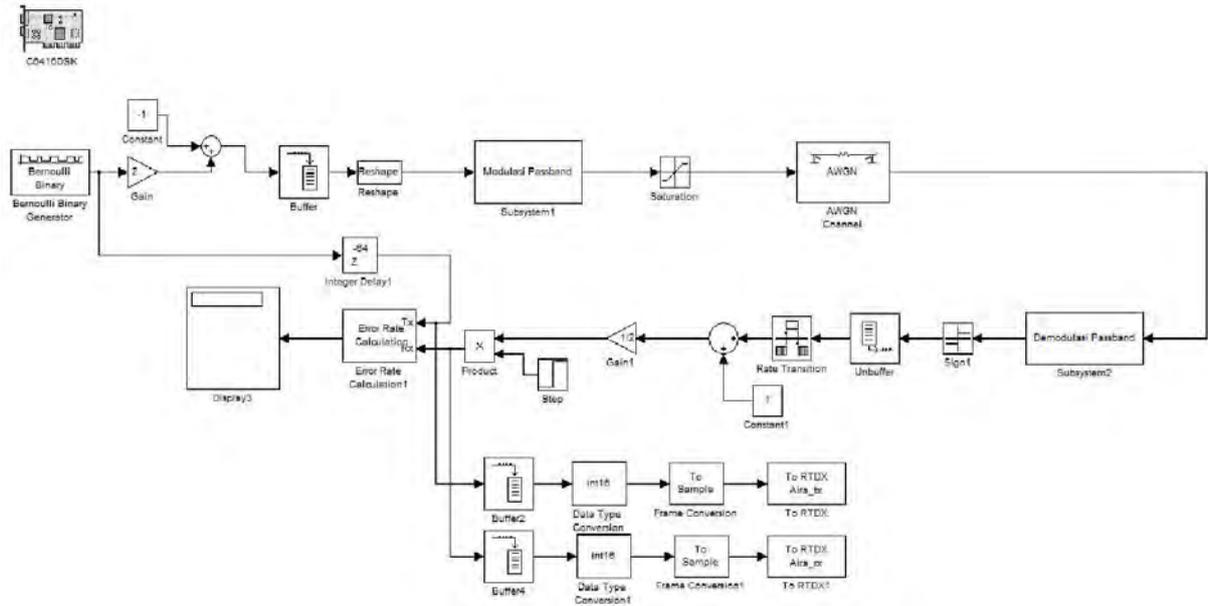
Gambar 3. 40 Pengaturan *Embedded IDE Link CC*

1.4.4 Proses Pengujian Keandalan Sistem

Sistem yang telah diimplementasikan kedalam DSK TMS320C6416 diuji kinerjanya dengan parameter BER (Bit Error Rate) terhadap variasi E_b/N_0 tertentu. Variasi E_b/N_0 yang terdapat pada kanal AWGN diubah-ubah sehingga didapatkan hasil kinerja sistem terhadap perubahan E_b/N_0 yang dilakukan. Setelah *Matlab Simulink* terintegrasi dengan *CCS v3.3*, maka sistem yang telah dibuat langsung di-build, tetapi sebelumnya blok sistem harus ditambahkan blok RTDX. Blok RTDX pada implementasi digunakan untuk mengambil hasil keluaran dari sistem yang telah diimplementasikan kedalam DSK. Terdapat 4 blok yang dibutuhkan dalam blok RTDX ini yaitu blok *Buffer*, *Data Type Conversion*, *Frame Conversion*, dan *To RTDX*. Pada blok *To RTDX* dapat diberi channel name secara bebas dan unik, contohnya "Aira_tx". Dalam sistem ini, *buffer* yang diberikan sebesar 500.



Gambar 3. 41 Blok *RTDX*



Gambar 3. 42 Blok Implementasi Reduksi PAPR Menggunakan Metode *Clipping*

Semua sistem akan diuji pada laju bit 1 kbps dan dengan kondisi jumlah bit yang dikirimkan adalah sebanyak 100.000 bit. Sistem ini akan dilewatkan kedalam kanal AWGN yang divariasikan nilai Eb/N0. Variasi nilai Eb/N0 dari -10 dB sampai 0 dB. Apabila terdapat kesalahan pada sisi penerima, maka jumlah bit salah akan dibandingkan dengan jumlah bit yang dikirim sehingga nantinya didapatkan nilai BER (Bit Error Rate).

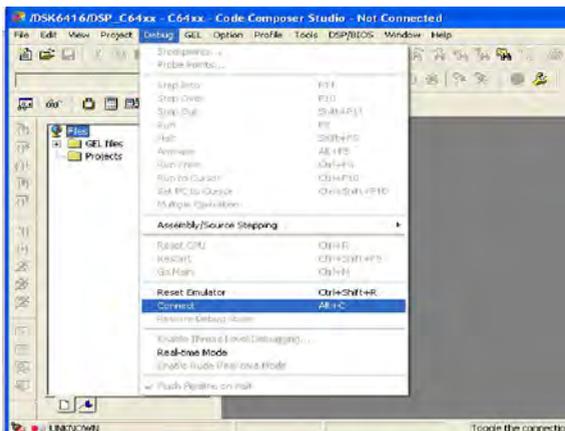
Untuk memulai implementasi, pastikan perangkat DSK TMS320C6416T sudah terhubung dengan PC/laptop. Berikut langkah – langkah dalam menjalankan implementasi :

1. Buka software 6416 DSK CCStudio v3.3. Klik kanan, *run as Administrator*



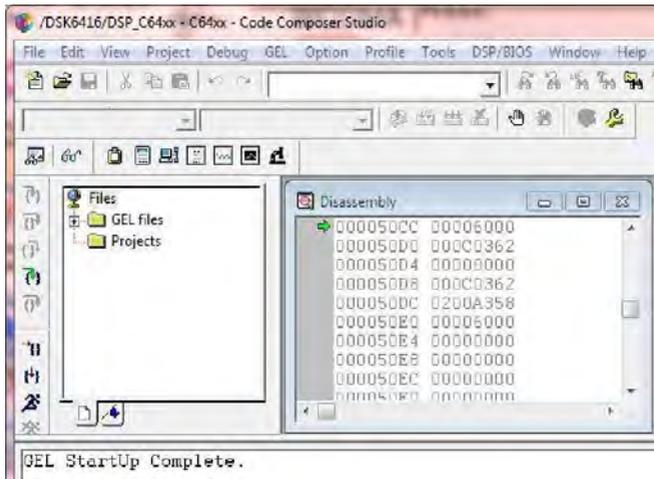
Gambar 3. 43 Ikon 6416 DSK CCStudio v3.3

2. Untuk menghubungkan software CCS dengan DSK board, pilih menu *debug*, klik *connect*



Gambar 3. 44 Setting *connect* software dengan DSK Board

Setelah *connect* berhasil, tampilan software CCS seperti Gambar 3.45.



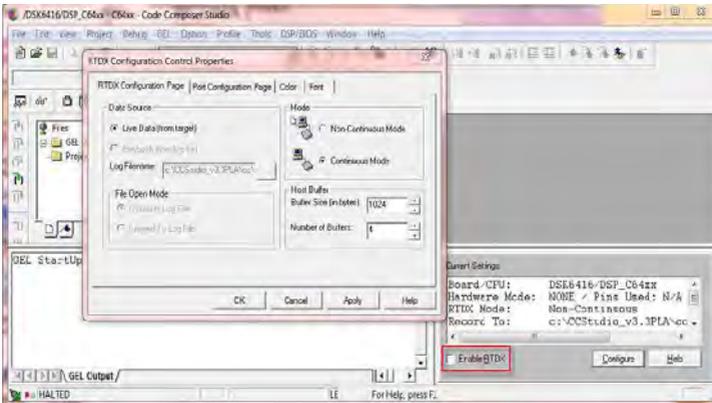
Gambar 3. 45 Connect software dengan DSK Board Berhasil

3. Setting konfigurasi RTDX. Pilih menu *Tools* > *RTDX* pilih *Configuration Control* dan pengaktifan RTDX dapat dilakukan dengan cara memilih *Enable RTDX*.



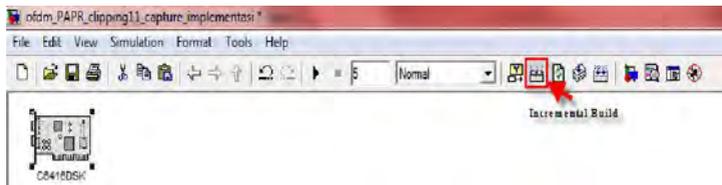
Gambar 3. 46 Menu *Config* RTDX

Kemudian pilih ‘*Continuous Mode*’ pada *RTDX Configuration Page*. Setelah itu centang ‘*Enable RTDX*’. Apabila RTDX telah aktif, DSK yang telah di program akan mengirim data yang hasil keluaran sistem sehingga data tersebut nantinya dapat diolah kedalam komputer menggunakan Matlab.



Gambar 3. 47 Configuration RTDX

4. File simulink yang telah dibuat dapat diimplementasikan pada DSK dengan cara klik *incremental build* yang terdapat pada simulink. Menu *incremental build* ini berfungsi untuk mengkomunikasikan Matlab dengan Code Composer Studio sehingga file simulink yang telah dibuat dapat menjadi file project CCS (*.pj) dimana berisi hasil konversi file simulink kedalam bahasa C dan *assembly*.



Gambar 3. 48 Button Incremental Build

[Halaman ini sengaja dikosongkan]

BAB IV

PENGUJIAN DAN ANALISA DATA

Pada bab ini dibahas tentang analisa hasil simulasi dan implementasi reduksi PAPR pada sinyal OFDM menggunakan teknik *clipping*. Ada 2 hal utama yang akan dianalisa pada bab ini, yaitu :

1. Analisa penurunan PAPR dengan metode *clipping*.
Untuk menganalisa nilai PAPR dengan metode *clipping*, parameter yang diubah – ubah adalah level *clipping* (A), *initialseed Bernoulli*, banyak bit yang dikirim.
2. Analisa *Bit Error Rate* (BER)
Analisa BER dengan mengubah nilai Eb/No dari blok AWGN (Additive White Gaussian Noise). Hasil dari pengujian sistem ditampilkan pada grafik BER. Pengujian hasil simulasi sistem dibandingkan terhadap hasil implementasi sistem yang dilakukan.

1.1 Analisa Penurunan PAPR Dengan Metode *Clipping*

Setelah dilakukan simulasi sistem beberapa kali dengan mengubah parameter level *clipping* (5 - 9 dengan step 0.5), *initialseed bernoulli*, banyak bit yang dikirim (100, 1000, 10000, 100000), jumlah *subcarrier* yang digunakan (8, 16, 24 *subcarrier*) maka diperoleh data penurunan nilai PAPR.

1.1.1 Perubahan Parameter Level *Clipping*

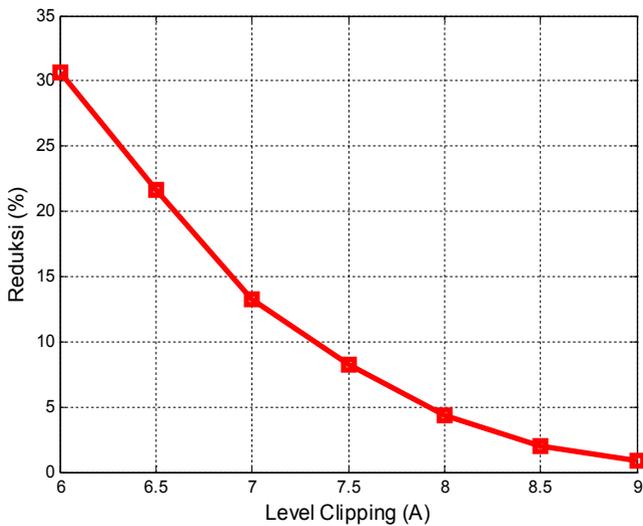
Simulasi pertama adalah menghitung nilai PAPR pada 100000 bit data menggunakan 8 *subcarrier* dan *initialseed* = 61 dengan perubahan parameter level *clipping*. Level *clipping* yang digunakan adalah mulai 5 sampai 9 dengan selisih 0.5.

Berdasarkan data pada tabel 4.1, perubahan level *clipping* atau amplitude maksimal dari sinyal OFDM yang akan dikirim sangat mempengaruhi penurunan nilai PAPR. Sebelum di reduksi, amplitude maksimal dari sinyal OFDM adalah 9.18 dan minimumnya 9.06. Setelah itu dihitung kuadrat daya maksimal dan rata-rata untuk menentukan nilai PAPR. Dari 100000 bit data yang dikirim, diperoleh PAPR rata-rata sebelum direduksi sebesar 6.81.

Tabel 4. 1 Tabel Penurunan PAPR dengan Perubahan Level *Clipping*

Level <i>Clipping</i> (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
9	6.81	6.748	0.910426	3.18
8.5	6.81	6.671	2.041116	3.01
8	6.81	6.511	4.390602	2.83
7.5	6.81	6.249	8.237885	2.65
7	6.81	5.862	13.9207	2.47
6.5	6.81	5.336	21.64464	2.3
6	6.81	4.721	30.67548	2.12

Kemudian untuk mengetahui pengaruh level clipping pada reduksi PAPR, amplitudo sinyal tersebut dipotong mulai dari level 9, 8.5, 8, 7.5, 7, 6.5, 6. Artinya, apabila level *clipping* = 9, amplitudo dari sinyal OFDM yang dikirim adalah -9 sampai 9. Grafik penurunan reduksi PAPR terhadap level *clipping* bisa dilihat pada Gambar 4.1.



Gambar 4. 1 Grafik Reduksi PAPR Terhadap Level *Clipping*

Berdasarkan grafik pada Gambar 4.1, dapat diketahui bahwa semakin kecil level *clipping* maka reduksi PAPR semakin besar. Dimana saat level *clipping* = 9, reduksi PAPR yang dihasilkan hanya 0.91% sedangkan saat level *clipping* = 6, reduksi PAPR yang dihasilkan mencapai 30.67%.

1.1.2 Perubahan Parameter *Initialseed*

Simulasi kedua adalah menghitung nilai PAPR pada 100000 bit data dan 8 *subcarrier* dengan perubahan parameter *Initialseed*. *Initialseed* yang diberikan adalah 61, 62, 63, 64, 65. Untuk data pada *initialseed* 61, bisa dilihat pada tabel 4.1 yang sudah ditampilkan pada pembahasan sebelumnya, sedangkan lainnya dilihat pada Tabel 4.2, 4.3, 4.4, 4.5.

Tabel 4. 2 Tabel Penurunan PAPR *Initialseed* = 62

Level <i>Clipping</i> (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
9	6.774	6.713	0.900502	3.18
8.5	6.774	6.635	2.051963	3.01
8	6.774	6.478	4.369649	2.83
7.5	6.774	6.224	8.11928	2.65
7	6.774	5.848	13.66991	2.47
6.5	6.774	5.33	21.3168	2.3
6	6.774	4.715	30.39563	2.12

Tabel 4. 3 Tabel Penurunan PAPR *Initialseed* = 63

Level <i>Clipping</i> (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
9	6.798	6.739	0.867902	3.18
8.5	6.798	6.658	2.059429	3.01
8	6.798	6.494	4.471904	2.83
7.5	6.798	6.234	8.296558	2.65
7	6.798	5.885	13.43042	2.47
6.5	6.798	5.331	21.57988	2.3
6	6.798	4.719	30.58252	2.12

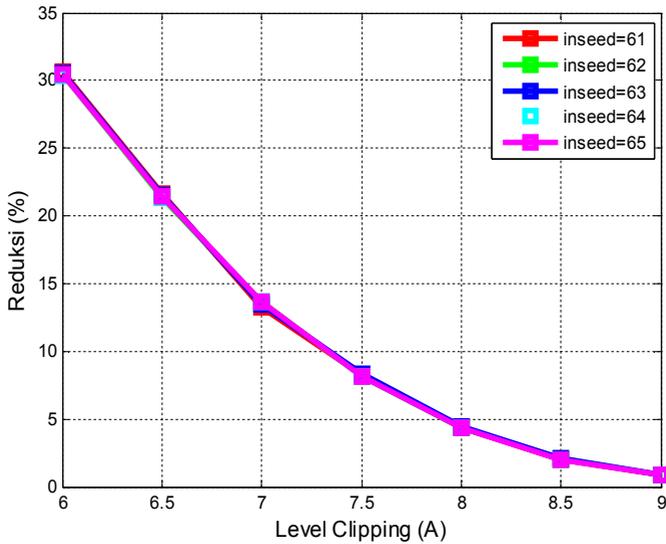
Tabel 4. 4 Tabel Penurunan PAPR *Initialseed* = 64

Level Clipping (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
9	6.768	6.711	0.842199	3.18
8.5	6.768	6.636	1.950355	3.01
8	6.768	6.477	4.299645	2.83
7.5	6.768	6.22	8.096927	2.65
7	6.768	5.845	13.63771	2.47
6.5	6.768	5.322	21.36525	2.3
6	6.768	4.713	30.36348	2.12

Tabel 4. 5 Tabel Penurunan PAPR *Initialseed* = 65

Level Clipping (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
9	6.783	6.724	0.869822	3.18
8.5	6.783	6.648	1.99027	3.01
8	6.783	6.487	4.363851	2.83
7.5	6.783	6.229	8.167478	2.65
7	6.783	5.853	13.71075	2.47
6.5	6.783	5.327	21.46543	2.3
6	6.783	4.715	30.48798	2.12

Dari Tabel 4.1 hingga 4.5, dapat dikatakan bahwa parameter *initialseed* tidak terlalu mempengaruhi penurunan reduksi PAPR. Hal ini terlihat dari grafik yang ditunjukkan pada Gambar 4.4, dimana dengan perbedaan nilai *initialseed* ternyata hasil reduksi PAPR-nya saling berhimpit. Misal pada *initialseed* 62 dan 64 untuk level clipping = 8, reduksi PAPR yang dihasilkan hampir sama yaitu sekitar 4.3%. Karena pada dasarnya *initialseed* disini berpengaruh pada pengacakan bit data yang akan dikirim. Selama probabilitas antara bit “0” dan “1” sama, yaitu “0.5” maka nilai berapapun *initialseed* yang diberikan akan menghasilkan reduksi PAPR yang hampir sama.

