

**SIMULASI PEMODELAN MIMO SPATIAL
MULTIPLEXING UNTUK APLIKASI WiMAX (802.16d)**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana

Pada Jurusan Teknik Elektro Fakultas Teknologi Industri

Universitas Islam Indonesia



Disusun oleh:

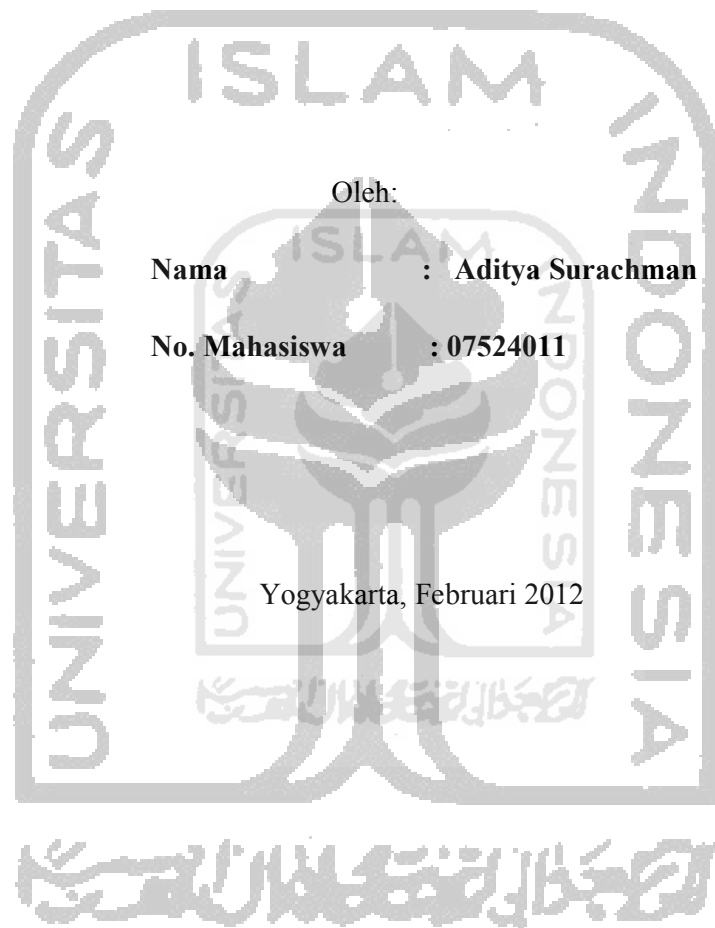
Nama : Aditya Surachman

No. Mahasiswa : 07524011

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2012

LEMBAR PENGESAHAN PEMBIMBING
SIMULASI PEMODELAN MIMO SPATIAL MULTIPLEXING
UNTUK APLIKASI WiMAX (802.1d)
TUGAS AKHIR



Pembimbing I

Pembimbing II

Tito Yuwono, ST., M. Sc

Dr., Eng, Hendra Setiawan, ST., MT

LEMBAR PENGESAHAN PENGUJI
SIMULASI PEMODELAN MIMO SPATIAL MULTIPLEXING
UNTUK APLIKASI WiMAX (802.16d)
TUGAS AKHIR

Oleh:

Nama : Aditya Surachman

No. Mahasiswa : 07524011

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana pada Jurusan Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Maret 2012

Tim Penguji

Ketua

Tito Yuwono, ST., M.Sc

Anggota I

Dr., Eng. Hendra Setiawan, ST., MT

Anggota II

Medilla Kusriyanto S.T., M.Eng

Mengetahui,

Ketua Jurusan Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia

Tito Yuwono, ST., M. Sc



*Kupersembahkan karya pertamaku ini
Untuk orang-orang yang kukasihi dan kucintai....
Ayahanda, Ibunda tercinta
Kakak dan kedua adikku yang tersayang
Semoga senantiasa diberikan rahmat,
hidayah serta maghfirah dari ALLAH SWT... Amin*

ABSTAKSI

Perkembangan Teknologi WiMAX-OFDM yang mengarah kepada pemenuhan akan kebutuhan kapasitas yang tinggi, serta penggunaan *bandwidth* yang efisien. Maka digunakanlah teknologi *Multiple Input Multiple Output* (MIMO) yang sangat mendukung dalam hal tersebut. Penggunaan teknik MIMO telah diketahui mampu meningkatkan performansi sistem. Tugas akhir ini akan dirancang dan disimulasikan pemodelan sistem MIMO-*Spatial Multiplexing* yang memenuhi spesifikasi standar WiMAX 802.16d. Sistem yang digunakan dalam simulasi mengacu pada parameter-parameter yang ada pada *fixed*-WiMAX tipe *Wireless MAN-OFDM*, dengan jumlah antena MIMO 2Tx-2Rx, dengan model kanal yang digunakan adalah AWGN. Dari hasil simulasi didapat BER dari beberapa masukan SNR mulai dari 0-60 dB. Ketika SNR = 28 dB, BER = 0, maka pada kondisi MIMO akan mempunyai kualitas kanal yang lebih baik, dibandingkan dengan yang bukan MIMO sebesar 29 dB, BER = 0. Itu artinya akan semakin besar pula kemungkinan jalur itu dipakai untuk lalu-lintas komunikasi data dan sinyal dalam kecepatan tinggi.

Kata kunci : WiMAX-OFDM, MIMO-SM, QPSK, AWGN, BER.

HALAMAN MOTTO

*“Optimis, Karena Hidup Terus Mengalir Dan
Kehidupan Terus Berputar”*

Zaman semakin canggih dan modern, kemajuan Teknologi pun terus berkembang. Perkembangan IT di dunia semakin pesat, seperti yang diketahui bahwa Globalisasi banyak membawa budaya asing ke dalam bangsa kita. IT sering berubah fungsi menjadi alat kulturisasi budaya asing tanpa kita sadari. Namun tidak seperti media televisi, cetak atau radio yang harus melalui penyeleksian oleh pemerintah. Banyak diantara budaya asing tersebut yang membawa dampak positif maupun negatif bagi moral maupun budaya bangsa kita.

Pengaruh positif yang ditimbulkan oleh kemajuan Teknologi yaitu : Meningkatnya rasa percaya diri. Dengan adanya kemajuan teknologi saat-saat ini, dapat mempermudah kita dalam menjalin silaturahmi tanpa harus datang ke tempatnya langsung. Karena sudah adanya SMS, Email, telephone dan lain-lain.

Pengaruh negatif yang ditimbulkan oleh kemajuan Teknologi yaitu : Kemerosotan moral di kalangan warga masyarakat, khususnya di kalangan remaja dan pelajar yang dapat menyebabkan kemiskinan dalam rohani. Namun manusia tidak bisa menipu diri sendiri akan kenyataan bahwa teknologi mendatangkan berbagai efek negatif bagi manusia.

KATA PENGANTAR



Puji syukur penulis panjatkan kehadirat Allah SWT atas rahmat, berkah, dan hidayah-Nya sehingga pembuatan Tugas Akhir dengan judul “*Simulasi Pemodelan MIMO Spatial Multiplexing pada WiMAX 802.16d*”. ini dapat diselesaikan dengan baik. Tugas Akhir ini disusun untuk memperoleh Gelar Sarjana pada Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.

Tugas Akhir ini mencoba untuk mendukung penelitian baik teoritis dan eksperimental Teknologi WiMAX-OFDM, sebuah uji simulasi yang didasarkan pada standar IEEE 802.16d khususnya pada penggunaan *fixed-WiMAX* yang akan dikembangkan untuk mendukung eksperimentasi teknik baru akses tetap *broadband wireless* pada PHY-MAC dan lapisan jaringan.

Akhirnya penulis berharap bahwa Tugas Akhir ini dapat berguna bagi seluruh pembaca sebagai langkah awal tumbuh mengembangkan teknologi informasi khususnya di Indonesia.

Kritik dan saran dapat dialamatkan ke aditya.surachman@ymail.co.id

UCAPAN TERIMA KASIH

Begitu banyak pihak yang telah membantu, dan begitu banyak ucapan terima kasih yang tidak terucap. Lewat lembar inilah penulis ingin sampaikan beribu terima kasih yang tak terhingga kepada:

1. Allah SWT dan Rasullullah Muhammad SAW yang telah membimbing dan memberikan petunjuk hidup.
2. Bapak Tito Yuwono, ST., M. Sc, selaku Ketua Jurusan Teknik Elektro dan juga sebagai dosen pembimbing I yang telah banyak memberikan masukan, nasihat dan petunjuk kepada penulis dalam pembuatan laporan ini.
3. Bapak Dr., Eng, Hendra Setiawan, ST., MT, selaku dosen pembimbing II, atas bimbingannya penulis sangat terbantu dalam perancangan simulasi serta dalam penulisan laporan.
4. Kedua orang tua penulis; Ayahanda Sumarno dan Ibunda Eko Wahyuni, yang banyak memberikan support serta kasih sayang dan doa yang tak terhingga, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan sebaik-baiknya.
5. Kakakku Astried yang telah memberikan arahan-arahan petunjuk bagaimana membuat laporan skripsi yang baik dan benar.
6. Kedua Adikku; Fanny dan Rian yang selalu memberikan senyuman indah, Sehingga penulis termotifasi.
7. Terima kasih kepada Om Hari dan Bule Lis yang mana telah memberikan tempat yang indah bagi penulis dalam berkarya, dan kedua putrinya Afi dan Amara yang selalu memberikan keceriaan di rumah, yang membuat penulis slalu ingat dengan kampung halaman.
8. Untuk mbah Kakung dan mbah Putri, Penulis ucapkan banyak terima kasih sudah diberikan tempat tinggal bagi penulis.
9. Bundaku siemut-mut yheny yang selalu menemaniku dalam mengerjakan Tugas Akhir ini, yang memang terkadang menyita waktuku, untuk terdiam sesaat. Dari itu semua penulis sangat berbangga hati telah memilikimu.

1. Mas Alvin Sahroni, atas jasanya dan saran-sarannya, penulis ucapkan banyak terimakasih.
2. Teman-temanku dan sahabatku Angkatan 07; Agus, Ari, Hardi, Rudi, Sapta, Tomi. Kalianlah yang selalu memberikan ide-ide baru bagaimana membuat hasil yang baik itu menjadi lebih baik, dimulai dari bagaimana kita melihat hal-hal yang kecil kemudian menjadi hal yang besar, Thanks to all.
3. Semua pihak yang mana penulis tidak dapat sebutkan satu persatu.

Yogyakarta, Februari 2012



Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	iii
PERSEMBAHAN	iv
ABTRAKSI	v
HALAMAN MOTTO	vi
KATA PENGANTAR	vii
UCAPAN TERIMA KASIH	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
DAFTAR SINGKATAN	xvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	1
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	2
1.5. Sistematik Penulisan	2
BAB II TINJAUAN PUSTAKA	4
2.1. Standar WiMAX-OFDM	4
2.2. Struktur <i>Frame</i> 802.16d	5
2.3. Teknologi OFDM	7
2.4. Teknologi MIMO	9
BAB III PERANCANGAN SIMULASI	10
3.1. Penentuan Parameter	10

3.2. Simulasi MATLAB	12
3.2.1. Paramereter MAC	13
3.2.2. Pembangkit Data	15
3.2.3. Blok <i>Randomization</i>	15
3.2.4. Blok <i>Encoder</i>	16
3.2.5. <i>Convolutional Enkoder</i>	18
3.2.6. Blok Parameter Selektor	19
3.2.7. Penyisipan (<i>Interleaver</i>)	19
3.2.8. <i>IQ Mapper</i>	20
3.2.9. <i>Serial to Parallel</i> MIMO	22
3.2.10. Modulasi OFDM	22
3.2.11. Kanal AWGN	25
3.2.12. Estimasi Kanal OFDM	25
3.2.13. <i>Parallel to Serial</i> OFDM	27
3.2.14. <i>Parallel to Serial</i> MIMO	27
3.2.15. <i>IQ Demapper</i>	27
3.2.16. <i>Deinterleaver</i>	29
3.2.17. Parameter <i>Upsample</i>	29
3.2.18. Parameter <i>Add</i>	29
3.2.19. Algoritma <i>Veterbi</i> pada <i>Convolutional Decoder</i>	30
3.2.20. <i>Decoder Reed Solomon</i>	31
3.2.21. Penyusunan Kembali Bit (<i>Derandomization</i>)	32
3.2.22. Parameter MAC <i>Receiver</i>	33
3.3. Bahasa Pemrograman	34
 BAB IV ANALISIS DAN PEMBAHASAN	 36
4.1. Pendahuluan	36
4.2. Analisis <i>Input</i> dan <i>Output</i> Data	36
4.2.1. Data <i>Randomization</i>	36
4.2.2. Data <i>Encoder</i>	37
4.2.3. Data <i>Convolutional Encoder</i>	38
4.2.4. Data Selektor	39

4.2.5. Data <i>Interleaver</i>	40
4.2.6. Data IQ <i>Mapper</i>	41
4.2.7. Data Modulasi OFDM	42
4.2.8. Data AWGN	43
4.2.9. Data Demodulasi OFDM	44
4.2.10. Data IQ <i>Demapper</i>	45
4.2.11. Data <i>Deinterleaver</i>	45
4.2.12. Data <i>Upsample</i>	46
4.2.13. Data <i>Add</i>	47
4.2.14. Data <i>Convolutional Decoder</i>	47
4.2.15. Data <i>Decoder</i>	48
4.2.16. Data <i>Derandomization</i>	49
4.3. Analisis Pengujiana kanal AWGN	50
4.3.1. Model pengujian SNR dari 0dB s/d 60dB Pada WiMAX ber-MIMO	50
4.3.2. Model pemrograman untuk SNR vs BER	57
BAB V PENUTUP	61
5.1. KESIMPULAN	61
5.2. SARAN	62
DAFTAR PUSTAKA	63
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1	Bentuk <i>Frame</i> 802.16D	6
Gambar 2.2	Spektrum kanal OFDM	8
Gambar 2.3	Pemisahan sinyal OFDM pada kanal <i>subcarrier</i>	8
Gambar 2.4	Pembagian pengukuran FFT dari <i>user</i>	9
Gambar 3.1	Diagram alir Simulasi WiMAX MIMO- <i>Spatial Multiplexing</i>	12
Gambar 3.2	Simulasi dengan 2Tx-2Rx antena MIMO (a) Model pengirim, (b) Model penerima	13
Gambar 3.3	Parameter MAC dan Blok Parameter Pembentuknya	14
Gambar 3.4	Blok <i>Randomization</i>	16
Gambar 3.5	Blok <i>Encoder</i>	17
Gambar 3.6	Blok Konvolusi <i>Encoder</i> dengan laju kode $\frac{1}{2}$	18
Gambar 3.7	Blok Konvolusi <i>Encoder</i>	18
Gambar 3.8	Blok Parameter selektor	19
Gambar 3.9	Blok <i>Interleaver</i>	20
Gambar 3.10	Blok IQ <i>Mapper</i>	21
Gambar 3.11	Posisi dan konstelasi sinyal <i>pilot</i>	22
Gambar 3.12	Blok OFDM	23
Gambar 3.13	Blok Kanal AWGN	25
Gambar 3.14	Blok Data OFDM	26
Gambar 3.15	Blok IQ <i>Demapper</i>	28
Gambar 3.16	Blok <i>Deinterleaver</i>	29
Gambar 3.17	Blok Parameter <i>Upsample</i>	29
Gambar 3.18	Blok Parameter <i>Add</i>	30
Gambar 3.19	Blok <i>Convolutional Dekoder</i>	30
Gambar 3.20	Blok <i>Decoder</i>	31
Gambar 3.21	Blok <i>Derandomization</i>	33
Gambar 3.22	Blok Parameter MAC <i>Receiver</i>	34
Gambar 4.1	<i>Input</i> dan <i>Output Randomization</i>	36
Gambar 4.2	<i>Input</i> dan <i>Output Encoder</i>	37

Gambar 4.3	<i>Input dan Output Convolutional Encoder</i>	38
Gambar 4.4	<i>Input dan Output Selektor</i>	39
Gambar 4.5	<i>Input dan Output Interleaver</i>	40
Gambar 4.6	<i>Input dan Output IQ Mapper</i>	41
Gambar 4.7	<i>Input dan Output Modulasi OFDM</i>	42
Gambar 4.8	<i>Input dan output data AWGN</i>	43
Gambar 4.9	<i>Input dan Output Demodulasi OFDM</i>	44
Gambar 4.10	<i>Input dan Output IQ Demapper</i>	45
Gambar 4.11	<i>Input dan Output Deinterleaver</i>	45
Gambar 4.12	<i>Input dan Output Upsample</i>	46
Gambar 4.13	<i>Input dan Output Add</i>	47
Gambar 4.14	<i>Input dan Output Convolutional Decoder</i>	47
Gambar 4.15	<i>Input dan Output Decoder</i>	48
Gambar 4.16	<i>Input dan Output Derandomization</i>	49
Gambar 4.17	Data terima untuk SNR 0dB	50
Gambar 4.18	Data terima untuk SNR 1dB	50
Gambar 4.19	Data terima untuk SNR 2dB	50
Gambar 4.20	Data terima untuk SNR 3dB	51
Gambar 4.21	Data terima untuk SNR 4dB	51
Gambar 4.22	Data terima untuk SNR 5dB	51
Gambar 4.23	Data terima untuk SNR 6dB	51
Gambar 4.24	Data terima untuk SNR 7dB	52
Gambar 4.25	Data terima untuk SNR 8dB	52
Gambar 4.26	Data terima untuk SNR 9dB	52
Gambar 4.27	Data terima untuk SNR 10dB	52
Gambar 4.28	Data terima untuk SNR 11dB	53
Gambar 4.29	Data terima untuk SNR 12dB	53
Gambar 4.30	Data terima untuk SNR 13dB	53
Gambar 4.31	Data terima untuk SNR 14dB	53
Gambar 4.32	Data terima untuk SNR 15dB	54
Gambar 4.33	Data terima untuk SNR 16dB	54
Gambar 4.34	Data terima untuk SNR 17dB	54

Gambar 4.35	Data terima untuk SNR 18dB	54
Gambar 4.36	Data terima untuk SNR 19dB	55
Gambar 4.37	Data terima untuk SNR 20dB	55
Gambar 4.38	Data terima untuk SNR 21dB	55
Gambar 4.39	Data terima untuk SNR 22dB	55
Gambar 4.40	Data terima untuk SNR 23dB	56
Gambar 4.41	Data terima untuk SNR 24dB	56
Gambar 4.42	Data terima untuk SNR 25dB	56
Gambar 4.43	Data terima untuk SNR 26dB	56
Gambar 4.44	Data terima untuk SNR 27dB	57
Gambar 4.45	Data terima untuk SNR 28dB s/d 60dB	57
Gambar 4.46	Grafik data BER vs SNR (AWGN)	58



DAFTAR TABEL

Tabel 2.1 Perbandingan Standar IEEE 802.16	5
Tabel 3.1 Parameter Model Lapisan Fisik	10
Tabel 3.2 Parameter Frekuensi WiMAX-OFDM	11
Tabel 3.3 Pemetaan bit pada <i>mapping</i> QPSK	21



DAFTAR SINGKATAN

ACK	: <i>Acknowledgement</i>
AWGN	: <i>Additive White Noise Gaussian</i>
BE	: <i>Best Effort</i>
BER	: <i>bit error rate</i>
BPSK	: <i>Base Phase Shift Keying</i>
BS	: <i>Base Station</i>
CP	: <i>Cyclic Prefix</i>
CPE	: <i>Customer Premises Equipment</i>
CQICH	: <i>Channel-Quality Indicator Channel</i>
dB	: <i>Decibel</i>
DSL	: <i>Digital Subscriber Line</i>
FCH	: <i>Frame Control Header</i>
FDD	: <i>Frequency Division Duplex</i>
FEC	: <i>Forward Error Correction</i>
FFT	: <i>Fast Fourier Transform</i>
IEEE	: <i>Institute of Electrical and Electronics Engineers</i>
IFFT	: <i>Inverse Fast Fourier Transform</i>
ISI	: <i>Inter Symbol Interferensi</i>
LDPC	: <i>Low Density Parity Check</i>
LFSR	: <i>Linear Feedback Shift Register</i>
LOS	: <i>Line of Sight</i>
MAN	: <i>Metropolitan Area Network</i>
MAC	: <i>Media Access Control</i>
MIMO	: <i>Multiple Input Multiple Output</i>
MIMO-SM	: <i>Multiple Input Multiple Output-Spatial Multiplexing</i>
MS	: <i>Mobile Station</i>
NLOS	: <i>Non Line of Sight</i>
OFDM	: <i>Orthogonal Frequency Division Multiplexing</i>

OFDMA	: <i>Orthogonal Frequency Division Multiple Access</i>
p2s	: <i>Parallel-to-Serial</i>
PHY	: <i>Physical Layer</i>
QAM	: <i>Quadrature Amplitude Modulation</i>
Qos	: <i>Quality of Service</i>
QPSK	: <i>Quadrature Phase Shift Keying</i>
RF	: <i>Radio Frequency</i>
RS	: <i>Reed Solomon</i>
Rx	: <i>Receiver</i>
s2p	: <i>Serial-to-Parallel</i>
SNR	: <i>Signal to-Noise Ratio</i>
SM	: <i>Spatial Multiplexing</i>
S-OFDMA	: <i>Scalable-Orthogonal Frequency Division Multiple Access</i>
SS	: <i>Subcarrier Station</i>
SSID	: <i>Service Set Identifier</i>
SUI	: <i>Stanford University Intern</i>
TDD	: <i>Time Division Duplex</i>
TDM	: <i>Time Division Multiplexing</i>
TDMA	: <i>Time Division Multiple Access</i>
Tx	: <i>Transmitter</i>
WiMAX	: <i>Worldwide interoperability for Microwave Access</i>
WMAN	: <i>Wireless Metropolitan Wireless Access</i>
XOR	: <i>Xclusive OR</i>
ZP	: <i>Zero Padding</i>
ZF	: <i>Zero Forcing</i>

BAB I

PENDAHULUAN

1.1. Latar Belakang

802.16-2004 merupakan standar yang berbasis 802.16 dan 802.16a dengan beberapa perbaikan. 802.16-2004 diperuntukan bagi layanan yang bersifat *fixed* dan *nomadic*. Terdapat dua opsi dalam transmisi pada 802.16d, yaitu *Time Division Duplex* (TDD) maupun *Frequency Division Duplex* (FDD). Sistem ini menggunakan *Orthogonal Frequency Division Multiplexing* (OFDM) dengan jumlah *carrier* 256 dan mendukung untuk kondisi lingkungan *Line of Sight* (LOS) dan *Non Line of Sight* (NLOS). Standar ini beroperasi pada band frekuensi 3,5 GHz dan 5,8 GHz.

