



UNIVERSITAS INDONESIA

**RANCANG BANGUN *BASEBAND* WIMAX PADA DSK
TMS320C6713 DENGAN MENGGUNAKAN SIMULINK**

SKRIPSI

**MAULANA ISHAK
04 05 03 054 Y**

**FAKULTAS TEKNIK
PROGRAM STUDI ELEKTRO
DEPOK
JULI 2009**



UNIVERSITAS INDONESIA

**RANCANG BANGUN *BASEBAND* WIMAX PADA DSK
TMS320C6713 DENGAN MENGGUNAKAN SIMULINK**

SKRIPSI

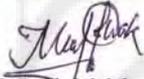
Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**MAULANA ISHAK
04 05 03 054 Y**

**FAKULTAS TEKNIK
PROGRAM STUDI ELEKTRO
DEPOK
JULI 2009**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Maulana Ishak
NPM : 04 05 03 054 Y
Tanda Tangan : 
Tanggal : 6 Juli 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Maulana Ishak
NPM : 040503054Y
Program Studi : Teknik Elektro
Judul Skripsi : **RANCANG BANGUN *BASEBAND* WIMAX
PADA DSK TMS320C6713 DENGAN
SIMULINK**

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Elektro, Fakultas Teknik Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr.Ir. Arman Djohan Diponegoro, M.Eng



Penguji : Dr.Ir. Purnomo Sidhi Priambodo, M.Sc



Penguji : Abdul Muis, ST, M.Eng, Phd



Ditetapkan di : Depok

Tanggal : 6 Juli 2009

KATA PENGANTAR

Puji syukur saya panjatkan kepada Allah Subhanahu Wa Ta'ala, karena atas berkat dan rahmat-Nya saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa selama masa penelitian sampai pada penyusunan skripsi ini banyak pihak yang telah membantu. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Dr. Ir. Arman Djohan, M.Eng. selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
2. Orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; serta
3. Seluruh pihak yang telah membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Allah Subhanahu Wa Ta'ala berkenan membalas kebaikan semua pihak yang telah membantu dalam penelitian dan penyusunan skripsi ini. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Depok, Juli 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Maulana Ishak
NPM : 04 05 03 054 Y
Departemen : Elektro
Fakultas : Teknik
Jenis karya : Skripsi

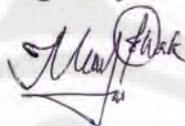
demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

Rancang Bangun Baseband WiMAX pada DSK TMS320C6713 dengan Menggunakan Simulink.

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 6 Juli 2009
Yang menyatakan



(Maulana Ishak)

ABSTRAK

Nama : Maulana Ishak
Program Studi : Teknik Elektro
Judul :

Rancang Bangun *Baseband* WiMAX pada DSK TMS320C6713 dengan Menggunakan Simulink.

Kebutuhan akan layanan *broadband* yang semakin meningkat dan didorong dengan aplikasi yang beragam membuat penyedia layanan akses mencari alternatif teknologi yang dapat mampu memenuhi layanan yang optimal. WiMAX (*worldwide interoperability for microwave access*, IEEE.802.16) adalah teknologi *Broadband Wireless Access* (BWA) yang dikembangkan secara khusus dari teknologi *Orthogonal Frequency Division Multiplexing* (OFDM) dan menggunakan teknik pengkodean kanal yang berlapis pada lapisan fisik untuk masing-masing aplikasi *fixed/mobile* sehingga dapat mendukung lingkungan yang *non-line-of sight* dengan data kecepatan transmisi yang tinggi dan mobilitas yang tinggi, sehingga WiMAX merupakan solusi dari permasalahan tersebut.

Pada skripsi ini, dilakukan rancang bangun *baseband* WiMAX dengan menggunakan DSK (*Digital Signal Processing Starter Kit*) TMS320C6713 berbasis Simulink. Dari hasil rancang bangun didapatkan bahwa rancang bangun *baseband* WiMAX dapat dibangun dengan menggunakan DSP (*Digital Signal Processing*) Processor.

Kata kunci:

WiMAX, *Orthogonal*, Pemancar, Penerima, OFMD, QPSK, *Reed-Solomon*, *Viterbi*, kode konvolusi.

ABSTRACT

Name : Maulana Ishak
Study Program : Electrical Engineering
Title :

Design and Construction of WiMAX Baseband using the DSK TMS320C6713 Operated by Simulink.

Recently broadband service demand are increase, this case encourage by various service make service provider to create an alternative access technology to provide optimum service. WiMAX is Broadband Wireless Access (BWA) technology which was developed from technology specifically othogonal frequency division multiplexing (OFDM) and using a chappel coding on the physical layer for each fixed/mobile application, so that it can support non-line-of-sight environment with speed data transmission and mobility rate high, so that WiMAX is the solution to these problems.

In this research, baseband WiMAX is built using DSK (Digital Signal Processing Starter Kit) TMS 320C6713 based on simulink. The result shows that baseband WiMAX can be built by using DSP (Digital Signal Processing) Processor.

Keywords:

WiMAX, Orthogonal, transmitter, Receiver, OFMD, QPSK, Reed-Solomon, Viterbi, convolution code.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	xiii
DAFTAR SINGKATAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Penulisan	3
BAB 2 WIMAX	4
2.1 WiMAX	4
2.2 Teknologi WiMAX.....	4
2.2.1 <i>Orthogonal Frequency Division Multiplexing (OFDM)</i>	5
2.2.1.1 <i>Sinyal Orthogonal</i>	7
2.2.1.2 <i>Prinsip Kerja OFDM</i>	9
2.2.1.3 <i>Penggunaan Cyclic Prefix</i>	11
2.2.2 <i>Teknik Pengkodean Kanal</i>	12
2.2.2.1 <i>Reed-Solomon</i>	12
2.2.2.2 <i>Convulasi Encoder</i>	13
2.2.3 <i>Quardrature Phase Shift Keying (QPSK)</i>	14
2.2.4 <i>Kanal Propagasi</i>	14
2.2.4.1 <i>Kanal AWGN</i>	15
BAB 3 RANCANG BANGUN BASEBAND WIMAX	16
3.1 Simulink	16
3.2 Penjelasan Umum Pemodelan Simulasi <i>Baseband</i> WiMAX...	17
3.3 Blok Diagram <i>Baseband</i> WiMAX Pada Simulink	18
3.3.1 Blok Diagram <i>Baseband Transmitter</i> WiMAX	19
3.3.1.1 Blok Data	19
3.3.1.2 Blok Pengacakan (<i>Randomizer</i>).....	21
3.3.1.3 Blok <i>Encoder</i>	23
3.3.1.4 Blok <i>Convolutional Encoder</i>	25
3.3.1.5 Blok <i>Interleaver</i>	27
3.1.3.6 Blok <i>Modulator</i>	27
3.1.3.7 Blok <i>Normalize</i>	27

3.1.3.8 blok OFDM <i>Symbol</i>	28
3.1.3.9 Blok Kanal AWGN	31
3.3.2 <i>Receiver</i>	32
3.4 <i>Digital Signal Processor Starter Kit TMS320C6713</i>	33
BAB 4 ANALISA HASIL SIMULASI	38
4.1 Pengambilan dan Analisis Data pada Simulink	38
4.2 Pengambilan dan Analisis Data pada DSK TMS320C6713	47
4.2.1 <i>Real Time Data Exchange (RTDX)</i>	47
BAB 5 KESIMPULAN	49
DAFTAR ACUAN	50

DAFTAR GAMBAR

Gambar 2.1 Spektrum <i>Frequency Division Multiplexing</i> (FDM).....	6
Gambar 2.2 Spektrum <i>Orthogonal Frequency Division Multiplexing</i> (OFDM)	6
Gambar 2.3 Modulator OFDM dengan N <i>subcarrier</i>	10
Gambar 2.4 Penerima OFDM dengan N <i>subcarrier</i>	10
Gambar 2.5. Spektral sinyal multicarrier (a) non OFDM, (b) OFDM	11
Gambar 2.6 Penambahan <i>Cyclic Prefix</i> pada simbol	11
Gambar 2.7 pemodelan kanal AWGN	15
Gambar 3.1 cara-cara menjalankan Simulink	16
Gambar 3.2 Diagram Blok Sistem Transmisi	17
Gambar 3.3 Rancang bangun <i>baseband</i> WiMAX	18
Gambar 3.4 Blok diagram <i>baseband transmitter</i> WiMAX	19
Gambar 3.5 Blok diagram <i>baseband receiver</i> WiMAX	19
Gambar 3.6 Parameter blok sumber data	20
Gambar 3.7 Parameter blok <i>integer to bit converter</i>	21
Gambar 3.8 Parameter blok <i>PN sequence Generator</i>	22
Gambar 3.9 Parameter Blok <i>bit to integer converter</i>	23
Gambar 3.10 Proses pemendekan dan <i>puncture</i>	24
Gambar 3.11 Parameter blok <i>Reed-Solomon encoder</i>	25
Gambar 3.12 <i>Convolutional Encoder</i> dengan rate = $\frac{1}{2}$	26
Gambar 3.13 Parameter Blok <i>convolutional encoder</i>	26
Gambar 3.14 Blok diagram pada blok OFDM <i>modulation</i>	28
Gambar 3.15 parameter blok <i>Multiport Selector</i>	29
Gambar 3.16 parameter blok <i>Matrix Concatenation</i>	30
Gambar 3.17 Parameter blok <i>Add Cyclic Prefix</i>	31
Gambar 3.18 algoritma penanaman model ke dalam TMS320C6713 sebagai <i>embedded target</i>	34
Gambar 3.19 Rangkaian <i>baseband</i> WiMAX untuk DSK	34
Gambar 3.20 Rangkaian <i>baseband</i> WiMAX DSK RTDX	36
Gambar 3.21 Proses pembuatan program C.....	37
Gambar 4.1 Pencantuman blok <i>to workspace</i> dan <i>scope</i>	38

Gambar 4.2 Data kirim.....	39
Gambar 4.3 Gelombang sinyal OFDM <i>transmitter</i>	43
Gambar 4.4 spektrum frekuensi pada <i>transmitter</i>	44
Gambar 4.5 Gelombang sinyal OFDM <i>receiver</i>	44
Gambar 4.6 spektrum frekuensi pada <i>receiver</i>	45
Gambar 4.7 data terima.....	45
Gambar 4.8 perbandingan data	46
Gambar 4.9 <i>Output</i> rangkaian <i>baseband</i> WiMAX dengan RTDX.....	47
Gambar 4.10 <i>Output</i> rangkaian <i>baseband</i> WiMAX	48

DAFTAR TABEL

Tabel 2.1	Pemetaan bit untuk <i>mapping</i> QPSK.....	14
Tabel 3.1	Tabel XOR.....	22
Tabel 3.2	Urutan Pembagian Sinyal pada <i>Multiport Selector</i>	8
Tabel 4.1	variabel “ <i>data input</i> ”	40
Tabel 4.2	Variabel ” <i>randomize</i> ” dan ” <i>Reed-Solomon</i> ”	41
Tabel 4.3	“ <i>interleaver</i> ” dan variabel “ <i>mapping</i> ”	42
Tabel 4.4	“ <i>interleaver</i> ” dan variabel “ <i>mapping</i> ”	43

DAFTAR LAMPIRAN

Lampiran 1 Tabel Nilai Lengkap Variabel Data <i>Input</i> , <i>Randomize</i> , dan <i>Reed-Solomon</i>	52
Lampiran 2 Tabel Nilai Lengkap Variabel <i>Interleaver</i> , <i>mapping</i> dan modulasi OFDM	59



DAFTAR SINGKATAN

AWGN	<i>Additive White Gaussian Noise</i>
BWA	<i>Broadband Wireless Access</i>
CCS	<i>Code Composer Studio</i>
CS	<i>Single Carrier</i>
DSK	<i>Digital Signal Processing Starter Kit</i>
FDM	<i>Frequency Division Multiplexing</i>
FFT	<i>Fast Fourier Transform</i>
GF	<i>Galois Field</i>
IFFT	<i>Inverse Fast Fourier Transform</i>
ISI	<i>intersymbol interference</i>
LoS	<i>line-of-sight</i>
MAC	<i>Medium Access Control</i>
MAN	<i>Metropolitan Area Network</i>
NLoS	<i>non line-of-sight</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
PN	<i>pseudorandom noise</i>
QAM	<i>Quadratul amplitude Modulation</i>
QPSK	<i>Quardrature Phase Shift Keying</i>
RTDX	<i>Real Time Data Exchange</i>
WiMAX	<i>worldwide interoperability for microwave access</i>

BAB 1 PENDAHULUAN

1.1 LATAR BELAKANG

WiMAX (*worldwide interoperability for microwave access*, IEEE.802.16) merupakan teknologi *Broadband Wireless Access* (BWA) yang dikembangkan secara khusus dari teknologi *Orthogonal Frequency Division Multiplexing* (OFDM) untuk mencapai cakupan daerah yang luas dengan kecepatan tinggi dan tambahan *multiple access* yang mungkin bisa diaplikasikan untuk sistem komunikasi *cellular* masa depan. Pada teknologi OFDM ini, spektrum sinyal dibagi-bagi ke dalam berbagai *sub-carrier* yang saling berpotongan tanpa mengalami interferensi satu dengan lain, selain itu teknologi ini juga membuat informasi yang dipancarkan lebih tahan terhadap *frequency selective fading*.

Pada standar IEEE 802.16e ditawarkan beberapa mekanisme teknik pengkodean kanal. Pilihan yang digunakan pada skripsi ini adalah teknik pengkodean kanal dua tahap yaitu *Reed-Solomon* untuk pengkodean bagian luarnya (*outer code*) dan kode konvolusi pada bagian pengkodean dalam (*inner code*). Alasan penggunaan pengkodean berlapis agar sistem handal terhadap *random error* dan *burst error*, sedangkan Teknik modulasi yang ditawarkan pada IEEE 802.16e adalah adaptif (*adaptive modulation*) untuk BPSK, QPSK, 16-QAM dan 64-QAM, yang digunakan pada skripsi ini adalah modulasi QPSK.

Teknologi *baseband* WiMAX yang diterapkan pada *Digital Signal Processing Starter Kit* (DSK) TMS320C6713 ini dibuat selain untuk mempelajari teknologi *baseband* WiMAX tetapi juga untuk membuat sebuah rancang bangun dari *baseband* WiMAX. Pembuatan dari *baseband* WiMAX ini dibuat dengan DSK TMS320C6713 dengan terlebih dahulu membuat model simulasi *baseband* WiMAX pada Simulink. Setelah model pada Simulink selesai dibuat, maka model telah siap untuk diterapkan pada DSK. Penerapan model pada DSK tidak membutuhkan pemrograman ulang karena Simulink akan membuat kode bahasa pemrograman C untuk dapat diterapkan dalam DSK. Kode bahasa pemrograman ini akan dijalankan dalam perangkat lunak *Code Composer Studio* (CCS) dan dari perangkat lunak ini kemudian program dijalankan pada DSK.

1.2 PERUMUSAN MASALAH

Masalah yang dibahas adalah pembuatan rancang bangun *baseband* WiMAX pada DSK TMS320C6713. Pembuatan dimulai dengan membuat model pada Simulink untuk kemudian diterapkan pada DSK dengan menggunakan perangkat lunak tambahan *Code Composer Studio* (CCS). Beberapa penyesuaian harus dilakukan agar blok yang dibuat dapat diimplementasikan pada DSK.

1.3 TUJUAN PENELITIAN

Tujuan penelitian ini adalah untuk mendapatkan rancang bangun dari *baseband* WiMAX pada DSK TMS320C6713 dengan berbantuan Simulink.

1.4 BATASAN MASALAH

Hal-hal yang tidak dilakukan pada skripsi ini adalah :

1. Tidak melakukan uji coba pada berbagai jenis sumber *input*.
2. Tidak melakukan uji coba dengan berbagai macam jenis gangguan.
3. Tidak melakukan uji coba hasil gelombang pada *oscilloscope*.

1.5 METODOLOGI PENELITIAN

Metodologi yang digunakan adalah merancang *baseband* WiMAX. Hal pertama yang dilakukan adalah mempelajari teknologi WiMAX dan kemudian mempelajari bagian-bagian dari *baseband* WiMAX. Penelitian kemudian dilanjutkan dengan membangun *baseband* WiMAX pada blok model simulasi pada Simulink. Setelah hasil yang dikehendaki didapatkan, maka selanjutnya adalah menerapkan model simulasi pada DSK TMS320C6713. Penerapan pada DSK TMS320C6713 dilakukan dengan bantuan perangkat lunak tambahan yaitu *Code Composer Studio* (CCS).

1.6 SISTEMATIKA PENELITIAN

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang penulisan, perumusan masalah, tujuan penulisan, batasan masalah, metodologi penelitian, dan sistematika penulisan skripsi untuk memberikan gambaran umum mengenai permasalahan yang dibahas dalam skripsi ini.

BAB II WIMAX

Pada bab ini dibahas tentang dasar teori dan pengetahuan-pengetahuan dasar yang diperlukan untuk dapat memahami teknologi dari *baseband* WiMAX.

BAB III RANCANG BANGUN *BASEBAND* WIMAX

Bab ini membahas pembuatan rancang bangun *baseband* WiMAX pada perangkat lunak Simulink serta penerapannya pada perangkat keras DSK TMS320C6713.

BAB IV UJICOBA DAN ANALISIS

Bab ini berisi uji coba dan analisis dari model *baseband* WiMAX yang telah dibuat pada Simulink dan diterapkan pada DSK TMS320C6713.

BAB V KESIMPULAN

Pada bab ini berisi kesimpulan dari keseluruhan penelitian rancang bangun *baseband* WiMAX ini.

BAB II

WORLDWIDE INTEROPERABILITY FOR MICROWAVE ACCESS (WiMAX)

2.1 WiMAX

Worldwide Interoperability for Microwave Access (WiMAX) merupakan standar industri yang bertugas menginterkoneksi berbagai standar teknis yang bersifat global menjadi satu kesatuan. Standar ini mengatur penggunaan perangkat nirkabel untuk keperluan jaringan perkotaan (*Metropolitan Area Network/MAN*). Standar ini khususnya dirancang untuk memenuhi kebutuhan jaringan akan akses nirkabel berkecepatan tinggi atau BWA (*broadband wireless access*). Kehadiran teknologi ini diharapkan akan memungkinkan akses terhadap berbagai macam aplikasi multimedia *via* koneksi nirkabel dengan jarak antarperangkat yang lebih jauh.

WiMAX terbagi menjadi dua model pemanfaatan yang masing-masing diwakili oleh dua standar IEEE yang berbeda. Model pemanfaatan pertama adalah pemanfaatan *fixed-access* atau sambungan tetap yang menggunakan standar IEEE 802.16-2004, standar ini termasuk dalam golongan layanan *fixed wireless* karena menggunakan antena yang dipasang di lokasi pelanggan. Sementara model pemanfaatan kedua, sering disebut pemanfaatan *portable* atau *mobile* yang menggunakan standar IEEE 802.16e. Standar ini khususnya diimplementasikan untuk komunikasi data pada berbagai macam perangkat genggam, atau perangkat bergerak (*mobile*) seperti PDA atau *notebook*. [3]

2.2 Teknologi WiMAX

Standar IEEE 802.16-2004 dan IEEE 802.16e mencakup lapisan fisik (*PHY layer*) dan *Medium Access Control (MAC)*. Standar IEEE 802.16e untuk menggabungkan layanan komunikasi *fixed* dan *mobile* dalam satu *band* terlisensi. Cakupannya adalah untuk menghasilkan peningkatan standar IEEE 802.16a agar mendukung pergerakan *user* pada kecepatan *vehicular* dan untuk menggabungkan *fixed* dan *mobile* BWA. Fungsional untuk lapisan yang lebih tinggi adalah

mekanisme *handoff* antar *base station* atau sektor. *Band* frekuensi terlisensi yang cocok untuk *mobility* adalah antara 2 sampai 6 GHz.[3]

Standar IEEE 802.16-2004 menyebutkan bahwa ada tiga perbedaan lapisan fisik yang dapat digunakan pada konjungsi dengan lapisan MAC untuk memberikan jaringan *end-to-end* data yang dapat diandalkan. Berikut spesifikasi dari lapisan fisik :

- *Single Carrier* (SC) modulasi *air interface*
- *256-point FFT OFDM multiplexing sceme.*
- *2048-point FFT OFDMA sceme.*

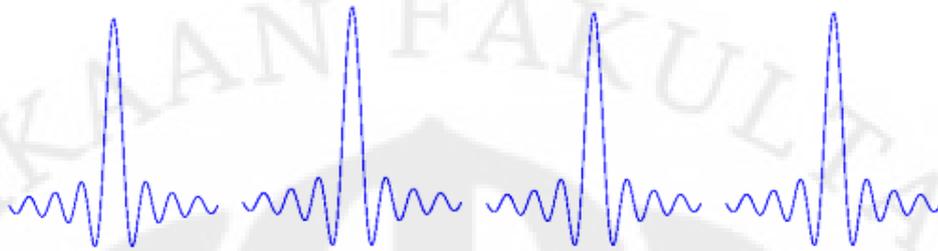
Single Carrier (SC) digunakan saat transmisi *line-of-sight* (LoS), sedangkan untuk transmisi *non line-of-sight* (NLoS) digunakan lebih tepatnya dua sistem dasar OFDM agar persamaan untuk proses sinyal *multicarrier* lebih sederhana. Standar *fixed* WiMAX menggunakan *256-point FFT OFDM* untuk spesifikasi pada lapisan fisik. Selain itu, sistem *fixed* WiMAX menyediakan jangkauan wilayah sampai 5 km dengan maksimum data rate sampai 70 Mbps pada sebuah 20 MHz kanal *bandwidth*, dan menawarkan kepada pengguna koneksi *broadband* tanpa membutuhkan *line-of-sight* langsung ke *base station*.