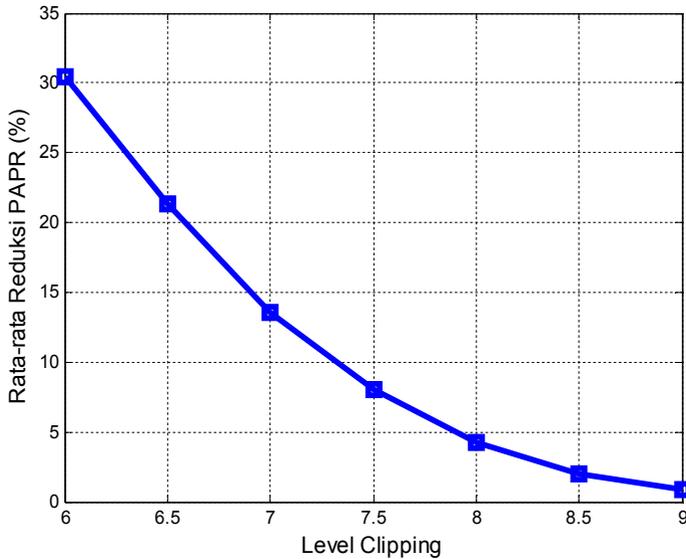


Gambar 4. 2 Grafik Reduksi PAPR Terhadap Level Clipping Pada *Initialseed* yang Berbeda

Selain 5 perubahan *initialseed* di atas, dilakukan 10 kali simulasi dengan 10 kali perubahan *initialseed*. Kemudian dihitung rata-rata dari 10 data hasil reduksi PAPR yang diperoleh. Rata-rata reduksi PAPR ditunjukkan pada Tabel 4.6 dan Gambar 4.3.

Tabel 4. 6 Tabel Reduksi PAPR Rata-rata

Level Clipping (A)	Reduksi (%)
9	0.826986
8.5	1.95085
8	4.288588
7.5	8.076189
7	13.60727
6.5	21.38607
6	30.43352



Gambar 4. 3 Grafik Hasil Reduksi PAPR Rata-rata

1.1.3 Perubahan Parameter Jumlah Bit yang Dikirim

Simulasi ketiga adalah menghitung nilai PAPR dengan *initialseed* =61 dan menggunakan 8 *subcarrier* pada jumlah bit data yang berbeda, yaitu mulai 100 bit, 1000 bit, 10000 bit, 100000 bit.

Tabel 4. 7 Tabel Penurunan PAPR Jumlah bit yang dikirim = 100

Level <i>Clipping</i> (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
9	5.899	5.899	0	3.18
8.5	5.899	5.899	0	3.01
8	5.899	5.899	0	2.83
7.5	5.899	5.683	3.661637566	2.65
7	5.899	5.471	7.255467028	2.47
6.5	5.899	5.273	10.61196813	2.3
6	5.899	4.58	22.35972199	2.12

Tabel 4. 8 Tabel Penurunan PAPR Jumlah bit yang dikirim = 1000

Level Clipping (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
9	6.873	6.821	0.756583733	3.18
8.5	6.873	6.738	1.96420777	3.01
8	6.873	6.558	4.583151462	2.83
7.5	6.873	6.294	8.424268878	2.65
7	6.873	5.909	14.02589844	2.47
6.5	6.873	5.376	21.78088171	2.3
6	6.873	4.735	31.10723119	2.12

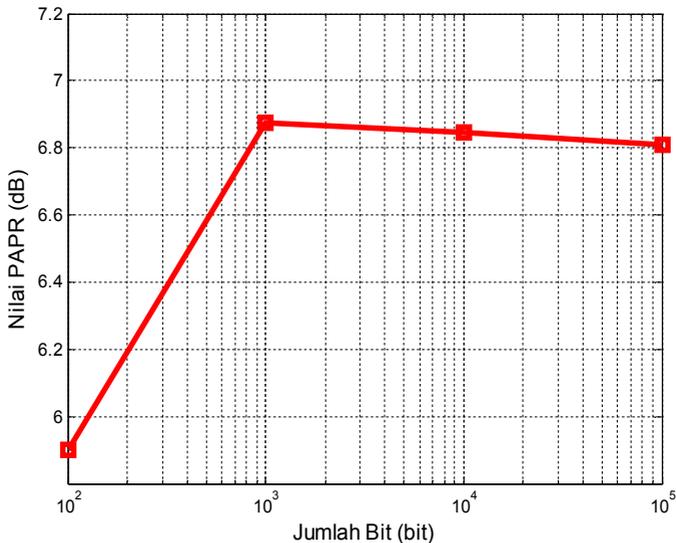
Tabel 4. 9 Tabel Penurunan PAPR Jumlah bit yang dikirim = 10000

Level Clipping (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
9	6.844	6.792	0.759789597	3.18
8.5	6.844	6.715	1.884862653	3.01
8	6.844	6.552	4.266510812	2.83
7.5	6.844	6.281	8.226183518	2.65
7	6.844	5.877	14.12916423	2.47
6.5	6.844	5.342	21.94623027	2.3
6	6.844	4.725	30.96142607	2.12

Tabel 4. 10 Tabel Penurunan PAPR Jumlah bit yang dikirim = 100000

Level Clipping (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
9	6.81	6.748	0.910426	3.18
8.5	6.81	6.671	2.041116	3.01
8	6.81	6.511	4.390602	2.83
7.5	6.81	6.249	8.237885	2.65
7	6.81	5.862	13.9207	2.47
6.5	6.81	5.336	21.64464	2.3
6	6.81	4.721	30.67548	2.12

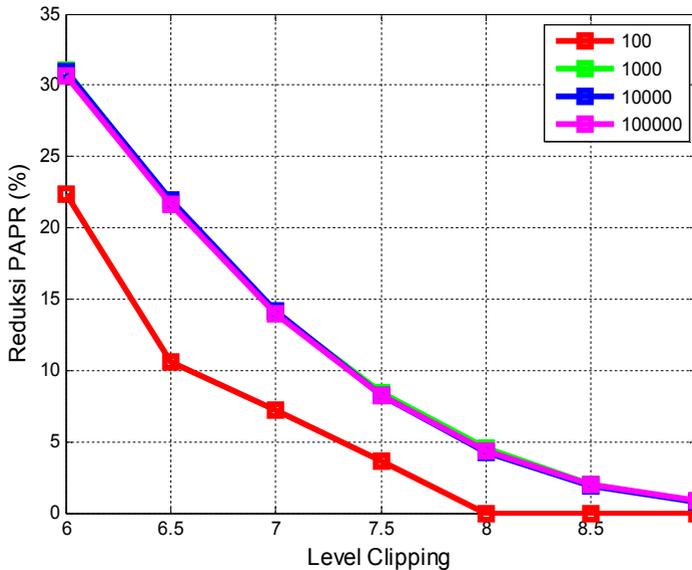
Dari Tabel 4.7 hingga 4.10 bisa diketahui bahwa saat data yang dikirim sebanyak 100000 bit, nilai PAPR rata-rata tanpa metode *clipping* yang diperoleh adalah 6.81. Ketika data 10000 bit, nilai PAPR rata-ratanya 6.844 dan ketika data 1000 nilai PAPR rata-ratanya 6.873. Dari ketiga jumlah bit data yang berbeda tersebut, seolah-olah dapat disimpulkan semakin banyak simbol OFDM yang dikirim maka nilai PAPR rata-rata yang diperoleh semakin kecil. Padahal sebenarnya tidak, karena saat jumlah bit yang dikirim 100, nilai PAPR rata-rata tanpa metode sebesar 5.899. Jadi jumlah bit yang dikirim hanya mempengaruhi seberapa banyak jumlah simbol OFDM yang dihitung nilai PAPRnya.



Gambar 4. 4 Grafik Nilai PAPR Tanpa Metode Terhadap Jumlah Bit yang Dikirim

Saat jumlah bit yang dikirim 100, reduksi PAPR yang dihasilkan tidak terlalu baik bahkan pada level *clipping* 9, 8.5, 8 nilai PAPR rata-rata antara sebelum dan sesudah *clipping* sama sehingga reduksi yang dihasilkan “0%”. Hal ini karena data yang dibangkitkan terlalu sedikit,

yaitu 100 bit sementara *delay*-nya 64 bit sehingga hanya tersisa 36 bit (2 simbol OFDM). Jadi, dari 2 simbol OFDM tersebut tidak ada yang nilai amplitudonya lebih dari 8 sehingga tidak ada sinyal yang terpotong amplitudonya dan tidak ada PAPR yang tereduksi. Sedangkan saat level *clipping* yang diberikan adalah 7.5, 7, 6.5, 6, baru ada proses pemotongan puncak amplitudo sinyal dan terjadi reduksi PAPR.



Gambar 4. 5 Grafik Reduksi PAPR Terhadap Level Clipping Pada Jumlah Bit yang Berbeda

Gambar 4.5 menunjukkan grafik perbandingan hasil reduksi PAPR terhadap level *clipping* yang diberikan pada jumlah bit yang berbeda. Jika dilihat dari grafik pada Gambar 4.5, perbedaan jumlah bit yang dikirim tidak mempengaruhi reduksi PAPR. Jumlah bit data yang dikirim hanya berpengaruh pada data statistik dari PAPR rata-rata. Semakin banyak jumlah bit yang dikirim, berarti semakin banyak pula jumlah simbol OFDM yang dihitung nilai PAPR-nya.

1.1.4 Perubahan Parameter Jumlah *Subcarrier*

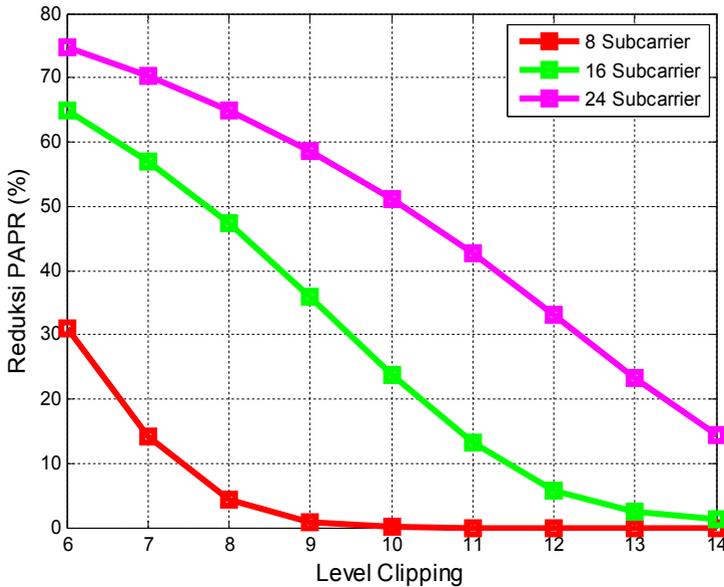
Simulasi keempat adalah menghitung nilai PAPR pada 10000 bit, *initialseed* = 61, dengan perubahan jumlah *subcarrier*, yaitu 8, 16, 24 *subcarrier*. Data reduksi PAPR pada pembahasan sebelumnya menggunakan 8 *subcarrier*. Dan disini, akan ditunjukkan data hasil reduksi PAPR menggunakan 16 *subcarrier* dan 24 *subcarrier*. Data hasil reduksi PAPR pada 16 *subcarrier* dan 24 *subcarrier* dapat dilihat pada Tabel 4.11 dan 4.12.

Tabel 4. 11 Tabel Penurunan PAPR Jumlah *Subcarrier* = 16

Level <i>Clipping</i> (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
14	8.299	8.199	1.204964	3.5
13	8.299	8.088	2.542475	3.25
12	8.299	7.823	5.735631	3
11	8.299	7.209	13.13411	2.75
10	8.299	6.333	23.6896	2.5
9	8.299	5.312	35.99229	2.25
8	8.299	4.372	47.31895	2
7	8.299	3.567	57.01892	1.75
6	8.299	2.903	65.01988	1.5

Tabel 4. 12 Tabel Penurunan PAPR Jumlah *Subcarrier* = 24

Level <i>Clipping</i> (A)	PAPR Rata-rata Tanpa metode	PAPR Rata-rata Metode <i>clipping</i>	Reduksi (%)	CR
14	9.159	7.833	14.47756	2.86
13	9.159	7.032	23.22306	2.65
12	9.159	6.132	33.04946	2.45
11	9.159	5.254	42.63566	2.25
10	9.159	4.475	51.14095	2.04
9	9.159	3.797	58.54351	1.84
8	9.159	3.215	64.89791	1.63
7	9.159	2.721	70.29152	1.43
6	9.159	2.306	74.82258	1.22



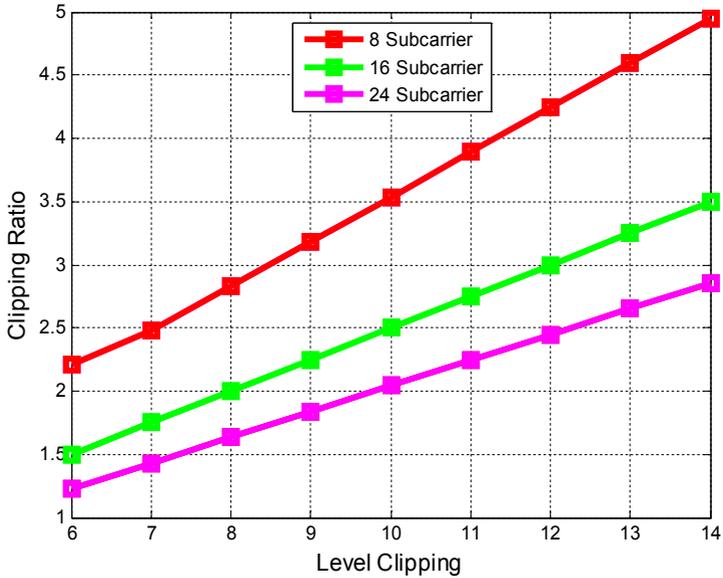
Gambar 4. 6 Grafik Reduksi PAPR Terhadap Level Clipping Pada Jumlah Subcarrier yang Berbeda

Berdasarkan grafik pada Gambar 4.6, untuk level *clipping* = 9, besar reduksi PAPR yang dihasilkan oleh 8 *subcarrier* = 0.759, 16 *subcarrier* = 35.992, 24 *subcarrier* = 58.543. Pada 8 *subcarrier*, Dari data tersebut, diketahui bahwa jumlah *subcarrier* sangat mempengaruhi nilai reduksi PAPR. Semakin banyak jumlah *subcarrier*, maka semakin banyak pula reduksi PAPR yang dihasilkan.

Perubahan jumlah *subcarrier* yang digunakan mempengaruhi nilai *Clipping Ratio* (CR). Perbandingan nilai *clipping ratio* ditunjukkan pada gambar 4.7. Saat level *clipping* = 9, nilai *clipping ratio* pada 8 *subcarrier* = 3.2, 16 *subcarrier* = 2.25, 24 *subcarrier* = 1.8. Dari data tersebut bisa ditarik kesimpulan bahwa semakin banyak jumlah *subcarrier*, maka nilai *clipping ratio* semakin kecil.