Sedangkan untuk penggunaan teknik antena MIMO, yang dimana telah diketahui dapat meningkatkan performansi sistem, salah satu skema yang ada pada MIMO adalah *spatial multiplexing*; yaitu skema yang menawarkan kelebihan peningkatan data *rate* teknik modulasi OFDM.

Pada Tugas akhir ini akan dirancang dan disimulasikan pemodelan sistem MIMO *spatial multiplexing* yang memenuhi spesifikasi standar *fixed-WiMAX* 802.16d. Sistem yang digunakan dalam simulasi mengacu pada parameter-parameter yang ada dalam *fixed-WiMAX wireless MAN-OFDM*, dengan menggunakan model *single* antena untuk WiMAX, dengan model kanal yang digunakan adalah *noise* AWGN.

1.2. Rumusan Masalah

Beberapa rumusan masalah pada Tugas akhir ini adalah:

1. Bagaimana membuat model, “Simulasi Pemodelan MIMO *Spatial Multiplexing* pada WiMAX 802.16d”.
2. Bagaimana membandingkan sistem WiMAX dengan sistem WiMAX ber-MIMO.
3. Bagaimana membandingkan pengujian BER dari beberapa masukan SNR mulai dari 0-60 dB pada sistem WiMAX.

1.3. Batasan Masalah

Agar penelitian menjadi terarah, maka parameter yang tidak menjadi objek penelitian diasumsikan ideal, tidak variatif dan mengikuti standar yang ada. Maka dari itu perlu dibuat batasan masalah, yaitu:

- a) Parameter sistem WiMAX mengacu pada standar IEEE 802.16-2004
- b) Tidak memperhitungkan pengaruh interferensi sistem lain.
- c) Akuisisi dan sinkronisasi dianggap ideal.
- d) Modulasi dianggap ideal dengan pemodelan *mapping* QPSK.
- e) Simulasi dilakukan dengan perangkat lunak MATLAB 7.8 2009a.

1.4. Tujuan Penelitian

Tujuan dari Tugas akhir ini adalah untuk mendukung penelitian dengan eksperimental Teknologi *fixed*-WiMAX Wireless Man-OFDM. Merupakan sebuah uji simulasi yang didasarkan pada standar IEEE 802.16-2004 yang akan dikembangkan untuk mendukung eksperimentasi teknik baru akses tetap *broadband wireless* di PHY, MAC dan lapisan jaringan. Domain diperoleh dari MIMO matriks yang digunakan untuk merumuskan skema ini dengan kompleksitas rendah.

1.5. Sistematika Penulisan

Dalam penulisan Tugas akhir ini dibagi dalam beberapa bab sesuai dengan urutan proses pengerjaan skripsi:

- a) **BAB I PENDAHULUAN**
Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan dari Tugas akhir ini.
- b) **BAB II TINJAUAN PUSTAKA**
Berisi penjelasan tentang WiMAX IEEE 802.16-2004 dan sistem MIMO secara umum.
- c) **BAB III PERANCANGAN SIMULASI**
Berisi pembahasan model simulasi, diagram alir program simulasi dan nilai parameter simulasi yang digunakan.

d) **BAB IV ANALISIS DAN PEMBAHASAN**

Berisi analisis terhadap hasil simulasi pada semua variasi sistem yang diuji.

e) **BAB V PENUTUP**

Berisi kesimpulan dari hasil simulasi serta saran yang dapat digunakan untuk penelitian berikutnya.



BAB II

TINJAUAN PUSTAKA

Teknologi *fixed*-WiMAX 802.16d. Merupakan teknik yang dilakukan dengan memecah dari kanal ke kanal yang terkecil dengan menggunakan antena canggih seperti antena MIMO. Penggunaan akses tetap ini meliputi; BS maupun penerima yang merupakan CPE dimana fungsi dari perangkat ini adalah sebagai media yang dapat menangkap SSID, dengan sebuah *wireless* dan membaginya kembali ke dalam akses jaringan. Dengan konfigurasi untuk layanan dipedesaan sampai radius 50 km, dan layanan di daerah berpenduduk padat di perkotaan untuk jarak 1-4 km, dengan data rate sampai 75Mbps. Dapat dibayangkan dengan teknologi ini peralatan *wireless point-to-multipoint*, NLOS, *last-mile access* dan solusi *backhaul* yang memungkinkan melengkapi, memperluas, bahkan menggantikan infrastruktur jaringan kabel atau DSL.

2.1. Standar WiMAX-OFDM

WiMAX-OFDM menawarkan keuntungan pada operator dengan lebarnya keterlambatan yang besar pada lingkungan NLOS. Dengan durasi simbol OFDM dan menggunakan *cyclic prefix* yang baik, sehingga OFDM dapat mengeliminasi masalah *Inter Symbol Interferensi* (ISI) dan kompleksitas dari *equalisasi* yang adaptif. Karena sinyal OFDM dapat mengubah *carrier* yang berdekatan saling tegak, dengan selektif *fading* yang dilokalisasi untuk *sub-carrier* yang relatif mudah untuk *diequalisasi*. Contohnya perbandingan antara sinyal OFDM dan sinyal *single carrier*, dengan informasi yang dikirim *parallel* untuk OFDM dan seri untuk *single carrier*.

Teknologi WiMAX (IEEE 802.16d) dapat mampu mengatasi dan mengurangi masalah pada N-LOS serta memiliki keunggulan yang disebabkan oleh penggunaan Teknologi OFDM. Dengan Teknologi OFDM maka cakupan daerah yang luas dapat mencapai hingga beberapa mil/sekitar 50-an kilometer dengan kecepatan tinggi sekitar 72 Mbps *wireless*.

Tabel 2.1 Perbandingan Standar IEEE 802.16

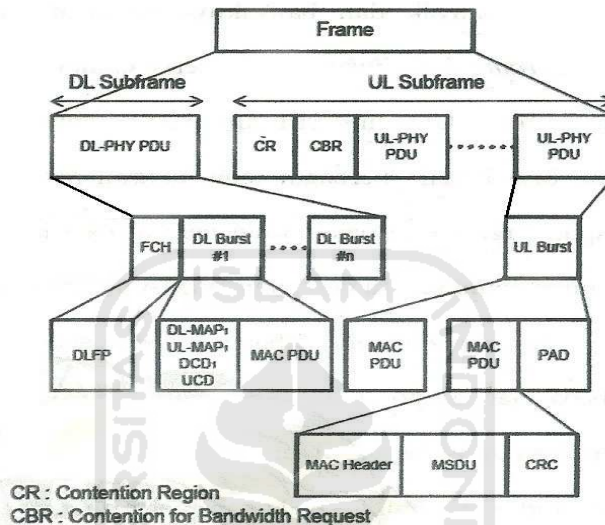
Standar	IEEE 802.16	IEEE 802.16a/802.16d	IEEE 802.16e
Awal Terbentuk	Desember 2001	802.16a: Januari 2003 802.16d: Juni 2004	Pertengahan ke dua 2005
Spektrum	10-66 GHz	2-11 GHz	2-11 GHz untuk <i>fixed</i> 2-6 GHz untuk <i>mobile</i>
Kondisi Kanal	<i>Fixed</i> LOS	<i>Fixed</i> NLOS	<i>Fixed</i> dan <i>mobile</i> NLOS
Arsitektur MAC	<i>Point-to-multipoint, mesh</i>	<i>Point-to multipoint, mesh</i>	<i>Point-to-multipoint, mesh</i>
Sekema Transmisi	<i>Single carrier</i>	<i>Single carrier</i> , 256 OFDM atau 2048 OFDMA	<i>Single carrier</i> , 258 OFDM atau <i>scalable</i> OFDM dengan 126, 512, 1024 atau 2048 <i>subcarriers</i>
<i>Bit Rate</i>	32 sampai 134 Mbps menggunakan frekuensi kanal 28 MHz	Hingga 70 Mbps menggunakan frekuensi kanal 20 MHz	Hingga 15 Mbps menggunakan frekuensi kanal 5 MHz
Modulasi	QPSK, 16 QAM dan 64 QAM	BPSK, QPSK, 16 QAM, 64 QAM	BPSK, QPSK, 16 QAM, 64 QAM
Mobiliti	Perangkat <i>wireless</i> tetap	Perangkat <i>wireless</i> tetap dan portabel	<i>Nomadic Mobility</i>
Implementasi WiMAX	Tidak ada	256 - OFDM sebagai <i>fixed</i> WiMAX	<i>Scalable</i> OFDMA sebagai <i>mobile</i> WiMAX
Frekuensi Per Kanal	20, 25 dan 28 MHz	Mulai dari 1,5 hingga 20 MHz	Mulai dari 1,5 hingga 20 MHz
<i>Radius Per Cell</i>	2 sampai 5 km	7-10 Km dengan kemampuan maksimal hingga 50 Km	2-5 Km.
<i>Multiplexing</i>	<i>Burst</i> TDM/TDMA	<i>Burst</i> TDM/TDMA/OFDMA	<i>Burst</i> TDM/TDMA/OFDM
<i>Duplexing</i>	TDD dan FDD	TDD dan FDD	TDD dan FDD

2.2. Struktur *Frame* WiMAX

Lapisan PHY dari WiMAX bertanggung jawab untuk alokasi *framing* diudara. Gambar 2.1 menunjukkan bahwa *frame* OFDM ketika beroperasi pada model TDD. *Frame* dibagi kedalam dua buah *subframe*; sebuah *frame* *downlink* yang diikuti oleh sebuah *frame* *uplink* setelah sebuah waktu jeda. Perbandingan antara *subframe* *uplink* dengan *downlink* berkisar dari 3:1 hingga 1:1 untuk mendukung profil trafik yang lain. WiMAX juga mendukung *Division Multiplexing*, dimana struktur *frame* sama kecuali baik pada *downlink* dan *uplink* ditransmisikan secara bersamaan melalui *carrier* yang berbeda. TDD

memungkinkan fasilitas pembangkit *bandwidth* antara *uplink* dan *downlink*, dan memiliki desain *transceiver* yang lebih sederhana.

Kekurangan dari TDD adalah kebutuhan untuk melakukan sinkronisasi di antara *Base Station* yang berbeda untuk memastikan keberadaan yang bebas interferensi.



Gambar 2.1 Bentuk *Frame* 8021.6d

Subframe terdiri dari sebuah *Frame Control Header* (FCH) yang menyediakan informasi konfigurasi *frame*, seperti panjang pesan MAP.

Beberapa pengguna dialokasikan dengan data wilayah di dalam *frame*, dan alokasi ini dispesifikasikan dipesan MAP *uplink* dan *downlink* (DL-MAP dan UL-MAP) yang *broadcast* mengikuti FCH pada *subframe downlink*. Pesan MAP meliputi profil *burst* untuk masing-masing pengguna, yang mana mendefinisikan modulasi dan skema coding yang digunakan *link*. Karena MAP berisi informasi yang penting yang diperlukan untuk mencapai semua pengguna, maka MAP sering dikirim *link* yang dapat diandalkan, seperti QPSK dengan *coding rate* 1/2.

WiMAX dianggap cukup fleksibel dalam hal bagaimana pengguna yang berjumlah banyak dan banyaknya paket *multiplexing* pada sebuah *frame*. Sebuah *frame downlink* mungkin berisi campuran data dengan ukuran-ukuran dan tipe yang berbeda. Ukuran dari *frame* juga bervariasi dari 2 ms hingga 20

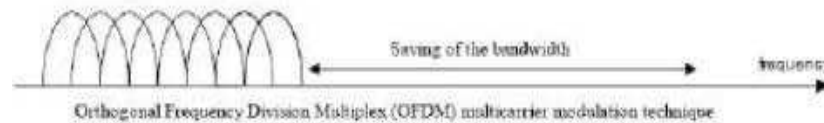
ms, dan setiap *frame* bisa berisi berbagai jenis ukuran *frame*, *frame* dengan ukuran sama ataupun bagian dari paket yang berasal dari lapisan yang lebih tinggi. Paling tidak, bagaimanapun, semua peralatan WiMAX hanya akan mendukung *frame* dengan 5 ms.

Subframe uplink dibentuk oleh beberapa data *uplink* dari pengguna yang berbeda. Sebuah porsi *subframe uplink* yang direvisikan untuk akses berbasis isi yang digunakan untuk berbagai jenis fungsi. *Subframe* ini pada umumnya digunakan untuk kanal *ranging* untuk melakukan performasi *close-loop frequency*, waktu, dan penyesuaian daya selama *entry* ke jaringan dan juga secara periodik setelah *entry*. Kanal *ranging* bisa digunakan oleh *Subcarrier Station* (SS) ataupun *Mobile Station* (MS) untuk membuat *uplink bandwidth requests*. Sebagai tambahan, data dengan *Best-Effort* bisa juga dikirim pada kanal berbasis isi ini, secara khusus ketika jumlah dari data yang akan dikirim terlalu kecil untuk memungkinkan merequest sebuah kanal tersendiri. Selain kanal *ranging* dan data trafik, *subframe uplink* memiliki *Channel-Quality Indicator Channel* (CQICH) untuk memungkinkan SS untuk memfeedback informasi kualitas kanal yang bisa dimanfaatkan oleh rencana dari *Base Station* dan kanal *Acknowledgement* (ACK) bagi *subcarrier station* yang memfeedback *downlink Acknowledgement*.

2. 3. Teknologi OFDM

Orthogonal Frequency Division Multiplexing (OFDM) merupakan sebuah teknik *multiplexing* sinyal melalui skema modulasi *multicarrier* dimana sebuah sinyal dengan *bit-rate* yang tinggi dibagi menjadi beberapa kanal dengan *bit-rate* yang lebih rendah dengan pita frekuensi yang sempit dan saling berdekatan dengan setiap kanal menggunakan frekuensi yang berbeda-beda. Modulasi *multicarrier* ini dapat mengurangi bahkan menghilangkan *Inter Symbol Interference* (ISI). Ide dasar dari modulasi *multicarrier* adalah membagi laju data yang tinggi ke dalam beberapa aliran laju *parallel* data rendah dan memodulasi setiap aliran *parallel* tersebut dengan *carriers* yang berbeda (*subcarriers*).

Frekuensi pada *subcarriers* tersebut tegak lurus (*orthogonal*) antara satu dengan lainnya sehingga secara teoritis telah mengeliminasi interferensi antar kanal. Dengan teknik modulasi *multicarrier* ini diharapkan dapat mengatasi masalah *multipath* dan *delay spread* yang sering menjadi kendala dalam komunikasi nirkabel.



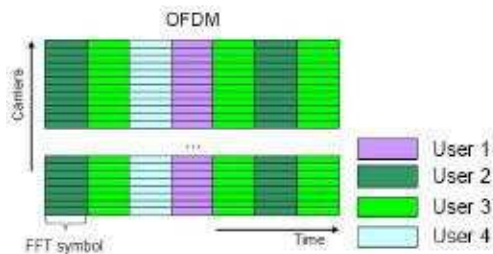
Gambar 2.2 Spektrum kanal OFDM

Gambar di atas menunjukkan bahwa OFDM memberikan efisiensi pada penggunaan spektrum dimana posisi setiap kanal saling berdekatan bahkan saling berimpitan satu sama lainnya. Pada OFDM sinyal-sinyal dipisahkan kedalam beberapa kanal subcarrier, lihat gambar 2.2, lalu dimodulasi dan dimultiplex untuk membentuk OFDM carrier. Subcarrier dapat dimultiplex dengan menggunakan metode OFDM yang disebut *transmission multicarrier* tersebut bersifat *orthogonal*. Sifat *orthogonalitas* memberikan transmisi secara simultan pada *subcarrier* yang banyak dalam ruang frekuensi yang sempit tanpa saling berinterferensi.



Gambar 2.3 Pemisahan sinyal OFDM pada kanal *subcarrier*

Teknik *multiplexing* OFDM diterapkan pada sistem WiMAX 802.16-2004 yang mempunyai FFT size sebesar 256 *subcarrier* dalam satu simbol modulasi. Karena FFT size tetap, maka spasi antara *subcarrier* disesuaikan berdasarkan *channel bandwidth*.



Gambar 2.4 Pembagian pengukuran FFT dari *user*

Pada teknik OFDM setiap *user* menempati satu *time-slot* termodulasi untuk ditransmisikan dengan memakai satu *channel bandwidth* penuh, lihat gambar di atas sehingga masing-masing *user* menempati satu *bandwidth* penuh untuk mengirim.

2.1.3. Teknologi MIMO

Multiple Input Multiple Output (MIMO) adalah Teknologi nirkabel dimana sinyal ditransmisikan menggunakan dua atau lebih antena yang sama, ada dua atau lebih antena di ujung penerima. Algoritma khusus yang diterapkan untuk mengekstrak pesan dari beberapa aliran informasi yang datang pada akhir penerima. MIMO meringankan efek dari *multipath distorsi*, RF; dimana bagian yang berbeda dari sinyal mengambil jalan yang berbeda antara pemancar dan penerima. Secara keseluruhan meningkatkan sinyal rasio kebisingan transmisi, sehingga kualitas *link* lebih dapat diandalkan dan sangat meningkat, baik jangkauan dan kapasitas dari sistem nirkabel.

Bentuk sistem antena canggih, biasanya digunakan untuk memfokuskan sinyal yang dikirimkan ke salah satu lokasi pelanggan dengan lebih efisien memanfaatkan kekuatan pemancar, namun MIMO lebih diterima secara luas untuk meningkatkan kapasitas dan *throughput*. Oleh karena itu MIMO diperlukan untuk sertifikasi WiMAX forum dan disebarluaskan pertama kali oleh operator yang paling mampu mendukung akses aplikasi tersebut.

BAB III

PERANCANGAN SIMULASI

3.1. Penentuan Parameter

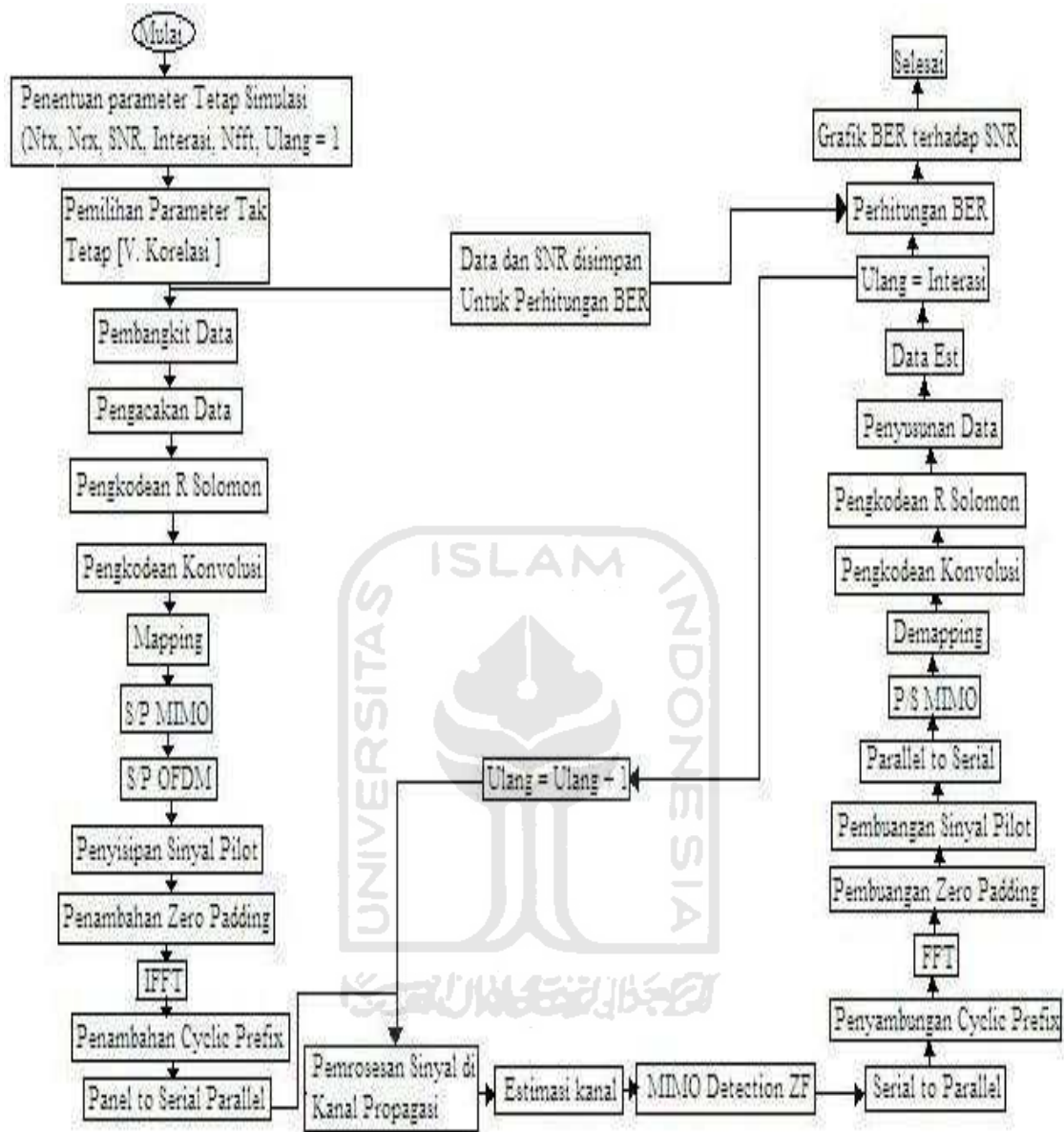
Dalam pemodelan simulasi penggunaan model parameter sangatlah penting agar dapat terdistribusi dengan baik dengan parameter yang sebenarnya sehingga nantinya dapat bersesuaian pada parameter yang sudah ada. Oleh karena itu penulis tentukan batasan-batasan apa saja yang penulis ingin sampaikan, tentunya sesuai dengan parameter yang sudah ada. Dapat dilihat pada tabel 3.1.