WiMAX sebagai salah satu sistem yang diunggulkan pada lapisan fisik untuk komunikasi 4G, didukung oleh beberapa teknik unggulan dalam hal sistem komunikasi yang dipakai. Teknik unggulan yang dimaksud di antaranya adalah *Orthogonal Frequency Division Multiplexing* (OFDM) dan pengkodean kanal yang digunakan yaitu *Reed-Solomon* dan Kode Konvolusi.[4]

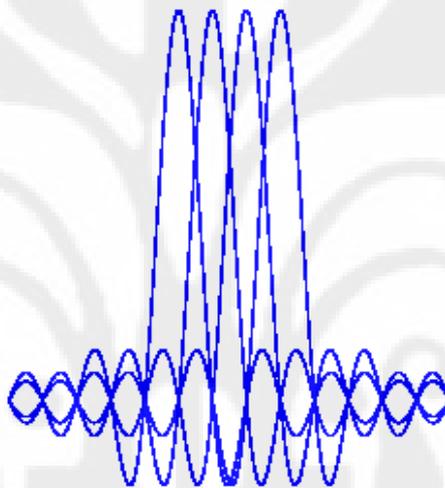
2.2.1 *Orthogonal Frequency Division Multiplexing* (OFDM)

OFDM merupakan bentuk khusus untuk FDM (*Frequency Division Multiplexing*). Pada FDM, suatu *bandwidth* tertentu dibagi menjadi beberapa kanal yang tersendiri. Agar tidak saling menginterferensi satu sama lain maka diberi jarak antar kanal (*guardband*), hal ini menyebabkan *bandwidth* yang digunakan tidak efisien. Sedangkan untuk OFDM, keempat kanal yang ada dalam satu *bandwidth* seakan-akan ditumpang tindihkan menjadi satu. Tidak ada jarak yang diberikan antar kanal (*guardband*). Dengan demikian, OFDM sangat efisien dalam penggunaan *bandwidth*. Gambar 2.1 menunjukkan model spektrum

frekuensi dari FDM yang terdiri dari empat kanal dalam satu *bandwidth* sedangkan Gambar 2.2 menunjukkan model spektrum frekuensi dari OFDM yang terdiri dari empat kanal dalam satu *bandwidth*.



Gambar 2.1 Spektrum *Frequency Division Multiplexing* (FDM)



Gambar 2.2 Spektrum *Orthogonal Frequency Division Multiplexing* (OFDM) [5]

Spektrum frekuensi kanal pada OFDM dapat ditumpangtindihkan dan tidak terjadi saling interferensi antar kanal, hal ini terjadi karena masing-masing sinyal transmisi dalam setiap kanal bersifat saling *orthogonal* satu dengan yang lain. Kanal-kanal tersebut tidak akan saling menginterferensi karena bernilai nol atau saling meniadakan, hal ini disebabkan oleh setiap kanal yang berdekatan jatuh tepat pada titik tengah spektrum yang membawa informasi (spektrum yang memiliki *power* tertinggi). Untuk mengatur supaya kanal spektrum satu dengan yang lainnya jatuh tepat pada titik tengah spektrum yang membawa informasi, setiap sinyal transmisi pada setiap kanal harus bersifat saling *orthogonal*. [6]

2.2.1.1 Sinyal *Orthogonal*

Sebuah sinyal dapat diinterpretasikan sebagai sebuah vektor. Seperti halnya sebuah vektor, sinyal dapat dioperasikan dengan operasi-operasi vektor baik berupa penambahan, perkalian baik dengan skalar ataupun dengan vektor lainnya. Dengan demikian, sinyal memenuhi sifat matematis dari ruang vektor.

Jika ada dua buah vektor dan diintegalkan yakni memenuhi Persamaan dibawah ini :

$$\int_0^T S_i(t)S_j(t)dt \quad \begin{cases} C & i=j \\ 0 & i \neq j \end{cases} \quad (2.1)$$

Maka hasil perkalian antara dua fungsi tersebut akan bernilai nol jika basis vektor antara kedua fungsi tersebut berbeda, maka perkalian antara kedua fungsi tersebut dikatakan *orthogonal*.

Jika kita terapkan pada transformasi Fourier, apabila kita mempunyai sebuah sinyal $x(t)$ dalam selang waktu $0 \leq t \leq T$ dan $x(t) = 0$ di tempat lain.

$$x(t) = \begin{cases} C & 0 \leq t \leq T \\ 0 & \text{selainnya} \end{cases} \quad (2.2)$$

Maka dengan menggunakan transformasi Fourier kita dapat menuliskannya sebagai:

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(2\pi f_k t) - \sum_{k=1}^{\infty} b_k \sin(2\pi f_k t) \quad (2.3)$$

Dengan a_k dan b_k adalah koefisien Fourier dengan:

$$a_k = \frac{2}{T} \int_0^T \cos(2\pi f_k t) x(t) dt \quad (2.4)$$

dan

$$b_k = -\frac{2}{T} \int_0^T \sin(2\pi f_k t) x(t) dt \quad (2.5)$$

dimana T adalah periode dari sinyal, t adalah waktu dan f_k adalah frekuensi dari gelombang sinusoidal dimana $f_k = k/T$ k adalah bilangan bulat ($k = 0,1,2,3,4,\dots$).

Jika kita analogikan ke dalam persamaan gelombang sinusoidal yakni dalam bentuk :

$$x(t) = A \sin \omega t \quad (2.6)$$

Dimana A adalah amplitude sedangkan $\sin \omega t$ adalah basis vektor dari $x(t)$ maka pada tranformasi Fourier koefisien Fourier (a_k dan b_k) adalah amplitude dari gelombang cosinus dan (*negatif*) sinus dari frekuensi f_k , sedangkan gelombang cosinus dan (*negatif*) sinus adalah basis dari transformasi Fourier dalam selang waktu T . Setiap sinyal dapat dituliskan sebagai kombinasi *linear* dari basis-basisnya.

Bentuk transformasi Fourier diatas dapat kita tuliskan kembali dalam bentuk umum bahwa sebuah fungsi dibangun dari kombinasi *linear* dari basis-basisnya dengan memandang sinyal $x(t)$ sebagai vektor berdimensi N ($\mathbf{x} \in \mathfrak{R}^N$) dalam vektor-vektor basis $\{\mathbf{v}_i\}_{i=1}^N$ sebagai berikut:

$$x = \sum_{i=1}^N \alpha_i \cdot v_i \quad (2.7)$$

koefisien α_i diperoleh dengan:

$$\alpha_i = v_i \cdot x \quad (2.8)$$

Basis-basis dari vektor yang digunakan disebut *orthonormal* jika antara dua basis vektor yang berbeda saling *orthogonal* satu sama lain. Dengan demikian, jika masing-masing basis vektor kita normalisasikan, maka perkalian titik dari dua basis vektor yang berbeda adalah nol. Sedangkan perkalian titik dari dua basis vektor yang sama memberikan nilai 1. Atau dapat dituliskan:

$$\mathbf{v}_i \cdot \mathbf{v}_k = \delta_{ik} \quad (2.9)$$

dimana δ_{ik} adalah *Kronecker Delta* ($\delta_{ik} = 0$ untuk $i \neq k$ dan $\delta_{ik} = 1$ untuk $i = k$).

Transformasi Fourier menguraikan sinyal ke dalam fungsi basisnya yaitu dalam bentuk cosinus dan (*negatif*) sinus dengan frekuensi $f_k = k/T$ dimana k adalah bilangan bulat sehingga untuk gelombang cosinus dan (*negatif*) sinus. Gelombang-gelombang tersebut akan saling *orthogonal* karena frekuensinya saling berkelipatan. Ini berhubungan dengan sifat bahwa dua sinyal sinusoidal saling *orthogonal* jika salah satu sinyal memiliki frekuensi sebesar kelipatan bulat

dari frekuensi sinyal yang lainnya. Sedangkan untuk frekuensi yang sama, sinyal cosinus dan (negatif) sinus saling *orthogonal*. Karenanya kita dapat melihat transformasi Fourier sebagai penguraian suatu sinyal kepada basis-basis fungsi yang saling *orthonormal*.

Pada aplikasi nyata, sinyal yang digunakan adalah sinyal kompleks. Jika kita mempunyai sebuah sinyal kompleks $s(t)$ pada selang waktu $[0, T]$ dan nol pada daerah lain, maka kita dapat menuliskan transformasi Fourier untuk sinyal kompleks sebagai:

$$s(t) = \sum_{k=-\infty}^{\infty} \alpha_k v_k(t) \quad (2.10)$$

dengan fungsi basis:

$$v_k(t) = \sqrt{\frac{1}{T}} \exp\left(j2\pi \frac{k}{T} t\right) \Pi\left(\frac{t}{T} - \frac{1}{2}\right) \quad (2.11)$$

fungsi-fungsi basisnya saling *orthonormal*, atau dapat dituliskan:

$$\int_{-\infty}^{\infty} v_i^*(t) v_k(t) dt = \delta_{ik} \quad (2.12)$$

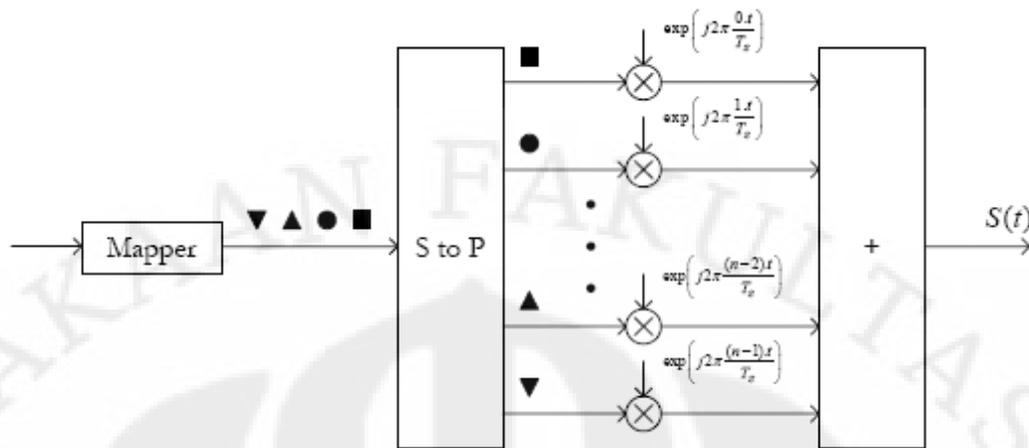
koefisien α_k didapat dengan:

$$\alpha_k = \int_{-\infty}^{\infty} v_i^*(t) s(t) dt \quad (2.13)$$

Untuk sinyal kompleks, fungsi basis yang dihasilkan dari transformasi Fourier adalah juga *orthonormal*. Dengan demikian, kita dapat melihat bahwa transformasi Fourier memiliki peranan yang besar dalam pembangkitan sinyal OFDM.

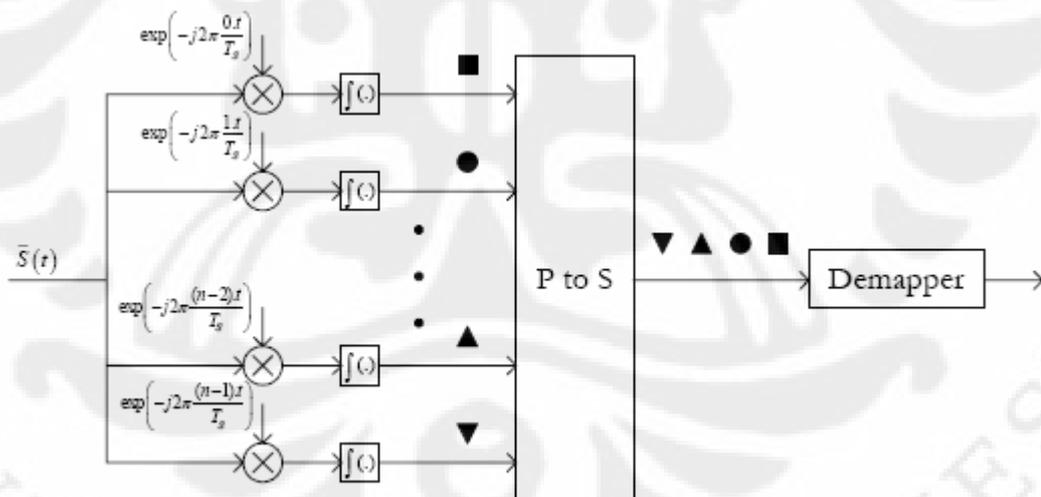
2.2.1.2 Prinsip kerja OFDM

Proses pembangkitan sinyal OFDM yang terjadi pada pengirim dapat dilihat pada Gambar 2.3. Prinsip kerja OFDM adalah membagi kecepatan transmisi data yang hanya dibawa oleh satu frekuensi pembawa menjadi lebih rendah kecepatan datanya ke dalam beberapa *subcarrier* yang saling ortogonal.



Gambar 2.3 Modulator OFDM dengan N subcarrier

Dalam wilayah frekuensi, aliran simbol hasil pemetaan yang masuk pada serial to parallel akan dibagi ke dalam beberapa jalur frekuensi. Simbol hasil proses paralel akan memodulasikan sinyal dari frekuensi *subcarrier* yang saling ortogonal. Proses yang terjadi pada perangkat OFDM penerima merupakan kebalikan dari proses yang terjadi pada perangkat pengirim, dapat dilihat pada Gambar 2.4.

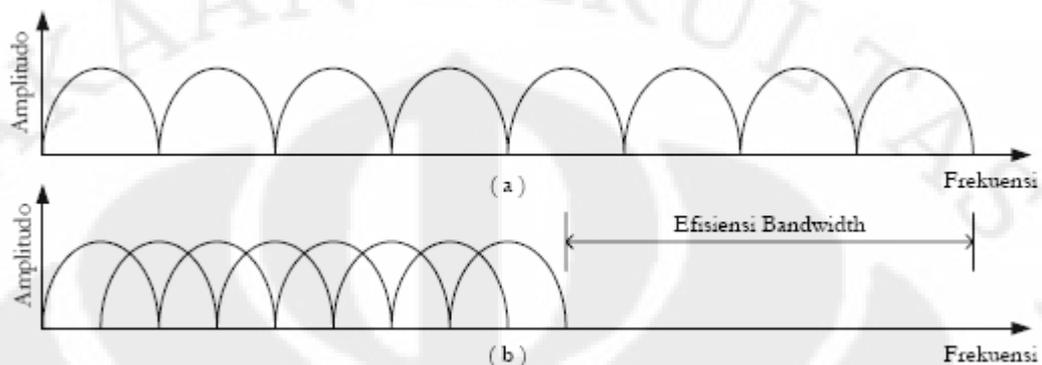


Gambar 2.4 Penerima OFDM dengan N subcarrier

Sinyal yang diterima $s(t)$ akan terlebih dahulu memodulasi sinyal *subcarrier* referensi yang terdapat pada bagian penerima, dikenal dengan proses demodulasi. Keluaran proses demodulasi akan dimasukkan ke dalam blok *parallel to serial* sehingga didapat deretan simbol yang dikirimkan. Tentu saja simbol

yang diterima tidak dijamin benar, tergantung kondisi kanal yang dilewati simbol tersebut.

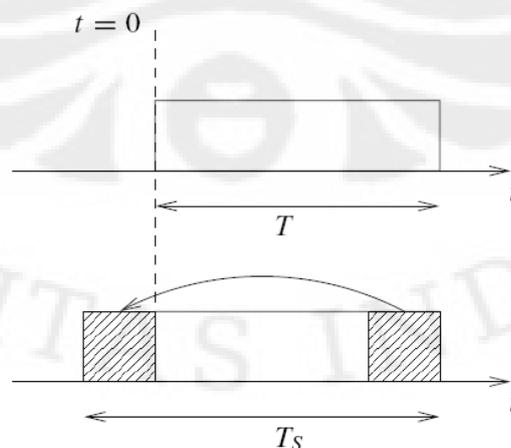
Gambar 2.5 menunjukkan salah satu kelebihan OFDM dibandingkan dengan FDM..



Gambar 2.5 Spektral sinyal multicarrier (a) non OFDM, (b) OFDM

2.2.1.3 Penggunaan Cyclic Prefix

Penambahan ekstensi siklis berfungsi sama dengan pemberian *guard interval* yaitu agar tidak terjadinya penundaan yang kemudian menyebabkan ISI dan IFI. Akan tetapi penambahan *guard interval* yang tidak tepat pada sistem komunikasi yang menggunakan OFDM akan membuat kehilangan sifat ortogonal pada *subcarrier* yang berdekatan. Agar sinyal OFDM tidak kehilangan sifat ortogonalnya maka harus dilakukan penambahan ekstensi siklis dengan cara mencuplik beberapa *frame* OFDM pada bagian akhir dan meletakkannya pada bagian awal sinyal OFDM tersebut, seperti yang terlihat pada Gambar 2.6 :



Gambar 2.6 Penambahan *Cyclic Prefix* pada simbol

2.2.2 Teknik Pengkodean Kanal.

Pada standar IEEE 802.16e ditawarkan beberapa mekanisme teknik pengkodean kanal. Pilihan yang digunakan pada penelitian ini adalah teknik pengkodean kanal dua tahap yaitu *Reed-Solomon* untuk pengkodean bagian luarnya (*outer code*) dan kode konvolusi pada bagian pengkodean dalam (*inner code*). Alasan penggunaan pengkodean berlapis agar sistem handal terhadap *random error* dan *burst error*. [4]

2.2.2.1 Reed-Solomon [2][7]

- *Reed-Solomon Encoder*

Kemampuan dari kode *Reed-Solomon* sangat tepat untuk diaplikasikan pada kesalahan yang terjadi pada *bursts*. Koreksi kesalahan *Reed-Solomon* adalah suatu rencana persandian yang bekerja membangun terlebih dulu suatu *polynomial* dari suatu simbol data untuk dipancarkan, dan kemudian mengirimkan suatu versi yang *oversampled* dari *polynomial* tersebut sebagai pengganti simbol yang asli dari data tersebut.

Suatu kode *Reed-Solomon* dilambangkan sebagai $RS(n, k, t)$ dengan simbol i -bit. Maksudnya adalah bahwa *encoder* memberikan k data simbol pada masing-masing i -bit dan menambahkan $2t$ simbol *parity* untuk memperbaiki n -simbol codeword. Jadi, n , k , dan t dapat didefinisikan sebagai berikut :

n : nomor dari *bytes* setelah penyandian.

k : number dari data *bytes* sebelum penyandian

t : nomor dari data *bytes* yang dapat diperbaiki

kemampuan koreksi kesalahan dari sebuah kode *Reed-Solomon* ditentukan oleh $(n-k)$. jika letak simbol yang salah tidak diketahui, maka kode *Reed-Solomon* dapat mengoreksi sampai t simbol, dimana t dilambangkan sebagai $t = (n-k)/2$.

Pada *Reed-Solomon* terdapat *Primitive polynomial* dan *generator polynomial*, *Primitive polynomial* digunakan untuk membuat symbol field dan juga dapat disebut sebagai suatu symbol field dari *generator polynomial*. Kode *generator Polynomial* digunakan untuk menghitung simbol *parity* dan mempunyai spesifikasi format yang ditetapkan, yakni :

$$g(x) = (x + \lambda^0)(x + \lambda^1)(x + \lambda^2)\dots(x + \lambda^{2^t-1}) \quad (2.14)$$

di mana λ adalah elemen *parity* dari *Galois field* di mana pesan masukannya telah didefinisikan.

- *Reed-solomon Decoder*

Reed-Solomon decoder melaksanakan operasi yang diperlukan untuk memecahkan kode sinyal, dan pada bagian akhir mendapatkan pesan yang asli yang dikirim dari sumber. Seperti pada semua blok penerima, *Reed-Solomon decoder* membalikkan langkah-langkah yang terjadi pada *encoder*, seperti yang dijelaskan pada *Reed-Solomon encoder*. jadi , *Reed-Solomon decoder* mengambilkan *codewords* yang panjangnya n setelah sinyal dipecahkan dan lalu mengembalikan pesan yang panjangnya k , prosesnya sama halnya dengan *Reed-Solomon encoder*, yang membedakan adalah pembalikan proses.