Selain itu, dari gambar tersebut juga dapat dilihat pengaruh *level clipping* terhadap nilai *clipping ratio*. Semakin besar nilai *level clipping*

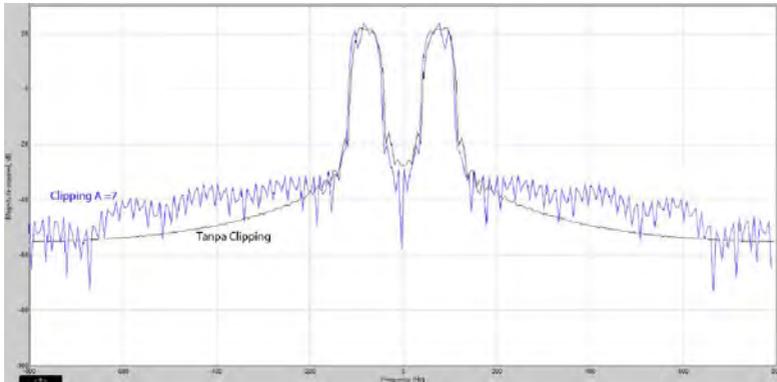
yang diberikan, maka semakin besar pula nilai *clipping ratio*. Hal ini sesuai dengan persamaan (2.9) dan (2.10), dimana *clipping ratio* sebanding dengan level *clipping* dan berbanding terbalik dengan jumlah *subcarrier*.



Gambar 4. 7 Grafik Perbandingan *Clipping Ratio* pada 8 *Subcarrier*, 16 *Subcarrier*, 24 *Subcarrier*

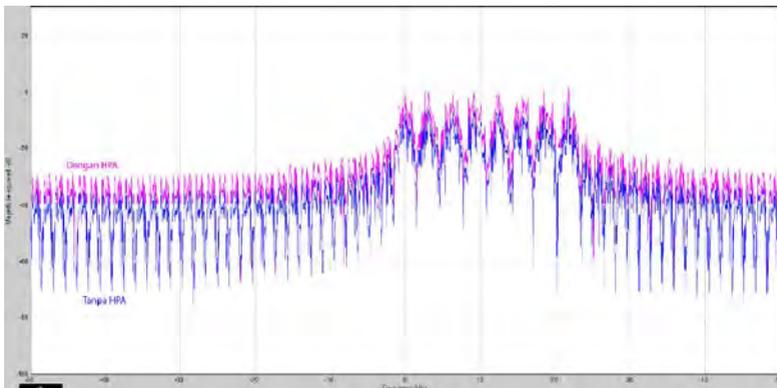
1.2 Analisa Spektrum Frekuensi

Selain itu, pengaruh metode reduksi PAPR juga bisa dilihat dari spektrum frekuensinya. Pada *subbab* ini dianalisa perbandingan spektrum frekuensi antara sebelum melewati HPA (sebelum *clipping* dan setelah *clipping*) dan setelah melewati HPA (sebelum *clipping* dan setelah *clipping*). Spektrum frekuensi ini diamati saat jumlah data yang dikirim 100000 bit dan menggunakan 8 *subcarrier*.



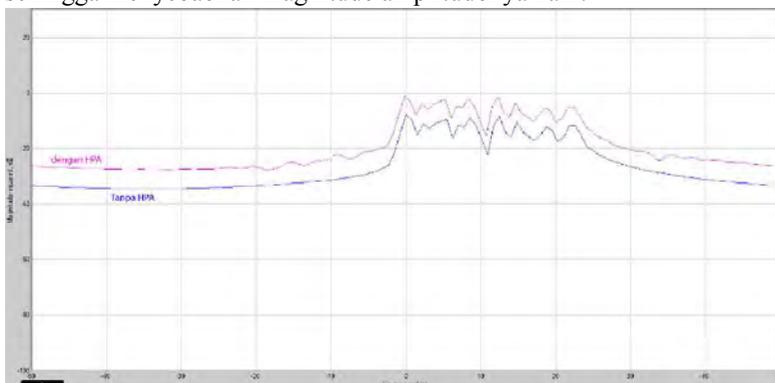
Gambar 4. 8 Perbandingan Spektrum Frekuensi Passband Tanpa *Clipping* dan Dengan *Clipping* ($A = 7$) Sebelum Melewati HPA

Gambar 4.8 merupakan perbandingan spektrum frekuensi *passband* tanpa *clipping* dan dengan *clipping* sebelum melewati HPA. Dari Gambar 4.8, diketahui bahwa spektrum frekuensi sebelum *clipping* lebih bagus daripada setelah menggunakan *clipping*. Setelah *clipping*, terdapat banyak ripple pada *side lobe*-nya dan pada *main lobe* juga ada beberapa ripple. Hal ini karena pada proses *clipping* terjadi pencacatan sinyal.



Gambar 4. 9 Spektrum Frekuensi Baseband Dengan HPA dan Tanpa HPA (2048 IFFT)

Gambar 4.9 merupakan perbandingan spectrum frekuensi *baseband* sebelum dan setelah melewati HPA OFDM *baseband* yang dengan 2048 IFFT sehingga spektrum frekuensi yang dihasilkan memiliki banyak ripple. Sedangkan Gambar 4.10 merupakan perbandingan spectrum frekuensi *baseband* sebelum dan setelah melewati HPA OFDM *baseband* yang dengan 128 IFFT. Spektrum frekuensi yang dihasilkan dengan 128 IFFT lebih halus daripada 2048 IFFT. Berdasarkan gambar tersebut, setelah melewati HPA magnitude amplitudonya naik sekitar 5 dB. Sebelum HPA amplitudonya -32.5 dB sedangkan setelah HPA amplitudonya -37.5 dB. Hal ini karena setelah melewati HPA, sinyal dikuatkan dahulu menggunakan amplifier sehingga menyebabkan magnitude amplitudonya naik.



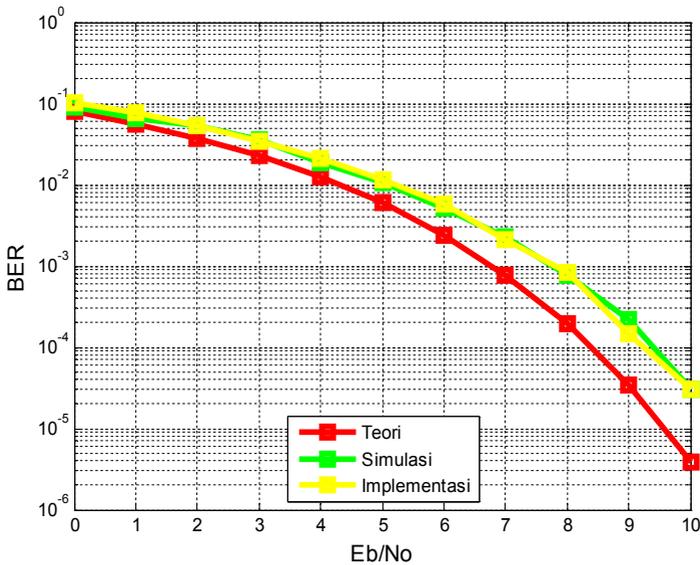
Gambar 4. 10 Spektrum Frekuensi Baseband Dengan HPA dan Tanpa HPA (128 IFFT)

1.3 Analisa *Bit Error Rate* (BER)

Pengujian dilakukan pada kanal AWGN dengan E_b/N_0 -10 dB sampai 0 dB. Secara teori, E_b/N_0 pada kanal AWGN diberi nilai 0 dB sampai 10 dB. Namun pada sistem OFDM yang telah dibuat ini mengalami pergeseran nilai E_b/N_0 sebanyak 10 nilai ke kiri sehingga menjadi -10 dB sampai 0. Hal ini karena ada parameter sistem yang digeser, yaitu frekuensi carrier pada OFDM. Simulink sendiri tetap mengasumsikan dalam 1 simbol OFDM terdapat 16 bit data tanpa pergeseran.

1.3.1 BER QPSK Secara Teori, Simulasi, dan Implementasi

Reduksi PAPR menggunakan metode *clipping* diimplementasikan dengan menggunakan modulasi QPSK dalam mengirimkan dan menerima informasi. Perhitungan dilakukan menggunakan rumus BER atau Probabilty Error modulasi QPSK menggunakan tabel Q-function. Penggunaan hasil pemodulasian QPSK tanpa metode akan menjadi acuan dalam membandingkan kinerja sistem. Perbandingan BER sistem dengan modulasi QPSK tanpa metode secara teori, simulasi, dan implementasi ditunjukkan pada Gambar 4.11.



Gambar 4. 11 Grafik Perbandingan BER QPSK Teori, Simulasi, dan Implementasi

Berdasarkan grafik pada Gambar 4.10 tersebut, diketahui bahwa nilai BER secara teori lebih bagus daripada BER sistem secara simulasi dan implementasi. Hal ini dikarenakan adanya OFDM pada sistem yang dibuat, sementara BER teori yang dibandingkan yaitu QPSK tanpa menggunakan OFDM.

1.3.2 BER Tanpa dan Dengan Metode *Clipping* Sebelum Melewati HPA

Parameter yang diubah - ubah atau divariasikan adalah level clipping yang digunakan, yaitu mulai 5 sampai 9 sedangkan parameter yang lainnya tetap, yaitu nilai *initialseed* yang digunakan adalah 61, jumlah bit yang dikirim adalah 100.000 bit, jumlah *subcarrier* adalah 8. Gambar 4.12 dan 4.13 menunjukkan grafik BER secara simulasi pada *monte carlo* dan Implementasi pada TMS.

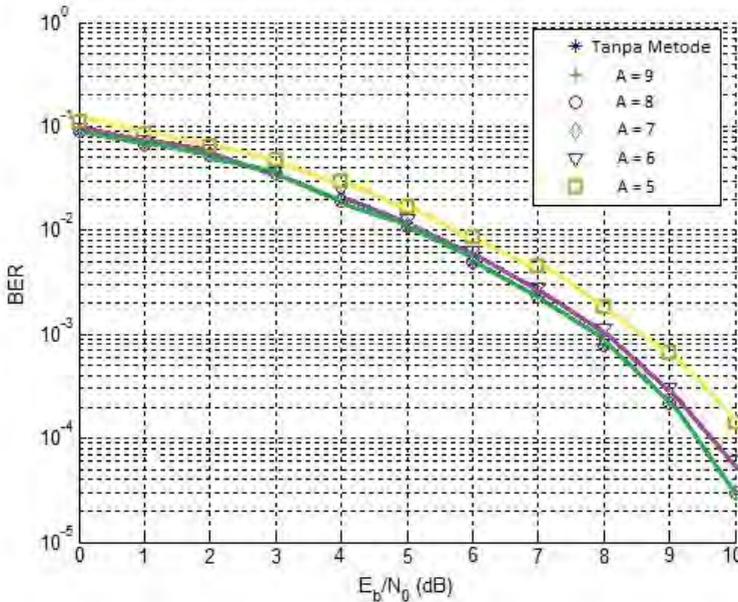
Tabel 4. 13 Tabel BER Dengan Variasi Level *Clipping* Secara Simulasi Sebelum HPA

EbNo	Sebelum Clipping	A = 9	A = 8	A = 7	A = 6	A = 5
0	0.09009	0.09009	0.09009	0.09132	0.09615	0.11013
1	0.06658	0.06658	0.06684	0.06859	0.07386	0.08489
2	0.05269	0.05269	0.05269	0.05426	0.06053	0.06540
3	0.03499	0.03499	0.03499	0.03511	0.03905	0.04822
4	0.01902	0.01909	0.01930	0.02053	0.02504	0.02946
5	0.01076	0.01076	0.01076	0.01110	0.01269	0.01693
6	0.00506	0.00506	0.00506	0.00536	0.00627	0.00844
7	0.00226	0.00228	0.00228	0.00252	0.00282	0.00451
8	0.00077	0.00077	0.00079	0.00089	0.00114	0.00184
9	0.00022	0.00022	0.00022	0.00024	0.00030	0.00066
10	3.00E-05	3.00E-05	3.00E-05	3.00E-05	6.00E-05	1.40E-04

Tabel 4. 14 Tabel BER Dengan Variasi Level *Clipping* Secara Implementasi pada TMS Sebelum HPA

EbNo	Sebelum Clipping	A = 9	A = 8	A = 7	A = 6	A = 5
0	0.1024	0.1020	0.1030	0.1038	0.1047	0.1063
1	0.0761	0.0758	0.0767	0.0782	0.0840	0.0864
2	0.0535	0.0558	0.0542	0.0550	0.0678	0.0715
3	0.0347	0.0343	0.0351	0.0357	0.0441	0.0453
4	0.0208	0.0217	0.0238	0.0218	0.0284	0.0294
5	0.0117	0.0115	0.0122	0.0121	0.0190	0.0192
6	0.0057	0.0065	0.0058	0.0060	0.0087	0.0094

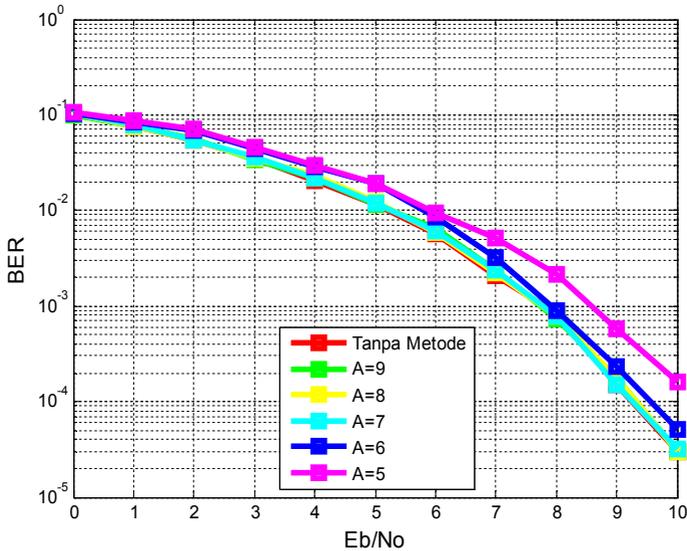
EbNo	Sebelum Clipping	A = 9	A = 8	A = 7	A = 6	A = 5
7	0.0021	0.0024	0.0022	0.0024	0.0032	0.0051
8	0.0008	0.0007	0.0008	0.0008	0.0009	0.0021
9	0.0001	0.0002	0.0002	0.0001	0.0002	0.0006
10	3.01E-05	3.01E-05	3.00E-05	3.20E-05	5.20E-05	1.60E-04



Gambar 4. 12 Grafik BER terhadap Eb/No dengan Variasi Level Clipping secara Simulasi Sebelum HPA

Pada Gambar 4.12, simulation0 = BER tanpa metode (sebelum clipping), simulation1 = level clipping 9, simulation2 = level clipping 8, simulation3 = level clipping 7, simulation4 = level clipping 6, simulation5 = level clipping 5. Berdasarkan grafik pada Gambar 4.11, diketahui bahwa level clipping berbanding terbalik dengan nilai BER. Semakin kecil level clipping, maka BER yang dihasilkan semakin besar. Saat level clipping bernilai 5, BER yang dihasilkan lebih besar daripada

lainnya. Hal ini karena adanya proses pemotongan sinyal pada sistem yang menyebabkan sinyal menjadi cacat sehingga nilai BER bertambah. Semakin banyak amplitude sinyal yang dipotong, maka semakin cacat sinyal tersebut sehingga BER semakin besar pula.



Gambar 4.13 Grafik BER terhadap Eb/No dengan Variasi Level *Clipping* secara Implementasi di TMS Sebelum HPA

Sementara Gambar 4.13 merupakan perbandingan BER terhadap Eb/No secara implementasi TMS. Nilai BER yang diperoleh tidak jauh beda dengan simulasi. Dalam grafik tersebut juga memiliki analisa yang sama dimana saat level *clipping* ($A = 5$) nilai BER yang dihasilkan lebih besar dari lainnya. Sebelum terjadi pemotongan sinyal nilai BER-nya adalah 0.1024. Namun setelah dipotong dengan level *clipping* $A=9$, $A=8$, $A=7$, $A=6$, $A=5$, nilai BER-nya semakin bertambah, yaitu 0.1020, 0.1030, 0.1038, 0.1047, 0.1063. Seperti penjelasan sebelumnya bahwa semakin banyak sinyal yang dipotong, maka BER yang dihasilkan semakin besar. Meskipun demikian, performansi sistem ini masih cukup handal karena kenaikan nilai tidak terlalu signifikan.