Tabel 3.1 Parameter Model Lapisan Fisik

No	Parameter Model Layer Fisik	Menggunakan	Standar <i>Fixed-WiMAX 802.16d</i>
1	Jumlah <i>Subcarrier</i>	256 <i>subcarrier</i>	256 <i>subcarrier</i>
2	<i>Subcarrier</i> untuk data	192 <i>subcarrier</i>	192 <i>subcarrier</i>
3	<i>Subcarrier</i> untuk <i>pilot</i>	8 <i>subcarrier</i>	8 <i>subcarrier</i>
4	<i>Subcarrier</i> untuk ZP	56 <i>subcarrier</i>	28 dan 56 <i>subcarrier</i>
5	<i>Bit Rate</i>	8 Mbps	Sampai dengan 75 Mbps, r 20 MHz
6	<i>Priode guard time</i>	1/32	1/32, 1/16, 1/8, 1/4
7	<i>Mapping</i>	QPSK	BPSK, QPSK, 16QAM, 64QAM
8	<i>Code Rate</i>	1/2	1/2, 2/3, 3/4
9	Frekuensi <i>carrier</i>	3,5 GHz	3,5 GHz dan 5.8 GHz
10	Model MIMO	<i>Spatial Multiplexing</i>	STBC, SM
11	Antena MIMO	2Tx-2Rx	2Tx-2Rx, 4Tx-4Rx
12	Model Kanal	AWGN	AWGN, <i>Rayleigh Fading</i> , SUI

Tabel 3.2 Parameter Frekuensi WiMAX-OFDM

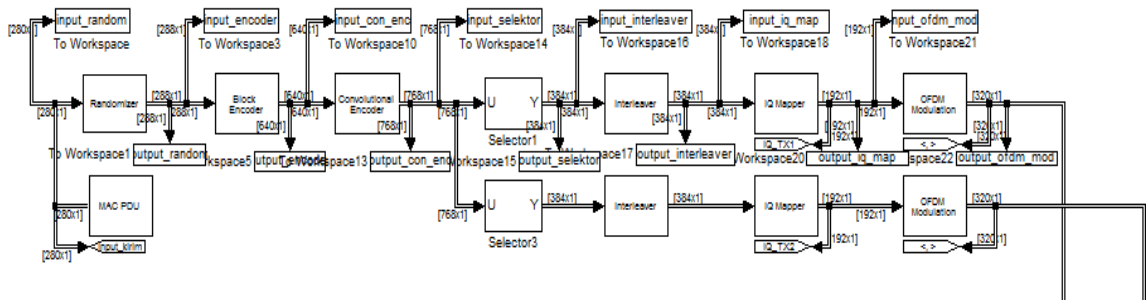
<i>Band Index</i>	<i>Frekuensi Band</i>	<i>Saluran Bandwidth</i>	<i>Ukuran OFDM FFT</i>	<i>Duplexing</i>	<i>Catatan</i>
<i>Profil Fixed WiMAX</i>					
1	3,5 GHz	3,5 MHz	256	FDD	Hasil telah terjamin
		3,5 MHz	256	FDD	
		7 MHz	256	FDD	
		7 MHz	256	FDD	
2	5,8 GHz	10 MHz	256	FDD	
<i>Profil Mobile WiMAX</i>					
1	2,3-2,4 GHz	5 MHz	512	TDD	Kedua <i>bandwith</i> harus supot dengan stasiun pemancar (MS)
		10 MHz	1,024	TDD	
		8,75 MHz	1,024	TDD	
2	2,305 GHz-2,320 GHz- 2,345 GHz- 2,360 GHz	3,5 MHz	512	TDD	
		5 MHz	512	TDD	
		10 MHz	1,024	TDD	
3	2,496 GHz-2,69 GHz	5 MHz	512	TDD	Kedua <i>bandwith</i> harus supot dengan stasiun pemancar (MS)
		10 MHz	1,024	TDD	
4	3,3-3,4 GHz	5 MHz	512	TDD	
		7 MHz	1,024	TDD	
		10 MHz	1,024	TDD	
5	3,4-3,8 GHz	5 MHz	512	TDD	
	3,4-3,6 GHz	7 MHz	1,024	TDD	
	3,6-3,8 GHz	10 MHz	1,024	TDD	



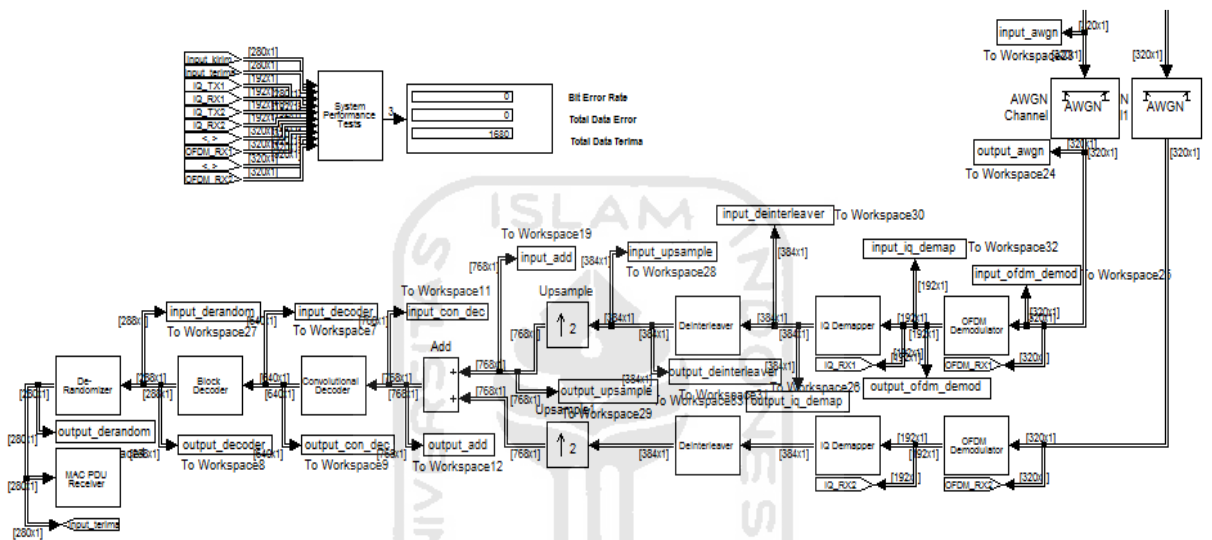
Gambar 3.1 Diagram alir Simulasi WiMAX MIMO-Spatial Multiplexing

3.2. Simulasi MATLAB

Pada simulasi ini digunakan perangkat lunak MATLAB 7.8 R2009a. Beberapa bagian dari simulasi ini menggunakan fungsi-fungsi yang telah ada di MATLAB, seperti pembangkit *randomization*, *encoder*, *decoder*. Model simulasi secara umum terlihat pada gambar 3.2 yang terdiri atas beberapa blok.



(a)



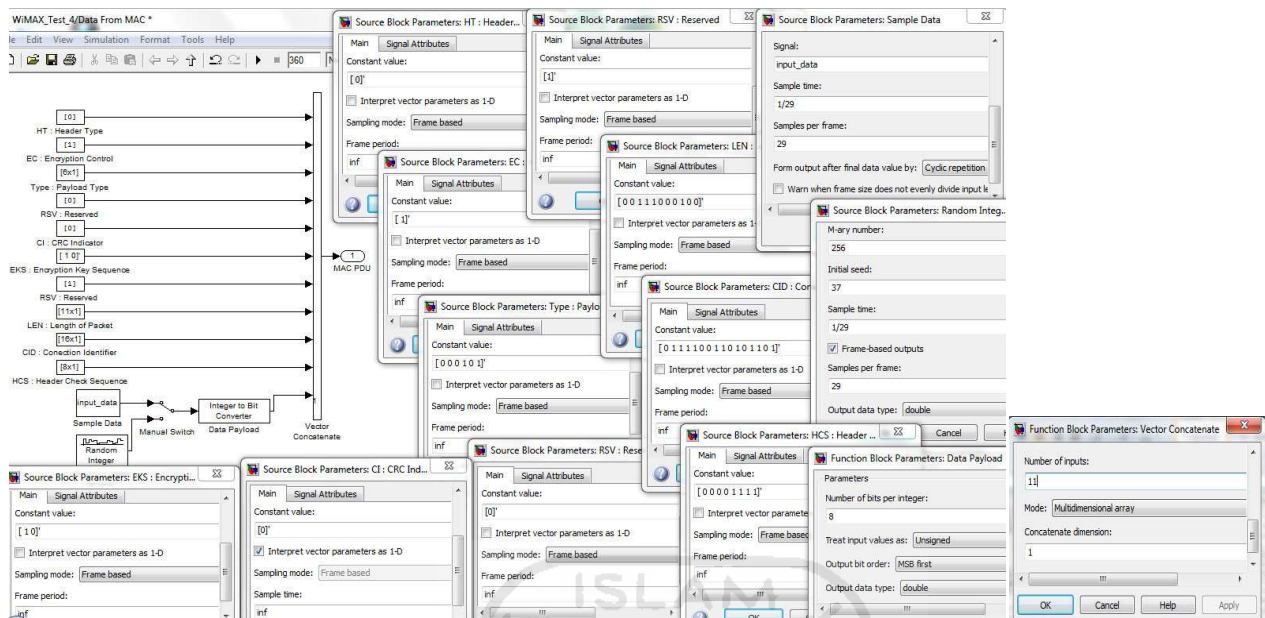
(b)

Gambar 3.2 Simulasi dengan 2Tx-2Rx antenna MIMO

(a) Model pengirim, (b) Model penerima

3.2.1. Parameter MAC

Merupakan bentuk penggabungan vektor dari beberapa blok, dimana dari beberapa blok ditentukan dengan nilai data tetap, dari penggabungan vektor juga telah digabungkan dengan aliran data percobaan dan pengacakan *integer generator* yang dihubungkan dengan *manual switch* yang keluarannya berupa data *payload* yang kemudian digabungkan lagi dengan penggabungan vektor.



Gambar 3.3 Parameter MAC dan Blok Parameter Pembentuknya

Alur perancangan simulasi :

- a. Parameter sumber HT (*Header Type*); pada parameter ini akan dimasukan dengan nilai konstan = $[0]'$, dengan model penarikan menggunakan bingkai mendasar, periode bingkai = inf (terus-menerus). Untuk perlengkapan sinyal menggunakan tipe data keluaran rangkap.
- b. Parameter sumber EC (*Encryption Control*); ditentukan dengan nilai tetap = $[1]'$.
- c. Parameter sumber (*Payload Type*); ditentukan dengan nilai tetap = $[0\ 0\ 0\ 1\ 0\ 1]'$.
- d. Parameter sumber RSV(*Reserved*); ditentukan dengan nilai tetap = $[0]'$.
- e. Parameter sumber CI (*CRC Indicator*); ditentukan dengan nilai tetap = $[0]'$
- f. Parameter EKS (*Encryption Key Sequence*); ditentukan dengan nilai tetap = $[10]'$.
- g. Parameter sumber RSV(*Reserved*); ditentukan dengan nilai tetap = $[1]'$ sama seperti EC.
- h. Parameter sumber LEN (*Length of Packet*); ditentukan dengan nilai tetap = $[0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0]'$.
- i. Parameter sumber CID (*Conection Indentifier*); ditentukan dengan nilai tetap = $[0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1]'$.

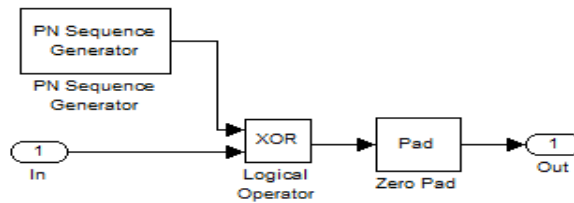
- j. Parameter sumber HSC (*Header Check Sequence*); ditentukan dengan nilai tetap = [0 0 0 0 1 1 1 1]'
- k. Parameter sumber data percobaan; dikondisikan dengan model sinyal masukan_data, kemudian untuk sampel durasi = 1/29, dengan jumlah sampel tiap-bingkai = 29 bit. Untuk bentuk keluaran setelah data akhir ditentukan dengan pengulangan siklis.
- l. Parameter sumber pembangkit pengacakan bilangan; diperoleh dengan jumlah Mary = 256 bit, kemudian untuk jumlah bibit awal = 37 bit, dengan sampel durasi = 1/29 yang ditentukan dengan model keluaran bingkai-mendasar. Untuk model tipe data keluaran terkondisi rangkap.
- m. Parameter muatan data; menggunakan deret 8 bit, dengan menyisipi nilai masukan selama *unsigned*, kemudian keluaran bit pesan dikondisikan dengan model MSB pertama, sedangkan untuk tipe data keluaran terkondisi rangkap.
- n. Parameter penggabungan vektor; menggunakan masukan bilangan 11 bit, dengan menggunakan model kesatuan luas dimensi, untuk penggabungan 1 dimensi bit.

3.2.2. Pembangkit Data

Pembangkit data yang digunakan pada pemodelan simulasi ini merupakan data yang dihasilkan oleh pembangkit Bernolli. Penggunaan pembangkit Bernolli dikarenakan probabilitas kemunculan bit "0" dan bit "1" yang dihasilkan sama atau terdistribusi seragam. Keluran pembangkit data memiliki dua tingkat tegangan yang membedakan kedua jenis bit tersebut.

3.2.3. Blok *Randomization*

Randomization dapat disebut pula sebagai aliran bit data yang akan diacak dengan modulo-2 dari bit keluaran *pseudorandom binary sequence generator*, dengan sumber informasi data (XOR) dan menggunakan model keluaran dari *Linear-Feedback Shift Register* (LFSR) yang memiliki karakteristik seperti; *sequence polynomial* $x^{15} + x^{14} + 1$ dengan bit 15 yang dapat diprogram. Setiap pengiriman paket data isi dari register *pseudorandom binary sequence* akan dibersihkan dan dimasukkan kembali dengan deretan bit dari *generator polinomial* [1 0 0 0 0 0 0 0 0 0 0 1 1] dan harus *clock* per-sekali bit proses.



Gambar 3.4 Blok *Randomization*

Alur perancangansimulasi:

a. Parameter *PN Sequence Generator*

Pada parameter ini akan dimasukan 15 bit, dengan masukan data *generator polinomial* = [1 0 0 0 0 0 0 0 0 0 0 1 1] dan masukan awalan = [0 0 0 1 1 1 0 1 1 1 1 0 0 0 1], sampel-durasi = 1/280 dan sampel tiap-bingkai = 280 bit.

b. Parameter *Logical Operator*

Pada Parameter ini akan digunakan kanal operator XOR, sebagai informasi data dari *pseudorandom binary sequence generator*.

c. Parameter *Zero Pad*

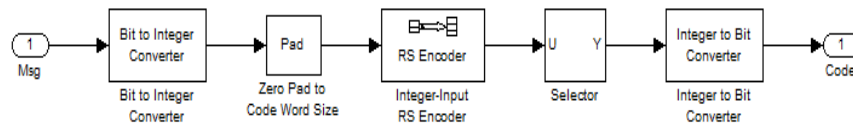
Parameter ini digunakan sebagai penambah sinyal nol yang berguna untuk menjaga masukan blok IFFT agar tetap berjumlah 256 jalur frekuensi. Dari masukan acak akan dibagi menjadi kumpulan data yang kecil sebanyak 288 bit.

3.2.4. Blok *Encoder*

Blok *encoder* yang merupakan bentuk dari masukan data yang berupa pembulatan pengacakan data yang dibagi menjadi kumpulan data yang lebih kecil seperti model algoritma *serial to parallel*, yang kemudian data ini diencoding dengan *code-rate* tertentu. Proses pesandian tersebut disebut *Forward Error Correction* (FEC) dimana tujuannya untuk meningkatkan kapasitas dari kanal dengan menambah bit *redundant* pada data informasi yang ditransmisikan. Proses penambahan bit *redundant* disebut saluran pesandian yang dipakai pada simulasi *covolutional encoder*, di dalam blok pesandian akan tersusun penggunaan *integer Reed Solomon*.

Reed Solomon disini digunakan pada pengkodean luar. Masukan pengkodean *Reed Solomon* yang digunakan pada simulasi ini menggunakan 256 level sinyal,

panjang data masukan pengkodean luar sepanjang 239 bit, dan panjang keluaran kata sandinya sepanjang 255 bit, dan agar didapat nilai koreksi sebesar 16 *byte* maka harus ditambahkan 32 bit pada masukan pengkodean luar ini. Pengkodean *Reed Solomon* memerlukan memori pada proses pengkodeannya, dan tidak akan terjadi proses sebelum semua memori terisi. Oleh sebab itu pengkodean ini akan menyebabkan keterlambatan yang cukup besar.



Gambar 3.5 Blok *Encoder*

Alur perancangan simulasi :

a. Parameter *Bit to Integer Converter*

Pada parameter ini akan ditentukan dengan jumlah *bit-rate* vektor yang akan digunakan untuk membagi menjadi kumpulan data yang lebih kecil, dengan jumlah vektor *bit-per integer* = 8 bit. Masukan bit dipesan lewat MSB pertama, setelah bit dikondisikan kemudian nilai bit akan dihilangkan selama *unsigned*. Untuk tipe data keluaran sama seperti masukan asal.

b. Parameter *Zero Pad to Code Word Size*

Pada parameter ini akan dilakukan pengukuran kolom pada pengkodean *reed solomon*, dengan pengukuran sinyal kolom sebanyak 239.

c. Parameter *Integer Input RS Encoder*

Pada parameter ini akan ditentukan panjang keluaran kata sandinya sepanjang 255 bit dan panjang data masukan pengkodean luar 239 bit. Dengan menetapkan metode *polynomial*; dengan *polynomial* sederhana dan *polynomial generator*.

d. Parameter Selektor

Parameter ini akan menggunakan 1 masukan dimensi dengan kondisi *one-based*. Dengan menyusun; $[220:243,184:239] = [(243-220)+(239-184)] = 80$ bit, dengan ukuran masukan sisi kiri = 255 bit.

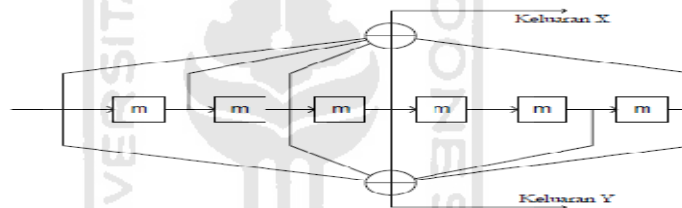
e. Parameter *Integer to Bit Converter*

Pada parameter ini akan ditentukan dengan jumlah *bit-rate* vektor yang akan digunakan untuk membagi menjadi kumpulan data yang lebih kecil, dengan jumlah vektor *bit per-integer* = 8 bit. Dan akan menghilangkan nilai masukan selama *unsigned*, kemudian dengan keluaran bit pesan MSB pertama akan diperoleh tipe data keluaran yang sama seperti masukan asal.

3.2.5. Convolutional Encoder

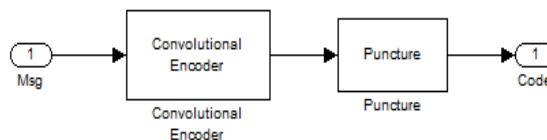
Pengkodean konvolusi digunakan pada pengkodean dalam (*inner code*). Pada perancangan pengkodean konvolusi ini digunakan 6 memori pergeseran, dan memiliki dua keluaran yang memiliki laju kode $\frac{1}{2}$.

Sedangkan *puncture*, yaitu proses secara sistematis menghapus bit dari aliran *output* dengan tujuan mereduksi sejumlah data yang akan ditransmisikan.



Gambar 3.6 Blok Konvolusi *Encoder* dengan laju kode $\frac{1}{2}$

Untuk *polinomial generator* yang diatur pada IEEE 802.16d adalah $G_1 = 171_{OTC}$ untuk X dan $G_2 = 133_{OTC}$ untuk Y. Meskipun pengkodean menggunakan memori sebagai penggeser, tetapi hal tersebut tidak terlalu menambah keterlambatan. Hal ini dikarenakan pengkodean memproses data secara berurutan dan tanpa harus menunggu isi memori penggeser terisi. Nilai keterlambatan yang ditambahkan oleh pengkodean konvolusi hanya sebesar 1 durasi bit, untuk konvolusi *decoder* menggunakan algoritma *veterbi*.



