

**SPATIAL MULTIPLEXING MIMO OFDM  
DENGAN MODEL KANAL SVD  
UNTUK MOBILE WIMAX**

**SKRIPSI**

**Oleh:**

**IGURA PRABU SENKY**

**04 04 03 707X**



**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
GENAP 2007/ 2008**

**SPATIAL MULTIPLEXING MIMO OFDM  
DENGAN MODEL KANAL SVD  
UNTUK MOBILE WIMAX**

**SKRIPSI**

**Oleh:**

**I GURA PRABU SENKY**

**04 04 03 707X**



**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN  
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
GENAP 2007/2008**

## **PERNYATAAN KEASLIAN SKRIPSI**

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

### **SPATIAL MULTIPLEXING MIMO OFDM DENGAN MODEL KANAL SVD UNTUK MOBILE WIMAX**

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada pendidikan Sarjana S1 Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan duplikasi dari skripsi yang telah dipublikasikan dan atau pernah dipakai untuk skripsi di lingkungan universitas Indonesia maupun di perguruan tinggi atau instansi manapun, kecuali bagian yang sumber informasinya telah dicantumkan sebagaimana mestinya.

Depok, 8 Mei 2008

I Gura Prabu Senky

NPM 040303707X

# LEMBAR PERSETUJUAN

Skripsi dengan judul :

**SPATIAL MULTIPLEXING MIMO OFDM  
DENGAN MODEL KANAL SVD  
UNTUK MOBILE WIMAX**

dibuat untuk melengkapi persyaratan kurikulum di Departemen Elektro Fakultas Teknik Universitas Indonesia. Tugas ini disetujui untuk diajukan dalam sidang skripsi.

Depok, 8 Mei 2008

Dosen Pembimbing,

Dr. Ir. Muhamad Asvial, M.Eng

NIP 132 094 574

## UCAPAN TERIMA KASIH

Puji dan syukur ke hadirat Allah SWT atas berkat dan rahmat-Nya sehingga skripsi ini dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada:

**Dr. Ir. Muhamad Asvial, M.Eng**

Selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan saran, bimbingan, pengarahan, dan kemudahan dalam menyelesaikan skripsi ini. Dan juga kedua orangtua serta kakak, adik, dan family saya tercinta yang telah mendukung baik secara moril maupun materil. Terakhir, ucapan terima kasih untuk teman-teman Elektro angkatan 2004 atas segala kebersamaannya selama ini.

I Gura Prabu Senky NPM 04 04 03 707 X Departemen Teknik Elektro	Pembimbing Dr. Ir. Muhamad Asvial, M.Eng
<b>SPATIAL MULTIPLEXING MIMO OFDM          DENGAN MODEL KANAL SVD UNTUK MOBILE WIMAX</b>	
<p><b>ABSTRAK</b></p> <p>Teknologi dalam bidang komunikasi nirkabel dalam kurun waktu terakhir menunjukkan perkembangan yang sangat pesat. Perkembangan tersebut mengarah kepada pemenuhan akan kebutuhan kapasitas yang tinggi, serta penggunaan bandwidth yang efisien dan kemampuan dalam menghadapi gangguan selama transmisi yang berujung pada minimnya eror selama pentransmisi data. Salah satu teknologi yang dapat memenuhi semua kebutuhan diatas adalah teknologi MIMO OFDM.</p> <p>Pada skripsi ini, disimulasikan dan dianalisis teknik MIMO OFDM dengan menggunakan teknik <i>Spatial Multiplexing</i> dalam pentransmisi datanya. Pemodelan kanal yang digunakan dalam simulasi ini berbasiskan operasi <i>Singular Value Decomposition</i>. Performansi sistem dalam simulasi ini dinyatakan dengan korespondensi Bit Error Rate (BER), Kapasitas Kanal dan <i>Throughput</i> terhadap Eb/No. Performansi sistem dilihat dengan tiga jenis modulasi, yaitu QPSK, 16 QAM dan 64 QAM. Unjuk kerja keseluruhan sistem divariasikan dengan jenis <i>Error Control Coding (ECC)</i> yang digunakan, dimana pada simulasi ini, terdapat tiga jenis ECC dengan <i>coderate</i> yang berbeda-beda. Selain itu, <i>user</i> dalam simulasi ini terdiri atas dua keadaan, yaitu <i>user</i> diam (<math>v = 0</math> m/s) dan <i>user</i> bergerak (<math>v = 10</math> m/s).</p> <p>Hasil simulasi menunjukkan bahwa unjuk kerja sistem makin baik dengan menggunakan Eb/No yang tinggi. Hal ini dilihat dari grafik BER, Kapasitas kanal, <i>throughput</i> dan juga sinyal konstelasi yang dihasilkan. Selain itu, dari hasil simulasi, diketahui unjuk kerja sistem paling baik didapatkan dengan menggunakan <i>coderate</i> 1/3.</p>	
<p><b>Kata Kunci : MIMO, OFDM, <i>Spatial Multiplexing</i>, SVD</b></p>	

I Gura Prabu Senky  
NPM 04 04 03 707 X  
Electrical Engineering Department

Counsellor  
Dr. Ir. Muhamad Asvial, M.Eng

SPATIAL MULTIPLEXING MIMO OFDM  
WITH SVD CHANNEL MODEL FOR MOBILE WIMAX

ABSTRACT

In recent years, wireless technology development shown rapid increasing. The development lead to fulfilled high capacity needs, using efficiently of bandwidth and ability to protect data from noise and minimizing error. Technology that suited with such conditions is MIMO OFDM.

In this undergraduate thesis, simulation and analysis of spatial multiplexing MIMO OFDM technique is proposed. Channel modeling that used in this simulation based on Singular Value Decomposition. The evaluation of system performance is proposed to Bit Error Rate (BER), Channel Capacity and Throughput parameter toward Eb/No value. Three type of modulation, that are QPSK, 16-QAM and 64-QAM are used in this simulation. For whole system performance, three type Error Control Coding is used in this simulation with different coderate value. Beside that, there are two user condition in this simulation, that are idle user ( $v = 0$  m/s) and mobile user ( $v = 10$  m/s).

The simulation results show that system performance with high Eb/No value is most better. This condition is shown for BER, channel capacity, throughput and signal space graph. In addition, we can see the performance with coderate 1/3 is also better than other coderate values.

**Keywords: MIMO, OFDM, *Spatial Multiplexing*, SVD**

## DAFTAR ISI

SPATIAL MULTIPLEXING MIMO OFDM DENGAN MODEL KANAL SVD UNTUK MOBILE WIMAX	I
SPATIAL MULTIPLEXING MIMO OFDM DENGAN MODEL KANAL SVD UNTUK MOBILE WIMAX	II
PERNYATAAN KEASLIAN SKRIPSI	III
LEMBAR PERSETUJUAN	IV
UCAPAN TERIMA KASIH	V
DAFTAR ISI	VIII
DAFTAR GAMBAR	X
DAFTAR TABEL	XII
DAFTAR SINGKATAN	XIII
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 TUJUAN	2
1.3 BATASAN MASALAH	2
1.4 METODOLOGI PENELITIAN	2
1.5 SISTEMATIKA PENULISAN	3
BAB II TEKNOLOGI MIMO-OFDM	5
2.1. SISTEM MULTIPLE INPUT MULTIPLE OUTPUT (MIMO)	5
2.1.1 Pemodelan Kanal MIMO	6
2.1.2 Teknik MIMO	7
2.1.2.1 <i>Spatial Multiplexing</i> [4]	8
2.1.2.2 <i>Spatial diversity</i>	10
2.2 SISTEM ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING (OFDM)	12
2.2.1 <i>OFDM Processing</i>	15
2.2.2 Keuntungan penggunaan teknik OFDM[8]	17
2.2.2.1 <i>Efisien dalam pemakaian frekuensi</i>	17
2.1.2.2 <i>Kuat menghadapi frequency selective Fading</i>	17
BAB III MODEL SISTEM MIMO OFDM DENGAN SPATIAL MULTIPLEXING	19
3.1 SISI <i>TRANSMITTER</i> (PENGIRIM)	19
3.1.1 <i>Channel Encoder</i>	19
3.1.2 <i>Interleaver</i>	21
3.1.3 Modulasi	21
3.1.3.1 <i>Quadrature Phase Shift Keying (QPSK)</i>	21
3.1.3.2 <i>QAM (Quadrature Amplitude Modulation)</i>	23
3.1.4 IFFT	24
3.1.5 Cyclic Prefix	24
3.1.6 <i>Upsampling</i>	25



3.1.7 <i>Upconversion</i>	25
3.2 KANAL TRANSMISI	26
3.2.1. SVD	26
3.2.2 AWGN dan <i>Fading</i>	28
3.2.3 Least Square	29
3.3 RECEIVER	29
3.3.1 <i>Downconversion</i>	30
3.3.2 <i>DOWNSAMPLING</i>	31
3.3.3. Penghapusan cyclic prefix	31
3.3.4 FFT	31
3.3.5 Demodulasi	31
3.3.5.1 <i>Demodulasi QPSK</i>	31
3.3.5.2 <i>Demodulasi QAM</i>	32
3.3.6. <i>Deinterleaver</i>	32
3.3.7 <i>Convolutional decoder</i>	32
<b>BAB IV HASIL SIMULASI DAN ANALISIS</b>	<b>33</b>
4.1 PARAMETER PERFORMANSI SISTEM	33
4.1.1 <i>Bit Error Rate</i>	34
4.1.2 Kapasitas Kanal	34
4.1.3 <i>Throughput</i>	35
4.2 HASIL SIMULASI DAN ANALISIS	36
4.2.1 Simulasi Berdasarkan ECC	36
4.2.1.1 <i>Simulasi dengan ECC = 0 ( Tanpa menggunakan ECC / Code rate ECC = 1 )</i>	36
4.2.1.2 <i>Simulasi dengan ECC = 2 (Code rate ECC = 1/2)</i>	39
4.2.2 Simulasi Berdasarkan Kecepatan	40
4.2.2.. <i>Simulasi dengan kecepatan 0 m/s</i>	40
4.2.2.2 <i>Simulasi dengan kecepaan 10 m/s</i>	42
4.2.3 Sinyal Konstelasi	43
4.2.3.1 <i>Sinyal Konstelasi Berdasarkan Eb/No</i>	43
4.2.3.2 <i>Sinyal Konstelasi Berdasarkan kecepatan</i>	45
<b>BAB V KESIMPULAN</b>	<b>48</b>
<b>DAFTAR ACUAN</b>	<b>49</b>
<b>DAFTAR PUSTAKA</b>	<b>51</b>
<b>LAMPIRAN</b>	<b>52</b>

## DAFTAR GAMBAR

GAMBAR 2.1 SISTEM MIMO SECARA UMUM	5
GAMBAR 2.2 PEMODELAN KANAL MIMO	7
GAMBAR 2.3 SPATIAL MULTIPLEXING	8
GAMBAR 2.4 SPATIAL MULTIPLEKSING PADA PENGIRIMAN DATA.	9
GAMBAR 2.5 SPATIAL DIVERSITY	11
GAMBAR 2.6 SISI PENGIRIM	11
GAMBAR 2.7 SISI PENERIMA	11
GAMBAR 2.8 KONSEP DASAR DARI OFDM	12
GAMBAR 2.9. PERBANDINGAN SPEKTRUM FDM DAN OFDM	13
GAMBAR 2.10. SPEKTRUM <i>SUBCARRIER</i> SINYAL OFDM	15
GAMBAR 2.11 SKEMA <i>TRANSMITTER</i> SISTEM OFDM	16
GAMBAR 2.12 PENYISIPAN GUARD INTERVAL DI ANTARA 2 SIMBOL	16
GAMBAR 2.13 SKEMA <i>TRECEIVER</i> SISTEM OFDM	17
GAMBAR 3.1 MODEL <i>TRANSMITTER</i> DARI SISTEM YANG DISIMULASIKAN	19
GAMBAR 3.2 TRELIS DIAGRAM	20
GAMBAR 3.3 ALIRAN BIT YANG MELALUI INTERLEAVER	21
GAMBAR 3.4 MODULATOR QPSK	22
GAMBAR 3.5 DIAGRAM FASA MODULASI QPSK	23
GAMBAR 3.6 MODULATOR QAM	23
GAMBAR 3.7 PENAMBAHAN <i>CYCLIC PREFIX</i> PADA FRAME	24
GAMBAR 3.8. <i>PREFILTER V TRANSMITTER</i> MIMO BERBASIS SVD	26
GAMBAR 3.9 MODEL <i>RECEIVER</i> SISTEM YANG DISIMULASIKAN	30
GAMBAR 3.10 <i>POSTFILTER U RECEIVER</i> MIMO BERBASIS SVD	30
GAMBAR 3.11 DEMODULATOR QPSK	31
GAMBAR 3.12 DEMODULATOR QAM	32
GAMBAR 4.1 BER VS $EB/NO$ ECC = 0	37
GAMBAR 4.2 KAPASITAS KANAL VS $EB/NO$ ECC = 0	37
GAMBAR 4.3 <i>THROUGHPUT</i> VS $EB/NO$ ECC = 0	37
GAMBAR 4.4 BER VS $EB/NO$ ECC = 2	39
GAMBAR 4.5 KAPASITAS KANAL VS $EB/NO$ ECC = 2	39
GAMBAR 4.6 <i>THROUGHPUT</i> VS $EB/NO$ ECC = 2	40

GAMBAR 4.7 BER VS EB/NO $V = 0$ M/S	41
GAMBAR 4.8 KAPASITAS KANAL VS EB/NO $V = 0$ M/S	41
GAMBAR 4.9 <i>THROUGHPUT</i> VS EB/NO $V = 0$ M/S	42
GAMBAR 4.10 BER VS EB/NO $V = 10$ M/S	42
GAMBAR 4.11 KAPASITAS KANAL VS EB/NO $V = 10$ M/S	43
GAMBAR 4.12 <i>THROUGHPUT</i> VS EB/NO $V = 10$ M/S	43
GAMBAR 4.13 SINYAL KONSTELASI QPSK BERDASARKAN EB/NO	44
GAMBAR 4.14 SINYAL KONSTELASI 16 QAM BERDASARKAN EB/NO	44
GAMBAR 4.15 SINYAL KONSTELASI 64 QAM BERDASARKAN EB/NO	44
GAMBAR 4.16 SINYAL KONSTELASI QPSK BERDASARKAN KECEPATAN	46
GAMBAR 4.17 SINYAL KONSTELASI 16 QAM BERDASARKAN KECEPATAN	46
GAMBAR 4.18 SINYAL KONSTELASI 64 QAM BERDASARKAN KECEPATAN	46

## DAFTAR TABEL

TABEL 3.1 TABEL KEADAAN SINYAL QPSK	22
TABEL 4.1 PARAMETER SIMULASI MIMO OFDM	33



## DAFTAR SINGKATAN

ADC	Analog Digital Converter
ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CP	Cyclic Prefix
DAC	Digital Analog Converter
DoA	Direction of Arrival
DoD	Direction of Departure
$E_b/N_0$	Energy bit per Noise
ECC	Error Control Coding
FDM	Frequency Division Multiplexing
FEC	Forward Error Correction
FFT	Fast Fourier Transform
GI	Guard Interval
GT	Guard Time
ICI	Inter Channel Interference
IFFT	Inverse Fast Fourier Transform
ISI	Inter Simbol Interference
LS	Least Square
MIMO	Multiple Input Multiple Output
OFDM	Orthogonal Frequency Division Multiplexing
QAM	Quardrature Amplitude Modulation
QPSK	Quardrature Phase Shift Keying
RX	<i>Receiver</i>
SNR	Signal to Noise Ratio
SM	Spatial Multiplexing
STC	Space Time Coding
SVD	Singular Value Decomposition
TX	<i>Transmitter</i>

# BAB I

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Dunia telekomunikasi mengalami perkembangan yang sangat signifikan dalam satu dasawarsa terakhir. Perkembangan tersebut tidak luput dari pengaruh kemajuan teknologi dalam *wireless access*. Dengan mengusung tema *Next Generation Network*, teknologi yang sudah ada sekarang terus ditingkatkan untuk mendapatkan kualitas sistem yang lebih baik lagi. Beberapa indikator yang mendapat sorotan penting dalam pengembangan teknologi tersebut adalah tuntutan akan akses dengan kecepatan tinggi dengan kualitas sinyal yang baik serta tahan terhadap gangguan dan interferensi.

OFDM merupakan suatu skema modulasi *multicarrier* yang mampu bekerja pada *multipath Fading* dan yang paling penting bisa memberikan solusi terhadap tuntutan akses layanan kecepatan tinggi secara real time dengan performansi yang baik[1]. Selain itu, terdapat suatu sistem antena yang bisa dikombinasikan dengan OFDM untuk memenuhi kebutuhan akan kualitas sinyal yang bagus dengan *data rate* yang tinggi tanpa harus menambah daya dan *bandwidth*. Sistem tersebut dinamakan MIMO (*Multiple Input Multiple Output*) [2].

Sistem antena MIMO yang diajukan pada skripsi ini menggunakan teknik *Spatial Multiplexing* untuk transmisi datanya, serta operasi *Singular Value Decomposition* untuk pemodelan kanalnya. Penggunaan SVD untuk pemodelan kanal, merupakan suatu pendekatan yang efisien dan optimal untuk mendapatkan kapasitas kanal yang tinggi sebagaimana dijelaskan dalam [3], ditambah lagi dengan penggunaan teknik *spatial multiplexing* yang mempunyai kelebihan dari segi kapasitas kanal yang besar [4].

Kombinasi sistem MIMO-OFDM, mempunyai kemampuan dalam menawarkan akses komunikasi yang cepat, reliable dan fleksibel serta efisien dalam penggunaan bandwidth[5].

Sistem *spatial multiplexing* MIMO OFDM yang disimulasikan pada skripsi ini menggunakan tiga jenis modulasi, yaitu QPSK, 16 QAM dan 64 QAM serta dilengkapi dengan penggunaan *Error Control Coding* (ECC) yang terdiri dari tiga jenis *coderate* yang berbeda. Selain itu, *user* dalam simulasi ini terdiri atas dua keadaan, yaitu *user* diam ( $v = 0$  m/s) dan *user* bergerak ( $v = 10$  m/s).

## 1.2 TUJUAN

Pada skripsi ini disimulasikan dan dianalisis teknik MIMO OFDM menggunakan *spatial multiplexing* dengan pemodelan kanal berbasiskan *Singular Value Decomposition* untuk mobile WiMAX.

## 1.3 BATASAN MASALAH

Pada skripsi ini, masalah dibatasi pada simulasi sistem MIMO OFDM 2 X 2 pada akses komunikasi bergerak dengan menggunakan teknik MIMO *Spatial Multiplexing*. Pemodelan kanal transmisi MIMO menggunakan operasi *singular value decomposition* (SVD). Selain itu, simulasi ini juga dibatasi hanya terdiri dari *single user*, kondisi *user* diam dan kondisi bergerak dengan kecepatan 10 m/s. Jika dilihat dari sisi WiMAX, simulasi ini hanya dibatasi pada penggunaan frekuensi *carrier* 3.6 GHz.

## 1.4 METODOLOGI PENELITIAN

Pada skripsi ini, penelitian dilakukan dengan studi literatur, kemudian hasil pembelajaran tersebut disimulasikan pada perangkat lunak Matlab (*Matrix Laboratory*).

## 1.5 SISTEMATIKA PENULISAN

Sistematika penulisan SKRIPSI ini adalah sebagai berikut :

**Bab 1 : Pendahuluan**

Bab ini menjelaskan latar belakang masalah, tujuan penulisan, batasan masalah dan sistematika penulisan skripsi ini.

**Bab 2 : Teknik MIMO-OFDM**

Bab ini memberikan penjelasan dasar mengenai konsep MIMO (*Multiple Input Multiple Output*) dan OFDM (*Orthogonal Frequency Division Multiplexing*)

**Bab 3 : Pemodelan Sistem**

Bab ini menjelaskan mengenai model sistem yang akan digunakan pada simulasi beserta parameter yang dianalisis

**Bab 4 : Hasil Simulasi dan Analisis**

Bab ini memaparkan hasil simulasi dan analisis yang rinci mengenai hasil tersebut.

**Bab 5 : Kesimpulan**



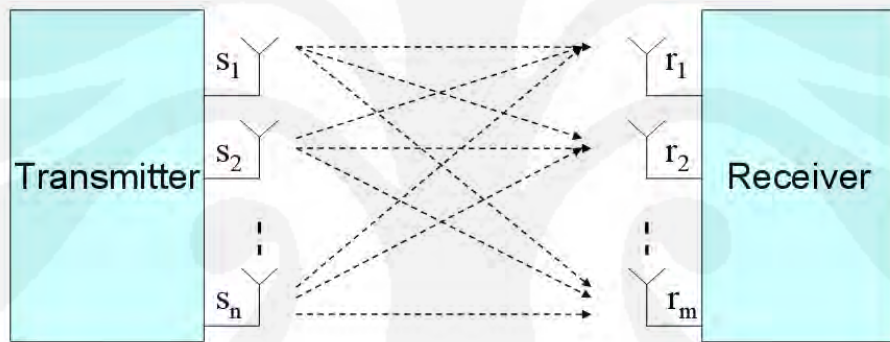


## BAB II

### TEKNOLOGI MIMO-OFDM

#### 2.1. Sistem Multiple Input Multiple Output (MIMO)

Sistem *Multiple-Input Multiple-Output* (MIMO) merupakan sistem yang terdiri dari sejumlah terminal (antena) pengirim dan penerima. Tidak seperti sistem antenna konvensional yang sangat rentan terhadap multipath, sistem MIMO justru bekerja sangat baik pada komponen multipath. Komponen multipath dieksploitasi untuk meningkatkan diversitas dan efisiensi bandwidth (bps/Hz) yang tekniknya akan dibahas kemudian.