1.3.3 BER Tanpa dan Dengan Metode *Clipping* Setelah Melewati HPA

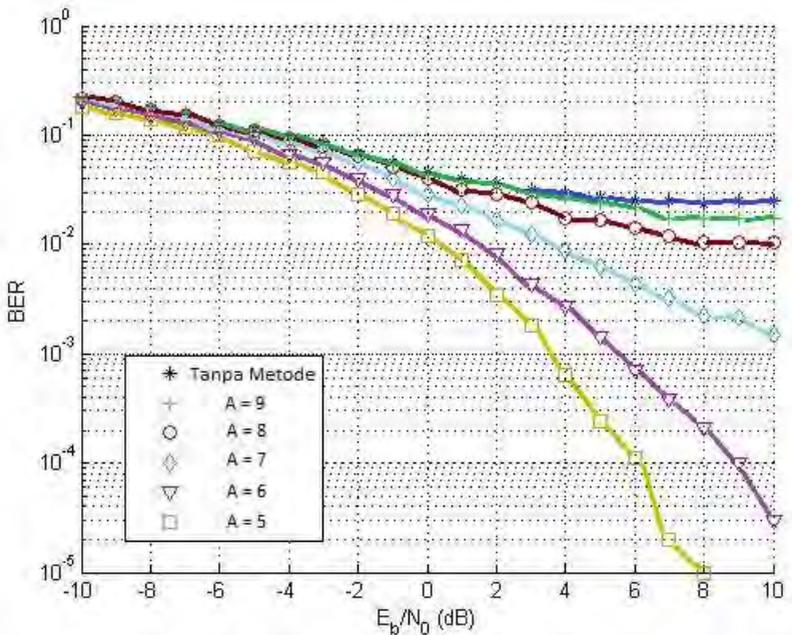
Ampilifier berfungsi untuk menguatkan sinyal sebelum ditransmisikan, sehingga penerima dapat menangkap sinyal dengan baik. Beberapa parameter penting dalam amplifier adalah *dynamic range* (*saturation point*), *bandwidth*, efisiensi, penguatan (*gain*), dan linearitas. Sinyal OFDM dengan PAPR yang tinggi akan menyebabkan amplifier bekerja pada kondisi saturasi, efisiensi rendah, dan terjadi distorsi.

Pada sub-bab ini dianalisa performansi sistem setelah melewati *High Power Amplifier* dengan melihat nilai BER yang dihasilkan. Parameter yang diubah adalah level *clipping* yaitu mulai 5 sampai 9 sedangkan parameter yang lainnya tetap, yaitu nilai *initialseed* yang digunakan adalah 61, jumlah bit yang dikirim adalah 100.000 bit, jumlah *subcarrier* adalah 8.

Tabel 4. 15 Tabel BER Dengan Variasi Level *Clipping* Secara Simulasi Setelah HPA

EbNo	Tanpa Metode	A=9	A=8	A=7	A=6	A=5
-10	0.2151	0.2151	0.2151	0.2146	0.2041	0.1852
-9	0.2008	0.2008	0.2008	0.1980	0.1792	0.1608
-8	0.1742	0.1742	0.1692	0.1664	0.1506	0.1410
-7	0.1595	0.1595	0.1468	0.1439	0.1285	0.1183
-6	0.1255	0.1255	0.1215	0.1183	0.1107	0.0987
-5	0.1151	0.1151	0.1107	0.1072	0.0961	0.0708
-4	0.0969	0.0969	0.0942	0.0896	0.0666	0.0571
-3	0.0864	0.0850	0.0783	0.0701	0.0568	0.0452
-2	0.0689	0.0688	0.0644	0.0565	0.0406	0.0288
-1	0.0561	0.0547	0.0505	0.0408	0.0291	0.0193
0	0.0452	0.0432	0.0403	0.0291	0.0193	0.0119
1	0.0401	0.0389	0.0320	0.0226	0.0136	0.0073
2	0.0361	0.0355	0.0288	0.0165	0.0082	0.0034
3	0.0320	0.0297	0.0245	0.0125	0.0044	0.0018
4	0.0297	0.0272	0.0172	0.0088	0.0027	0.0006
5	0.0272	0.0245	0.0165	0.0063	0.0014	0.0002
6	0.02549	0.02172	0.01426	0.00428	0.00071	0.00011
7	0.02546	0.01744	0.01201	0.00319	0.00038	0.00002
8	0.02305	0.01720	0.01053	0.00222	0.00021	0.00001
9	0.02544	0.01742	0.01034	0.00213	0.00010	0.00000
10	0.02544	0.01720	0.01034	0.00150	0.00003	0.00000

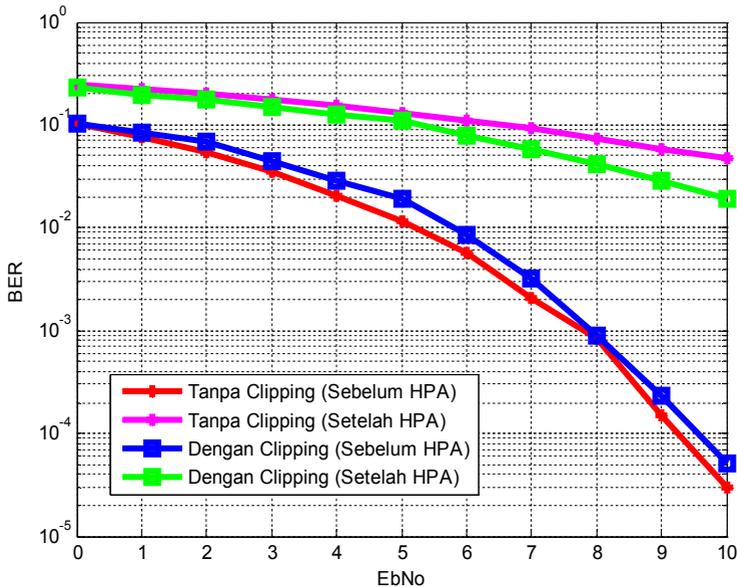
Grafik pada Gambar 4.14 menunjukkan analisa kerja kehandalan sistem setelah melewati HPA pada range E_b/N_0 mulai -10 sampai 10 dB. Dari grafik tersebut diketahui bahwa nilai BER saat menggunakan metode *clipping* lebih bagus daripada tanpa metode *clipping*. Semakin kecil level *clipping* yang digunakan, maka semakin bagus pula nilai BER yang dihasilkan. Misal pada saat $E_b/N_0 = 8$, BER tanpa metode yang diperoleh sebesar 0.02305, BER A = 9 yang diperoleh sebesar 0.0172, BER A = 8 yang diperoleh sebesar 0.01053, BER A = 7 yang diperoleh sebesar 0.00222, BER A = 6 yang diperoleh sebesar 0.00021, BER A = 5 yang diperoleh sebesar 0.00001. Data ini sudah menunjukkan adanya penurunan nilai BER yang dihasilkan dari tanpa metode hingga level *clipping* A = 5.



Gambar 4. 14 Grafik BER terhadap E_b/N_0 dengan Variasi Level *Clipping* secara Simulasi *Monte Carlo* Setelah HPA

1.3.4 Perbandingan BER Sebelum HPA dan Setelah HPA

Pada sub bab ini dianalisa perbandingan BER implementasi antara sebelum HPA dan setelah HPA, baik tanpa *clipping* maupun dengan *clipping*. Level clipping yang digunakan pada perbandingan ini adalah $A = 7$. Perbandingan BER tersebut dapat dilihat pada Gambar 4.15.



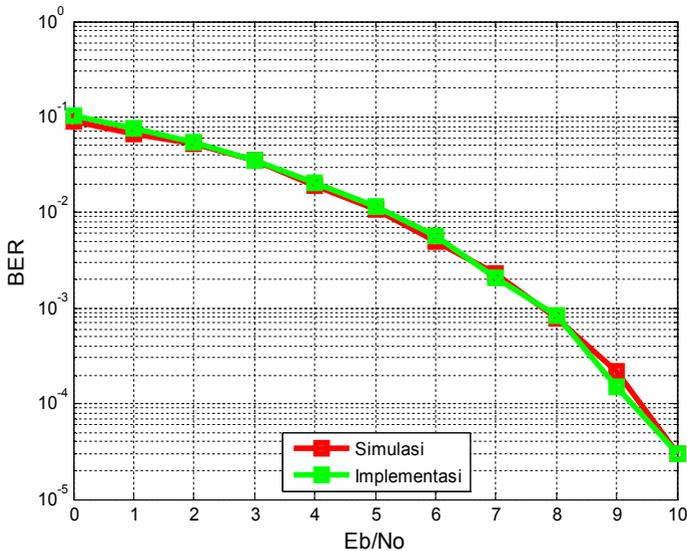
Gambar 4. 15 Perbandingan BER Sebelum HPA dan Setelah HPA

Berdasarkan Gambar 4.15, diketahui bahwa nilai BER sebelum melewati HPA lebih bagus daripada setelah melewati HPA. Saat sebelum melewati HPA, nilai yang dihasilkan masih bersifat linear sementara setelah melewati HPA, nilai yang dihasilkan bersifat nonlinear karena HPA yang ditambahkan pada sistem adalah tipe HPA nonlinear. Meskipun demikian, dalam proses reduksi PAPR ini yang ditekankan adalah pengaruh metode *clipping* terhadap analisa BER setelah melewati HPA.

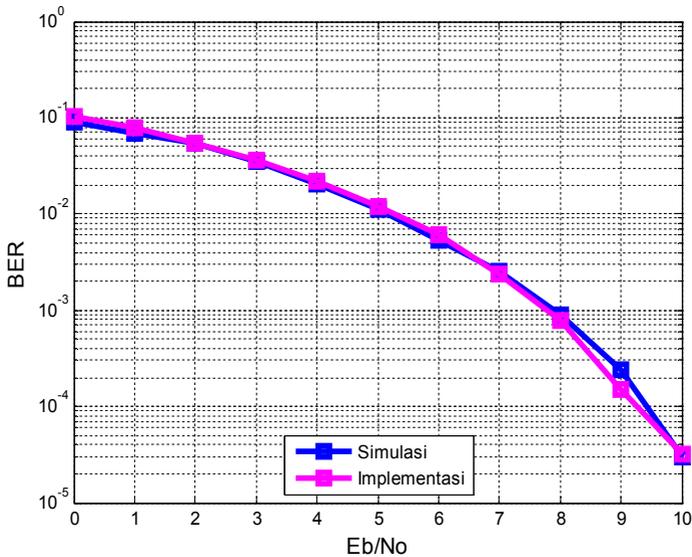
Sebelum melewati HPA, BER yang dihasilkan dengan *clipping* lebih jelek daripada tanpa *clipping*. Oleh karena itu perlu ditambahkan penguat nonlinear (HPA) untuk menganalisa kinerja sistem reduksi PAPR dengan metode *clipping*. Setelah ditambahkan HPA, BER yang dihasilkan dengan metode *clipping* lebih bagus daripada tanpa *clipping*. Jadi, bisa disimpulkan bahwa tujuan reduksi PAPR menggunakan metode *clipping* ini adalah untuk memperkecil nilai BER yang dihasilkan. Dimana paa sub bab sebelumnya telah disebutkan bahwa semakin kecil level *clipping* yang digunakan, maka semakin besar reduksi PAPR yang dihasilkan dan menyebabkan nilai BER semakin bagus.

1.3.5 Perbandingan BER Simulasi dan Implementasi

Untuk lebih jelasnya lagi tentang analisa perbandingan BER antara simulasi dan implementasi, bisa dilihat pada Gambar 4.16 sampai Gambar 4.19, baik sebelum HPA maupun setelah HPA.



Gambar 4. 16 Grafik Perbandingan BER Tanpa Metode antara Simulasi dan Implementasi Sebelum HPA



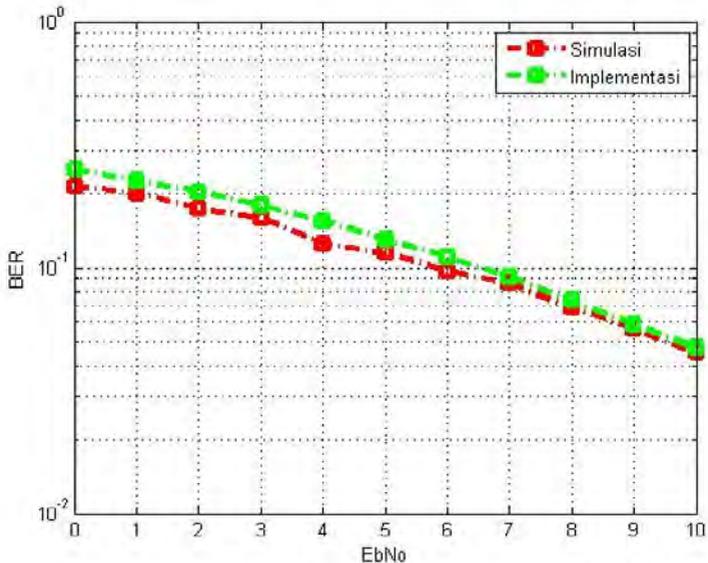
Gambar 4.17 Grafik Perbandingan BER antara Simulasi dan Implementasi pada Level *Clipping* ($A=7$) Sebelum HPA

Grafik pada Gambar 4.16 dan 4.17 merupakan BER sebelum melewati HPA. Berdasarkan grafik pada Gambar 4.16 dan 4.17, dapat dilihat bahwa nilai BER hasil simulasi tidak jauh berbeda dengan implementasi. BER maksimum yang diperoleh saat simulasi adalah 0.0913 sedangkan untuk implementasi, BER maksimumnya adalah 0.1038. Saat nilai $A = 5, 6, 8, 9$ nilai BER yang diperoleh saat simulasi dan implementasi juga hampir sama.

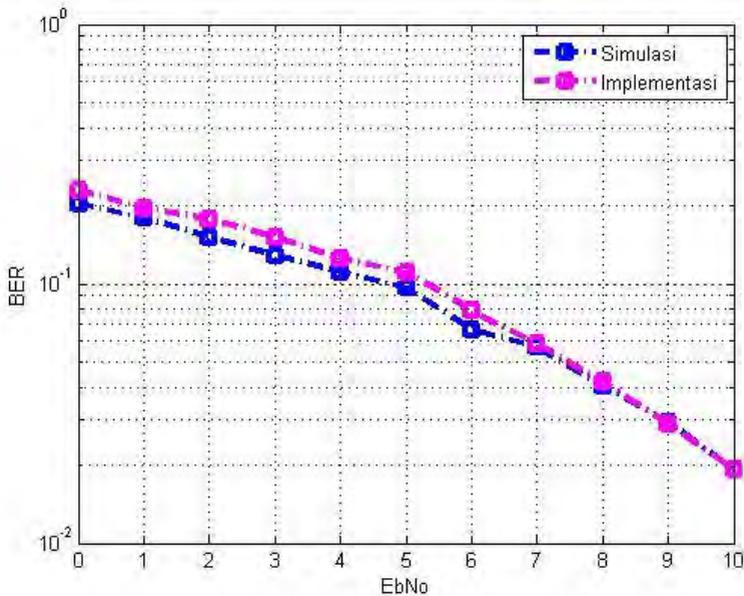
Selanjutnya adalah membandingkan simulasi dan implementasi untuk mengetahui seberapa handal sistem reduksi PAPR menggunakan metode *clipping* yang telah dibuat setelah melewati HPA. Berdasarkan Tabel 4.16, dapat dilihat bahwa nilai BER hasil simulasi sedikit berbeda dengan implementasi saat $E_b/N_0 = 0$ sampai 6, dimana BER implementasi sedikit lebih jelek daripada simulasinya. Namun perbedaan tersebut tidak terlalu signifikan. Hasil perbandingan antara simulasi dan implementasi ditunjukkan dari grafik pada Gambar 4.18 dan 4.19.