Gambar 3.7 Blok Konvolusi *Encoder*

Alur perancangan simulasi :

a. Parameter *Convolutional Encoder*

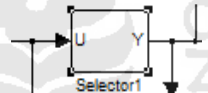
Parameter ini akan menggunakan struktur jari-jari dengan jari-jari *polynomial* (7, [171 133]), menggunakan sistem operasi *continous*.

b. Parameter *Puncture*

Pada parameter ini akan direduksi dengan vektor *puncture*; dengan mengasah kembali dengan masukan; ([1 0 10 1;1 10 10], 10, 1).

3.2.6. Blok Parameter Selektor

Parameter selektor pada simulasi ini berfungsi sebagai kanal pembagi untuk 2 saluran antenna MIMO, dan sebagai pemisah berupa data ganjil dan genap, oleh sebab itu pada pengisian vektor indeks untuk data awalnya apabila 1 maka berbentuk ganjil dan apabila 2 maka berbentuk genap.



Gambar 3.8 Blok Parameter Selektor

Alur perancangan simulasi:

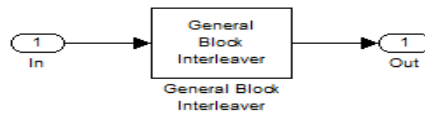
Pada parameter selektor akan dikondisikan, menggunakan 1 bit masukan dimensi dengan model indeks operasi *one-based*, yang didistribusikan dengan vektor indeks [1:2:768]; artinya untuk nilai 1 merupakan bentuk data pembagi ganjil, untuk nilai 2 adalah bentuk pembagian untuk data $768 = 768:2 = 384$ bit, sedangkan untuk ukuran *port* masukan = 768, merupakan bentuk pembagi menjadi 2 saluran.

3.2.7. Penyisipan (*Interleaver*)

Proses penyisipan berfungsi sebagai cara untuk meminimalisasi keretakan kesalahan selama transmisi data. Penyisipan akan menyusun data kembali dimana data yang berurutan akan dibagi menjadi beberapa blok berbeda. Penyisipan

berbentuk matriks, dengan aliran bit masukan akan mengisi matriks tersebut baris demi baris.

Keluaran dari penyisipan merupakan bit yang tersusun secara kolom. Penyisipan dari simulator telah ditetapkan dalam rentan data yang melewati sebuah matriks penyisipan dengan blok *input* baris demi baris dan kemudian mengirim isi matriks kolom dengan kolom, rumusan untuk blok ini adalah jumlah baris yang membentuk matriks.



Gambar 3.9 Blok *Interleaver*

Alur perancangan simulasi :

Pada parameter ini akan ditransformasikan lebar sinyal dengan panjang elemen; *int_idx*, dimana pada *int_idx* akan dijadikan parameter indeks pada blok *Interleaver*. Keluaran dari blok *Interleaver* akan menjadi masukan pada blok modulator dimana pada blok modulator terdapat beberapa parameter yang harus dimasukkan.

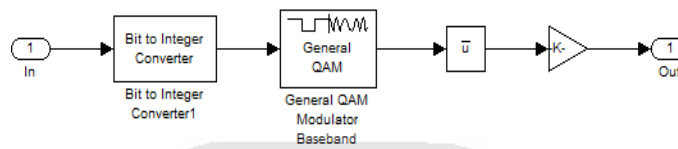
3.2.8. IQ Mapper

Hanya ada satu skema pemetaan simbol atau modulasi digital yang digunakan pada tugas akhir ini, skema tersebut adalah QPSK yang dalam hal ini nilai *roll of factor* filternya diasumsikan 0. Pemetaan QPSK sendiri memiliki efisiensi *bandwidth* dua kali lebih besar dari BPSK, karena dua bit disimbolkan dengan satu sinyal *mapper*.

Dengan *input* digital ke modulator QPSK yang merupakan sinyal biner, maka untuk menghasilkan 4 kondisi masukan yang berbeda harus dipakai 2 bit masukan; 00, 01, 10 dan 11. Masing-masing *level* sinyal disimbolkan dengan beda fasa 90^0 .

Tabel 3.3 Pemetaan bit pada *mapping* QPSK

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



Gambar 3.10 Blok IQ Mapper

Alur perancangan simulasi :

- a. Parameter *Bit to Integer Converter*
 Pada parameter ini akan dipakai 2 bit masukan. Masukan bit dipesan lewat MSB pertama, setelah bit dikondisikan kemudian nilai bit akan dihilangkan selama *unsigned*. Untuk tipe data keluaran sama seperti masukan asal.
- b. Parameter *General QAM Modulation Baseband*
 Parameter ini akan membagi sinyal *qamcost*, yang merupakan bagian dari bentuk pemetaan skema QPSK, dengan menggunakan 1 bit sampel per simbol.
- c. Parameter Fungsi *Math*
 Parameter fungsi *math*, dipetakan dengan fungsi sambung (*conjungction*) fungsi yang disambung adalah bagian dari sinyal *qamcost* yang akan dibentuk menjadi 4 kondisi masukan untuk pembentukan skema QPSK, dengan tipe sinyal keluaran yang berjalan sendiri (*auto*). dan akan mengalami durasi (-1 untuk menerima) sebesar -1.
- d. Parameter *Gain*
 Bentuk parameter *gain* ditentukan dengan $1/\sqrt{2}$ dengan perkalian model *element-wise* ($K.*u$). Sedangkan untuk durasinya = -1, tipe data keluaran yang

diperoleh sama seperti masukan asal, dengan model operasi pembulatan dasar. Sedangkan untuk tipe data parameter diperoleh sama seperti *input* asal.

3.2.9. *Serial to Parallel* MIMO

Pada parameter ini akan memecah deretan simbol kirim menjadi 2 buah deretan simbol sejajar yang nantinya dikirimkan melalui masing-masing antena pengirim yang juga berjumlah 2 buah. Karena MIMO-OFDM maka sebelum dikirimkan ke masing-masing antena, masing-masing deretan simbol sejajar tersebut dilakukan proses modulasi *multicarrier*.

3.2.10. Modulasi OFDM (*Serial to Paralel*)

Modulasi OFDM (*Serial to Paralel*), berguna untuk membagi sinyal yang masih dalam satu jalur frekuensi kedalam 192 jalur frekuensi yang berbeda dengan kecepatan keluaran yang lebih rendah pada setiap jalur frekuensinya.

Untuk pembangkit sinyal *pilot* dan penyisipan sinyal *pilot* pada standar IEEE 802.16d ditetapkan jumlah sinyal *pilot* sebanyak 8 sinyal. Sinyal *pilot* diperlukan pada teknik estimasi kanal. Sinyal *pilot* yang disisipkan merupakan sinyal *tone* yang memiliki konstelasi QPSK dan posisi sinyal sudah diketahui oleh penerima.

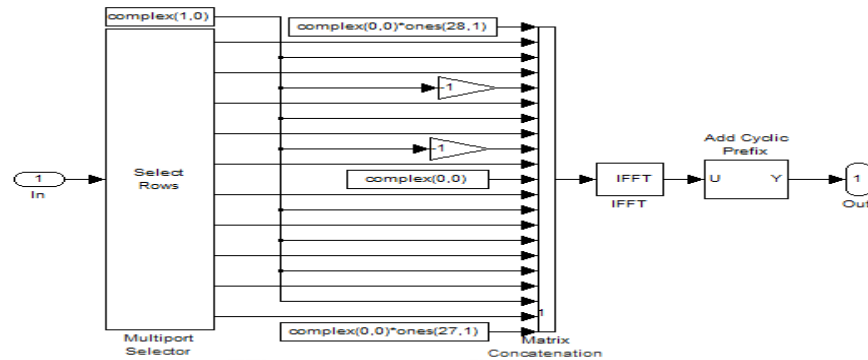
Parameter	Nomor Sinyal Pilot							
	1	2	3	4	5	6	7	8
Posisi	1	30	58	86	115	143	171	200
Konstelasi	1+i	-1+i	-1-i	1-i	1+i	-1+i	-1-i	1-i

Gambar 3.11 Posisi dan konstelasi sinyal *pilot*

Sedangkan pada kanal *Inverse Fast Fourier Transform* (IFFT) berfungsi sebagai OFDM *baseband* modulator, dimana setiap simbol akan memodulasi frekuensi *subcarrier* yang dibangkitkan oleh blok IFFT. Jumlah titik IFFT yang digunakan sebanyak 256 titik, penggunaan IFFT akan menjamin *orthogonalitas* antar *subcarrier*. Keluaran dari setiap proses transformasi 256 titik IFFT akan membentuk sebuah deret simbol OFDM.

Setelah pemasangan IFFT akan ditambahkan model *Cyclic Prefix* (Ekstensi Siklis). Berfungsi sama dengan pemberian *guard interval* pada sistem komunikasi lain, akan tetapi penambahan *guard interval* yang tidak tepat pada sistem

komunikasi yang menggunakan OFDM akan membuat kehilangan sifat *orthogonalnya* pada *subcarrier* yang berdekatan. Penambahan ekstensi siklis pada model yang dirancang sepanjang 1/32 dari total panjang kerangka OFDM.



Gambar 3.12 Blok OFDM

Alur perancangan simulasi :

a. Parameter sumber tetap 1

Pada parameter ini akan diberikan harga tetap (1,0), dengan menafsirkan parameter vektor seperti 1-D, kemudian untuk model penarikan sampel menggunakan skema sampel mendasar dan sampel durasi inf (terus-menerus).

b. Parameter *multiport* selektor

Perancangan *multiport* selektor dilakukan oleh metode parameter indeks dengan keluaran; {1:12, 13:36, 37:60, 61:84, 85:96, 97:108, 109:132, 133:156, 157:180, 181:192} = {(12x1)+(24x1)+(24x1)+(24x1)+(12x1)+(12x1)+(24x1)+(24x1)+(12x1)} = 192 *subcarrier*. Hasil 192 *subcarrier* diperuntukan sebagai data simbol terima dari OFDM.

c. Parameter sumber tetap 2

Parameter ini akan diberikan harga tetap (0,0)*ones(28,1). Dengan menafsirkan parameter vektor seperti 1-D, model penarikan sampel menggunakan skema sampel mendasar dan sampel durasi inf (terus-menerus). Untuk perlengkapan sinyal menggunakan tipe data keluaran penerima; menerima dari nilai harga tetap.

d. Parameter fungsi

Pada parameter ini akan diberikan 2 model parameter fungsi dengan masing-masing akan diberikan $gain = -1$, dengan perkalian model *element-wise* ($K.*u$). Sedangkan untuk durasinya = -1. Menggunakan tipe data keluaran penerima; menerima melewati kaidah bagian dalam. Sedangkan untuk model bilangan *integer* menggunakan skema tingkat. Kemudian untuk perlengkapan parameter akan digunakan tipe data parameter dengan penerima; menerima dengan melewati kaidah bagian dalam.

e. Parameter sumber tetap 3

Pada parameter ini akan diberikan harga tetap $(0, 0)*ones(27, 1)$. Dengan menafsirkan parameter vektor seperti 1-D, kemudian untuk model penarikan sampel menggunakan skema sampel mendasar dan sampel durasi inf (terus-menerus). Untuk perlengkapan sinyal menggunakan tipe data keluaran penerima; menerima dari harga tetap.

f. Parameter sumber tetap 4

Pada parameter ini akan diberikan harga tetap $(0, 0)$. Dengan menafsirkan parameter vektor seperti 1-D, kemudian untuk model penarikan sampel menggunakan skema sampel mendasar dan sampel durasi inf (terus-menerus). Untuk perlengkapan sinyal menggunakan tipe data keluaran penerima; menerima dari harga tetap.

g. Parameter IFFT

Untuk parameter IFFT, Untuk metode komputasi faktor *twiddle* menggunakan skema *lookup*, kemudian untuk metode skema dioptimalkan dengan kecepatan. Dengan ditentukan keluaran pembagi dari panjang FFT, dan ditentukan mengikuti panjang FFT dari *input* dimensi.

Pada operasi parameter *fixed-point*; untuk model pembulatan dipakai metode *floor* (tingkat) dan model *overflow* dipakai metode *wrap* (membungkus). Sedangkan untuk tipe data *fixed-point*; daftar sinus sama dengan panjang perkata dari masukan, hasil keluaran mengikuti kaidah yang melewati bagian dalam, himpunan mengikuti kaidah yang melewati bagian dalam, keluaran mengikuti kaidah yang melewati bagian dalam.

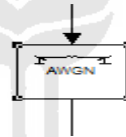
h. Parameter *Add Cyclic Prefix*

Pada parameter ini akan digunakan 1 masukan dimensi dengan model indeks *one-based*, dengan indeks [193:256 1:265], menggunakan ukuran *input port* 256 bit.

3.2.11. 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 terhingga dan respon frekuensi *flat* untuk semua frekuensi, yang kemudian diistilahkan dengan AWGN. Sehingga dapat melewati semua spektral daya sinyal tanpa distorsi.

Kerusakan sinyal hanya disebabkan oleh derau yang *White Gaussian* (AWGN).



Gambar 3.13 Blok Kanal AWGN

Alur perancangan simulasi :

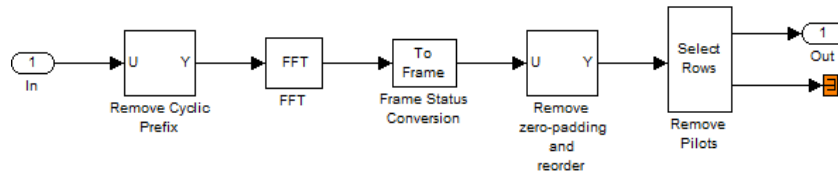
Pada parameter saluran AWGN akan ditentukan dengan benih awal = 1 bit, dengan menggunakan metode *signal to noise ratio* (SNR), dengan melihat pada uji simulasi yang dilakukan akan diberikan nilai SNR (dB) dari 0 s/d 60 dB, *input signal power* = 0.080325450151575 watt, dan kenaikan 10 dB.

3.2.12. Estimasi Kanal OFDM

Blok ini memanfaatkan sinyal *pilot* dari 8 *subcarrier* pilot untuk mengestimasi kanal. Estimasi dengan sinyal *pilot* ini hanya dilakukan pada WiMAX tanpa MIMO. Proses yang pertama terjadi pada blok estimasi kanal adalah pembagian sinyal *pilot* yang diterima dengan sinyal *pilot* referensi pada bagian pengirim. Proses selanjutnya adalah memisahkan bagian *in-phase* dan kwadratur hasil

pembagian tersebut dan dilakukan proses interpolasi agar didapatkan respon kanal bagian *in-phase* dan kwadratur simbol.

Keluaran interpolasi bagian *in-phase* dan kwadratur akan digabungkan, dengan (pada bagian kwadratur dikalikan dengan nilai “i” terlebih dahulu). Hasil penggabungan tersebut akan dikalikan kembali dengan simbol dan sinyal *pilot*.



Gambar 3.14 Blok Data OFDM

Alur perancangan simulasi :

a. Parameter *Remove Cyclic Prefix*

Parameter ini menggunakan 1 masukan dimensi, dengan model indeks *one-based*. Untuk data indeks = [65:320] yang berarti $320 - 65 = 256$ bit, sesuai dengan ukuran penggunaan FFT OFDM. Untuk ukuran *port* masukan = 320 bit.

b. Parameter FFT

Sama seperti parameter IFFT, untuk FFT ini merupakan pembalikan dari struktur data dari IFFT .

c. Parameter *Frame Conversion*

Pada parameter ini hanya menentukan metode parameter *frame-based* dari keluaran sinyalnya.

d. Parameter *Remove zero-padding* dan *reorder*

Parameter ini akan dialokasikan 1 masukan dimensi, dengan model indeks menggunakan skema *one-based*. Dengan data indeks [29:128 130:229] = $[(128 - 29) + (229 - 130)] = 99 + 99 = 198$ bit, dengan ukuran *port* masukan = 256 bit.

e. *Parameter Remove Pilot*

Parameter ini digunakan untuk melepaskan *pilot*, dengan parameter indeks ke *output*; {[1:12 14:37 39:62 64:87 89:112 114:137 139:162 164:187 189:200], [13 38 63 88 113 138 163 188]} = {[$(12 \times 1) + (24 \times 1) + (24 \times 1) + (24 \times 1) + (24 \times 1) + (24 \times 1) + (24 \times 1) + (24 \times 1) + (12 \times 1)$] [13+38+63+88+113+138+163+188 = 804/100]} = [192 8]. Dengan keluaran *subsystem* = 192 bit dan keluaran sinyal terminator = 8 bit.

3.2.13. *Parallel to Serial OFDM*

Perinsip *parallel to serial* ini berfungsi untuk mengubah simbol terima 192 *subcarrier* (256-8-56 *subcarrier*) menjadi satu deretan simbol terima seri.

3.2.14. *Parallel to Serial MIMO*

Perinsip dasarnya sama dengan *parallel to serial* pada OFDM bagian penerima, cuma disini fungsinya adalah menggabungkan kembali 2 buah deretan simbol sejajar menjadi satu deretan simbol lagi. Karena menggunakan sistem MIMO-OFDM, maka sebelum digabungkan, masing-masing deretan simbol sejajar tersebut dilakukan proses demodulasi *multicarrier*.

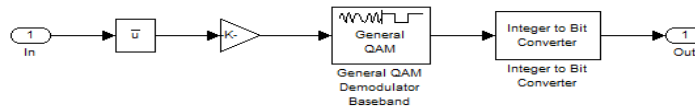
3.2.15. *IQ Demapper*

Blok ini berfungsi untuk menghilangkan *subcarrier* yang berisi sinyal nol karena tidak mengandung informasi. *Subcarrier* yang dihilangkan adalah sama dengan *subcarrier* yang berisi sinyal nol pada blok IFFT. Selain itu *subcarrier* sinyal *pilot* dihilangkan setelah sinyal *pilot* yang diterima digunakan untuk estimasi kanal.

Umumnya QAM demodulasi *baseband demodulating* pada sinyal pengirim menggunakan metode modulasi amplitudo kuadratur. Untuk model sampel berdasarkan pengirim, pengirim harus skalar. Untuk masukan berbasis bingkai, pengirim harus menjadi vektor kolom. Dalam kasus berbasis bingkai lebar bingkai pengirim merupakan hasil dari jumlah simbol dan percobaan per-nilai simbol.

Dalam kasus percobaan *input* akan berbasis masa percobaan dari *input* periode simbol dibagi dengan percobaan per-nilai simbol. Model *integer* untuk konverter sedikitnya dilakukan dengan penggabungan vektor dari nilai masukan *integer*

untuk vektor bit. Blok masukan harus diberikan dengan nilai *integer* dalam rentang $[-2^{(M-1)}, 2^{(M-1)} - 1]$, artinya $[(-2^{(M-1)}) (2^{(M-1)-1})]$ ketika mereka ditugaskan (*signed*) dan $[0, 2^M - 1]$, artinya $[0 (2^{(M-1)})]$ ketika mereka tidak ditugaskan (*unsigned*).



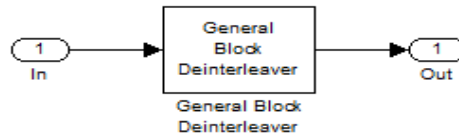
Gambar 3.15 Blok IQ Demapper

Alur perancangan simulasi :

- a. Parameter Fungsi *Math*
 Dalam parameter ini akan diberikan metode fungsi *conjungction*, dengan tipe keluaran sinyal yang akan berjalan sendiri (*auto*), dan akan mengalami waktu (-1 untuk penerima) sebesar -1. Untuk perlengkapan sinyal akan memenuhi proses *integer* hingga selesai.
- b. Parameter *Gain*
 Parameter ini menggunakan $gain1/\sqrt{2}$, dengan perkalian skema *element-wise* ($K,*u$), dan mengalami durasi (-1 untuk menerima) sebesar -1. Untuk perlengkapan sinyal, digunakan tipe data keluaran sama seperti *input* asal. Sedangkan untuk bilangan *integer* dipakai model tingkat. Kemudian untuk perlengkapan parameter akan digunakan tipe data parameter sama seperti masukan asal.
- c. Parameter *General QAM Demodulasi Baseband*
 Pada parameter ini akan digunakan metode konstelasi sinyal menggunakan *qamconst* atau ketetapan QAM, setiap simbol akan dikenai 1 bit.
- d. Parameter *Integerto Bit Converter*
 Pada parameter ini akan diberikan 2 bit *per-integer*, kemudian nilai bit akan dihilangkan selama *unsigned*, kemudian dengan keluaran bit pesan MSB pertama akan diperoleh tipe data keluaran yang sama seperti masukan asal.

3.2.16. *Deinterleaver*

Blok ini akan mengembalikan bentuk simbol terima yang berupa sinyal kompleks menjadi bit data terima.



Gambar 3.16 Blok *Deinterleaver*

Alur perancangan simulasi :

Pada parameter *general* blok *deinterleaver* ini hanya akan diberikan pada jumlah *element* yang akan dibangkitkan, kemudian untuk model penarikan sampel menggunakan skema sampel mendasar dan sampel durasi inf (terus-menerus).

3.2.17. Parameter *Upsample*

Parameter *upsample* merupakan bentuk sisipan $L-1$ diantara 0 yang berupa data sampel masukan; yang berfungsi sebagai data sampel pengganti, berupa bentuk data pertama dengan satu nilai bilangan *integer* dalam jarak $(0, L-1)$



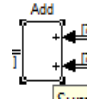
Gambar 3.17 Blok Parameter *Upsample*

Alur perancangan simulasi :

Pada parameter fungsi *upsample* akan diberikan model faktor *upsample*, $L = 2$, dengan sampel pembagi $(0 \text{ untuk } L-1) = 0$, dan untuk kondisisi awal $= 0$, dengan menggunakan metode bingkai mendasar model *maintain input frame rate*.