Gambar 2.1 Sistem MIMO secara umum

Pada gambar 2.1 diatas, terlihat sistem MIMO dengan antenna pengirim dan penerima yang lebih dari satu. Antena penerima akan menerima sinyal yang dikirimkan oleh antenna pengirim setelah sinyal tersebut dikalikan dengan suatu matriks kanal. Secara umum, dengan matriks kanal H, sinyal yang diterima oleh antenna penerima dapat dirumuskan sebagai berikut:

$$x_1 = h_{11}s_1 + h_{12}s_2 + \dots + h_{1N}s_N$$

$$x_2 = h_{21}s_1 + h_{22}s_2 + \dots + h_{2N}s_N$$

$$x_N = h_{N1}s_1 + h_{N2}s_2 + \dots + h_{NN}s_N$$

atau, secara umum dapat digabungkan ke dalam suatu persamaan, yaitu:

$$x_i(t) = \sum_{k=1}^K h_{ik}(t)s_k(t) \dots\dots\dots(1.1)$$

menghasilkan model sinyal sederhana sistem MIMO :

$$\mathbf{x}(t) = \mathbf{H}(t)\mathbf{s}(t) \dots\dots\dots(1.2)$$

dengan untuk semua  $N_t$  sinyal, digunakan notasi matriks:

$$\mathbf{s}(t) = \begin{pmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_L(t) \end{pmatrix}, \mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_L(t) \end{pmatrix}, \mathbf{H}(t) = \begin{pmatrix} h_{11}(t) & h_{12}(t) & \dots & h_{1K}(t) \\ h_{21}(t) & h_{22}(t) & \dots & h_{2K}(t) \\ \vdots & \vdots & \ddots & \vdots \\ h_{L1}(t) & h_{L2}(t) & \dots & h_{LK}(t) \end{pmatrix} \dots\dots(1.3)$$

matriks H merupakan matriks kanal MIMO yang dibentuk dari estimasi nilai  $h_{ij}$  pada kanal transmisi. Matriks ini akan berguna dalam mendapatkan kembali sinyal informasi pada sisi penerima. Sinyal informasi didapatkan dengan mengalikan *inverse* matriks H dengan sinyal pada sisi penerima (x), seperti terlihat pada persamaan berikut:

$$\mathbf{H}^{-1}(t)\mathbf{x}(t) = \mathbf{H}^{-1}(t)\mathbf{H}(t)\mathbf{s}(t) = \mathbf{I}_{N_t}\mathbf{s}(t) = \mathbf{s}(t) \dots\dots\dots(1.4)$$

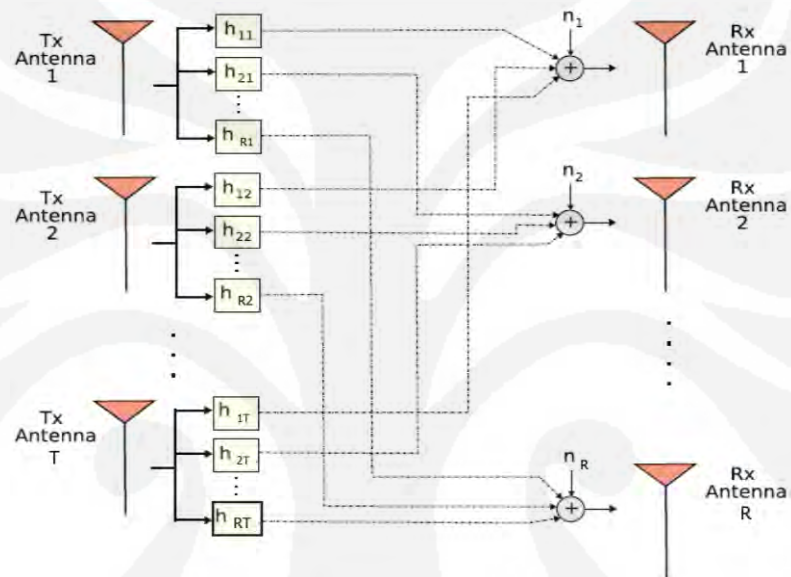
### 2.1.1. Pemodelan Kanal MIMO

Pengaruh karakter statistik *multidimensional* dari kanal *Fading* MIMO (matriks H) memiliki peran yang sangat signifikan pada performa sistem. Oleh karena itu, perancangan model kanal MIMO, menjadi suatu hal yang penting. Pemodelan kanal MIMO berdasarkan keadaan lingkungan fisiknya, dibagi menjadi dua, yaitu model *physical (Deterministic/geometric)* dan model *stochastic* [6].

Model *Deterministic*, yaitu model fisik melibatkan parameter fisik kanal di keadaan sebenarnya dari semua komponen *multipath*, seperti DoA (*Directions of Arrival*) dan DoD (*Directions of Departure*). Pemodelan ini berlaku untuk lingkungan *picocell / microcell*. Model ini dapat digunakan sebagai dasar untuk pemodelan secara *stochastic*. Contoh pemodelan ini adalah *Codit Model*. Model kedua, *stochastic* yang berarti berkarakteristik probabilitas atau acak, memberikan

permodelan secara statistik dari properti-properti spasial kanal pada elemen-elemen antena [6]. Model ini biasanya dipakai pada daerah pengukuran yang luas. Contoh dari model ini adalah model Metra MIMO pada 3GPP.

Variasi jalur tempuh yang berbeda-beda antara TX dengan RX sebagai fungsi waktu, lokasi, dan frekuensi (multipath *Fading*) dapat direpresentasikan dengan distribusi statistik. Untuk fokus ke masalah *Fading* ini, model geometris dapat ditransfer ke dalam model stochastic [6].



Gambar 2.2 Pemodelan kanal MIMO

Pada gambar 2.2 terlihat pemodelan kanal MIMO secara umum. Dari gambar, terlihat sistem MIMO dengan sejumlah  $T$  antena *transmitter* dan  $R$  antena *receiver*. Kanalnya direpresentasikan dengan matriks  $H$  yang memiliki  $R$  baris dan  $T$  kolom. Elemen-elemen matriksnya,  $h_{ij}$  merupakan fungsi transfer dari antena *transmitter* ke  $j$  ke antena *receiver* ke  $i$ . Dari gambar terlihat, sinyal yang diterima antena *receiver* merupakan penjumlahan sinyal dari semua antena *transmitter*.

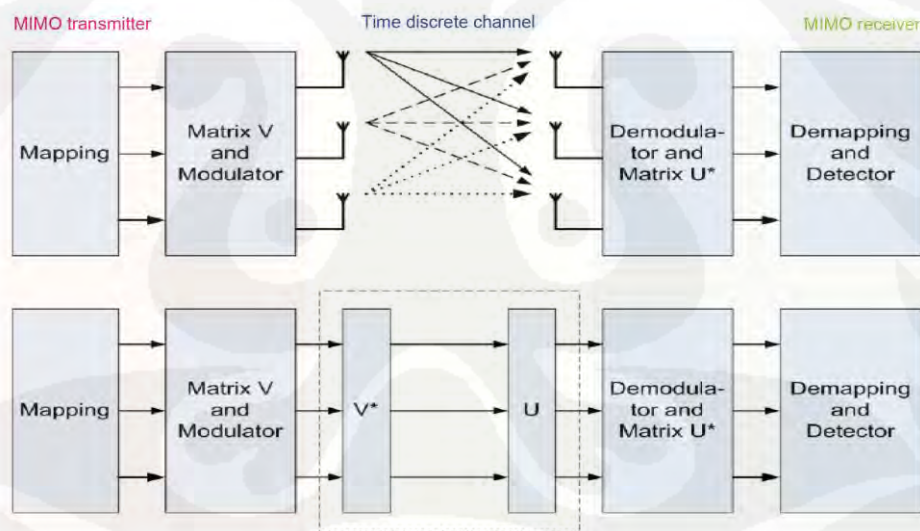
### 2.1.2 Teknik MIMO

Sistem MIMO dapat memanfaatkan keberadaan *multipath* untuk menciptakan sejumlah kanal ekuivalen yang seolah-olah terpisah satu sama lain, dimana pada kondisi normal keberadaan *multipath* justru merugikan karena

menimbulkan *fading*. Dalam aplikasinya, terdapat dua macam teknik MIMO yang digunakan dalam sistem komunikasi nirkabel dan bergerak yaitu:

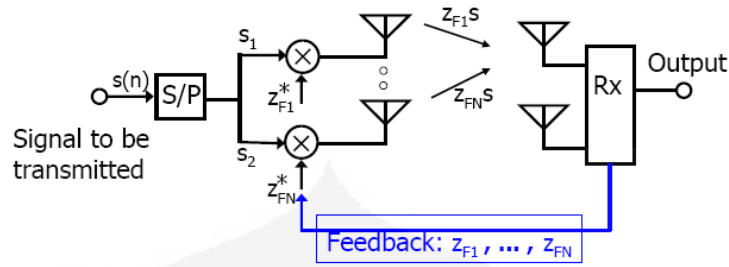
### 2.1.2.1. Spatial Multiplexing[4]

Teknik *spatial multiplexing* mengirimkan data yang berbeda secara paralel dan dikodekan secara paralel pula untuk setiap antenna transmisi. Tujuan utama penggunaan teknik ini adalah untuk mencapai kapasitas kanal yang besar, dengan memecah aliran data berlaju tinggi menjadi sejumlah aliran paralel sesuai dengan jumlah antenna *transmitter*, masing-masing dengan laju yang lebih rendah dari aliran aslinya. Aliran-aliran data ini dilewatkan pada matriks khusus yang berfungsi menggabung-gabungkan sinyal dari semua aliran dengan kombinasi tertentu untuk ditransmisikan melalui setiap antenna. Ini merupakan suatu proses multipleks yang berlangsung pada dimensi spasial karena setiap kombinasi data paralel ditujukan ke salah satu antenna *transmitter*.



Gambar 2.3 Spatial multiplexing

*Spatial multiplexing* dapat menambah spectrum efisiensi sehingga dapat menambah kecepatan transmisi data.



Gambar 2.4 Spatial multipleksing pada pengiriman data.

Untuk mengestimasi respon kanal pada sistem ini, diterapkan suatu saluran umpan balik informasi dari antenna *receiver* ke *transmitter*. Dengan adanya umpan balik ini, *transmitter* dapat mengetahui nilai matriks multipleks yang optimum untuk mendapatkan kapasitas kanal yang maksimal.

Operasi dekomposisi nilai singular (*singular value decomposition* atau SVD) merupakan suatu teknik yang digunakan untuk mendiagonalisasi suatu matriks dan menentukan nilai eigennya, dimana dalam hal ini, bertujuan untuk mengestimasi matriks respons kanal. Matriks respon kanal yang dihasilkan adalah

$$H = U_H \Sigma_H V_H^* \dots\dots\dots(1.5)$$

Dua matriks unitaris U dan V, yang dihasilkan oleh operasi ini adalah matriks multipleks dan matriks demultipleks yang digunakan oleh *transmitter* dan *receiver*. Konfigurasi sistem tersebut kemudian menjadi ekuivalen dengan sistem *transmitter-receiver* yang terhubung melalui sejumlah saluran paralel sebanyak T dan R, tergantung mana yang lebih kecil.

Bila T bernilai lebih kecil dari R, maka sistem ini seolah-olah memiliki T saluran yang terpisah satu sama lain untuk membawa T aliran data yang berbeda, masing-masing dengan laju rata-rata 1/T dari laju aliran data aslinya, padahal seluruh sistem multi antenna ini bekerja pada frekuensi yang sama. Dapat disimpulkan telah terjadi penghematan penggunaan *bandwidth* sebesar 1/T kali, atau dengan kata lain terjadi peningkatan kapasitas kanal sebesar T kali. Bayangkan kondisi ekstremnya, dengan sedikitnya 10 antenna pada masing-masing sisi *transmitter* dan *receiver*, aliran data sebesar 1 Mbps dapat dikirimkan ke *receiver* dengan *bandwidth* sekitar 100 kHz saja apabila digunakan modulasi dengan efisiensi 1 bps/Hz. Atau dari sudut pandang yang berlawanan, lebar spektrum 100 kHz yang sebelumnya hanya mampu membawa sinyal 100 kbps,

sekarang mampu mengangkut data berlaju 1 Mbps dengan menggunakan minimal 10 antena pada setiap sisi

Kapasitas kanal sistem MIMO dapat dihitung berdasarkan rumusan Shanon, yaitu:

$$C_{1x1} = E \left[ \log_2 \left( 1 + \frac{P|h_{11}|^2}{\sigma_n^2} \right) \right] \text{ bps/Hz} \dots\dots\dots(1.6)$$

Dengan menerapkan operasi SVD, kapasitas kanal Shannon dapat diubah menjadi:

$$\begin{aligned} C_{R \times T} &= E \left[ \log_2 \left( \det \left( I_{R_o} + \frac{P}{T\sigma_n^2} HH^h \right) \right) \right] \\ &= E \left[ \sum_{i=1}^{\text{rank}(H)} \log_2 \left( 1 + \frac{P}{T\sigma_n^2} \delta_i^2 \right) \right] \text{ bps/Hz} \dots\dots\dots(1.7) \end{aligned}$$

Dimana  $C$  adalah kapasitas kanal yang dihitung dalam [bps/Hz],  $E$  adalah ekspektasi pada semua realisasi kanal,  $P$  adalah daya rata-rata yang ditransmisikan pada kanal,  $H$  adalah penguatan kanal kompleks random,  $\sigma_n^2$  adalah *varians noise*,  $I_{R_o}$  adalah matriks identitas berdimensi  $\text{rank}(H)$  yang dilihat dari sisi *receiver*,  $T$  adalah jumlah kanal spasial,  $H^h$  adalah hermitian matriks  $H$ ,  $\text{rank}(H) \leq \min\{R,T\}$ ,  $\delta_i^2$  adalah kuadrat nilai singular yang berkaitan dengan daya dari tiap-tiap kanal hasil dekomposisi matriks  $HH^h$  dan  $P/\sigma_n^2$  adalah rasio *signal-to-noise* (SNR) rata-rata pada setiap sisi *receiver*. [4]

Selain itu, untuk mendapatkan kapasitas kanal yang lebih optimal, dapat digunakan teknik *waterfilling* dengan laju sub aliran (*substream*) transmisi yang lebih cepat melalui kanal dengan SNR yang lebih tinggi.

### 2.1.2.2 Spatial diversity

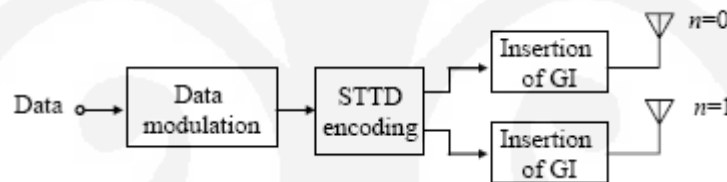
Pada teknik ini, setiap antena pengirim pada sistem MIMO mengirimkan data yang sama secara paralel dengan menggunakan *coding* yang berbeda. Tujuannya adalah mendapatkan kualitas sinyal setinggi mungkin dengan memanfaatkan teknik *diversity* pada *transmitter* dan *receiver*



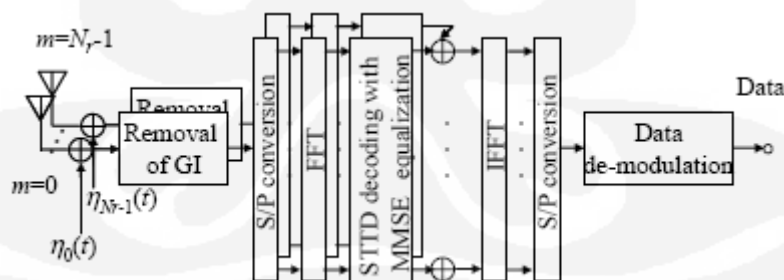
Gambar 2.5 Spatial diversity

*Diversity* secara konvensional diaplikasikan dengan pemasangan antena *array* pada sisi *receiver*, dengan harapan bahwa kualitas sinyal yang diterima dapat ditingkatkan dari sistem satu antena dalam kondisi kanal *fading* dengan adanya *multipath*. *Receive* and *transmit diversity* dapat menanggulangi *fading* dan secara signifikan dapat menambah *link quality* atau dengan kata lain dapat meningkatkan SNR.

Berikut merupakan blok diagram pada sisi pengirim dan penerima dari *Spatial diversity*:



Gambar 2.6 Sisi pengirim



Gambar 2.7 Sisi penerima

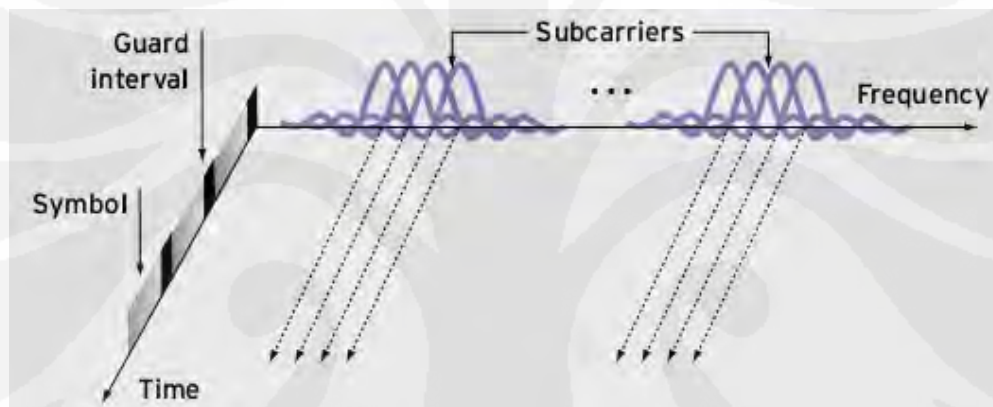
Peningkatan kualitas sinyal dapat dilihat berdasarkan nilai parameter penguatan *diversity* (*diversity gain*), yang harganya makin meningkat dengan makin besarnya tingkat *diversity*  $R$ , yaitu jumlah antena yang digunakan pada *receiver*. Penggunaan STC (*Space Time Coding*) pada sistem MIMO dengan sejumlah  $T$  antena *transmitter* dan  $R$  antena *receiver* menjanjikan kenaikan tingkat *diversity* menjadi  $T \times R$ . Sebagai gambaran, dengan 4 antena pada masing-



masing *transmitter* dan *receiver*, sistem MIMO dengan STC diharapkan mampu menyediakan tingkat *diversity* yang setara dengan metode konvensional menggunakan 16 antena pada *receiver*[7]

## 2.2. Sistem Orthogonal Frequency Division Multiplexing (OFDM)

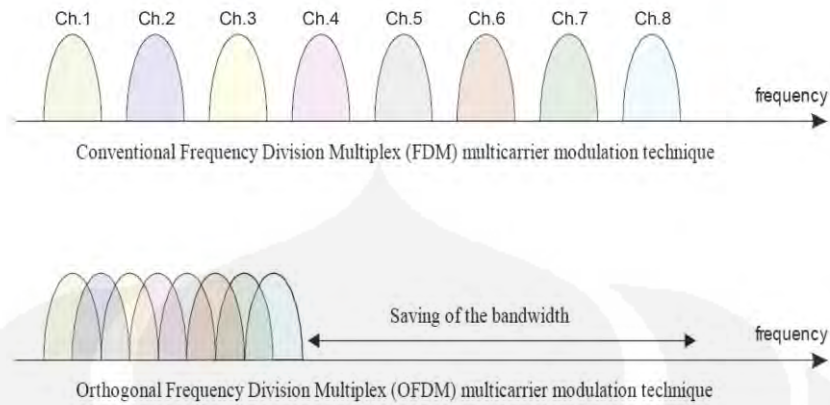
OFDM pada dasarnya merupakan suatu skema transmisi *multi carrier*, dimana menggabungkan suatu teknik modulasi dengan *multiplexing*. Konsep dasar dari OFDM adalah pembagian/penyebaran transmisi data melalui sejumlah *subcarriers*, dan dimodulasi pada *low rate*. Dengan demikian, suatu sinyal OFDM dapat dikatakan terdiri dari sekumpulan *narrowband carrier* yang ditransmisikan dari sumber yang sama pada frekuensi yang berbeda.



Gambar 2.8 Konsep dasar dari OFDM

Pada gambar 2.8 terlihat susunan sinyal OFDM, dimana secara umum terdiri dari sekumpulan *subcarriers* yang saling tegak lurus satu sama lain. Jarak diantara *subcarriers* merupakan kebalikan dari durasi setiap simbol, sehingga masing-masing *subcarriers* saling tegak lurus (*non-interfering*).[5]

Masing masing *subcarrier* saling ortogonal satu sama lain. Ortogonalitas inilah yang memungkinkan adanya *overlapping spectral* diantara *subcarrier*, dimana spektrum dari tiap-tiap *subcarriernya* mempunyai harga nol pada frekuensi pusat *subcarrier* lainnya sehingga tidak akan terjadi interferensi antar sub *carrier*. Hal inilah yang membedakan prinsip kerja OFDM dengan FDM, dimana pada FDM pembagian kanal hanya berdasarkan frekuensi, dengan diberi suatu jarak tertentu antar kanal (*Guard Interval*) untuk mencegah interferensi. Dari segi inilah, terlihat dengan jelas, bahwa OFDM lebih efisien dalam penggunaan bandwidth dibanding konsep FDM.