Tabel 4. 16 Tabel Perbandingan Simulasi dan Implementasi Setelah HPA

EbNo	Tanpa Metode		A = 6	
	Simulasi	Implementasi	Simulasi	Implementasi
0	0.2150538	0.25	0.2040816	0.2291
1	0.2008032	0.2268	0.1792115	0.1957
2	0.174216	0.2038	0.1506024	0.1776
3	0.1594896	0.1796	0.1285347	0.1511
4	0.1254705	0.1553	0.110742	0.1254
5	0.1150748	0.13	0.0960615	0.1105
6	0.0968992	0.11	0.0665779	0.0788
7	0.0864304	0.0918	0.0568182	0.0587
8	0.0688705	0.0734	0.040568	0.0417
9	0.0561482	0.0587	0.0291375	0.0288
10	0.045208	0.0473	0.0193311	0.0191



Gambar 4. 18 Grafik Perbandingan BER Tanpa Metode antara Simulasi dan Implementasi Setelah HPA



Gambar 4.19 Grafik Perbandingan BER antara Simulasi dan Implementasi pada Level *Clipping* ($A=7$) Setelah HPA

Perbedaan antara simulasi dan implementasi ini disebabkan karena saat pengambilan data secara real-time menggunakan blok RTDX berbeda dengan perhitungan nilai BER menggunakan *Bit Error Rate Analysis Tool* di Matlab. Dari keseluruhan sistem, dapat disimpulkan bahwa nilai BER yang dihasilkan tidak ada yang mencapai 10^{-5} . Kejadian ini disebabkan sistem rentan terhadap noise. Namun perbedaan antara simulasi dan implementasi tidak terlalu besar sehingga masih menunjukkan bahwa performansi sistem yang telah dibuat sudah cukup handal dan bekerja dengan baik.

Pada pengujian implementasi, metode sinkronisasi sinyal juga dibutuhkan untuk mengatasi kesalahan deteksi sistem di penerima sehingga nanti dalam perhitungan nilai BER juga tidak akan terjadi banyak kesalahan.

[Halaman ini sengaja dikosongkan]

BAB V

PENUTUP

1.1 Kesimpulan

Dari hasil simulasi, implementasi, dan analisa yang telah dilakukan pada Tugas akhir ini dapat diambil kesimpulan sebagai berikut:

1. Semakin kecil level *clipping* yang ditentukan, maka reduksi PAPR yang dihasilkan semakin besar. Pada saat $A = 6$, reduksi PAPR bisa mencapai $\geq 30\%$.
2. Semakin banyak *subcarrier* yang digunakan reduksi PAPR yang dihasilkan semakin besar. Reduksi PAPR pada 24 *subcarrier* bisa mencapai $\geq 50\%$.
3. Semakin besar nilai E_b/N_0 , maka semakin kecil nilai BER yang dihasilkan. Saat $E_b/N_0 = 0$, nilai BER $\pm 10^{-1}$ dan saat $E_b/N_0 = 10$, nilai BER mendekati 10^{-5} .
4. Sebelum melewati HPA, BER dengan metode *clipping* lebih besar daripada tanpa metode. Saat $E_b/N_0 = 10$, BER tanpa metode sebesar 3×10^{-5} sedangkan BER dengan metode *clipping* sebesar 14×10^{-5} .
5. Setelah melewati HPA, BER dengan metode *clipping* lebih kecil daripada tanpa metode. Saat $E_b/N_0 = 10$, BER tanpa metode sebesar 2.5×10^{-2} sedangkan BER dengan metode *clipping* mencapai 1×10^{-5} .
6. Dengan reduksi PAPR $\geq 50\%$ dapat menurunkan nilai BER hingga bernilai 1×10^{-5} .

1.2 Saran

Dari tugas akhir yang telah dilakukan, masih terdapat beberapa hal yang dapat disempurnakan. Berikut saran-saran untuk pengembangan sistem reduksi PAPR sinyal OFDM menggunakan metode *clipping* yang menggunakan platform DSP.

1. Simulasi dan implementasi dapat dikembangkan untuk ukuran tipe *clipping* yang berbeda.
2. Simulasi dan implementasi dapat dikembangkan untuk gangguan sinyal yang lebih bervariasi.
3. Simulasi dapat dikembangkan dengan menyertakan parameter – parameter sistem komunikasi digital lainnya.

DAFTAR PUSTAKA

- [1] Cho, Yong Soo, Won Young Yang, Chung-Gu Kang. *MIMO-OFDM Wireless Communicatios with MATLAB*. Singapore. 2010.
- [2] M. M. Rana. *Clipping Based PAPR Reduction Method for LTE OFDMA Systems*. International Journal of Electrical & Computer Sciences IJECS-IJENS Vol:10 No:05, October 2010.
- [3] Aschbacher, Ernst and Markus Rupp. *Modelling And Identification Of A Nonlinear Power – Amplifier With Memory For Nonlinear Digital Adaptive Pre – Distortion*. Institute of Communications and Radio – Frequency Engineering Vienna University of Technology. 2003.
- [4] Chassaing, Rulph., Reay, Donald. *Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK, second edition*:Wiley-Interscience. 2008.
- [5] Thompson Steve C., John G. Proakis, and James R. Zeidler. *The Effectiveness of Signal Clipping for PAPR and Total Degradation Reduction in OFDM Systems*. Center for Wireless Communications University of California, San Diego, 2005.
- [6] Gouda Mamdouh, Khaled Ali Shehata, Mohamed Hussien. *PAPR Reduction Performance for LTE OFDM Systems with Different Techniques*. International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May 2013.
- [7] Dewangan Leman, Mangal Singh, Neelam Dewangan. *A Survey of PAPR Reduction Techniques in LTE-OFDM System*. International Journal of Recent Technology and Engineering (IJRTE). ISSN: 2277-3878, Volume-1, Issue-5, November 2012.
- [8] V. Vijayarangan, DR. (MRS) R. Sukanesh. *An Overview Of Techniques For Reducing Peak To Average Power Ratiio And Its Selection Criteria For Orthogonal Frequency DivisionMultiplexing Radiio Systems*. National Engineering Collage, Kovilpatti. India. 2009.

- [9] W. Aziz, E Ahmed, G. Abbas, S. Saleem, Q. Islam. *PAPR Reduction In OFDM using Clipping and Filtering*. Departement Of Electrical Engineering, Institute Of Space Technology, Islamabad, Pakistan. 2012.
- [10] Triprasetyo Yudhi. *Analisa Dan Perbandingan Teknik Clipping Untuk Reduksi PAPR dan Kode Koreksi Error Untuk Performansi BER Pada OFDM*. Universitas Indonesia. 2011.
- [11] Randy Satria Putra. *Reduksi PAPR Pada OFDM Dengan Menggunakan Transformasi Wavelet Dan Teknik Clipping*. Universitas Indonesia. 2012.
- [12] M. Schetzen, "Nonlinier System Modelling Based on the Wiener Theory". Proceedings of the IEEE, vol. 69, no. 12, pp. 1557-1573, Dec. 1981.
- [13] Sengar, Suverna dan Bhattacharya, P.P. *Performance Improvement in OFDM System by PAPR Reduction. International Journal (SIPIJ) Vol.3, No.2, April 2012*.
- [14] Gavin, Hill. *Peak Power Reduction in Orthogonal Frequency Division Multiplexing Transmitter. Thesis Project. Victoria University. 2011*.
- [15] Mathworks. "Documentation". <http://www.mathworks.com/help/>

LAMPIRAN A

PROPOSAL TUGAS AKHIR

Jurusan Teknik Elektro
Fakultas Teknologi Industri - ITS

TE 091398 TUGAS AKHIR – 4 SKS

Nama Mahasiswa : Rafih Hikmah Puspita
Nomor Pokok : 2212105010
Bidang Studi : Telekomunikasi Multimedia
Tugas Diberikan : Semester Genap Th. 2013/2014
Dosen Pembimbing : 1. Dr. Ir. Suwadi, MT
2. Dr. Ir. Titiek Suryani, MT
Judul Tugas Akhir : **Analisa dan Implementasi Reduksi PAPR Sinyal OFDM Dengan Metode Clipping Pada DSK TMS320C6416T**
(*Analysis and Implementation PAPR Reduction of OFDM Signal Using Clipping Method On DSK TMS320C6416T*)

28 FEB 2014

Uraian Tugas Akhir :

OFDM merupakan sebuah teknik *multiplexing* sinyal melalui skema *multicarrier modulation* dimana sebuah sinyal dengan *bit-rate* yang tinggi dibagi menjadi beberapa kanal dengan *bit-rate* yang lebih rendah dengan pita frekuensi yang sempit dan saling berdekatan dimana setiap kanal menggunakan frekuensi yang berbeda-beda. Sinyal OFDM terdiri dari sejumlah subcarrier yang dimodulasi secara terpisah. Jika masing-masing subcarrier memiliki fasa sinyal yang koheren, sinyal OFDM yang dihasilkan akan memiliki nilai perbandingan PAPR (*Peak to Average Power Ratio*) yang besar.

Solusi untuk mengatasi PAPR yang tinggi ini adalah dengan meminimalkan nilai PAPR pada sinyal OFDM yang akan ditransmisikan. Dengan kata lain, diperlukan skema reduksi PAPR pada bagian transmitter sinyal OFDM. Ada beberapa teknik untuk menurunkan nilai PAPR yaitu, *Partial Transmit Sequence (PTS)*, *Selective Mapping (SLM)*, *Clipping, Interleaving, Sub Block Coding, Peak Windowing*.

Metode yang akan dibahas pada tugas akhir ini adalah metode *clipping*. Pada tugas akhir ini, dibuat implementasi reduksi PAPR dengan metode *clipping* menggunakan board DSK TMS320C6416T. Reduksi PAPR dengan metode *clipping* yang diimplementasikan pada sebuah prosesor DSP diharapkan dapat menjadi suatu ide yang cukup baik untuk membantu menganalisa penurunan PAPR pada sinyal OFDM.

Keyword : *OFDM, PAPR, Clipping, DSK TMS*

Dosen Pembimbing I

Dr. Ir. Suwadi, MT

NIP. 1968 0818 199303 1 002

Dosen Pembimbing II

Dr. Ir. Titiek Suryani, MT

NIP. 1964 1130 198903 2 001



Mengetahui,
Jurusan Teknik Elektro FTI – ITS
Kepala,

Dr. Tri Arif Sarjibno, ST, MT
NIP. 1970 0212 199512 1 001

Menyetujui,
Bidang Studi Telekomunikasi Multimedia
Koordinator,

Dr. Ir. Endroyono, DEA
NIP. 1965 0404 199102 1 001

A. JUDUL

Analisa dan Implementasi Reduksi PAPR Sinyal OFDM Dengan Metode *Clipping* Pada DSK TMS320C6416T

B. RUANG LINGKUP

1. PAPR
2. OFDM
3. DSK (DSP Starter Kit) TMS320C6416T
4. *Clipping*

C. LATAR BELAKANG

Orthogonal Frequency Division Multiplexing (OFDM) merupakan suatu skema modulasi multi carrier yang mampu bekerja pada multipath fading dan yang paling penting bisa memberikan solusi terhadap tuntutan akses layanan kecepatan tinggi secara realtime dengan performansi yang baik. Selain itu, terdapat suatu sistem antena yang bisa dikombinasikan dengan OFDM untuk memenuhi kebutuhan akan kualitas sinyal yang bagus dengan data rate yang tinggi tanpa harus menambah daya dan bandwidth. Sinyal OFDM terdiri dari sejumlah subcarrier yang dimodulasi secara terpisah. Jika masing-masing subcarrier memiliki fasa sinyal yang koheren, sinyal OFDM yang dihasilkan akan memiliki nilai perbandingan PAPR (*Peak to Average Power Ratio*) yang besar.

Nilai PAPR yang besar menghasilkan beberapa kerugian seperti, meningkatnya kompleksitas ADC dan DAC serta mengurangi efisiensi dari RF poweramplifier. Peak mengalami distorsi nonlinear disebabkan oleh ketidaksempurnaan amplifier. Intermodulasi terjadi karena efek distorsi nonlinear. Hal ini dapat diinterpretasikan sebagai *Inter Carrier Interferences* (ICI) dan *out-of-bandradiation* (OOB). Oleh karena itu, PAPR yang tinggi harus direduksi. Solusi untuk mengatasi PAPR yang tinggi ini adalah dengan meminimalkan nilai PAPR pada sinyal OFDM yang akan ditransmisikan. Dengan kata lain, diperlukan skema pereduksi PAPR pada bagian transmitter sinyal OFDM. Ada

beberapa teknik untuk menurunkan nilai PAPR yaitu, *Partial Transmit Sequence (PTS)*, *Selective Mapping (SLM)*, *Clipping*, *Interleaving*, *Sub Block Coding*, *Peak Windowing*. Teknik tersebut memperoleh reduksi PAPR dengan dampak peningkatan daya sinyal transmisi, peningkatan BER, data-rate loss, peningkatan kompleksitas komputasi dan sebagainya.

Metode yang akan dibahas pada tugas akhir ini adalah metode *clipping*. Sebelumnya terdapat beberapa penelitian tentang reduksi PAPR yang menggunakan metode *clipping* dengan berbagai variasi *clipping*. Tetapi penelitian tersebut masih berupa simulasi menggunakan matlab. Oleh karena itu pada tugas akhir ini, dibuat implementasi reduksi PAPR dengan metode *clipping* menggunakan board DSK TMS320C6416T. Reduksi PAPR dengan metode *clipping* yang diimplementasikan pada sebuah prosesor DSP diharapkan dapat menjadi suatu ide yang cukup baik untuk membantu menganalisa penurunan PAPR pada sinyal OFDM. Alasan penggunaan prosesor DSP didasari oleh kemampuan perangkat DSP yang lebih mudah untuk dimodifikasi dan dikalibrasi. Selain itu, prosesor DSP tersebut mudah didapat di pasaran, berkecepatan tinggi serta dapat digunakan untuk mengoperasikan banyak aplikasi atau algoritma.

D. PERUMUSAN MASALAH

Perumusan masalah pada Tugas Akhir ini adalah

1. Metode apa yang digunakan untuk mengurangi PAPR pada sinyal OFDM?
2. Bagaimana membuat implementasi untuk mereduksi PAPR?

E. BATASAN MASALAH

Permasalahan pada Tugas Akhir ini dibatasi pada:

1. Metode reduksi PAPR yang digunakan adalah *Clipping*.
2. Implementasi penurunan PAPR menggunakan Matlab Simulink dan DSK TMS320C6416T.

F. TUJUAN

Tujuan dari pengerjaan Tugas Akhir ini adalah :

1. Menghitung *Bit Error Rate* (BER).
2. Merancang teknik reduksi PAPR dengan metode *clipping* pada DSK TMS320C6416T.
3. Menganalisa teknik reduksi PAPR dengan metode *clipping*.

G. TINJAUAN PUSTAKA DAN DASAR TEORI

1. OFDM

OFDM (*Orthogonal Frequency Division Multiplex*) adalah bentuk khusus dari multi-carrier modulation (MCM) yang membagi sebuah aliran data informasi menjadi beberapa aliran paralel untuk ditransmisikan melalui beberapa sub-carrier. Dalam OFDM, data masukan dialirkan ke beberapa sub-carrier paralel ortogonal yang memiliki laju data yang lebih rendah. Ortogonal merupakan sifat matematika dari dua vektor yang saling tegak lurus. Pada sistem komunikasi, sinyal-sinyal dikatakan ortogonal jika mereka berdiri sendiri tanpa saling mengganggu satu sama lain. Sifat ortogonal dari vektor sinyal ini memungkinkan beberapa sinyal informasi dikirimkan pada kanal yang sama tanpa mengalami interferensi. Dua set sinyal dikatakan ortogonal, jika integral perkalian keduanya dalam satu interval sama dengan nol. Berikut bentuk persamaan dari sinyal ortogonal [1] :

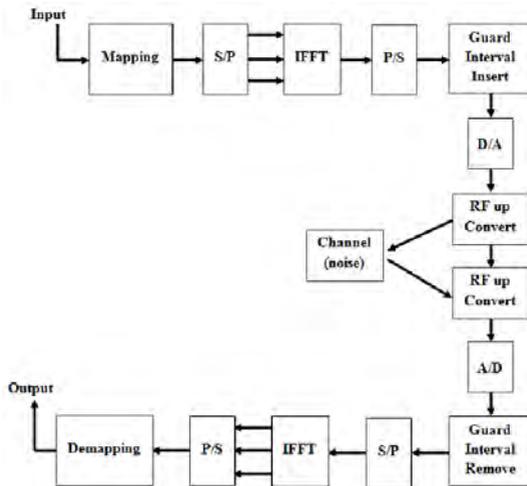
Continuous Time :

$$\int_0^T \cos(2\pi f_0 t) x \cos(2\pi f_0 t) dt = 0 \quad (n \neq m) \quad (1)$$

Discrete Time:

$$\sum_{k=0}^{N-1} \cos\left(\frac{2\pi kn}{N}\right) x \cos\left(\frac{2\pi kn}{N}\right) = 0 \quad (n \neq m) \quad (2)$$

OFDM dapat juga dikatakan variasi dari FDM (*Frekuensi Division Multiplexing*), yang menempatkan sinyal-sinyal sub-carrier sedekat mungkin, sehingga dapat menghemat pemakaian lebar-bandwidth, namun tetap mempertahankan sifat ortogonal antar sinyal. Berikut adalah blok diagram OFDM secara garis besar.