2.2.2.2 konvolusi *Encoder*[2][7]

Convolutional Encoder adalah sebuah blok yang berfungsi untuk mengkodekan urutan dari *input* vektor biner untuk menghasilkan *output* vektor biner. Pada umumnya *convolutional codes* dispesifikasikan berdasarkan tiga parameter, yaitu (n , k dan m) dimana :

n = jumlah bit keluaran

k = jumlah bit masukan

m = jumlah *memory register*

Kuantitas dari k/n disebut dengan *code rate*, yang merupakan acuan tingkat efisiensi dari kode. Pada umumnya parameter k dan n berkisar antara 1 sampai 8, m berada diantara 2 sampai 10 dan *code rate* dari *encoder* antara $1/8$ sampai $7/8$. Biasanya *Convolutional Code* mendefinisikan parameternya dalam variabel (n , k , K), dimana :

$$\text{Constraint Length, } K = k(m - 1) \quad (2.15)$$

Suatu konvolusi *encoder* menerima pesan yang panjangnya k dan menghasilkan *codewords* n bit. Secara umum, hal itu terdiri dari suatu *shift register* L segmen, di mana L menandakan panjangnya batasan tersebut.. Jika terjadi suatu garis koneksi dari daftar pergeseran, maka diwakilkan oleh sebuah

bit " satu" di oktal penyajian dari *polynomials*, dan jika tidak ada koneksi diwakili oleh sebuah bit " nol".

2.2.3 *Quardrature Phase Shift Keying* (QPSK)

Hanya ada satu skema pemetaan simbol atau modulasi digital yang digunakan pada skripsi ini, skema tersebut adalah (*Quardrature Phase Shift Keying*) QPSK. Metode QPSK ini adalah suatu metode yang lebih kompleks dalam sistem Pengiriman, Memakai pergeseran phase perkalian 90^0 , Tiap urutan 2 bit dinyatakan dengan *phase* yang berbeda. Tujuannya agar pengiriman data lebih cepat dan penggunaan bandwidth medianya lebih efisien.

Pemilihan dari skema modulasi yang digunakan sangat tergantung dari kondisi kanal radio. Misalkan pada kondisi kanal yang baik, akan digunakan skema modulasi yang kompleksitasnya lebih tinggi, hal ini akan membuat pengirim dapat mengirim data dengan kecepatan yang lebih tinggi. Pada tabel 2.1 menunjukkan pemetaan bit pada QPSK :

Tabel 2.1 Pemetaan bit untuk *mapping* QPSK[4]

Bit	Simbol
00	$1+i$
01	$-1+i$
11	$-1-i$
10	$1-i$

2.2.4 Kanal Propagasi

Pada proses transmisi, sinyal yang sampai ke penerima tidak hanya melewati satu jalur tetapi datang dari berbagai jalur (*multipath*). Sinyal-sinyal *multipath* tersebut akan mengalami pergeseran fasa dan *delay* yang selalu berubah. Pengaruh dari perbedaan panjang lintasan sinyal akan mengakibatkan pergeseran relatif fasa antara komponen fasa utama yang bersuperposisi dengan komponen fasa lintasan lain, hal ini akan mengakibatkan penguatan atau redaman sinyal terima. Akibat dari perlakuan demikian maka tingkat sinyal terima di penerima akan mengalami fluktuasi. Apabila tingkat sinyal terima berada di

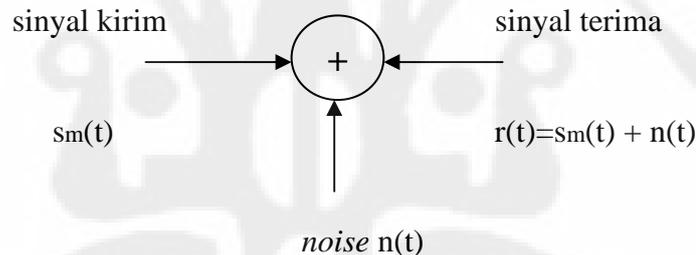
bawah nilai ambang batas penerima maka sinyal tersebut mengalami *fading*. *Fading* yang terjadi akibat adanya *multipath* biasa dinamakan *multipath fading*.

Pada sistem ini, karakteristik kanal diwakili oleh adanya *noise* yang berdistribusi *Gaussian*, *Additive White Gaussian Noise* (AWGN).

2.2.4.1 Kanal AWGN

Pada kanal transmisi selalu terdapat penambahan *noise* yang timbul karena akumulasi *thermal noise* dari perangkat penerima. *Noise* yang menyertai sinyal pada sisi penerima dapat didekati dengan model matematis statistik *Gaussian* dengan rata-rata (*mean*) nol, variansi 1 dan rapat *spektral* daya yang tersebar merata pada lebar pita frekuensi tidak berhingga yang kemudian diistilahkan dengan AWGN.

Additive White Gaussian Noise (AWGN) merupakan model kanal sederhana dan umum dalam suatu sistem komunikasi. Model kanal ini dapat digambarkan seperti Gambar 2.7 :



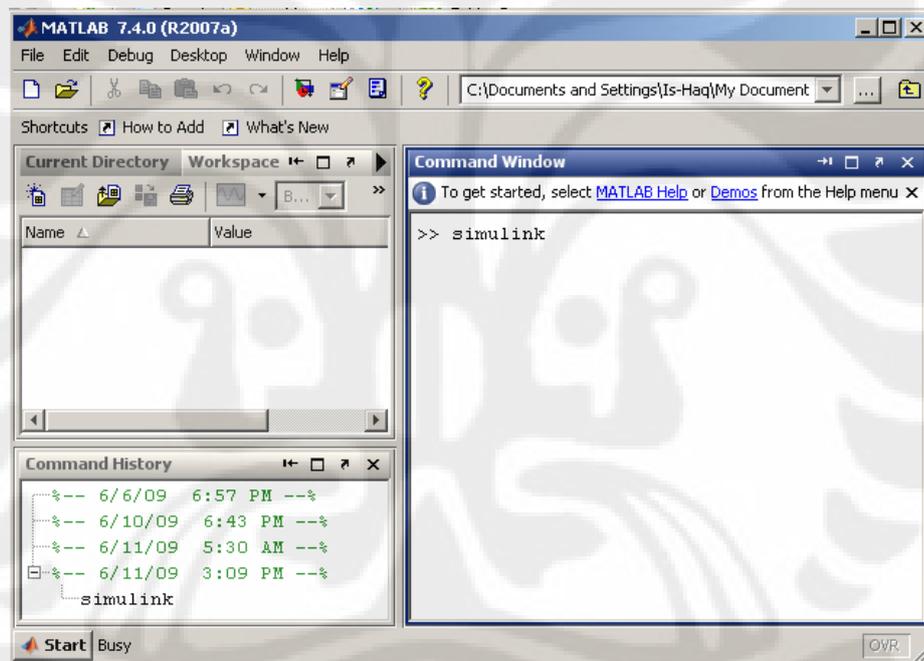
Gambar 2.7 pemodelan kanal AWGN

Sinyal yang dikirim $x(t)$, pada kanal akan terkena derau AWGN $n(t)$. Sehingga sinyal yang diterima menjadi $y(t) = s(t) + n(t)$.

BAB III RANCANG BANGUN BASEBAND WIMAX

3.1 SIMULINK

Simulink adalah suatu perangkat yang sudah terintegrasi dalam program Matlab. Simulink biasa digunakan untuk pemodelan, simulasi dan analisa sistem. Dengan menggunakan perangkat lunak ini kita dapat membangun blok-blok model untuk mensimulasi sistem yang kita kehendaki. Matlab yang digunakan pada pembuatan simulasi ini adalah matlab versi R2007a. Cara menggunakan Simulink hanya dengan mengetikkan perintah "simulink" pada *command window* dari matlab. Cara lain adalah cukup dengan menekan tombol Simulink pada *toolbar* matlab seperti pada Gambar 3.1.



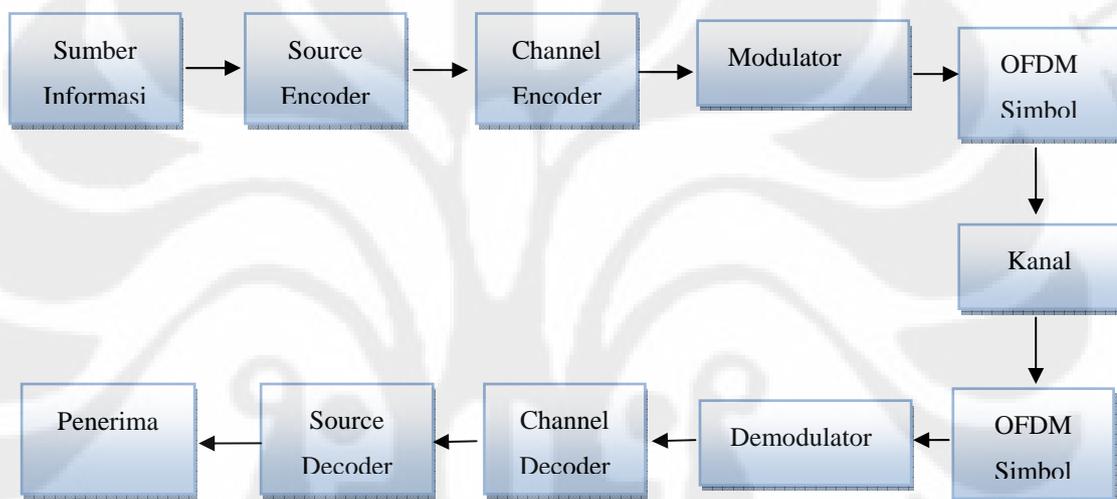
Gambar 3.1 cara menjalankan Simulink

Dengan menjalankan program ini, kita akan diberikan pilihan blok-blok model yang dapat kita gabungkan satu dengan yang lain untuk membentuk simulasi sistem yang ingin dibuat.

3.2 Penjelasan Umum Pemodelan Simulasi *Baseband* WiMAX

Skripsi ini akan menganalisis penyampaian data informasi yang dikirim melalui suatu sumber data ke penerima sumber informasi tersebut dengan menggunakan *platform* perancangan simulasi WiMAX. Kanal yang digunakan pada simulasi ini adalah model kanal berderau AWGN. Untuk mengirimkan suatu informasi dari pengirim ke penerima, informasi tersebut terlebih dilakukan beberapa proses, diantaranya yaitu *channel coding*.

Tahapan proses *channel coding* dalam sistem pemodelan *baseband* WiMAX dapat dilihat pada gambar 3.2 :



Gambar 3.2 Diagram Blok Sistem Transmisi

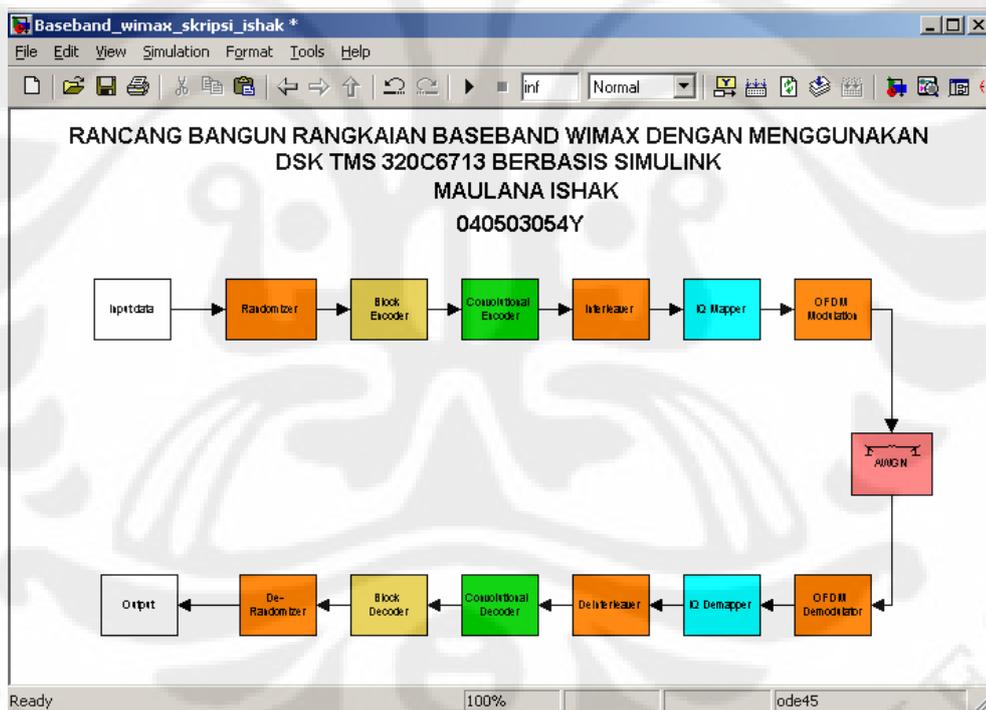
Sumber informasi yang berasal dari *source code* akan dikirim sampai ke penerima, sebelum sumber informasi sampai ke penerima, sumber informasi tersebut harus melalui beberapa proses yaitu *channel coding*. Proses *channel coding* tersebut diantaranya adalah melalui *source encoder* dimana sumber informasi di kode ulang, lalu masuk ke *channel encoder*, blok ini berfungsi untuk menjaga informasi atau data digital dari *error* yang mungkin terjadi selama proses transmisi dengan cara menambahkan bit redundansi (tambahan) ke dalam data yang akan dikirimkan, pada *Channel code* dapat digunakan untuk mendeteksi *error* dan juga untuk mengoreksi *error* tersebut, setelah itu masuk ke *modulator*. Sumber informasi yang telah dikode ulang tersebut masuk ke OFDM simbol sehingga informasi yang dikirim dalam ke adalah saling *orthogonal*, informasi tersebut siap dikirim ke bagian penerima, selanjutnya data informasi tersebut

terkena gangguan dalam hal ini gangguan yang dipakai kanal AWGN. Setelah melalui kanal informasi, data informasi tersebut diterima oleh penerima setelah melalui OFDM simbol, *demodulator*, *channel decoder*, dan *source decoder*.

Perancangan model simulasi WiMAX sebagai *platform* dari mengacu pada beberapa pilihan yang dikeluarkan oleh organisasi dan beberapa peneliti sebelumnya. Namun acuan utama adalah standar IEEE 802.16-2004 dan IEEE 802.16e. Standar IEEE 802.16e merupakan standar akses *wireless* pita lebar yang memiliki laju data yang bervariasi tergantung dari laju keluaran pengkodean kanal, jenis pemetaan dan penambahan *guard interval* yang digunakan.

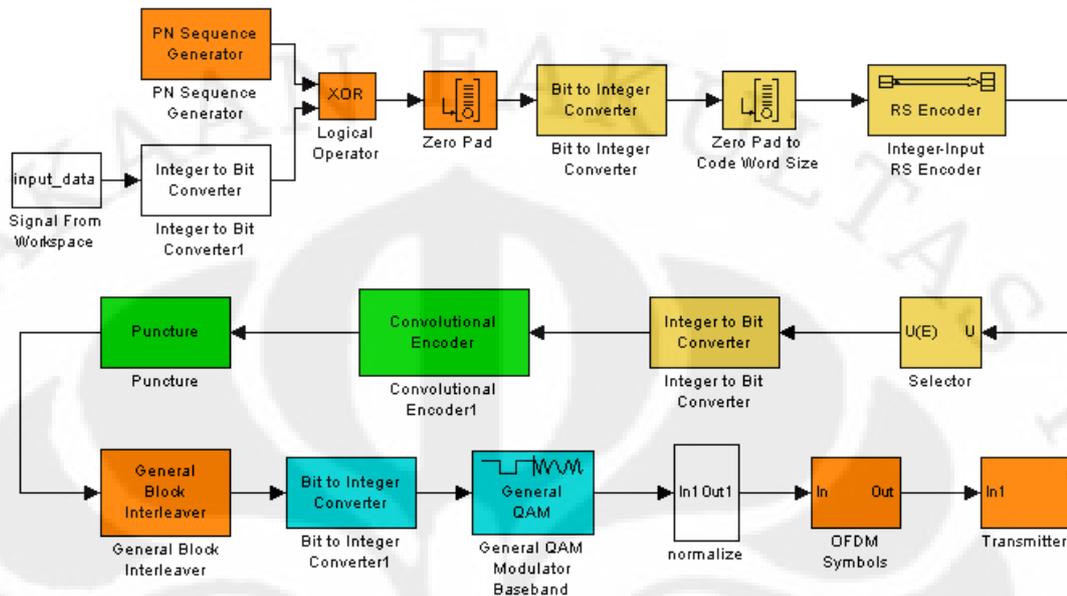
3.3 Blok Diagram *Baseband* WiMAX Pada Simulink

Dengan menggunakan Simulink, maka dibentuklah blok diagram untuk *baseband* WiMAX yang ditunjukkan pada Gambar 3.3 berikut:

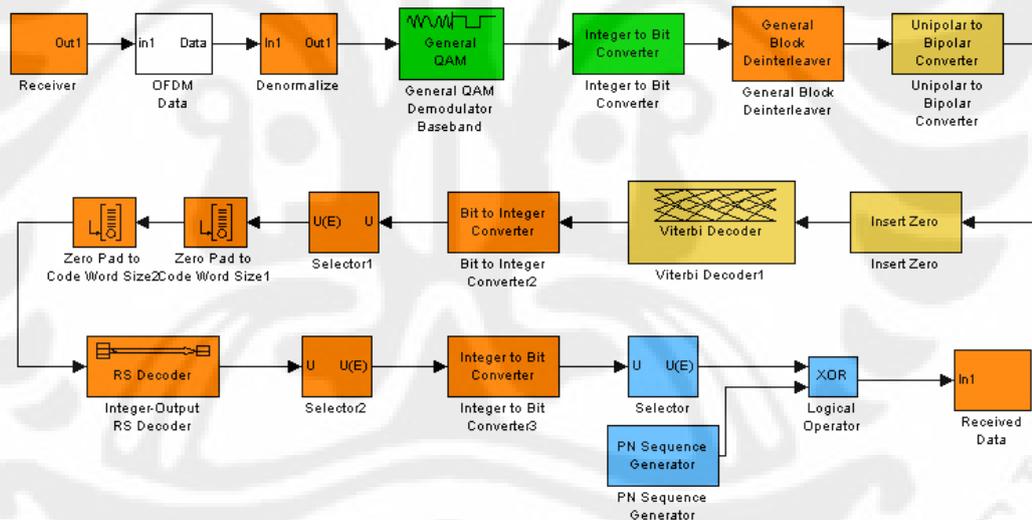


Gambar 3.3 Rancang bangun *baseband* WiMAX

Gambar 3.4 dan Gambar 3.5 menunjukkan diagram blok yang ada pada subblok baseband WiMAX :



Gambar 3.4 Blok diagram *baseband transmitter* WiMAX



Gambar 3.5 Blok diagram *baseband receiver* WiMAX

3.3.1 Blok Diagram *Baseband Transmitter* WiMAX

3.3.1.1 Blok Data

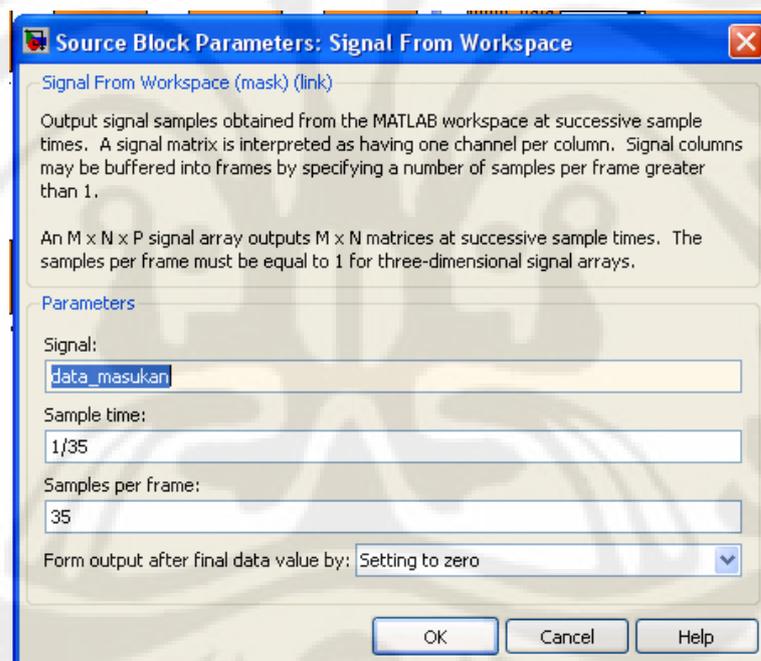
Pada blok data terdiri dari beberapa *sub-blok* diantaranya adalah *input* data dan *integer to bit converter*.