Gambar 2.9. Perbandingan spektrum FDM dan OFDM

Konsep ortogonalitas, dapat dijelaskan secara matematis sebagai berikut [4]: Misalkan simbol serial termodulasi QPSK dengan periode simbol  $T$  dibagi ke dalam sejumlah  $N$  aliran pengganti paralel. Setelah konversi paralel  $N$  maka periode simbol pada setiap aliran pengganti meningkat menjadi  $T_s = T \times N$ . Sinyal *baseband* ekuivalen yang disusun dari sejumlah  $N$  aliran paralel ditulis

$$s(t) = \sum_{n=-\infty}^{\infty} \sum_{k=0}^{N-1} X_{n,k} e^{j2\pi f_k (t-T_s)} \dots\dots\dots(1.8)$$

$X_{n,k}$  adalah simbol modulasi QPSK yang ditransmisikan pada *subcarrier* ke  $k$  dari simbol OFDM ke  $n$  dan  $f_k$  adalah frekuensi pusat *subcarrier* ke  $k$ . Jika spasi frekuensi di antara dua *subcarrier* adalah merupakan kelipatan  $1/T_s$  atau  $f_k = \frac{k}{T_s}$ , dimana  $k$  adalah bilangan bulat, maka *down-converting* sinyal yang diterima adalah

$$\begin{aligned} \hat{X}_k &= \frac{1}{2T_s} \int_{-T_s}^{T_s} \left[ \sum_{k=0}^{N-1} X_k c_k(t) \right] c_k^*(t) dt \\ &= \frac{1}{2T_s} \int_{-T_s}^{T_s} \left[ \sum_{k=0}^{N-1} X_k e^{j2\pi f_{k_1} t} \right] e^{-j2\pi f_{k_2} t} dt \\ &= \frac{1}{2T_s} \int_{-T_s}^{T_s} \left[ \sum_{k=0}^{N-1} X_k e^{j2\pi \frac{k_1}{T_s} t} \right] e^{-j2\pi \frac{k_2}{T_s} t} dt \\ &= \frac{1}{2T_s} \sum_{k=0}^{N-1} X_k \int_{-T_s}^{T_s} e^{j2\pi \frac{t}{T_s} (k_1 - k_2)} dt \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2T_s} \sum_{k=0}^{N-1} X_k \int_{-T_s}^{T_s} e^{j2\pi \frac{t}{T_s} \Delta k} dt \\
&= \frac{1}{2T_s} \sum_{k=0}^{N-1} X_k \left[ \int_{-T_s}^{T_s} \cos(2\pi \Delta k \frac{t}{T_s}) dt + j \int_{-T_s}^{T_s} \sin(2\pi \Delta k \frac{t}{T_s}) dt \right] \\
&= \frac{1}{2T_s} \sum_{k=0}^{N-1} X_k \left[ \frac{\sin(2\pi \Delta k)}{\pi \Delta k} + 0 \right] \\
&= \sum_{k=0}^{N-1} X_k \left[ \frac{\sin(2\pi \Delta k)}{2\pi \Delta k} \right] \\
&= \sum_{k=0}^{N-1} X_k \text{sinc}(2\pi \Delta k) \\
&= \sum_{k=0}^{N-1} X_k \text{sinc}(2\pi(k_1 - k_2)) \\
\hat{X}_k &= \begin{cases} X_k & , \text{ untuk } k_1 = k_2 \\ 0 & , \text{ untuk } k_1 \neq k_2 \end{cases}
\end{aligned}$$

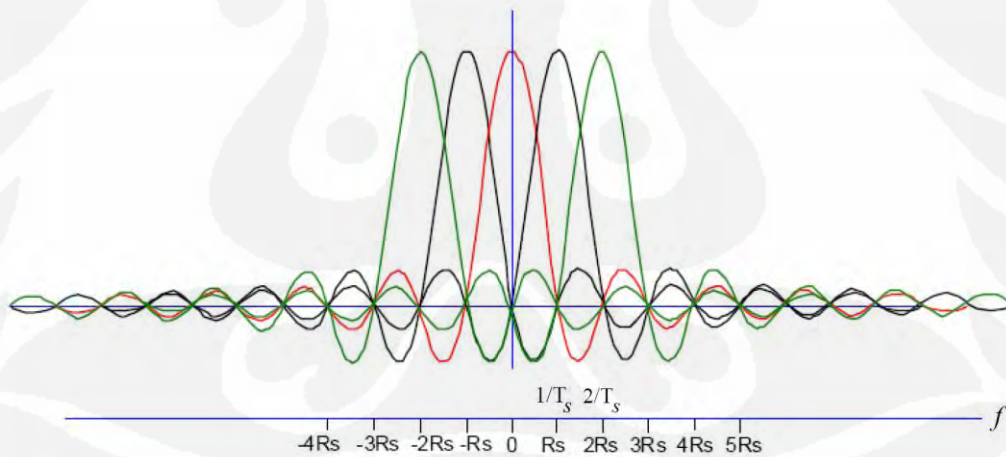
Hasil tersebut di atas dicapai oleh karena integrasi :

$$\begin{aligned}
\int_{-T_s}^{T_s} \cos\left[\frac{2\pi \Delta k}{T_s} t\right] dt &= \left[\frac{T_s}{2\pi \Delta k} t\right] \int_{-T_s}^{T_s} \cos\left[\frac{2\pi \Delta k}{T_s} t\right] d\left[\frac{2\pi \Delta k}{T_s} t\right] \\
&= \frac{T_s}{2\pi \Delta k} \sin\left(\frac{2\pi \Delta k}{T_s} t\right) \Bigg|_{-T_s}^{T_s} \\
&= \frac{T_s}{2\pi \Delta k} [\sin(2\pi \Delta k) - \sin(-2\pi \Delta k)] \\
&= \frac{T_s}{2\pi \Delta k} [(\sin(2\pi \Delta k)) - (-\sin(2\pi \Delta k))] \\
&= T_s \frac{\sin 2\pi \Delta k}{\pi \Delta k}
\end{aligned}$$

dan integrasi :

$$\begin{aligned}
j \int_{-T_s}^{T_s} \sin \left[ \frac{2\pi \Delta k}{T_s} t \right] dt &= j \left[ \frac{T_s}{2\pi \Delta k} \right] \int_{-T_s}^{T_s} \sin \left[ \frac{2\pi \Delta k}{T_s} t \right] d \left[ \frac{2\pi \Delta k}{T_s} t \right] \\
&= j \frac{T_s}{2\pi \Delta k} \int_{-T_s}^{T_s} \sin \left[ \frac{2\pi \Delta k}{T_s} t \right] d \left[ \frac{2\pi \Delta k}{T_s} t \right] \\
&= -j \frac{T_s}{2\pi \Delta k} \cos \left( \frac{2\pi \Delta k}{T_s} t \right) \Bigg|_{-T_s}^{T_s} \\
&= -j \frac{T_s}{2\pi \Delta k} [\cos(2\pi \Delta k) - \cos(-2\pi \Delta k)] \\
&= -j \frac{T_s}{2\pi \Delta k} [\cos(2\pi \Delta k) - \cos(2\pi \Delta k)] \\
&= 0
\end{aligned}$$

Dari uraian matematis diatas, terlihat bahwa masing-masing sub-carrier saling tegak lurus. Spektrum yang dihasilkan oleh teknik OFDM ini dapat dilihat pada gambar berikut:



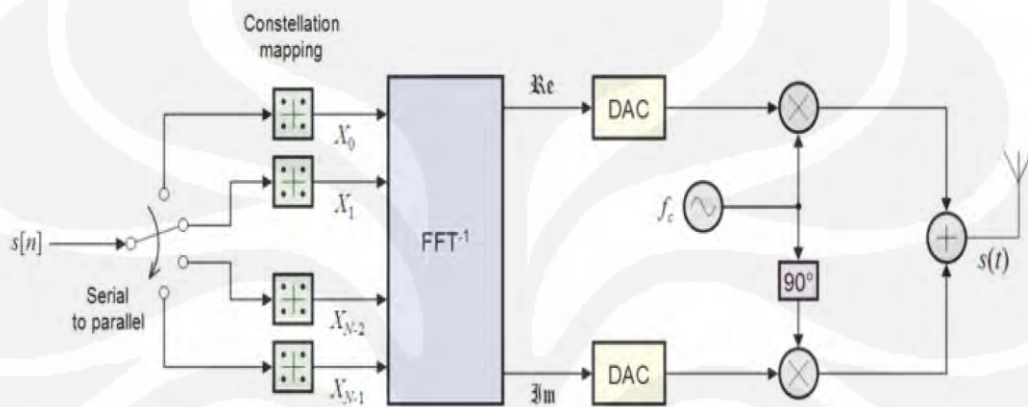
Gambar 2.10. Spektrum subcarrier sinyal OFDM

### 2.2.1. OFDM Processing

Secara umum, proses transmisi dari sistem OFDM dapat dijelaskan sebagai berikut:

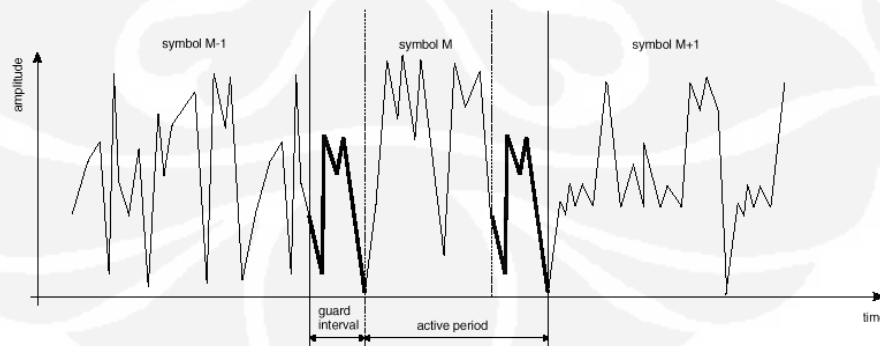
Masukan data serial pada sisi *transmitter*, dikodekan untuk menambahkan *redundancy check* yang bertujuan untuk *error recovery* pada saat ditransmisikan,

selain itu juga *interleaving* data untuk mengantisipasi kanal yang *error*. Kemudian, data dikonversi dari serial ke paralel dan dipetakan dengan menggunakan signal *mapper* dari  $x$  bits untuk membentuk bilangan kompleks. Pemetaan ini terjadi pada setiap sub *carrier* dengan menggunakan modulasi biasa, seperti QPSK dan QAM. Dengan menggunakan *Inverse Fast Fourier Transform* (IFFT), hasil modulasi tadi diubah menjadi suatu sinyal multi *subcarrier* OFDM. Proses yang terjadi pada sisi *transmitter* dapat dilihat pada gambar 2.11 berikut.



Gambar 2.11 Skema *transmitter* sistem OFDM

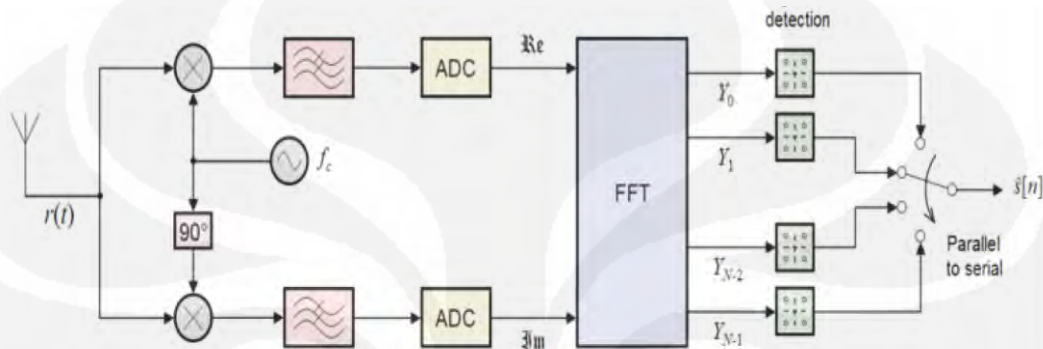
Sinyal ini dikonversi kembali ke serial untuk transmisi. Sebelum ditransmisikan, sebuah *guard interval* disisipkan diantara simbol untuk mencegah *Intersymbol Interference (ISI)* yang disebabkan distorsi multipath. *Cyclic prefix* bertindak sebagai *Guard Interval* di mana panjangnya melebihi delay pada kanal propagasi *multipath*. Penyisipan *Guard Interval* dapat dilihat pada gambar 2.11 berikut.



Gambar 2.12 Penyisipan guard interval di antara 2 simbol

Pada gambar 2.11, terlihat posisi *Guard Interval* pada setiap simbol. *Guard Interval* disisipkan pada setiap simbol dengan tujuan mencegah terjadinya ISI.

Selanjutnya, simbol diskret tersebut dikonversi ke analog oleh DAC (*Digital Analog Converter*) dan kemudian di filter menggunakan *Low Pass Filter* untuk *up-conversion* dan ditransmisikan melalui antenna pemancar (*transmitter*). Pada sisi penerima (*receiver*), proses yang berkebalikan terjadi, untuk mendapatkan sinyal data kembali. Proses pengolahan sinyal pada sisi *receiver* dapat dilihat pada gambar 2.13 berikut.



Gambar 2.13 Skema *receiver* sistem OFDM

## 2.2.2 Keuntungan penggunaan teknik OFDM[8]

Berikut merupakan keuntungan teknik OFDM secara umum:

### 2.2.2.1 . Efisien dalam pemakaian frekuensi

OFDM adalah salah satu jenis *multicarrier* (FDM), tetapi memiliki efisiensi pemakaian frekuensi yang jauh lebih baik. Pada OFDM *overlap* antar frekuensi yang bersebelahan diperbolehkan, karena masing-masing sudah saling ortogonal, sedangkan pada sistem *multicarrier* konvensional untuk mencegah interferensi antar frekuensi yang bersebelahan perlu diselipkan frekuensi penghalang (*guard band*), dimana hal ini memiliki efek samping berupa menurunnya kecepatan transmisi bila dibandingkan dengan sistem *single carrier* dengan lebar spektrum yang sama. Sehingga salah satu karakteristik dari OFDM adalah tingginya tingkat efisiensi dalam pemakaian frekuensi. Selain itu pada *multicarrier* konvensional juga diperlukan *band pass filter* sebanyak frekuensi yang digunakan, sedangkan pada OFDM cukup menggunakan FFT saja.

### 2.1.2.2. Kuat menghadapi *frequency selective Fading*

Karakter utama yang lain dari OFDM adalah kuat menghadapi *frequency selective Fading*. Dengan menggunakan teknologi OFDM, meskipun jalur komunikasi yang digunakan memiliki karakteristik *frequency selective Fading*

(dimana *bandwidth* dari kanal lebih sempit dibanding *bandwidth* dari transmisi sehingga mengakibatkan pelemahan daya terima secara tidak seragam pada beberapa frekuensi tertentu), tetapi tiap sub *carrier* dari sistem OFDM hanya mengalami *flat Fading* (pelemahan daya terima secara seragam). Pelemahan yang disebabkan oleh *flat Fading* ini lebih mudah dikendalikan, sehingga performansi dari sistem mudah untuk ditingkatkan.

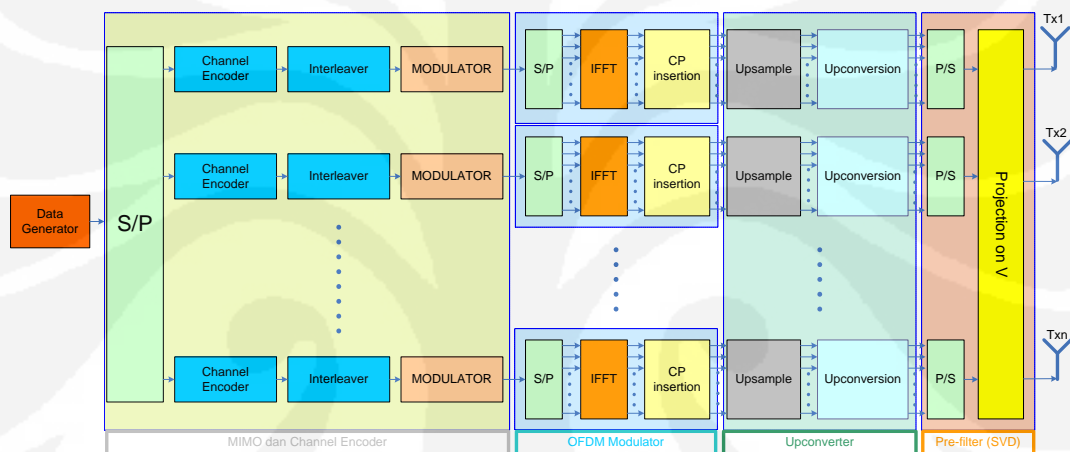
Selain beberapa keuntungan diatas, terdapat beberapa kekurangan dari sistem OFDM, yaitu dari segi *frequency offset* dan distorsi linier. Sistem OFDM sangat sensitif terhadap *carrier frequency offset* yang disebabkan oleh *jitter* pada gelombang pembawa (*carrier wave*) dan juga terhadap Efek Doppler yang disebabkan oleh pergerakan baik oleh stasiun pengirim maupun stasiun penerima. Selain itu, karena teknologi OFDM merupakan sebuah sistem modulasi yang menggunakan multi-frekuensi dan multi-amplitudo, sehingga sistem ini mudah terkontaminasi oleh distorsi nonlinear yang terjadi pada *amplifier* dari daya transmisi.

# BAB III

## MODEL SISTEM MIMO OFDM DENGAN SPATIAL MULTIPLEXING

### 3.1 Sisi *Transmitter* (Pengirim)

Skema *transmitter* dari sistem yang disimulasikan dapat dilihat pada gambar 3.1. Gambar 3.1 mengilustrasikan tahap-tahap pemrosesan sinyal yang terjadi pada sisi *transmitter* mulai dari sumber data informasi, sampai pengiriman oleh antenna *transmitter*.



Gambar 3.1 Model *transmitter* dari sistem yang disimulasikan

#### 3.1.1 *Channel Encoder*

Data inputan yang berupa *random integer* dibagi menjadi kumpulan data yang lebih kecil setelah melalui algoritma *Serial to Paralel*. Kemudian data ini, diencoding dengan *code rate* tertentu. Proses *encoding* ini disebut juga *forward error correction (FEC)*, dimana bertujuan untuk meningkatkan kapasitas dari kanal dengan menambahkan bit *redundant* pada data informasi yang ditransmisikan. Proses penambahan bit *redundant* tersebut juga dinamakan *channel coding*. Teknik *channel coding* yang dipakai pada simulasi ini adalah *Convolutional encoding*.

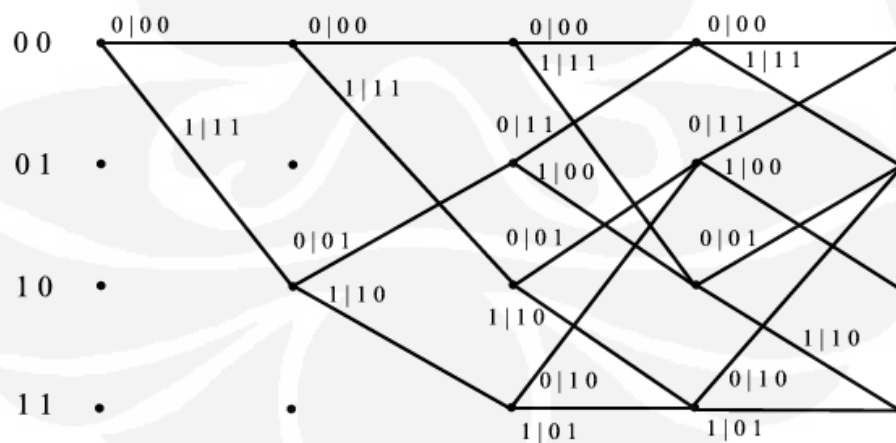
*Convolutional encoder* berfungsi untuk menurunkan *error rate* selama pentransmisian data, atau dengan kata lain berfungsi sebagai pendeteksi dan sekaligus memperbaiki error selama pentransmisian data, tanpa harus meminta



pengirim untuk mengirimkan kembali data yang sama. Prinsip kerja dari *convolutional encoder* adalah dengan men-*xor*-kan beberapa *bit* masukan sebelumnya dengan *bit input* pada saat itu. Sistem *xor* ini mencakup bagaimana alur operasi *xor* dan berapa lama suatu *bit* akan mempengaruhi keluaran *bit* lain.

Pada sistem MIMO OFDM yang disimulasikan, terdapat 3 pilihan nilai *code rate*, yaitu: 3/2, 1/2 dan 1/3. *Coding rate* pada *convolutional encoder* merupakan nilai perbandingan antara jumlah keluaran dengan jumlah masukan pada satu waktu yang bersamaan. Besar nilai *code rate* akan menentukan pemetaan output dari *convolutional encoding*. Sebagai contoh. *Code rate* 3/2 berarti, 3 bit masukan dipetakan menjadi 2 bit keluaran. *Code rate* yang bernilai kecil, berarti jumlah bit kode lebih banyak dibanding jumlah bit informasi, begitu juga sebaliknya. Pemilihan nilai *code rate* sangat penting dalam menjaga performa sistem. Disatu sisi, Jika nilai *coding rate* semakin besar, performanya dalam mendeteksi dan memperbaiki *error* juga semakin baik tetapi sebagai konsekuensinya *encoder* semakin tidak efisien. Karena jumlah *bit* yang akan dikirimkan menjadi lebih banyak untuk jumlah *bit* informasi yang sama.

Pada simulasi ini, teknik *Convolutional encoder* menggunakan algoritma *Trellis* (*Trellis diagram*). Diagram dari algoritma *Trellis* dapat dilihat pada gambar 3.2.

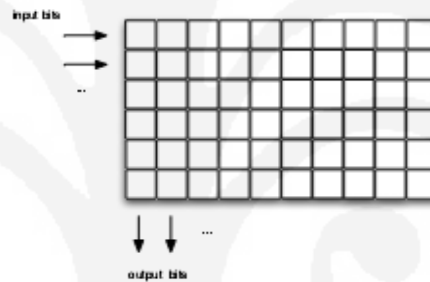


Gambar 3.2 Trellis Diagram

Gambar 3.2 diatas, merupakan contoh diagram Trellis dengan 2 *shift register* . Terdapat 4 keadaan dari diagram Trellis tersebut, yaitu 00, 01, 10, 11. Sebagai contoh, jika bit masukan adalah 1 1 0 0 , maka keluaran dari *convolutional encoder* adalah 11, 10, 10, 11.

### 3.1.2 Interleaver

Proses *Interleaver* berfungsi sebagai cara untuk meminimalisasi *burst error* selama transmisi data. *Interleaver* akan menyusun kembali data dimana data yang berurutan akan dibagi menjadi beberapa blok berbeda. *Interleaver* disini berbentuk sebuah matriks. Aliran bit inputan akan mengisi matriks tersebut baris demi baris. Keluaran dari *interleaver* merupakan bit bit yang tersusun secara kolom.



Gambar 3.3 Aliran bit yang melalui interleaver

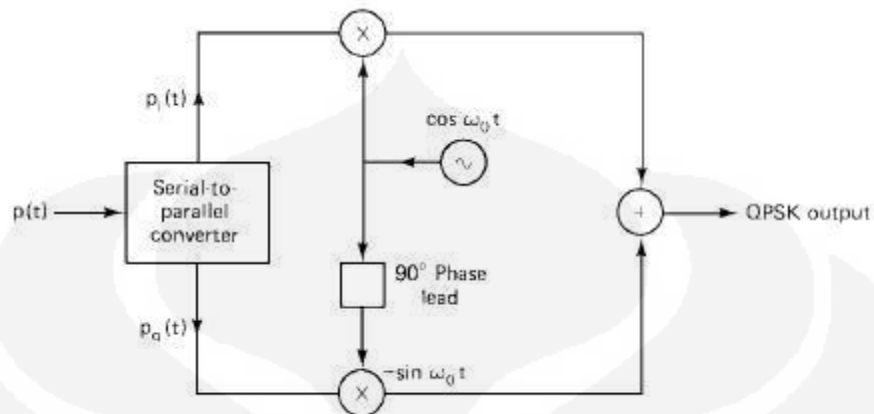
### 3.1.3 Modulasi

Data hasil keluaran dari *Interleaver* dipetakan menjadi simbol-simbol melalui proses modulasi. Dalam sistem ini, jenis modulasi yang digunakan ada tiga jenis, yaitu QPSK, 16 QAM dan 64 QAM. Kemudian dari hasil stream serial data, dibagi menjadi paralel sesuai dengan jumlah subkanal pada OFDM.