Gambar 1 : Blok diagram OFDM [1]

2. PAPR dari Sinyal OFDM

Dalam sistem OFDM, sinyal OFDM terdiri atas penjumlahan sejumlah subcarrier yang dimodulasi secara independen. Apabila fasa setiap *subcarrier* sama maka akan dihasilkan daya sinyal maksimum sebesar N kali daya rata-ratanya. Perbandingan antara daya sinyal maksimum dengan daya rata-ratanya biasa disebut sebagai *Peak to Average Power Ratio* (PAPR). Pada suatu waktu, penjumlahan ini sangat besar dan pada waktu yang lain sangat kecil, artinya nilai daya maksimum dari sinyal pasti selalu lebih besar dari daya rata-ratanya. Nilai PAPR yang besar merupakan tantangan dalam mengimplementasikan OFDM karena dapat mengurangi efisiensi dan meningkatkan cost dari RF power amplifier. Nilai PAPR yang besar pada OFDM membutuhkan amplifier dengan dynamic range yang lebar untuk mengakomodasi amplitude sinyal. Jika hal ini tidak terpenuhi maka akan terjadi distorsi linear yang menyebabkan subcarrier menjadi tidak lagi orthogonal dan pada akhirnya menurunkan kinerja OFDM.

Sinyal OFDM terdiri dari sejumlah *subcarrier* yang dimodulasi sendiri-sendiri sehingga dapat menghasilkan perbandingan antara daya puncak dan daya rata-rata (*Peak to Average Power Ratio*) yang cukup besar ketika dijumlahkan (superposisi). Ketika N sinyal dengan fasa yang sama dijumlahkan akan menghasilkan daya puncak sebesar N kali dari daya rata-rata. Daya puncak didefinisikan sebagai daya dari gelombang sinus dengan amplitudo sama dengan nilai *envelope* maksimum. Secara matematis, PAPR suatu sinyal $S(t)$ dideskripsikan sebagai berikut [1] :

$$\text{PAPR} = \frac{P_{max}}{P_{av}} = \frac{\max_{0 \leq t \leq T_s} |S(t)|^2}{\frac{1}{T_s} \int_0^{T_s} |S(t)|^2 dt} \quad (3)$$

Dimana

$$|S(t)|^2 = [\sqrt{S(t)S^*(t)}]^2 \quad (4)$$

3. Clipping

Nilai PAPR yang besar akan mengakibatkan sebaran spektrum yang signifikan (*out of band*), *inter symbol interference* (ICI) dan distorsi *in-band* ketika dilewatkan melalui divais nonlinear. Hal tersebut dapat diatasi dengan menggunakan amplifier linear atau dengan mem-*backoff* titik operasi suatu *amplifier* nonlinear yang mengakibatkan menurunnya efisiensi daya.

Cara yang paling mudah untuk mengatasi PAPR yang tinggi adalah dengan memotong (*clipping*) sinyal masukan sebelum ke *amplifier*. Karena probabilitas terjadinya sinyal dengan peak yang tinggi sangat kecil, maka *clipping* merupakan suatu teknik yang efektif untuk menurunkan PAPR. Namun, *clipping* merupakan proses yang nonlinear dan akan mengakibatkan distorsi *in-band* yang cukup signifikan yang akibatnya akan menurunkan BER (*Bit Error Rate*). *Peak power* pada sinyal OFDM akan dibatasi dengan *threshold* melalui persamaan berikut [4] :

$$x(t) = s(t), \text{ untuk } |s(t)| \leq A \quad (5)$$

$$x(t) = Ae^{j\varphi t}, \text{ untuk } |s(t)| > A \quad (6)$$

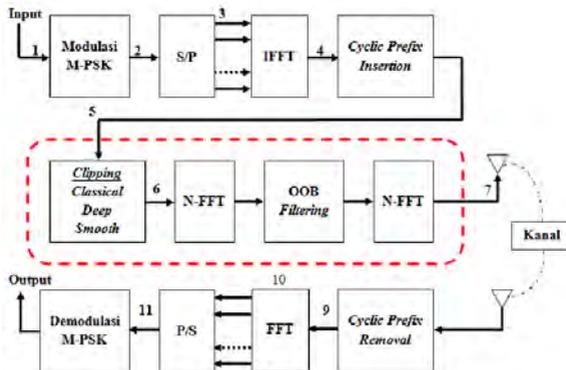
$$\varphi t = \arg(s(t)) \quad (7)$$

Dengan $x(t)$ adalah sinyal OFDM hasil *Clipping*, $s(t)$ merupakan simbol OFDM, A ialah nilai *threshold Clipping*, dan φt ialah fasa *Clipping*. Dari persamaan diatas dihasilkan sinyal yang amplitudonya tidak melebihi A , dengan fasa yang sama dengan aslinya. Nilai A ditentukan oleh parameter *Clipping Ratio* (CR) yang dinyatakan persamaaan berikut [4] :

$$CR = \frac{A}{\sigma} \quad (8)$$

$$\sigma = \sqrt{N} \quad (9)$$

Dengan CR adalah Clipping Ratio, A merupakan nilai threshold Clipping, σ ialah nilai pembandingan, dan N merupakan jumlah subcarrier OFDM. Diagram blok transmitter OFDM dengan disertai clipping dan filtering seperti diperlihatkan gambar berikut:



Gambar 2 : OFDM dengan *Clipping* [4]

4. DSK (DSP Starter Kit) TMS320C6416T

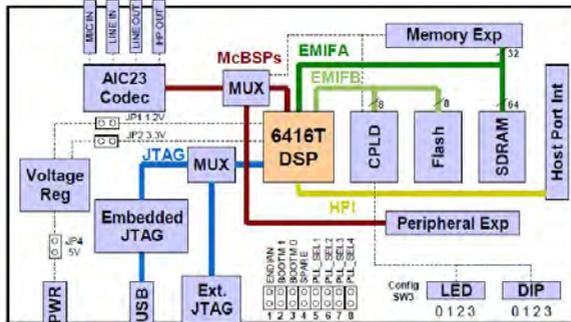
DSK TMS320C6416T merupakan multi-layer board berukuran 8.75 x 4.5 inch (210 x 115 mm) yang disuplai daya eksternal +5 volt. DSK ini terdiri dari sebuah 1 GHz DSK TMS320C6416T fixedpoint digital signal processor dan 16 bit stereo codec TLV 320AIC23 untuk input dan output. Codec AIC23 menyediakan ADC dan DAC dengan clock 12 MHz dan sampling rate 8 – 96 kHz. DSK meliputi

16 MB pada Synchronous Dynamic Random (SDRAM) dan 256 kB pada flash memory.

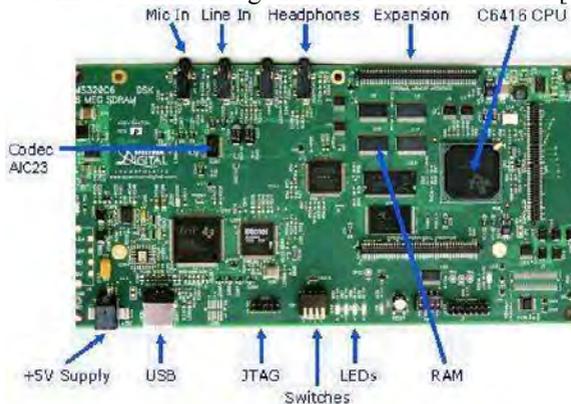
Terdapat empat konektor pada DSK untuk input dan output :

- MIC IN untuk input dari microphone.
- LINE IN untuk input dari function generator,
- LINE OUT untuk output , dan
- HEADPHONE untuk output pada headphone.

Dip switch dalam DSK dapat difungsikan sesuai program dan menyediakan fungsi kontrol. Terdapat pengatur tegangan yang menyediakan 1,26 V untuk processor dan 3,3 V untuk sistem memori dan kelengkapan lain.



Gambar 3 : Blok Diagram DSK TMS320C6416T [5]



Gambar 4 : Board DSK TMS320C6416T [5]

TMS320C6416 pada DSK adalah floating point processor yang didasari pada arsitektur VLIW (Very-Long-Instruction-Word). Memori internal meliputi 2 level penyimpanan daya, yang terdiri dari 32 KB pada level penyimpanan program (L1P) dan level 1 penyimpanan data (L1D), dan 1024 kB pada level 2 pembagian memori antara program dan data. Selain itu, juga mempunyai sebuah interface untuk memerintah synchronous dan memori (SDRAM dan SBSRAM) dan asynchronous memori (SPRAM dan EPROM). Memiliki 8 fungsi atau *execution units* dalam 6 ALU dan 2 multiplier units, 32 - bit address bus ke 4 GB (gigabytes) address, dan 2 set 32 - bit general - purpose registers.

H. METODOLOGI

Metode yang digunakan dalam pengerjaan Tugas Akhir ini terdiri dari beberapa tahapan pengerjaan, yaitu :

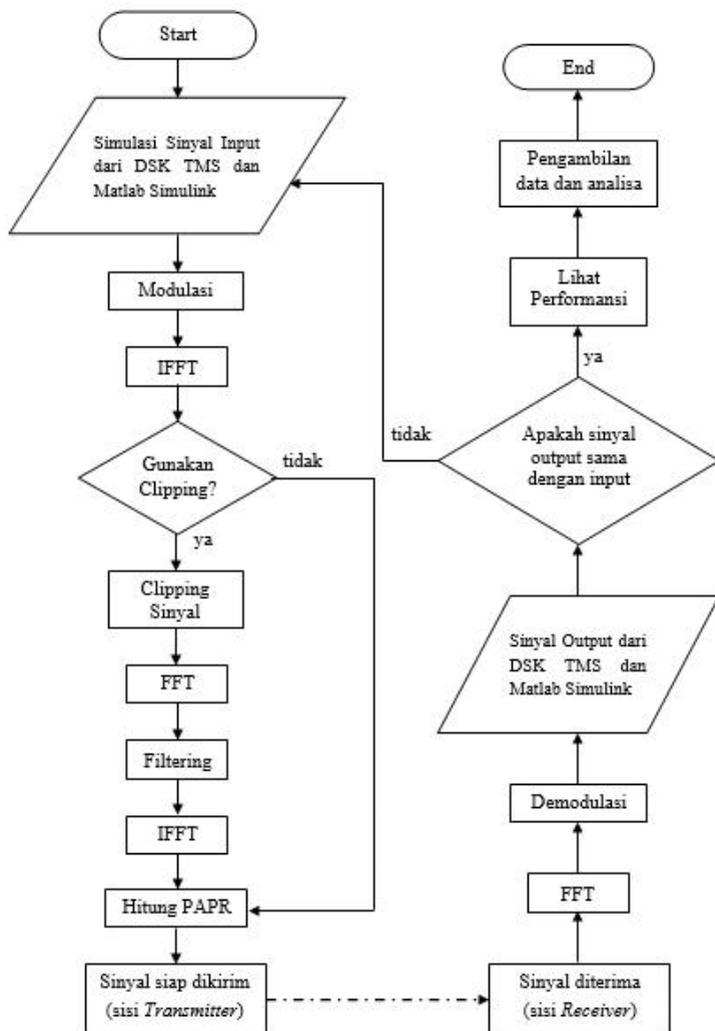
1. Studi literature
Mencari refensi yang berkaitan dengan pemancar dan penerima sinyal OFDM dengan metode *clipping* dan implementasinya pada perangkat DSK TMS320C64616T. Literatur yang digunakan meliputi buku referensi, jurnal dan dokumen internet.
2. Perancangan sistem pemancar dan penerima sinyal OFDM
Pemodelan sistem yang akan disimulasikan, meliputi perancangan sistem pemancar dan penerima sinyal OFDM dengan menggunakan perangkat lunak Matlab Simulink.
3. Integrasi Simulink pada DSK TMS320C6416T
Proses integrasi model sistem yang telah dibuat dari Matlab Simulink ke DSK TMS320C6416T melalui *interface* perangkat lunak Code Composer Studio.
4. Perancangan simulasi
Merupakan cara atau metode yang akan digunakan dalam pengujian sistem yang telah dibuat.
5. Analisa hasil keluaran sistem
Meliputi analisa hasil dari keluaran sistem pada Simulink dibandingkan dengan hasil keluaran pada Osiloskop.

6. Kesimpulan

Penarikan kesimpulan dari kinerja sistem yang selanjutnya dapat ditarik sebuah kesimpulan

Flowchart pengerjaan tugas akhir ini bisa dilihat pada gambar 5. Pada sisi pengirim, tahap pertama adalah membangkitkan sinyal input (sinyal informasi) dari DSK TMS dan matlab Simulink. Bit informasi dipetakan menggunakan modulasi QPSK menjadi symbol kompleks. Sekuen symbol dijadikan parallel, kemudian dilakukan transformasi IFFT yang menghasilkan simbol OFDM. Setelah itu terdapat proses penyisipan *cyclic prefix* . Kemudian dilakukan seleksi kondisi apakah menggunakan proses *clipping* pada sinyal OFDM atau tidak. Apabila tidak, langsung dihitung nilai PAPR – nya. Apabila iya, dilakukan proses *clipping* sinyal untuk reduksi PAPR dan dilakukan *filtering* untuk menurunkan OOB. Setelah itu sinyal pada domain waktu ditransmisikan melalui kanal.

Pada sisi penerima, sinyal OFDM dihilangkan *cyclic prefix*-nya dan ditransformasi menggunakan FFT. Sinyal OFDM pada domain frekuensi yang telah dijadikan serial didemodulasi. Kemudian dilihat sinyal output dari DSK TMS dan matlab Simulink. Apabila sinyal output sama dengan sinyal input, maka dilakukan analisa terhadap parameter yang mempengaruhi penurunan nilai PAPR.



Gambar 5. Flowchart Pengerjaan Tugas Akhir

I. JADWAL KEGIATAN

Kegiatan	Minggu Ke-															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Studi literatur																
Pemodelan sinyal dari DSK TMS																
Pemodelan sinyal di Matlab Simulink																
Analisa performansi PAPR dengan metode <i>clipping</i>																
Analisa performansi PAPR dengan metode lainnya																
Pengambilan data																
Analisis Data																
Penyusunan laporan																

J. DAFTAR PUSTAKA

- [1] M. M. Rana. *Clipping Based PAPR Reduction Method for LTE OFDMA Systems*. International Journal of Electrical & Computer Sciences IJECS-IJENS Vol:10 No:05, October 2010.
- [2] Thompson Steve C., John G. Proakis, and James R. Zeidler. *The Effectiveness of Signal Clipping for PAPR and Total Degradation Reduction in OFDM Systems*. Center for Wireless Communications University of California, San Diego, 2005.
- [3] Gouda Mamdouh, Khaled Ali Shehata, Mohamed Hussien. *PAPR Reduction Performance for LTE OFDM Systems with Different Techniques*. International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May 2013.
- [4] Dewangan Leman, Mangal Singh, Neelam Dewangan. *A Survey of PAPR Reduction Techniques in LTE-OFDM System*. International Journal of Recent Technology and Engineering (IJRTE). ISSN: 2277-3878, Volume-1, Issue-5, November 2012.
- [5] Chassaing, Rulph., Reay, Donald. (2008). *Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK, second edition*:Wiley-Interscience.