❖ Blok Data_masukan *signal from workspace*

Input data merupakan suatu sumber data yang akan digunakan sebagai sumber informasi yang akan diproses pada simulasi ini, Pada simulasi *baseband* WiMAX dengan menggunakan Simulink ini, *input* data yang diberikan adalah sebagai berikut :

```
data_masukan = sscanf(['45 29 C4 79 AD 0F 55 28 AD 87 B5 76 1A
9C 80 50 45 1B 9F D9 2A 88 95 EB AE B5 2E 03 4F 09 14 69 58 0A
5D'], '%x');
```

Input data tersebut berupa baris vektor yang dijadikan sebagai sumber data yang digunakan untuk model simulasi *baseband* WiMAX ini. Selain itu, parameter sample time yang digunakan adalah 1/35 artinya lamanya waktu yang digunakan untuk 1 *frame*-nya adalah 1 detik sedangkan keluaran yang dihasilkan adalah dalam bentuk *frame* dimana *samples per frame*-nya bernilai 35 bit. Berikut Gambar 3.6 blok sumber data :

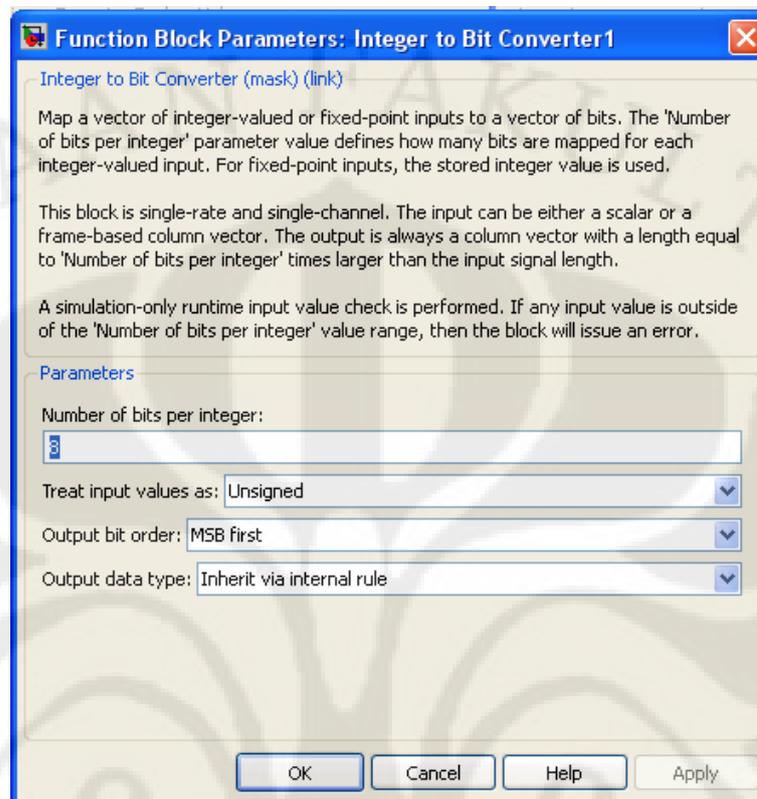


Gambar 3.6 Parameter blok sumber data

❖ *Integer to Bit Converter*

Sub-blok ini berfungsi untuk mengubah setiap nilai *integer* (bilangan bulat) yang berasal dari *input* data yang sudah dalam bentuk *frame* menjadi bilangan biner dengan lebar bit sebesar 8 bit per *integer* seperti terlihat pada Gambar 3.7. Nilai dari bit ini yang nantinya akan di-XOR-kan dengan nilai bit

yang berasal dari blok *PN sequence Generator* yang berfungsi sebagai *noise* pada simulasi ini.

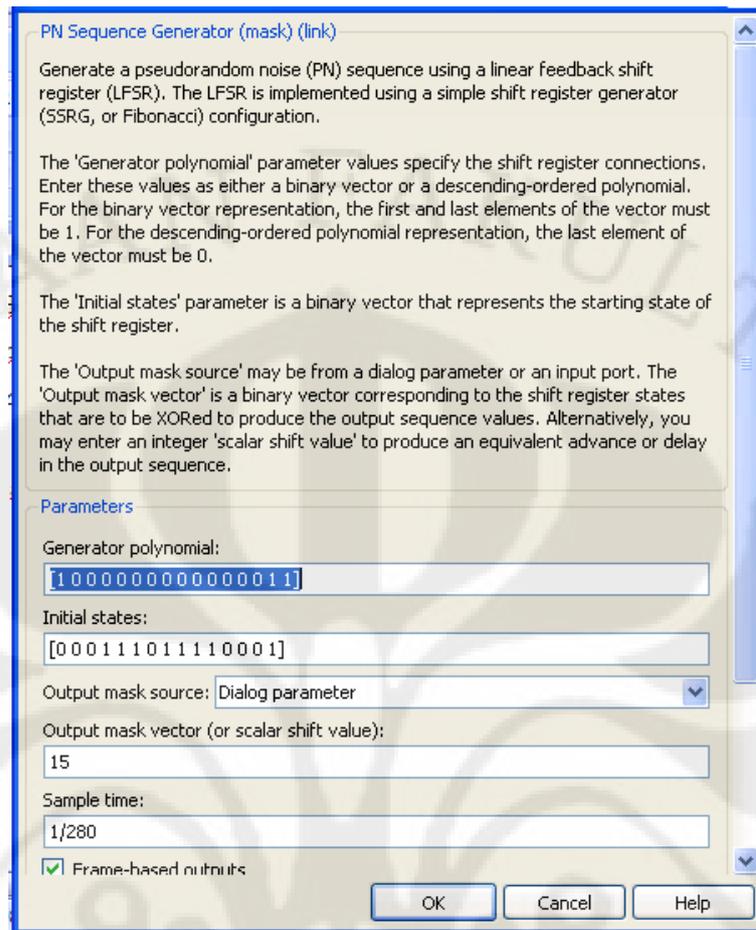


Gambar 3.7 Parameter blok *integer to bit converter*

3.3.1.2 Blok Pengacakan (*Randomizer*)

❖ Blok *PN Sequence Generator* / Pengacakan Bit (*Randomization*)

Blok *PN sequence Generator* adalah suatu rangkaian blok yang berfungsi untuk membangkitkan suatu nilai *pseudorandom binary*. Aliran bit data akan diacak dengan penambahan dari bit keluaran *pseudorandom binary sequence generator*. Metode pengacakan yang digunakan merupakan *sequence polynomial* dengan 15 bit yang dapat ditentukan oleh pengguna. Pada simulasi ini, *Generator polynomial* yang digunakan adalah [1 0 0 0 0 0 0 0 0 0 0 0 0 1 1], seperti terlihat pada Gambar 3.8 sehingga *sequence polynomial* menjadi $x^{15} + x^{14} + 1$, seperti yang terlihat pada Gambar 3.8. Pada setiap pengiriman *frame* data, isi dari *register pseudorandom binary sequence* dibersihkan dan diisi kembali dengan deretan bit [0 0 0 1 1 1 0 1 1 1 1 0 0 0 1], sedangkan parameter *sample time* yang digunakan adalah 1/280 dan keluaran yang dihasilkan adalah dalam bentuk *frame* dimana samples per *frame*-nya bernilai 280 bit.



Gambar 3.8 Parameter blok *PN sequence Generator*

❖ Blok XOR

Hasil Keluaran dari blok *input data* yang telah diubah menjadi bit dengan menggunakan *integer to bit converter* di-XOR-kan dengan blok *PN sequence Generator*, artinya apabila bit yang bertemu antara bit yang ada di *input data* dengan bit yang berada pada Blok *PN sequence* berlainan (0 dan 1) maka keluaran dari bit tersebut adalah bernilai 1. Berikut daftar tabel 3.1 diagram XOR :

Tabel 3.1 Table XOR

A	B	C
Zero	Zero	0
Zero	Nonzero	1
Nonzero	Zero	1
Nonzero	Nonzero	0

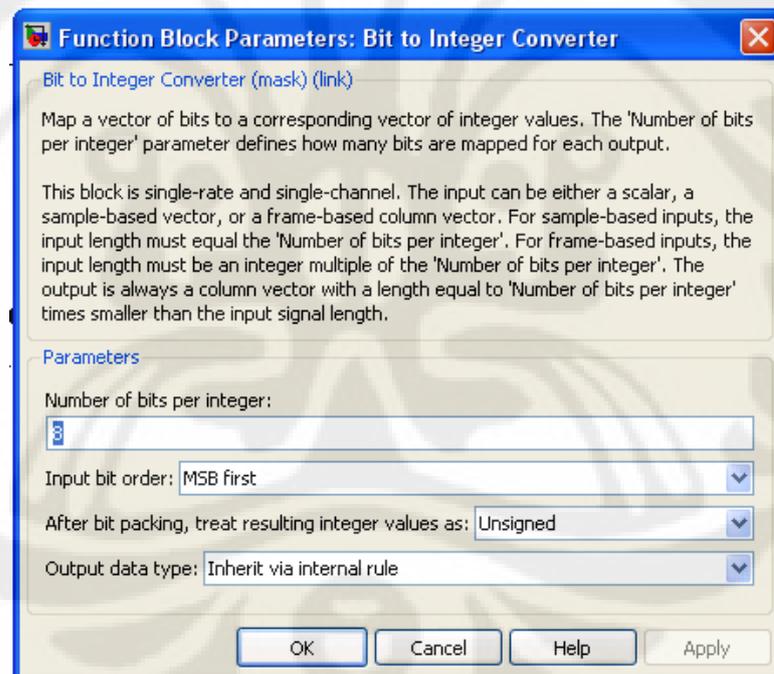
❖ Blok *Zero Pad*

Blok *Zero pad* ini berfungsi sebagai penambah sinyal yang di set 1 tujuannya adalah untuk menjaga masukan blok selanjutnya agar sesuai dengan yang diinginkan, dengan cara menambahkan *zero pad* pada masukan keluaran dari XOR tersebut.

3.3.1.3 Blok *Encoder*

❖ Blok *Bit to Integer Converter*

Blok *bit to Integer Converter* berfungsi untuk mengubah sinyal biner yang beratribut *frame* menjadi sebuah sinyal *integer* (bilangan bulat), sehingga data informasi yang keluar berupa bilangan bulat, hal ini dilakukan karena data keluaran yang berasal dari *zero pad* tersebut akan masuk ke blok *Reed-Solomon encoder*, dimana *Reed-Solomon encoder* masukkannya berupa bilangan *integer*, berikut Gambar 3.9 yang memperlihatkan blok *Bit to Integer Converter* :



Gambar 3.9 Parameter Blok *bit to integer converter*

❖ Blok *Zero Pad to Code Word Size*

Karena ketika bit yang diubah menjadi *integer* tidak sesuai dengan panjangnya nilai *integer* maka bilangan *integer* tersebut di tambahkan *zero pad*-nya agar nilai yang masuk ke *Reed-Solomon encoder* sesuai dengan yang diinginkan.

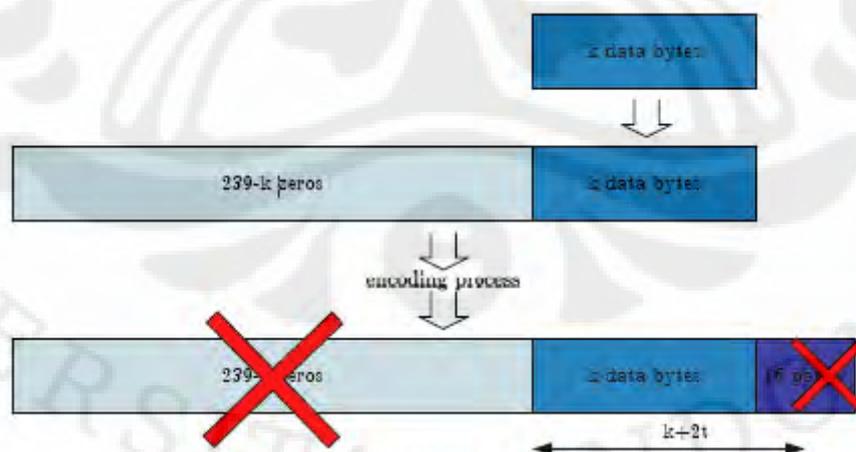
❖ Blok *Reed-Solomon Encoder*

Sebagaimana yang telah ditetapkan dalam standar IEEE 802.16-2004 dan IEEE 802.16e, masukan pengkodean *Reed-Solomon* yang digunakan pada model memiliki 256 level sinyal. Panjang data masukan pengkodean luar (k) adalah sepanjang 239 bytes, panjang keluaran *codeword*-nya (n) adalah sepanjang 255 bytes, terlihat pada Gambar 3.11 maka nilai untuk $n-k$ adalah 16, maka nilai m , yakni besar *Galois Field* (GF) yang menentukan nilai *primitive polynomial* $p(x)$ adalah sebesar $(n-k)/2$ yakni 8, maka *primitive polynomial* dan *generator polynomial* yakni sebagai berikut :[1]

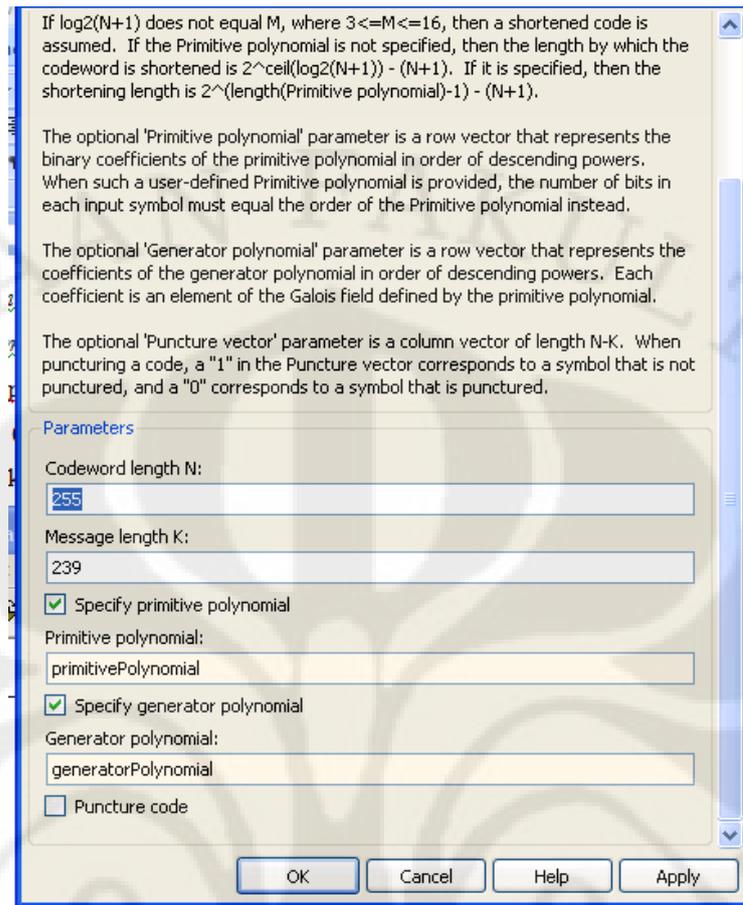
$$p(x) = x^8 + x^4 + x^3 + x^2 + 1 \quad (3.1)$$

$$g(x) = (x + \lambda^0)(x + \lambda^1)(x + \lambda^2)\dots(x + \lambda^{2t-1}) \quad (3.2)$$

Untuk membuat kode *Reed-Solomon* yang fleksibel, yaitu menyediakan variabel *block size* dan kemampuan koreksi variabel, dipendekkan dan *punctured*-kan. Ketika suatu blok dipendekkan sampai k bytes, $239-k$ nol bytes ditambahkan sebagai awalan, dan setelah proses penyaditan, $239-k$ tersebut, nol bytes yang tadi dihilangkan. ketika sekali proses memendekkan telah dilaksanakan, banyaknya simbol yang keluar masuk dari perubahan penyaditan *Reed-Solomon*, dan juga banyaknya simbol yang dapat dikoreksi, t . Dengan proses *puncture*, hanya $2t$ yang pertama dari total 16 bytes *parity* akan diproses. Gambar 3.10 menunjukkan *Reed-Solomon* penyaditan, pemendekkan, dan proses *puncture*.



Gambar 3.10 Proses pemendekkan dan *puncture*

Gambar 3.11 Parameter blok *Reed-Solomon encoder*❖ Blok *Integer to Bit Converter*

Keluaran dari *Reed-Solomon* ini diubah kembali menjadi bit dengan menggunakan blok *integer to bit converter*.

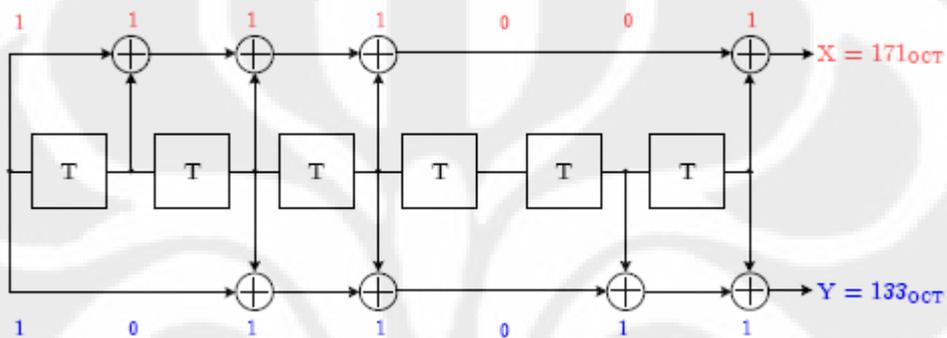
3.3.1.4 Blok *Convolutional Encoder*❖ Blok *Convolutional Encoder*

Setelah proses *Reed-Solomon encoder*, bit data selanjutnya di-*encoder*-kan oleh *convolusi encoder*, berdasarkan standar IEEE 802.16-2004 dan IEEE 802.16e *convolusi encoder* mempunyai suatu *native rate* 1/2 dan suatu batasan panjang 7, hal ini menunjukkan bahwa *encoder* memiliki satu buah *input* dan menghasilkan dua buah *output*. *Generator polynomial* digunakan untuk memperoleh dua keluaran kode bit, yang dilambangkan oleh X dan Y, ditetapkan menurut persamaan berikut :[1]

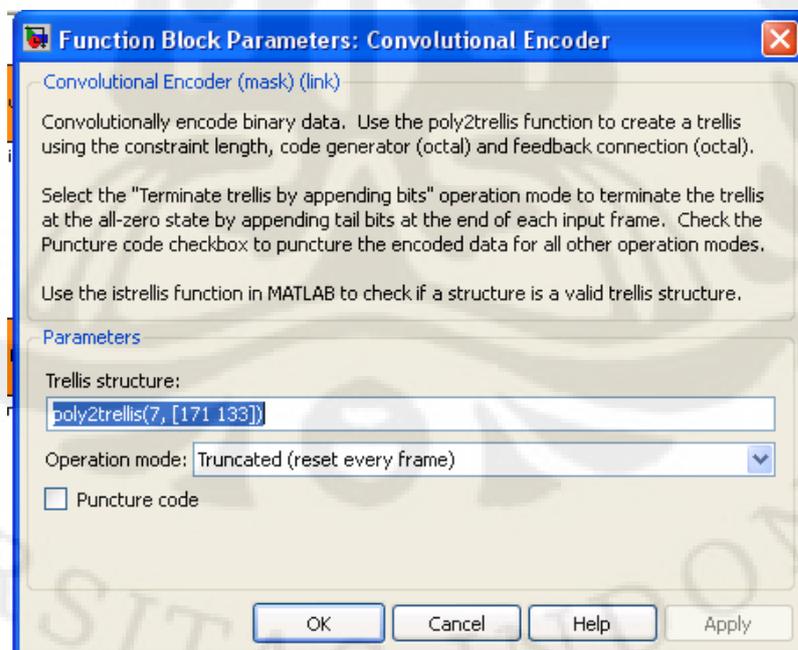
$$G1 = 171_{\text{OCT}} \text{ for } X \quad (3.3)$$

$$G2 = 133_{\text{OCT}} \text{ for } Y \quad (3.4)$$

Bilangan 171 133 merupakan bilangan octal yang menunjukkan keadaan awal dari isi register sebagaimana yang ditunjukkan pada Gambar 3.12, sehingga pada simulink ditulis sebagaimana yang ditunjukkan pada Gambar 3.13. Bilangan tersebut menunjukkan bilangan biner 1111001 dan 1011011. Bilangan biner tersebut apabila dibaca menurut aturan bilangan oktal maka akan menjadi 1 111 001 dan 1 011 011



Gambar 3.12 convolusi *encoder* dengan binary rate 1/2



Gambar 3.13 Parameter Blok *convolutional encoder*

3.3.1.5 Blok *Interleaver*

❖ *General Block Interleaver*

Pada Blok ini berfungsi untuk menyusun kembali *input* tanpa mengulang atau mengabaikan masukan tersebut. Pengurutan elemen tersebut dapat ditentukan sendiri dengan cara memasukkan elemen yang diinginkan.