#### 3.1.3.1 Quadrature Phase Shift Keying (QPSK)

QPSK memetakan bit bit informasi menjadi simbol, dimana setiap simbolnya memiliki 2 bit informasi. QPSK dapat diperoleh dengan penggabungan 2 modulasi BPSK. Bit stream yang masuk pada modulator  $p(t)$ , dibagi menjadi 2 stream yaitu  $p_t(t)$  dan  $p_q(t)$ . Bit stream  $p_t(t)$  akan dimodulasikan dengan  $\cos(\omega_0 t)$

dan  $p_q(t)$  akan dikalikan dengan  $\sin(\omega_0 t)$ , seperti yang terlihat pada gambar 3.4 berikut:



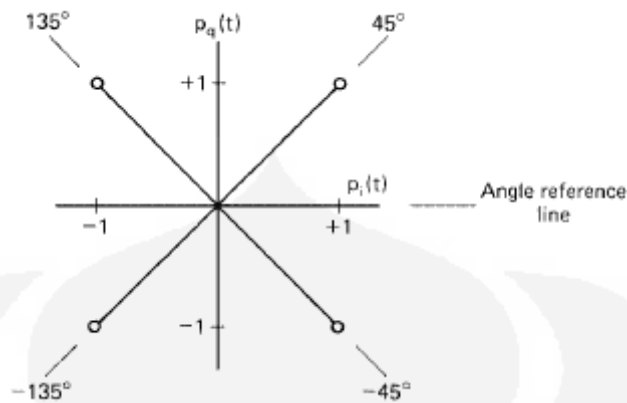
Gambar 3.4 Modulator QPSK

Dua bit stream diatas digabungkan kembali menjadi stream QPSK dengan kondisi seperti yang ditampilkan table 3.1 berikut:

$p_i(t)$	$p_q(t)$	QPSK
1	1	$\text{Cos}(\omega_0 t) - \sin(\omega_0 t) = \sqrt{2}\text{cos}(\omega_0 t + 45^\circ)$
1	-1	$\text{Cos}(\omega_0 t) + \sin(\omega_0 t) = \sqrt{2}\text{cos}(\omega_0 t - 45^\circ)$
-1	1	$-\text{Cos}(\omega_0 t) - \sin(\omega_0 t) = \sqrt{2}\text{cos}(\omega_0 t + 135^\circ)$
-1	-1	$-\text{Cos}(\omega_0 t) + \sin(\omega_0 t) = \sqrt{2}\text{cos}(\omega_0 t - 135^\circ)$

Tabel 3.1 Tabel keadaan sinyal QPSK

Kata *Quardrature* mengacu kepada empat kemungkinan keadaan fasa *carrier* (4-PSK) pada satu waktu. Empat fasa tersebut mengacu kepada sudut dari bit yang dimodulasikan, yaitu  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ . Hal ini terlihat pada gambar 3.5 berikut.



Gambar 3.5 Diagram fasa modulasi QPSK

3.1.3.2 QAM (Quadrature Amplitude Modulation)

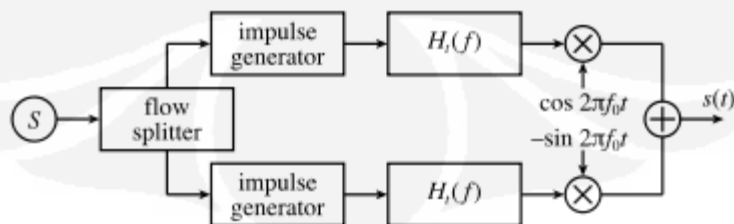
QAM merupakan modulasi dengan kombinasi fasa dan amplitude. Teknik ini mengirim sinyal dengan empat fase berbeda. Pada simulasi ini, jenis modulasi QAM yang digunakan adalah 16 QAM dan 64 QAM.

Modulasi 16 QAM memiliki 4 bit informasi untuk setiap simbolnya, sehingga memiliki  $2^4$  keadaan fasa. Disisi lain, 64 QAM memiliki  $2^6$  keadaan fasa, karena tiap simbolnya terdiri dari 6 bit informasi.

Modulasi QAM membagi sinyal yang ditransmisikan menjadi dua bagian/bit stream, yaitu *In-phase* dan *Quardrature phase*. Kedua bagian ini berbeda fasa 90 derajat, karena bit stream *In-phase* dikalikan dengan sinyal kosinus, sedangkan bagian *quardrature* dikalikan dengan sinyal sinus. Dalam pentransmisian data, kedua bagian tersebut digabungkan, menurut persamaan 3.1 berikut:

$$s(t) = I(t) \cos(2\pi f_0 t) + Q(t) \sin(2\pi f_0 t) \dots\dots\dots(3.1)$$

Berikut merupakan blok diagram modulasi dari QAM dengan frekuensi *carrier*  $f_0$  dan  $H_t$  respon frekuensi dari filter transmisi.



Gambar 3.6 Modulator QAM

Modulasi QAM memiliki keuntungan dari segi kecepatan dalam transmisi data dan efisien dalam penggunaan bandwidth. Hal ini dikarenakan banyaknya

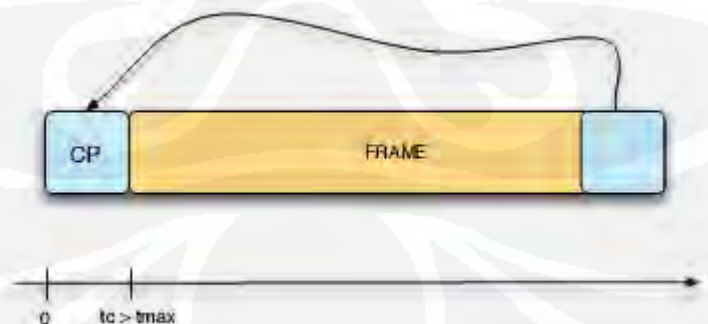
jumlah bit informasi yang diwakilkan oleh satu simbol. Makin tinggi orde modulasinya, makin cepat transmisi datanya. Namun, seiring dengan makin tingginya orde modulasi, modulasi ini makin sensitif terhadap noise.

### 3.1.4 IFFT

Parameter subkanal OFDM menggunakan parameter pada mobile wimax. Lebar masing-masing kanal bandwidth adalah 10 MHz. Dengan terdiri dari 256 *subcarrier* yang akan menjadi subkanal dengan frekuensi *carrier* antara 3400-3600 MHz atau terletak pada band frekuensi 3.5 GHz. Setiap subkanal, dimodulasi dengan IFFT sehingga data sekarang direpresentasikan dalam *time domain*. IFFT dan juga FFT pada *receiver* merupakan komponen utama dalam sistem OFDM. Operasi pemetaan linier antara data simbol kompleks dan simbol OFDM kompleks akan menghasilkan sebuah ketahanan terhadap kanal *multipath Fading*. Simbol kompleks OFDM hasil IFFT ini kemudian digabungkan kembali menjadi serial.

### 3.1.5 Cyclic Prefix

Tujuan utama penambahan *cyclic prefix* pada data adalah untuk mempertahankan properties ortogonalitas sinyal selama durasi waktu tertentu. Selain itu, *Inter Symbol Interference* (ISI) dan *Inter channel interference* (ICI) dapat dicegah dengan menambahkan *guard interval* pada awal frame.



Gambar 3.7 Penambahan *Cyclic prefix* pada frame

Gambar 3.7 memperlihatkan struktur frame setelah ditambahkan *Cyclic prefix*. *Cyclic prefix* ditambahkan pada awal frame dan diperoleh dari beberapa sampel simbol OFDM.

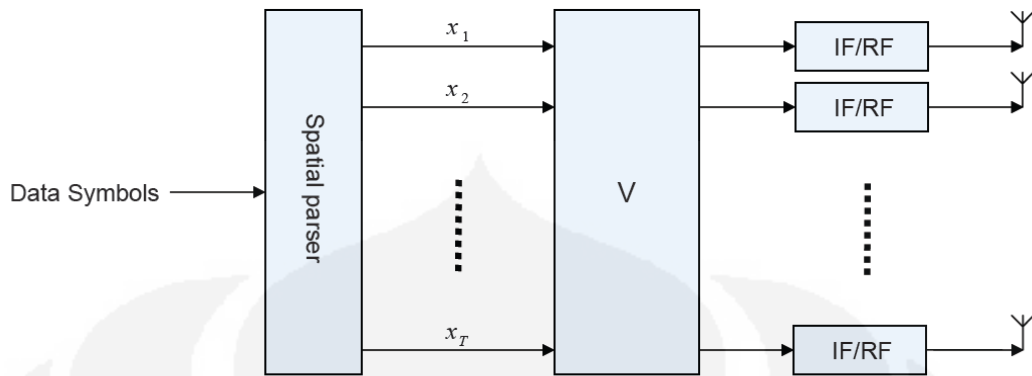
### 3.1.6 Upsampling

Upsampling merupakan proses peningkatan sampling rate dari sebuah sinyal. Up-sampling pada sistem MIMO OFDM diperlukan untuk meningkatkan akurasi sinkronisasi dan estimasi kanal. Berbeda dengan proses upsampling pada umumnya yang menyisipkan zeros diantara bit informasi, pada simulasi ini, upsampling yang digunakan adalah dengan mengulang masing-masing simbol sebanyak  $n$  kali, dimana  $n$  adalah faktor upsampling nya.

### 3.1.7 Upconversion

*Up-conversion* bertujuan untuk memudahkan transmisi simbol kompleks. Bagian real dan imajiner di-up-conversi menggunakan dua *carrier* yang saling orthogonal (sinus dan cosinus). Sinyal hasil up-conversi terbagi menjadi bagian sinyal *inphase* (I) dan *quadrature* (Q). Kedua sinyal ini kemudian dijumlahkan untuk dikirim melalui *transmitter*. Up-conversion disini menggunakan intermediate frekuensi  $F_i = 10$  KHz.

Selanjutnya, sebelum sinyal dikirim melalui antena *transmitter*, sinyal dikalikan dengan matriks  $V$  sebagaimana terlihat pada gambar 3.8, dimana matriks ini berfungsi sebagai *prefilter* bagi sinyal informasi. Hal ini dimaksudkan untuk mendapatkan simbol-simbol yang akan dipancarkan melalui masing-masing antena *transmitter*. Penggunaan *prefilter* dengan matriks  $V$  ini berkaitan dengan penggunaan *Singular Value Decomposition* pada kanal MIMO. Matriks  $V$  juga sering disebut sebagai vektor singular kanan dari matriks respon kanal  $H$ . Idanya adalah agar masing-masing kanal dapat memaksimalkan kapasitas kanal secara independen terhadap kanal yang lainnya. Selain itu kanal juga dapat digunakan pada *rate* yang optimum, independen terhadap kanal spasial yang lain sehingga dapat meningkatkan *throughput* keseluruhan sistem.



Gambar 3.8. Prefilter  $V$  transmitter MIMO berbasis SVD

### 3.2 Kanal Transmisi

#### 3.2.1. SVD

SVD (*Singular Value Decomposition*) merupakan suatu metoda untuk mendiagonalisasi suatu matriks dan mendapatkan nilai eigennya. Jika kolom dan baris dari matriks  $\mathbf{H}$  adalah independen satu terhadap yang lainnya, maka dengan menggunakan metode faktorisasi SVD, matriks kanalnya dapat didekomposisi ke dalam bentuk perkalian dari tiga matriks berikut

$$\mathbf{H} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^h \dots\dots\dots(3.2)$$

$\mathbf{U}$  dan  $\mathbf{V}$  adalah matriks unitaris non singular ortogonal dengan nilai absolut 1, dimana transposenya merupakan invers dari matriks itu sendiri.  $\mathbf{V}^h$  adalah hermit atau transposisi kompleks konjugasi matriks unitaris  $\mathbf{V}$ .  $\mathbf{U}$  memiliki dimensi  $R \times R$  dan dibentuk oleh vektor eigen ortonormal matriks korelasi  $\mathbf{H}\mathbf{H}^h$ , vektor kolom ortogonalnya memuat vektor kiri singular matriks  $\mathbf{H}$ .  $\mathbf{V}$  memiliki dimensi  $T \times T$  dibentuk oleh vektor eigen orthonormal matriks korelasi  $\mathbf{H}^h\mathbf{H}$ , vektor kolom ortogonalnya memuat vektor singular kanan matriks  $\mathbf{H}$ .  $\mathbf{\Sigma}$  adalah matriks diagonal berdimensi  $R \times T$  yang entri-entrinya adalah nilai *singular* matriks  $\mathbf{H}$  pada diagonal utama, dimana elemennya bernilai positif atau nol.

$$\sum_{i=1}^2 U_{ik} U_{in} = \delta_{kn} \quad \begin{matrix} 1 \leq k \leq 2 \\ 1 \leq n \leq 2 \end{matrix} \dots\dots\dots(3.3)$$

$$\sum_{j=1}^2 V_{jk} V_{jn} = \delta_{kn} \quad 1 \leq k \leq 2$$

$$1 \leq n \leq 2 \dots\dots\dots(3.4)$$

$$\Sigma_H = \begin{bmatrix} \lambda_{H,max} & 0 \\ 0 & \lambda_{H,max} \end{bmatrix} \dots\dots\dots(3.5)$$

Melalui SVD ini, suatu matriks didiagonalisasi, dimana hal ini berarti tercipta sub kanal spasial MIMO 2x2 secara terpisah dan independent [9]. Nilai singular pada matriks diagonal merupakan akar pangkat dua dari nilai eigen matriks:

$$D = H^*H$$

$\lambda_{H,1}$  dan  $\lambda_{H,2}$  merupakan nilai eigen dari matriks D, yang menggambarkan *channel gain*.

Vektor  $V_i$  dari matriks  $V_H$  merupakan *transmit weight factor* untuk eksitasi dari nilai singular  $\sqrt{\lambda_i}$  dan *receive weight factor* adalah matriks  $U_i^*$  pada matriks  $U_H$ . Dengan demikian, kanal independent menjadi:

$$HV_1 = \sqrt{\lambda_{H,1}}U_1 \dots\dots\dots(3.6)$$

$$HV_2 = \sqrt{\lambda_{H,2}}U_2 \dots\dots\dots(3.7)$$

Seperti yang sudah dijelaskan sebelumnya, bahwa data informasi (x) dipremultiplikasi dengan *prefilter* V dan nantinya, pada *receiver* akan terjadi perkalian sinyal yang diterima (y) dengan *postfilter*  $U^h$ . Dengan demikian, secara umum, prinsip kerja MIMO OFDM berdasarkan SVD dapat dituliskan sebagai berikut:

$$y = HVx + \eta \dots\dots\dots(3.8)$$

$$\tilde{y} = U^h y \dots\dots\dots(3.9)$$

Dengan demikian, sinyal yang diterima setelah *postfilter*, adalah:

$$\begin{aligned} \tilde{y} &= U^h(HVx + \eta) \\ &= U^h((U\Sigma V^h)Vx + \eta) \\ &= U^hU\Sigma V^hVx + U^h\eta \\ \tilde{y} &= \Sigma x + \tilde{\eta} \dots\dots\dots(3.10) \end{aligned}$$

Selanjutnya, dengan membagi persamaan diatas dengan nilai eigen matriks  $\Sigma$ , pada sisi akhir *receiver*, didapatkan kembali sinyal informasi yang dikirimkan.



### 3.2.2 AWGN dan Fading

*Fading* merupakan karakteristik utama dalam propagasi radio bergerak. *Fading* dapat didefinisikan sebagai perubahan fase, polarisasi dan atau level dari suatu sinyal terhadap waktu. Definisi dasar dari *fading* yang paling umum adalah yang berkaitan dengan mekanisme propagasi yang melibatkan refraksi, refleksi, difraksi, hamburan dan redaman dari gelombang radio.

Fenomena *fading* yang terjadi dapat dimodelkan secara matematis menurut distribusi Rayleigh, dan lebih dikenal dengan *Rayleigh Fading Model*. Pemodelan dengan *Rayleigh Fading* sering dipakai bila lingkungan dari propagasi yang ada memiliki banyak objek sehingga akan menghamburkan sinyal sebelum sampai di penerima. Pada simulasi ini, *Rayleigh Fading Model* digunakan untuk menjelaskan perubahan waktu dari selubung sinyal *fading* datar (*flat Fading*) yang diterima, atau selubung dari satu komponen multipath. Telah diketahui bahwa selubung dari jumlah antara dua sinyal derau gaussian membentuk distribusi Rayleigh.

Distribusi Rayleigh mempunyai fungsi kerapatan probabilitas (*probability density function* - pdf) diberikan oleh persamaan 3.11 berikut [10][11]:

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) & (0 \leq r \leq \infty) \\ 0 & (r < 0) \end{cases} \quad \dots\dots\dots(3.11)$$

dengan  $\sigma^2$  merupakan daya rata-rata sinyal yang diterima terhadap waktu sebelum deteksi tegangan (envelope) dan  $r$  nilai mean

Pada simulasi ini, juga digunakan kanal Gaussian, yaitu kanal yang dipengaruhi oleh *Additive White Gaussian Noise* (AWGN). AWGN adalah suatu derau *thermal* yang merusak sinyal dalam bentuk aditif. Artinya, derau ini ditambahkan ke dalam sinyal utama yang memiliki kepadatan spektrum daya yang sama untuk semua daerah frekuensi. Kanal ini tidak memperhitungkan faktor-faktor perusak seperti *fading*, sehingga bisa disebut kanal ideal [10][11].

AWGN dan *rayleigh Fading* digunakan untuk merepresentasikan kanal transmisi yang sesuai dengan kanal transmisi sebenarnya. Gangguan AWGN divariasikan dengan parameter *Signal to Noise Ratio* (SNR) dan gangguan *rayleigh fading* divariasikan dengan parameter kecepatan gerak relatif *mobile*

*station* dengan *base station* ( $v$ ). Variasi kecepatan gerak juga merupakan variasi pergeseran maksimum frekuensi Doppler (*Doppler shift*). Nilai realistik untuk *Doppler Shift* adalah antara 4 Hz untuk gerak lambat (*pedestrian*) hingga 80 Hz untuk gerak sangat cepat (*full mobility*). Keadaan *multipath* juga direpresentasikan oleh gangguan *rayleigh Fading*. Keadaan *multipath* ini digambarkan memiliki 3 *path* yang datang dengan waktu tunda yang berbeda-beda. *Delay spread* untuk keadaan *multipath* ini adalah  $4 \cdot 10^{-8}$  detik dan termasuk *delay spread* untuk keadaan *indoor* yang bernilai antara  $1 \cdot 10^{-7}$  detik hingga  $1 \cdot 10^{-9}$  detik. *Gain* untuk setiap *path* yang datang biasanya bernilai antara -20 dB hingga 0 dB.[12]

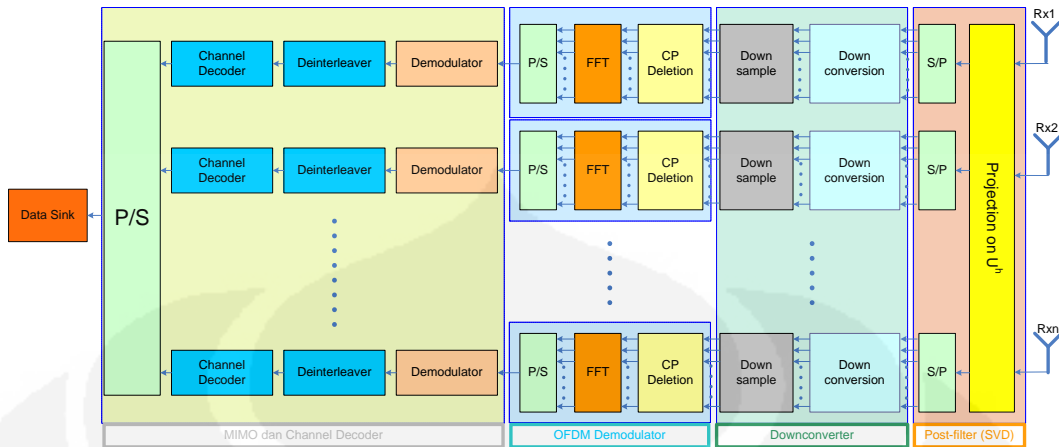
### 3.2.3 Least Square

*Least square* digunakan untuk mengestimasi respon kanal sistem. Estimasi kanal digunakan untuk mendapatkan nilai matriks respon kanal  $H$ . Estimasi kanal dilakukan pada frame pertama yang berisikan *training sequence*. Pada *transmitter*, dilakukan *frame packing* dimana frame pertama akan berisikan *training sequence*, sedangkan frame selanjutnya berisikan data informasi beserta *guard interval* (*Cyclic prefix*). Penyusunan ini akan mempermudah pemrosesan data, dimana sebelum memroses data informasi, dilakukan estimasi kanal terlebih dahulu.

Sebelum diestimasi, sistem menggunakan suatu matriks kanal  $H$  yang sudah didefinisikan, namun nilai dari matriks kanal tersebut tidak diketahui. Disini, *least square* bertujuan untuk mendapatkan nilai matriks respon kanal tersebut.