[Halaman ini sengaja dikosongkan]

LAMPIRAN B

MONITORING KEGIATAN TUGAS AKHIR



ITS
Institut
Teknologi
Sepuluh Nopember

FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
Kampus ITS Gedung B & C Sukolilo Surabaya - 60111
Telp. (031) 594 7302, 5994251 s/d 54 Pes. 1206
Fax. (031) 5931237
e-mail : ed@www.its.ac.id
Website: <http://www.ee.its.ac.id>

MONITORING KEGIATAN TUGAS AKHIR

Nama Mahasiswa : Ratih Hikmah Puspita Nrp 2212105010
Judul Tugas Akhir : Analisa dan Implementasi Reduksi PAPR Cegah OFDM Dengan Metode Clipping Pada PSK TMS 3200A4LT

Bulan Proposal Disahkan :
Dosen Pembimbing 1 : Dr. Ir. Suwadi, MT NIP 19680818 199303 1002
Dosen Pembimbing 2 : Dr. Ir. Triek Suryani, MT NIP 19641130 198908 2001

No	Tanggal	Urutan Kegiatan	Tanda Tangan		Mahasiswa	Keterangan
			Pembimbing (1)	Pembimbing (2)		
1	26/2014 2	Memperajari sistem OFDM				
2	7/2014 3	Membuat Modulasi Baseband BPSK				
3	14/2014 3	Membuat Modulasi Baseband QPSK				
4	21/2014 3	Membuat Modulasi Baseband QPSK OFDM				
5	28/2014 3	Membuat Demodulasi Baseband BPSK OFDM				
6	4/2014 4	Memperajari teknik reduksi PAPR clipping				
7	17/2014 4	Membuat sistem reduksi PAPR dg clipping				
8	23/2014 4	Uji coba simulasi sistem yg telah dibuat (link BER)				
9	7/2014 5	Membandingkan sistem antara cekatan dan setelah metode				
10	13/2014 5	Integrasi PSK TMS 3200A4LT dg Matlab Simulink dan CS				
11	19/2014 5	Membandingkan simulasi dan implementasi (hasil BER)				
12	23/2014 5	Konsultasi hasil reduksi PAPR yg telah diperoleh				
13	30/2014 5	Revisi Buku TA Bab I, II dan III				
14	5/2014 6	Revisi Buku TA Bab I - V serta Lampiran				
15						

[Halaman ini sengaja dikosongkan]

LAMPIRAN C

LISTING PROGRAM

#Listing Program RTDX

```
clear all;
clc;
cc = ccstdsp;
l=1;
rtdx_ichan = cc.rtdx;
rtdx_ichan.set('timeout', 100);
rtdx_ichan.configure(64000,2);
rtdx_ichan.open('Aira_tx','r');
rtdx_ichan.open('Aira_rx','r');
rtdx_ichan.enable;
cc.run;
enable(rtdx_ichan,'Aira_tx');
enable(rtdx_ichan,'Aira_rx');
jum_data_tx=msgcount(rtdx_ichan,'Aira_tx')
jum_data_rx=msgcount(rtdx_ichan,'Aira_rx')
ukuran=201;%jumlah bit= ukuran x buffer
a=ones(ukuran,500);b=ones(ukuran,500);
for i=1:1:ukuran
    a(i,:)=readmsg(cc.rtdx,'Aira_tx','int16');
    b(i,:)=readmsg(cc.rtdx,'Aira_rx','int16');
end
aa=a(:,1:1:ukuran);
bb=b(:,1:1:ukuran);
err=xor(aa,bb);
jum=size(aa);
ber=sum(sum(err))/(jum(1,1)*jum(1,2))
```

[Halaman ini sengaja dikosongkan]

DAFTAR RIWAYAT HIDUP



Ratih Hikmah Puspita dilahirkan di Banyuwangi, Jawa Timur, pada tanggal 27 Maret 1990, merupakan putri pertama dari pasangan Bapak Sunanto dan Ibu Ririn Tri Winarni. Pada tahun 2001, penulis menamatkan sekolahnya di SDN Tapanrejo V Muncar dan melanjutkan ke SMPN 1 Cluring lalu melanjutkan ke SMAN 1 Genteng. Setelah tamat SMA pada tahun 2007, penulis melanjutkan pendidikan di Jurusan Matematika Universitas Airlangga, kemudian pada tahun 2008 pindah di D3 PENS-ITS program studi Teknik Telekomunikasi dan lulus pada tahun 2011. Setelah itu pada tahun 2012 penulis melanjutkan ke jenjang sarjana di Jurusan Teknik Elektro ITS program studi Telekomunikasi Multimedia. Pada tahun 2014 penulis mengambil tugas akhir di Laboratorium Komunikasi Multimedia B304. Alamat email penulis aira.luph.pink@gmail.com.

[Halaman ini sengaja dikosongkan]

LAMPIRAN A

PROPOSAL TUGAS AKHIR

Jurusan Teknik Elektro
Fakultas Teknologi Industri - ITS

TE 091398 TUGAS AKHIR – 4 SKS

Nama Mahasiswa : Rafih Hikmah Puspila
Nomor Pokok : 2212105010
Bidang Studi : Telekomunikasi Multimedia
Tugas Diberikan : Semester Genap Th. 2013/2014
Dosen Pembimbing : 1. Dr. Ir. Suwadi, MT
2. Dr. Ir. Titiek Suryani, MT
Judul Tugas Akhir : **Analisa dan Implementasi Reduksi PAPR Sinyal OFDM Dengan Metode Clipping Pada DSK TMS320C6416T**
(*Analysis and Implementation PAPR Reduction of OFDM Signal Using Clipping Method On DSK TMS320C6416T*)

28 FEB 2014

Uraian Tugas Akhir :

OFDM merupakan sebuah teknik *multiplexing* sinyal melalui skema *multicarrier modulation* dimana sebuah sinyal dengan *bit-rate* yang tinggi dibagi menjadi beberapa kanal dengan *bit-rate* yang lebih rendah dengan pita frekuensi yang sempit dan saling berdekatan dimana setiap kanal menggunakan frekuensi yang berbeda-beda. Sinyal OFDM terdiri dari sejumlah subcarrier yang dimodulasi secara terpisah. Jika masing-masing subcarrier memiliki fasa sinyal yang koheren, sinyal OFDM yang dihasilkan akan memiliki nilai perbandingan PAPR (*Peak to Average Power Ratio*) yang besar.

Solusi untuk mengatasi PAPR yang tinggi ini adalah dengan meminimalkan nilai PAPR pada sinyal OFDM yang akan ditransmisikan. Dengan kata lain, diperlukan skema reduksi PAPR pada bagian transmitter sinyal OFDM. Ada beberapa teknik untuk menurunkan nilai PAPR yaitu, *Partial Transmit Sequence (PTS)*, *Selective Mapping (SLM)*, *Clipping, Interleaving, Sub Block Coding, Peak Windowing*.

Metode yang akan dibahas pada tugas akhir ini adalah metode *clipping*. Pada tugas akhir ini, dibuat implementasi reduksi PAPR dengan metode *clipping* menggunakan board DSK TMS320C6416T. Reduksi PAPR dengan metode *clipping* yang diimplementasikan pada sebuah prosesor DSP diharapkan dapat menjadi suatu ide yang cukup baik untuk membantu menganalisa penurunan PAPR pada sinyal OFDM.

Keyword : *OFDM, PAPR, Clipping, DSK TMS*

Dosen Pembimbing I

Dr. Ir. Suwadi, MT

NIP. 1968 0818 199303 1 002

Dosen Pembimbing II

Dr. Ir. Titiek Suryani, MT

NIP. 1964 1130 198903 2 001



Mengetahui,
Jurusan Teknik Elektro FTI – ITS
Kepala,

Dr. Tri Arif Sarjibno, ST, MT

NIP. 1970 0212 199512 1 001

Menyetujui,
Bidang Studi Telekomunikasi Multimedia
Koordinator,

Dr. Ir. Endroyono, DEA

NIP. 1965 0404 199102 1 001

A. JUDUL

Analisa dan Implementasi Reduksi PAPR Sinyal OFDM Dengan Metode *Clipping* Pada DSK TMS320C6416T

B. RUANG LINGKUP

1. PAPR
2. OFDM
3. DSK (DSP Starter Kit) TMS320C6416T
4. *Clipping*

C. LATAR BELAKANG

Orthogonal Frequency Division Multiplexing (OFDM) merupakan suatu skema modulasi multi carrier yang mampu bekerja pada multipath fading dan yang paling penting bisa memberikan solusi terhadap tuntutan akses layanan kecepatan tinggi secara realtime dengan performansi yang baik. Selain itu, terdapat suatu sistem antena yang bisa dikombinasikan dengan OFDM untuk memenuhi kebutuhan akan kualitas sinyal yang bagus dengan data rate yang tinggi tanpa harus menambah daya dan bandwidth. Sinyal OFDM terdiri dari sejumlah subcarrier yang dimodulasi secara terpisah. Jika masing-masing subcarrier memiliki fasa sinyal yang koheren, sinyal OFDM yang dihasilkan akan memiliki nilai perbandingan PAPR (*Peak to Average Power Ratio*) yang besar.

Nilai PAPR yang besar menghasilkan beberapa kerugian seperti, meningkatnya kompleksitas ADC dan DAC serta mengurangi efisiensi dari RF poweramplifier. Peak mengalami distorsi nonlinear disebabkan oleh ketidaksempurnaan amplifier. Intermodulasi terjadi karena efek distorsi nonlinear. Hal ini dapat diinterpretasikan sebagai *Inter Carrier Interferences* (ICI) dan *out-of-bandradiation* (OOB). Oleh karena itu, PAPR yang tinggi harus direduksi. Solusi untuk mengatasi PAPR yang tinggi ini adalah dengan meminimalkan nilai PAPR pada sinyal OFDM yang akan ditransmisikan. Dengan kata lain, diperlukan skema pereduksi PAPR pada bagian transmitter sinyal OFDM. Ada

beberapa teknik untuk menurunkan nilai PAPR yaitu, *Partial Transmit Sequence (PTS)*, *Selective Mapping (SLM)*, *Clipping*, *Interleaving*, *Sub Block Coding*, *Peak Windowing*. Teknik tersebut memperoleh reduksi PAPR dengan dampak peningkatan daya sinyal transmisi, peningkatan BER, data-rate loss, peningkatan kompleksitas komputasi dan sebagainya.

Metode yang akan dibahas pada tugas akhir ini adalah metode *clipping*. Sebelumnya terdapat beberapa penelitian tentang reduksi PAPR yang menggunakan metode *clipping* dengan berbagai variasi *clipping*. Tetapi penelitian tersebut masih berupa simulasi menggunakan matlab. Oleh karena itu pada tugas akhir ini, dibuat implementasi reduksi PAPR dengan metode *clipping* menggunakan board DSK TMS320C6416T. Reduksi PAPR dengan metode *clipping* yang diimplementasikan pada sebuah prosesor DSP diharapkan dapat menjadi suatu ide yang cukup baik untuk membantu menganalisa penurunan PAPR pada sinyal OFDM. Alasan penggunaan prosesor DSP didasari oleh kemampuan perangkat DSP yang lebih mudah untuk dimodifikasi dan dikalibrasi. Selain itu, prosesor DSP tersebut mudah didapat di pasaran, berkecepatan tinggi serta dapat digunakan untuk mengoperasikan banyak aplikasi atau algoritma.

D. PERUMUSAN MASALAH

Perumusan masalah pada Tugas Akhir ini adalah

1. Metode apa yang digunakan untuk mengurangi PAPR pada sinyal OFDM?
2. Bagaimana membuat implementasi untuk mereduksi PAPR?

E. BATASAN MASALAH

Permasalahan pada Tugas Akhir ini dibatasi pada:

1. Metode reduksi PAPR yang digunakan adalah *Clipping*.
2. Implementasi penurunan PAPR menggunakan Matlab Simulink dan DSK TMS320C6416T.

F. TUJUAN

Tujuan dari pengerjaan Tugas Akhir ini adalah :

1. Menghitung *Bit Error Rate* (BER).
2. Merancang teknik reduksi PAPR dengan metode *clipping* pada DSK TMS320C6416T.
3. Menganalisa teknik reduksi PAPR dengan metode *clipping*.

G. TINJAUAN PUSTAKA DAN DASAR TEORI

1. OFDM

OFDM (*Orthogonal Frequency Division Multiplex*) adalah bentuk khusus dari multi-carrier modulation (MCM) yang membagi sebuah aliran data informasi menjadi beberapa aliran paralel untuk ditransmisikan melalui beberapa sub-carrier. Dalam OFDM, data masukan dialirkan ke beberapa sub-carrier paralel ortogonal yang memiliki laju data yang lebih rendah. Ortogonal merupakan sifat matematika dari dua vektor yang saling tegak lurus. Pada sistem komunikasi, sinyal-sinyal dikatakan ortogonal jika mereka berdiri sendiri tanpa saling mengganggu satu sama lain. Sifat ortogonal dari vektor sinyal ini memungkinkan beberapa sinyal informasi dikirimkan pada kanal yang sama tanpa mengalami interferensi. Dua set sinyal dikatakan ortogonal, jika integral perkalian keduanya dalam satu interval sama dengan nol. Berikut bentuk persamaan dari sinyal ortogonal [1] :

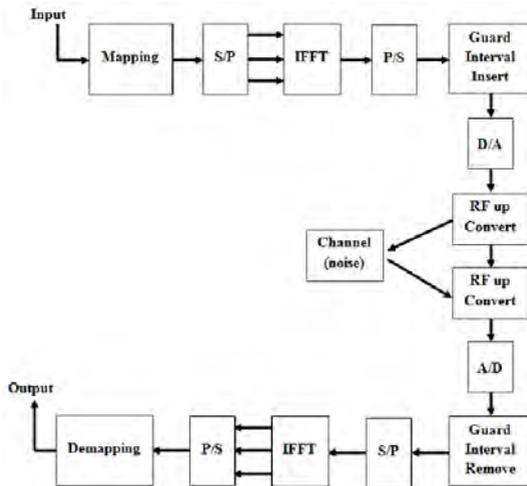
Continuous Time :

$$\int_0^T \cos(2\pi f_0 t) x \cos(2\pi f_0 t) dt = 0 \quad (n \neq m) \quad (1)$$

Discrete Time:

$$\sum_{k=0}^{N-1} \cos\left(\frac{2\pi kn}{N}\right) x \cos\left(\frac{2\pi kn}{N}\right) = 0 \quad (n \neq m) \quad (2)$$

OFDM dapat juga dikatakan variasi dari FDM (*Frekuensi Division Multiplexing*), yang menempatkan sinyal-sinyal sub-carrier sedekat mungkin, sehingga dapat menghemat pemakaian lebar-bandwidth, namun tetap mempertahankan sifat ortogonal antar sinyal. Berikut adalah blok diagram OFDM secara garis besar.



Gambar 1 : Blok diagram OFDM [1]

2. PAPR dari Sinyal OFDM

Dalam sistem OFDM, sinyal OFDM terdiri atas penjumlahan sejumlah subcarrier yang dimodulasi secara independen. Apabila fasa setiap *subcarrier* sama maka akan dihasilkan daya sinyal maksimum sebesar N kali daya rata-ratanya. Perbandingan antara daya sinyal maksimum dengan daya rata-ratanya biasa disebut sebagai *Peak to Average Power Ratio* (PAPR). Pada suatu waktu, penjumlahan ini sangat besar dan pada waktu yang lain sangat kecil, artinya nilai daya maksimum dari sinyal pasti selalu lebih besar dari daya rata-ratanya. Nilai PAPR yang besar merupakan tantangan dalam mengimplementasikan OFDM karena dapat mengurangi efisiensi dan meningkatkan cost dari RF power amplifier. Nilai PAPR yang besar pada OFDM membutuhkan amplifier dengan dynamic range yang lebar untuk mengakomodasi amplitude sinyal. Jika hal ini tidak terpenuhi maka akan terjadi distorsi linear yang menyebabkan subcarrier menjadi tidak lagi orthogonal dan pada akhirnya menurunkan kinerja OFDM.

Sinyal OFDM terdiri dari sejumlah *subcarrier* yang dimodulasi sendiri-sendiri sehingga dapat menghasilkan perbandingan antara daya puncak dan daya rata-rata (*Peak to Average Power Ratio*) yang cukup besar ketika dijumlahkan (superposisi). Ketika N sinyal dengan fasa yang sama dijumlahkan akan menghasilkan daya puncak sebesar N kali dari daya rata-rata. Daya puncak didefinisikan sebagai daya dari gelombang sinus dengan amplitudo sama dengan nilai *envelope* maksimum. Secara matematis, PAPR suatu sinyal $S(t)$ dideskripsikan sebagai berikut [1] :

$$PAPR = \frac{P_{max}}{P_{av}} = \frac{\max_{0 \leq t \leq T_s} |S(t)|^2}{\frac{1}{T_s} \int_0^{T_s} |S(t)|^2 dt} \quad (3)$$

Dimana

$$|S(t)|^2 = [\sqrt{S(t)S^*(t)}]^2 \quad (4)$$

3. Clipping

Nilai PAPR yang besar akan mengakibatkan sebaran spektrum yang signifikan (*out of band*), *inter symbol interference* (ICI) dan distorsi *in-band* ketika dilewatkan melalui divais nonlinear. Hal tersebut dapat diatasi dengan menggunakan amplifier linear atau dengan mem-*backoff* titik operasi suatu *amplifier* nonlinear yang mengakibatkan menurunnya efisiensi daya.