Pada standar IEEE 802.16e blok interleaver ini, pada persamaan 3.5 memberikan hubungan antara k , m_k , dan j_k , index bit sebelum dan sesudah pada tahapan pertama dan kedua interleaver secara berturut-turut, dimana N_c adalah jumlah bit pada blok dan M adalah alphabet pada modulasi (2 untuk QPSK, 4 untuk QAM dan 6 untuk 64 QAM) dan d adalah parameter yang dapat diubah-ubah, dimana nilainya dibuat 12 :[1]

$$m_k = \left(\frac{N_c}{d} \right) k_{\text{mod}(d)} + \text{floor} \left(\frac{k}{d} \right) \quad (3.5)$$

$$j_k = s \cdot \text{floor} \left(\frac{m_k}{s} \right) + \left(m_k + N_c - \text{floor} \left(\frac{d \cdot m_k}{N_c} \right) \right)_{\text{mod}(d)} \quad (3.6)$$

3.1.3.6 Blok Modulator

❖ Blok *Bit to Integer Converter*

Seperti yang telah diketahui diatas bahwa blok *bit to integer converter encoder* adalah suatu blok yang berfungsi untuk mengubah sinyal biner yang beratribut *frame* menjadi sebuah sinyal *integer* (bilangan bulat). Diubahnya bilangan bit ke dalam *integer* karena data tersebut akan masuk ke blok *general QAM modulator baseband*, dimana masukannya harus berupa *integer* per 2 bit, oleh karena itu nilai yang diset pada blok ini adalah 2.

❖ *General QAM Modulator Baseband*

General QAM modulator baseband menggunakan modulasi *Quadratul amplitude*. Keluaran dari blok modulasi *baseband* QAM ini menggambarkan sinyal yang termodulasi.

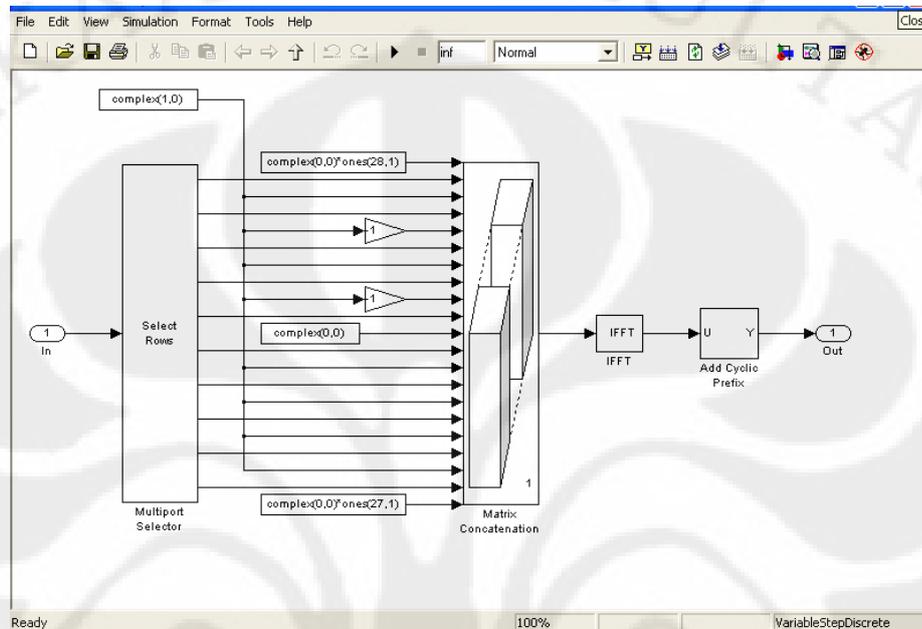
3.1.3.7 Blok *Normalize*

Pada bagian ini data diubah yang awalnya berupa bilangan integer menjadi bilangan dalam bentuk conjugasi, yakni seperti persamaan berikut :

$$\text{conj}(Z) = \text{real}(Z) - i \cdot \text{imag}(Z)$$

3.1.3.8 blok OFDM *Symbol*

Blok *OFDM modulation* ini adalah sebuah *subsystem* yang berfungsi untuk memodulasikan sinyal informasi yang kita miliki dari blok sebelumnya. Blok ini terdiri dari beberapa blok dasar sebagai berikut (Gambar 3.14):



Gambar 3.14 Blok diagram pada blok OFDM *modulation*

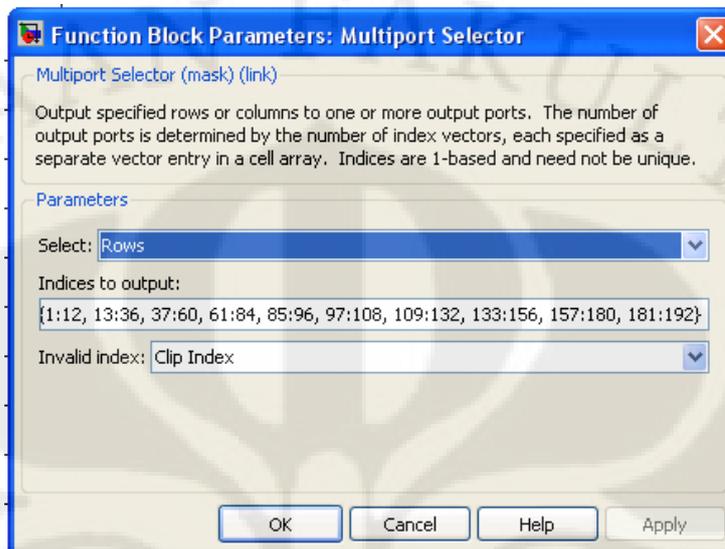
❖ Blok *Multiport Selector*

Pada blok *multiport selector* sinyal sumber yang merupakan data *frame* dengan lebar 192 kemudian dibagi-bagi ke dalam beberapa bagian, tabel 3.2 memperlihatkan urutan pembagian sinyal pada *multiport selector*. Blok *select row* ini mengerjakan bagian *serial-to-parallel* dari blok diagram OFDM.

Tabel 3.2 urutan pembagian sinyal pada *multiport selector*

Bagian	Urutan data
1	1 - 12
2	13 - 36
3	37 - 60
4	61 - 84
5	85 - 96
6	97 - 108
7	109 - 132
8	133 - 156
9	157 - 180
10	181 - 192

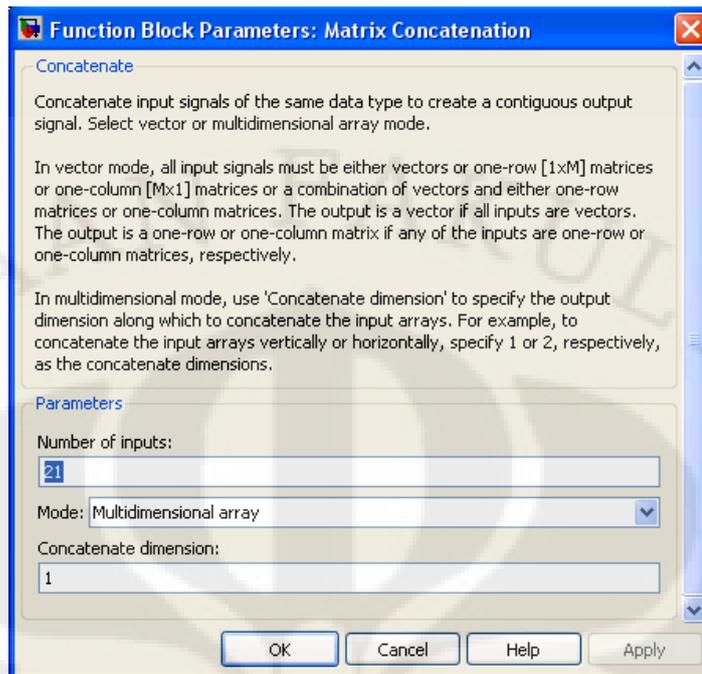
Pembagian tersebut dilakukan dengan memasukkan nilai sesuai dengan Tabel diatas, pada parameter *Indices to output* seperti dapat kita lihat pada Gambar 3.15.



Gambar 3.15 parameter blok *Multiport Selector*

❖ Blok *Matrix Concatenation*

Blok ini mengerjakan bagian dari *parallel-to-serial* dari modulasi OFDM. Pada Gambar 3.11 kita melihat bahwa masukan pada blok ini tidak hanya data sumber informasi tetapi juga beberapa nilai nol. Dengan tambahan nilai tersebut, lebar *frame* menjadi 256. Bilangan-bilangan tersebut berfungsi dalam pembentukan spektrum frekuensi dari sinyal OFDM. Beberapa literatur menjelaskan bahwa dengan memperbanyak nilai-nilai nol yang disisipkan pada modulasi OFDM, pembentukan spektrum frekuensi dari sinyal OFDM akan semakin baik. Pada simulasi ini kita menggunakan FFT length 256 dengan *frame* data 192. Parameter dari blok ini dapat dilihat pada Gambar 3.16.



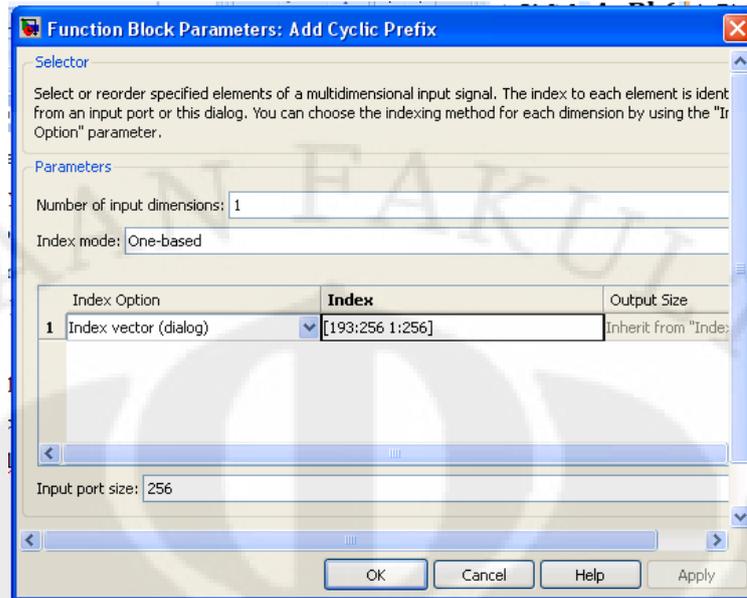
Gambar 3.16 parameter blok *Matrix Concatenation*

❖ Blok IFFT

Pembentukan sinyal OFDM dapat dengan mudah dilakukan dengan menggunakan algoritma IDFT seperti telah dijelaskan sebelumnya. Blok IFFT ini adalah blok yang mengerjakan penghitungan *Inverse Fast Fourier Transform* dari sinyal. Panjang sinyal yang melalui blok ini adalah 256 sebagaimana keluaran dari blok *OFDM Modulation*. Keluaran dari blok ini adalah sinyal yang telah orthogonal dengan panjang *frame* 256.

❖ Blok *Add Cyclic Prefix*

Penggunaan *add cyclic prefix* adalah untuk mengkompensasi ketidak-sinkronan dari sinyal yang akan diterima. Blok ini mengerjakan bagian penambahan *cyclic prefix* pada sinyal OFDM yang telah terbentuk dari IFFT. Pada Gambar 3.17 terlihat pada bagian "*elements (-1 for all elements)*" diisikan nilai [193:256 1:256].



Gambar 3.17 Parameter blok *Add Cyclic Prefix*

Ini berarti bahwa blok ini mengerjakan *cyclic prefix* dengan mengambil data dari elemen 193 sampai dengan 256 dan ditambahkan pada bagian depan dari elemen. Dengan demikian keluaran dari blok ini adalah sebesar:

$$256 + (256 - 193 + 1) = 320$$

3.1.3.9 Blok Kanal AWGN

Pada blok AWGN, akan dilakukan perhitungan *Signal-to-Noise Ratio* (SNR) yang memiliki parameter-parameter sebagai berikut :

$$SNR = 10 \log \frac{S_0}{N_0} \quad (3.7)$$

dimana :

- SNR = Rasio antara daya *noise* dan daya dari sinyal.
- S_0 = Daya dari sinyal
- N_0 = Daya dari *noise*

3.3.2 Receiver

Pada Blok simulasi penerima (*Receiver*) pelaksanaan pengiriman data yang berasal dari pemancar (*transmitter*) pada dasarnya sama dengan perubahan bentuk informasi data yang terjadi pada blok simulasi penerima (*transmitter*), yang membedakan adalah proses yang terjadi pada blok simulasi penerima (*receiver*) perubahan data informasi merupakan kebalikan dari proses perubahan bentuk informasi data yang terjadi pada blok simulasi pemancar (*transmitter*), pada blok simulasi penerima (*receiver*) data informasi yang berasal dari blok OFDM symbol diubah kembali samapai seperti data informasi yang dikirim dari pemancar (*transmitter*), perubahan tersebut melalui beberapa proses, proses tersebut itulah yang merupakan kebalikan dari proses yang terjadi pada simulasi blok pemancar (*transmitter*), sebagai contohnya adalah data yang berasal dari blok OFDM symbol harus dikembalikan bentuk datanya seperti bentuk data yang sebelum masuk ke blok OFDM symbol, yakni dengan cara menggunakan blok OFDM data, dimana pada blok ini didalamnya terdapat *sub*-blok, diantaranya yaitu :

- ❖ Blok *Remove Cyclic Prefix*

Blok *remove cyclic prefix* ini merupakan blok yang fungsinya berkebalikan dengan blok *remove cyclic prefix*, fungsi dari blok ini adalah untuk memisahkan sinyal sebenarnya dengan ekstensi siklis yang kemungkinan telah terkena efek *intersymbol interference* (ISI) akibat pengaruh *multipath*. Proses yang terjadi pada model adalah dengan mencuplik 256 baris waktu terakhir pada setiap matrik sinyal wilayah waktu.

- ❖ Blok FFT

Blok *Fast Fourier Transform* (FFT) ini merupakan blok yang memiliki fungsi yang berkebalikan dengan blok IFFT, yakni berfungsi sebagai *baseband demodulator* dengan cara mengubah sinyal wilayah waktu ke wilayah frekuensi. Jumlah titik FFT yang digunakan sama dengan jumlah titik yang digunakan oleh IFFT yaitu 256-titik.

- ❖ Blok *Frame Status Conversion* dan Blok *Remove Zero-padding and recorder*

Kedua blok ini merupakan *inverse* dari blok *matrix concatenation*, yakni mengembalikan data informasi menjadi beberapa bagian yang berurutan dalam

bentuk *parallel* yang awalnya dalam bentuk *serial*, lalu dari data tersebut yang lebar *frame*-nya 256 diubah menjadi lebar *frame* awalnya yakni 192 dengan cara menghilangkan nilai nol pada data informasi tersebut dengan menggunakan blok *remove zero-padding*.

❖ **Blok *Remove Pilots***

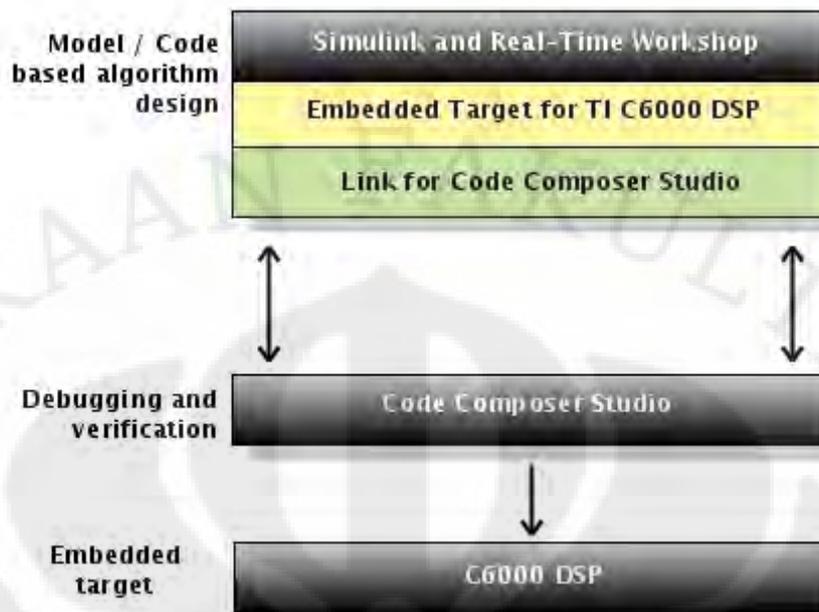
Fungsi dari blok ini adaklah berkebalikan dengan fungsi dari blok *multiport selector*, yakni data yang memiliki lebar *frame* 192 tersebut yang masih bentuk pengelompokan – pengelompokan kecil dijadikan satu kelompok, dengan cara menggunakan fungsi dari blok *remove pilots*.

Untuk *sub*-blok selanjutnya yang ada pada blok simulasi penerima (*receiver*), proses perubahan data informasi diubah samapai dalam bentuk data informasi yang berasal dari sumber informasi, prosesnya tidak jauh berbeda dengan proses yang diatas hanya saja urutan prosesnya berkebalikan.

3.4 Digital Signal Processor Starter Kit TMS320C6713

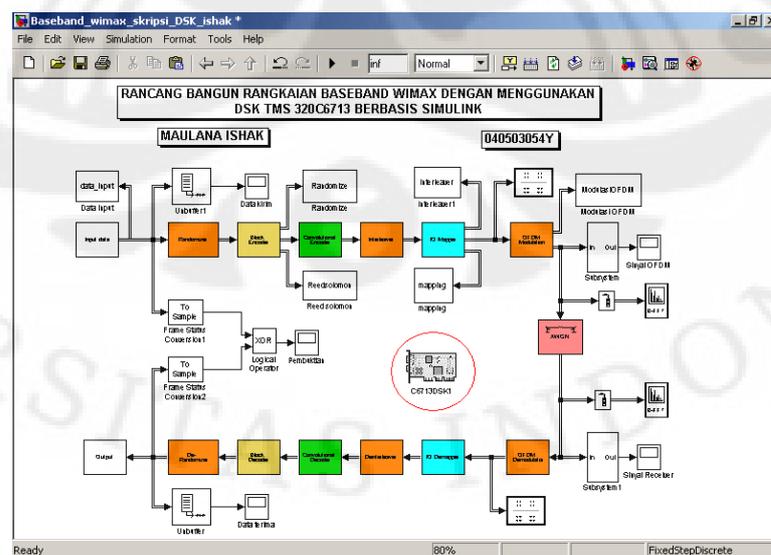
Blok diagram sebagaimana dibuat dalam Simulink telah siap untuk diimplementasikan pada sebuah papan *processor*. Pada rancang bangun ini, papan *processor* yang digunakan adalah DSP Starter Kit TMS320C6713. Proses rancang bangun ini menggunakan *target preference* C6713DSK yang tersedia dalam library Simulink. Selain target preference dibutuhkan *Real Time Workshop*, *Embedded Target for TI (Texas Instrument) C6000 DSP*, dan *Link for CCS* untuk menghubungkan Simulink dengan DSK. Ketiga hal tersebut dapat ditemukan di Simulink dan harus dilakukan pengaturan konfigurasi.

Proses debugging dari Simulink ke DSK memerlukan perangkat lunak lain yaitu Code Composer Studio (CCS) yang merupakan sebuah komposer bahasa C/C++ yang digunakan untuk memrogram papan DSK dengan menggunakan bahasa C/C++. Maka sebenarnya yang dilakukan oleh Simulink adalah membangun kode pemrograman dalam bahasa C/C++ dari model yang disimulasikan. Gambar 3.18 menunjukkan algoritma dari penerapan Simulink pada DSK.