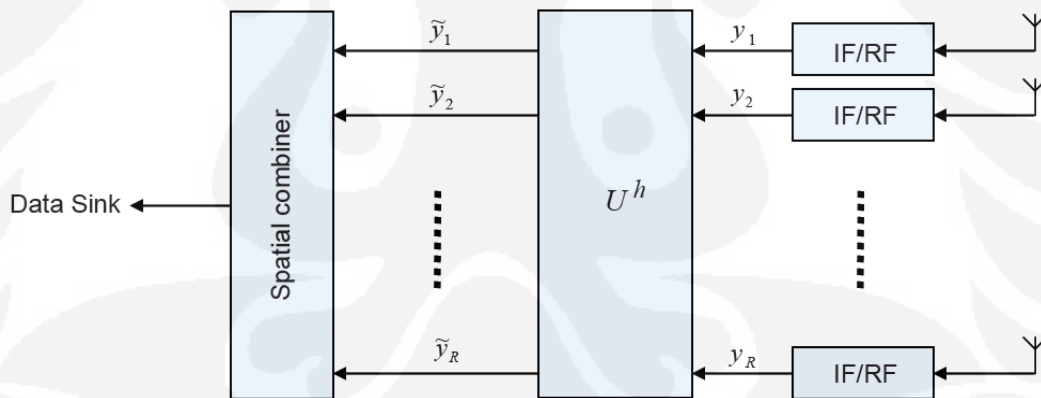
### 3.3 Receiver

Blok diagram dari sisi *receiver* diilustrasikan pada gambar 3.8. Pada gambar 3.8 tersebut, terlihat tahap-tahap pemrosesan sinyal, mulai dari penerimaan sinyal oleh antena penerima, sampai diperolehnya kembali data informasi.



Gambar 3.9 Model *receiver* sistem yang disimulasikan

Pada *receiver*, sinyal yang diterima melakukan perlakuan yang merupakan kebalikan dari proses pada *transmitter*. Karena menggunakan operasi SVD, maka simbol yang diterima oleh antenna penerima akan diteruskan terlebih dahulu ke *postfilter* untuk dikalikan dengan vektor kolom ortogonal matriks demultipleks  $U^h$ , yang juga sering disebut sebagai vektor singular kiri dari matriks respon kanal  $H$ . Hal ini diilustrasikan pada gambar 3.10.



Gambar 3.10. *Postfilter*  $U$  receiver MIMO berbasis SVD

### 3.3.1. Down-Conversion

Sinyal dari antenna penerima merupakan sinyal *passband*. Untuk pemrosesan pada sisi penerima, maka sinyal ini dikonversi menjadi sinyal *baseband*.

### 3.3.2 Down sampling

Selanjutnya, sinyal di-*down sample* dengan *rate* yang merupakan kebalikan dari *rate upsampling* pada *transmitter*. Dengan demikian, *rate* untuk *down sampling* adalah  $1/L$  ( $1/4$ ), karena pada sisi transiter *rate* dari *upsampling* adalah 4.

### 3.3.3. Penghapusan cyclic prefix

Proses merupakan penghapusan cyclic prefix yang sebelumnya ditambahkan pada sisi *transmitter*.

### 3.3.4 FFT

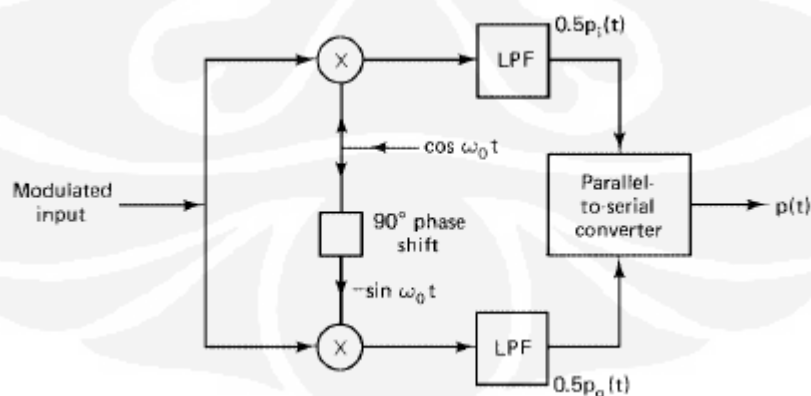
Data ditransmformasikan kembali ke frekuensi domain menggunakan fungsi *Fast Fourir Transform* (FFT) yang merupakan kebalikan dari fungsi IFFT.

### 3.3.5 Demodulasi

Dengan demodulasi, simbol ditransformasikan kembali ke dalam bentuk bits, tergantung jenis modulasi yang digunakan pada sisi *transmitter*. Sebelum didemodulasi, terlebih dahulu dilihat konstelasi dari sinyal yang diterima.

#### 3.3.5.1 Demodulasi QPSK

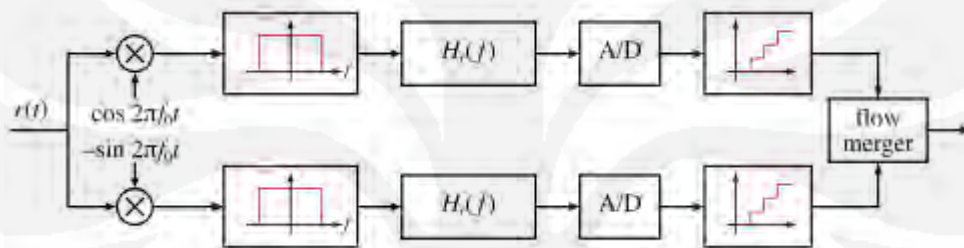
Proses demodulasi ini merupakan kebalikan dari proses modulasi pada sisi *transmitter*. Setiap simbol dipetakan kembali menjadi bit bit informasi, seperti terlihat pada gambar 3.9 berikut:



Gambar 3.11 Demodulator QPSK

### 3.3.5.2 Demodulasi QAM

Demodulasi dilakukan dengan konsep *demodulator* ASK yang kemudian masing masing bit stream digabung kembali menjadi satu bit stream sinyal informasi. Pada praktiknya, terdapat *phase delay* antara *transmitter* dan *receiver* yang nantinya dapat diatasi dengan menggunakan *synchronization* pada bagian *receiver*. Pada aplikasi *mobile*, terdapat efek *Doppler shift* yang menggeser nilai frekuensi asli pada *transmitter*, sehingga diperlukan adanya tuning pada komponen kosinus dan sinus yang memerlukan phase reference dengan menggunakan *phase lock loop*.



Gambar 3.12 Demodulator QAM

### 3.3.6. Deinterleaver

Merupakan proses yang berkebalikan dengan *interleaver*, dimana bits akan mengisi matriks kolom per kolom. Output dari *deinterleaver* merupakan matriks baris.

### 3.3.7 Convolutional decoder

*Decoder* akan membentuk algoritma *Viterbi decoding* untuk menghasilkan bit data informasi dari bit yang dikodekan.

## BAB IV

### HASIL SIMULASI DAN ANALISIS

Simulasi MIMO OFDM dengan teknik *spatial multiplexing* ini menggunakan berbagai macam parameter, yang mana dapat dilihat pada tabel 4.1. Pada simulasi, digunakan tiga jenis modulasi yaitu modulasi QPSK, QAM 16 dan QAM 64. Selain itu, simulasi juga dilengkapi dengan *Convolutional Encoding* dimana terdiri dari tiga jenis *code rate*.

#### 4.1 Parameter Performansi Sistem

Dalam simulasi MIMO OFDM ini, unjuk performansi dari sistem dapat dilihat dari tiga jenis parameter, yaitu Bit Error Rate (BER), Kapasitas kanal dan *Throughput*, dimana ketiga jenis parameter tersebut pada simulasi ini bergantung pada variasi nilai EbNo.

Sistem parameter	Nilai Parameter
Teknik Modulasi	QPSK, 16 QAM dan 64 QAM.
Error Control Coding	Convolutional coderate 2/3, 1/2 dan 1/3
Jumlah <i>subcarrier</i>	256
Jumlah data <i>subcarrier</i>	256
Jumlah pilot <i>subcarrier</i>	256
Durasi simbol OFDM	16 $\mu$ s
Guard Interval	3.2 $\mu$ s
Frekuensi <i>carrier</i>	3.6 GHz

Tabel 4.1 Parameter Simulasi MIMO OFDM

Tabel diatas merupakan parameter umum yang digunakan dalam simulasi ini. Simulasi ini, terdiri atas tiga pilihan modulasi, yaitu QPSK, 16 QAM dan 64 QAM, yang nantinya akan dilihat performa dari ketiga modulasi tersebut terhadap sistem yang disimulasikan. Teknik ECC yang digunakan disini, terdiri atas 3 jenis,

dimana jenis ECC tersebut tergantung pada nilai *code rate* yang digunakan. Selain itu, sistem juga bisa disimulasikan tanpa menggunakan ECC sama sekali. Jumlah *subcarrier* menunjukkan pembagian *carrier* untuk transmisi OFDM, yaitu 256. Masing masing *subcarrier* nantinya akan membawa data informasi. Khusus untuk frame pertama, *subcarrier* hanya membawa data pilot (*training sequence*) yang digunakan untuk estimasi kanal pada sisi *receiver*. Jumlah *training sequence* yang digunakan adalah 256 bit. Frame selanjutnya berisikan data informasi, dimana data informasi tersebut berukuran 256 bit, dan ditransmisikan mulai dari frame ke 2 sampai frame terakhir. Masing masing simbol OFDM, mempunyai durasi antar simbol sebesar 16  $\mu$ s, dimana nilai ini merupakan asumsi yang digunakan pada simulasi ini. Asumsi lain yang digunakan adalah waktu durasi *cyclic prefix* untuk masing masing simbol adalah 3.2  $\mu$ s. Baris terakhir dari table diatas adalah frekuensi *carrier*, yaitu sebesar 3.6 GHz, yang merupakan frekuensi *carrier* untuk WiMAX.

#### 4.1.1 Bit Error Rate

BER merupakan parameter yang paling utama dalam menentukan kualitas suatu sistem. BER yang tinggi menandakan, informasi yang diterima pada sisi *receiver* mengalami banyak perubahan selama ditransmisikan. Pada simulasi ini, BER dihitung dengan membandingkan sinyal hasil generator data dengan sinyal hasil demodulasi pada sisi *receiver*. Prinsip kerja penghitungan BER disini, secara umum merupakan fungsi xor antara dua data, dimana jika hasil xor tersebut bernilai 1, maka terdapat satu bit yang error. Penghitungan BER terus berlanjut, sampai semua bit pada sinyal informasi di *transmitter* di-xor-kan dengan sinyal hasil demodulasi pada sisi *receiver*. Persentase BER dihitung berdasarkan rasio antara jumlah bit yang eror dengan jumlah bit total.

#### 4.1.2 Kapasitas Kanal

Sebagaimana yang sudah dijelaskan pada subbab 2.1.2, kapasitas kanal yang digunakan pada simulasi ini memakai rumusan Gans dan Foschini yang berdasar pada teori informasi Shannon, dimana persamaannya adalah [4]:

$$C = \log_2 \left( \det \left( I_{R_0} + \frac{SNR}{T} HH^h \right) \right) \text{ bits / s. / Hz.} \dots\dots\dots(4.1)$$

$I_{R_0}$  adalah suatu matriks identitas berdimensi  $R_0$  yang dilihat dari sisi *receiver*, T adalah jumlah antena *transmitter*, SNR adalah rasio *signal to noise*, dan H adalah matriks respon kanal.

Pada sistem OFDM, dimana setiap sinyal berada dalam deretan waktu juga dapat dikembangkan sebagai sinyal *narrow-band* dalam domain frekuensi, yang mana karakteristik tiap subkanal dapat didefinisikan sebagai frekuensi gelombang *carrier* yang berhubungan dengan sinyal tersebut. Oleh karena itu, persamaan kapasitas kanal diatas secara langsung dapat diterapkan untuk menganalisis metode transmisi informasi MIMO-OFDM dan karakteristik transmisinya yang relevan. Bila jumlah diskrit subkanal frekuensi *flat* dalam kanal sistem dinotasikan dengan N, maka kapasitas subkanalnya dapat ditulis

$$C = \frac{1}{N} \sum_{i=1}^{N-1} \log_2 \left( \det \left( I_{R_0} + \frac{SNR}{T} H(i)H(i)^h \right) \right) \text{ bits / s. / Hz.} \dots\dots\dots(4.2)$$

Karena sistem MIMO OFDM disini menggunakan operasi SVD, maka kapasitas kanal tersebut dapat diformulasikan ke dalam model ekuivalen paralel independen

$$C = \sum_{i=1}^{rank(H)} \log_2 \left( 1 + \frac{SNR}{T} (\delta_i^f)^2 \right) \text{ bits / s. / Hz.} \dots\dots\dots(4.3)$$

Dimana  $\delta_i$  merupakan nilai eigen dari matriks diagonal S.

### 4.1.3 Throughput

Berdasarkan [13], *throughput* sistem dinyatakan dengan fungsi maksimum *datarate* R yang dapat dicapai sistem dan besarnya PER (*Packet Error Rate*), sebagaimana terlihat pada persamaan 4.4.

$$\delta = R \cdot \{1 - PER(\gamma)\} \dots\dots\dots(4.4)$$

Dengan  $\delta$  adalah besar nilai *throughput* dan R adalah data rate transmisi. Sedangkan PER adalah *packet error rate* pada nilai SNR  $\gamma$ .

Dari persamaan 4.4, terlihat *throughput* sistem ditentukan oleh besarnya PER, dimana PER dipengaruhi juga oleh BER sistem. Dengan demikian, BER secara langsung juga mempengaruhi *throughput* sistem, dimana BER yang tinggi



akan membuat *throughput* sistem rendah.

Teknik *Spatial multiplexing* MIMO yang dipakai pada simulasi ini, mempunyai keuntungan dalam hal meningkatkan *throughput* sistem, karena data yang dikirimkan merupakan pembagian dari data utama, tergantung jumlah antena yang digunakan.

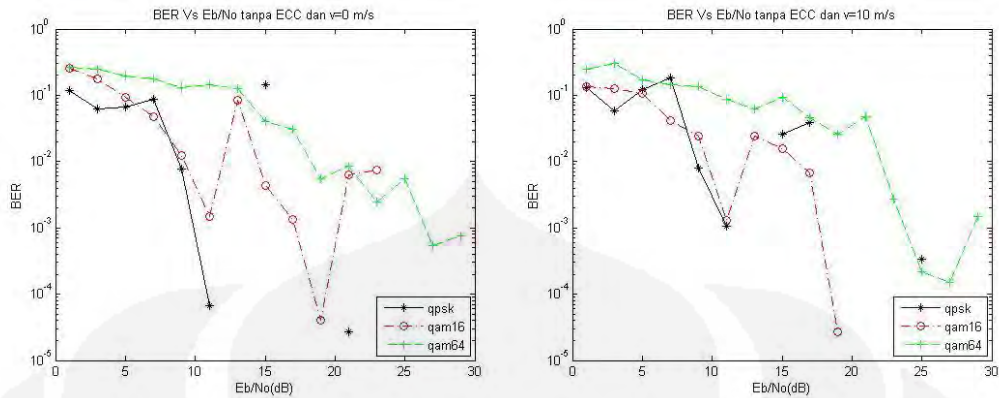
## 4.2 Hasil Simulasi dan Analisis

Hasil simulasi secara umum dibagi menjadi dua bagian, yaitu grafik performansi sistem, berupa BER, kapasitas kanal, *throughput* terhadap EbNo dan gambar hasil konstelasi. Performansi sistem, dianalisis dengan kondisi sistem yang menggunakan ECC dan tanpa menggunakan ECC. Selain itu, terdapat kondisi lain yang digunakan pada simulasi ini dalam melihat performa sistem, yaitu kondisi *User* / penerima diam ( $V = 0$  m/s) dan kondisi *user* / penerima bergerak ( $V = 10$  m/s).

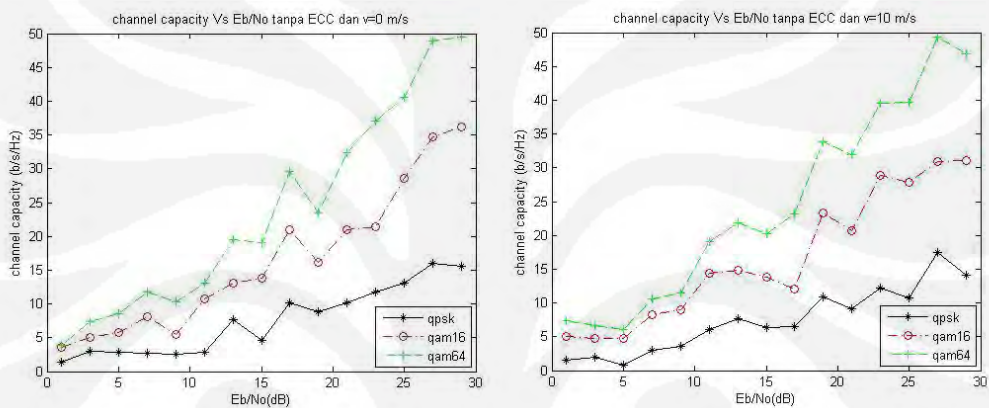
### 4.2.1 Simulasi Berdasarkan ECC

#### 4.2.1.1 Simulasi dengan $ECC = 0$ (Tanpa menggunakan ECC / Code rate $ECC = 1$ )

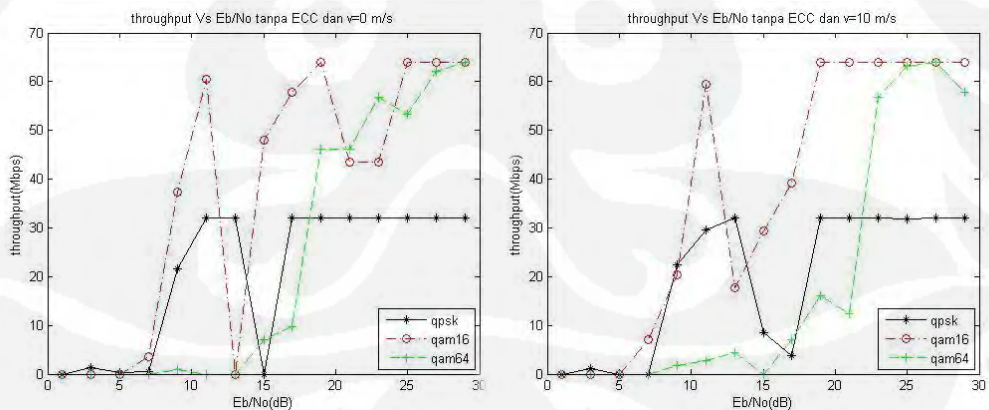
Simulasi pertama dilakukan dengan kondisi tanpa menggunakan ECC. Unjuk kerja dari sistem, sesuai dengan parameter performansi yang sudah didefinisikan diatas, yaitu BER, kapasitas kanal dan *throughput* dapat dilihat pada gambar 4.1, 4.2, 4.3, secara berurutan. Terdapat dua kondisi pada ketiga gambar tersebut. Grafik pertama (sebelah kiri) adalah *user* dalam keadaan diam ( $v = 0$  m/s), sedangkan grafik kedua (sebelah kanan) merupakan unjuk kerja sistem dengan *user* bergerak ( $v = 10$  m/s).



Gambar 4.1 BER vs Eb/No ECC = 0



Gambar 4.2 Kapasitas Kanal vs Eb/No ECC = 0



Gambar 4.3 Throughput vs Eb/No ECC = 0

Unjuk kerja parameter BER , terlihat pada gambar 4.1. BER untuk sistem dengan kondisi *user* diam lebih kecil dibanding sistem dengan *user* bergerak. Hal ini terlihat jelas dengan melihat kecenderungan grafik seiring makin naiknya nilai Eb/No, dimana grafik untuk *user* diam memiliki rata rata BER yang jauh lebih

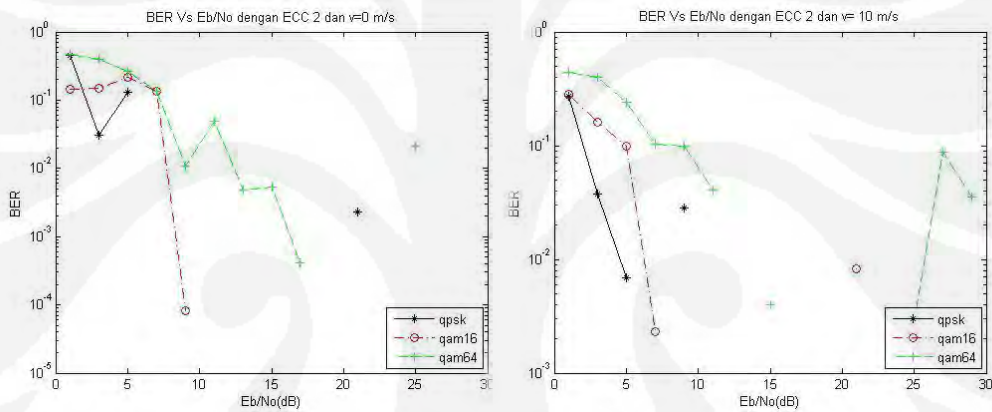
kecil secara keseluruhan. Hal ini dikarenakan, untuk *user* yang bergerak, transmisi datanya dipengaruhi oleh parameter kecepatan atau dengan kata lain dipengaruhi oleh efek pergeseran Doppler. Selain itu, dengan nilai  $E_b/N_0$  yang sama, terlihat juga BER sistem yang menggunakan modulasi 64 QAM lebih besar dibanding dua jenis modulasi lainnya. Sedangkan, untuk tipe modulasi 16 QAM, nilai BERnya relative lebih tinggi dibanding modulasi QPSK. Kedua hal tersebut dilihat sebagai sesuatu yang wajar, karena modulasi QPSK hanya menggunakan 2 simbol, sedangkan modulasi 16 QAM dan 64 QAM memodulasikan 4 dan 6 simbol, sehingga lebih rentan terhadap *noise* selama transmisi. Grafik BER untuk modulasi QPSK, dan juga 16 QAM terlihat adanya garis yang putus. Bagian yang putus menunjukkan BER bernilai nol. Secara keseluruhan dapat dilihat bahwa seiring dengan makin naiknya nilai  $E_b/N_0$ , BER dari sistem makin kecil. Hal ini dikarenakan dengan naiknya nilai  $E_b/N_0$ , maka SNR makin tinggi sehingga secara langsung berpengaruh terhadap kenaikan nilai BER sistem.

Unjuk kerja kapasitas kanal dapat dilihat pada gambar 4.2. Dari gambar tersebut terlihat bahwa kapasitas kanal makin meningkat seiring bertambahnya nilai  $E_b/N_0$ . Selain itu, kapasitas kanal untuk modulasi 64 QAM lebih besar dibanding dua jenis modulasi lainnya. Dari grafik juga terlihat, kapasitas kanal sistem dengan *user* bergerak sedikit lebih rendah dibanding keadaan *user* diam. Namun, perbedaan tersebut tidak terlalu signifikan.