3.2.18. Parameter *Add*

Parameter *add* dipakai untuk menjumlahkan antara dua atau lebih blok prameter menjadi satu keluarannya blok parameter.



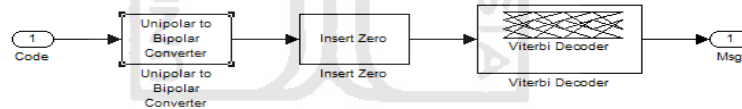
Gambar 3.18 Blok Parameter *Add*

Alur perancangan simulasinya :

Dengan memberikan simbol “++” dua kali untuk dua parameter yang akan dijumlahkan dan dimasukkan dengan sampel durasi = -1, dengan bentuk model rektangular.

3.2.19. Algoritma *Viterbi* pada *Convolutional Dekoder*

Untuk mendapatkan kembali data inputan pengkodean konvolusi, dapat digunakan *viterbi dekoder*. Parameter yang harus ditentukan pada *viterbi dekoder* adalah cara pemutusan dan mengikuti asal ukuran. Cara pemutusan yang digunakan adalah pemutusan *unquantized* dan dimasukkan nilai 8 yang merupakan pengikut asal ukuran. Nilai keterlambatan yang ditambahkan *viterbi dekoder* sangat tergantung dari nilai mengikuti asal ukuran yang ditetapkan.



Gambar 3.19 Blok *Convolutional* Dekoder

Alur perancangan simulasi :

a. Parameter *Unipolar to Bipolar Converter*

Pada parameter ini akan dikonfigurasi dengan dua bilangan *M-ary*, dengan menggunakan polaritas *negatif*, dan tipe data keluaran yang sama seperti masukan asal.

b. Paramet *Insert Zero*

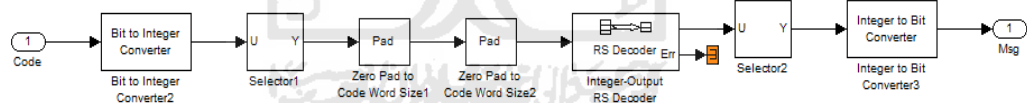
Pada parameter ini akan digunakan skema vektor sisipan nol yang dimasukkan dengan data *reshape* ([1 0 1 0 1;1 1 0 1 0], 10, 10).

c. Parameter *Veterbi Decoder*

Dalam penggunaan parameter *veterbi decoder* dilakukan skema struktur jari-jari dengan 2 *polynomial* jari-jari (7, [171 133]). Untuk cara pemutusan digunakan parameter tipe pemutusan *unquantized*. Dengan ukuran pembalik keadaannya = 8 bit yang merupakan pengikut asal ukuran, dan menggunakan mode operasi *continuous*. Untuk tipe datanya menggunakan data keluaran mengikuti kaidah bagian dalam.

3.2.20. Dekoder Reed-Solomon

Proses pertama yang terjadi pada dekoder ini adalah mengelompokkan deretan bit keluaran blok *viterbi decoder* menjadi 8 bit, atau yang lebih dikenal dengan *byte*, dan akan diteruskan dengan proses pengelompokan deretan *byte* menjadi 255 *byte*. Dengan panjang bingkai bit keluaran *dekoder Reed-Solomon* sepanjang 223 *byte* yang memiliki kemampuan maksimal pengkoreksian kesalahan sebanyak 16 *byte* kesalahan. Sama seperti *Reed-Solomon encoder*, *Reed-Solomon dekoder* memerlukan memori pada proses *decoding* (pemecah kode). Sebelum proses *buffering* (pengelompokan) 255 *byte* selesai, maka proses *decoding* tidak dapat dilakukan.



Gambar 3.20 Blok *Decoder*

Alur perancangan simulasi :

a. Parameter *Bit to Integer Converter*

Proses pertama yang terjadi pada dekoder ini adalah mengelompokkan deretan bit keluaran blok *viterbi decoder* menjadi 8 bit, Masukan bit dipesan lewat MSB pertama, setelah bit dikondisikan kemudian nilai bit akan dihilangkan selama *unsigned*. Untuk tipe data keluaran sama seperti masukan asal.

b. Parameter Selektor1

Dari parameter selektor 1 akan dialokasikan pembagian nilai data indeks [2:80, 1], yang merupakan indeks vektor dialog dengan ukuran *port* masukan 80 bit, menggunakan model indeks tipe *one-based*, dengan 1 input dimensi.

c. Parameter *Zero pad to Code Word Size*

Pada tahap ini akan dilakukan proses pengelompokan deretan *byte*, untuk *pad* sinyal dimulai dengan ukuran pertama yang dikelompokkan menjadi 239 *byte*. Sedangkan untuk ukuran kedua akan dilakukan proses pengelompokan deretan *byte* menjadi 255 *byte*, dengan *pad* sinyal yang dinyatakan selesai.

d. Parameter *Integer Output RS Decoder*

Untuk tahapan berikutnya akan dilakukan pemasukan pengukuran berdasarkan pengelompokan *byte*. Untuk N panjang *codeword* = 255 *byte* dan untuk K panjang pesan = 239 *byte*.

e. Parameter Selektor 2

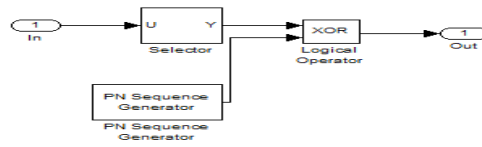
Untuk parameter selektor ke dua akan dialokasikan pembagian nilai data indeks [204:239], yang merupakan indeks vektor (dialog) terhadap *port* masukan 239 *byte*, dengan model indeks *one-based*, dengan 1 bit masukan dimensi.

f. *Integer to Bit Converter*

Untuk parameter *integer to bit converter* akan dilakukan pengelompokan deretan bit keluaran blok *viterbi decoder* menjadi 8 bit, Kemudian akan dilakukan penetapan nilai *input* selama *unsigned*, dengan bit masukan pesan menggunakan metode MSB pertama. Untuk tipe data keluaran sama seperti masukan asal.

3.2.21. Penyusunan Kembali Bit (*Derandomization*)

Derandomization atau penyusunan kembali bit diperlukan untuk mendapatkan kembali data informasi yang sebelumnya sudah diacak. Proses penyusunan kembali sama seperti proses pengacakannya yaitu dengan menggunakan modulo-2 data keluaran dari *pseudorandom binary sequence generator* yang sama seperti pada blok pengirim.



Gambar 3.21 Blok *Derandomization*

Alur perancangan simulasi :

a. Parameter Selektor

Pada parameter selektor akan dilakukan pembagian nilai data indeks [1:280], yang merupakan indeks vektor dialog terhadap *port input 288 byte*, dengan model indeks *one-based*, dan menggunakan 1 bit *input* dimensi.

b. Parameter *PN Sequence Generator*

Sama seperti model masukan pada *randomization* yang merupakan bentuk aliran bit data yang akan diacak, untuk *derandomization* ini merupakan pembalikan bentuk aliran acak tersebut untuk mendapati nilai pembentukan awal. Pada parameter ini akan dimasukan 15 bit, dengan *input* data *generator polinomial* = [1 0 0 0 0 0 0 0 0 0 0 0 1 1], dengan masukan awalan = [0 0 0 1 1 1 0 1 1 1 1 0 0 0 1], sampel-durasi = 1/280 dan sampel tiap-bingkai = 280 bit; merupakan jumlah yang akan disimulasikan dalam 1 bit bingkai data.

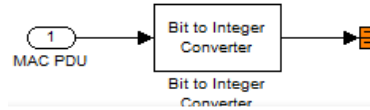
c. Parameter *Logical Operator*

Pada Parameter ini akan digunakan kanal operator XOR, sebagai informasi data dari *pseudorandom binary sequence generator*. Perpotongan gambar dilakukan dengan model *rectangular*, menggunakan 2 *port* masukan dengan sampel waktu = -1. Pada bagian perlengkapan sinyal digunakan tipe *float* (rangkap) sebagai keluaran tipe datanya.

3.2.22. Parameter MAC Receiver

MAC pada *receiver* merupakan bentuk aliran data yang telah dibangkit oleh MAC pada *transmitter*. Dengan skema masukan bit pesan yang menggunakan model MSB pertama, akan dihasilkan nilai *integer* selama *unsigned*. Artinya konversi data dari pengirim ke penerima sudah diberhentikan, karena dari bentuk pengembalian data oleh *derandomization* sudah terkondisi seperti sumber

awalnya. Sehingga untuk MAC pada *receiver* ini hanya mengkodekan kembali aliran bit pesan menjadi bentuk model pertama.



Gambar 3.22 Blok Parameter MAC *Receiver*

Alur perancangan simulasi :

Parameter *Bit to Integer converter*. Pada parameter ini akan diberikan 8 bit, dengan skema masukan bit pesan menggunakan model MSB pertama. Setelah bit ditentukan, maka akan menghasilkan nilai *integer* selama *unsigned*, dengan tipe data keluaran yang sama dengan masukan asal.

3.3. Bahasa Pemrograman

Dalam simulasi ini pertama-tama dimasukan sebuah parameter yang diperlukan yaitu berupa *source* (sumber), dengan perintah :

```
input_data = 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');
randomized_data = sscanf(['D4 BA A1 12 F2 74 96 30 27 D4 88 9C 96 E3 A9 52 B3 15 AB FD 92
53 07 32 C0 62 48 F0 19 22 E0 91 62 1A C1'], '%x');
rs_encoded_data = sscanf(['49 31 40 BF D4 BA A1 12 F2 74 96 30 27 D4 88 9C 96 E3 A9 52 B3 15
AB FD 92 53 07 32 C0 62 48 F0 19 22 E0 91 62 1A C1 00'], '%x');
conv_encoded_data = sscanf(['3A 5E E7 AE 49 9E 6F 1C 6F C1 28 BC BD AB 57 CD BC CD E3
A7 92 CA 92 C2 4D BC 8D 78 32 FB BF DF 23 ED 8A 94 16 27 A5 65 CF 7D 16 7A 45 B8 09
CC'], '%x');
interleaved_data = sscanf(['77 FA 4F 17 4E 3E E6 70 E8 CD 3F 76 90 C4 2C DB F9 B7 FB 43 6C F1
9A BD ED 0A 1C D8 1B EC 9B 30 15 BA DA 31 F5 50 49 7D 56 ED B4 88 CC 72 FC 5C'], '%x');
```

Data yang dihasilkan kemudian diubah menjadi biner dan hasilnya akan diacak. Sebelum memasuki *encoder*, data yang bernilai biner perlu diubah dulu menjadi desimal. Kemudian akan diproses oleh *reed-solomon encoder* dengan parameter *primitive polynomial* dan *generator polynomial* yang dibangkitkan dengan perintah :

```
Gen_Poly = [1 gf(2,8) ^ 0];
for Index = 1 : 15
```

```

Gen_Poly = conv (Gen_Poly, [1 gf (2,8) ^ Index]);
end
Prim_Poly = [1 0 0 0 1 1 1 0 1];

```

Dari hasil *RS-Encoder* akan diteruskan melalui blok *Convolutional Encoder* dan *Puncture* akan masuk ke blok *interleaver*, dengan rumusan-rumusan yang akan dibangkitkan melalui *m-file* program matlab sebagai berikut :

```

Ncbps = 384;
Ncpc = 2;
k = 0 : Ncbps - 1;
mk = (Ncbps/12) * mod (k,12) + floor (k/12);
s = ceil (Ncpc/2);
jk = s * floor (mk/s) + mod (s, mk + Ncbps - floor (12 * mk/Ncbps));
[s, int_idx] = sort (jk);

```

Dimana "int_idx" akan dijadikan parameter indeks pada blok *Interleaver*. Keluaran dari blok *Interleaver* akan menjadi masukan pada blok modulator dimana pada blok modulator terdapat beberapa parameter yang harus dimasukkan. Pada blok ini digunakan sekema 2 QAM dengan *normalisation constant for until average power* (C_m) adalah $1/\sqrt{2}$; dan simbol *alphabet* (A_s) adalah $1+j, 1-j, -1+j, -1-j$. Dimana parameter tersebut akan dibangkitkan dengan *m-file* pada program matlab sebagai berikut :

```

Ry = ones (2,1) * [+1 -1];
Iy = [+1 -1]' * ones (1,2);
qamconst = Ry + j * Iy;
qamconst = qamconst (:)/sqrt (2);

```

BAB IV

ANALISIS DAN PEMBAHASAN

4.1. Pendahuluan

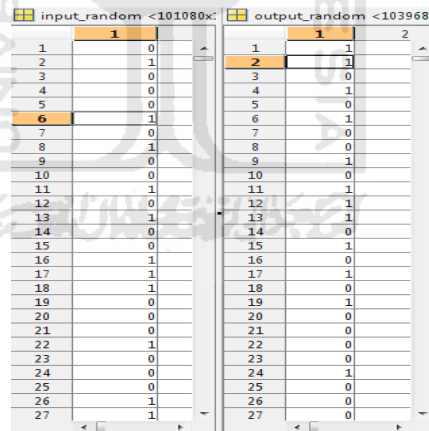
Pada bab ini seperti yang telah dijelaskan pada Bab 1 adalah mengkaji pengaruh penggunaan WiMAX-OFDM, dengan beberapa variasi yang telah dijelaskan pada bab 3 dengan beberapa parameter simulasi khususnya WiMAX mengacu pada nilai-nilai yang ada pada standar IEEE 802.16d.

Hal-hal yang diamati untuk tujuan analisis antara lain :

- a) Analisis *Input* dan *Output* data
- b) Analisis pengujian kanal AWGN

4.2. Analisis *Input* dan *Output* data

4.2.1. Data *Randomization*



input_random <101080x:		output_random <103968x:	
	1	1	2
1	0	1	1
2	1	2	3
3	0	3	0
4	0	4	1
5	0	5	0
6	1	6	1
7	0	7	0
8	1	8	0
9	0	9	1
10	0	10	0
11	1	11	1
12	0	12	1
13	1	13	1
14	0	14	0
15	0	15	1
16	1	16	0
17	1	17	1
18	1	18	0
19	0	19	1
20	0	20	0
21	0	21	0
22	1	22	0
23	0	23	0
24	0	24	1
25	0	25	0
26	1	26	0
27	1	27	0

Gambar 4.1 *Input* dan *Output* data *Randomization*

Pengoperasian kanal *randomization* akan secara langsung memetakan fungsi dan fitur sistem secara sistematis dimana pada operasinya, yakni akan berusaha mengurangi variansi dari perkiraan dengan penyebaran *sampling* yang bersifat acak.

Karena bentuk keluaran dari masukan *randomization* menggunakan model keluaran dari *Linear-Feedback Shift Register (LFSR)*, yang merupakan bentuk dari pengaruh arus balik dari proses pergeseran data acak *randomization*, sehingga bila kita lihat pada gambar 4.1 mulai dari data masukan nomer 6 akan terjadi pergeseran pada keluaran data yang dimulai dari nomer 2.

Dari analisis tersebut dapat diperoleh bahwa *input* data dari *randomization* akan langsung dipetakan oleh *output* data *randomization* untuk mengurangi penyebaran *sampling* yang tidak dapat diperkirakan. Sehingga kesempatan akan terjadinya perpindahan data akan secara langsung dipetakan tanpa adanya keterlambatan.

4.2.2. Data Encoder

input_encoder <103968x>		output_encoder <231040>	
1	2	1	2
1	1	1	0
2	1	2	1
3	0	3	0
4	1	4	0
5	0	5	0
6	1	6	0
7	0	7	1
8	0	8	1
9	1	9	0
10	0	10	1
11	1	11	1
12	1	12	1
13	1	13	1
14	0	14	0
15	1	15	0
16	0	16	0
17	1	17	0
18	0	18	0
19	1	19	1
20	0	20	0
21	0	21	1
22	0	22	1
23	0	23	1
24	1	24	0
25	0	25	1
26	0	26	0
27	0	27	1

Gambar 4.2 Tabel *Input* dan *Output Encoder*

Bentuk dari aliran data *encoder* merupakan jenis kode yang memiliki perbedaan mendasar dari blok kode, dimana urutan bit informasi tidak dikelompok-kelompokan dalam blok-blok yang berbeda sebelum dikodekan.

Proses yang terjadi adalah bit informasi sebagai masukan secara kontinyu dipetakan kedalam urutan bit *output encoder*.

Pada proses *encoding* yang diterapkan akan melakukan operasi biner yang dikhususkan pada bit-bit masukan, sehingga bit-bit sumber dikatakan dikonvolusi untuk menghasilkan deretan bit keluaran. Hal ini juga menyebabkan bit-bit keluaran tidak hanya bergantung pada bit-bit yang

diproses, namun juga bergantung pada deretan sebelumnya dari bit-bit sumber dengan demikian membutuhkan suatu bentuk dari memori. Dalam prakteknya bentuk memori ini dapat berupa *shift register* dengan panjang tertentu.

Dari gambar 4.2, dapat dianalisis bahwa pada keluaran *encoder* akan terjadi keterlambatan yang cukup besar. Karena dari proses pengkodean *Reed Solomon* yang dimana memerlukan memori pada proses pengkodeannya, yang tidak terjadi proses sebelum semua memori terisi. Oleh sebab itu antara masukan dan keluaran dari *encoder*, tidak terlambat geseran data.

4.2.3. Data Convolutional Encoder

input_con_enc	output_con_enc
1	0
2	1
3	0
4	0
5	0
6	0
7	1
8	1
9	0
10	1
11	1
12	1
13	1
14	0
15	0
16	0
17	0
18	0
19	1
20	0
21	1
22	1
23	1
24	0
25	1
26	0
27	1

Gambar 4.3 Input dan Output data Convolutional Encoder

Pada pengkodean konvolusi *encoder*, dimana tiap *codeword* selain bergantung pada pesan yang bersesuaian juga tergantung pada m blok pesan sebelumnya. Satu set *codeword* dengan masukan k, keluaran n dan tingkat memori m yang disebut dengan kode konvolusi (n,k,m). Dengan kode *rate* didefinisikan $R = k/n$. Karena pengkodean konvolusi memiliki tingkat memori maka harus diimplementasikan dengan rangkaian logika *sequensial*.

Kode konvolusi dapat dihasilkan dengan cara melewati urutan bit informasi melalui sejumlah tingkat *shift register*. Pada umumnya *shift register* terdiri dari N (k bit) tingkat dan m *generator polynomial*. Dengan data masukan yang digeser sepanjang k bit *shift register* pada satu kali waktu,

karena jumlah bit keluaran untuk tiap k bit masukan adalah n bit, dengan kode rate $R_c = k/n$.

Seperti yang dijelaskan pada analisis diatas bahwa bentuk dari pengkodean ini menggunakan memori sebagai pergeseran. Ini terlihat bahwa pada masukan data mulai dari nomer 20 akan terjadi pergeseran data yang dimana pergeseran tersebut telah dialokasikan pada keluaran mulai dari nomer 14, dan sempat terjadi keterlambatan. Nilai keterlambatan yang ditambahkan oleh pengkodean konvolusi hanya sebesar 1 durasi bit

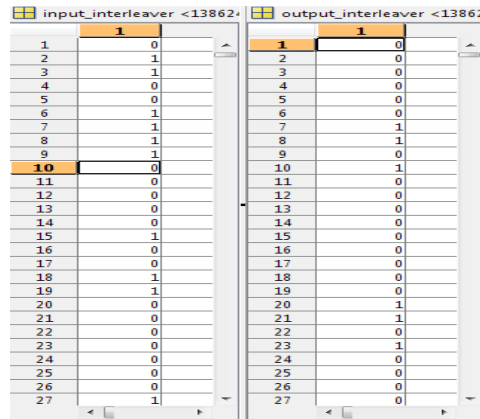
4.2.4. Data Selektor

input_selektor <277248x>		output_selektor <138624x>	
Index	Bit	Index	Bit
1	1	1	0
2	0	2	1
3	1	3	1
4	1	4	0
5	1	5	0
6	1	6	1
7	0	7	1
8	0	8	1
9	0	9	1
10	1	10	0
11	1	11	0
12	1	12	0
13	1	13	0
14	0	14	0
15	1	15	1
16	1	16	0
17	1	17	0
18	0	18	1
19	0	19	1
20	1	20	0
21	0	21	0
22	1	22	0
23	0	23	0
24	1	24	0
25	0	25	0
26	0	26	0
27	0	27	1

Gambar 4.4 *Input dan Output* data Selektor

Pengkodean data selektor menggunakan 1 bit masukan dimensi dengan rentang masukan dimulai pada posisi nomer 1 yang digeser pada keluarannya di nomor 4, pergeseran bit ini terbentuk secara terpisah-pisah antara bit kode 0 dan 1. Karena sesuai fungsinya sebagai kanal pembagi.