Gambar 3.18 algoritma penanaman model ke dalam TMS320C6713 sebagai *embedded target*

Untuk membuat simulasi pada Simulink dapat dijalankan pada DSK maka terdapat beberapa konfigurasi yang harus ditambahkan dan diubah. Diantara yang perlu ditambahkan adalah menambahkan blok C6713DSK dari library browser ke model yang telah dibuat. Blok ini menandakan bahwa simulasi yang kita buat siap untuk diimplementasikan pada papan DSK TMS320C6713 sebagai *embedded target*. Pada Gambar 3.19 memperlihatkan diagram blok *baseband* WiMAX yang akan di-*burn* pada DSK :

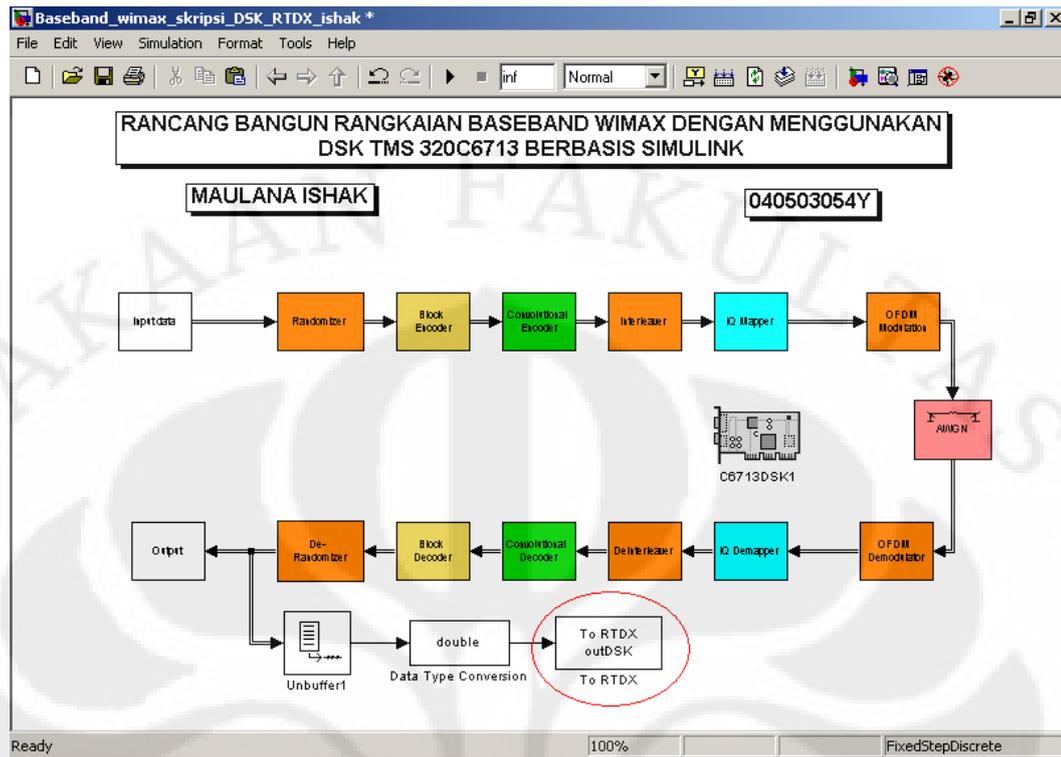


Gambar 3.19 Rangkaian baseband WiMAX untuk DSK

Pengaturan konfigurasi parameter model rangkaian *baseband* WiMAX terhadap ketiga hal tersebut dapat dilakukan dengan memilih menu *simulation* yang ada pada *toolbar* lalu memilih konfigurasi parameter. Selanjutnya akan muncul kotak konfigurasi parameter model seperti yang dapat dilihat pada Gambar 3.11. Pengaturan yang harus dilakukan terhadap konfigurasi parameter model rangkaian penerima OFDM yaitu :

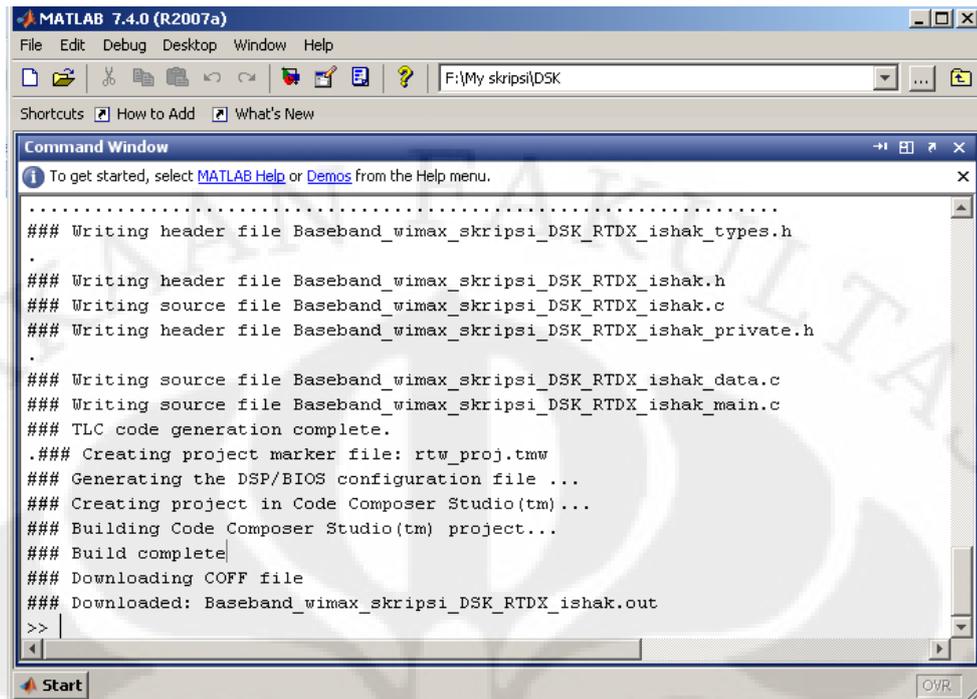
1. Pada *tab solver*, mengubah *type* menjadi *fixed step* dan *solver* menjadi *discrete* pada menu *solver options*.
2. Pada *tab optimization*, melakukan *uncheck block reduction* dan *implement logic signals as boolean data* yang terdapat pada menu *simulation and code generation*.
3. Pada *tab hardware implementation*, mengubah *device type* yang terdapat pada menu *embedded hardware* menjadi TI C6000.
4. Pada *tab real time workshop* di menu *target selection*, mengubah *system target file* menjadi *ccslink_ert.tlc*.
5. Pada *tab real time workshop subtab debug*, melakukan *check verbose build* pada menu *build process*.
6. Pada *tab real time workshop subtab Link for CCS* di menu *project option*, mengubah *system stack size* menjadi 8192.

Selanjutnya dilakukan proses *Incremental Building*. *Incremental Building* dapat dilakukan dengan memilih menu yang terdapat di *tools* lalu *real time workshop* kemudian pilih *build model*. Akan tetapi sebelum melakukan proses *Incremental Building* semua scope dihilangkan dan untuk melihat *output* dengan menggunakan media komputer maka pengujian model rangkaian *baseband* dilakukan dengan menggunakan *Real Time Data Exchange (RTDX)* seperti dapat dilihat pada Gambar 3.20



Gambar 3.20 Rangkaian baseband WiMAX DSK RTDX

Setelah dilakukan proses *build model*, maka matlab akan membuat kode program dalam bahasa C sebagai pengganti blok Simulink untuk dijalankan di DSK. Proses pembuatan program dalam bahasa C oleh matlab ditampilkan pada command window seperti terlihat pada Gambar 3.21.



```

MATLAB 7.4.0 (R2007a)
File Edit Debug Desktop Window Help
F:\My skripsi\DSK
Shortcuts How to Add What's New
Command Window
To get started, select MATLAB Help or Demos from the Help menu.
.....
### Writing header file Baseband_wimax_skripsi_DSK_RTDX_ishak_types.h
.
### Writing header file Baseband_wimax_skripsi_DSK_RTDX_ishak.h
### Writing source file Baseband_wimax_skripsi_DSK_RTDX_ishak.c
### Writing header file Baseband_wimax_skripsi_DSK_RTDX_ishak_private.h
.
### Writing source file Baseband_wimax_skripsi_DSK_RTDX_ishak_data.c
### Writing source file Baseband_wimax_skripsi_DSK_RTDX_ishak_main.c
### TLC code generation complete.
.### Creating project marker file: rtw_proj.tmw
### Generating the DSP/BIOS configuration file ...
### Creating project in Code Composer Studio(tm)...
### Building Code Composer Studio(tm) project...
### Build complete
### Downloading COFF file
### Downloaded: Baseband_wimax_skripsi_DSK_RTDX_ishak.out
>>
Start OVR

```

Gambar 3.21 Proses pembuatan program C

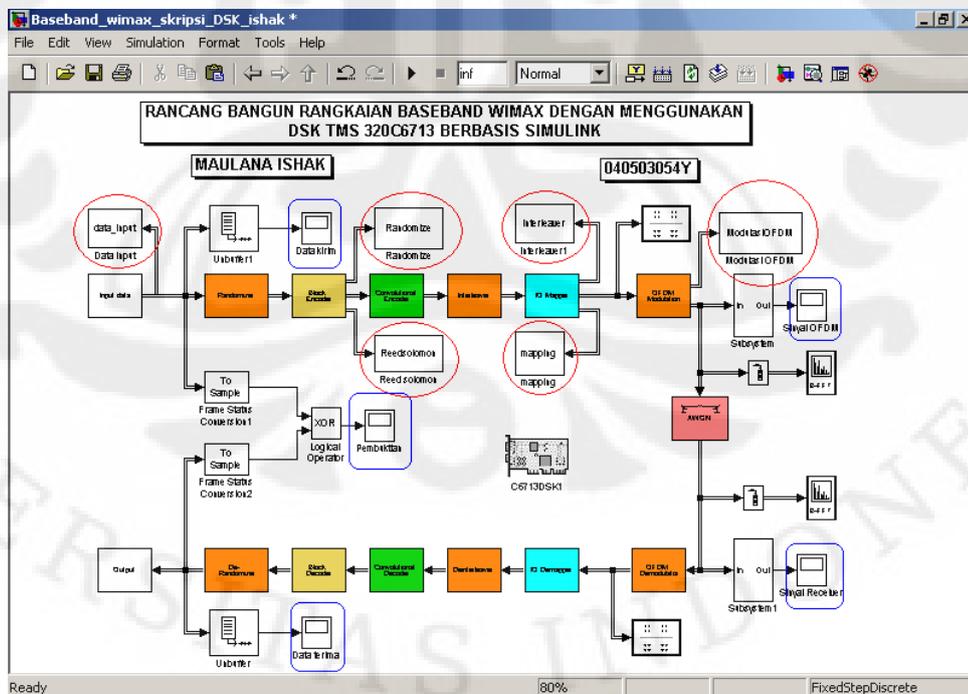
Setelah program C berhasil dibuat maka program tersebut akan dibebankan ke dalam DSK. Model ini kemudian akan berjalan pada perangkat DSK secara *real time*.

BAB IV UJI COBA DAN ANALISIS

Uji coba dan analisis terhadap model rangkaian *baseband* WiMAX yang telah dibuat dilakukan dengan dua metode, metode yang pertama dilakukan uji coba dan analisis simulasi yang telah dibuat di simulink dan metode yang kedua dilakukan uji coba dan analisis simulasi simulink yang telah diterapkan dengan menggunakan alat DSK TMS320C6713. Kedua ujicoba dan analisis tersebut dilakukan dengan cara membandingkan data yang telah diproses di *receiver* dengan data yang dikirim dari *transmitter*.

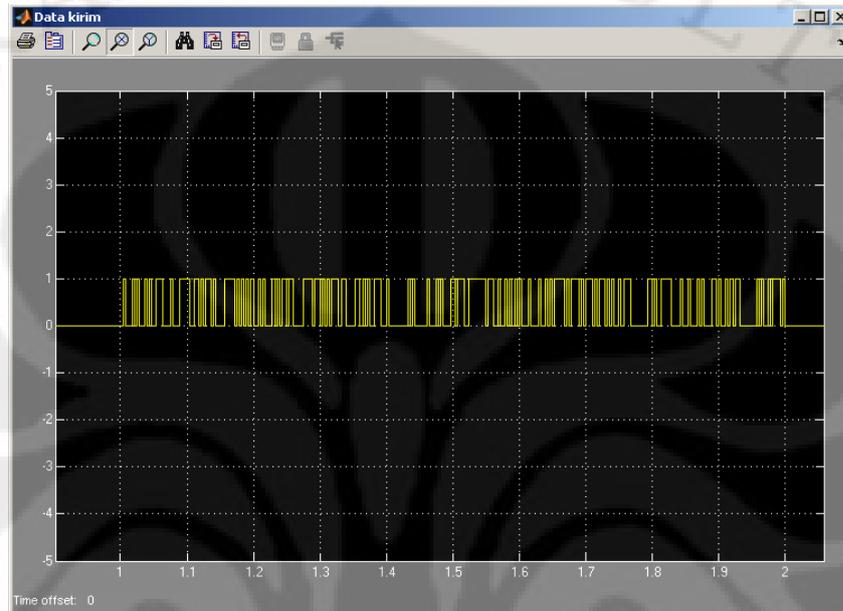
4.1 Pengambilan dan Analisis Data pada Simulink

Pada ujicoba simulink ini, langkah yang dilakukan adalah mengambil data dengan cara memberikan blok *to workspace* dan *scope*. Data yang diambil tambahan blok *to workspace* diberikan nama variabel data *input*, *randomize*, *Reed-Solomon*, *interleaver*, *mapping*, dan modulasi OFDM, sedangkan data yang diberikan blok *scope* pada blok data kirim, sinyal OFDM *transmitter*, sinyal OFDM *receiver*, dan data terima, seperti yang terlihat pada Gambar 4.1 :



Gambar 4.1 Pencantuman blok *to workspace* dan *scope*

Pada keluaran data kirim akan dibandingkan dengan data yang didapat dari sisi *receiver*. Pada sisi *transmitter* data dibangkitkan dengan menggunakan data *input* yang telah dimasukkan ke dalam simulasi setelah diubah ke bilangan biner sehingga data yang dikirim hanya bernilai satu dan nol secara bergantian seperti pada Gambar 4.2.



Gambar 4.2 Data kirim

Pada Gambar 4.2 data berupa bit-bit baru keluar setelah detik ke 1 sampai dengan detik ke 2, dimulai dari detik ke 1 karena data yang dihasilkan adalah *frame based*. Data *frame based* akan dikirim apabila jumlah *samples per frame* telah memenuhi konfigurasi yang telah dibuat sehingga untuk mendapatkan jumlah bit sesuai dengan *samples per frame* yang diinginkan, maka bit-bit data harus disimpan terlebih dahulu sampai jumlah *samples per frame* terpenuhi sehingga terjadi *delay* seperti ditunjukkan Gambar 4.2, sedangkan data yang dikirim hanya sampai detik ke 2 karena data masukan yang dikirim berupa data yang dimasukkan secara manual, sehingga akan berhenti sampai data yang dimasukkan telah diproses seluruhnya.

Pada Tabel 4.1 menunjukkan data masukan beserta perubahan dalam bentuk bit :

Tabel 4.1 variabel “data *input*”

Elemen	<i>Input-an data</i>	Bit
1	69	10001010
2	41	00101001
3	196	11000100
4	121	01111001
5	173	10101101
6	15	00001111
7	85	01010101
8	40	00101000
9	173	10101101
10	135	10000111
11	181	10110101
12	118	01110110
13	26	00011010
14	156	10011100
15	128	10000000
16	80	10100000
17	69	01000101
18	27	00011011
19	159	10011111
20	217	11011001
21	42	00101010
22	136	10001000
23	149	10010101
24	235	11101011
25	174	10101110
26	181	10110101
27	46	00101110
28	3	00000011
29	79	01001111
30	9	00001001
31	20	00010100
32	105	01101001
33	88	01011000
34	10	00001010
35	93	01011101

Selanjutnya data *input-an* menuju blok *randomize* sebelum masuk ke blok *Reed-Solomon* untuk dilakukan pengkodean, proses *randomize* sampai ke blok *Reed-Solomon* setelah di ubah dalam bentuk integer per 8 bit, Untuk *Reed-*

Solomon sampai dengan elemen 255 sesuai dengan panjang *codeword* yang diinginkan, menghasilkan data pada Tabel 4.2 berikut ini :

Tabel 4.2 variabel “*randomize*” dan variabel “*Reed-Solomon*”

Elemen	<i>Integer randomize</i>	Elemen	Blok <i>Reed-Solomon</i>
1	212	1	0
2	186	:	:
3	161	:	:
4	18	203	0
5	242	204	212
6	116	205	186
7	150	206	161
8	48	207	18
9	39	208	242
10	212	209	116
11	136	210	150
12	156	211	48
13	150	212	39
14	227	213	212
15	169	214	136
16	82	215	156
17	179	216	150
18	21	217	227
19	171	218	169
20	253	219	82
21	146	220	179
22	83	221	21
23	7	222	171
24	50	223	253
25	192	224	146
26	98	225	83
27	72	226	7
28	240	227	50
29	25	228	192
30	34	229	98
31	224	230	72
32	145	231	240
33	98	232	25
34	26	233	34
35	193	234	224
36	255	235	145

Setelah dari blok *Reed-Solomon* pengkodeaan data dilanjutkan ke pengkodean konvolusi dan *interleaver* untuk dilakukan proses pengkodean. Sebelum masuk ke blok modulasi QAM untuk dilakukan proses modulasi, bit-bit yang telah dikodekan ubah kedalam bentuk integer, dimana 2 bit untuk 1 nilai integer, berikut hasil pengkodean dan hasil perubahan ke bentuk integer, dari data tersebut akan di-*mapping* oleh blok QPSK, berikut tabel 4.3 yang diambil dari elemen pertama sampai ke elemen 20 :

Tabel 4.3 variabel "*interleaver*" dan variabel "*mapping*"

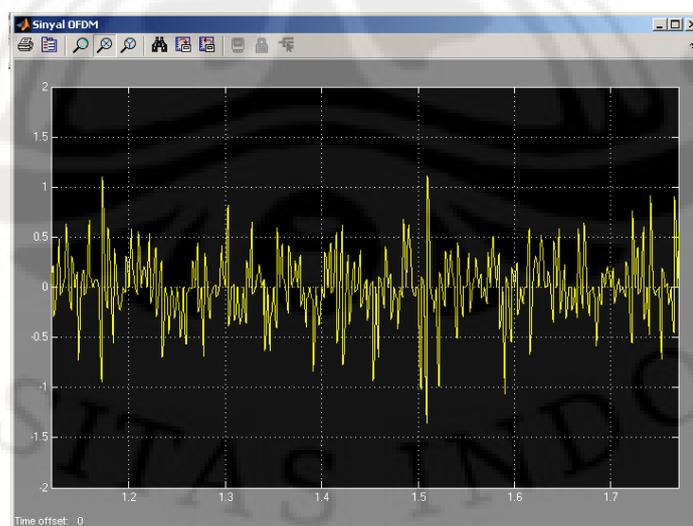
elemen	Setelah Konvolusi dan <i>interleaver</i>	<i>Mapping</i>
1	1	$0.707106781186548 - 0.707106781186548i$
2	2	$-0.707106781186548 + 0.707106781186548i$
3	1	$0.707106781186548 - 0.707106781186548i$
4	3	$-0.707106781186548 - 0.707106781186548i$
5	3	$-0.707106781186548 - 0.707106781186548i$
6	3	$-0.707106781186548 - 0.707106781186548i$
7	2	$-0.707106781186548 + 0.707106781186548i$
8	2	$-0.707106781186548 + 0.707106781186548i$
9	1	$0.707106781186548 - 0.707106781186548i$
10	0	$0.707106781186548 + 0.707106781186548i$
11	3	$-0.707106781186548 - 0.707106781186548i$
12	3	$-0.707106781186548 - 0.707106781186548i$
13	0	$0.707106781186548 + 0.707106781186548i$
14	1	$0.707106781186548 - 0.707106781186548i$
15	1	$0.707106781186548 - 0.707106781186548i$
16	3	$-0.707106781186548 - 0.707106781186548i$
17	1	$0.707106781186548 - 0.707106781186548i$
18	3	$-0.707106781186548 - 0.707106781186548i$
19	3	$-0.707106781186548 - 0.707106781186548i$
20	2	$-0.707106781186548 + 0.707106781186548i$

Setelah di-*mapping* oleh QPSK data-data tersebut diproses kembali sebelum masuk ke blok modulasi OFDM, yakni di konjugasikan dan dikalikan gain sebesar $\frac{1}{\sqrt{2}}$, sehingga data yang dihasilkan pada tabel 4.4 berikut ini :

Tabel 4.4 variabel “modulasi OFDM”

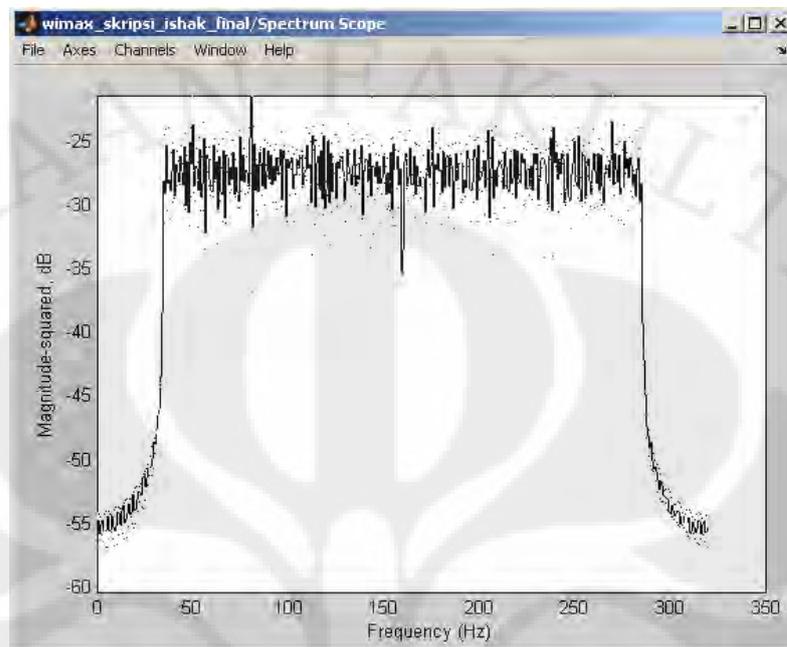
Elemen	Modulasi OFDM
1	-0.0117187500000000 - 0.0312500000000000i
2	0.0325728671592399 + 0.00859407520977960i
3	0.00202160949974735 - 0.0244516507347647i
4	-0.0650292468922605 + 0.0247708596682588i
5	0.0279761643135147 + 0.0133958539007940i
6	-0.0231036576467969 + 0.00612945835983571i
7	0.0353122489113658 - 0.00657664975491613i
8	0.00197257970781509 - 0.0115621673008002i
9	0.0404863141583750 - 0.00404292807809550i
10	-0.0207493391316643 - 0.0172181765721329i
11	-0.0161506391819665 + 0.0344047613858343i
12	-0.0672712622958545 - 0.00167503907399161i
13	0.00914242636026917 - 0.0159087031145790i
14	0.0732810307841988 + 0.0132505455192835i
15	0.000441937619738897 - 0.000820733210831053i
16	-0.0238576606116672 - 0.0104946620648038i
17	0.0321072788470175 + 0.00198264229366233i
18	-0.0196342627316235 - 0.00360702018637060i
19	-0.0364767764359875 + 0.000612389946975591i
20	0.00570276755279806 + 0.0257696674102890i

Gelombang sinyal setelah dilakukan proses modulasi OFDM tampak seperti pada Gambar 4.3.