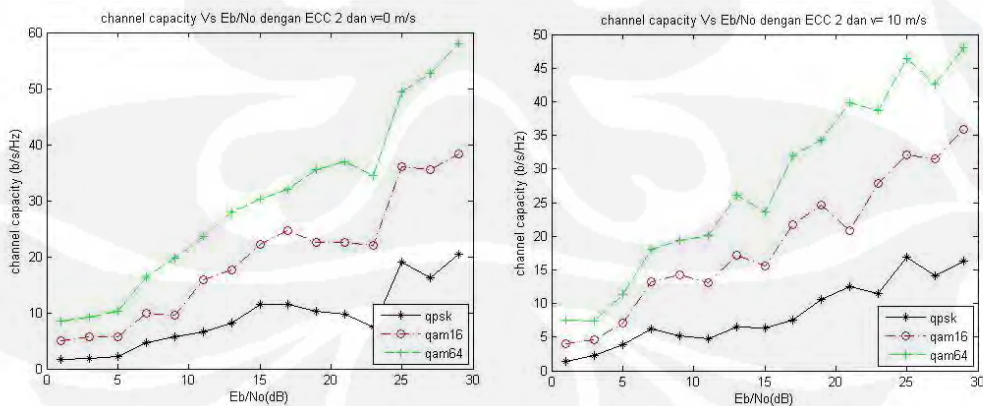
Gambar 4.3 merepresentasikan *throughput* dari sistem dengan berbagai variasi nilai  $E_b/N_0$ . Pada gambar, terlihat adanya grafik yang mendatar dan ada yang naik turun. Grafik yang mendatar, menandakan *throughput* yang maksimal, yang berarti nilai BER adalah nol atau mendekati nol. Dapat kita lihat, untuk modulasi QAM 16 dan QPSK, BER nya bernilai nol pada saat  $E_b/N_0$  nya adalah 19 dB sampai 30 dB. Sehingga grafik *throughput* untuk nilai  $E_b/N_0$  dari 19 dB sampai 30 dB adalah datar, dengan *throughput* maksimal, yaitu bernilai 65 Mbps untuk 16 QAM dan 32 Mbps untuk QPSK. Yang menjadi perhatian adalah grafik *throughput* untuk modulasi 64 QAM, dimana mempunyai BER yang cukup tinggi, sehingga secara langsung mempengaruhi *throughput* dari sistem, dimana nilai *throughput* berfluktuasi, dan tidak mencapai *throughput* maksimal, karena nilai BER yang dicapai, tidak pernah bernilai nol.

#### 4.2.1.2 Simulasi dengan ECC = 2 (Code rate ECC = 1/2)

Hasil simulasi dapat dilihat pada gambar 4.4, 4.5, 4.6. Untuk grafik BER, secara umum dapat dilihat bahwa BER untuk kondisi *user* diam, BER nya relatif lebih rendah dibanding *user* dalam keadaan bergerak. Modulasi QPSK dan 16 QAM, nilai BER-nya hampir sama, dan mencapai nilai nol masing masing pada  $E_b/N_0 = 5\text{dB}$  dan  $9\text{ dB}$  secara berurutan untuk QPSK dan 16 QAM. Perbedaan yang sedikit mencolok terlihat pada modulasi 64 QAM, dimana BER untuk kondisi *user* diam, pada  $E_b/N_0$  yang sudah tinggi ( $17\text{dB}$ ) sudah mencapai nol, sedangkan untuk kondisi *user* bergerak, masih terdapat nilai BER yang cukup besar.



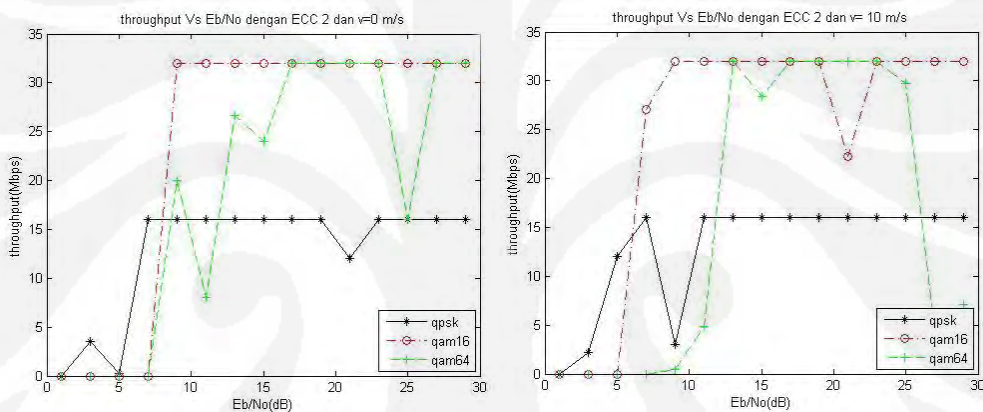
Gambar 4.4 BER vs Eb/No ECC = 2



Gambar 4.5 Kapasitas Kanal vs Eb/No ECC = 2

Unjuk kerja kapasitas kanal terlihat pada gambar 4.5. Dari Gambar 4.5 terlihat sistem dengan kondisi *user* diam dan *user* bergerak, kapasitas kanalnya tidak terlalu berbeda. Sedangkan performa parameter *throughput* terlihat pada gambar 4.6. Dari grafik terlihat, kapasitas kanal maksimal dari kedua kondisi

sistem adalah hampir sama, untuk masing masing modulasi yang digunakan. *Throughput* dari sistem dipengaruhi oleh besarnya BER. Dengan demikian, pada saat BER nya bernilai nol, maka kapasitas kanal akan maksimal. Seperti yang terlihat pada Gambar 4.6, untuk sistem dengan *user* diam, kapasitas kanal untuk modulasi QAM, mencapai nilai maksimalnya pada saat  $E_b/N_0 = 9$  dB. Begitu juga dengan kondisi *user* bergerak. Dengan demikian, secara umum, dapat disimpulkan bahwa, performa sistem untuk kondisi *user* diam dan bergerak ( $v = 10$  m/s) tidak mempunyai perbedaan yang signifikan. Namun, untuk lebih khususnya, performa sistem dengan kondisi *user* diam sedikit lebih bagus dibanding sistem dengan *user* bergerak.



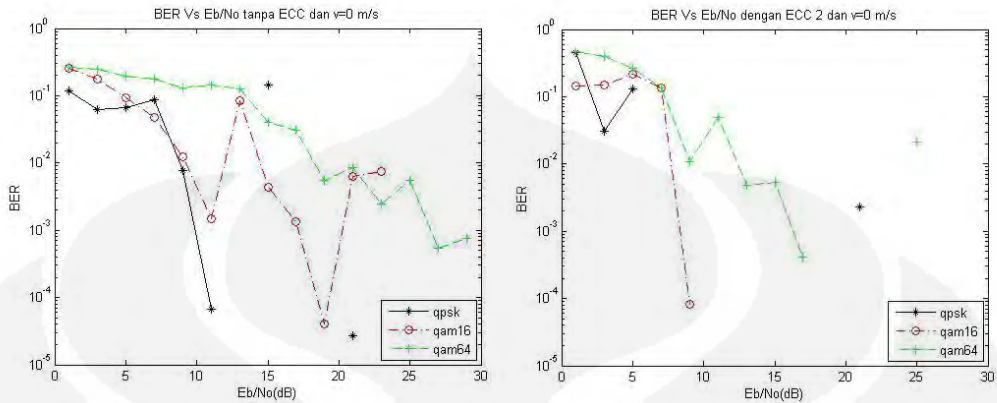
Gambar 4.6 *Throughput vs Eb/No* ECC = 2

## 4.2.2 Simulasi Berdasarkan Kecepatan

### 4.2.2.1. Simulasi dengan kecepatan 0 m/s

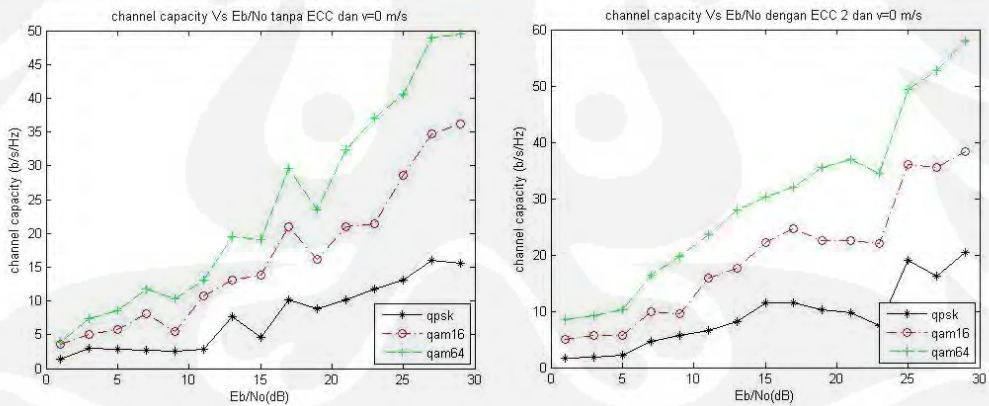
Gambar 4.7 menampilkan unjuk kerja sistem berdasarkan parameter BER. Dari gambar tersebut dapat dilihat secara jelas nilai BER untuk sistem yang menggunakan ECC jauh lebih baik dibanding sistem tanpa menggunakan ECC. Hal ini telah membuktikan bahwa penggunaan ECC sangat signifikan pengaruhnya dalam meminimalisasi *bit error rate* selama transmisi data. Dari grafik terlihat bahwa sistem dengan menggunakan ECC, untuk modulasi QPSK, 16 QAM, dan 64 QAM telah mencapai BER bernilai nol pada saat  $E_b/N_0 = 7$  dB, 9 dB dan 17 dB. Sedangkan untuk kondisi sistem tanpa menggunakan ECC, nilai

BER = 0 baru dicapai pada saat  $E_b/N_0 = 11$  dB dan 23 dB untuk modulasi QPSK dan 16 QAM. Untuk modulasi 64 QAM, nilai BER nya tidak mencapai nol.



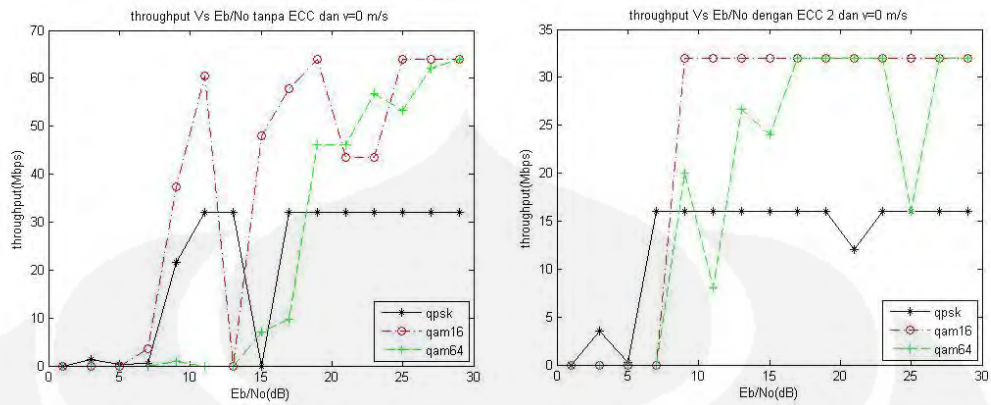
Gambar 4.7 BER vs  $E_b/N_0$   $V = 0$  m/s

Unjuk kerja parameter kapasitas kanal, direpresentasikan oleh Gambar 4.8. Dari gambar tersebut, terlihat bahwa kapasitas kanal dengan sistem tanpa menggunakan ECC lebih rendah dibanding sistem yang menggunakan ECC. Hal ini dikarenakan pengaruh BER terhadap kapasitas kanal sistem. Sehingga, sistem dengan nilai BER yang kecil, akan mempunyai kapasitas kanal yang besar.



Gambar 4.8 Kapasitas Kanal vs  $E_b/N_0$   $V = 0$  m/s

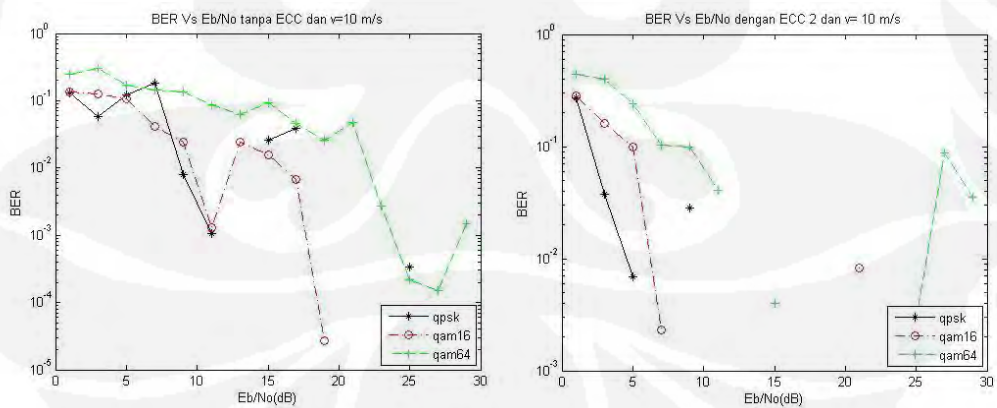
Dari segi parameter *throughput*, analisisnya sama dengan parameter sebelumnya, dimana *throughput* yang bagus dicapai dengan sistem yang menggunakan ECC, karena parameter *throughput* dipengaruhi oleh nilai BER. *Throughput* yang maksimal dicapai ketika  $BER = 0$ . Dengan demikian, karena sistem dengan ECC mencapai BER yang bernilai nol pada  $E_b/N_0$  yang lebih kecil dibanding sistem tanpa menggunakan ECC, maka *throughput* maksimalnya juga didapat lebih cepat.



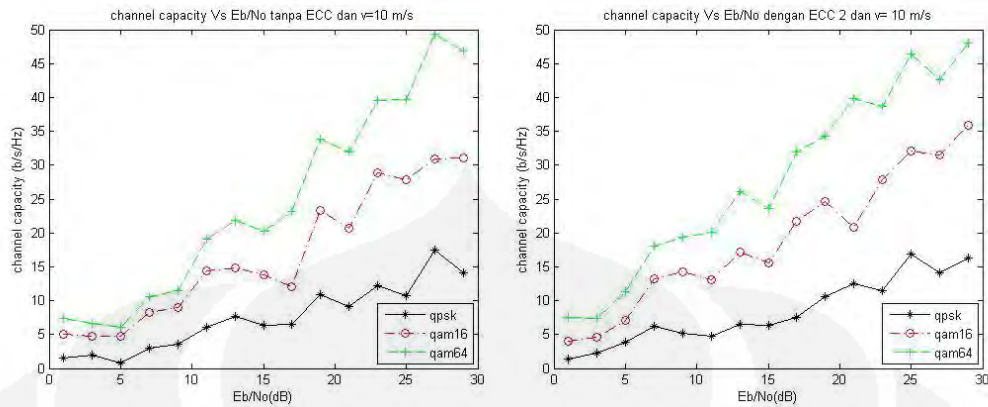
Gambar 4.9 Throughput vs Eb/No V = 0 m/s

#### 4.2.2.2 Simulasi dengan kecepatan 10 m/s

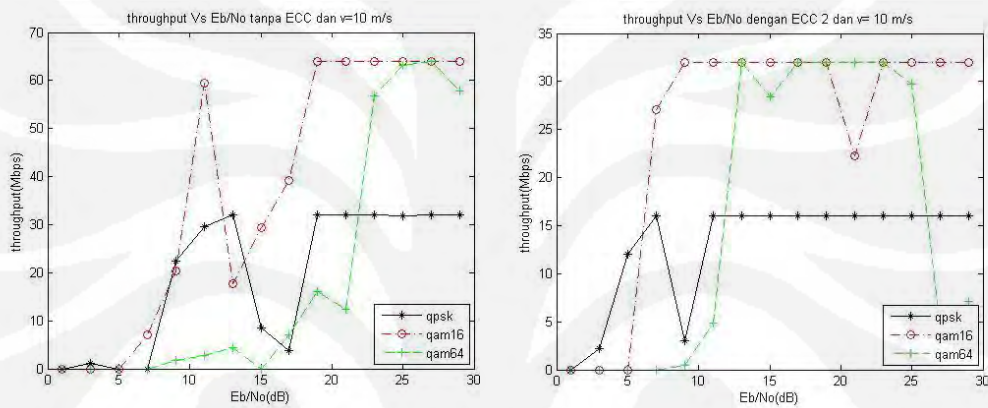
Hasil simulasi sistem dengan kondisi *user* bergerak, dapat dilihat pada Gambar 4.10, 4.11, 4.12 masing masing untuk unjuk kerja BER, kapasitas kanal dan *throughput* sistem secara berurutan. Analisis hasil simulasi hampir sama dengan sistem dengan kondisi *user* diam, dimana sistem yang menggunakan ECC mempunyai performansi yang jauh lebih baik dibanding sistem tanpa menggunakan ECC, baik itu untuk parameter BER, kapasitas kanal, maupun *throughput*.



Gambar 4.10 BER vs Eb/No V = 10 m/s



Gambar 4.11 Kapasitas kanal vs Eb/No V = 10 m/s



Gambar 4.12 Throughput vs Eb/No V = 10 m/s

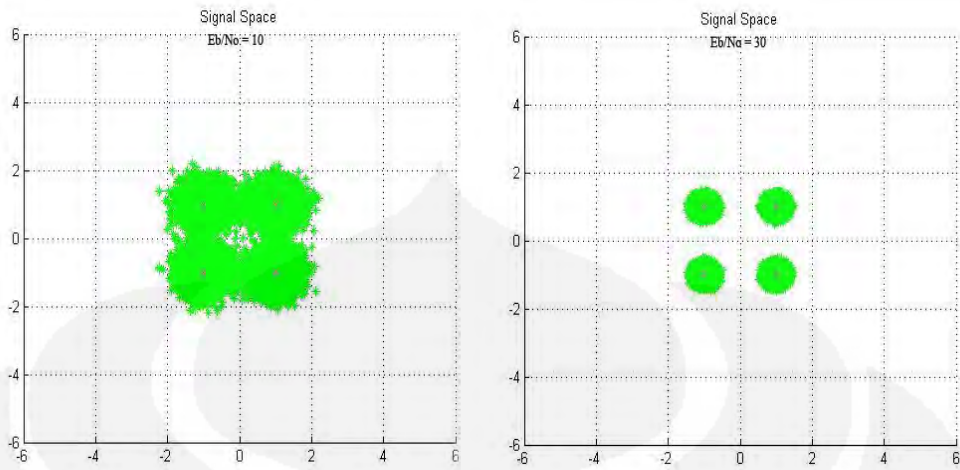
### 4.2.3 Sinyal Konstelasi

Pada bagian ini, akan dibahas mengenai sinyal konstelasi (*Signal Space*) yang merepresentasikan sinyal yang diterima pada sisi *receiver*. Simulasi dilakukan berdasarkan dua parameter, yaitu Eb/No dan kecepatan.

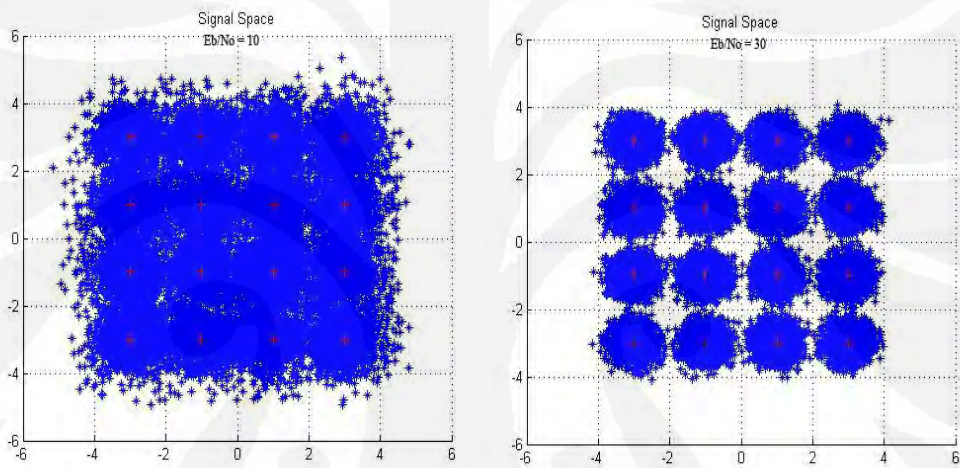
#### 4.2.3.1 Sinyal Konstelasi Berdasarkan Eb/No

Gambar 4.13, 4.15, 4.16 menampilkan sinyal konstelasi dari modulasi QPSK, 16 QAM dan 64 QAM secara berurutan. Gambar tersebut dihasilkan berdasarkan dua jenis nilai Eb/No, yaitu 10 dB dan 30 dB. Teknik ECC yang dipakai adalah ECC dengan  $code\ rate = \frac{1}{2}$ , dan sistem disimulasikan dengan kondisi *user* diam ( $v = 0$  m/s). Pada gambar hasil konstelasi tersebut, terdapat 2 jenis titik, yaitu (+) dan (\*). Titik dengan tanda (+) menandakan sinyal hasil modulasi pada sisi *transmitter*, sedangkan tanda (\*) merepresentasikan sinyal pada sisi *receiver*.