Cara yang paling mudah untuk mengatasi PAPR yang tinggi adalah dengan memotong (*clipping*) sinyal masukan sebelum ke *amplifier*. Karena probabilitas terjadinya sinyal dengan peak yang tinggi sangat kecil, maka *clipping* merupakan suatu teknik yang efektif untuk menurunkan PAPR. Namun, *clipping* merupakan proses yang nonlinear dan akan mengakibatkan distorsi *in-band* yang cukup signifikan yang akibatnya akan menurunkan BER (*Bit Error Rate*). *Peak power* pada sinyal OFDM akan dibatasi dengan *threshold* melalui persamaan berikut [4] :

$$x(t) = s(t), \text{ untuk } |s(t)| \leq A \quad (5)$$

$$x(t) = Ae^{j\varphi t}, \text{ untuk } |s(t)| > A \quad (6)$$

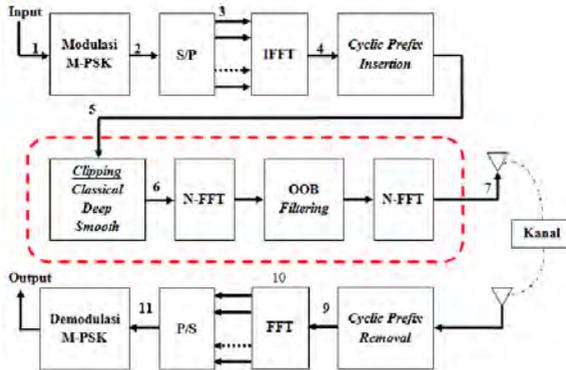
$$\varphi t = \arg(s(t)) \quad (7)$$

Dengan $x(t)$ adalah sinyal OFDM hasil *Clipping*, $s(t)$ merupakan simbol OFDM, A ialah nilai *threshold Clipping*, dan φt ialah fasa *Clipping*. Dari persamaan diatas dihasilkan sinyal yang amplitudonya tidak melebihi A , dengan fasa yang sama dengan aslinya. Nilai A ditentukan oleh parameter *Clipping Ratio* (CR) yang dinyatakan persamaan berikut [4] :

$$CR = \frac{A}{\sigma} \quad (8)$$

$$\sigma = \sqrt{N} \quad (9)$$

Dengan CR adalah Clipping Ratio, A merupakan nilai threshold Clipping, σ ialah nilai pembanding, dan N merupakan jumlah subcarrier OFDM. Diagram blok transmitter OFDM dengan disertai clipping dan filtering seperti diperlihatkan gambar berikut:



Gambar 2 : OFDM dengan *Clipping* [4]

4. DSK (DSP Starter Kit) TMS320C6416T

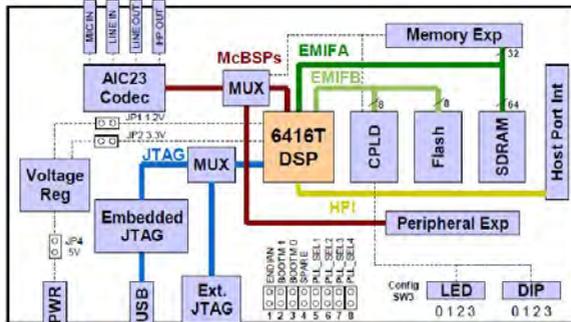
DSK TMS320C6416T merupakan multi-layer board berukuran 8.75 x 4.5 inch (210 x 115 mm) yang disuplai daya eksternal +5 volt. DSK ini terdiri dari sebuah 1 GHz DSK TMS320C6416T fixedpoint digital signal processor dan 16 bit stereo codec TLV 320AIC23 untuk input dan output. Codec AIC23 menyediakan ADC dan DAC dengan clock 12 MHz dan sampling rate 8 – 96 kHz. DSK meliputi

16 MB pada Synchronous Dynamic Random (SDRAM) dan 256 kB pada flash memory.

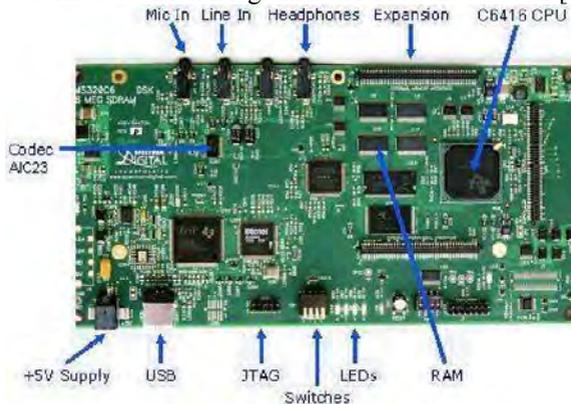
Terdapat empat konektor pada DSK untuk input dan output :

- MIC IN untuk input dari microphone.
- LINE IN untuk input dari function generator,
- LINE OUT untuk output , dan
- HEADPHONE untuk output pada headphone.

Dip switch dalam DSK dapat difungsikan sesuai program dan menyediakan fungsi kontrol. Terdapat pengatur tegangan yang menyediakan 1,26 V untuk processor dan 3,3 V untuk sistem memori dan kelengkapan lain.



Gambar 3 : Blok Diagram DSK TMS320C6416T [5]



Gambar 4 : Board DSK TMS320C6416T [5]

TMS320C6416 pada DSK adalah floating point processor yang didasari pada arsitektur VLIW (Very-Long-Instruction-Word). Memori internal meliputi 2 level penyimpanan daya, yang terdiri dari 32 KB pada level penyimpanan program (L1P) dan level 1 penyimpanan data (L1D), dan 1024 kB pada level 2 pembagian memori antara program dan data. Selain itu, juga mempunyai sebuah interface untuk memerintah synchronous dan memori (SDRAM dan SBSRAM) dan asynchronous memori (SPRAM dan EPROM). Memiliki 8 fungsi atau *execution units* dalam 6 ALU dan 2 multiplier units, 32 - bit address bus ke 4 GB (gigabytes) address, dan 2 set 32 - bit general - purpose registers.

H. METODOLOGI

Metode yang digunakan dalam pengerjaan Tugas Akhir ini terdiri dari beberapa tahapan pengerjaan, yaitu :

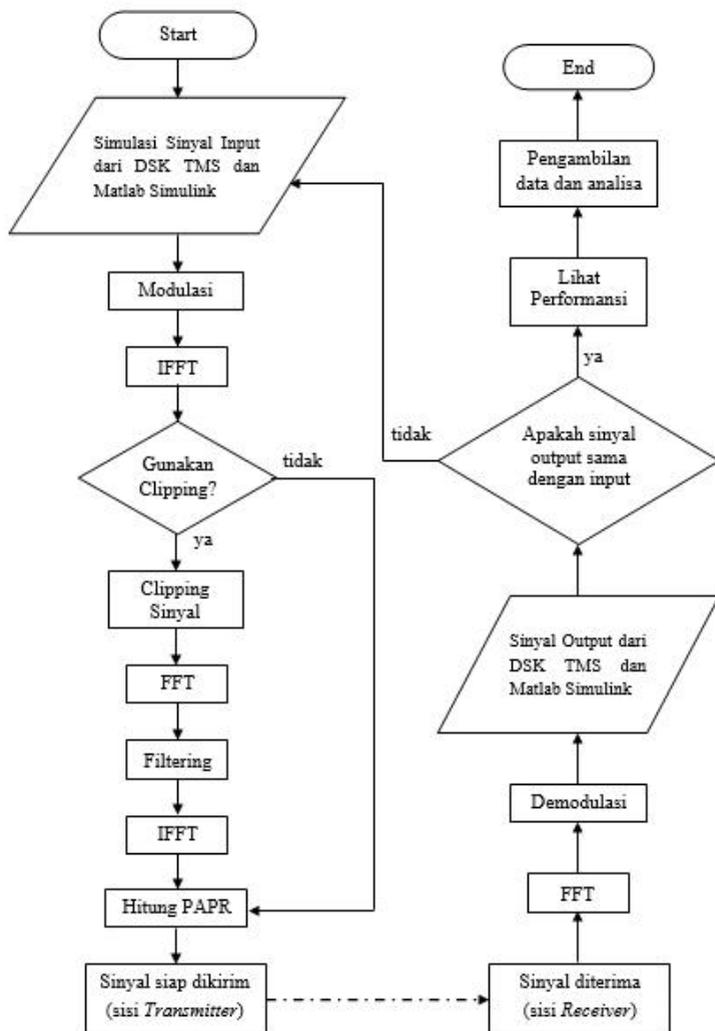
1. Studi literature
Mencari refensi yang berkaitan dengan pemancar dan penerima sinyal OFDM dengan metode *clipping* dan implementasinya pada perangkat DSK TMS320C64616T. Literatur yang digunakan meliputi buku referensi, jurnal dan dokumen internet.
2. Perancangan sistem pemancar dan penerima sinyal OFDM
Pemodelan sistem yang akan disimulasikan, meliputi perancangan sistem pemancar dan penerima sinyal OFDM dengan menggunakan perangkat lunak Matlab Simulink.
3. Integrasi Simulink pada DSK TMS320C6416T
Proses integrasi model sistem yang telah dibuat dari Matlab Simulink ke DSK TMS320C6416T melalui *interface* perangkat lunak Code Composer Studio.
4. Perancangan simulasi
Merupakan cara atau metode yang akan digunakan dalam pengujian sistem yang telah dibuat.
5. Analisa hasil keluaran sistem
Meliputi analisa hasil dari keluaran sistem pada Simulink dibandingkan dengan hasil keluaran pada Osiloskop.

6. Kesimpulan

Penarikan kesimpulan dari kinerja sistem yang selanjutnya dapat ditarik sebuah kesimpulan

Flowchart pengerjaan tugas akhir ini bisa dilihat pada gambar 5. Pada sisi pengirim, tahap pertama adalah membangkitkan sinyal input (sinyal informasi) dari DSK TMS dan matlab Simulink. Bit informasi dipetakan menggunakan modulasi QPSK menjadi symbol kompleks. Sekuen symbol dijadikan parallel, kemudian dilakukan transformasi IFFT yang menghasilkan simbol OFDM. Setelah itu terdapat proses penyisipan *cyclic prefix* . Kemudian dilakukan seleksi kondisi apakah menggunakan proses *clipping* pada sinyal OFDM atau tidak. Apabila tidak, langsung dihitung nilai PAPR – nya. Apabila iya, dilakukan proses *clipping* sinyal untuk reduksi PAPR dan dilakukan *filtering* untuk menurunkan OOB. Setelah itu sinyal pada domain waktu ditransmisikan melalui kanal.

Pada sisi penerima, sinyal OFDM dihilangkan *cyclic prefix*-nya dan ditransformasi menggunakan FFT. Sinyal OFDM pada domain frekuensi yang telah dijadikan serial didemodulasi. Kemudian dilihat sinyal output dari DSK TMS dan matlab Simulink. Apabila sinyal output sama dengan sinyal input, maka dilakukan analisa terhadap parameter yang mempengaruhi penurunan nilai PAPR.



Gambar 5. Flowchart Pengerjaan Tugas Akhir

I. JADWAL KEGIATAN

Kegiatan	Minggu Ke-															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Studi literatur																
Pemodelan sinyal dari DSK TMS																
Pemodelan sinyal di Matlab Simulink																
Analisa performansi PAPR dengan metode <i>clipping</i>																
Analisa performansi PAPR dengan metode lainnya																
Pengambilan data																
Analisis Data																
Penyusunan laporan																

J. DAFTAR PUSTAKA

- [1] M. M. Rana. *Clipping Based PAPR Reduction Method for LTE OFDMA Systems*. International Journal of Electrical & Computer Sciences IJECS-IJENS Vol:10 No:05, October 2010.
- [2] Thompson Steve C., John G. Proakis, and James R. Zeidler. *The Effectiveness of Signal Clipping for PAPR and Total Degradation Reduction in OFDM Systems*. Center for Wireless Communications University of California, San Diego, 2005.
- [3] Gouda Mamdouh, Khaled Ali Shehata, Mohamed Hussien. *PAPR Reduction Performance for LTE OFDM Systems with Different Techniques*. International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May 2013.
- [4] Dewangan Leman, Mangal Singh, Neelam Dewangan. *A Survey of PAPR Reduction Techniques in LTE-OFDM System*. International Journal of Recent Technology and Engineering (IJRTE). ISSN: 2277-3878, Volume-1, Issue-5, November 2012.
- [5] Chassaing, Rulph., Reay, Donald. (2008). *Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK, second edition*:Wiley-Interscience.

[Halaman ini sengaja dikosongkan]

LAMPIRAN B

MONITORING KEGIATAN TUGAS AKHIR



ITS
Institut
Teknologi
Sepuluh Nopember

FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
Kampus ITS Gedung B & C Sukolilo Surabaya - 60111
Telp. (031) 594 7302, 5994251 s/d 54 Pes. 1206
Fax. (031) 5931237
e-mail : ed@www.its.ac.id
Website: <http://www.ee.its.ac.id>

MONITORING KEGIATAN TUGAS AKHIR

Nama Mahasiswa : Ratih Hikmah Puspita Nrp 2212105010
Judul Tugas Akhir : Analisa dan Implementasi Reduksi PAPR Cegah OFDM Dengan Metode Clipping Pada PSK TMS 3200A4LT

Bulan Proposal Disahkan :
Dosen Pembimbing 1 : Dr. Ir. Suwadi, MT NIP 19680818 199303 1002
Dosen Pembimbing 2 : Dr. Ir. Triek Suryani, MT NIP 19641130 198908 2001

No	Tanggal	Urutan Kegiatan	Tanda Tangan		Mahasiswa	Keterangan
			Pembimbing (1)	Pembimbing (2)		
1	26/2014 2	Memperajari sistem OFDM				
2	7/3/2014	Membuat Modulasi Baseband BPSK				
3	14/3/2014	Membuat Modulasi Baseband QPSK				
4	21/3/2014	Membuat Modulasi Baseband QPSK OFDM				
5	28/3/2014	Membuat Demodulasi Baseband BPSK OFDM				
6	4/4/2014	Memperajari teknik reduksi PAPR clipping				
7	17/4/2014	Membuat sistem reduksi PAPR dg clipping				
8	23/4/2014	Uji coba simulasi sistem yg telah dibuat (link BER)				
9	7/5/2014	Membandingkan sistem antara cekatan dan setelah metode				
10	13/5/2014	Integrasi PSK TMS 3200A4LT dg Matlab Simulink dan CS				
11	19/5/2014	Membandingkan simulasi dan implementasi (hasil BER)				
12	23/5/2014	Penyusunan hasil reduksi PAPR yg telah diperoleh				
13	30/5/2014	Revisi Buku TA Bab I, II dan III				
14	5/6/2014	Revisi Buku TA Bab I - V serta Lampiran				
15						

[Halaman ini sengaja dikosongkan]

LAMPIRAN C

LISTING PROGRAM

#Listing Program RTDX

```
clear all;
clc;
cc = ccstdsp;
l=1;
rtdx_ichan = cc.rtdx;
rtdx_ichan.set('timeout', 100);
rtdx_ichan.configure(64000,2);
rtdx_ichan.open('Aira_tx','r');
rtdx_ichan.open('Aira_rx','r');
rtdx_ichan.enable;
cc.run;
enable(rtdx_ichan,'Aira_tx');
enable(rtdx_ichan,'Aira_rx');
jum_data_tx=msgcount(rtdx_ichan,'Aira_tx')
jum_data_rx=msgcount(rtdx_ichan,'Aira_rx')
ukuran=201;%jumlah bit= ukuran x buffer
a=ones(ukuran,500);b=ones(ukuran,500);
for i=1:1:ukuran
    a(i,:)=readmsg(cc.rtdx,'Aira_tx','int16');
    b(i,:)=readmsg(cc.rtdx,'Aira_rx','int16');
end
aa=a(:,1:1:ukuran);
bb=b(:,1:1:ukuran);
err=xor(aa,bb);
jum=size(aa);
ber=sum(sum(err))/(jum(1,1)*jum(1,2))
```

[Halaman ini sengaja dikosongkan]

DAFTAR RIWAYAT HIDUP



Ratih Hikmah Puspita dilahirkan di Banyuwangi, Jawa Timur, pada tanggal 27 Maret 1990, merupakan putri pertama dari pasangan Bapak Sunanto dan Ibu Ririn Tri Winarni. Pada tahun 2001, penulis menamatkan sekolahnya di SDN Tapanrejo V Muncar dan melanjutkan ke SMPN 1 Cluring lalu melanjutkan ke SMAN 1 Genteng. Setelah tamat SMA pada tahun 2007, penulis melanjutkan pendidikan di Jurusan Matematika Universitas Airlangga, kemudian pada tahun 2008 pindah di D3 PENS-ITS program studi Teknik Telekomunikasi dan lulus pada tahun 2011. Setelah itu pada tahun 2012 penulis melanjutkan ke jenjang sarjana di Jurusan Teknik Elektro ITS program studi Telekomunikasi Multimedia. Pada tahun 2014 penulis mengambil tugas akhir di Laboratorium Komunikasi Multimedia B304. Alamat email penulis aira.luph.pink@gmail.com.

[Halaman ini sengaja dikosongkan]