4.2.5. Data Interleaver



input_interleaver <13862>		output_interleaver <13862>	
	1		1
1	0	1	0
2	1	2	0
3	1	3	0
4	0	4	0
5	0	5	0
6	1	6	0
7	1	7	1
8	1	8	1
9	1	9	0
10	0	10	1
11	0	11	0
12	0	12	0
13	0	13	0
14	0	14	0
15	1	15	0
16	0	16	0
17	0	17	0
18	1	18	0
19	1	19	0
20	0	20	1
21	0	21	1
22	0	22	0
23	0	23	1
24	0	24	0
25	0	25	0
26	0	26	0
27	1	27	0

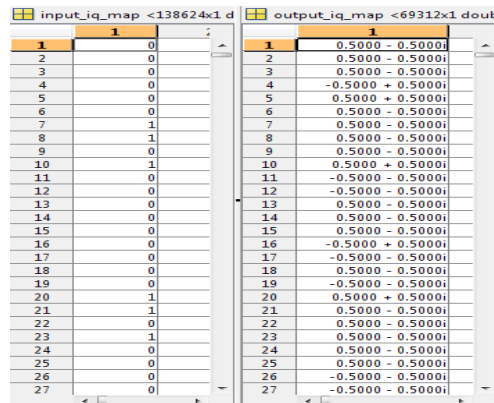
Gambar 4.5 Input dan Output data Interleaver

Salah satu cara metode *interleaver* adalah membagi data *serial* kedalam matriks ($n \times m$) secara berurutan per kolom. Kemudian data dalam matriks tersebut diubah ke dalam data *serial* lagi dengan pembacaan per baris.

Untuk skema *bit-interleaver multiplexer*, dimana akan menerima dan memultiplex data dari sejumlah kanal-kanal sinkron berdasar bit-demi bit. Data yang termultiplex berisi semua karakter sinkronisasi dan kontrol yang dibangkitkan oleh terminal. Karena setiap bit data akan dikirimkan oleh *multiplexer* kesirkuit pembawa segera setelah bit ini diterima, sehingga tidak diperlukan penyangga yang berkapasitas besar dan pengiriman menjadi lebih kecil.

Dari analisis tersebut diperoleh bahwa keluaran dari data *interleaver* merupakan bit yang tersusun secara kolom, itu artinya bahwa pada bentuk kelurannya akan berupa data yang tersusun sesuai dengan metode masukan data, dapat dilihat bahwa mulai dari masukan data nomer 10 akan bergeser pada data keluaran 1 dari keluaran *interleaver*. Sehingga data yang diperoleh akan tersusun baris demi baris dan kemudian mengirim isi matriks kolom dengan kolom.

4.2.6. Data IQ Mapper



input_iq_map <138624x1 d		output_iq_map <69312x1 doub	
Index	Value	Index	Value
1	1	1	0.5000 - 0.5000i
2	0	2	0.5000 - 0.5000i
3	0	3	0.5000 - 0.5000i
4	0	4	-0.5000 + 0.5000i
5	0	5	0.5000 + 0.5000i
6	0	6	0.5000 - 0.5000i
7	1	7	0.5000 - 0.5000i
8	1	8	0.5000 - 0.5000i
9	0	9	0.5000 - 0.5000i
10	1	10	0.5000 + 0.5000i
11	0	11	-0.5000 - 0.5000i
12	0	12	-0.5000 - 0.5000i
13	0	13	0.5000 - 0.5000i
14	0	14	0.5000 - 0.5000i
15	0	15	0.5000 - 0.5000i
16	0	16	-0.5000 + 0.5000i
17	0	17	-0.5000 - 0.5000i
18	0	18	0.5000 - 0.5000i
19	0	19	-0.5000 - 0.5000i
20	1	20	0.5000 + 0.5000i
21	1	21	0.5000 - 0.5000i
22	0	22	0.5000 - 0.5000i
23	1	23	0.5000 - 0.5000i
24	0	24	0.5000 - 0.5000i
25	0	25	0.5000 - 0.5000i
26	0	26	-0.5000 - 0.5000i
27	0	27	-0.5000 - 0.5000i

Gambar 4.6 Input dan Output data IQ Mapper

Data IQ *mapper* disini merupakan suatu bentuk dari pembagian kanal yang disesuaikan dengan modulusnya. Pada perakteknya akan digunakan modulasi QPSK, untuk modulasi QPSK, data akan dibagi kedalam 2 fase, yaitu *Q-phase* dan *I-phase*. Dimulai dari proses *encoder*, data akan melewati kode konvolusi dengan *rate* 1/2. Lalu data akan disebar oleh *Pseudonoise code*. Hasil dari penyebaran data dengan *Pncode* itu selanjutnya akan melewati kanal, sehingga akan dijadikan dalam satu fase untuk ditransmisikan menjadi bentuk aliran data.

Bentuk aliran data pada *mapping* QPSK yang sejatinya terdiri dari empat buah level sinyal yang berupa empat kode *binary* yaitu; 00, 01, 11, 10, yang kemudian akan dimodulasikan digital, oleh sebab itu dari masukan data biner akan dimasukan dengan 2 bit masukan. Sehingga untuk keluaran data akan terpotong menjadi data yang lebih kecil.

4.2.7. Data Modulasi OFDM

input_ofdm_mod <69312x1 d		output_ofdm_mod <115520x	
1	0.5000 - 0.5000i	1	0.0078 - 0.0234i
2	0.5000 - 0.5000i	2	0.0413 + 0.0324i
3	0.5000 - 0.5000i	3	-0.0389 - 0.0067i
4	-0.5000 + 0.5000i	4	-0.0161 - 0.0401i
5	0.5000 + 0.5000i	5	0.0385 - 0.0137i
6	0.5000 - 0.5000i	6	-0.0303 + 0.0590i
7	0.5000 - 0.5000i	7	-0.0004 + 0.0019i
8	0.5000 - 0.5000i	8	0.0141 - 0.0271i
9	0.5000 - 0.5000i	9	0.0033 + 0.0205i
10	0.5000 - 0.5000i	10	0.0077 - 0.0050i
11	-0.5000 - 0.5000i	11	-0.0163 - 0.0196i
12	-0.5000 - 0.5000i	12	-0.0076 - 0.0068i
13	0.5000 - 0.5000i	13	-0.0074 + 0.0130i
14	0.5000 - 0.5000i	14	0.0223 + 0.0007i
15	0.5000 - 0.5000i	15	-0.0223 - 0.0124i
16	-0.5000 + 0.5000i	16	0.0438 + 0.0309i
17	-0.5000 - 0.5000i	17	0.0076 + 0.0262i
18	0.5000 - 0.5000i	18	-0.0077 - 0.0448i
19	-0.5000 - 0.5000i	19	-0.0911 - 0.0217i
20	0.5000 + 0.5000i	20	0.0233 + 0.0364i
21	0.5000 - 0.5000i	21	0.0509 - 0.0224i
22	0.5000 - 0.5000i	22	0.0193 - 0.0152i
23	0.5000 - 0.5000i	23	-0.0006 + 0.0051i
24	0.5000 - 0.5000i	24	-0.0271 + 0.0564i
25	0.5000 - 0.5000i	25	-0.0213 - 0.0054i
26	-0.5000 - 0.5000i	26	0.0381 - 0.0020i
27	-0.5000 - 0.5000i	27	-0.0290 - 0.0531i

Gambar 4.7 Input dan Output data Modulasi OFDM

Pemakaian kanal OFDM menggunakan spektrum yang lebih efisien dengan memisahkan kanal yang berdekatan/*overlapping*. Ini dilakukan dengan membuat semua *carrier* yang dikonfersikan oleh IFFT, yang akan saling *orthogonal* satu sama lain, dengan merubah data yang tadinya *serial to-parallel* menjadi *parallel to-serial*. Istilah *orthogonal* dalam modulasi *multicarrier* mengandung makna hubungan matematis antara frekuensi yang digunakan, yaitu dua atau lebih sinyal dikatakan *orthogonal* (saling tegak lurus) jika saling bebas satu sama lain.

Bila dilihat dari gambar 4.7, dimana bentuk masukan data OFDM merupakan bentuk aliran data yang telah dibagi dengan metode pembagian sinyal yang masih dalam satu jalur frekuensi data, yang merupakan suatu aliran data yang masih *orthogonal*, yang kemudian akan dipecah menjadi data frekuensi yang lebih kecil pada setiap jalur frekuensi sebagai bentuk dari keluarannya. Data yang sudah dibagi dalam bentuk kecil ini akan dipakai sebagai aliran pembangkit data pada pemetaan dasar pembentukan kostelasi modulasi QPSK.

4.2.8. Data AWGN

input_awgn <115520x1 do		output_awgn <115520x1 do	
	1		1
1	0.0078 - 0.0234i	1	0.0082 - 0.0233i
2	0.0413 + 0.0324i	2	0.0409 + 0.0325i
3	-0.0389 - 0.0067i	3	-0.0388 - 0.0067i
4	-0.0161 - 0.0401i	4	-0.0164 - 0.0403i
5	0.0385 - 0.0137i	5	0.0384 - 0.0141i
6	-0.0303 + 0.0590i	6	-0.0301 + 0.0590i
7	-0.0004 + 0.0019i	7	-0.0004 + 0.0019i
8	0.0141 - 0.0271i	8	0.0144 - 0.0270i
9	0.0033 + 0.0205i	9	0.0033 + 0.0206i
10	0.0077 - 0.0050i	10	0.0080 - 0.0049i
11	-0.0163 - 0.0196i	11	-0.0161 - 0.0193i
12	-0.0076 - 0.0068i	12	-0.0078 - 0.0068i
13	-0.0074 + 0.0130i	13	-0.0076 + 0.0133i
14	0.0223 + 0.0007i	14	0.0226 + 0.0009i
15	-0.0223 - 0.0124i	15	-0.0224 - 0.0123i
16	0.0438 + 0.0309i	16	0.0439 + 0.0312i
17	0.0076 + 0.0262i	17	0.0078 + 0.0262i
18	-0.0077 - 0.0448i	18	-0.0077 - 0.0451i
19	-0.0911 - 0.0217i	19	-0.0912 - 0.0222i
20	0.0233 + 0.0364i	20	0.0235 + 0.0364i
21	0.0509 - 0.0224i	21	0.0513 - 0.0222i
22	0.0193 - 0.0152i	22	0.0193 - 0.0153i
23	-0.0006 + 0.0051i	23	-0.0009 + 0.0053i
24	-0.0271 + 0.0564i	24	-0.0272 + 0.0565i
25	-0.0213 - 0.0054i	25	-0.0217 - 0.0055i
26	0.0381 - 0.0020i	26	0.0384 - 0.0017i
27	-0.0290 - 0.0531i	27	-0.0290 - 0.0531i

Gambar 4.8 *Input dan Output* data AWGN

Dari hasil *dispreeding* (penyebaran data) atau perpecahan data frekuensi yang lebih kecil, kemudian data tersebut akan dikirimkan melalui kanal AWGN sehingga data informasi akan bercampur dengan *noise* yang mengalami *error*. Kemudian pada sisi *receiver* sinyal informasi yang bercampur *noise* tersebut akan dibagi lagi menjadi 2 fase kembali sebelum masuk ke proses *dispreeding* dengan kode yang sama seperti digunakan pada sisi pemancar.

Dari proses AWGN tersebut, telah dilakukan proses pengacakan pada setiap jalur frekuensi, sehingga nantinya akan terbentuk *noise-noise* yang akan merata pada lebar pita frekuensi. Kondisi seperti ini terlihat pada bentuk keluaran data AWGN yang sudah teracak-acak dari bentuk masukannya.

4.2.9. Data Demodulasi OFDM

input_ofdm_demod <115520		output_ofdm_demod <6931	
	1		1
1	0.0082 - 0.0233i	1	0.4997 - 0.4934i
2	0.0409 + 0.0325i	2	0.5005 - 0.4964i
3	-0.0388 - 0.0067i	3	0.4989 - 0.5027i
4	-0.0164 - 0.0403i	4	-0.5039 + 0.4931i
5	0.0384 - 0.0141i	5	0.5003 + 0.4929i
6	-0.0301 + 0.0590i	6	0.5032 - 0.5010i
7	-0.0004 + 0.0019i	7	0.5017 - 0.4972i
8	0.0144 - 0.0270i	8	0.4998 - 0.4991i
9	0.0033 + 0.0206i	9	0.5023 - 0.5041i
10	0.0080 - 0.0049i	10	0.4997 + 0.5036i
11	-0.0161 - 0.0193i	11	-0.5028 - 0.4983i
12	-0.0078 - 0.0068i	12	-0.5071 - 0.4978i
13	-0.0076 + 0.0133i	13	0.4999 - 0.4968i
14	0.0226 + 0.0009i	14	0.5003 - 0.4986i
15	-0.0224 - 0.0123i	15	0.4946 - 0.5029i
16	0.0439 + 0.0312i	16	-0.4984 + 0.4969i
17	0.0078 + 0.0262i	17	-0.4984 - 0.4999i
18	-0.0077 - 0.0451i	18	0.5010 - 0.5001i
19	-0.0912 + 0.0222i	19	-0.5010 - 0.5053i
20	0.0235 + 0.0364i	20	0.5002 + 0.5006i
21	0.0513 - 0.0222i	21	0.5024 - 0.4976i
22	0.0193 - 0.0153i	22	0.4998 - 0.4978i
23	-0.0009 + 0.0053i	23	0.5008 - 0.4985i
24	-0.0272 + 0.0565i	24	0.5008 - 0.5015i
25	-0.0217 - 0.0055i	25	0.5031 - 0.5007i
26	0.0384 - 0.0017i	26	-0.5071 - 0.4954i
27	-0.0290 - 0.0531i	27	-0.4983 - 0.5020i

Gambar 4.9 *Input* dan *Output* data Demodulasi OFDM

Proses yang terjadi pada data demodulasi OFDM sisi penerima adalah kebalikan dari proses yang dilakukan pada modulasi OFDM sisi pemancar. Pada penerima, pengaruh kanal akan diabaikan, berupa gelombang digital dan kemudian akan dikonversi kembali kedalam bentuk simbol dengan menggunakan FFT, FFT merupakan suatu metoda praktis untuk memperbaiki data dari pembawa (*carrier*) yang *overlap*.

Kemudian data dari demodulasi OFDM yang merupakan bentuk proses dari pemisahan bagian *in-phas*a dan kwadratur dari hasil pembagian sinyal *pilot*, akan dilakukan proses interpolasi agar didapatkan respon kanal bagian *in-phas*a dan kuadratur. Kemudian untuk bagian keluarannya akan digabungkan kembali dengan model pembagian sinyal *pilot*.

4.2.10. Data IQ Demapper

input_iq_demap <69312x1 do		output_iq_demap <138624x1 c	
1	2	1	2
1	0.4997 - 0.4934i	1	0
2	0.5005 - 0.4964i	2	0
3	0.4989 - 0.5027i	3	0
4	-0.5039 + 0.4931i	4	0
5	0.5003 + 0.4929i	5	0
6	0.5032 - 0.5010i	6	0
7	0.5017 - 0.4972i	7	1
8	0.4998 - 0.4991i	8	1
9	0.5023 - 0.5041i	9	0
10	0.4997 + 0.5036i	10	1
11	-0.5028 - 0.4983i	11	0
12	-0.5071 - 0.4978i	12	0
13	0.4999 - 0.4968i	13	0
14	0.5003 - 0.4986i	14	0
15	0.4946 - 0.5029i	15	0
16	-0.4984 + 0.4969i	16	0
17	-0.4984 - 0.4999i	17	0
18	0.5010 - 0.5001i	18	0
19	-0.5010 - 0.5053i	19	0
20	0.5002 + 0.5006i	20	1
21	0.5024 - 0.4976i	21	1
22	0.4998 - 0.4978i	22	0
23	0.5008 - 0.4985i	23	1
24	0.5008 - 0.5015i	24	0
25	0.5031 - 0.5007i	25	0
26	-0.5071 - 0.4954i	26	0
27	-0.4983 - 0.5020i	27	0

Gambar 4.10 *Input dan Output IQ Demapper*

Pada pemrosesan data *demapping* atau pemetan kembali oleh modulasi QPSK merupakan perubahan simbol dari hasil keluaran FFT menjadi bentuk biner. Dengan setiap satu simbol akan dijadikan dua bit data biner hal ini mengacu pada pemetaan modulasi QPSK, dimana setiap satu simbol terdiri dari dua buah data biner. Dari penggabungan pembagian dari sinyal *pilot* akan dikembalikan kekeluaran data IQ *demapper*, seperti pada masukan IQ *mapper*.

4.2.11. Data Deinterleaver

input_deinterleaver <138624x		output_deinterleaver <138624	
1	2	1	2
1	0	1	0
2	0	2	1
3	0	3	1
4	0	4	0
5	0	5	0
6	0	6	1
7	1	7	1
8	1	8	1
9	0	9	1
10	1	10	0
11	0	11	0
12	0	12	0
13	0	13	0
14	0	14	0
15	0	15	1
16	0	16	0
17	0	17	0
18	0	18	1
19	0	19	1
20	1	20	0
21	1	21	0
22	0	22	0
23	1	23	0
24	0	24	0
25	0	25	0
26	0	26	0
27	0	27	1

Gambar 4.11 *Input dan Output data Deinterleaver*

Aliran data dari *deinterleaver* sejatinya merupakan metode untuk mengumpulkan kembali bit-bit *codeword* yang telah disebar, menjadi deretan *codeword* yang sebenarnya, sebagaimana yang didapat pada *codeword* sebelum memasuki *interleaver*. Dimana bit akan mengisi matriks kolom per kolom. Kemudian untuk *output deinterleaver* berupa matriks baris.

4.2.12 Data Upsample

input_upsample <138624x1 d...		output_upsample <277248x1 d...	
	1	2	
1	0		1
2	1		2
3	1		3
4	0		4
5	0		5
6	1		6
7	1		7
8	1		8
9	1		9
10	0		10
11	0		11
12	0		12
13	0		13
14	0		14
15	1		15
16	0		16
17	0		17
18	1		18
19	1		19
20	0		20
21	0		21
22	0		22
23	0		23
24	0		24
25	0		25
26	0		26
27	1		27

Gambar 4.12 Input dan Output data Upsample

Pada data tabel *upsamle* ini yang merupakan bentuk aplikasi dari data pertama yang telah digantikan dengan sisipan diantara 0 yang berupa data sampel masukan yang akan digantikan. Bentuk perubahan ini terlihat bahwa masukan data mulai dari nomer 6 yang berupa data '1' akan kembali ke nomer 6 yang berupa '0' pada keluran *upsampel*.

4.2.13. Data Add

input_add <277248x1 double:			output_add <277248x1 double:		
	1	2		1	2
1	0		1	0	
2	0		2	0	
3	1		3	1	
4	0		4	1	
5	1		5	1	
6	0		6	1	
7	0		7	0	
8	0		8	0	
9	0		9	0	
10	0		10	1	
11	1		11	1	
12	0		12	1	
13	1		13	1	
14	0		14	0	
15	1		15	1	
16	0		16	1	
17	1		17	1	
18	0		18	0	
19	0		19	0	
20	0		20	1	
21	0		21	0	
22	0		22	1	
23	0		23	0	
24	0		24	1	
25	0		25	0	
26	0		26	0	
27	0		27	0	

Gambar 4.13 *Input dan Output data Add*

Data parameter *add* berfungsi sebagai penjumlahan data dari dua data yang tercabang untuk disatukan menjadi satu jalur kanal. Bentuk dari pengkodean ini menggunakan pergeseran memori dalam proses penjumlahan data. Oleh karena itu pada pergeseran datanya akan terjadi keterlambatan yang cukup besar, pergeseran data dapat dilihat dari masukan mulai dari nomer 2 yang akan digeser kekeluaran mulai dari nomer 19.

4.2.14. Data Convolutional Decoder

input_con_dec <277248x1 double:			output_con_dec <231040x1 double:		
	1	2		1	2
1	0		1	0	
2	0		2	0	
3	1		3	0	
4	1		4	0	
5	1		5	0	
6	1		6	0	
7	0		7	0	
8	0		8	0	
9	0		9	0	
10	1		10	1	
11	1		11	0	
12	1		12	0	
13	1		13	0	
14	0		14	0	
15	1		15	1	
16	1		16	1	
17	1		17	0	
18	0		18	1	
19	0		19	1	
20	1		20	1	
21	0		21	1	
22	1		22	0	
23	0		23	0	
24	1		24	0	
25	0		25	0	
26	0		26	0	
27	0		27	1	

Gambar 4.14 *Input dan Output data Convolutional Decoder*

Kode konvolusi *dekoder* menggunakan metode algoritma *viterbi*. Dimana *Convolutional code* dengan *viterbi decoder* adalah sebuah teknik *Forward Error Correction* (FEC) yang sesuai dengan sebuah sinyal yang berada dalam saluran AWGN.

Cara kerja dari algoritma *viterbi* sendiri adalah dengan mencari jarak minimum terkecil yang memiliki nilai akumulasi *error* matriks yang terkecil. Untuk mendapatkan nilai akumulasi *error matrix* yang terkecil dari keluaran konvolusi *decoder* diperlukan data rentang *state output* dari konvolusi *enkoder* tersebut.

Dapat dilihat pada gambar 4.14 bahwa pada *convolutional decoder* yang merupakan data dari pengembalian data asal, dengan nilai keterlambatan yang sangat tergantung dengan nilai data asal. Untuk bentuk pergeserannya dapat dilihat pada masukan mulai dari nomer 2 yang digeser kekeluaran data dari mulai nomer 17

4.2.15 Data Decoder

input_decoder <231040x1 do		output_decoder <103968x1 dc	
1	2	1	2
1	0	1	1
2	0	2	1
3	0	3	0
4	0	4	1
5	0	5	0
6	0	6	1
7	0	7	0
8	0	8	0
9	0	9	1
10	1	10	0
11	0	11	1
12	0	12	1
13	0	13	1
14	0	14	0
15	1	15	1
16	1	16	0
17	0	17	1
18	1	18	0
19	1	19	1
20	1	20	0
21	1	21	0
22	0	22	0
23	0	23	0
24	0	24	1
25	0	25	0
26	0	26	0
27	1	27	0

Gambar 4.15 *Input dan Output data Decoder*

Dalam proses *decoding*, nilai yang akan mempengaruhi performansi adalah banyaknya memori data dalam menentukan *output* blok *decoder*. Jadi semakin banyak memori data maka akan didapat nilai *output* yang memenuhi syarat $cH^T = 0$, dimana c merupakan *codeword* dan H merupakan data matriks yang akan menyeimbangkan bentuk data yang akan dikirimkan.