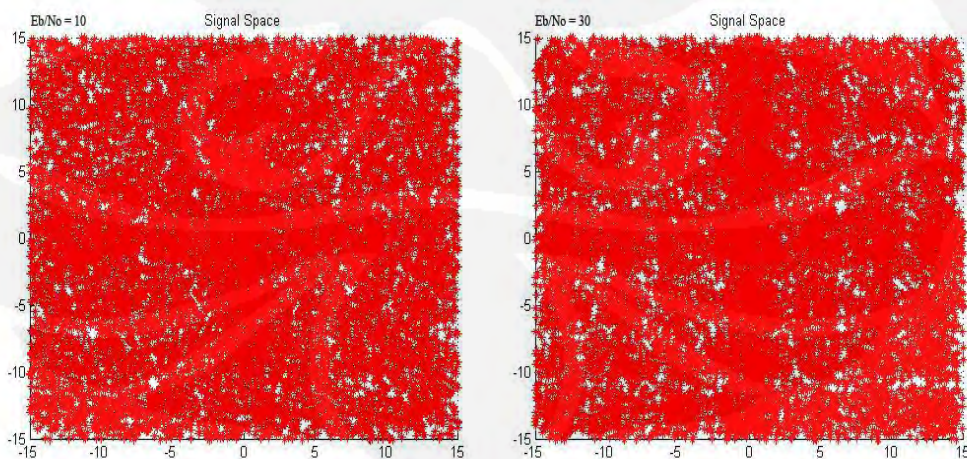




Gambar 4.13 Sinyal konstelasi QPSK berdasarkan Eb/No



Gambar 4.14 Sinyal konstelasi 16 QAM berdasarkan Eb/No



Gambar 4.15 Sinyal konstelasi 64 QAM berdasarkan Eb/No

Gambar 4.13 merupakan sinyal konstelasi dari modulasi QPSK. Dari gambar tersebut, data dilihat bahwa sinyal konstelasi untuk Eb/No yang lebih tinggi (30 dB) lebih baik. Pada grafik terlihat juga, dengan Eb/No yang tinggi, maka sinyal pada sisi penerima (\*) lebih banyak berkumpul disekitar sinyal pada sisi pengirim (+).

Hal yang sama juga terlihat pada Gambar 4.14 dan 4.15 yang merepresentasikan sinyal konstelasi dari modulasi 16 QAM dan 64 QAM. Untuk nilai Eb/No yang tinggi, sinyal konstelasinya terlihat lebih bagus, ditandai dengan halusnnya pola konstelasi yang terbentuk.

Dengan demikian, dari ketiga gambar diatas, dapat disimpulkan bahwa sinyal konstelasi dipengaruhi secara langsung oleh besar nilai Eb/No. Alasannya adalah, nilai Eb/No menentukan besarnya nilai SNR , yang hubungannya dapat dilihat pada persamaan berikut:

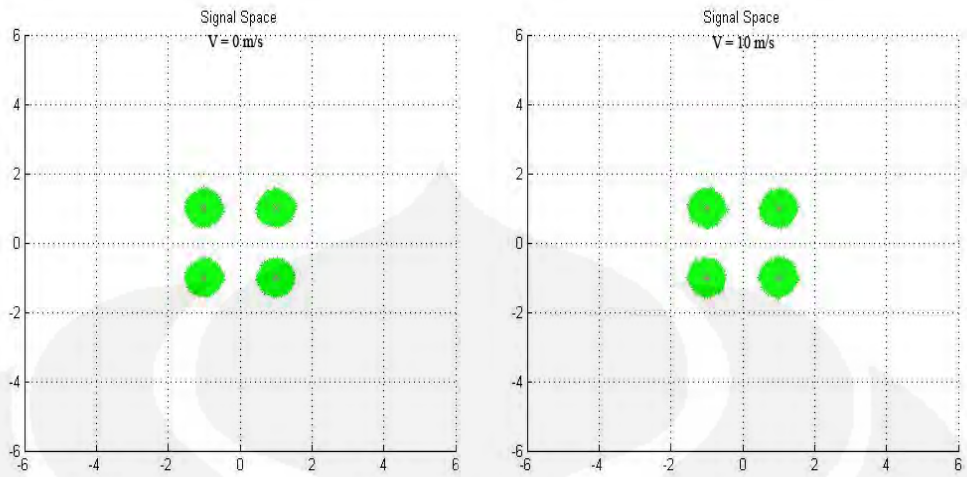
$$SNR = \frac{Eb}{No} + 10\log_{10}(k \times coderate) + 10\log_{10}(nsamp) \dots\dots(4.5)$$

Dimana k merupakan indeks modulasi, yaitu k=2 untuk QPSK, k=4 untuk 16-QAM dan k=6 untuk 64-QAM. *Coderate* menyatakan orde dari *Convolutional Encoding* (pada simulasi ini, *coderate* yang dipakai adalah 1/2), dan *nsamp* merupakan banyak sample dari sinyal simbol.

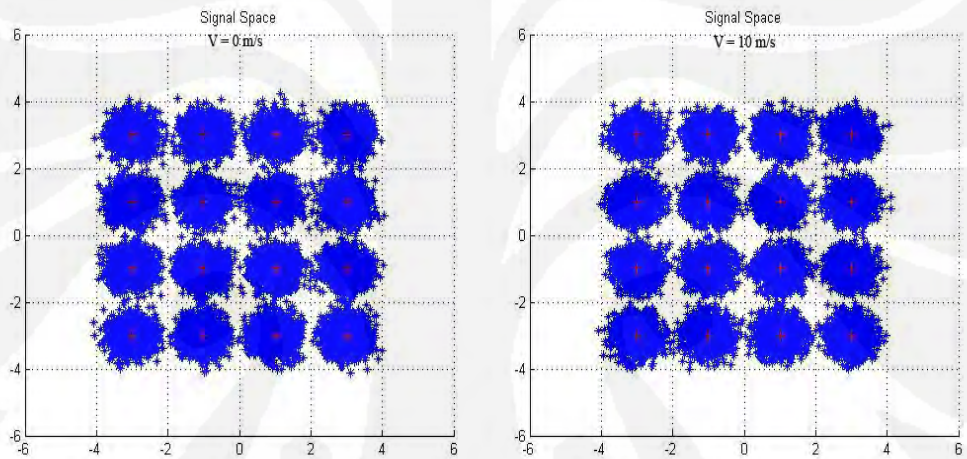
Berdasarkan persamaan diatas, dapat dilihat pengaruh langsung Eb/No terhadap SNR. Makin besar nilai Eb/No maka SNR akan semakin tinggi pula, sehingga perbandingan *noise* dengan sinyal informasi untuk kanal AWGN semakin kecil. Dengan demikian, hal ini juga berimbas pada nilai BER yang semakin kecil. BER yang kecil akan membuat *throughput* makin besar.

#### 4.2.3.2 Sinyal Konstelasi Berdasarkan kecepatan

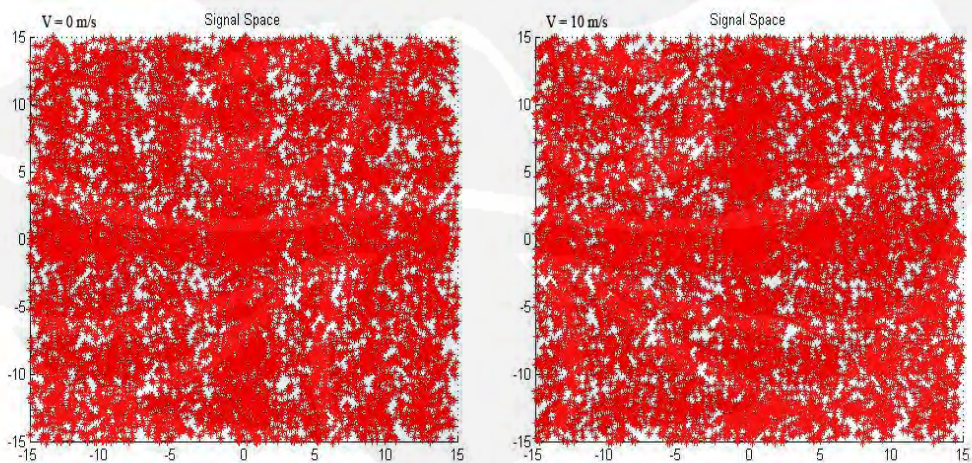
Pada bagian ini, akan dilihat sinyal konstelasi berdasarkan besarnya kecepatan *user*. Terdapat dua nilai kecepatan pada simulasi ini, yaitu v=0 (yang menandakan *user* diam) dan v=10 (*user* bergerak). Disini, untuk simulasinya digunakan nilai Eb/No = 30 dB dan teknik ECC yang digunakan adalah dengan *coderate* = 1/2.



Gambar 4.16 Sinyal konstelasi QPSK berdasarkan kecepatan



Gambar 4.17 Sinyal konstelasi 16 QAM berdasarkan kecepatan



Gambar 4.18 Sinyal konstelasi 64 QAM berdasarkan kecepatan

Sinyal konstelasi untuk modulasi QPSK, 16 QAM dan 64 QAM direpresentasikan pada gambar 4.16, 4.17, 4.18 secara berurutan. Yang pertama kita lihat adalah sinyal konstelasi untuk modulasi QPSK. Secara sekilas, tidak terlihat adanya perbedaan diantara sinyal konstelasi untuk kecepatan 0 m/s dan 10 m/s. Namun, jika dilihat lebih lanjut, akan terdapat perbedaan dari kehalusan pola yang terbentuk, yang berarti banyaknya sinyal dengan tanda (\*) yang berkumpul disekitar tanda (+). Dari gambar 4.16 dapat dilihat bahwa, untuk sistem dengan  $v=10$  m/s, sinyal konstelasinya sedikit kurang bagus. Hal ini menunjukkan adanya pengaruh kecepatan terhadap sinyal konstelasi yang dihasilkan. Hal ini lebih terlihat jelas pada gambar 4.17 dan 4.19 yang menggunakan modulasi 16 QAM dan 64 QAM.

Adanya perbedaan hasil sinyal konstelasi tersebut dikarenakan besar kecepatan yang mempengaruhi nilai pergeseran frekuensi Doppler. Kecepatan yang besar, akan membuat pergeseran frekuensi Doppler makin besar pula. Pengaruh kecepatan terhadap pergeseran frekuensi Doppler dapat dilihat pada persamaan berikut ini:

$$F_{\text{doppler}} = (v/\lambda) \cdot \cos \varphi \dots \dots \dots (4.6)$$

Dengan  $F_{\text{doppler}}$  adalah besarnya pergeseran frekuensi pembawa,  $v$  adalah kecepatan gerak relatif *mobile station* terhadap *base station*,  $\lambda$  merupakan panjang gelombang dari frekuensi pemancar dan  $\varphi$  adalah sudut datang yang dibentuk antara path tersebut dengan arah gerak *mobile station*.

Pergeseran frekuensi yang dikarenakan kecepatan tersebut, menyebabkan perbedaan frekuensi untuk mensampling data pada sisi penerima. Hal ini membuat data hasil demodulasi pada sisi penerima akan berbeda pula, yang nantinya akan menyebabkan nilai BER semakin besar.

## BAB V KESIMPULAN

Dari hasil simulasi MIMO OFDM dengan spatial multiplexing berbasis SVD, dapat ditarik beberapa kesimpulan seperti berikut:

1. Sistem MIMO OFDM dengan teknik spatial *multiplexing* mempunyai keunggulan dari segi kecepatan transfer data
2. Sistem yang dilengkapi dengan ECC mempunyai unjuk kerja yang jauh lebih baik dibanding sistem tanpa menggunakan ECC, baik dari segi BER, kapasitas kanal, maupun *throughput* sistem.
3. Sistem dengan kondisi *User* diam memiliki unjuk kerja yang sedikit lebih baik dibanding sistem dengan keadaan *user* bergerak, walaupun perbedaannya tidak terlalu signifikan.
4. Makin besar *code rate* yang digunakan untuk ECC, maka performa sistem makin baik
5. Sinyal konstelasi untuk sistem dengan menggunakan ECC lebih bagus dibanding sistem tanpa menggunakan ECC
6. Sinyal konstelasi untuk sistem dengan kondisi *user* diam lebih bagus dibanding sistem dengan *user* bergerak

## DAFTAR ACUAN

- [1] \_\_\_\_\_, "*Orthogonal Frequency Division Multiplexing (OFDM)*", Application for wireless Communication with Coding
- [2] Michael Steer, "*Beyond 3G*," IEEE Microwave Magazine No.1 (Februari, 2007).
- [3] Jaehak Chung, Yungsoo Kim, Eungsun Kim, "*Multiple Antenna System for 802.16 Systems*", IEEE 802.16 Broadband Wireless Access Working Group.2001
- [4] Abdullah Mandan. "Pemodelan Kanal MIMO OFDM Sistem Komunikasi UWB Menggunakan Singular Value Decomposition". Skripsi, Program Pasca Sarjana Fakultas Teknik UI, Depok, 2008.
- [5] Philippe Duplessis, "*Exploiting OFDM and MIMO to take UMTS beyond HSDPA / HSUPA*", Nortel Technical Journal, Issue 2, Nortel, Kanada
- [6] Allert van Zelst, "MIMO-OFDM For Wireless LAN," Disertasi Ph.D, Technische Universiteit of Eindhoven, Eindhoven, 2004.
- [7] Kamal Hamzah. "Sistem Komunikasi Nirkabel Dan Bergerak Dengan Menggunakan Teknik MIMO OFDM". Seminar, Program Sarjana Fakultas Teknik UI, Depok, 2008.
- [8] \_\_\_\_\_, "*Mengenal Teknologi Frequency Division Multiplexing (OFDM) pada Komunikasi Wireless*". Diakses 7 mei 2008 dari elektro indonesia.  
[www.elektroindonesia.com](http://www.elektroindonesia.com)
- [9] Wang, Yin, et al. "*MIMO with MUMS*". Project course in signal processing and Digital Communication. KTH, Swedia. 2004
- [10] Wahyu Mahendra. "Simulasi Sistem MIMO-STBC dengan Smart Antenna Algoritma Referensi Temporal ". Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2008.

- [11] K Sam Shanmugam, "Digital and Analog Communication Systems" (New York: John Wiley & Sons, 1979).
- [12] Gunawan Hendro. "Simulasi WCDMA dan WiMAX". Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2006
- [13] Tim C W Schenk, Guido Dolmans, Isabella Modonesi. "*Throughput of a MIMO OFDM based WLAN System*". Proc. Syposium IEEE Benelux Chapter on Communications and vehicular Technology, Belgium. 2004

## DAFTAR PUSTAKA

- Shanmugam, K. Sam, "Digital and Analog Communication Systems" New York, John Wiley & Sons, 1979.
- Wang, Yin, *et al.* "MIMO with MUMS". Project course in signal processing and Digital Communication. KTH, Swedia. 2004
- Schonemann, Nikolaus, *et al.* "SIMO OFDM System". Project course in signal processing and Digital Communication. KTH, Swedia. 2006
- Tung, Kim Thanh, *et al.* MIMO Smart Antenna over Radio. Project course in signal processing and Digital Communication. KTH, Swedia. 2003
- Jia Liu, *et al.* "OFDM Project". Project course in signal processing and Digital Communication. KTH, Swedia. 2005
- Qiu Xun, *et al.* "OFDM In Multipath". Project course in signal processing and Digital Communication. KTH, Swedia. 2007
- Huang, Jinliang, Svante Signell. "A Comparison of Two Algorithms in MIMO OFDM" Department of Microelectronics and IT Royal Institute of Technology. Swedia
- Jafarian, H Zamiri, G Gulak. "Adaptive Channel SVD Estimation for MIMO OFDM Systems". Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada
- Dung, Dao Nguyen, *et al.* "Performance Analysis of MIMO OFDM Systems using Indoor Wideband MIMO Channel Measurement Data". Tokyo Institute of Technology. 2006
- Bregman, Svante. Spatial Multiplexing Using Partial Channel. Royal Institute of Technology. Swedia, 2004
- Prasetya, Budi. "Peningkatan Kinerja Sistem Komunikasi dengan : MIMO, OFDM dan Beamforming". Kuliah Kapita Selekt, 2006
- [www.wikipedia.org](http://www.wikipedia.org)
- [www.elektroindonesia.com](http://www.elektroindonesia.com)



# LAMPIRAN

## Simulasi Parameter

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               SKRIPSI                               %
%-----%
%           SPATIAL MULTIPLEXING MIMO OFDM           %
%           DENGAN MODEL KANAL SVD                   %
%           UNTUK MOBILE WIMAX                       %
%-----%
%           I Gura Prabu Senky                       %
%           040403707X                               %
%           Departemen Teknik Elektro               %
%           Fakultas Teknik Universitas Indonesia   %
%           2008                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;
clc;

%Pilihan Modulasi dan Error Control Coding
pilihanmodulasi=input('modulasi=');
pilihanecc=input('ecc=');

output=[];

%MIMO 2 x 2
R=2;
T=2;

%parameterOFDM%
Nc = 256;           % jumlah subkanal OFDM
Si = 256;           % jumlah data
GI = Nc/4;         % panjang Guard Interval
L=16;

%channel rayleigh%
sum_alpha=0;
EbNoi=1:2:30;
v=10;%m/s

fs = 25e4;
fc = 3e4;
Fc=3.6e9;          % Frekuensi WiMAX
c=3e8;
Fdop=round((fc*v/c));

tsamp=1/(fs);
tsym=4*tsamp;
nsamp=tsym/tsamp;
aaa=10*log10(nsamp);

%persiapan masuk kanal%
ch = rayleighchan(tsamp,Fdop,1e-7*[0 0.2 0.4],[-3.98 -3.01 0]);
```

```

ch.ResetBeforeFiltering = 0;

%parameter convolutional encoding%
tb=20;
trell1 = poly2trellis([5 4],[23 35 0;0 5 13]); % Trellis
trell2 = poly2trellis(7, [171 133]); % Define trellis.
trell3 = poly2trellis(7, [171 165 133]); % Define trellis.
if pilihanecc==1
    trel = trell1;
    decdelay=2*tb;
    coderate=3/2;
elseif pilihanecc==2
    trel = trell2;
    decdelay=tb;
    coderate=2;
elseif pilihanecc==3
    trel = trell3;
    decdelay=tb;
    coderate=3;
end

%error counting%
count_error=0;

%training sequence%
Ep=2;
load('train_256.mat');
training1=ms1.*Ep;
training2=ms2.*Ep;
len_train=length(training1);

interval=1;
alpha=sqrt(0.5);

%%%%%%%%%% TRANSMITTER %%%%%%%%%%%
%Generate input
panjangblok=576;
banyakblok=128;
message=randint(1,panjangblok*banyakblok);

for nnn=1:length(EbNoi)
    EbNo=EbNoi(nnn);

%convolutional encoding%
if pilihanecc==0
    code=message;
    coderate=1;
else
    code=convenc(message,trel);
end

%interleaver message%
code1=interleaver(code);

%modulasi message%%
if pilihanmodulasi==1
    [qpsk_I,qpsk_Q]=qpsk_map(code1);

```

```

chan_I=qpsk_I;
chan_Q=qpsk_Q;
k=2;
elseif pilihanmodulasi==2
    [qam_I,qam_Q]=qam16_map(code1);
    chan_I=qam_I;
    chan_Q=qam_Q;
    k=4;
else
    [qam_I,qam_Q]=qam64_map(code1);
    chan_I=qam_I;
    chan_Q=qam_Q;
    k=4;
end

messagesiap=chan_I+j*chan_Q;
siapsig=reshape(messagesiap,512,[]);
train=[training1.';training2.'];
siapsigtrain=cat(2,train,siapsig);
[brs1,klm1]=size(siapsigtrain);

snr = EbNo + 10*log10(k*coderate)-aaa;

%%% frame 1 %%%                untuk estimasi kanal
messtca=siapsigtrain(:,1);
mesa=reshape(messtca,2,[]);
mesgrd1a=mesa(1,:);
mesgrd2a=mesa(2,:);
H=crandn(2,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% penentuan channelest
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
trainifft1a=real(mesgrd1a)';
trainifft2a=real(mesgrd2a)';
for aa=1:2
    h111a=H(aa,:);
    for ii=1:length(trainifft1a)
        ooo=[trainifft1a(ii);trainifft2a(ii)];
        ooo=awgn(ooo,snr,'measured');
        outesta(:,ii)=(h111a)*ooo;
    end
    fad1 = abs(filter(ch, ones(size(outesta))));
    fadedSig1 = fad1.*outesta;
    xxa=[trainifft1a(1:256) trainifft2a(1:256)];
    yya= fadedSig1(1:256).';
    hesta(aa,:)=(inv(xxa'*xxa))*(xxa'*yya);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Kanal SVD %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[U,S,V]=svd(hesta);

%%%frame2 dst%%%                frame data
output=[];
for ii=2:klm1
    messtc=siapsigtrain(:,ii);
    mes=reshape(messtc,2,[]);

```

```

mes1=mes(1,:);
mes2=mes(2,:);

mesifft1=ifft(mes1);
mesifft2=ifft(mes2);

mesgrd1=[mesifft1(Nc-GI+1:Nc) mesifft1];
mesgrd2=[mesifft2(Nc-GI+1:Nc) mesifft2];

% Follow with rectangular pulse shaping (Upsample) %%%%.
mesgrd1 = rectpulse(mesgrd1,4);
mesgrd2 = rectpulse(mesgrd2,4);

% Up-conversion
mesgrd1up = upconv(mesgrd1,fc,fs);
mesgrd2up = upconv(mesgrd2,fc,fs);

mesgrd=[mesgrd1up;mesgrd2up];

%%%%%%%%% PREFILTER WITH V %%%%%%%%%%
mesgrdd=V*inv(S)*mesgrd;
mesgrdd1=mesgrdd(:);

%%%%%%%%% KANAL %%%%%%%%%%
mesgrddd=awgn(mesgrdd1,snr,'measured');
mesgrddd1=reshape(mesgrddd,2,[]);
kan=H*mesgrddd1;
kan1=kan(:);
fad = abs(filter(ch, ones(size(kan1)))));
fadedSig = fad.*kan1;
kan2=reshape(fadedSig,2,[]);

%%%%%%%%% RECEIVER %%%%%%%%%%
%%%%%%%%% POST FILTER WITH U' %%%%%%%%%%
rec=U'*kan2;

rx1=rec(1,:);
rx2=rec(2,:);

%%down conversion
rx1down = downconv(rx1,fc,fs);
rx2down = downconv(rx2,fc,fs);

rx1 = intdump(rx1down,4);
rx2 = intdump(rx2down,4);

%%%%%%%%% Guard Interval Removal %%%%%%%%%%
rexx1=rx1(GI+1:end);
rexx2=rx2(GI+1:end);

%%%%%%%%% F F T %%%%%%%%%%
rexxft1=fft(rexx1);
rexxft2=fft(rexx2);

rec=[rexxft1;rexxft2];

```

```

out=rec(:);
output=[output;out];

kapkanal(ii)=kapasitaskanal1(snr,T,S,hesta);
end
buff_I=real(output);
buff_Q=imag(output);