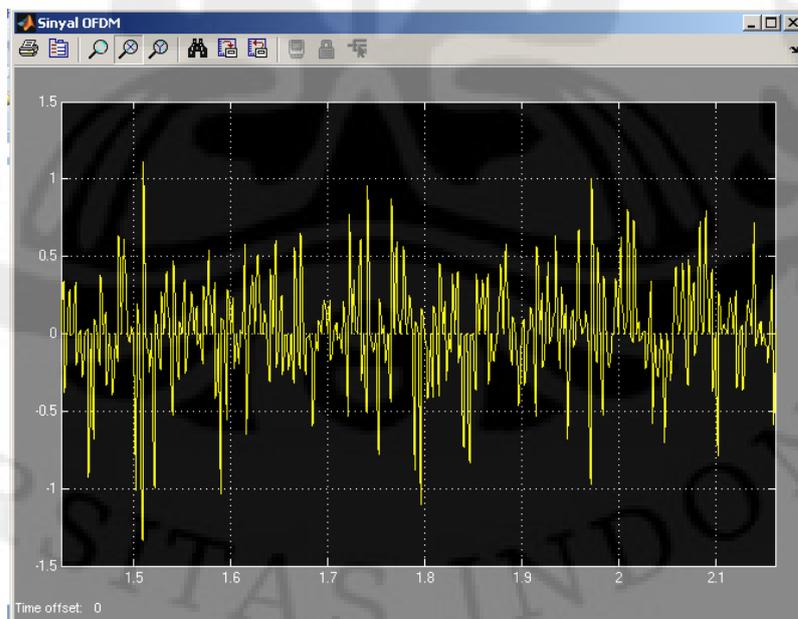
Gambar 4.3 Gelombang sinyal OFDM transmitter

Gambar 4.4 menunjukkan spektrum frekuensi dari sinyal OFDM yang dikeluarkan oleh *transmitter* sebagai berikut:

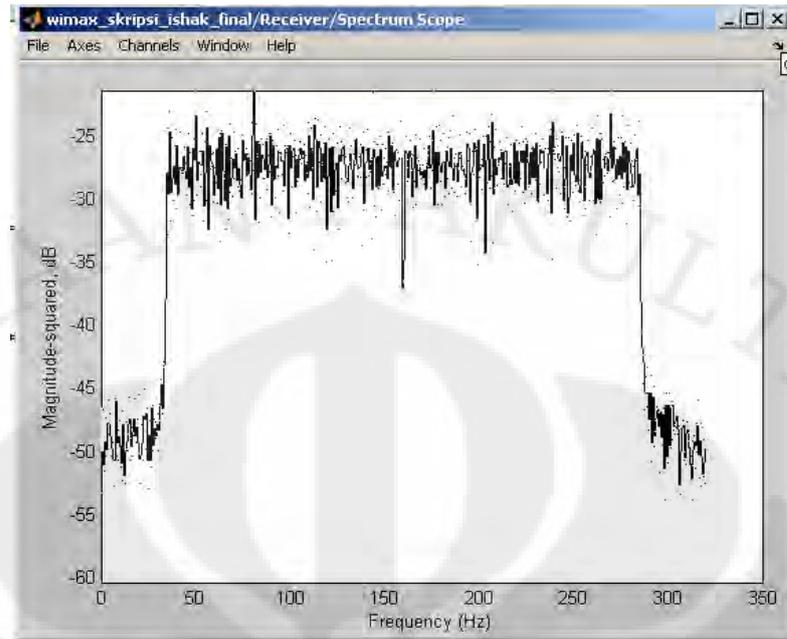


Gambar 4.4 spektrum frekuensi pada *transmitter*

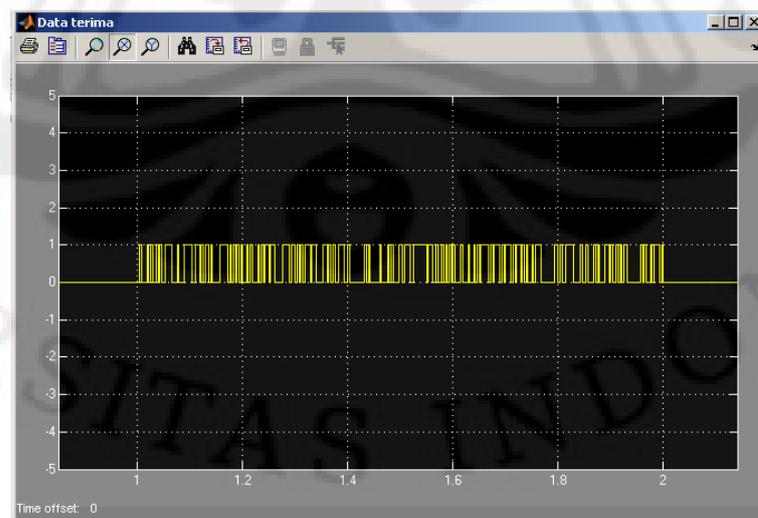
Sinyal tersebut dikirim ke *transmitter* dengan melewati kanal AWGN dimana nilai SNR-nya adalah 30, berikut sinyal dan spektrum yang dihasilkan terlihat pada Gambar 4.5 dan Gambar 4.6 berikut ini:



Gambar 4.5 Gelombang sinyal OFDM *receiver*

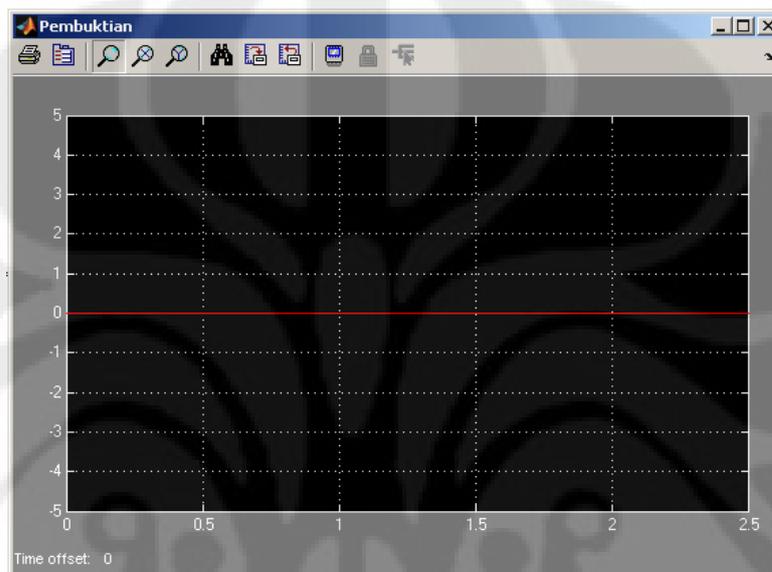
Gambar 4.6 spektrum frekuensi pada *receiver*

Spektrum sinyal setelah melewati kanal AWGN memiliki sedikit perbedaan dengan spektrum sinyal sebelum melewati kanal AWGN, pada spektrum transmiter terlihat lebih halus dibandingkan pada spektrum *receiver*, hal ini disebabkan noise yang diberikan oleh kanal AWGN tersebut. Setelah dilakukan proses pengiriman, maka selanjutnya dilakukan proses demodulasi terhadap sinyal untuk mendapatkan kembali sinyal informasi yang asli. Setelah melewati proses demodulasi maka didapatkan sinyal informasi yang asli. Data hasil demodulasi dapat dilihat pada Gambar 4.7.



Gambar 4.7 Data terima

Pengujian data yang telah diolah di *receiver* dilakukan dengan membandingkannya dengan data yang dikirim dari *transmitter*. Perbandingan data dilakukan dengan melakukan operasi logical XOR. Dengan membandingkan sinyal yang dikirim oleh *transmitter* dengan sinyal yang diterima di *receiver* yang telah diproses, dapat diketahui bahwa data yang diterima sama dengan yang dikirim seperti terlihat pada *scope* pembuktian pada Gambar 4.8.



Gambar 4.8 Perbandingan data

Grafik pada *scope* pembuktian selalu bernilai nol karena dilakukan proses XOR terhadap data yang dikirim dengan data yang diterima. Apabila data yang dikirim dengan data yang diterima bernilai sama maka hasil XOR sama dengan nol. Oleh karena data yang dikirim dengan data yang diterima selalu bernilai nol maka hasil XOR juga akan selalu bernilai nol. Hal ini menunjukkan bahwa rancang bangun model *baseband* WiMAX yang telah diuji sudah yang aturan secara teoritis.

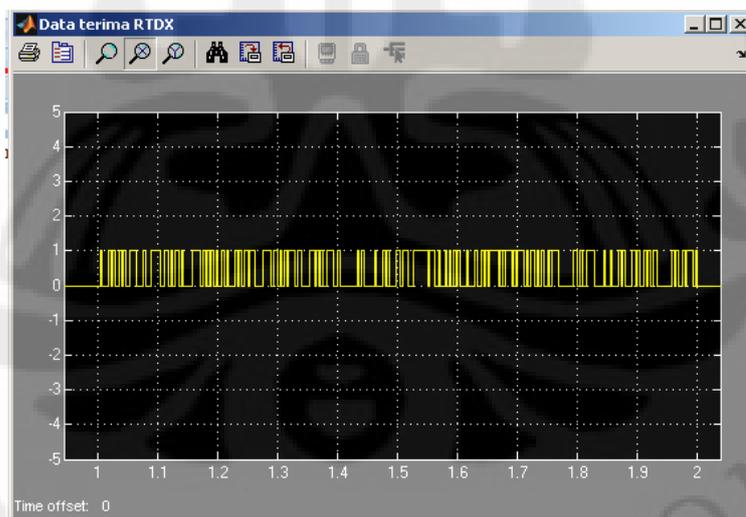
4.2. Pengambilan dan Analisis Data pada DSK TMS320C6713

Pada ujicoba dengan menggunakan DSK TMS320C6713, pengujian dilakukan dengan menggunakan dua cara yaitu dengan menggunakan *Real Time Data Exchange* (RTDX) dan *storage oscilloscope*.

4.2.1 Real Time Data Exchange (RTDX)

Ujicoba dilakukan dengan menggunakan RTDX karena *output* berupa bit-bit. Apabila *output* yang berupa bit-bit dilihat dengan menggunakan *storage oscilloscope* maka *output* yang terlihat akan mengalami gangguan disebabkan oleh adanya proses sampling sinyal *digital* menjadi sinyal *analog* sedangkan jika menggunakan RTDX maka *output* yang berupa bit-bit akan dikirim ke komputer dan data diolah secara *digital* sehingga tidak mengganggu atau merusak *output*.

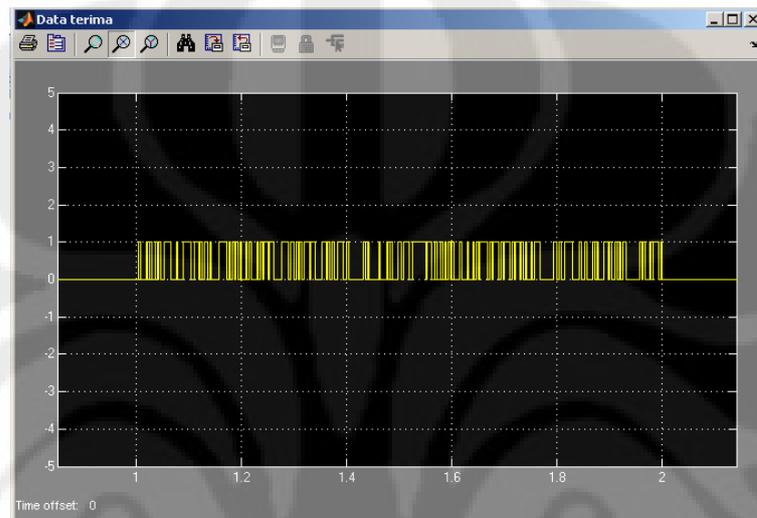
Pengujian kebenaran data *output* dari *receiver* dilakukan dengan membandingkan data *output* RTDX dengan data *output* didapat dari Simulink. *Input* pada pengujian ini sama dengan *input* pada pengujian simulink yakni data *input* yang dimasukkan. *Output* dari rangkaian *baseband* WiMAX dengan menggunakan RTDX didapatkan data seperti Gambar 4.9.



Gambar 4.9 *Output* rangkaian *baseband* WiMAX dengan RTDX

Output dari rangkaian *baseband* WiMAX yang didapat dari Simulink dapat dilihat pada Gambar 4.7. Dari kedua Gambar tersebut dapat diketahui bahwa data *output* yang didapat dengan menggunakan RTDX pada Gambar 4.9

sama dengan data *output* yang berasal dari simulink pada Gambar 4.7. Data *output* yang berasal dari Simulink sama dengan data yang dikirim dari *transmitter*. Oleh karena itu, data *output* yang didapat dari RTDX sama dengan data yang dikirim dari *transmitter* pada Gambar 4.2. Hal ini menunjukkan bahwa rancang bangun model rangkaian *baseband* WiMAX yang diuji sudah berfungsi dengan semestinya.



Gambar 4.10 *Output* rangkaian *baseband* WiMAX

BAB V KESIMPULAN

Berdasarkan hasil analisis dapat disimpulkan :

1. Aliran data dari rangkaian *baseband* WiMAX yang diukur pada *workspace* dari MATLAB telah memberikan hasil yang sesuai dengan teori dan tujuan.
2. Rangkaian *baseband* WiMAX dapat dibangun dengan menggunakan DSK TMS320C6713 dan telah berhasil diimplementasikan serta dapat bekerja sesuai dengan yang direncanakan.
3. Pada implementasi rangkaian *baseband* WiMAX ini pada DSK TMS320C6713 dengan menggunakan simulink digunakan *noise* dengan nilai SNRnya 30 dB memberikan hasil di *receiver* yang tidak berbeda jauh dengan hasil masukan data pada bagian *transmitter*.
4. Hasil perbandingan data antara data yang dikirim dengan data yang diterima memberikan hasil yang sesuai dengan tujuan dari skripsi ini.

DAFTAR ACUAN

- [1] Andrews ,Jeffrey G., Ghosh, Arunabha,dan Muhamed, Rias., "*Fundamental of WIMAX Understanding Broadband Wireless Network*", Prentice Hall *First Edition*, 2007.
- [2] A. M. Michelson, and A. H. Levesque: "*Error Control Techniques for Digital Communications*," Wiley-Interscience Publications, 1985.
- [3] Giovanni, P., "*Functional Specifications of the Adaptive Modem IEEE 802.16*", Multichannel Adaptive Information System, 2004.
- [4] IEEE, "802.16TM *IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems*", IEEE Standards, 2004.
- [5] Langton, Charan., "*Orthogonal Frequency Division Multiplexing Tutorial*", 2004.
- [6] Pietikäinen, Kari., "*Orthogonal Frequency Division Multiplexing*", Postgraduate Course in Radio Communications, 2004.
- [7] S. B. Wicker: "*Error Control Systems for Digital Communication and Storage*," School of Electrical and Computer Engineering, Georgia Institute of Technology, Prentice Hall, 1995.
- [8] WiMAX Forum: "*Mobile WiMAX. Part I: A Technical Overview and Performance Evaluation*", Agustus 2006.

DAFTAR PUSTAKA

Matlab help, *signal processing toolbox*, mathworks inc.

Matlab help, *communications blockset*, mathworks inc.

Ifeachor, E.C. dan Jervis, B.K., "*Digital Signal Processing*", Prentice Hall *Second Edition*, 2001

Diponegoro, Arman Djohan., Suryanegara, Muhammad, "*Rancang Bangun Rangkaian OFDM dengan Menggunakan DSK TMS C6713 Berbasis Simulink*". Proposal Grant Riset ICT. Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia., 2007

S. Haykin, *Communication Systems*, 4th ed (New York: John Wiley & Sons, 2001)

LAMPIRAN 1

Tabel Nilai Lengkap Variabel Data Input, Randomize, dan Reed-Solomon

Elemen	Variabel "data input"		Variabel "randomize" dan variabel "Reed-Solomon"	
	<i>Input-an data</i>	Bit	<i>Integer Randomize</i>	<i>Reed-Solomon</i>
1	69	10001010	212	0
2	41	00101001	186	0
3	196	11000100	161	0
4	121	01111001	18	0
5	173	10101101	242	0
6	15	00001111	116	0
7	85	01010101	150	0
8	40	00101000	48	0
9	173	10101101	39	0
10	135	10000111	212	0
11	181	10110101	136	0
12	118	01110110	156	0
13	26	00011010	150	0
14	156	10011100	227	0
15	128	10000000	169	0
16	80	10100000	82	0
17	69	01000101	179	0
18	27	00011011	21	0
19	159	10011111	171	0
20	217	11011001	253	0
21	42	00101010	146	0
22	136	10001000	83	0
23	149	10010101	7	0
24	235	11101011	50	0
25	174	10101110	192	0
26	181	10110101	98	0
27	46	00101110	72	0
28	3	00000011	240	0
29	79	01001111	25	0
30	9	00001001	34	0
31	20	00010100	224	0
32	105	01101001	145	0
33	88	01011000	98	0
34	10	00001010	26	0
35	93	01011101	193	0

(Lanjutan)

36			255	0
37				0
38				0
39				0
40				0
41				0
42				0
43				0
44				0
45				0
46				0
47				0
48				0
49				0
50				0
51				0
52				0
53				0
54				0
55				0
56				0
57				0
58				0
59				0
60				0
61				0
62				0
63				0
64				0
65				0
66				0
67				0
68				0
69				0
70				0
71				0
72				0
73				0
74				0
75				0
76				0
77				0
78				0

(Lanjutan)

79				0
80				0
81				0
82				0
83				0
84				0
85				0
86				0
87				0
88				0
89				0
90				0
91				0
92				0
93				0
94				0
95				0
96				0
97				0
98				0
99				0
100				0
101				0
102				0
103				0
104				0
105				0
106				0
107				0
108				0
109				0
110				0
111				0
112				0
113				0
114				0
115				0
116				0
117				0
118				0
119				0
120				0
121				0

(Lanjutan)

122				0
123				0
124				0
125				0
126				0
127				0
128				0
129				0
130				0
131				0
132				0
133				0
134				0
135				0
136				0
137				0
138				0
139				0
140				0
141				0
142				0
143				0
144				0
145				0
146				0
147				0
148				0
149				0
150				0
151				0
152				0
153				0
154				0
155				0
156				0
157				0
158				0
159				0
160				0
161				0
162				0
163				0
164				0

(Lanjutan)

165				0
166				0
167				0
168				0
169				0
170				0
171				0
172				0
173				0
174				0
175				0
176				0
177				0
178				0
179				0
180				0
181				0
182				0
183				0
184				0
185				0
186				0
187				0
188				0
189				0
190				0
191				0
192				0
193				0
194				0
195				0
196				0
197				0
198				0
199				0
200				0
201				0
202				0
203				0
204				212
205				186
206				161
207				18

(Lanjutan)

208				242
209				116
210				150
211				48
212				39
213				212
214				136
215				156
216				150
217				227
218				169
219				82
220				179
221				21
222				171
223				253
224				146
225				83
226				7
227				50
228				192
229				98
230				72
231				240
232				25
233				34
234				224
235				145
236				98
237				26
238				193
239				255
240				35
241				190
242				76
243				143
244				84
245				185
246				166
247				216
248				214
249				47
250				6

(Lanjutan)