Sehingga akan terjadi pengelompokan deret bit keluaran dari blok *viterbi decoder* yang dimana memerlukan memori data pada proses *decoding*. Dapat dilihat pada gambar 4.15, telah didapati pergeseran dari masukan mulai dari nomer 9 yang akan bergeser kekeluaran mulai dari nomer 18, dan terjadi keterlambatan yang disebabkan dari olah proses memori data.

4.2.16. Data Derandomization

input_derandom <103968x1 d			output_derandom <101080x1		
	1	2		1	2
1	1	1	1	0	1
2	1	1	2	1	1
3	0	1	3	0	1
4	1	1	4	0	1
5	0	1	5	0	1
6	1	1	6	1	1
7	0	1	7	0	1
8	0	1	8	1	1
9	1	1	9	0	1
10	0	1	10	0	1
11	1	1	11	1	1
12	1	1	12	0	1
13	1	1	13	1	1
14	0	1	14	0	1
15	1	1	15	0	1
16	0	1	16	1	1
17	1	1	17	1	1
18	0	1	18	1	1
19	1	1	19	0	1
20	0	1	20	0	1
21	0	1	21	0	1
22	0	1	22	1	1
23	0	1	23	0	1
24	1	1	24	0	1
25	0	1	25	0	1
26	0	1	26	1	1
27	0	1	27	1	1

Gambar 4.16 Input dan Output data Derandomization

Bentuk dari data *derandomization* merupakan alur penyusunan kembali bit yang sebelumnya telah diacak. Pada dasarnya penyusunan kembali bit diperlukan untuk mendapatkan kembali data informasi yang sebelumnya sudah diacak sehingga akan didapati kembali data pergeseran dari masukan menuju kekeluaran. Model pergeserannya telah diperlihatkan pada gambar 4.16, dengan masukan mulai dari nomer 3 akan digeser ke data keluaran nomer 5.

4.3. Analisis Pengujian Kanal AWGN

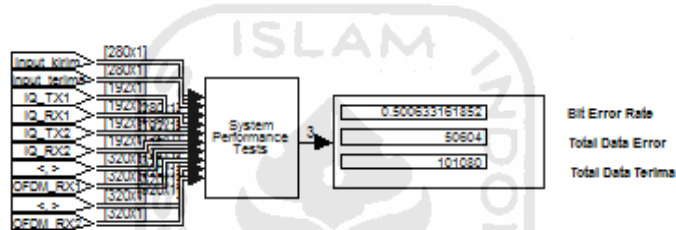
Dalam Analisis simulasi kanal AWGN ini bertujuan untuk membandingkan perbedaan antara keluaran dari BER dengan beberapa nilai SNR mulai dari 0 sampai 60 dB dengan kenaikan masing-masing 10 dB.

4.3.1. Model pengujian SNR dari 0 dB s/d 60 dB pada WiMAX ber-MIMO

Dengan batas-batas pengujian :

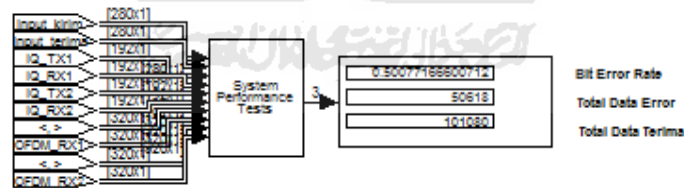
- Total data terima : 101080
- Paket data yang dikirim : 360
- *Input* daya sinyal : 0.080325450151575 watt

1. SNR : 0 dB, *Bit Error Rate* : 0.500633161852



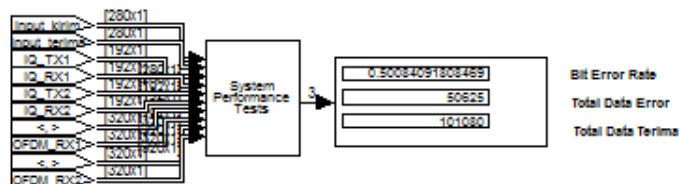
Gambar 4.17 Data terima untuk SNR 0 dB

2. SNR : 1 dB, *Bit Error Rate* : 0.50077166600712



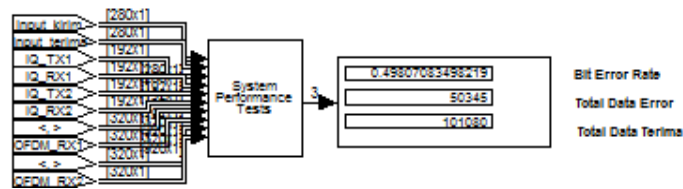
Gambar 4.18 Data terima untuk SNR 1 dB

3. SNR : 2 dB, *Bit Error Rate* : 0.5008409180869



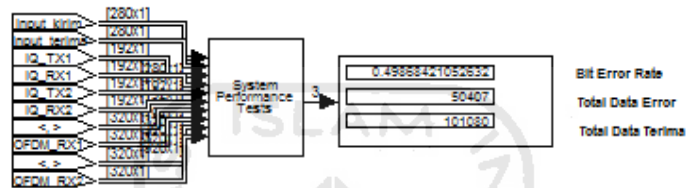
Gambar 4.19 Data terima untuk SNR 2 dB

4. SNR : 3 dB, *Bit Error Rate* : 0.49807083498219



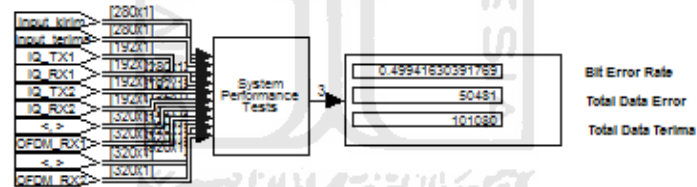
Gambar 4.20 Data terima untuk SNR 3 dB

5. SNR : 4 dB, *Bit Error Rate* : 0.49868421052632



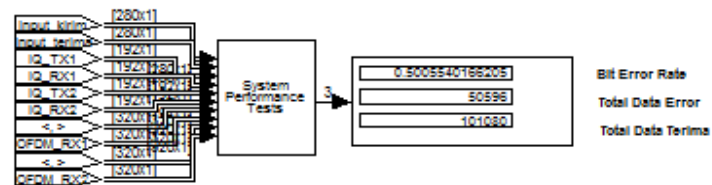
Gambar 4.21 Data terima untuk SNR 4 dB

6. SNR : 5 dB, *Bit Error Rate* : 0.49941630391769



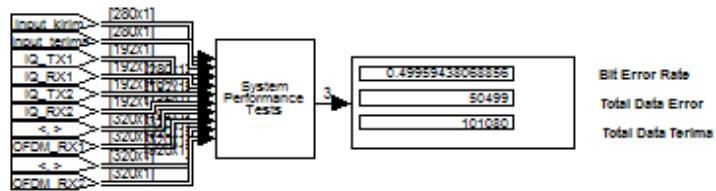
Gambar 4.22 Data terima untuk SNR 5 dB

7. SNR : 6 dB, *Bit Error Rate* : 0.5005540166205



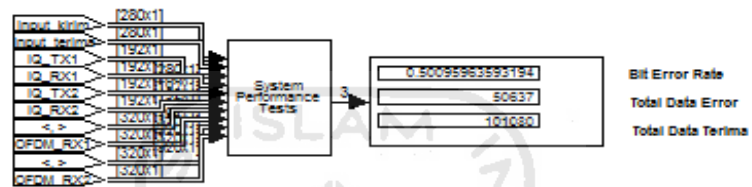
Gambar 4.23 Data terima untuk SNR 6 dB

8. SNR : 7 dB, *Bit Error Rate* : 0.49959428068856



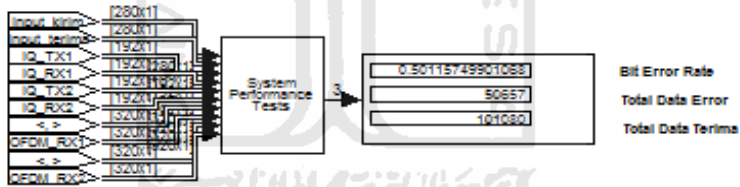
Gambar 4.24 Data terima untuk SNR 7 dB

9. SNR : 8 dB, *Bit Error Rate* : 0.50095963593194



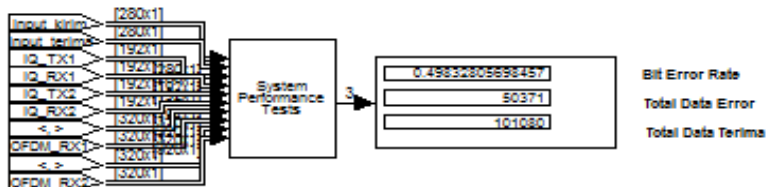
Gambar 4.25 Data terima untuk SNR 8 dB

10. SNR : 9 dB, *Bit Error Rate* : 0.50115749901068



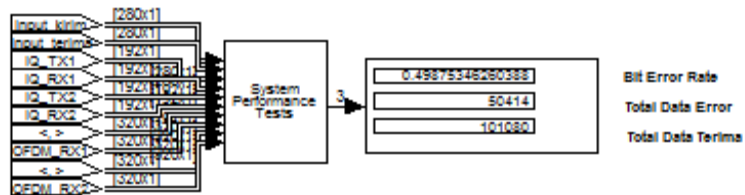
Gambar 4.26 Data terima untuk SNR 9 dB

11. SNR : 10 dB, *Bit Error Rate* : 0.49832805698457



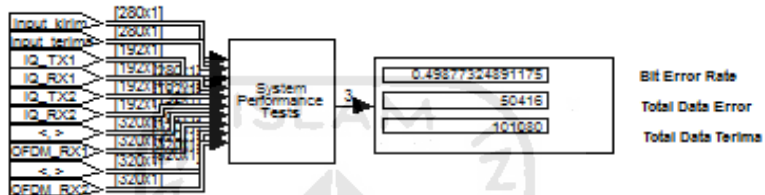
Gambar 4.27 Data terima untuk SNR 10 dB

12. SNR : 11 dB, *Bit Error Rate* : 0.49875346260388



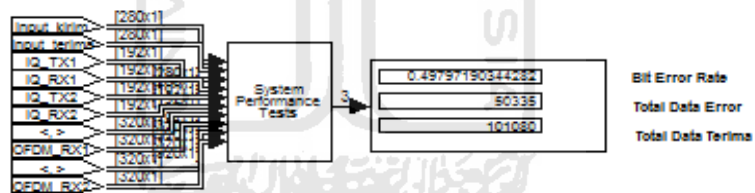
Gambar 4.28 Data terima untuk SNR 11 dB

13. SNR : 12 dB, *Bit Error Rate* : 0.49877324891175



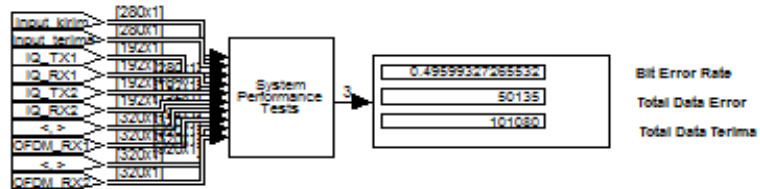
Gambar 4.29 Data terima untuk SNR 12 dB

14. SNR : 13 dB, *Bit Error Rate* : 0.49797190344282



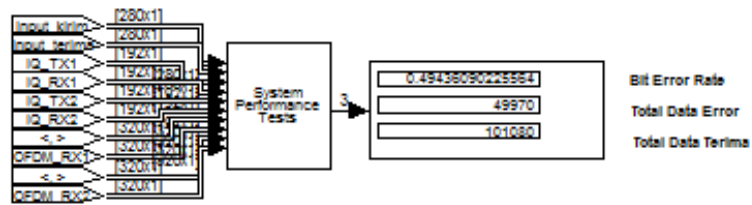
Gambar 4.30 Data terima untuk SNR 13 dB

15. SNR : 14 dB, *Bit Error Rate* : 0.49599327265532



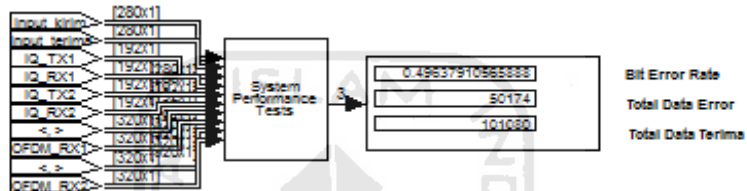
Gambar 4.31 Data terima untuk SNR 14 dB

16. SNR : 15 dB, *Bit Error Rate* : 0.49436090225564



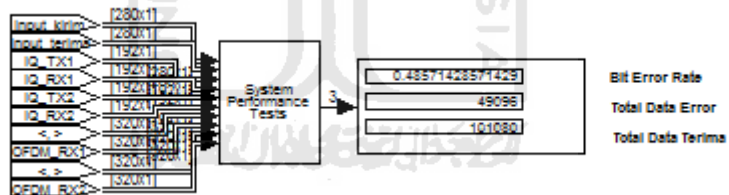
Gambar 4.32 Data terima untuk SNR 15 dB

17. SNR : 16 dB, *Bit Error Rate* : 0.49637910565888



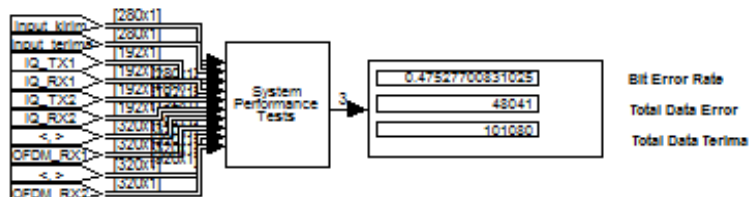
Gambar 4.33 Data terima untuk SNR 16 dB

18. SNR : 17 dB, *Bit Error Rate* : 0.48571428571429



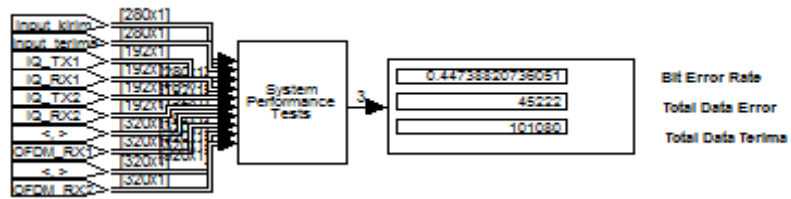
Gambar 4.34 Data terima untuk SNR 17 dB

19. SNR 18 dB, *Bit Error Rate* : 0.47527700831025



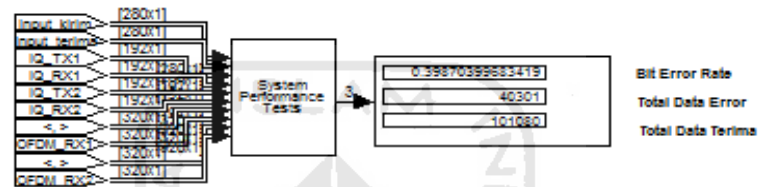
Gambar 4.35 Data terima untuk SNR 18 dB

20. SNR : 19 dB, *Bit Error Rate* : 0.44738820736051



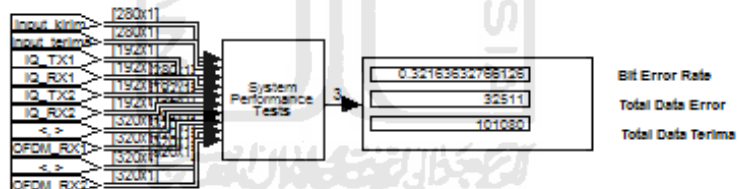
Gambar 4.36 Data terima untuk SNR 19 dB

21. SNR : 20 dB, *Bit Error Rate* : 0.39870399863419



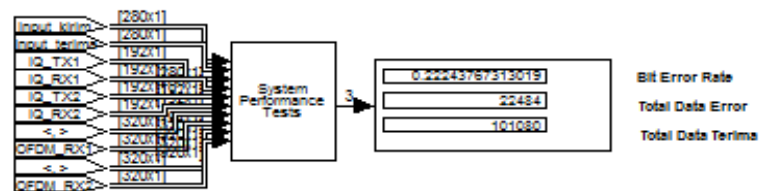
Gambar 4.37 Data terima untuk SNR 20 dB

22. SNR : 21 dB, *Bit Error Rate* : 0.32163632766126



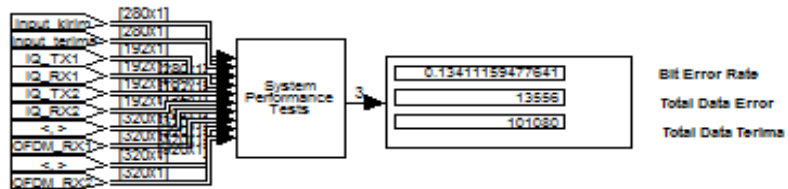
Gambar 4.38 Data terima untuk SNR 21 dB

23. SNR : 22 dB, *Bit Error Rate* : 0.22243767313019



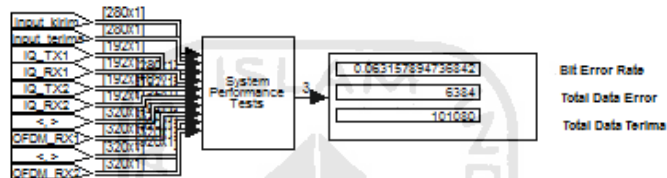
Gambar 4.39 Data terima untuk SNR 22 dB

24. SNR : 23 dB, *Bit Error Rate* : 0.13411189477641



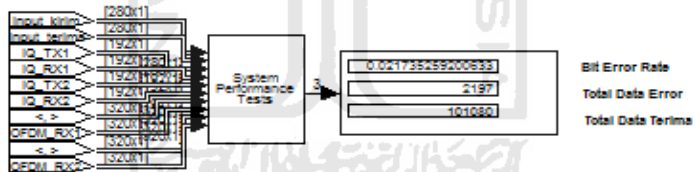
Gambar 4.40 Data terima untuk SNR 23 dB

25. SNR : 24 dB, *Bit Error Rate* : 0.063157894736842



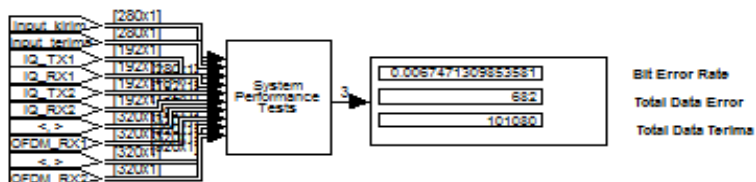
Gambar 4.41 Data terima untuk SNR 24 dB

26. SNR : 25 dB, *Bit Error Rate* : 0.021735259200633



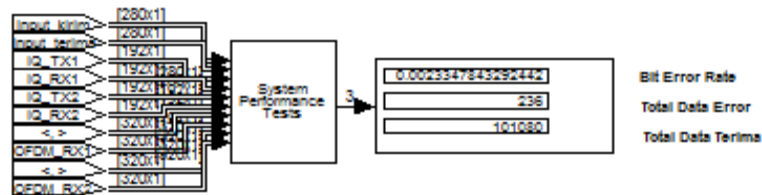
Gambar 4.42 Data terima untuk SNR 25 dB

27. SNR : 26 dB, *Bit Error Rate* : 0.0067471309853381



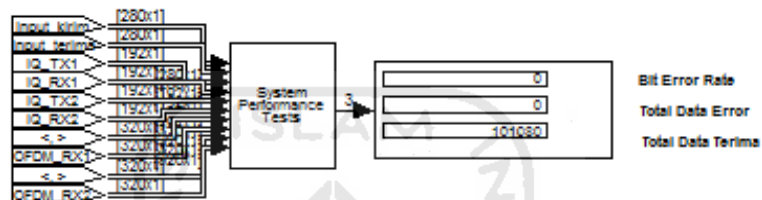
Gambar 4.43 Data terima untuk SNR 26 dB

28. SNR : 27 dB, *Bit Error Rate* : 0.0023347843292442



Gambar 4.44 Data terima untuk SNR 27 dB

29. Model SNR : 28 dB s/d 60 dB, *Bit Error Rate* : 0



Gambar 4.45 Data terima untuk SNR 28 dB s/d 60 dB

4.3.2. Model pemrograman untuk SNR vs BER

```
>> figure(1)
```

```
%WiMAX BER-MIMO%
```

```
>> SNR1=[0:60];
```

```
>> i=1:length(SNR1);
```

```
>> BER(i)=0;t=[0.500633162 0.500771666 0.500840918 0.498070835 0.498684211
0.499416304 0.500554017 0.499594281 0.500959636 0.501157499 0.498328057
0.498753463 0.498773249 0.497971903 0.495993273 0.494360902 0.496379106
0.485714286 0.475277008 0.447388207 0.3987039990.3216363280.222437673
0.134111595 0.063157895 0.0217352590.006747131 0.002334784 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
```

```
%WiMAX Tanpa MIMO%
```

```
>> SNR2=[0:60];
```

```
>> j=1:length(SNR2);
```

```
>> BER(j)=0;s=[0.50172140878512 0.50116739216462 0.49954491491888
0.49805104867432 0.50130589631975 0.50139493470518 0.50204788286506
0.50083102493075 0.4992876929165 0.49687376335576 0.49924812030075
```

```

0.49778393351801 0.50011871784725 0.50137514839731 0.49937673130194
0.49823901859913 0.49752671151563 0.49686387020182 0.48842500989315
0.47585081123862 0.44031460229521 0.36162445587653 0.24254056193114
0.11744163039177 0.047170557973882 0.015098952908587 0.0040265136525524
0.0010091017016225 0.0017807677087455 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0];

```

```
>> semilogy(t,'blue');
```

```
>> hold
```

```
Current plot held
```

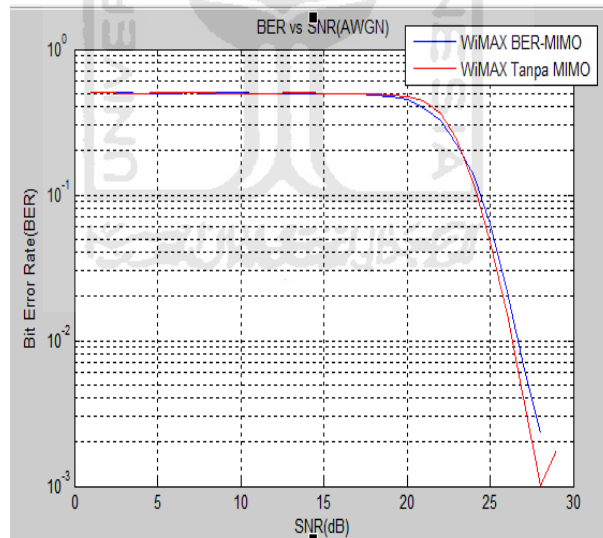
```
>> semilogy(s,'red');
```

```
>> grid
```

```
>> title('BER vs SNR(AWGN)');
```

```
>> xlabel('SNR(dB)');
```

```
>> ylabel('Bit Error Rate(BER)');
```



Gambar 4.46 Grafik data BER vs SNR (AWGN)

SNR yang merupakan tolak ukur kualitas dari suatu jaringan, dengan kekuatan derau (*noise level*). *Noise* merupakan sinyal listrik yang tidak diinginkan. Pada sinyal yang tidak diinginkan ini dalam suatu proses komunikasi, merupakan suatu faktor pembatas utama dalam sistem komunikasi data. Bila *noise* terjadi dalam suatu sistem komunikasi maka sistem komunikasi akan mengalami gangguan. Gangguan yang terjadi dapat menyebabkan proses komunikasi terganggu atau bahkan dapat memutuskan proses komunikasi.