%%%%%%%%%%%%% DEMODULASI %%%%%%%%%%%%%%
if pilihanmodulasi==1
    %signal_qpsk(buff_I,buff_Q);
    outputakh1=detector_qpsk(buff_I,buff_Q);
    outputakh1=outputakh1';
elseif pilihanmodulasi==2
    %signal_space(buff_I,buff_Q,1);
    outputakh1=detector_16qam(buff_I,buff_Q,1);
    outputakh1=outputakh1';
else
    %signal3_space(buff_I,buff_Q,1);
    outputakh1=detector_64qam(buff_I,buff_Q,1);
    outputakh1=outputakh1';
end

%%%%%%%%%%%%% deinterleaver %%%%%%%%%%%%%%
outputt=deinterleaver(outputakh1);

%%%%%%%%%%%%% convolutional decoding %%%%%%%%%%%%%%
if pilihanecc==0
    decoded=outputt;
    decoded1=decoded(:);
else
    decoded = vitdec(outputt,trel,tb,'cont','hard'); % Decode.
    decoded1 = decoded(decdelay+1:end);
    decoded1=decoded1(:);
    mestam=message((length(message)-decdelay+1):end);
    decoded1 =[decoded1;mestam(:)];
end

%%%%%%%%%%%%% menghitung BER %%%%%%%%%%%%%%

[num_bern(1,nnnn)]=biterr(message,decoded1')

%menghitung simbol error
messagesym=reshape(message,[],k*Si);
decoded1sym=reshape(decoded1,[],k*Si);
frames=bi2de(messagesym);
framedecoded1=bi2de(decoded1sym);
[num_symer(1,nnnn)]=symerr(frames,framedecoded1)

%menghitung kapasitas kanal
totkap=(1/klm1)*sum(kapkanal);
chancap(1,nnnn)=real(totkap)

%menghitung throughput
datarate=Si*k*(1/coderate)*(1/tsym);
throughput(1,nnnn)=datarate*(1-symer(1,nnnn))

```

```

%kosongkan semua variabel
decoded1=[];
outputakhir=[];
a6=[];
H=[];
hest=[];
snr=[];
out=[];
hasil1=[];
hasil2=[];
end

%%%%%%%%%% plot grafik %%%%%%%%%%%
figure(1);
semilogy(EbNoi,bern(1,:), '-k*');
xlabel('Eb/No (dB) ');
ylabel('BER');
title('BER Vs Eb/No ');

figure(2);
plot(EbNoi, chancap(1,:), '-k*');
xlabel('Eb/No (dB) ');
ylabel('channel capacity (b/s/Hz)');
title('channel capacity Vs Eb/No ');

figure(3);
throughput=throughput./1e6;
plot(EbNoi,throughput(1,:), '-k*');
xlabel('Eb/No (dB) ');
ylabel('throughput (Mbps)');
title('throughput Vs Eb/No ');

```

## Simulasi Signal Space

```

clear all;
close all;
clc;

%Pilihan Modulasi dan Error Control Coding
pilihanmodulasi=input('modulasi=');
pilihanecc=input('ecc=');
output=[];

%MIMO 2 x 2
R=2;
T=2;

%parameterOFDM%
Nc = 256;           % jumlah subkanal
S = Nc-P;          % jumlah data
GI = Nc/4;         % panjang guard interval
L=16;

%channel rayleigh%
sum_alpha=0;

```

```

EbNo=30;
v=10;%m/s

fs = 96e3;
fc = 10e3;
Fc=3.6e9;           % Frekuensi WiMAX
c=3e8;
Fdop=round((fc*v*cos(pi/6)/c));

tsamp=1/(fs);
tsym=4*tsamp;
nsamp=tsym/tsamp;
aaa=10*log10(nsamp);

%persiapan masuk kanal%
ch = rayleighchan(tsamp,Fdop,1e-7*[0 0.2 0.4],[-3.98 -3.01 0]);
ch.ResetBeforeFiltering = 0;

%parameter convolutional encoding%
tb=20;
trell1 = poly2trellis([5 4],[23 35 0;0 5 13]); % Trellis
trell2 = poly2trellis(7, [171 133]); % Define trellis.
trell3 = poly2trellis(7, [171 165 133]); % Define trellis.
if pilihanecc==1
    trel = trell1;
    decdelay=2*tb;
    coderate=3/2;
elseif pilihanecc==2
    trel = trell2;
    decdelay=tb;
    coderate=2;
elseif pilihanecc==3
    trel = trell3;
    decdelay=tb;
    coderate=3;
end

%error counting%
count_error=0;

%training sequence%
Ep=2;
load('train_256.mat');
training1=ms1.*Ep;
training2=ms2.*Ep;
len_train=length(training1);

%feedback every frame for adptive modulation as well as weighting
interval=1;
alpha=sqrt(0.5);

% TRANSMITTER %%
%Generate input
panjangblok=240;
banyakblok=128;
message=randint(1,panjangblok*banyakblok);
%convolutional encoding%

```

```

if pilihanecc==0
    code=message;
    coderate=1;
else
    code=convenc(message,trel);
end

%interleaver message%
code1=interleaver(code);

%modulasi message%%%%
if pilihanmodulasi==1
    [qpsk_I,qpsk_Q]=qpsk_map(code1);
    chan_I=qpsk_I;
    chan_Q=qpsk_Q;
    k=2;
elseif pilihanmodulasi==2
    [qam_I,qam_Q]=qam16_map(code1);
    chan_I=qam_I;
    chan_Q=qam_Q;
    k=4;
else
    [qam_I,qam_Q]=qam64_map(code1);
    chan_I=qam_I;
    chan_Q=qam_Q;
    k=6;
end

messagesiap=chan_I+j*chan_Q;
siapsig=reshape(messagesiap,512,[]);
train=[training1.';training2.'];
siapsigtrain=cat(2,train,siapsig);
[brs1,klm1]=size(siapsigtrain);

snr = EbNo + 10*log10(k*coderate)-aaa;

%%frame 1%%
messtca=siapsigtrain(:,1);
mesa=reshape(messtca,2,[]);
mesgrd1a=mesa(1,:);
mesgrd2a=mesa(2,:);
H=crandn(2,2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% penentuan channelest
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
trainifft1a=real(mesgrd1a)';
trainifft2a=real(mesgrd2a)';
for aa=1:2
    h111a=H(aa,:);
for ii=1:length(trainifft1a)
    ooo=[trainifft1a(ii);trainifft2a(ii)];
    ooo=awgn(ooo,snr,'measured');
    outesta(:,ii)=(h111a)*ooo;
end
fad1 = abs(filter(ch, ones(size(outesta))));
fadedSig1 = fad1.*outesta;
xxa=[trainifft1a(1:256) trainifft2a(1:256)];
yya= fadedSig1(1:256).';

```



```

hesta(aa, :)=(inv(xxa'*xxa))*(xxa'*yya);
end
[U,S,V]=svd(hesta);

%%%frame2 dst%%%
output=[];
for ii=2:klm1
messtc=siapsigtrain(:,ii);
mes=reshape(messtc,2,[]);
mes1=mes(1,:);
mes2=mes(2,:);

mesifft1=ifft(mes1);
mesifft2=ifft(mes2);

mesgrd1=[mesifft1(Nc-GI+1:Nc) mesifft1];
mesgrd2=[mesifft2(Nc-GI+1:Nc) mesifft2];

% Follow with rectangular pulse shaping (Upsample) %%%%.
mesgrd1 = rectpulse(mesgrd1,4);
mesgrd2 = rectpulse(mesgrd2,4);

% Up-conversion
mesgrd1up = upconv(mesgrd1,fc,fs);
mesgrd2up = upconv(mesgrd2,fc,fs);

mesgrd=[mesgrd1up;mesgrd2up];

%%%%%%%%%% PREFILTER WITH V %%%%%%%%%%%
mesgrdd=V*inv(S)*mesgrd;
mesgrdd1=mesgrdd(:);

%%%%%%%%%% KANAL %%%%%%%%%%%
mesgrddd=awgn(mesgrdd1,snr,'measured');
mesgrddd1=reshape(mesgrddd,2,[]);
kan=H*mesgrddd1;
kan1=kan(:);
fad = abs(filter(ch, ones(size(kan1)))));
fadedSig = fad.*kan1;
kan2=reshape(fadedSig,2,[]);

%%%%%%%%%% RECEIVER %%%%%%%%%%%
%%%%%%%%%% POST FILTER WITH U' %%%%%%%%%%%
rec=U'*kan2;

rx1=rec(1,:);
rx2=rec(2,:);

%%down conversion
rx1down = downconv(rx1,fc,fs);
rx2down = downconv(rx2,fc,fs);

rx1 = intdump(rx1down,4);
rx2 = intdump(rx2down,4);

%%%%%%%%% Guard Interval Removal %%%%%%%%%%

```

```

rexx1=rx1 (GI+1:end);
rexx2=rx2 (GI+1:end);

%%%%%%%%%% F F T %%%%%%%%%%%
rexfft1=fft (rexx1);
rexfft2=fft (rexx2);

rec=[rexfft1;rexfft2];

out=rec (:);
output=[output;out];

kapkanal (ii)=kapasitaskanal1 (snr, T, S, hesta);
end
buff_I=real (output);
buff_Q=imag (output);

%%%%%%%%%% DEMODULASI %%%%%%%%%%%
if pilihanmodulasi==1
    signal_qpsk (buff_I,buff_Q);
    outputakh1=detector_qpsk (buff_I,buff_Q);
    outputakh1=outputakh1';
elseif pilihanmodulasi==2
    signal_space (buff_I,buff_Q,1);
    outputakh1=detector_16qam (buff_I,buff_Q,1);
    outputakh1=outputakh1';
else
    signal3_space (buff_I,buff_Q,1);
    outputakh1=detector_64qam (buff_I,buff_Q,1);
    outputakh1=outputakh1';
end

%%%%%%%%%% deinterleaver %%%%%%%%%%%
outputt=deinterleaver (outputakh1);

%%%%%%%%%% convolutional decoding %%%%%%%%%%%
if pilihanecc==0
    decoded=outputt;
    decoded1=decoded (:);
else
    decoded = vitdec (outputt,trel,tb, 'cont', 'hard'); % Decode.
    decoded1 = decoded (decdelay+1:end);
    decoded1=decoded1 (:);
    mestam=message ((length (message)-decdelay+1):end);
    decoded1 =[decoded1;mestam (:)];
end

%%%%%%%%%% menghitung BER %%%%%%%%%%%
a6=length (decoded1);
for k = 1:a6
    if (message (k) ~=decoded1 (k))
        count_error = count_error+1;
    end
end
BER=count_error/a6

%menghitung kapasitas kanal

```

```
totkap=(1/klm1)*sum(kapkanal);
a=real(max(kapkanal))
```

## Interleaver

```
function [codenn]=interleaver(source)
%interleaver
coden=reshape(source,8,[]);
coden1=coden(1,:);
coden1=coden1(:);
coden2=coden(2,:);
coden2=coden2(:);
coden3=coden(3,:);
coden3=coden3(:);
coden4=coden(4,:);
coden4=coden4(:);
coden5=coden(5,:);
coden5=coden5(:);
coden6=coden(6,:);
coden6=coden6(:);
coden7=coden(7,:);
coden7=coden7(:);
coden8=coden(8,:);
coden8=coden8(:);
codenn=[coden1;coden2;coden3;coden4;coden5;coden6;coden7;coden8];
end
```

## Modulasi

- QPSK

```
function [Ich,Qch]=qpsk_map(bit_source)
% 10 | 00
% ---|--->
% 11 | 01

len=length(bit_source);
block=floor(len/2);

for i=1:block
    if(bit_source(2*i-1)==0 & bit_source(2*i)==0)
        qam_symbol(i)=1+j;
    elseif(bit_source(2*i-1)==0 & bit_source(2*i)==1)
        qam_symbol(i)=1-j;
    elseif(bit_source(2*i-1)==1 & bit_source(2*i)==0)
        qam_symbol(i)=-1+j;
    elseif(bit_source(2*i-1)==1 & bit_source(2*i)==1)
        qam_symbol(i)=-1-j;
    end
end
Ich=real(qam_symbol)/sqrt(2);
Qch=imag(qam_symbol)/sqrt(2);
```

- 16-QAM

```
function [qam_I,qam_Q]=qam16_map(bit_source)
```

```

%Algorithm:
% divide the bits into blocks of 4 bits each
%
% Gray mapping
% 0010 0110 1110 1010
% 0011 0111 1111 1011
% 0001 0101 1101 1001
% 0000 0100 1100 1000
%
% like(0010), every two bits map to a symbol in{1,-1,3,-3}
% 00    -3
% 01    -1
% 11     1
% 10     3
%
% the first 2 bits in block corresponds to the In_phase part and
% last 2 bits corresponds to the Quadrature-phase part.
%
% normalize
% ((1^2+1^2)+(1^2+3^2)+(1^2+3^2)+(3^2+3^2))/4=10
% (1+3*j)/sqrt(10)
% by normalizing, the BER deteriorate 11 dB
%
% Output:
% qam_I+j*qam_Q is the symbol we derive after mapping
%

len=length(bit_source);
block=floor(len/4);

bit_source=bit_source(1:4*block);%discard the left bits
bit_matrix=reshape(bit_source,2,[]);

for i=1:block*2
    b(:,i)=bit_matrix(:,i);

    if(b(:,i)==[0;0])
        qam_symbol(i)=-3;
    elseif(b(:,i)==[0;1])
        qam_symbol(i)=-1;
    elseif(b(:,i)==[1;1])
        qam_symbol(i)=1;
    elseif(b(:,i)==[1;0])
        qam_symbol(i)=3;
    end
end
qam_I=qam_symbol(1:2:end)/sqrt(10);
qam_Q=qam_symbol(2:2:end)/sqrt(10);

```

- **64-QAM**

```

function b_hat=detector_64qam(I,Q,power_h);
len=length(I);
b_hat=zeros(1,6*len);
aa=( (1^2+1^2)+(5^2+5^2)+(9^2+9^2)+(13^2+13^2)+2*(1^2+5^2)+2*(9^2+5^2)+2*(13^2+9^2)+2*(9^2+1^2)+2*(13^2+5^2)+2*(13^2+1^2) )/16;
symbol_power=aa;
i=1;

```

```

for k=1:len
    if (I(k)>=0)
        b_hat(i)=1;
        if (I(k)*sqrt(symbol_power)<=7*power_h)
            b_hat(i+1)=1;
            if (I(k)*sqrt(symbol_power)<=3*power_h)
                b_hat(i+2)=1;
            end
        else
            if (I(k)*sqrt(symbol_power)<=11*power_h)
                b_hat(i+2)=1;
            end
        end
    else
        if (I(k)*sqrt(symbol_power)>-7*power_h)
            b_hat(i+1)=1;
            if (I(k)*sqrt(symbol_power)>-3*power_h)
                b_hat(i+2)=1;
            end
        else
            if (I(k)*sqrt(symbol_power)>(-11)*power_h)
                b_hat(i+2)=1;
            end
        end
    end
end
if (Q(k)>=0)
    b_hat(i+3)=1;
    if (Q(k)*sqrt(symbol_power)<=7*power_h)
        b_hat(i+4)=1;
        if (Q(k)*sqrt(symbol_power)<=3*power_h)
            b_hat(i+5)=1;
        end
    else
        if (Q(k)*sqrt(symbol_power)<=11*power_h)
            b_hat(i+5)=1;
        end
    end
end
else
    if (Q(k)*sqrt(symbol_power)>-7*power_h)
        b_hat(i+4)=1;
        if (Q(k)*sqrt(symbol_power)>-3*power_h)
            b_hat(i+5)=1;
        end
    else
        if (Q(k)*sqrt(symbol_power)>(-11)*power_h)
            b_hat(i+5)=1;
        end
    end
end
end
i=i+6;
end

```

## Upconversion

```

function out=upconv(in,fc,fs)
% fc : Carrier frequency ,fs : Sampling frequency

t=0:length(in)-1;

```

```
cos_table=cos(2*pi*(fc/fs)*t);
out=in.*cos_table;
```

## Downconversion

```
function out = downconv(in,fc,fs)
% fc : Carrier frequency ,fs : Sampling frequency

t=0:length(in)-1;

cos_table=2*cos(2*pi*(fc/fs)*t);
out=in.*cos_table;
```

## Kapasitas Kanal SVD

```
function C=kapasitaskanal(SNR,T,S,H)

SNR = 10 ^ (SNR/10);
for ai=1:rank(H)
ddd=(1+(SNR/T)*S(ai,ai)^2);
%%setiap subkanal diasumsikan memiliki matriks kanal yang sama
pada setiap
%%transmsisi
C(ai)=log2(ddd);
end
C=sum(C);
```

## Signal Space

- **QPSK**

```
function signal_qpsk(Ich,Qch)
hold on

len_data=length(Ich);
correct_sig_x=[-1,-1,1,1];
correct_sig_y=[-1,1,-1,1];

x=Ich*sqrt(2);
y=Qch*sqrt(2);

axis([-6 6 -6 6])

plot(x,y,'*g')
plot(correct_sig_x,correct_sig_y,'mx')
title('Signal Space')
grid on
drawnow
```

- **16-QAM**

```
function signal_space(Ich,Qch,power_h)
hold on

len_data=length(Ich);
correct_sig_x=[-3,-3,-3,-3,-1,-1,-1,-1,1,1,1,1,3,3,3,3];
correct_sig_y=[-3,-1,1,3,-3,-1,1,3,-3,-1,1,3,-3,-1,1,3];
```

```

x=Ich*sqrt(10);
y=Qch*sqrt(10);

axis([-6 6 -6 6])

plot(x/power_h,y/power_h,'*b')
plot(correct_sig_x,correct_sig_y,'r+')
title('Signal Space')
grid on
drawnow

```

- **64-QAM**

```

function signal3_space(Ich,Qch,power_h)
hold on

len_data=length(Ich);
correct_sig_x=[-13,-13,-13,-13,-13,-13,-13,-13,-9,-9,-9,-9,-9,-9,-9,-9,-5,-5,-5,-5,-5,-5,-5,-5,-5,-1,-1,-1,-1,-1,-1,-1,-1,-1,...
9,-9,-5,-5,-5,-5,-5,-5,-5,-5,-1,-1,-1,-1,-1,-1,-1,-1,-1,...

1,1,1,1,1,1,1,1,5,5,5,5,5,5,5,5,9,9,9,9,9,9,9,9,13,13,13,13,13,13,
13,13];
correct_sig_y=[-13,-9,-5,-1,1,5,9,13,-13,-9,-5,-1,1,5,9,13,-13,-9,-5,-1,1,5,9,13,-13,-9,-5,-1,1,5,9,13,...
9,-5,-1,1,5,9,13,-13,-9,-5,-1,1,5,9,13,-13,-9,-5,-1,1,5,9,13,...
-13,-9,-5,-1,1,5,9,13,-13,-9,-5,-1,1,5,9,13,-13,-9,-5,-1,1,5,9,13,-13,-9,-5,-1,1,5,9,13];

x=Ich*sqrt(138);
y=Qch*sqrt(138);

axis([-15 15 -15 15])

plot(x/power_h,y/power_h,'*r')
plot(correct_sig_x,correct_sig_y,'r+')
title('Signal Space')
grid on
drawnow

```

## DEMODULASI

- **QPSK**

```

function b_hat=detector_qpsk(I,Q);
% 10 | 00
% ---|--->
% 11 | 01
len=length(I);
b_hat=zeros(1,2*len);

for k=1:len
    if(I(k)<=0)
        b_hat(2*k-1)=1;
    end
    if(Q(k)<=0)
        b_hat(2*k)=1;
    end
end

```

```

end
end

```

### • 16-QAM

```

function b_hat=detector_16qam(I,Q,power_h);
% Gray mapping
% 0010 0110 1110 1010
% 0011 0111 1111 1011
% 0001 0101 1101 1001
% 0000 0100 1100 1000
len=length(I);
b_hat=zeros(1,4*len);
symbol_power=((1^2+1^2)+(1^2+3^2)+(1^2+3^2)+(3^2+3^2))/4;

i=1;
for k=1:len
    if(I(k)>=0)
        b_hat(i)=1;
        if(I(k)*sqrt(symbol_power)<=2*power_h)
            b_hat(i+1)=1;
        end
    else
        if(I(k)*sqrt(symbol_power)>-2*power_h)
            b_hat(i+1)=1;
        end
    end
    if(Q(k)>=0)
        b_hat(i+2)=1;
        if(Q(k)*sqrt(symbol_power)<2*power_h)
            b_hat(i+3)=1;
        end
    else
        if(Q(k)*sqrt(symbol_power)>-2*power_h)
            b_hat(i+3)=1;
        end
    end
    i=i+4;
end

```

### • 64-QAM

```

function [qam_I,qam_Q]=qam64_map(bit_source)
len=length(bit_source);
block=floor(len/6);

aa=((1^2+1^2)+(5^2+5^2)+(9^2+9^2)+(13^2+13^2)+2*(1^2+5^2)+2*(9^2+5^2)+2*(13^2+9^2)+2*(9^2+1^2)+2*(13^2+5^2)+2*(13^2+1^2))/16;

bit_source=bit_source(1:6*block);
bit_matrix=reshape(bit_source,3,[]);

for i=1:block*2
    b(:,i)=bit_matrix(:,i);
    if(b(:,i)==[0;0;0])
        qam_symbol(i)=-13;
    elseif(b(:,i)==[0;0;1])
        qam_symbol(i)=-9;
    elseif(b(:,i)==[0;1;0])
        qam_symbol(i)=-5;
    end
end

```



```

elseif (b(:,i)==[0;1;1])
    qam_symbol(i)=-1;
elseif (b(:,i)==[1;1;1])
    qam_symbol(i)=1;
elseif (b(:,i)==[1;1;0])
    qam_symbol(i)=5;
elseif (b(:,i)==[1;0;1])
    qam_symbol(i)=9;
elseif (b(:,i)==[1;0;0])
    qam_symbol(i)=13;
end
end
qam_I=qam_symbol(1:2:end)/sqrt(aa);
qam_Q=qam_symbol(2:2:end)/sqrt(aa);

```

## Deinterleaver

```

function [codenn]=deinterleaver(source)
%deinterleaver
zzz=source;
z1=reshape(zzz,[],8);
z2=z1';
codenn=z2(:);
end

```