251				159
252				3
253				8
254				241
255				209



LAMPIRAN 2

Tabel Nilai Lengkap Variabel *Interleaver*, *mapping* dan modulasi OFDM

Elemen	variabel " <i>interleaver</i> " dan variabel " <i>mapping</i> "		Variabel "Modulasi OFDM"
	setelah <i>interleaver</i>	<i>Mapping</i>	Modulasi OFDM
1	1	0.707106781186548 - 0.707106781186548i	-0.0117187500000000 - 0.0312500000000000i
2	2	-0.707106781186548 + 0.707106781186548i	0.0325728671592399 + 0.00859407520977960i
3	1	0.707106781186548 - 0.707106781186548i	0.00202160949974735 - 0.0244516507347647i
4	3	-0.707106781186548 - 0.707106781186548i	-0.0650292468922605 + 0.0247708596682588i
5	3	-0.707106781186548 - 0.707106781186548i	0.0279761643135147 + 0.0133958539007940i
6	3	-0.707106781186548 - 0.707106781186548i	-0.0231036576467969 + 0.00612945835983571i
7	2	-0.707106781186548 + 0.707106781186548i	0.0353122489113658 - 0.00657664975491613i
8	2	-0.707106781186548 + 0.707106781186548i	0.00197257970781509 - 0.0115621673008002i
9	1	0.707106781186548 - 0.707106781186548i	0.0404863141583750 - 0.00404292807809550i
10	0	0.707106781186548 + 0.707106781186548i	-0.0207493391316643 - 0.0172181765721329i
11	3	-0.707106781186548 - 0.707106781186548i	-0.0161506391819665 + 0.0344047613858343i
12	3	-0.707106781186548 - 0.707106781186548i	-0.0672712622958545 - 0.00167503907399161i
13	0	0.707106781186548 + 0.707106781186548i	0.00914242636026917 - 0.0159087031145790i
14	1	0.707106781186548 - 0.707106781186548i	0.0732810307841988 + 0.0132505455192835i
15	1	0.707106781186548 - 0.707106781186548i	0.000441937619738897 - 0.000820733210831053i
16	3	-0.707106781186548 - 0.707106781186548i	-0.0238576606116672 - 0.0104946620648038i
17	1	0.707106781186548 - 0.707106781186548i	0.0321072788470175 + 0.00198264229366233i
18	3	-0.707106781186548 - 0.707106781186548i	-0.0196342627316235 - 0.00360702018637060i
19	3	-0.707106781186548 - 0.707106781186548i	-0.0364767764359875 + 0.000612389946975591i
20	2	-0.707106781186548 + 0.707106781186548i	0.00570276755279806 + 0.0257696674102890i
21	0	0.707106781186548 + 0.707106781186548i	-0.0208319691473926 - 0.00855704645880452i
22	3	-0.707106781186548 - 0.707106781186548i	0.0438159801905249 - 0.0310811181216370i
23	3	-0.707106781186548 - 0.707106781186548i	0.0226806665511224 - 0.0106533345997175i
24	2	-0.707106781186548 + 0.707106781186548i	-0.00531169350267099 + 0.0293557066118378i
25	3	-0.707106781186548 - 0.707106781186548i	-0.00892568291876294 + 0.0550476622745183i
26	2	-0.707106781186548 + 0.707106781186548i	-0.0195153891116763 - 0.0422253657322917i
27	1	0.707106781186548 - 0.707106781186548i	-0.0130467787087979 - 0.0211525712026254i
28	2	-0.707106781186548 + 0.707106781186548i	0.00184117879681948 - 0.0198402498448371i
29	1	0.707106781186548 - 0.707106781186548i	0.00801384801616984 + 0.00605439334019892i
30	3	-0.707106781186548 - 0.707106781186548i	0.0154157219899932 + 0.00841873383643580i
31	0	0.707106781186548 + 0.707106781186548i	0.00577983410868939 + 0.0373745916792238i
32	0	0.707106781186548 + 0.707106781186548i	-0.00470224636205865 + 0.0147342001366505i
33	1	0.707106781186548 - 0.707106781186548i	-0.00619447827198011 - 0.0276213586400995i
34	3	-0.707106781186548 - 0.707106781186548i	0.00961163134449542 - 0.00907031649693604i
35	2	-0.707106781186548 + 0.707106781186548i	-0.00181408284221231 - 0.0168415029552300i

(Lanjutan)

36	0	$0.707106781186548 + 0.707106781186548i$	$-0.0305983475629204 - 0.0206216201588977i$
37	3	$-0.707106781186548 - 0.707106781186548i$	$0.00915358851748701 + 0.0438613561917588i$
38	0	$0.707106781186548 + 0.707106781186548i$	$-0.0188027659754011 - 0.00319814619241103i$
39	3	$-0.707106781186548 - 0.707106781186548i$	$0.0497493188603527 - 0.0209409112554319i$
40	1	$0.707106781186548 - 0.707106781186548i$	$0.00754356011394552 + 0.0402734134381849i$
41	0	$0.707106781186548 + 0.707106781186548i$	$-0.0150695229372859 + 0.00762417136892552i$
42	3	$-0.707106781186548 - 0.707106781186548i$	$0.0112692783801208 - 0.0216849588758699i$
43	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0300145663829420 - 0.0325971212104766i$
44	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0339670704370905 + 0.0118030035287698i$
45	1	$0.707106781186548 - 0.707106781186548i$	$0.0112999728814979 + 0.0194051431954110i$
46	3	$-0.707106781186548 - 0.707106781186548i$	$0.0375106671900964 - 0.0524289679880764i$
47	1	$0.707106781186548 - 0.707106781186548i$	$0.0337620291201318 + 0.0386523663120121i$
48	2	$-0.707106781186548 + 0.707106781186548i$	$-0.0260939627733302 + 0.0432642583706929i$
49	2	$-0.707106781186548 + 0.707106781186548i$	$-0.000410921010391975 - 0.00703549856492321i$
50	0	$0.707106781186548 + 0.707106781186548i$	$-0.0335727708223314 - 0.0336914863021774i$
51	0	$0.707106781186548 + 0.707106781186548i$	$-0.0243674797251124 + 0.0142360075162668i$
52	0	$0.707106781186548 + 0.707106781186548i$	$0.0462637619923915 - 0.0388302965754542i$
53	3	$-0.707106781186548 - 0.707106781186548i$	$0.00546763958631528 + 0.00329097640590380i$
54	0	$0.707106781186548 + 0.707106781186548i$	$0.0449683724881510 + 0.0489120449403710i$
55	1	$0.707106781186548 - 0.707106781186548i$	$-0.0628060504049639 - 0.0476639853055036i$
56	0	$0.707106781186548 + 0.707106781186548i$	$-0.0265617953455235 + 0.00546128116147387i$
57	0	$0.707106781186548 + 0.707106781186548i$	$0.0143551822653153 + 0.0445673152216232i$
58	2	$-0.707106781186548 + 0.707106781186548i$	$0.0396905026623798 + 0.0159199710618468i$
59	3	$-0.707106781186548 - 0.707106781186548i$	$0.0198677801883021 - 0.0584279699047118i$
60	0	$0.707106781186548 + 0.707106781186548i$	$-0.0470547975460277 + 0.0243433572068663i$
61	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0119252968943894 - 0.00903589394771074i$
62	1	$0.707106781186548 - 0.707106781186548i$	$0.0292107172788010 - 0.0242811697363047i$
63	2	$-0.707106781186548 + 0.707106781186548i$	$-0.0142177163952623 - 0.0171291102781135i$
64	2	$-0.707106781186548 + 0.707106781186548i$	$0.0180753373675813 + 0.0407406776204975i$
65	1	$0.707106781186548 - 0.707106781186548i$	$-0.0195312500000000 + 0.0390625000000000i$
66	2	$-0.707106781186548 + 0.707106781186548i$	$0.0181492980119622 - 0.0113372400024943i$
67	2	$-0.707106781186548 + 0.707106781186548i$	$-0.0341719738165507 - 0.0212773562904042i$
68	1	$0.707106781186548 - 0.707106781186548i$	$0.0224656333387481 - 0.00429860431342278i$
69	2	$-0.707106781186548 + 0.707106781186548i$	$0.0276870027363871 - 0.00680086108961178i$
70	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0334602536924637 - 0.0204220347063691i$
71	1	$0.707106781186548 - 0.707106781186548i$	$0.0122033465441823 + 0.00420720693661334i$
72	3	$-0.707106781186548 - 0.707106781186548i$	$0.0111278012989877 + 0.0285926637253594i$
73	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0342725604380075 + 0.00556467953203733i$
74	3	$-0.707106781186548 - 0.707106781186548i$	$-0.00402492854806936 - 0.0227910880757505i$
75	2	$-0.707106781186548 + 0.707106781186548i$	$0.00795288630779793 + 0.0619733812241416i$
76	3	$-0.707106781186548 - 0.707106781186548i$	$0.0303890072152335 - 0.0401266621883113i$
77	1	$0.707106781186548 - 0.707106781186548i$	$-0.00181890150846191 - 0.0447458724492469i$
78	0	$0.707106781186548 + 0.707106781186548i$	$-0.000281676366852262 + 0.0108360047659616i$

(Lanjutan)

79	0	$0.707106781186548 + 0.707106781186548i$	$0.00547305164344376 + 0.0256014308371835i$
80	2	$-0.707106781186548 + 0.707106781186548i$	$-0.0307436970938154 + 0.00647087230519889i$
81	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0308199007174489 - 0.0193427178379003i$
82	0	$0.707106781186548 + 0.707106781186548i$	$0.0437615772608618 + 0.0131381189561110i$
83	3	$-0.707106781186548 - 0.707106781186548i$	$-0.00453264467762666 + 0.0101385452974078i$
84	0	$0.707106781186548 + 0.707106781186548i$	$0.0104772338990457 - 0.0115516715040534i$
85	3	$-0.707106781186548 - 0.707106781186548i$	$-0.000469417377107175 + 0.0271813511244554i$
86	3	$-0.707106781186548 - 0.707106781186548i$	$0.0108044927897512 - 0.0550847079208999i$
87	0	$0.707106781186548 + 0.707106781186548i$	$0.00324963970225402 + 0.0214444141563953i$
88	1	$0.707106781186548 - 0.707106781186548i$	$-0.0338313281284228 - 0.0350303661764300i$
89	2	$-0.707106781186548 + 0.707106781186548i$	$-0.0284277075001410 + 0.0618647550783356i$
90	1	$0.707106781186548 - 0.707106781186548i$	$0.0306464241781056 + 0.000452717432923452i$
91	2	$-0.707106781186548 + 0.707106781186548i$	$0.0241268413771821 - 0.0167688850345879i$
92	2	$-0.707106781186548 + 0.707106781186548i$	$0.00789922628330675 - 0.00569552721421724i$
93	2	$-0.707106781186548 + 0.707106781186548i$	$-0.00997311032781134 + 0.000706185846519145i$
94	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0243455684776926 - 0.0172313348466811i$
95	3	$-0.707106781186548 - 0.707106781186548i$	$-0.00303694330082788 - 0.0104137443979589i$
96	0	$0.707106781186548 + 0.707106781186548i$	$0.0221594175707498 + 0.0596816085899114i$
97	3	$-0.707106781186548 - 0.707106781186548i$	$-0.00619447827198010 - 0.0321978151840597i$
98	2	$-0.707106781186548 + 0.707106781186548i$	$5.12376910663664e-05 - 0.00246022784368813i$
99	3	$-0.707106781186548 - 0.707106781186548i$	$0.0229879938632457 + 0.00971139698836725i$
100	1	$0.707106781186548 - 0.707106781186548i$	$-0.0402982533803365 + 0.00348433677460667i$
101	0	$0.707106781186548 + 0.707106781186548i$	$0.000154093032046802 - 0.0331892762343566i$
102	0	$0.707106781186548 + 0.707106781186548i$	$0.0236129539015128 + 0.0122819366877200i$
103	2	$-0.707106781186548 + 0.707106781186548i$	$-0.00941880363696931 - 0.0165618850485308i$
104	2	$-0.707106781186548 + 0.707106781186548i$	$-0.00523945219807535 + 0.0699082646510470i$
105	0	$0.707106781186548 + 0.707106781186548i$	$0.0367745129306610 - 0.0430532348314291i$
106	1	$0.707106781186548 - 0.707106781186548i$	$-0.0330650839482815 + 0.0183754875763407i$
107	3	$-0.707106781186548 - 0.707106781186548i$	$0.00406576484674180 - 0.0270497473715491i$
108	0	$0.707106781186548 + 0.707106781186548i$	$0.00861836731252903 - 0.000855106518309591i$
109	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0350176875229908 - 0.0284580995143590i$
110	1	$0.707106781186548 - 0.707106781186548i$	$0.0254932172249875 + 0.0371346035309255i$
111	2	$-0.707106781186548 + 0.707106781186548i$	$0.00434962055681975 + 0.0246437689242817i$
112	0	$0.707106781186548 + 0.707106781186548i$	$-0.0314449475809898 - 0.00878877392586321i$
113	3	$-0.707106781186548 - 0.707106781186548i$	$0.0701777277706651 - 0.0104225728043723i$
114	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0346953605521318 - 0.00905229784210822i$
115	2	$-0.707106781186548 + 0.707106781186548i$	$-0.00301427113174752 - 0.00633623874739276i$
116	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0261538181767796 + 0.00117010156887219i$
117	3	$-0.707106781186548 - 0.707106781186548i$	$0.0408817962446036 + 0.000449500994591656i$
118	2	$-0.707106781186548 + 0.707106781186548i$	$-0.0409376176109563 + 0.0306387687087942i$
119	3	$-0.707106781186548 - 0.707106781186548i$	$0.0118494710950223 - 0.0377969268307381i$
120	0	$0.707106781186548 + 0.707106781186548i$	$0.00395889553434434 + 0.0276409691690345i$
121	2	$-0.707106781186548 + 0.707106781186548i$	$-0.000316142477003328 - 0.0182681125995553i$

(Lanjutan)

122	1	$0.707106781186548 - 0.707106781186548i$	$0.0328999283581632 + 0.00629775605861913i$
123	2	$-0.707106781186548 + 0.707106781186548i$	$-0.00373813383811970 + 0.0145700430871777i$
124	3	$-0.707106781186548 - 0.707106781186548i$	$-0.000234016214158909 - 0.0123117252284395i$
125	0	$0.707106781186548 + 0.707106781186548i$	$-0.0524772657746579 - 0.0310136407030676i$
126	3	$-0.707106781186548 - 0.707106781186548i$	$-0.00677176781671756 + 0.0352182157192724i$
127	0	$0.707106781186548 + 0.707106781186548i$	$0.0209042416899090 + 0.0169587269476219i$
128	1	$0.707106781186548 - 0.707106781186548i$	$0.0213477713527990 - 0.0318801769557831i$
129	2	$-0.707106781186548 + 0.707106781186548i$	$0.0351562500000000 - 1.04083408558608e-17i$
130	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0224845765566578 - 0.0214182709930081i$
131	1	$0.707106781186548 - 0.707106781186548i$	$-0.0363268673771462 + 0.0421780049810968i$
132	1	$0.707106781186548 - 0.707106781186548i$	$0.0115199987868683 + 0.0431858218498834i$
133	2	$-0.707106781186548 + 0.707106781186548i$	$-0.00367852419051632 - 0.0516163383198948i$
134	3	$-0.707106781186548 - 0.707106781186548i$	$-0.00183725445870815 - 0.0606621522597660i$
135	2	$-0.707106781186548 + 0.707106781186548i$	$0.0142832822598689 + 0.0628184203371839i$
136	2	$-0.707106781186548 + 0.707106781186548i$	$0.0347811153430367 - 0.00253076449821675i$
137	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0407904402776504 + 0.00370336513258656i$
138	1	$0.707106781186548 - 0.707106781186548i$	$-0.00549391237858152 - 0.0199864948419628i$
139	2	$-0.707106781186548 + 0.707106781186548i$	$-0.0171264459749288 + 0.0221567229218855i$
140	2	$-0.707106781186548 + 0.707106781186548i$	$0.0211162837550593 + 0.0154893125204990i$
141	0	$0.707106781186548 + 0.707106781186548i$	$0.00832830173802845 - 0.0276795186796245i$
142	3	$-0.707106781186548 - 0.707106781186548i$	$0.0146411994433594 - 0.0220407775454648i$
143	0	$0.707106781186548 + 0.707106781186548i$	$0.0156315973078382 + 0.0173966650944178i$
144	0	$0.707106781186548 + 0.707106781186548i$	$-0.0295265251063566 + 0.0348181250785651i$
145	1	$0.707106781186548 - 0.707106781186548i$	$-0.0256351919349379 - 0.0677186857497021i$
146	1	$0.707106781186548 - 0.707106781186548i$	$0.00326572735097889 + 0.0661533390790085i$
147	1	$0.707106781186548 - 0.707106781186548i$	$-0.0248330515847532 + 0.0149095307218500i$
148	1	$0.707106781186548 - 0.707106781186548i$	$0.0525838661055488 - 0.0735197454519614i$
149	1	$0.707106781186548 - 0.707106781186548i$	$0.0291833999881049 + 0.0112582945062777i$
150	1	$0.707106781186548 - 0.707106781186548i$	$-0.0147871441854425 + 0.00155721581268733i$
151	0	$0.707106781186548 + 0.707106781186548i$	$-0.0119156016117804 + 0.0437284062512478i$
152	0	$0.707106781186548 + 0.707106781186548i$	$-0.0280437781626325 + 0.0170876076633052i$
153	1	$0.707106781186548 - 0.707106781186548i$	$-0.00721703298274562 - 0.0338983905464984i$
154	0	$0.707106781186548 + 0.707106781186548i$	$0.0398455987155970 - 0.0304755489129299i$
155	2	$-0.707106781186548 + 0.707106781186548i$	$-0.0592902696842448 - 0.00834921028080788i$
156	1	$0.707106781186548 - 0.707106781186548i$	$0.0393914165597751 + 0.0227981028520701i$
157	1	$0.707106781186548 - 0.707106781186548i$	$0.0299653977956273 + 0.0193087162265031i$
158	3	$-0.707106781186548 - 0.707106781186548i$	$-0.00872458451271759 + 0.0176027812352539i$
159	3	$-0.707106781186548 - 0.707106781186548i$	$0.0132460124687015 - 0.0136053312775773i$
160	1	$0.707106781186548 - 0.707106781186548i$	$-0.0577279491852135 - 0.0441399651014338i$
161	2	$-0.707106781186548 + 0.707106781186548i$	$-0.0172430217280199 + 0.0276213586400995i$
162	3	$-0.707106781186548 - 0.707106781186548i$	$0.0896593995523709 + 0.0458511965595149i$
163	1	$0.707106781186548 - 0.707106781186548i$	$-0.0498871083288369 - 0.0613902427498780i$
164	2	$-0.707106781186548 + 0.707106781186548i$	$0.00121695300539624 - 0.0142473608904179i$

(Lanjutan)

165	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0302373636410896 + 0.00587020952517906i$
166	2	$-0.707106781186548 + 0.707106781186548i$	$0.0412847631154004 + 0.0579917217385872i$
167	3	$-0.707106781186548 - 0.707106781186548i$	$-0.00225510974206518 - 0.0127440359602816i$
168	1	$0.707106781186548 - 0.707106781186548i$	$0.00327979361368752 - 0.0222344590146730i$
169	2	$-0.707106781186548 + 0.707106781186548i$	$0.0190397917951522 + 0.0166331060994015i$
170	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0209551816453699 - 0.0153978905292827i$
171	1	$0.707106781186548 - 0.707106781186548i$	$-0.0274857031313593 + 0.0100688674828008i$
172	0	$0.707106781186548 + 0.707106781186548i$	$0.00981189069061117 - 0.0610161921512052i$
173	2	$-0.707106781186548 + 0.707106781186548i$	$-0.00527785738389359 + 0.0491299065814636i$
174	0	$0.707106781186548 + 0.707106781186548i$	$0.00147702307500463 + 0.0273262169091452i$
175	2	$-0.707106781186548 + 0.707106781186548i$	$0.0155006144398974 - 0.0159777186383497i$
176	1	$0.707106781186548 - 0.707106781186548i$	$0.0231001549731463 + 0.0107843659138115i$
177	2	$-0.707106781186548 + 0.707106781186548i$	$0.00309174718623274 - 0.0274505448911166i$
178	0	$0.707106781186548 + 0.707106781186548i$	$-0.0371509766549456 + 0.0265202192915100i$
179	3	$-0.707106781186548 - 0.707106781186548i$	$0.0207984439390045 - 0.0503379995311385i$
180	0	$0.707106781186548 + 0.707106781186548i$	$-0.00760180700734627 + 0.0210934495338423i$
181	1	$0.707106781186548 - 0.707106781186548i$	$-0.0350881171219201 + 0.0251205785943688i$
182	3	$-0.707106781186548 - 0.707106781186548i$	$0.0233049172293212 - 0.00731434150153755i$
183	0	$0.707106781186548 + 0.707106781186548i$	$-0.0283983781467440 - 0.00625570150788697i$
184	2	$-0.707106781186548 + 0.707106781186548i$	$0.0337034604391385 + 0.0174757222360824i$
185	3	$-0.707106781186548 - 0.707106781186548i$	$0.0655749117657619 - 0.0344665869496431i$
186	3	$-0.707106781186548 - 0.707106781186548i$	$-0.0309285404293648 - 0.00298734939980709i$
187	3	$-0.707106781186548 - 0.707106781186548i$	$-0.00215815829073088 + 0.0431129164392052i$
188	0	$0.707106781186548 + 0.707106781186548i$	$-0.0874218440930509 - 0.0187453085001449i$
189	1	$0.707106781186548 - 0.707106781186548i$	$0.0116238463884448 - 0.00151107512579476i$
190	1	$0.707106781186548 - 0.707106781186548i$	$0.0508680008188885 + 0.00402329217497322i$
191	3	$-0.707106781186548 - 0.707106781186548i$	$0.00334682431266830 - 0.0231244018873508i$
192	1	$0.707106781186548 - 0.707106781186548i$	$0.0139512727551922 + 0.0108797165045943i$