Berdasarkan gambar 4.47, bentuk garis BER dari keluaran SNR yang berupa *thermal noise* atau *white noise*, merupakan suatu *noise* yang dibangkitkan oleh sumber diluar sistem komunikasi. Kemudian dari hasil data simulasi akan diperoleh analisis; ketika SNR = 28 dB, BER = 0, maka pada kondisi MIMO akan mempunyai kualitas kanal yang lebih baik, dibandingkan dengan yang bukan MIMO sebesar 29 dB, BER = 0. Itu artinya akan semakin besar pula kemungkinan jalur itu dipakai untuk lalu-lintas komunikasi data dan sinyal dalam kecepatan tinggi.

Dimana :

$$SNR = P_{signal}/P_{noise}$$

P= Power atau Daya

Contoh : SNR = Ps/Pn

$$= -16 \text{ dBm}/-39 \text{ dBm}$$

$$= 0.03 \text{ mW}/0.00013 \text{ mW}$$

$$= 230.77 \text{ mW} = 23 \text{ dB}$$

Karena nilai SNR diukur pada nilai pada satu benda, contoh pada suatu kabel berarti nilai-nilai yang mempengaruhi *P_{signal}* maupun *P_{noise}* adalah sama. Yaitu:

$$P = V^2/R$$

V = Tegangan

R = Hambatan

Untuk sinyal maka V dapat disetarakan dengan A (Amplitudo):

$$P = A^2/R$$

$$P_{signal} = (A_{signal})^2/R \text{ kabel}$$

$$P_{noise} = (A_{noise})^2/R \text{ kabel}$$

Karena R kabel bernilai sama, Maka :

$$\text{SNR} = (\text{Asignal}/\text{Anoise})^2$$

Atau dalam satuan *deciBel* (dB), yaitu satuan yang biasa untuk mengukur sinyal,

$$\text{SNRdb} = 10 \log \text{SNR}$$

$$= 10 \log (\text{Asignal}/\text{Anoise})^2$$

$$= 20 \log (\text{Asignal}/\text{Anoise})$$



BAB V

PENUTUP

5.1. KESIMPULAN

Berdasarkan perancangan sistem dan hasil simulasi yang didapat maka dalam pembuatan sistem “*Simulasi Pemodelan MIMO Spatial Multiplexing pada WiMAX 802.16d*” dapat disimpulkan beberapa hal, yaitu :

- a. Dengan memanfaatkan prinsip OFDM, dimana dapat meningkatkan kapasitas kanal dengan simbol yang dikirimkan *parallel*. Maka dari penggunaan teknik MIMO *Spatial Multiplexer* akan didapati kapasitas kanal yang besar, yakni dengan memecah aliran data berlaju tinggi menjadi sejumlah aliran *parallel*, masing-masing dengan laju yang lebih rendah dari aliran aslinya.
- b. Antara grafik keluaran SNR dari WiMAX ber-MIMO dan WiMAX tanpa MIMO, mempunyai model alur yang sama. Dengan MIMO yang mempunyai kualitas kanal jauh lebih baik sebesar 28 dB, dibandingkan dengan yang bukan MIMO sebesar 29 dB.

5.2. SARAN

Untuk mengembangkan sistem di masa yang akan datang maka dapat disarankan beberapa hal sebagai berikut :

1. Perlu adanya analisis lebih lanjut mengenai perbandingan kinerja antara WiMAX dengan sistem komunikasi nirkabel yang lain. Sehingga akan terlihat lebih jelas lagi mengenai ketangguhan suatu sistem komunikasi WiMAX.
2. Perlu dibahas lebih lanjut mengenai varian pengkodean WiMAX yang lain, sehingga akan terlihat perbedaan bentuk analisis kerjanya dengan varian parameter yang lainnya. Misalnya dengan metode SUI.



DAFTAR PUSTAKA

1. Nugroho Agus Prasetyo, “Simulasi Unjuk Kerja Simulasi WiMAX”, TA/SEKJUR/TE/2008/082, Universitas Islam Indonesia, Yogyakarta, 2008.
2. Andika Wisnu Wardana, “Simulasi *Baseband* Prosesor WiMAX (Bagian Pengirim)”, TA/SEKJUR/TE/2010/013, Universitas Islam Indonesia, Yogyakarta, 2010.
3. Rustam Efendi, “*LIMITED FEEDBACK PRECODING DAN MIMO SPATIAL MULTIPLEXING UNTUK APLIKASI 802.16e*”, Tesis, STT Telkom, Bandung, 2007.
4. Proffesor Roberto Cristi, “*Wireless Communications with Matlab and Simulink: IEEE 802.16 (WiMax) Physical Layer*”, Dept of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943 August 2009.
5. Gunawan Wibisono, Gunadi Dwi Hartanto, “WiMAX, Teknologi *Broadband Wireless Access* (BWA) Kini dan Masa Depan”, INFORMATIKA, Bandung, 2006
6. WAHANA KOMPUTER, “Kupas Tuntas Teknologi WiMAX”, ANDI, Yogyakarta, 2009

PEMROGRAMAN DENGAN MATLAB 7.8 2009a

```
clear;
clc;
%-----
% Part 1
% Test Data for the WiMAX PHY Layer from Document 'ETSI TS 102 177 V1.3.2(2006-03)'
%-----
input_data = 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');
randomized_data = sscanf(['D4 BA A1 12 F2 74 96 30 27 D4 88 9C 96 E3 A9 52 B3 15 AB FD
92 53 07 32 C0 62 48 F0 19 22 E0 91 62 1A C1'], '%x');
rs_encoded_data = sscanf(['49 31 40 BF D4 BA A1 12 F2 74 96 30 27 D4 88 9C 96 E3 A9 52 B3
15 AB FD 92 53 07 32 C0 62 48 F0 19 22 E0 91 62 1A C1 00'], '%x');
conv_encoded_data = sscanf(['3A 5E E7 AE 49 9E 6F 1C 6F C1 28 BC BD AB 57 CD BC CD E3
A7 92 CA 92 C2 4D BC 8D 78 32 FB BF DF 23 ED 8A 94 16 27 A5 65 CF 7D 7A 45 B8 09
CC'], '%x');
interleaved_data = sscanf(['77 FA 4F 17 4E 3E E6 70 E8 CD 3F 76 90 C4 2C DB F9 B7 FB 43 6C
F1 9A BD ED 0A 1C D8 1B EC 9B 30 15 BA DA 31 F5 50 49 7D 56 ED '.....
'B4 88 CC 72 FC 5C'], '%x');
%-----
% Part 2
% Forward Error Correction (FEC):
% Generator and Primitive Polynomials
% Generator Polynomial is a Galois Field Array [1 59 13 104 189 68 209 30 8 163 65 41 229 98 50 36 59]
% Primitive Polynomial is [1 0 0 0 1 1 1 0 1]
%-----
Gen_Poly = [1 gf(2,8)^0];
for Index = 1 : 15
Gen_Poly = conv (Gen_Poly, [1 gf(2,8)^ Index]);
end
Prim_Poly = [1 0 0 0 1 1 1 0 1];
%-----
% Interleaver
% Ncbps 384, 768, 1152 for QPSK, 16-QAM, 64-QAM Respectively
% Ncpc 2, 4, 6 for QPSK, 16-QAM, 64-QAM
```

```
Ncbps = 384;
Ncpc = 2;
k = 0 : Ncbps - 1;
mk = (Ncbps/12) * mod (k,12) + floor (k/12);
s = ceil (Ncpc/2);
jk = s * floor (mk/s) + mod (s, mk + Ncbps - floor (12 * mk/Ncbps));
[s, int_idx] = sort (jk);
```

```
%End Part 2
```

```
%-----
```

```
%QPSK Modulator
```

```
Ry = ones(2,1) * [+1 -1];
```

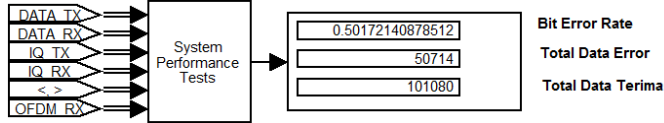
```
Iy = [+1 -1]*ones (1,2);
```

```
qamconst = Ry + j * Iy;
```

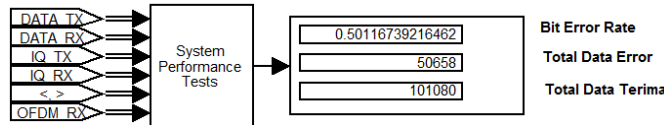
```
qamconst = qamconst (/sqrt (2));
```



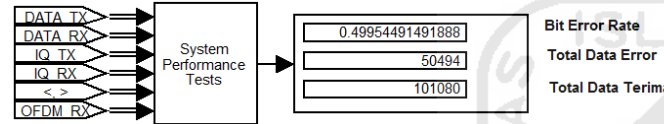
PENGUJIAN SNR DARI 0 dB s/d 60 dB PADA WiMAX TANPA MIMO



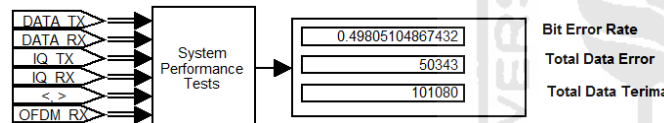
SNR : 0 dB, *Bit Error Rate* : 0.50172140878512



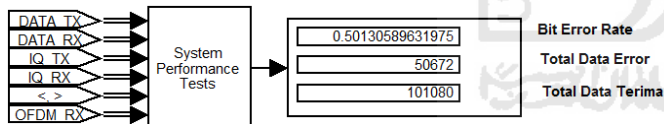
SNR : 1 dB, *Bit Error Rate* : 0.50116739216462



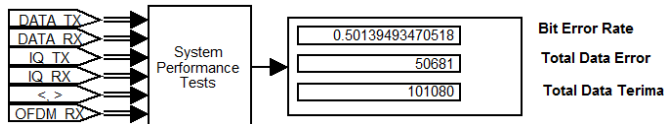
SNR : 2 dB, *Bit Error Rate* : 0.49954491491888



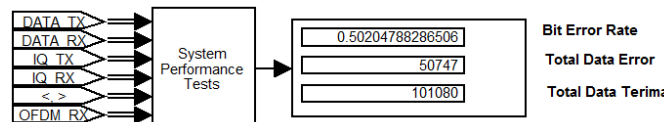
SNR : 3 dB, *Bit Error Rate* : 0.49805104867432



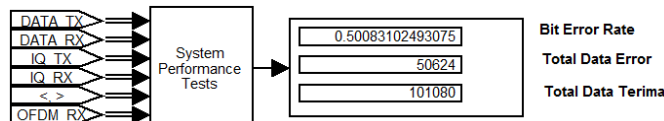
SNR : 4 dB, *Bit Error Rate* : 0.50130589631975



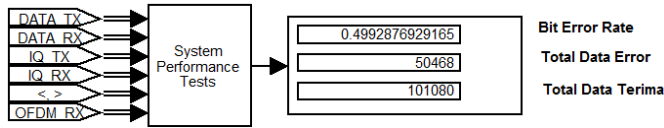
SNR : 5 dB, *Bit Error Rate* : 0.50139493470518



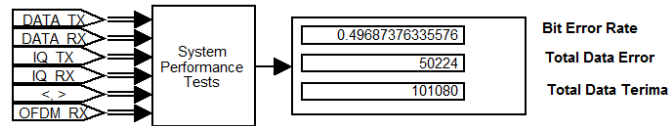
SNR : 6 dB, *Bit Error Rate* : 0.50204788286506



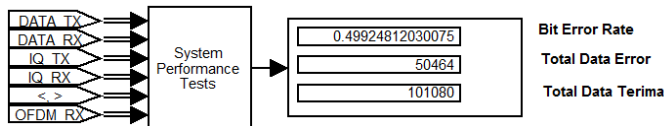
SNR : 7 dB, *Bit Error Rate* : 0.50083102493075



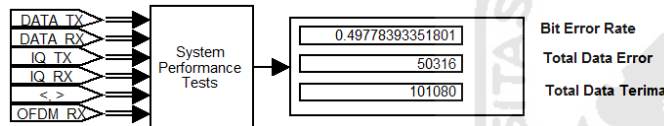
SNR : 8 dB, *Bit Error Rate* : 0.4992876929165



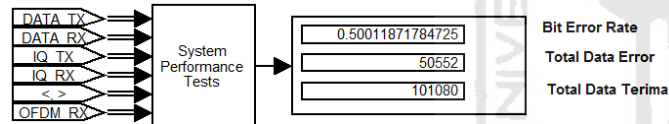
SNR : 9 dB, *Bit Error Rate* : 0.49687376335576



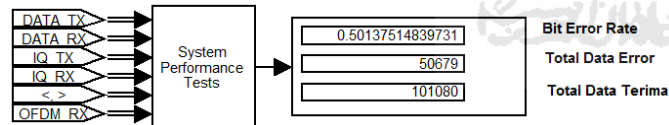
SNR : 10 dB, *Bit Error Rate* : 0.49924812030075



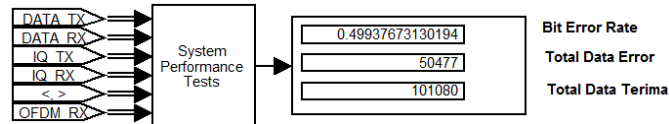
SNR : 11 dB, *Bit Error Rate* : 0.49778393351801



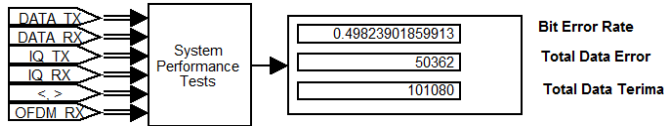
SNR : 12 dB, *Bit Error Rate* : 0.50011871784725



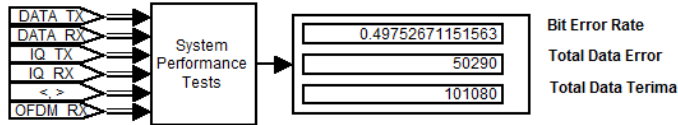
SNR : 13 dB, *Bit Error Rate* : 0.50137514839731



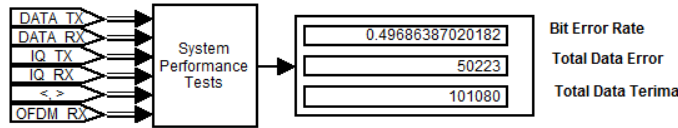
SNR : 14 dB, *Bit Error Rate* : 0.49937673130194



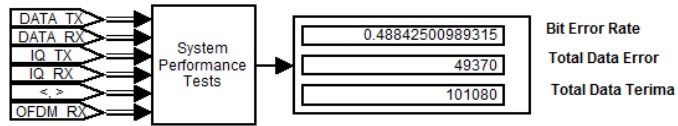
SNR : 15 dB, *Bit Error Rate* : 0.49823901859913



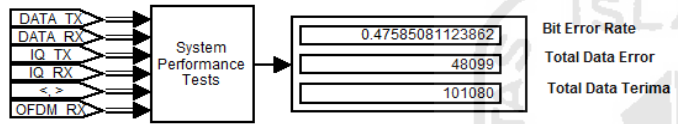
SNR : 16 dB, *Bit Error Rate* : 0.49752671151563



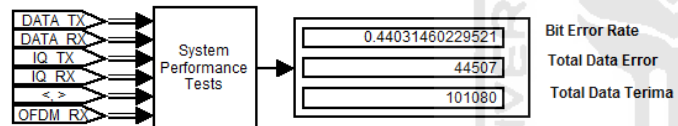
SNR : 17 dB, *Bit Error Rate* : 0.49686387020182



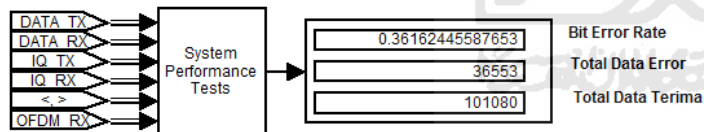
SNR : 18 dB, *Bit Error Rate* : 0.48842500989315



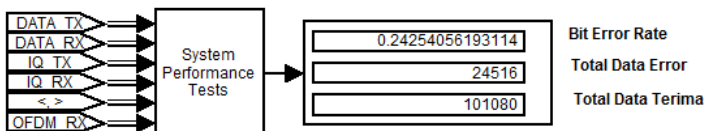
SNR : 19 dB, *Bit Error Rate* : 0.47585081123862



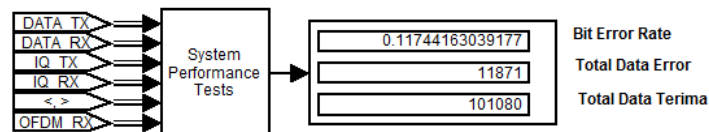
SNR : 20 dB, *Bit Error Rate* : 0.44031460229521



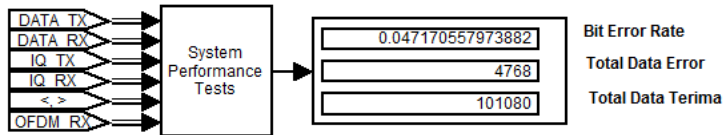
SNR : 21 dB, *Bit Error Rate* : 0.36162445587653



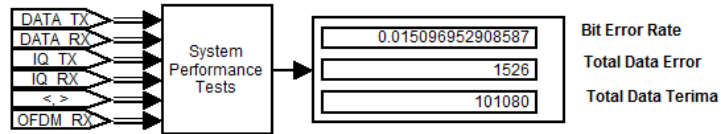
SNR : 22 dB, *Bit Error Rate* : 0.24254056193114



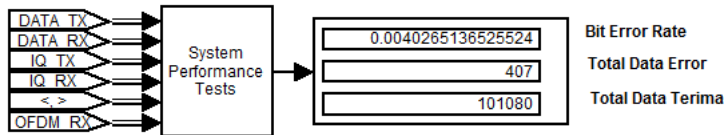
SNR : 23 dB, *Bit Error Rate* : 0.11744163039177



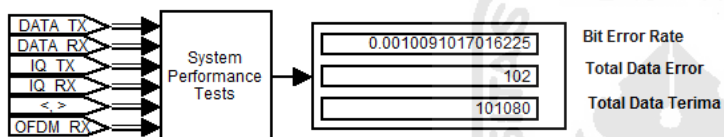
SNR : 24 dB, *Bit Error Rate* : 0.047170557973882



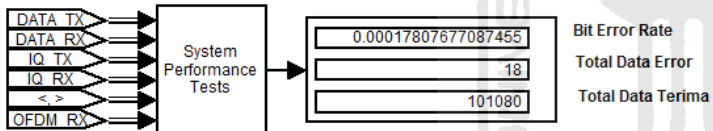
SNR : 25 dB, *Bit Error Rate* : 0.015096952908587



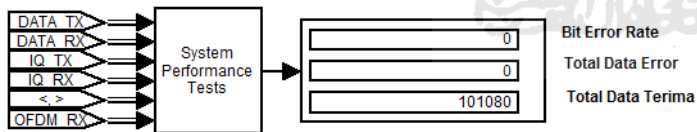
SNR : 26 dB, *Bit Error Rate* : 0.0040265136525524



SNR : 27 dB, *Bit Error Rate* : 0.0010091017016225



SNR : 28 dB, *Bit Error Rate* : 0.00017807677087455



SNR : 29-60 dB, *Bit Error Rate* : 0